NTNU

Norwegian University of
Science and Technology

Enhancing reflection by collaboratively capturing experiences in a timeline

Anders Klund Kristiansen
Andreas Storlien

Master of Science in Informatics
Submission date: June 2011
Supervisor: Monica Divitini, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

Experience is the foundation for learning. Through reflection, users can achieve new learning outcomes based on the learning experience they are involved in. Boud et.al defines reflection as a process where the experience is revisited, feelings are re-attended and the experience is re-evaluated [7]. This process can be supported by technology in several ways. Previous work using timelines as a basis for collaborative work has shown to support reflective learning and knowledge construction. In for instance [30], timelines have been used to create shared environments for reflective learning.

The main objective of this thesis was to develop a tool to enhance reflection by supporting collaborative capturing of learning experiences in a timeline.

The tool was developed following an iterative user-centered approach. The tool is grounded in theory of reflection and experience based learning. Design choices have been informed by the state of the art in tools supporting reflection and feedback from evaluations. An intermediate evaluation was conducted with an expert in the field of reflection in learning. The final evaluation was performed at the general assembly of a EU-project working in the field of reflection and learning. 25 users used the tool over the course of 3 days.

The work resulted in an Android application where users can collaboratively construct timelines by annotating it with notes, pictures, videos and sound. The tool encourages users to interact and reflect by assessing and contributing to each others annotations. By integrating the tool with other applications for reflection, we showed how several sources of data can enhance reflection on learning experiences.

The main contributions of this thesis is a set of characteristics identified to be important for enhancing reflection when capturing collaborative learning experiences in a timeline.

Preface

This is the Master's thesis of IT3900 - Informatics Postgraduate Thesis, written by Andreas Storlien and Anders Kristiansen from August 2010 to June 2011 at the Norwegian University of Science and Technology(NTNU). The Master's thesis is the final delivery of the master programme in Informatics, study path System Development and Human-Computer-Interaction.

We would like to thank our supervisor, Monica Divitini, for valuable feedback and advices throughout the course of this thesis. We appreciate the inspiring opportunity you gave us by letting us go to Germany.

We would also like to thank Birgit R. Krogstie and Simone Mora for great support. At last we would like to thank the participants of the MIRROR Project general assembly that patiently helped us evaluate our tool.

Trondheim, June 1, 2011

Andreas Storlien

Anders Kristiansen

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

In this thesis, a design project will be presented where the goal was to design a mobile tool for use as support for learning and reflection in collaborative learning environments. The purpose of the tool is to make it easier for the users to collect and remember their learning experiences, connect feelings with the experiences and represent them in an abstract, but know concept - a timeline. All in a collaborative setting. The challenge will be to capture the learning and help the learner understand what they have experienced. In other words, help them reflect on the learning situation. Learning is largely unplanned and can evolve in many different directions, and this makes it important to facilitate reflection to support learning. As Boud et.al [7] presents it; reflection is a process where the experience is revisited, feelings are re-attended and the experience is re-evaluated.

1.1 Context and Background

The use of day to day mobile technology has for some years been a natural part of many peoples' lives. We take for granted that a high tech gadget with Internet, camera, phone, GPS and video possibilities is something we carry around everywhere in our pockets. We take notes, plan our day and record our thoughts using these devices, and later we share the content with our friends and family.

When we say mobile technology, the most common references are Personal digital assistants (PDAs), mobile phones/smartphones and tablets (like the iPad from Apple), that is personal and portable. All these devices can con-

nect to a large variety of information sources and enable us to communicate with each other anytime and anywhere. These possibilities are utilized most commonly in our social life and in correlation with our work life. Integration with social media and electronic mail systems have become an important factor for newer technologies, and with these systems, new requirements to hardware have become essential. Many of the new mobile devices now comes with large high resolution touch screens that enables the user to more easily interact with the content. Also, cellular wireless standards like 3G and 4G enables these devices to be connected to the Internet at all times, which enables a wave of possible uses. Still, most of the development is for personal, social or work related usage [32].

For educational purposes however, there is still a large potential in using mobile technologies. For several years, these device have more or less been excluded from schools beneath college and university level, when they at the same time have become a natural part of our day to day life. Mobility in learning is not only desired because it physically activates the learner, but also because it enables in situ (in context), and anytime, anywhere learning [18]. This kind of learning, where the user can engage in creative, constructive and generative activities has proved itself to be extremely educational because it allows the learners "to synthesize the information as well as provoke further investigation" [18]. By supporting content creation in situ, mobile systems can help learners to interact with the surroundings where the learning takes place, and this can further enhance the amount of knowledge gained.

Rich social interactions, context awareness and Internet connectivity makes embedded, ubiquitous and networked mobile technology a distinct candidate for further enhancing the learning experience in schools and workplace. As learning moves more and more out of the traditional settings, such technologies can have a large impact on learning.

1.1.1 Domain

Figure 1.1 shows the domains in which the work in this thesis will be performed. The main focus in the thesis will be in the intersection between collaboration technology, mobile technology and technology enhanced learning.

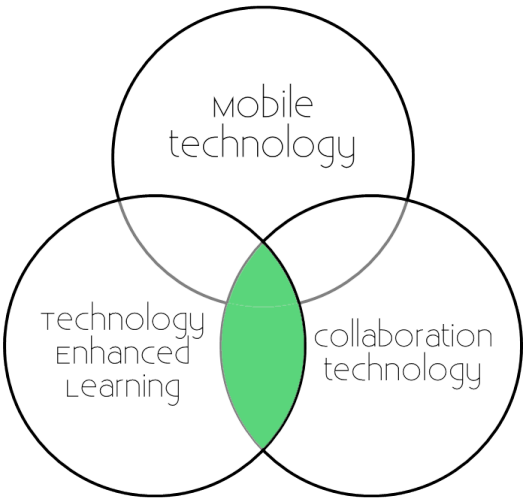


Figure 1.1: The research domain

Collaboration Technology

Collaboration technology (or collaboration software) is a wide area of research. However, research within two areas of collaboration technology stands out, namely computer supported collaborative work (CSCW) and computer supported collaborative learning (CSCL).

CSCW emerged in the 1980s as an effort by technologists to learn from economists, social psychologists, anthropologists, organizational theorists, educators, and anyone else who could shed light on group activity [21]. The drive of CSCW should be directed towards designing systems embodying an even deeper understanding of the nature of cooperative forms and practices [43]. CSCL is a pedagogical approach where learning is defined to take place via social interaction using technology. This kind of learning is characterized by the sharing and construction of knowledge among the participants using technology as their primary source of cooperation [46].

The tool we have developed belongs to the domain of collaboration technology as it offers the users the opportunity to participate in group based learning activities. These activities has not been defined to be in the area of either CSCW or CSCL, but rather both, as the tool is general enough to support different usage in different environments.

Mobile Technology

Due to the ever increasing amount of users of mobile technology, the field of mobile technology is wide, and in steady growth. We will contribute to this area by developing an application for the Android operating system [1]. Android is a software stack for mobile devices that includes operating system, middleware and key applications developed by the Open Handset Alliance [2] together with the owner of Android, Google. This means that the tool we has developed is available for all Android devices, which almost exclusively consists of mobile phones and tablets.

Technology Enhanced Learning

Technology enhanced learning (TEL) refers to the supporting of any learning activity through technology. Hannafin et.al [22] uses the term "Technology-enhanced, student centered learning environments" to describe the use of TEL in a learning environment with students. These environments use technology to enable methods for students to engage in learning activities that "enables individuals to address unique learning interests and needs, study multiple levels of complexity, and deepen understanding." The technology is supposed to enrich thinking and learning, as well as supporting reflection and collaboration in the learning environments. The goal is to "provide socio-technical innovations for learning practices, regarding individuals or organizations, independent of time, place and pace."

A part of TEL that will be central during this thesis is "mobile learning". O' Malley et.al [39] defines Mobile learning as "Any sort of learning that happens when the learner is not at a fixed, predetermined location, or learning that happens when the learner takes advantage of the learning opportunities offered by mobile technology". The goal of our tool is to support mobile learning both by offering possibilities to utilize its functionality in situ and providing uses that can help the user both learn from and reflect on the work being performed.

[1] `http://www.android.com/`
[2] `http://www.openhandsetalliance.com/`

1.1.2 Core Concepts

In this section we will try to illuminate some of the concepts that is important for the work described in this thesis. First of all, we have the concept of a *learning experience*, which is a part of the process to produce a learning outcome. The *timeline* is used as a representation of the learning experience, and a part of this thesis is to find out if this is a good representation or not. At last, we will explain the model for *reflection* much of the work in this thesis relies on.

Learning Experience

The process of how an experience evolve over time has by Anselm Strauss [48] been defined as a trajectory. In this definition the trajectory is described as "the course of how any experienced phenomenon as it evolves over time" and "the actions and interactions contributing to its evolution". Based on this we define a learning experience as an experience that over time leads to a learning outcome. According to David Kolb et.al [29], for a learning experience to occur there must exists certain abilities in the learner. First, the learning must be willing to actively be involved in the experience. Second, the learner must be able to reflect on the experience. Third, the learner must possess and be able to use analytical skills to conceptualize the experience. And fourth, the learner must posses decision making and problem solving skills to be able to create new knowledge outcomes based on the experience.

In this thesis we want to support the first two of these attributes by giving the users a way to involve themselves in and build their learning experiences, as well as providing ways for the users to reflect on this learning experience.

Timeline

A timeline is often used as a project artifact. The presentation is most of the time a long line or bar labeled with dates and events connected to these dates. It is often used to show happenings along a period of time. The timeline as a metaphor is easily grasped by people and most people know about the concept and can explain a timeline. It also maps well to the concept of a learning experience by using time as a factor to show how the process of creating a learning experience evolves over time.

The tool we are developing use the timeline as a representation for a learning

experience over time. We support the users with collecting their learning experiences and organizing the data collected in a timeline where each item is placed at its time of creation. The timeline will then also function as an abstraction of the concept of a learning trajectory presented in Section 1.1.2, describing the process of how an experience came to be.

Reflection

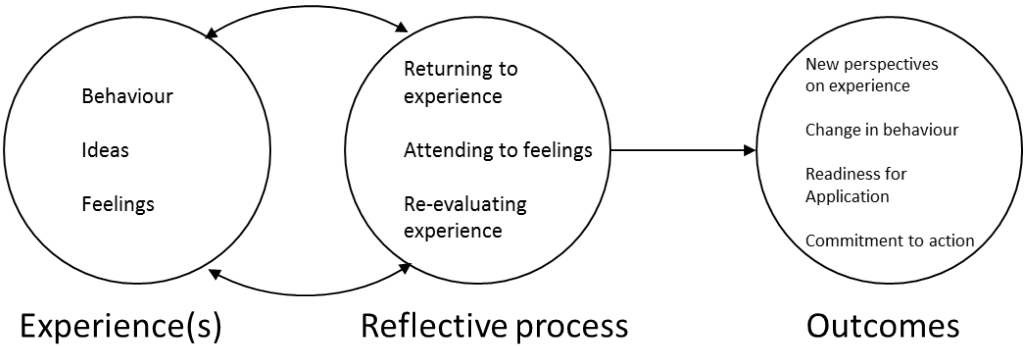


Figure 1.2: A model of the reflective process. From Boud et.al. 1985 [7]

In the model presented in Figure 1.2, Boud et.al [7] presents reflection as a process with several steps without any fixed sequence. The major theme in the model is that experience is what is reflected upon, including the behavior, feelings and ideas connected to that experience. In the reflective process you return to the experience, attend to connected feelings and re-evaluate the experience. In the model, an iteration between the experience and reflections is indicated, meaning that the reflective process and the experience can intervene before the outcome becomes clear. The outcome then consists of new perspectives, changes in the behavior, readiness for application and commitment to action. All the steps of the process does not need to be included to achieve an instance of reflection, but they are guidelines for understanding the reflective process.

In this thesis we utilize this model by trying to capture the events in the experience, as a snapshot in the timeline. This will make it easier to return to the experience and reflect. The challenge will be to capture the ideas, behavior and feelings that is involved in the experience and using technology to support the reflective process.

1.2 Research Questions

1.2.1 Main Research Question

- **How to enhance reflection by capturing collaborative learning experiences in a timeline?**

To answer the main research question we developed a tool for collaborative reflection on the Android platform. The refinement of the main research question into a set of research questions guiding the investigation of the problem domain was done by developing and evaluating this tool.

Refinement of Research Questions

- Sub RQ1: What functionality should a mobile collaborative tool support to help users share their learning experiences?

With this research question we wanted to find out if the functionality we developed for the tool were sufficient for supporting shared learning experiences. To answer this, we developed the tool with several functionality that, based on research in theory and related work, should help the users achieve this goal. These functionality includes sharing information between devices, creating collaborative environments that the users can take a part of (groups) and organization of this data for the users.

- Sub RQ2: How does a timeline work as a representation of a learning experience on a mobile device?

To represent the learning experience of a user we choose to visualize it as a timeline. In our tool we created a user interface that presents all data back to the user in a timeline based view with much resemblance to a calendar, with different levels of granularity. We want to find out if this representation works, and how the user can benefit from having their experiences collected in a timeline.

- Sub RQ3: How should data be presented to enhance learning by reflection?

In our tool, we allow the user to collect several types of data. How this is data is presented back to the user is important for several reasons. Usability is a big issue as well as how this data is perceived by the users. With our tool, we wanted to find out if the presentation of the data to the user could

work in enhancing learning by reflection aside from the actual content of the data. How can the presentation of the data help trigger a reflective process?

- Sub RQ4: What kind of user interactions should the tool support to trigger reflective processes?

Different user interaction can foster reflection in different ways. We added some functionality that we, during the evaluation, wanted to find out if could help trigger reflection amongst the users.

1.3 Research Approach

The research approach adopted in this thesis is the result of working with a design-oriented project. The background of our project is grounded in the area of reflection and learning theory and it provides as result a tool for supporting reflection. This tool is evaluated based on our research questions.

1.3.1 Research Context

FABULA

The vision of FABULA is a city that, with the help of seamless networks, becomes a learning arena for its students, with services that allow people not only to access learning material anytime and anywhere, but also, and most importantly, to take an active role in collaborative processes of knowledge construction and sharing [15]. The focus in FABULA is on learning experiences outside the classroom, situated in a city where learning comes from exploration, interaction and serendipity [16].

The principal objective of FABULA is to develop novel principles and technical solutions for learning enabled by seamless roaming in mobile networks, with focus on services that foster the city learning geographies and ecology's and enable new relationships among learners and communities [16].

The work carried out in this master thesis will apply to sub-goal 4 described in the FABULA project description [16]: "To develop and demonstrate to end user communities proof-of-concept applications based on the tools provided by the project."

The MIRROR Project

The MIRROR Project is a European Union funded project which goal is to empower and engage professional employees when reflecting on their performance and experience at work. This is to support learning engaging reflection in their work so they can learn more quickly and solve work-related problems more creatively. The deliveries of the project will be a set off applications that will support the employees of recording, sharing and learning from previous work experiences. These applications will be available in The MIRROR Appsphere as presented in Figure 1.3. The project last for 4 years and will through this do theoretical work, requirement analysis, user studies and development in the domain of reflective learning. The problem domain research will be evaluated through real life contact in businesses ranging from care homes for people with dementia to contract negation support in business environments.

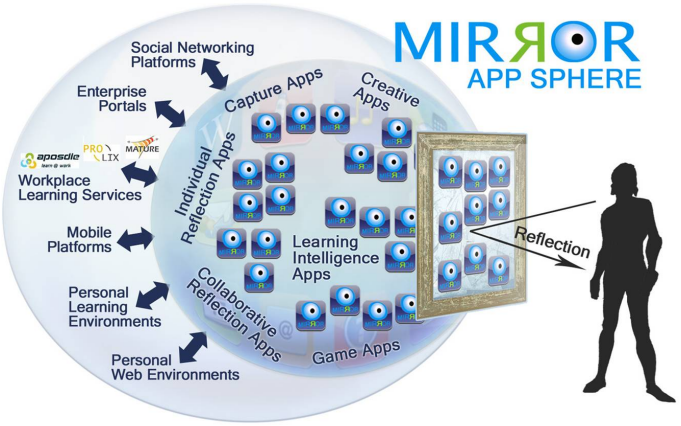


Figure 1.3: A representation of the MIRROR Appsphere, showing user domains, problem domains, application domains and outcome

The consortium partners of the MIRROR Project range from professional businesses like British Telecom through research institutes like the FZI Research Center for Information Technologies, where the general assembly that we took place at was arranged. NTNU is one of the consortium partners.

The MIRROR Project is currently in its first year of work, and was at the time of our involvement focusing on theory, user studies and prototype demonstrations, where we took part in the latter.

MIRROR web-site `http://www.mirror-project.eu`

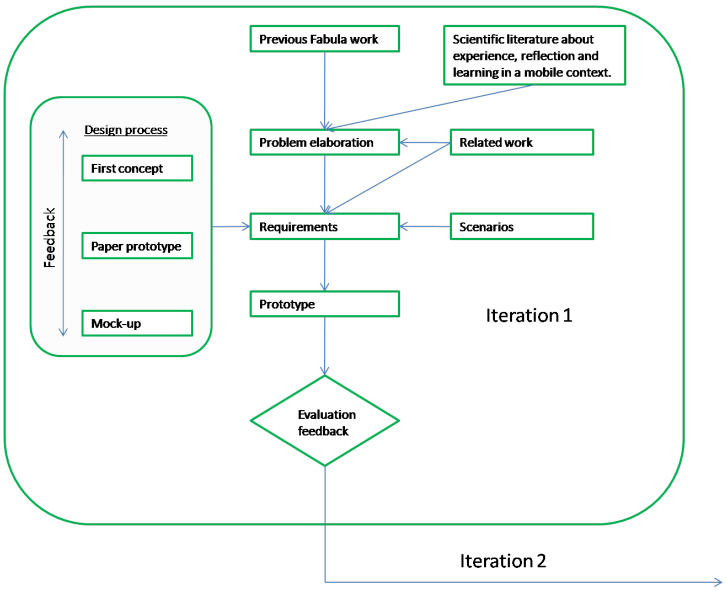


Figure 1.4: A model of the research approach used in iteration 1

1.3.2 Research Method

The development process was split into three separate iterations, where the first consisted of small iterations on ideas, design and feedback. The goal of the first iteration was to create the first prototype of the tool for individual usage. The second iteration goal was to further develop the prototype to support collaborative work on shared timelines, using the individual features of the first prototype as a basis. In the third iteration the focus was the integration of the prototype with other applications.

1.3.3 Iteration 1

Figure 1.4 shows a model of the research approach used in the first iteration of the development. The model describes how we started reading theory to understand the concepts of reflection and learning (See Chapter 3 for the theoretical background of the thesis), as well as looking at this theory used in related work(See Chapter 4). Together with the previous work done in the Fabula project (See Section 1.3.1), a problem elaboration(See Chapter 5) was created. This further lead us to the process of creating the first requirements (See Section 5.4) of the application which is described in the next section, “Steps of the development”. The first iteration ended with a expert evaluation

of the work done so far.

Steps of the Development

We had several steps of design and feedback before we were able to start developing the prototype. The design process can be split into three different steps:

- First concept
- Paper prototyping
- Computer mock-up

In the beginning of the process we had a meeting with the Fabula group at IDI - NTNU (see Section 1.3.1), where we presented our preliminary work and ideas for the prototype, as well as a scenario where we described possible usage of the application. We began designing the application based on the feedback we got from this meeting together with discussions with our supervisor. Following the process of design, prototyping and construction described in [40], the design activities began once a set of requirements had been established. As a result of the feedback we got on our first concepts ideas, we were able to create a first set of preliminary requirements and user stories.

The next step of the development was to use these requirements to create a visual image of our application. Both to help us as developers, but also for further feedback from our research environment. We started paper prototyping based on our requirements and user stories. This gave us ideas on how the application could look like, and made it possible for us to get feedback on design choices and the look & feel of the application. We used the feedback from this step to create a computer mock-up with even more specific details than the paper prototype. This resulted in us being able to start programming the first prototype of the application.

Evaluation

The first iteration of the development process ended with an expert evaluation of the first prototype. We presented the prototype to an expert on reflection and discussed how we further could develop the application to be more suitable for the goal of the thesis.

1.3.4 Iteration 2

During the second iteration, we were approached by a member of the MIRROR Project [3] to integrate our system with one of their scenarios. This made an impact on our work in the terms of adding some requirements to the application and by giving us a new scenario to evaluate. Still, the main purpose in iteration 2 was to focus on the collaborative aspect of the prototype.

The second prototype was thus developed with requirements gathered from the first iteration evaluation as well as input from the MIRROR Project scenario, except the specifics of the integration that where moved to a new third iteration. The third iteration was not planned from the start, and we felt that it was important for us to focus on the original requirements of the prototype in iteration 2 before we started with the integration. Still, with identifying requirements connected to the integration already in iteration 2 we were able to form the development of the second prototype to easier accept the new requirements of the integration in the next iteration.

The second prototype was presented at a workshop in Saarbrücken, Germany, connected to the MIRROR Project. A representative from NTNU presented our application to the participants, collected data based on the feedback received and acquired new requirements for iteration 3.

A model of the research approach in iteration 2 can be seen in Figure 1.5.

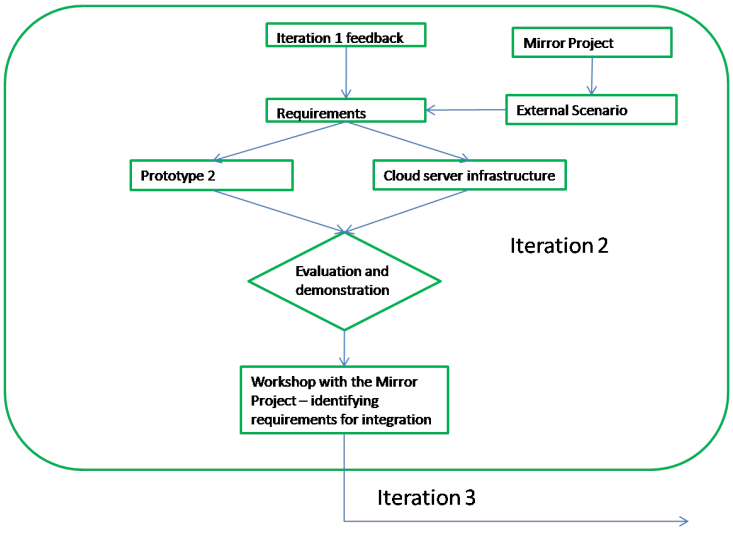


Figure 1.5: A model of the research approach used in iteration 2

[3] http://www.mirror-project.eu

1.3.5 Iteration 3

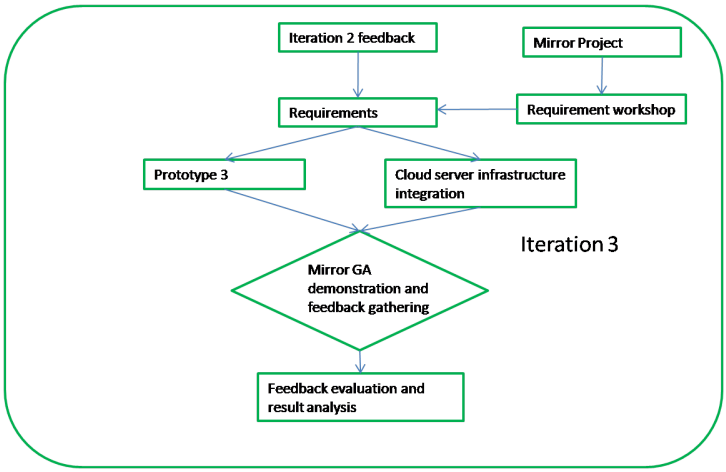


Figure 1.6: A model of the research approach used in iteration 3

Iteration 3 was conducted on the background of the proposed integration of our prototype with the applications provided by the MIRROR Project. Initially the plan was to perform two iterations, focusing on individual and collaborative reflection. However, with the new opportunities presented by the MIRROR Project, both in terms of further developing the application as well as evaluation possibilities, we decide to introduce a new iteration where we focused on the integration between the applications. We had a workshop where we discussed and set the requirements for the integration with the representative from the MIRROR Project, creating the foundation making it possible to get our application to interact with the proposed scenario. This work is further described in Chapter 8. Over the course of a couple of weeks, in cooperation with the MIRROR Project, we developed several interfaces that made it possible for other application to interact with ours.

The final evaluation of the prototype took place during the 3 day long general assembly of the MIRROR Project in Karlsruhe, Germany between the 10-12. of May. During the general assembly several of the attendees used our application as well as other applications that were integrated with ours. We collected data based on direct feedback, observation and data analysis during and after the general assembly. This evaluation is further described in Chapter 9.

1.4 Report Outline

In this section we will describe the organization of the rest of the chapters in the thesis.

Chapter 2 gives a brief overview the Timeline Android application. The objective is to present concepts and features of the application on a high level.

Chapter 3 investigates the theoretical background of experiences and experience based learning.

Chapter 4 presents some of the work already performed in the domains of technology for learning and reflection. This related work is used as basis for our work on the thesis.

Chapter 5 defines the problem and presents the high level requirement identified to investigate the problem.

Chapter 6 describes the work performed during iteration 1 of the development, which was about collecting data. This includes our first ideas, the concepts of the application, design choices made and the first evaluation.

Chapter 7 describes the work performed during iteration 2 which was about collaboration. This describes the work done in order to use the application in a collaborative setting. This includes work done on both the client as well as the implementation of the server.

Chapter 8 describes the work performed during iteration 3 which was about integrating the application with external applications. The includes adapting both the client and the server side of the application.

Chapter 9 describes the final evaluation of the application performed in the MIRROR general assembly in Germany. This chapter ends with a discussion of the results gathered against our research questions.

Chapter 10 concludes the thesis with a summary, our contributions, an evaluation of our work and ideas for future work.

Chapter 2

Timeline Application

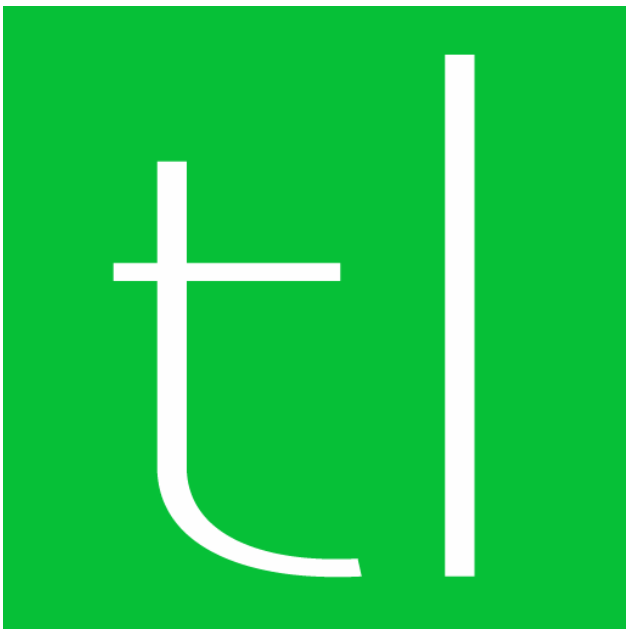


Figure 2.1: The logo of the Timeline Application

In this chapter we will introduce the Timeline application, which is the prototype we developed to support this thesis. The focus will be to explain the concept of the application, its features and describing its functionality with screen shots. To get an overview of what the rest of the work in this thesis are centered around, we feel it is important that the prototype we developed is described on a high level. The process of how we developed the prototype and took the different design decisions is in more detail described in Chapter 6, 7 and 8.

2.1 What is It?

The Timeline application is a prototype developed to help answering the research questions of this thesis. The application is developed for the Android operating system and will work on all Android devices with Android version 2.2 or newer.

2.2 What does it Do?

The prototype is an application that allows the users to share their experiences in learning, work or social life with their friends, co-workers and family. Users can choose what to share and what to keep private. The application is designed to support both individual and collaborative usage, both in offline and online mode.

You can create timelines to preserve your experiences and choose to invite other participants to take part in your discoveries and share their own with you. The application supports collection of several different data types, namely short notes, pictures, video, mood and sound. When captured or created, these data types are placed in a timeline as an event in the experience. A user can create several different timelines to use in different environments. The application is also integrated with the Android operating system so you can share events from other applications into your timelines.

2.3 What is the Goal?

The goal of the prototype is to give users a tool that can help them reflect on their learning experiences in both an individual and collaborative setting. Through the development process of the application, it has become a general tool that supports collection and representation of data, with some additional features that we, based on theory and related work, introduced to help trigger reflection amongst the users.

2.4 Functionality

In this section we will describe the functionality that the application supports. First we will explain the Dashboard view, from where the user can start working with the application. Then we will describe the functionality that directly works in the domain of reflection. These are separated into three different areas that will be presented separately:

- Collecting data - How a user can create and populate a timeline.
- Collaborating - How several users can work together in groups to collaboratively create timelines.
- Reflection Triggers - Functionality developed especially for the purpose of finding out if they can help users reflect on their work.

2.4.1 The Dashboard

All the user interaction in the application begins from the Dashboard view (Figure 2.2). From this view, the user has 6 options to choose from:

Figure 2.2: The Main Menu of the application called the Dashboard

- **add timeline** gives the user the option to create a new timeline. From this menu, the user can also choose if a timeline is to be shared with a group or kept individual. This is an attribute that can be changed later.
- **my groups** gives the user the option to organize groups. This includes creating new groups, inviting users to groups or leave a group.
- **my tags** organizes the tags created for events (explained in Section 7.3.3). This includes creating or deleting tags or create new timelines based on tags.
- **my private timelines** gives the user the option to choose between all the **individual** timelines the user has created. This means that these timelines are only accessible from the device the user is working with.
- **my shared timelines** gives the user the option to choose between all the **collaborative** timelines the user are a part of. This means every timeline that is shared with a group.
- **synchronize** lets the user synchronize all his shared timelines with the server. This will synchronize the users shared timelines to a state where all the content shared by other users working on the same timelines will be accessible.

2.4.2 Collecting Data

To collect data, the easiest way is for the user to have a timeline open. From the timeline screen the user have access to a toolbar that lets the user add different types of data to the timeline. An empty timeline focused on the 8. of May is shown in Figure 2.3. On the bottom of the screen you can see the toolbar which are described in Figure 2.4.

When captured or created, the content will be visible at the time of creation in the timeline as the example in Figure 2.5 shows.

The user can navigate a timeline to look at different periods of time or choose between different views that shows different time intervals like a month, week, day or hour.

In addition to adding different content directly by using the toolbar in the timeline view, the user can also collect data from other applications on the device by sharing directly into a selected timeline. This can for instance be from the Gallery application and lets the user add data to a timeline that

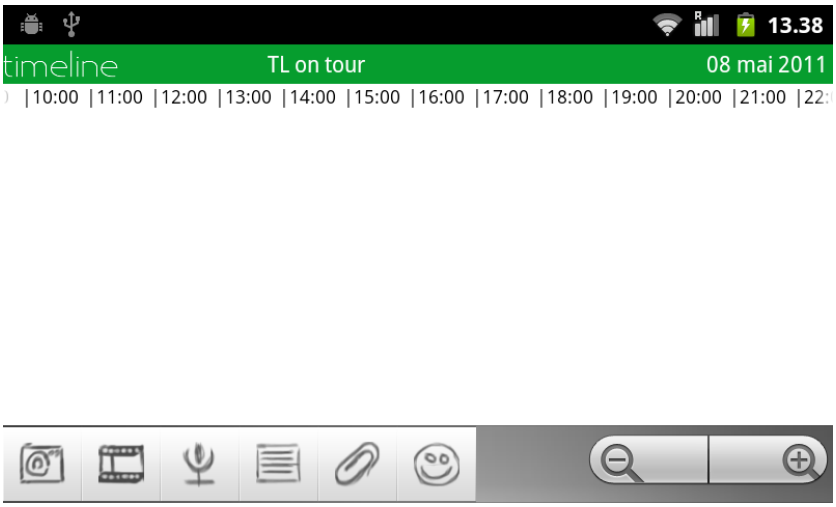


Figure 2.3: An empty timeline with the toolbar displayed at the bottom of the screen

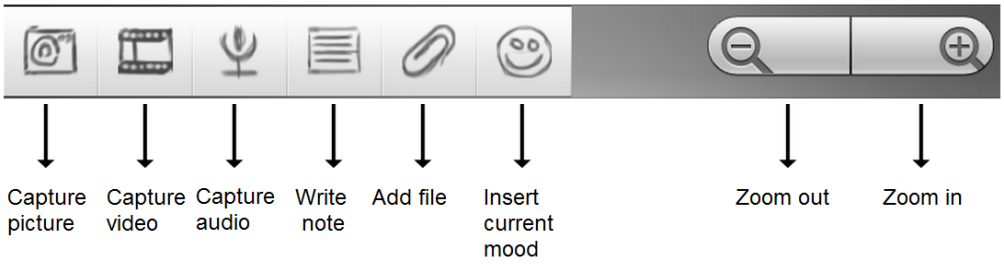


Figure 2.4: A description of the toolbar for populating the timeline with content

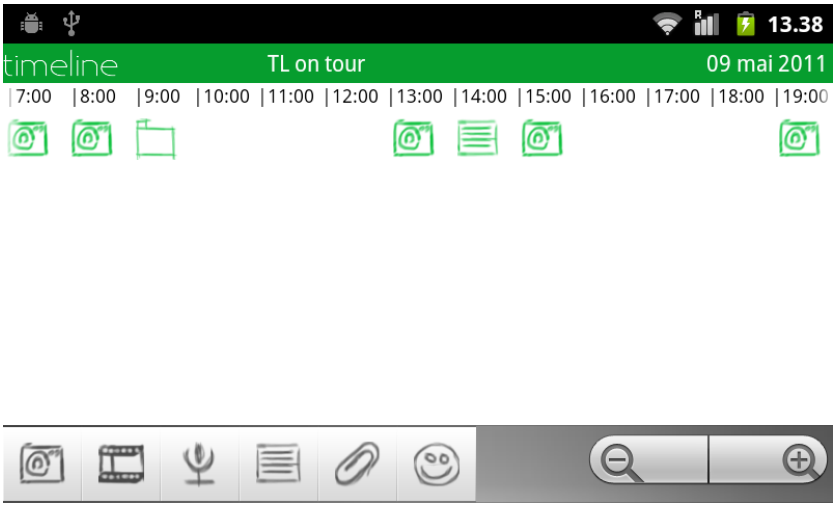


Figure 2.5: A timeline after it has been populated with different types of data

in the first place may not have been intended for it. This is a functionality supported by the Android system to allow sharing between applications.

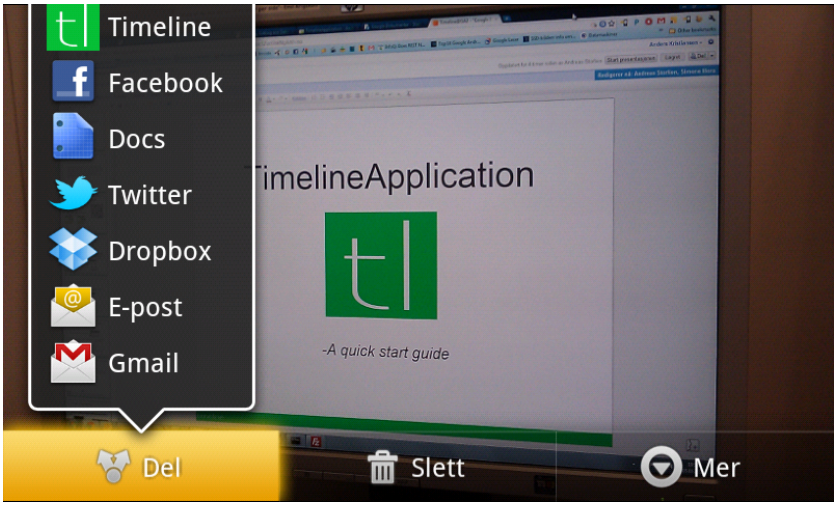


Figure 2.6: An option of sharing a picture from the Gallery application of an Android device into the Timeline application

2.4.3 Collaborating

The application support several different collaborative functionality between devices. We will now describe these functionality.

Groups

To collaborate, a user has to be a part of a group with one or more users with their own Android device. A user can create groups, invite other users to groups, be invited to groups or leave groups by using **my groups**. Timelines can be connected to groups, and the content in those timelines will be shared between all the users in the group.

Sharing Timelines and Data

When participating in a group, a user can share a timeline with that group. When the user adds content to the timeline, this content will by default be set to **not shared**, meaning that each users has to explicitly decide what content to be shared with the rest of the group. By opening an event from

a timeline, the user can decide to share it, as shown in Figure 2.7. All items that can be collected in a timeline, can also be shared.

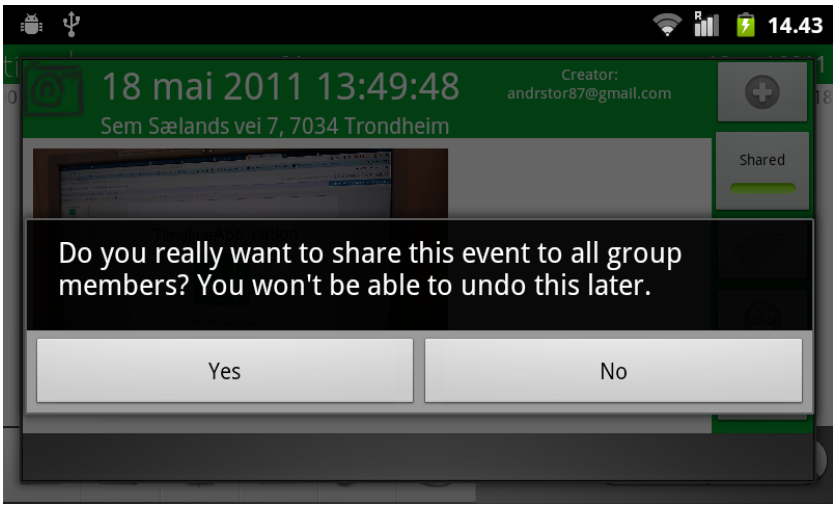


Figure 2.7: A confirmation dialog asking the user if he really want to share the item after the share button has been clicked

To receive the content that other uses has shared to a group, the user need to press the synchronization button on the dashboard. This will automatically synchronize the users shared timelines with the server, downloading all content to the device. This content will then appear in the timeline it was shared to and be available for the user to interact with.

2.4.4 Reflection Triggers

Some functionality have been added to the application with the goal of having the users interact with the data they have collected, thus triggering reflection.

Tagging

A user can tag each separate event in a timeline with user decided tags. The goal of this is to help the user organize their material and it can help the user reflect because the tagging part requires the user to go back and look at data previously captured. The user can also create new timelines based on the tagged events. For instance, from the tag menu, a user can select one or more tags and create a new timeline based on these. A new timeline will

then be created with all events connected to these tags already present in the timeline, even if they in the first place was located in different timelines.

E-mail Report

Some times it may be better to work on other platforms then the mobile phone. The Timeline application supports sending all the content gathered in a users timelines as an e-mail. This lets the user work with an organized overview of all the data captured with the application on his computer.

Mood Average

A user can insert his mood into the timeline, represented by one of four smileys ranging from happy to sad. These smileys are connected to some values, that lets the server calculate the average mood of a group working on a timeline. If every group member insert their mood to a shared timeline, each user can use this functionality to see the average mood of the timeline, giving an idea of how the work in the group is perceived by its group members.

Content Assessment

Each event in a timeline can be opened in a separate view. In this view, a user has the option to asses the content of the event by adding an emoticon to the event. In a shared timeline, this functions as feedback to the creator of the event. An example is shown in Figure 2.8

Content Commenting

Another way of giving feedback to another users content, is to add additional items to an already created event. This can for instance be a note to comment something, or a related picture. With this functionality the user can interact with each other through a shared timeline. Figure 2.9 shows how one user has commented on the picture of another users presentation.

Figure 2.8: An event can be assessed by adding emoticons to it

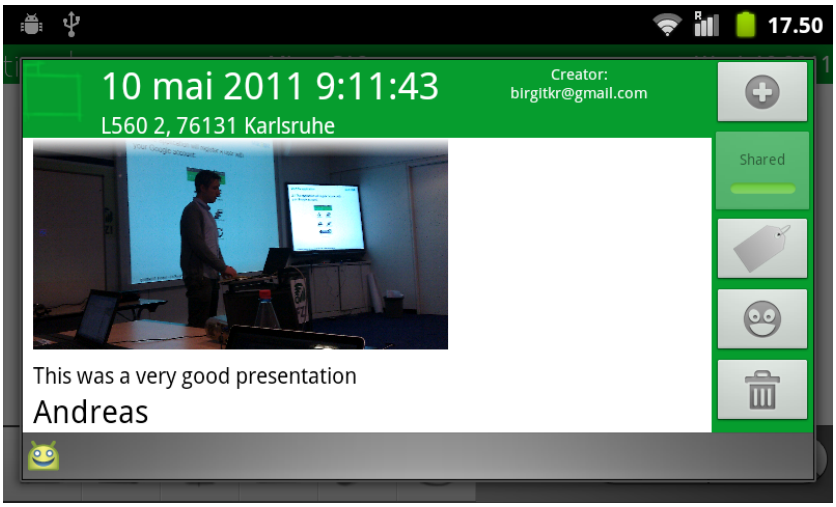


Figure 2.9: Several users can work in the same event. Here exemplified by giving feedback to each other

Chapter 3

Theoretical Background

In this chapter we will elaborate on the background theory that creates the foundations for learning as a field of research as well as how technology can be a part of this domain.

3.1 What is an Experience?

"Experience is the foundation of, and stimulus for, learning". This is one of the assumptions that Boud et.al [2] base experience-based-learning (see Section 3.2) on. In other words they say that all learning necessarily needs to be based in some kind of experience to occur. Another assumption is that *"the learner actively construct their own experience"*. What they mean is that each individual has their own representation of the experience that they have experienced.

Both these assumptions are important to take into consideration when we design for our application. First of all we want to support the learner in collecting and saving experiences, and easing access to them when they are re-visited. Second, we want to represent the experiences back to the learner in a way that can help the learner reflect on the experience or present it in a collaborative setting to achieve a learning outcome. In their model of the reflective process, Boud et.al [2] uses the terms ideas, behavior and feelings to model an experience. In our application we want to capture these human factors and connect them with experiences that the learners goes through, and with this make the learners aware of these factors as they progress through a learning experience.

3.2 Experience Based Learning

"Learning is a process whereby knowledge is created through the transformation of experience(Kolb, 1984 [29])"

Experience based learning (or EBL) is a concept that has been around for a long time. The distinguishing feature of EBL is that the experience of the learner has a central part in all instances of the learners learning or teaching. These experience can comprise of earlier events in life, earlier learning activities, current life events or participation in activities facilitated by others. The key part of EBL is that the learner, deliberate or unconsciously, analyze their experience by reflecting, evaluating and reconstructing the experience. This analysis can happen individually, collaboratively or sometimes both. The goal is that this review of the experience is going to lead to further action. [2]

As mentioned above, reflection is an important part in EBL. One of the most influential characters within the modern research of EBL is David Kolb, who with his "Experiential Learning" (1984 [29]) introduced the experiential learning cycle highly influenced by the work of Kurt Lewinsky [2]. Figure 3.1 shows the cycle where one of the steps included is that of observation and reflection as a part of the learning process.

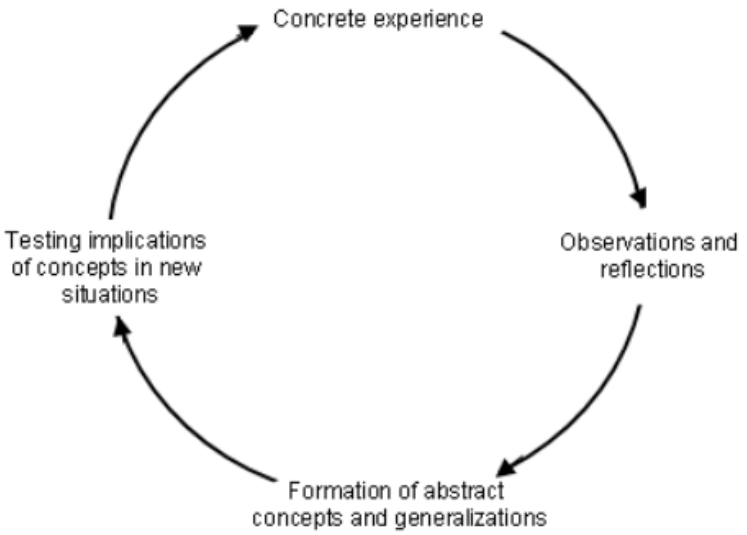


Figure 3.1: The model of experiential learning [29]

When defining the key characteristics of experienced learning, Boud et.al [2] also emphasize the importance of reflection upon earlier experiences in order to "add and transform them into deeper understanding". They continue: "The quality of reflective thought by the learner is of greater significance to the eventual learning outcomes than the nature of the experience itself". Also, in the same text, the process of debriefing and reflecting is considered an essential step. The experience alone is not necessarily educative at all. This shows how much of an importance that the reflection part of a learning process has for the learning experience.

3.3 Using Technology to Facilitate Reflection

Based on two case studies, Seale and Cann [44] presents some interesting factors that may identify how technology can help facilitate learning. They emphasize that tutors has an important role in providing guidelines and goals for how the technology should be used. Brockbank and McGill [8] argues that tutors can draw the attention to the reflective process by modeling it for the users. This shows that it is important to provide some guidelines for how the technology should be used to achieve the reflective goals if a tutor is not present. Also, the communication between the learners is regarded as an important factor. McLoughlin and Oliver [35] argues that facilitating reflection in collaborative learning can be achieved by making the learners interact with each other. This can be facilitated by collaborative technology.

Another paper, "Using Technology to Foster Reflective Learning in Higher Education" by Strampel and Oliver [47] concludes with saying that when creating an environment for reflection that is optimal for students engaging in reflection, in this case by using technology, it is necessary to understand the thought process of which the students engage in when reflecting. Students are often told to reflect on their work. As Boud et.al. [7] puts it:

> *the activity of reflection is so familiar that, as teachers or trainers, we often overlook it in formal learning settings, and make assumptions about the fact that not only is it occurring, but it is occurring effectively for everyone in the group.*

Therefore, it is important when facilitating reflection in technology to not assume that providing features for obtaining learning experiences will start reflective thinking. The technology must also facilitate the reflective thinking [47].

3.4 Learning in Situ

In the context of learning in natural environments, research has shown that experience based strategies for learning has the most engaging, effective and enduring learning experiences, rather than teacher-directed strategies [4]. These strategies has been identified as being:

- Learning by doing
- Being in the environment
- Real life learning
- Sensory engagement
- Local context

"Learning by doing" is that the students gets to execute activities themselves in a natural context, not just being told what the activity is. The importance of "Being in the environment" comes to hand to let the students visualize and understand the scale and importance of problems, for instance environmental issues. "Real life learning" has to do with how realistic and connected to the real life the activities to be performed are, for instance if they are based on real places, real issues and authentic tasks. In a natural learning environment, students should be able to use all their senses to explore and experience the environment in the learning process, something which the authors gave the name "Sensory engagement". At last, "Local context" encourage students to explore and investigate environmental problems in "their own backyard" These strategies has been used in for instance [51], and in this research the authors emphasizes the importance of collecting data as a basis for enhancing the reflection in the learning experiences (See Related Work in Chapter 4.3).

3.5 Making the Process Explicit

In "Designing technology to support reflection" by Lin et.al [33] the authors does a literature review into the problem domain of designing technology to support reflection. In their results they identify design features that promotes reflecting thinking in technology. One of these features, called "Process display", is to show the learners explicitly what they are doing to solve a specific task. This makes it possible for the learners to reflect on the process and "the students own thinking becomes an object for reflection". The authors further writes that the reflection needs to happen at two levels: a) reflection on the

product (the product in this case is the result of the work) and b) reflection on the process of how the product was created. The important thing here is the reflection on the process, which often is tacit learning and hard to capture whereas the product of learning is explicit and easier to conceptualize.

In other words, "Process display" means to create technology that makes learning that normally is tacit, explicit and overt. By making the process of the learning explicit, the technology helps facilitate reflection on what and how learning is occurring. It involves the use of technology to show the students work as well as the process that lead to the finished product.

The concept of process display fit well with how we wanted the Timeline tool to facilitate learning and reflection. By letting the user collect his experiences and presenting this back, the user can see the evolvement of the experience over time, and make the process of the learning more explicit. This gives the user a history of his work. Together with triggers for reflection, for instance sharing of events, assessments of events and tagging, this can help reflection on the road taken to reach the goal of the process.

Examples of "Process display" in tools and user studies can be found in [6, 18, 41]. These tools are presented in the Related Work section in Chapter 4.

3.6 Levels of Reflection

Reflection is a wide concept, and it is being used in diverse ways. In [19] Fleck and Fitzpatrick synthesize the literature looking on aspects such as purposes of reflection, conditions for reflection and levels of reflection (where the levels capture the behaviors and activities associated with reflection). The authors points out that "[...] the interest in reflection and technologies to support reflection has expanded beyond these more traditional domains to a range of new areas, with reflection as a topic in its own right". They say that reflection is a time consuming process, that requires the right environment and encouragement to happen. The authors identifies five levels of reflection that spans from the lowest level where the user merely describes the situation to the highest level, where the learner is "taking into consideration aspects beyond the immediate context, for example moral and ethical issues, and wider sociohistorical and politico-cultural contexts". The techniques for supporting reflection on the different levels identified in [19] will be used as a background when designing the Timeline application.

Chapter 4

Related Work

In this chapter we describe the related work in the domain of technology for learning and reflection that we used as a basis for designing the Timeline application.

4.1 Collaboration

In the paper "Tools for Students Doing Mobile Fieldwork" [41] a *wiki* that supported collaboration was extended with a mobile tool to support mobile collection and contribution to the wiki. The students worked in groups when gathering data. The data was automatically uploaded to the wiki. The authors found it beneficial that the students were aware of the others' work, known in CSCW as *awareness*, and therefore an extension to the wiki was created placing the groups' work in a timeline.

To view and organize the captured data a web application was created. The web application allowed the users to browse objects based on time or location. In time view the objects were arranged horizontally after time, and spread out vertically to allow more object to be visible. The users can zoom the *timeline* to allow different time perspectives.

In "The Design and Implementation of a Mobile Learning Resource" [45] the aim of the research was to create a prototype that would enable children between 9 and 11 years to capture learning events in the field, to annotate, share and organize them into resources for learning and to communicate directly to other learners and teachers.

In “Mobile Stories 1.0” [18] collaboration was supported in the sense that multiple devices could connect to the same story, making the latest change saved as the latest version of the story. The goal was that the collaborative work of creating these story pages was going to help the children reflect on the information they collected. The story pages changed as the exploration went forward and when the different teams of children could see what the other teams was creating. The sharing of information helped the children evolve their part of the story as they saw the information collected by the other teams.

The authors of “Mobile Stories 1.0” [18] conclude that their experience shows that it is important to create possibilities for synchronizing data that is already created, but that in the creation process (taking picture, writing note, recording audio), this is less important.

CaseLine is an application developed for supporting collaborative work and planning between citizens and municipal caseworkers in Denmark [6]. The goal was to create a common information space to ease the coordination between actors. CaseLine acts as a common collaboration object where coordination and information exchange can take place. This work shows how a timeline can be used for sharing information.

4.2 Collection and Representation

In “Tools for Students Doing Mobile Fieldwork” [41] the mobile tool was built for mobile phones running Symbian 3^{rd} edition. The application collects photos, videos, text and audio recordings, and automatically uploads them to a web server. If the same user creates many events on the timeline in a short amount of time, they are grouped together. We see the same behaviour in “ButterflyNet: A Mobile Capture and Access System for Field Biology Research” [52], where the system automatically associate photos, notes and other data containing timestamps. For example, if a biologist writes a note, followed by a photo shortly after, the two are associated.

Results from “ButterflyNet: A Mobile Capture and Access System for Field Biology Research” [52] showed that the participants readily understood the automatic time-based association. However there were problems of visualizing the data when much information is gathered in a short time frame. The authors therefore emphasizes the need to provide the user with a way to adjust and visualize the granularity of automatic associations in capture

and access systems. ButterflyNet supports access to captured information through the ButterflyNet Browser. The browser has a timeline visualization that allows users to jump to content by time and date. The rest of the screen shows the notes and photos.

In "Noising around: Investigation in Mobile learning [51] a more case-specific mobile capturing tool was developed. The tool has the ability to record the noise level in the environment using the devices' microphone. The tool does not have the possibility to record the result, thus it has to be recorded using another tool, or in this case using a paper sheet. The tools only purpose is to work as a sensor for collecting the information needed to answer the question of the exercise.

The prototype of "HandLeR" [45] includes operations for actions to enable the learner to capture and annotate events such as images, sound and written notes, to perform experiments and to converse with teachers and learners. The outcome of the captured events is stored in a "learning object", which can be a test, a simulation or a lecture. The learning objectives can be accessed and presented on a timeline, a spacial map or an "idea map". Since the learning objectives are represented as XML, it also enables the learner to create a personal website on a learning topic.

4.3 Approaches to Reflection

In "Noising around: Investigation in Mobile learning" [51] the research showed that the student often used the paper resources to take unspecified notes and created drawings on them to express their feelings. They also write that it is very important when developing these kind of applications to manage to create a balance between the doing and the attending of activities to support both exploration and reflection in the learning activities that takes place.

The paper "Mobile Stories 1.0" [18] takes another approach on reflection giving the student a more active role within the mobile tool. With this tool, the students can create their own story pages with information about the place they are visiting. These pages can include text, photos and/or audio. The stories often has a beginning that the students can start with and complete themselves. Together, these story pages with text, pictures and audio creates a story of the experience that the students took part in.

Computer Supported Intentional Learning Environments (CSILE) is a computer-supported knowledge medium constructed to support intentional learning

and is used by Scardamalia and Bereiter [42] to describe how a learning environment can be designed to support reflection in learning. This system allows several computers to connect with a server where the students can use a collaborative environment to share text and graphical data. Labeled notes are used to share ideas, relevant topics or pieces of information. This is used to discuss questions and theories as well as compare and contrast different perspectives. In this environment the students can learn individually and collaboratively by presenting their own understanding and getting feedback on their work from others. They reflect on their own learning by contrasting themselves with peers and by responding to peers comments on their work.

4.4 Technology

Table 4.1 presents the different technologies we have found being use in related work and research. The research also include different advantages and disadvantages the technologies provide. The problem however, is that most of the research done in these papers took place some years ago, and as we all know, technology is advancing rapidly. Therefore, it can both be hard to take technological advices from work done even if it is only a couple of years old.

Paper	**Technology**
"Tools for Students Doing Mobile Fieldwork"	Nokia with Symbian and web-based application with Ajax.
"Noising around: Investigation in Mobile learning"	iPod Touch
"Mobile Stories 1.0"	Windows Mobile
"ButterflyNet"	Nokia digital pen, "Smart Camera" and OQO handheld with webcam running Windows XP. The findings were presented in The ButterflyNet Browser written in Java J2SE 5.0.
"HandLeR"	Custom prototype
"CaseLine"	Web-based(not specified specific technology)

Table 4.1: Related work and used technology

4.5 Existing Timeline Tools

Table 4.2 gives an overview of some applications using the different kinds of timeline visualizations. We used these examples as input when deciding how to visualize the timeline in the Timeline application.

Name	Layout	Content generation
Google News Timeline [27]	Horizontal time, vertical content	News from different sources
TimeGlider [34]	Horizontal time, vertical content	User generated
Google Timemap [28]	Horizontal time, vertical content, map	User generated content connected to maps
CaseLine [6]	Horizontal time and content	User generated, municipial work, collaboration
SIMILE [38]	Horizontal time, vertical content	User generated
SmartHistory [1]	Horizontal time and content	World history
Wiki awareness extension[41]	Horizontal time, vertical zoom	System generated from user editing
Twitter [49]	Vertical time, no zoom	User generated content
Facebook [17]	Vertical time, no zoom	User and system generated content

Table 4.2: Example of different timeline tools, their layout type and content

4.6 Conclusions Found in Related Work

From the related work we have examined several different works done in the domain of learning and reflection. From this research we have taken some results and guidelines that we have used when we took design choices for the Timeline application. The main conclusion we drew from this work are presented in Table 4.3.

Table 4.3: Conclusions from Related Work

Article	Conclusion
"Tools for Students Doing Mobile Fieldwork" [41]	It is important to create a tool that is simple enough to use that, in an educational setting, students actually can find the time to use it. Students in the article found the application to be "very simple, but powerful".
"Noising around: Investigation in Mobile learning" [51]	It is important to manage to create a balance between the doings and attending of activities to support both exploration and reflection in the learning activities that takes place.
"Mobile stories 1.0" [18]	Concludes that it is more important to support synchronization of data that is already gathered, but that in the creation process (taking picture, writing note, recording audio) this is less important.
"ButterflyNet" [52]	The participants of this research understood the automatic time-based association. However there were problems of visualizing the data when much information is gathered in a short time frame. The authors therefore emphasizes the need to provide the user with a way to adjust and visualize the granularity of automatic associations in capture and access systems.
"The Design and Implementation of a Mobile Learning Resource" [45]	Further evaluation of the usefulness of mobile technology for learning should be delayed until technology has been developed that is easy and intuitive to use. One should however take into account that the paper was written in 2002.
"The Design and Implementation of a Mobile Learning Resource" [45]	Another useful conclusion is that a successful learning organizer should fit into the daily activities of informal learning. It should be able to capture and recall an object or event they would otherwise forget. Mobile learning is more mediated by its context than classroom instruction. Context includes time, location, the learners trajectories, goals and motivation, the surrounding resources, co-learners and other available conversants.

4.7 Relevance for our Work

In the decisions we have taken during the course of developing the Timeline application, we have had to look back at the previous work done in the domain of technology for reflective learning. It has been important especially in deciding what data to support, how to visualize this data and how to perform the sharing of the data. Before the first iteration the research done in these papers were important guidelines for taking certain decisions when the basis of our knowledge and experience was lacking.

Chapter 5

Problem Elaboration

In this chapter we will elaborate further on our problem by defining our task and presenting our high level requirements.

5.1 Problem Definition

To answer our research questions we will develop a tool for mobile devices allowing users to collaboratively construct, share and reflect upon learning experiences that they annotate with data. As a representation for the learning experiences we have chosen to use the timeline representation.

The tool will be designed with reflection in mind, but the goal is to make it a general tool that does not only fit one usage scenario, but can be used by different possible user groups. Our goal is to use this tool to answer our research questions, and still make it scalable, usable and modifiable enough to make it applicable for different scenarios.

5.1.1 Designing Technology for Reflection

With the enormous amount of information available, and the easy access different technology gives us, learners must not only be able find the information, but also learn how to makes sense of it. To make conscious decisions about the uses of information, learners have to step back and reflect on the information they collect. Problems are not solved by just reading and writing. Reflection on both the process for and the actual solving of problems is

needed to achieve a good result and to further enhance the ability to learn. These skills are important in both learning with and without technology, but when being supported by technology however, this technology should promote these aspects within learning [33].

The goal is to offer technology that not only makes information search and collection efficient, but also provides scaffolds that supports reflective thinking and problem solving. In our task, this means utilizing the collaboration of creating shared learning experiences.

5.2 Scenarios

Based in the work done in theoretical background and related work, we developed two scenarios to give an insight into what we wanted our tool to support in terms of collaborative learning and reflection. The high level requirements were developed with these scenarios in mind. However, when we evaluated the tool, none of these scenarios were similar to the actual setting of the evaluation. This can make the use of this scenarios in the thesis somewhat inconsistent with the result we obtained through our evaluation. Still, the scenarios were important for the requirement elaboration and the design choices made in the development and can help enlighten some of these.

The two scenarios in the next sections envisions the usage of the Timeline application in two different settings, first on a school trip, then during a school project. The first scenario were developed before we started developing the application, based on the work in theory and on the feedback received during the first stages of the work done in this thesis. The second were developed after the first iteration, more concerning the collaborative aspect of the application, and used as a walkthrough to describe the possible usage of the application for the evaluator in the first iteration evaluation (See Section 6.5).

5.2.1 First Scenario - Using the tool on a School Trip

In this scenario, our users are students in a school class traveling on a field trip to visit the Norwegian University of Science and Technology (NTNU), campus Gløshaugen. The children attending is about ten to twelve years old.

The scenario is set up to follow the notion of experience-based learning as illustrated in Figure 5.1. This figure the cycle of a learning experience, and tries to illustrate how a learning experience unfolds. The experience starts

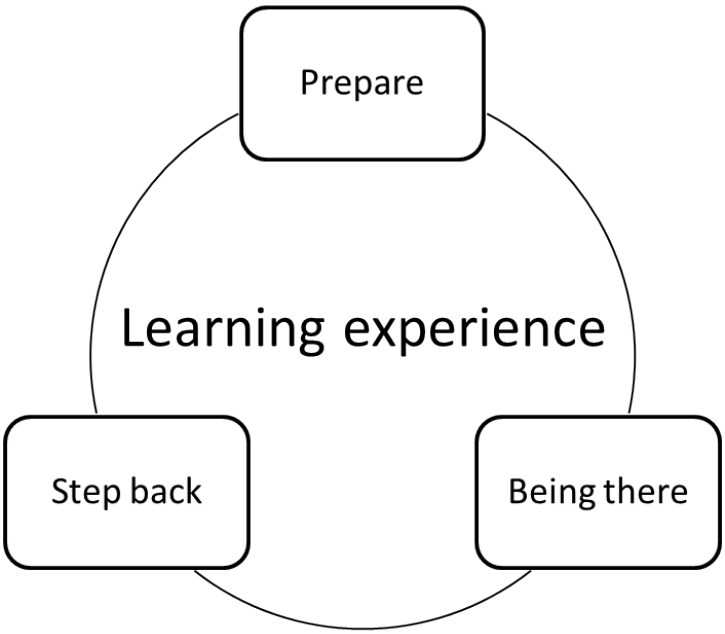


Figure 5.1: Model of experience based learning

with a preparation stage where the learners prepare for some form off upcoming event, for instance a field trip. Then comes the stage of "Being there", where the students are at the field trip location, exploring, discussing, looking, reading etc. Information is collected and new experiences occur. After the field trip, the students "Step Back" to reflect on the things that happened during the field trip, and on the experiences they remember. This reflection process consist of returning to the experience, attending to the feelings and re-evaluating the thing that occurred, for instance what was good/bad, fun/-boring etc [30].

Phase 1: Preparing

The students are divided into groups, and each group is handed a mobile device with the "timelineApp" pre-installed. The groups are told that they will soon travel on a field trip to NTNU. For preparation purposes the students are given the assignment of finding information about NTNU, for instance history and education fields. As the students gets more and more familiar with NTNU, they are told to pick the findings they find interesting and add these to the timeline so they can further explore these items on the field

trip. Figure 5.2 shows examples of collected information in the preparing phase. The events themselves are not that important, it is more the process of collection and discussions around the selections that prepares the learning experience.

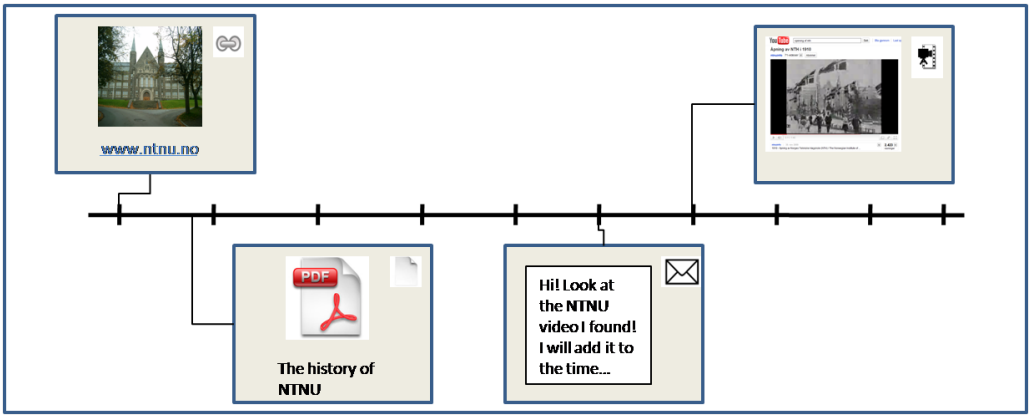


Figure 5.2: Phase 1: Examples of research done before the learning experience, represented on a timeline

Phase 2: "Being there"

Still in groups, the students explore the Gløshaugen campus. Their assignment is to find out more about the history of the university, and further explore the findings they had in the preparation phase. Whenever they find something relevant, they can collect the information using the timeline application. This can be notes, pictures, videos, sound clips and so on, as illustrated in Figure 5.3.

Phase 3: "Step back"

After the field trip, the students are told to pick the events in the timeline that they feel was most interesting/fun/special and share these with the rest of the class. All the events can then be represented on a common timeline, that the whole class can see together. Then, in the "Step back" process the students can look at similar findings, popular findings or special findings. Are there some patterns in the findings? Are some events more connected than others? Does a trail separate from the main objective? Figure 5.4 illustrate a large collection of events on a timeline. Different colors or tags can be

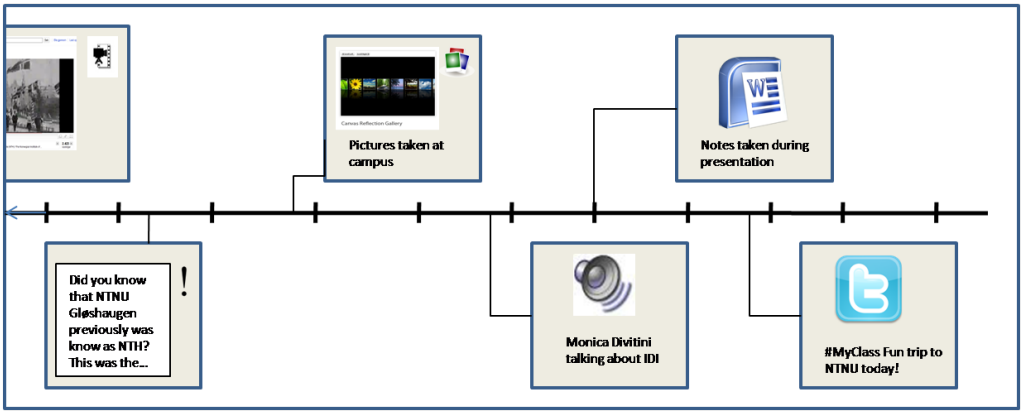


Figure 5.3: Phase 2: Examples of events collected during the field trip, added to the same timeline used in the preparation

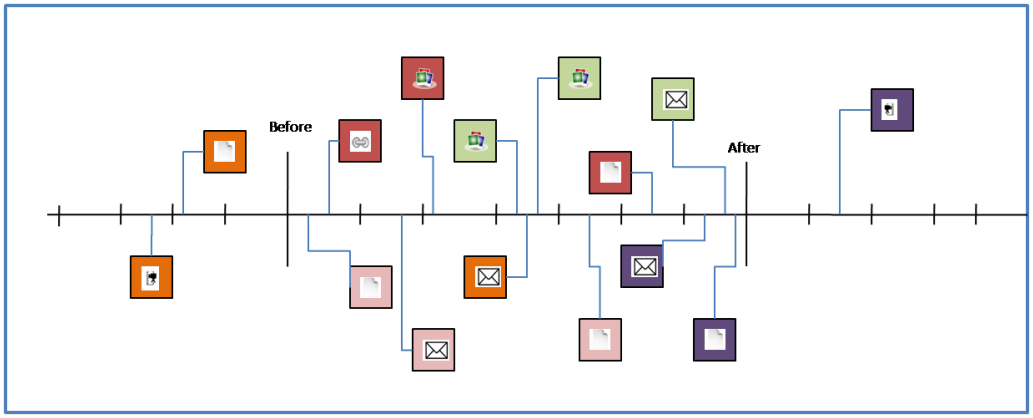


Figure 5.4: Phase 3: The most popular events collected during the field trip

used to separate the events from each other and create different connections amongst them.

5.2.2 Second Scenario - Using the tool in a School Project

At the Norwegian University of Science and Technology, NTNU, a group of students are taking the course Experts in Teamwork [1]. In the course Experts

[1]Experts in Teamwork is a course where students from different education programs is put together in groups working on different social useful projects. The goal of Experts in Team is experiential learning, meaning that the students gets interdisciplinary project experience that they later can reflect on and learn from - `http://www.ntnu.no/eit`

in Team, the focus is not only on the project itself, but also on the process of which the group goes through to reach their goal, including project processes, group roles, distribution of work etc. At the end of the course, the students have to present both their project results and their project process results and experiences.

The group is going to work on mapping the construction industry in the city of Trondheim by looking at smart solutions that can help save both money and the environment. During this project work, the group is using the Timeline application for collecting information and to get an overview of the progress in the project. The group hopes that this will enlighten different challenges the group has come upon in the process. Each member of the group uses an Android device with the application installed. One of the members create a new group and invites all the others to join. A new timeline covering the project is created by the group. The individuals have their own private timeline that reflects the content of the group timeline, as well as containing private entries that the members can choose to share to the group timeline if they find it relevant.

At the beginning of the project the students uses the first day to get to know each other, this results in the first entry of the timeline, a group picture and a note that describes the first day of work. In the following weeks, the members does individual literature surveys to learn more about the subject. The relevant findings are shared by each member by linking web-pages and documents in events on the timeline. Each student can then go in to these events, look at the material and either asses it (based on if they think it is relevant or not) or they can add their own notes to the events, sharing their thoughts about the content. This way, each member can work individually on collecting information, while the group can keep an overview of what each member is working on as well as giving feedback on it.

As the focus of this work also is on the group process, the students are encouraged to share their own thoughts about the work progress in the timeline. If they don't want to share their personal thoughts, they can still add them to their private version of the timeline, and later on decide whether or not they want to share it to the group timeline.

Later on in the project, the group goes on several field trips to collect data, either individually or together with other group members. On these trips the students gather data by taking notes and pictures, recording interviews and filming construction sites, all by using the functionality the Timeline application provides. All the data are later shared from the individual devices to the group timeline, so that every member of the group can look at it, use

it and give feedback. After one of these field trips, Karl - a group member, is writing a report on the subject of the trip, and uses the timeline as a reference when writing the data, looking at notes, pictures and listening to an interview they conducted. All the data has been collected in events placed on the timeline at the time they were added during the field trip. Karl also remembers that in the beginning stages of the project, Kari - another member of the group, added a link to a relevant paper in the timeline. Karl search the timeline to find this entry and uses it as background for his report. He also puts a note in the event where the paper entry was found, telling the other members that he used this source as background material for the report he is writing.

At the end of the project the group is going to deliver a report on the project as well as a report on the project process. In a addition to all the data and information collected, the group uses the timeline to describe how their work progressed. When was the progress on top? Were there periods with conflicts? Which discoveries had the biggest impact on the project? The group answers these question both by looking at the shared timeline in terms of when the population of the timeline is the largest, when few entries where made and which entries got the most attention. They can also use each individual members' timeline to look at personal comments or entries in the timeline if they in some way are connected to episodes in the project period which had an impact on the group process.

5.3 High Level Requirements

In this section we will describe the high level requirements for the application by considering the core concepts in a learning situation where our application can support the learners in their activities. We have elaborated about this theory in Chapter 3 and identified related work in Chapter 4. In this section we will describe five high level requirements for our tool based on an analysis of theory, related work and scenarios.

5.3.1 Collecting Experiences

As we are using a mobile device for collecting information, we define an experience in our application to be a collection of events that the learner captures. An event can consist of one, or several annotations. An annotation

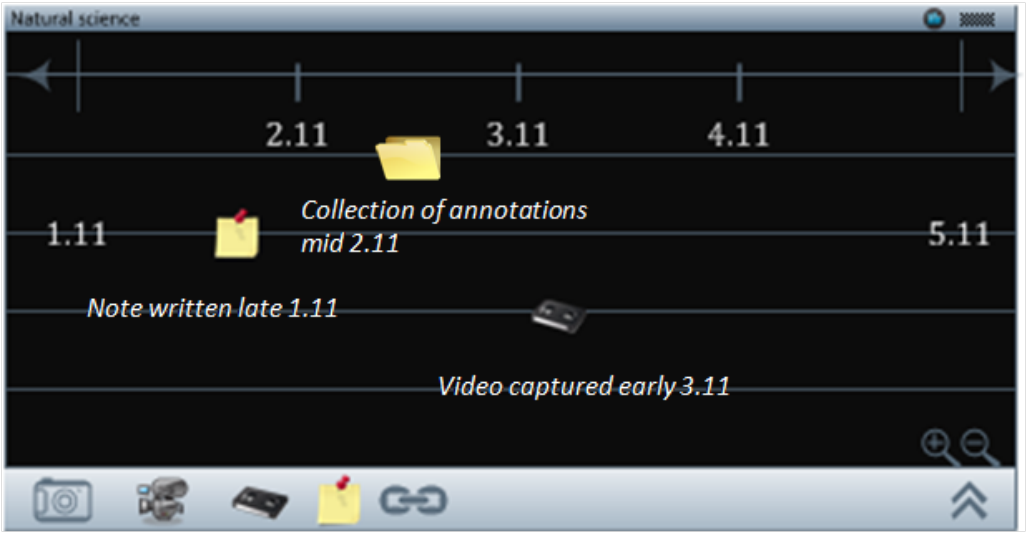


Figure 5.5: Mock Up with different experiences scattered through a week.

in this case can be whatever the user sees as a valuable happening to capture, for instance in the form of a picture.

One of the basic requirements for the application is to allow the learner to use it for collecting and saving data of different happenings, or what we call "events", as annotations in the application. All these events should together create an experience. The collected experience with events function as *the foundation of, and stimulus for, learning [7].* The events consists of a collection of items that is connected to the happening. The user should be able to open an event and visually be presented with its collected content. To separate different events, they should be tagged with a timestamp for when they were created and placed as an annotation in the timeline accordingly. See Figure 5.5 for an illustration of events captured and presented to the user in a timeline according to when each event was created. Each event contains details about the captured items connected to this event.

5.3.2 Representation - Using the Timeline

Collection of information is not a unique property of mobile applications. As for the collection of "experiences" as we call it, this itself does not separate much from collecting information. For instance the multi-platform application Evernote [14] is a well known example of just that. To separate us from other information-collecting applications we want to represent the ex-

periences as small collections of information in a timeline according to the time they were created. Another requirement is to create a timeline based representation of the experience that the users can utilize to put their experiences into the context of time. The concept of learning experiences as well as the choice of using the timeline as a representation of the learning experiences was introduced in Section 1.1.2. We want to look at the timeline as an representation a of learning experience and see how the concepts map together. This is one of our research questions (See Section 1.2 - RQ2).

5.3.3 Being in the Environment

Another important requirement for the application is that it has to enable in situ(in context) usage. The user should be able to collect and interact with content when he/she is in the situation. "Being in the environment" is a experience-based learning strategy that encourages the learner to experience and appreciate the natural setting the learning takes part in [4]. Also, research show that in situ usage makes mobility a core part of the technology, and not just an extra feature of the application [18]. This makes it important that the application not only supports collection of experiences, but also gives the possibility to interact with the experiences through other resources, made available by the application. This can for instance be the sharing of experiences, further research into experiences or creating connections between experiences.

Another important aspect of being in the environment is availability. Several resources of a mobile device requires Internet access to function. This is often unavailable or very expensive for the users when using the devices in a non-urban environment. Nevertheless, it is important for our tool to still offer the possibilities of collecting experiences even if such resources are unavailable. Therefore, an option of using the tool in offline mode is very important and certainly a requirement.

5.3.4 Sharing the Timeline - Collaborating

We don't want this to merely be a information collecting or a personal diary type of application. As stated earlier, we want to use this application in the context of learning and reflection. Therefore we have to separate our work from the masses of applications that exists doing the things mentioned. One of the aspects we want to explore is therefore collaboration. If several learners

can collaboratively work on the same timeline, share their timelines to other learners or groups and/or give and receive feedback on their timelines - will this help foster the reflective process in the learning experience?

The application developed should create a collaborative environment where learners can experience the learning together, sharing and reflecting. The application should be the environment where individuals can present their learning experience to each other so that a collaborative understanding of the learning process can be reflected upon. Previous work shows that when providing users with a collaborative environment, the learning outcome for the participants is high [18, 42].

Basically two things should be supported with this application; sharing a timeline with another user or a group of users, and working collaboratively on the same timeline, in either real time or with time based synchronization so changes are reflected to each individual user.

5.3.5 Re-visit Experiences - Reflecting

As a large focus of this work is reflection in learning, one of the requirements of the application is to foster reflection. Boud et.al [7] describes reflection as an activity where people "recapture their experience, think about it, mull it over and an evaluate it". They focus on three aspects:

- **Returning to experience** - Recalling what has happened, looking at it "a second time".
- **Attending to feelings** - Using helpful feelings connected to the experience or removing obstructive ones.
- **Evaluating experiences** - Using the knowledge acquired and re-examining the experience, integrate this new knowledge into ones conceptual framework.

We can also use these three aspects when defining the requirements for reflection in our application. First we want to give the learners the opportunity to *Return to the experience*. This is not a straight forward task. The learners must be aware that they have had an experience because an experience is not only a sensation, it also entails thinking. Boud et.al [7] argues that "an experience has within it judgment, thought and connectedness with other experiences". Therefore, to make it easier to recall an experience, it should be put in a larger context, where different experiences together creates the learning experience. This can be supported by letting the learners "save"

their experiences in different contexts. In our case, this context will be a timeline. A specific timeline should reflect a specific experience and contain connected and related events that together will shape the learning experience the learner has evolved. As memories of experiences dwindle over time, a technological support for saving the physical (place, context, environment) and the mental (feelings, ideas, behavior) part of an experience is important. Also, the experience must be presented back to the learner in a way that helps the user recall what he or she was experiencing.

Second, as mentioned, experiences are connected with feelings which help shape the experience for an individual. This is hard to cover with technology. Feelings in a reflective process is very tacit and hard for a human to share, and thus difficult to capture with technology. Hildrum [23] argues that sharing tacit knowledge in a non-verbal setting is possible to some extent if the participants has a shared frame of reference. In our application, the shared frame of reference will be a timeline and its content. As this will be a collaborative application, we want to make it possible for users to attach feelings to their own and/or other users experiences (timeline). It is important that the feelings are represented to the user as something simple and easy to understand. The "like" functionality of the social network site Facebook [17] is an example of a feeling connected to an event that we will look to implement in our application. This will be how we *attend to feelings*. See Section 6.2.5 for a presentation of how this is utilized.

Third, we want to make the collected experiences a resource for the learner to re-visit the experience - working with the content in a collaborative setting. This way we can support the user in *using the acquired knowledge* and *re-examining the experience* to lead to outcomes as presented in the model of reflection by Boud et.al:

- Create new perspective on experiences
- Create changes in behavior
- Apply new methods to the learning
- Commit new action based on the experience

(See Figure 1.2)

5.4 Requirement Analysis

A requirement analysis was performed in order to support elaboration and implementation of the solution. In this section we will present the requirements made for development of the first prototype, both in terms of client and system requirements. These are functional requirements identified from the high level requirements presented in the previous section and is created to map the concepts of the Timeline application to the available technology.

We prioritized by assigning each requirement a priority and a time estimate (in approximately work hours) and used a Return of Investment(ROI) [2] formula to calculate a ROI number. The list of requirements are sorted by this ROI number, making the requirement with the highest ROI number the highest priority.

Some of these requirements are dependent on each other and therefore we did not implement requirements from top till bottom, but used the list as a starting point for development. We were not planning on implementing every requirement because of the time limitation we had, but all requirements in the list were still viable in terms of our goals. Also, other requirements were added later, as new inputs came to our attention.

We identify the requirements throughout this thesis by connecting them to the iteration(I) they were developed in and to the system they belong to. This gives us these types of requirement identificators:

- CRI-X - Client requirements in iteration I
- ARI-X - Architecture requirements in iteration I
- SRI-X - Server requirements in iteration I
- WSRI-X - Web service requirements in iteration I
- ERI-X - External requirements in iteration I

5.4.1 System Requirements

In Table 5.1 the basic requirement for the application is presented. These are connected to the storage and the user interface of the application. The requirements are result of investigations of how Android applications are

[2] A performance measure used to evaluate the efficiency of an investment or to compare the efficiency of a number of different investments.

constructed, hence the SQLite database, content providers and activities. No prioritizing is performed on these requirements because all later requirements are dependent on them (except requirement AR1-5) to be feasible.

Table 5.1: System requirements for iteration 1

ID	Name	Description
AR1-1	Persistent Storage	Create SQLite database support for creating and performing actions on persistent databases using the application.
AR1-2	Databases	Create databases for different functionality: timelines, notes, picture/video/audio URI's, linktables etc.
AR1-3	Content Providers	Create interfaces for the app to interact with the database by performing actions like insert, delete and update.
AR1-4	Timeline-view activity	Design and develop a view to represent a timeline with the possibilities of handling touch gestures and visualizing data based on their attributes (time captured).
AR1-5	Main menu-view activity	Create a main menu screen where the user can choose different activities in the application
AR1-6	Models	Create models for all the objects that the application will handle. For instance timeline, event, picture, note, video etc.

5.4.2 First Iteration Client Requirements

The requirements presented in Table 5.2 are the ones we prioritized when developing the first prototype of the application. These requirements are mainly connected to functionality the user can perform or different types of data that the application can handle. To create these requirements we built a number of user stories elaborated from the scenarios and the high level requirements. These users stories are built as follows: "As a [<role>], I want [<goal/desire>] so that [<benefit>]".

Table 5.2: Client requirements for iteration 1.

ID	Name	User Story	Prio	Est.	ROI
CR1-10	Save timelines	As a [user], I want to [have my timeline saved automatically in the current state], so I can [work with it on a later time].	10	80	0.50
CR1-19	Open item	As a [user], I want to [open a item on the timeline], so I can [see it's details].	10	50	0.20
CR1-8	Show timeline interval	As a [user], I want to [see time intervals for my timeline], so I can [watch different time intervals for the timeline].	7	6	0.18
CR1-14	Add Picture	As a [user], I want to [be able to add a picture to an event] so I can [connect related items in one event].	9	30	0.13
CR1-11	Browse timelines	As a [user], I want to [be able to browse the timelines I currently have access to], so I can [choose the one I want to work with].	10	20	0.13
CR1-18	Zoom timeline	As a [user], I want to [be able to zoom in and out on a timeline], so I can [see the timeline in the context of a day, week, month, semester].	6	50	0.12
CR1-15	Add Note	As a [user], I want to [be able to add a note to an event] so I can [connect related items in one event].	10	10	0.11
CR1-4	Write Note	As a [user], I want to [write a note], so I can [access it from the timeline].	10	5	0.11
CR1-9	Create new timelines	As a [user], I want to [create a new timeline], so I can [use it to capture items].	10	5	0.11
CR1-1	Capture Picture	As a [user], I want to [capture a picture with the camera], so I can [watch it from the timeline].	9	10	0.10
CR1-7	Scroll timeline	As a [user], I want to [use my finger to scroll the timeline], so I can [watch different time periods for the timeline].	7	20	0.09

Continued on next page

Table 5.2 – continued from previous page

CR1-13	Group items	As a [user], I want to [be able to see items in a group when they are added with close time intervals], so I can [more easily keep my timeline clean].	5	40	0.08
CR1-12	Name timelines	As a [user], I want to [be able to give my timeline a name], so I can [separate it from other timelines].	7	1	0.07
CR1-20	Asses events	As a [user] I want to [be able to asses an event], so I can [connect the event to my feelings]	4	40	0.07
CR1-16	Add Video	As a [user], I want to [be able to add a video to an event] so I can [connect related items in one event].	4	30	0.06
CR1-17	Add Audio	As a [user], I want to [be able to add audio to an event] so I can [connect related items in one event].	4	30	0.06
CR1-6	Delete item	As a [user], I want to [delete an item on the timeline], so I can [keep my timeline as I want].	5	2	0.05
CR1-2	Capture Audio	As a [user], I want to [capture a audio recording with the microphone], so I can [listen to it from the timeline].	4	10	0.04
CR1-3	Capture Video	As a [user], I want to [capture a video with the camera], so I can [watch it from the timeline].	4	10	0.04
CR1-21	Capture Video	As a [user], I want to [capture a video with the camera], so I can [watch it from the timeline].	4	10	0.04
CR1-5	Edit Note	As a [user], I want to [open a note item on the timeline], so I can [edit it].	3	5	0.03

Chapter 6

Iteration 1 - Collecting data

In this chapter we go thoroughly into how the first iteration of the development process took place. The goal was to develop a prototype application to demonstrate the use of different annotations to collect experiences by using the mobile device in situ, thus the iteration name "Collecting data". In this iteration our focus was on the individual collection of data over an arbitrary period of time. Arbitrary meaning that the application in itself is so general that it can be used for both a short or a long period of time without the user loosing any of the main functionality.

In the following sections we describe our first ideas, followed by the main concepts of the application. Then we present the design choices taken in iteration 1, before we present the evaluation of the first prototype.

6.1 First Ideas

The first steps of iteration 1 were to decide what type of information to collect, and how to visualize the timeline. To do this, we started by defining a set of scenarios illustrating the possible usages of the envisioned application. These scenarios (Section 5.2) helped us create the first requirements of what the application should do, and how it could do it.

We presented these first concept thoughts and sketches to the Fabula group and got feedback on several things. First, the discussion was on how to visualize the timeline. Second, we were asked to decide on what type of data we wanted to support collection of. Which type of annotations is most relevant when it comes to learning and reflection? Also, we had to find a way

to separate this tool from all the other data collection tools that already are in the market. These issues became the focus of the prototype in iteration 1.

6.2 Timeline Concepts

Figure 6.1 illustrates the core concepts used in the application, and how they relate. The application lets the user create several timelines, associated with different themes. Each timeline represents a learning experience that the user, with the application, create by collecting data to the timeline.

The timeline can consist of several events. An event is a collection of one or more items. These events are spread throughout the timeline according to the time they were created, and together - over time - they create a visualization of a learning experience that the user can benefit from.

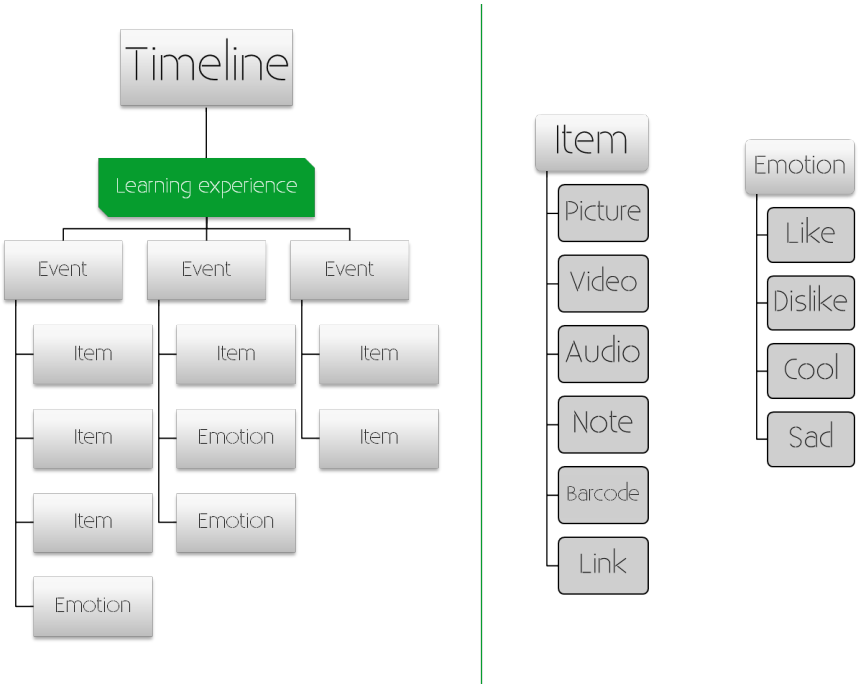


Figure 6.1: The different concepts in the application

6.2.1 Learning Experience Represented as a Timeline

In the application we want to support reflection on a learning experience by making it easier for the user to collect events and activities connected to the

experience. When we visualize these events and activities in the timeline, we can help users, to a certain extent, identify the process they have been through. Our idea is that the timeline represents a learning experience in a learning context, for instance a school project, a trip to the museum or even just a specific subject that is of the users interest. So, one single timeline equals a learning experience for the user.

6.2.2 Events

Events are snapshots of happenings that the learning experience consists of. An event can contain one or many items. The number of items connected to an event depends on what representation that suits the situation best. A picture can tell a lot, but a short note that describes the situation, feelings, ideas and thoughts can further enhance the learning experience by better supporting reflection. Previous research has shown that several different representations of data can be suitable for enhancing a learning experience. See for instance [18, 36, 19].

The grouping of items in events are based on related work that shows that items that are collected in a short period of time should be grouped together [41, 52].

6.2.3 User Collected Items

In the application items that utilize the embedded features of the mobile device are supported. These are the items that the user has the option of collecting using the application:

- **Picture:**
  A picture captured with the camera of the mobile device.
- **Video:**
  A video film captured with the camera of the mobile device.
- **Audio Recording:**
  An audio recording recorded with the microphone of the mobile device.
- **Note:**
  A note is a simple text note with a subject and main content.
- **Attachment:**
  An attachment is any external content that can be linked to the time-

line. In addition to the four item types described in this section, one can add information from a barcode using the internal camera or a link to an URL[1].

To use these types of data in our application, all we had to do was to integrate with the device' native systems. Second, several of the related work done in the theme of reflection, learning and collaboration have used the same types of data with success, for instance [50, 18, 41, 45].

6.2.4 System Collected Attribute

The system captures two attributes, time and location, and connects it with the user collected events so they can be placed accordingly in the timeline.

6.2.5 Emoticons

To capture the users' feelings about the events, the application provides easy access to emoticons. Emoticons are small icons that represents the users feelings defined by [20] as *"Emoticons are a simple and concise way of visually showing user the general emotion behind a feedback tag"*. The prototype application supports 4 different emoticons:

Like 
Cool 
Dislike 
Sad 

In the first version of the prototype, the user can use these emoticons to assess an event. As we move forward into realizing the collaborative aspect of the application in the next iteration, the emoticons create a possibility for users working together to asses each others events to further enhance the feeling aspect of the reflection process as seen in Boud's model presented in Figure 1.2. By encourage user to assess events we want to achieve the second level of reflection, as stated by Fleck and Fitzpatrick in [19].

[1]Uniform Resource Locator

6.3 Design Choices

Through several steps of the development, different design choices were made that shaped the application. Before the actual development began, we went through a design process that resulted in several design choices. In this section we will elaborate on how these steps culminated into these design choices.

6.3.1 Visualizing and Populating the Timeline

The first design choice was to figure out how we wanted the applications look and feel to be. When designing for a mobile device there are several limitations that is important to take under consideration. For instance screen size, user interaction, connectivity to the Internet, built in functionality and hardware. During the paper prototyping, different ideas were explored and many were thrown away. We looked at many distinct design possibilities, most importantly on how the timeline should be represented. Several options were discussed, amongst them was representing the timeline as a line with connected annotations, which is what most people relate to when they hear the word "Timeline". But what we decided was to use the events annotated with icons to build the timeline. The advantage of this approach is that the timeline is not locked to a time interval, but rather build itself as new events are annotated:

An event is captured at a specific time. We understood that by placing the events on the screen according to time, the events themselves would create the timeline. An example of the idea of representing the timeline this way can be seen in Figure 6.2.

Also, we had to decide whether we wanted the timeline to be represented vertically or horizontally. Several usages of timelines already exists. Some of these were identified in Chapter 4, Related Work, and summarized in Table 4.2. Many of the tools examined uses similar layouts, spreading the time-span out in a horizontal manner and organizing the content vertically such as the web application created in "Tools for Students Doing Mobile Fieldwork" [41]. Another example can be seen in Figure 6.3 where the time is showed horizontally while the content is stacked on top of each other according to when each happening took place. Even if they did not take place at the exact same time, for the sake of room on the screen, they are stacked vertically.

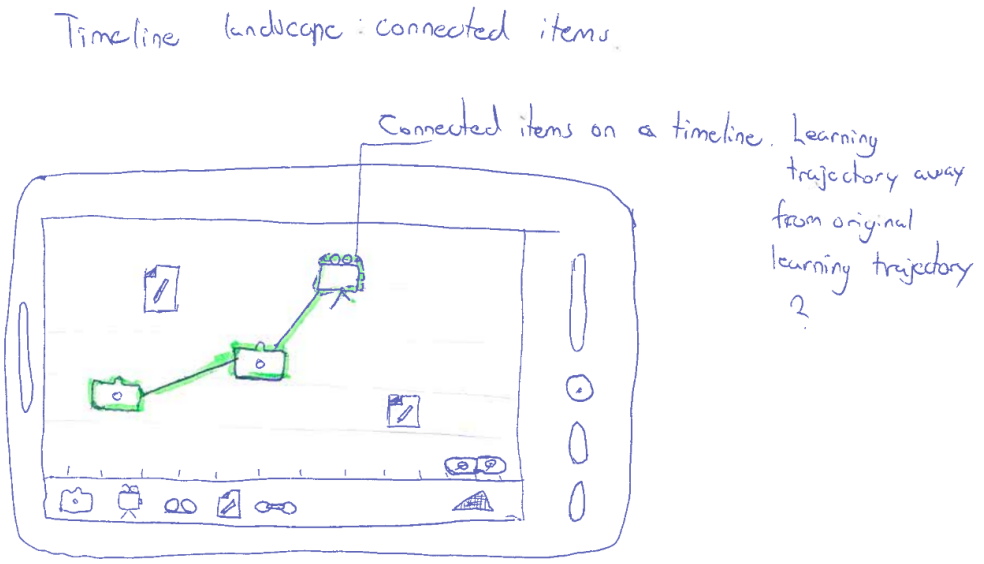


Figure 6.2: A paper prototype representing how different events captured at different times builds a timeline

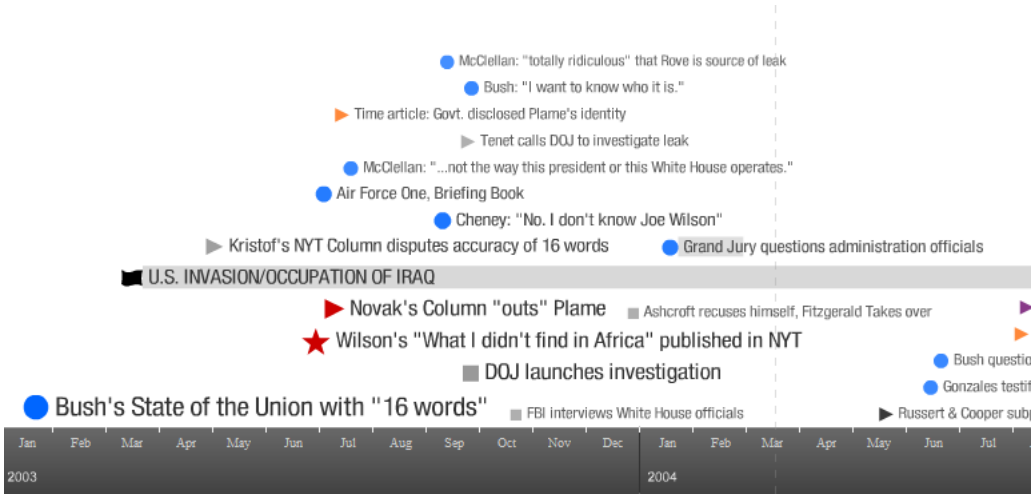


Figure 6.3: A screen shot taken from the TimeGlider [34] timeline Web tool showing a horizontal time span and vertically placed content according to time

Several of the tools use this kind of visual representation of their timeline, and as this also fits for our application, out of all our ideas, we choose to create a horizontal timeline, with optionally vertical stacking of content if the content is captured at a similar time. For example, all annotations captured within the same hour stack vertically if the hour view is chosen. This is shown in another illustration from our paper prototyping (Figure 6.4).

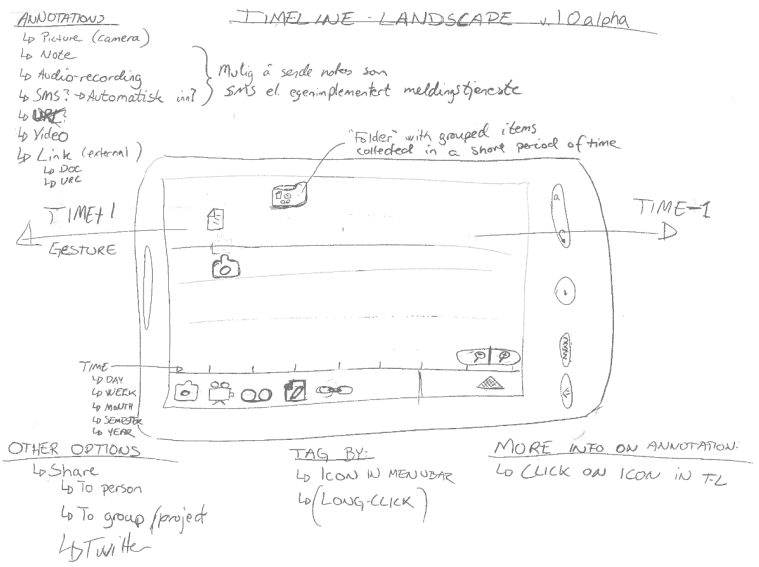


Figure 6.4: An illustration from the paper prototyping, showing the application in landscape mode with a horizontal timeline

Another important decision to make was the time span of the timeline. Should it be an hour, a day or a week? As it is hard to predict the time granularity for a specific experience, we wanted this to be the choice of the user. The authors of "ButterflyNet: A Mobile Capture and Access System for Field Biology Research" [52] are also emphasizing the need to provide the user with a way to adjust and visualize the granularity of collected data in capture and access systems. The time span navigation is shown in figure 6.5.

To zoom in the user simply clicks on the desired zoom level, e.g. clicking on a date in the month view(the longest time span), zooms in to the view of that date. Clicking on an hour in the day view zooms in to the view of this hour. The hour view is the lowest time span in the application. To zoom out, we have added a zoom button in the lower right corner, as well as implementing the zoom out functionality to the Android back button(see Section 6.3.3).

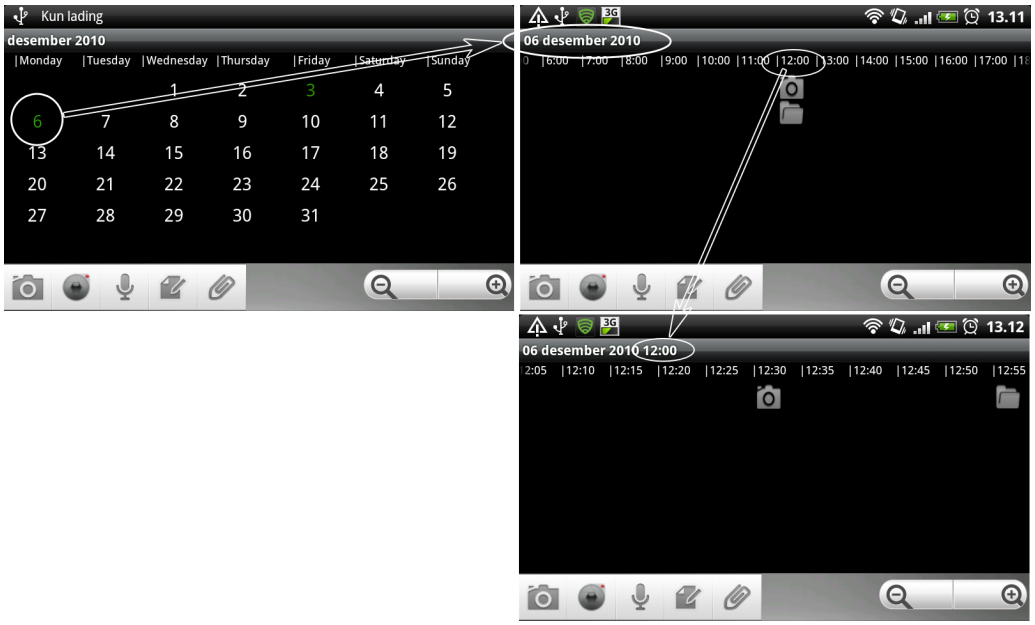


Figure 6.5: The navigation between different zoom levels in the timeline

6.3.2 Utilize Native Android Applications

Early in the design process we were planning to integrate the timeline with social media like Facebook or Twitter but as we started to do some research, we found out that this type of integration already existed (For instance Memolane [2]) and for our part it would be more like a fancy feature then something relevant for our research questions.

Even if we discarded the social media integration, much of the feedback we got on our first concept was about letting the user add separate content like stored pictures, notes, links or e-mails to the timeline. This would open the possibility of timeline content collected outside the context of the application, which could make the application more applicable in the case of not forcing the user to always collect data with the application.

This design choice was realized by integrating the application with the Android operating system, making the Timeline application available as a choice when sharing content from different native applications in the system. This is a feature the Android operating system provides and this makes it possible to share from one application to another. Figure 6.6 shows how this is done in practice.

[2] `http://www.memolane.com`

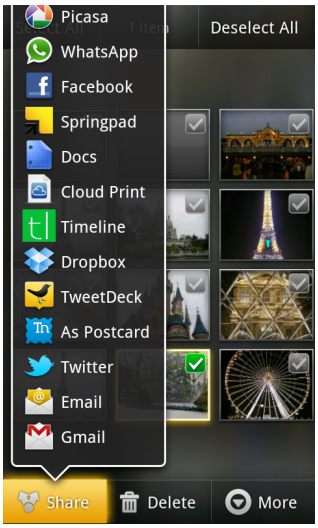


Figure 6.6: The application is integrated with the system, allowing content to be shared between the system and the application. In this case, sharing a photo from the gallery to the Timeline application

6.3.3 Menus and Navigation

The navigation of the application user interface is important for giving the user a good experience. As this is something that is common among all mobile applications, there are many guidelines available to assure that the look and feel of the navigation is good. The Android [3] operating system developed by Google Inc [4], has its own user interface guidelines that we have followed when creating this part of the application. The guidelines we followed are [25] (UI guidelines of creating menus and navigation patterns by Google) and [37] (UI guidelines presented in terms of the official Twitter application for Android, by Twitter [5] and Google).

Dashboard

The dashboard showcase the features, as well as providing easy access to the common functions of the application. This is one of Google's proposed design patterns for Android [37].

[3] `http://www.android.com`
[4] `http://www.google.com`
[5] `http://www.twitter.com`

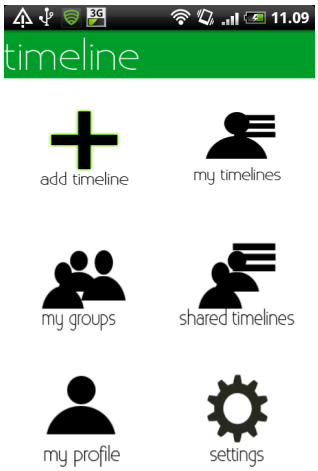


Figure 6.7: Screenshot of dashboard after iteration 1

We hope the dashboard encourages the user to explore and use the features of the application.

The dashboard in iteration 1 consisted of these options:

- add timeline: Lets the user create a new timeline.
- my timelines: Lets the user browse and open private timelines.
- shared timelines: Lets the user browse and open shared timelines (not implemented in iteration 1).
- my groups: Lets the user create and add users to groups (not implemented in iteration 1).
- my profile: Lets the user see the profile he is working on in the application (Google Account [6], not implemented in iteration 1).
- settings: application settings (not implemented in iteration 1).

Tool bar

When the user creates a new timeline, the timeline view is shown. At this screen, the tool bar of the application is always visible. From this bar the

[6] `https://www.google.com/accounts/`

user can add data to the timeline, as well as zoom in and out on the timeline itself. A screen shot of the tool bar can be seen in Figure 6.8

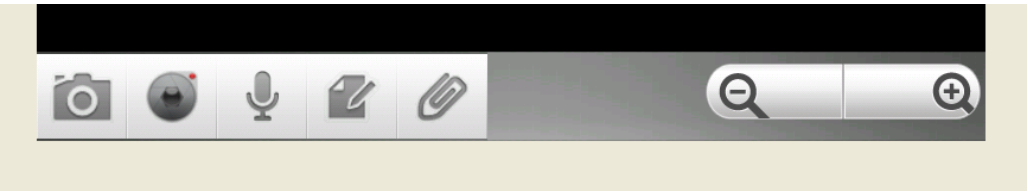


Figure 6.8: The tool bar in the timeline screen of the application

Event view

Figure 6.9 shows a screen shot of the event view in the application. The event view is presented with the different item types stacked on top of each other, and the user can scroll up and down to see each one. The top of the view displays the date and time, as well as location information of when and where the event was created. On the right side the user has several different options. Amongst others to add additional items or to share the event with other users. The bottom of the screen shows an assessment of the event in terms of emoticons added to it.

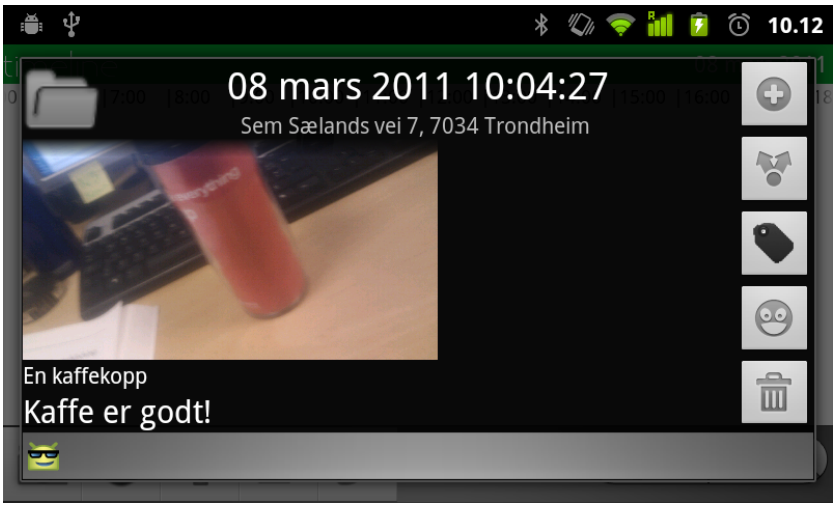


Figure 6.9: A screen shot of the event view in the application

User Navigation

The navigation of the application can be split into two parts, navigating the different menus and navigating the timeline. In the menus, the user navigates by clicking both software and hardware buttons. A software button is a button that is placed in the screen(programmatically), while hardware buttons are physical buttons on the device. Standard hardware buttons on Android devices are back, home and menu. We will utilize these standard buttons by adding additional functions to the current screen using the menu button [7], and returning to the previous screen by using the back button. This saves precious screen real estate[8] which can be used to display the timeline data in a better manner. In the timeline view the user, as well as clicking buttons, can use the swipe movement on the screen to look at different intervals of time on the screen. This can for instance be the next hour, next day, next month depending on the granularity the user has chosen for the timeline.

Long pressing an item will bring up extra options to perform on that specific item, such as delete or edit. This is normal behavior for Android applications, and similar to a right-click context menu in a desktop operating system. [9]

6.4 Client Architecture

The architecture of the client is basic. It is a model view-adapter (MVA) architecture, meaning that the view (e.g. user interface) is completely separated from the model (e.g., data structures, storage, domain models). They are instead connected through adapters, that has the responsibility of (to a certain extent) the business logic and grinding of the data so that the view can show it. The advantage of this architecture is that the view is oblivious of the model so that dependencies can be avoided. Also, with this separation, several different views can use the same models without having to change the models, just relying on the adapters. The MVA architecture is a standard way of implementing Android applications.

[7] Android options menu - `http://developer.android.com/guide/practices/ui_guidelines/menu_design.html#options_menu`

[8] Screen real estate `http://www.usabilityfirst.com/glossary/screen-real-estate/`

[9] Android context menu - `http://developer.android.com/guide/practices/ui_guidelines/menu_design.html#context_menu`

Figure 6.10 illustrates the MVA architecture of the first prototype. The views consists of several activities. An activity is the component of the Android framework that provides a screen which the user can interact in order to do something. Several adapter objects exists as a bridge between these activities and the underlying data. The adapters provide access to these data items. The model consists of the code that takes care of storing the data. The domain models is the object representation of the data, e.g., a note object and its attributes like content, creator and title. Content providers [10] are specific for the Android framework and allows application to share data between each other. This API also provides interfaces for easy interaction with databases. The content providers are used to access the databases and perform actions on them, as for instant inserting, querying and updating content. At last we have the database managers that is responsible for setting up the storage and to give the application access when needed.

6.4.1 Domain Models

The domain models of the first prototype are:

- Experience
- Event
- EventItems: Picture, Note, Video, Audio

The Experience contains several Events. The Events functions as a container for different items and has attributes like location coordinates, creator and date of origin. An Event can contain one or more EventItems. An EventItem can be one of the types: Note, Video, Audio, Picture, that all, except the note, contains an attribute linking to a file. Basically, this means that the models are connected in a hierarchy sharing several attributes. This is illustrated in Figure 6.11. This kind of hierarchy is used so that it will be easy to extend the application with new types of Events or Items as all the common attributes of these are available in the BaseEvent or the EventItems classes. The Event class also contains a list of Emotion objects that are the emotions that users connect to a specific event.

[10] Android's description of the ContentProvider framework: `http://www.developer.android.com/guide/topics/providers/content-providers.html`

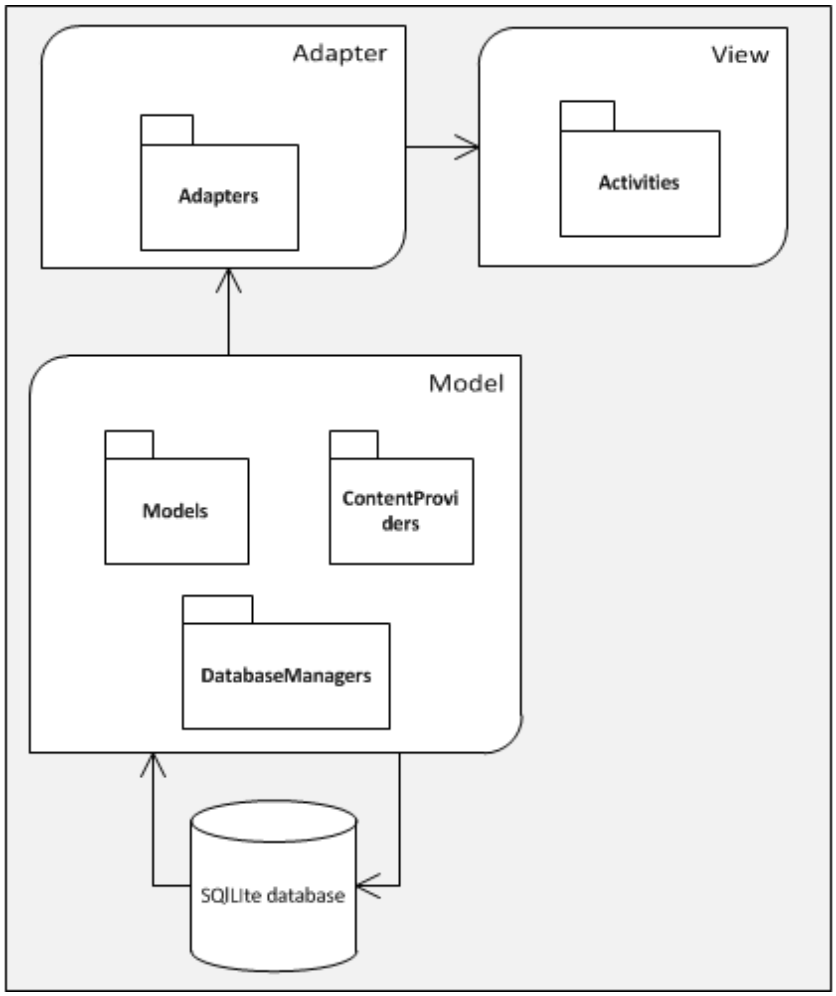


Figure 6.10: Model View Adapter Architecture of the Client Prototype

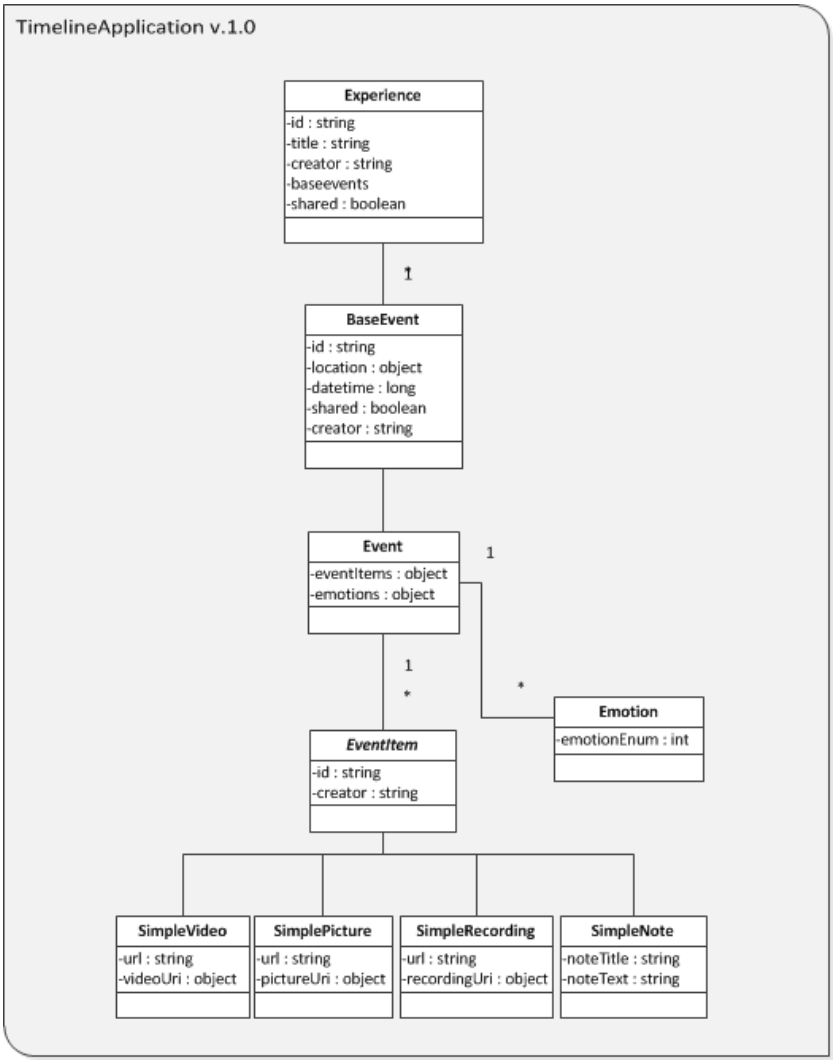


Figure 6.11: Client UML diagram in iteration 1

6.5 First Iteration Evaluation

In this section we will describe how iteration 1 of the development was evaluated, with concern on the evaluation goals, evaluation method and evaluation results.

6.5.1 Evaluation Goal

The goal of this evaluation was to answer the questions presented below:

- How is reflection supported by the application?
- How should we change the current functionality to better support reflection?
- What should we implement in the next iteration to further enhance the application and meet our overall goal.

Also, we had some ideas that we wanted feedback on to decide whether or not we should go further with them. These ideas are presented in Table 6.1.

6.5.2 Evaluation Method

The evaluation was an expert evaluation with a Post Doc from NTNU that is a member of the EU-funded project MIRROR. The evaluator is an expert in the field of reflection in learning, and also had some previous knowledge of our thesis as a member of the FABULA research group [30]. However, the evaluator was never directly involved in our thesis.

The evaluation was conducted as a type of expert walkthrough as described in "Interaction Design - Beyond Human Computer Interaction" [40]. The main difference from this evaluation method compared to the one in the book is that we were not conducting a evaluation on the user interface of the application. Our focus was rather on how the application can support reflection in learning.

We began with a presentation of the main concepts of the application and continued with a walkthrough of the application used in a specific scenario, showing the possibilities and limitations of the application at its current state (Appendix A). We then had an open discussion with the evaluator on the problems, limitations or advantages that the evaluator had noticed. At this

Number	Name	Description
1	Combine timelines and events	Give the user the possibility to combine several timelines or create a new timeline with starting point in another timeline.
2	Distributed timelines	Several users can work together on the same timeline, using a server for synchronizing the work done between involved users.
3	Tagging	Make it possible for the user to tag events in a timeline so that they will be easier to organize and find later.
4	Link items based on tags	Use tags to visualize connection between events in a timeline. If there are different themes of events in a timeline, we can visualize their connection.
5	Identify events based on assessment	Use the emotion assessments to separate events from each other in a timeline. Good, Bad, Cool, Sad events. See Emoticons in Section 6.2.5.
6	Facilitated learning	Support for a feature where a user can insert questions, a quiz, an assignment or other learning objectives into another user's timeline to facilitate learning.
7	Present the timeline in other medias	Give the user the possibility to export timeline to other medias, typically a web resource. For instance presenting the timeline in a browser.
8	Integrating	Integrate the application closer to the Android system, for instance SMS straight into timeline. Integrate with social media like Facebook or Twitter. Integrate with resources like Dropbox [10] or source control systems.

Table 6.1: Different ideas presented at the expert evaluation

point in time we had not presented any of our thoughts or ideas about the further development of the application. We also held back on "defending" problems or limitations noticed that we knew where going to get fixed or implemented in the future. This was because we wanted a nuanced view of our application without the developers steering the discussion in any way.

Next we asked the evaluator to present her initial ideas on how we could improve the application with focus on reflection in learning, still without presenting any of our own thoughts about the subject.

After this part, we had prepared a set of ideas and possibilities that could be used for enhancing the use of the application as a tool for reflection. Many of these were similar to the ones the evaluator already had presented in the previous part. We went through each of these ideas and discussed with the evaluator if they where feasible, any good or needed some improvement. The feedback on the application from the evaluator showed that we shared a lot of ideas on how we could further develop the application to better support reflection in learning.

6.5.3 Overall Feedback

The evaluation itself gave us the impression that the evaluator was overall satisfied with the functionality of the application when thinking in terms of reflection. As an overall feedback on the already existing functionality, the evaluator said that the choices we had made on the data and the representation was satisfying as it allowed several types of data to populate an event and thus creating an environment for reflection.

However, for the application to support reflection in group work, the current state of the application seemed rather thin when it comes to the collaboration aspect. As of that time, as the evaluator emphasized, the only collaborative aspect of the application was that it was possible to link (to a note annotation) e-mail, web-pages and such into the timeline. As the evaluator expressed: "*If this is supposed to help the group it isn't too much use if all the information is stored separately on each individuals device*". The evaluator then emphasized the importance of supporting collaboration in the application if we were to use the scenario we had presented during the walkthrough. At this point of the evaluation we emphasized that the goal of the first iteration was to develop the applications individual requirements, this meaning creating the features making it possible for a user to individually collect data and getting them represented in a timeline. And also, that the goal of this

evaluation was to get feedback on ideas on how to perform the collaboration part of the application in the next iteration.

At the end of the evaluation, the evaluator gave us some general advice that she thought would be useful for us to remember during iteration 2:

- To look at things in comparison with each other could be very useful. As shown in the mid circle of Boud's model of the reflective process(Figure 1.2)
- Users often tend to use functionality different than intended by the developers.
- Be aware of what aspects of collaboration that would be important to cover and reflect on, and choose what's available in the given range of time you have available.

6.5.4 Feedback Related to the Current State of the Application

On the overall design of the application we got some feedback on how the timeline itself was presented. The evaluator pointed out that the user should be familiar with navigating a calendar on a smartphone, and that it was important to make it so intuitive that the users actually used the application instead of for instance pen and paper.

The evaluator was also interested in knowing how we could use the application to show sharing patterns between users. Who shares what with whom, how much have a user shared and so on. This is something the evaluator emphasized could be a *"starting point for discussion about reflection on the process a group has been through in their collaboration"*. The relevance of this, the evaluator pointed out, is based on what the group want to use the application for. Should the application be a tool for helping the group do their work, or should it be a tool for tracking the process a group goes through when working on a project? It is important for a group to agree on the usage of such an application before the work begins, so that everyone agrees on how to utilize it. This may be hard to take into account, because individual users tends to use the same tools in different ways.

Another interesting discussion also occurred. Basically it was an issue that in some cases the timeline view may not be the best way to represent the experience. This would be for instance when an experience occurs over a

very short period of time so that the time aspects becomes less important. A related issue is also evident in [52] who states that it is difficult to visualize the data when much information is gathered in a short time frame. One suggestion here was to create a location based view to show the events in an experience based on *where* they were captured. In some cases, as the evaluator expressed, this could be more useful for the users because they then could look at an actually map of where they performed work and collected information. After some discussion about the theme, we concluded that it would be a nice feature and in some cases also relevant for reflection, but we had to consider how much effort it would take to implement it before we could take a decision.

6.5.5 Feedback on Ideas

In this section we will list the ideas presented in Table 6.1 and present the feedback and discussion on each idea.

- Idea number 1 - Combine timeline and events

The evaluators feedback on this idea was mostly related to that it could be a nice or helpful functionality, but not so much for improving reflection. It should not be prioritized.

- Idea number 2 - Distributed timeline

This was the aspect of the evaluation that received the most attention. If the application is going to fulfill its role as a mobile tool for supporting reflection in group work, some sort of distributed work must be supported. The discussion is touched on earlier in this evaluation but to summarize what was discussed:

Several user should be able to work on the same timeline distributed. Functionality to synchronize the work should be present, either automatically or by choice of each user (clicking some sort of synchronization button in the application). The evaluator said that this is relevant not only because it makes the application more usable, but also in terms of reflection because it gives the users the opportunity to choose what to share to whom and when. If you later on can go back and look at what has been shared, it becomes a process where the user have to *step back* to see if information has been relevant or not, or if something not shared maybe should have been shared.

The data being shared is also relevant here, because when we look back at Boud's model on the reflective process [7](See Figure 1.2) it is the ideas, behavior and feelings connected to an experience that are reflected upon. Thus it is important that the data being captured and shared includes these items. The evaluator was interested in how we had made emotions (See Section 6.2.5) a part of an event, where the thought was that user could look at other users events and asses them based on how they feel about the event. Further the evaluator expressed some thoughts on better capturing this "feelings part" of the application with for instance collecting the average mood of a group in a timeline, and in some way visualizing this back to the users. In a group process, if you can look at the work done and compared it to how the group members feels at the same time, it helps the user to reflect on why the process may have unfolded as it did. She meant this should be something to consider to better capture the reflective process presented in Boud's model.

- Idea number 3 & 4 - Tagging

The evaluator thoughts on this idea were mostly that it would be useful for the user to organize the content of the timelines in the application, but for better supporting reflection, it should be more evolved than just the possibility of tagging an event with a suiting tag to find back to it easily later. An idea we discussed went more into how we can aggregate the data based on tags. That we for instance could create a new timeline based on one or more types of tags. As the evaluator expressed: "*I can imagine that the computer structure lying beneath should give possibilities of showing different types of timelines based on different criteria, for instance tags*".

- Idea number 5 - Identify events based on assessment

On the issue of using the assessments as a type of tag to identify events and create new timelines, the evaluator expressed that this was only important if the timeline was considered in a historical aspect. For instance, if all user mark an event as unimportant at one point in time, but then later goes back and changes this assessment to be more important. The evaluator felt it could be interesting to log these changes; *"You can imagine that the history of these changes enlightens the reflection in the group.* But as for the use of assessments in the same way as tags, the evaluator expressed that tags in itself was more suiting.

- Idea number 6 - Facilitated learning

From the experience of the evaluator, it is seldom that a group wants a supervisor or a facilitator included in the group process sphere. The groups want it to be informal and control it in their own way. As soon as other participants get access, this changes. So a big issue here is to decide what to share with the supervisor, without it having an effect on the sharing inside the group. Another option would be to give the advisor role to a member of the group, making that member responsible for facilitating on the timeline. The evaluator also expressed that this seemed a bit out of our scope.

- Idea number 7 & 8 - Integrate with other media

The evaluator viewed integration with other systems as interesting and that it shows the flexibility of a system, but she also raised some concerns. Integration with social media like Twitter or Facebook could cause problems with filtering all the information, as much could be of lesser relevance. By integrating with Dropbox or source control systems, the application would tend more towards a project control system and then again raise other issues. If this where to be implemented, the evaluator recommended to take under consideration two questions; what is the utility value during the project and what is the utility value for reflection. *"A lot of things can be useful for reflection, but there is limits to what you can create in one tool and what you should do based on your research questions".*

6.5.6 Feedback on Ideas from the Evaluator

As mentioned previously, the evaluator also had some ideas on how we could further develop the application. The idea that received the most attention was the idea of having some kind of mood map available in the application. The evaluator liked our feature of having emotions of the users connected to events, and could see this further developed in the context of a multi-user usage of the application. She would like to know if seeing the events connected with positive emotions was shared more often than negative ones. This could show a pattern of how events are shared, and this pattern could further enhance reflection around the process of sharing events between users. A concrete suggestion from the evaluator was to create a kind of mood map to put in the timeline that could show the feelings of the users on events - over time, and compare the mood to the shared events. As the evaluator expressed: *"Positive and negative feelings are one dimension, and what you*

have shared is another dimension. Connecting these two dimensions and look for a pattern could be interesting".

6.5.7 Refinement Based on the Evaluation

Based on the feedback from the evaluation, a review of the application was conducted both for the current functionality and to decide how we should prioritize developing new functionality. The full list of the requirements identified after the evaluation of the work in iteration 1 can be found in Table 6.2.

Table 6.2: Requirements prioritized based on feedback of iteration 1.

ID	Name	User Story	Prio	Est.	ROI
CR2-2	Share data	As a [user] I want to [be able to share data to a user I am connected with]	10	60	0.25
CR2-1	Compare timelines	As a [user] I want to [compare my timelines with others timelines] to see relations, differences and similarities.	8	40	0.13
CR2-4	Group timeline	As a [user] I want to [create a group timeline] where I can share my events with a group	10	20	0.13
CR2-3	Connect users	As a [user] I want to [connect to other user so I can share timeline content with them]	8	30	0.11
CR2-10	Asses shared event	As a [user] I want to [be able to asses an event shared to me through another users timeline]	9	10	0.10
CR2-5	Web view of timeline	Export a timeline to an XML based view of timelines: `http://timeline.codeplex.com/`	6	30	0.09
CR2-6	Tagging events	As a [user] I want to [be able to tag events]	8	5	0.08
CR2-7	Visualize tags	Events with different tags should be visualized in way that separates their look. Ex: color outline on icons, lines between events with the same tag	8	5	0.08

Continued on next page

Table 6.2 – continued from previous page

CR2-12	Show mood view	Based on assessments, there should be possible to show a mood view of the timeline. Showing ”mood swings” of the timeline period.	5	40	0.08
CR2-8	Add tag	As a [user] I want to [mark an event with a tag]	8	1	0.08
CR2-9	Create tag cloud view	Create a view that can show a tag cloud	6	20	0.08
CR2-11	Lock Events	As a [user] I want to [lock my events] so they are private.	7	5	0.07
CR2-13	Filter timelines based on tags	A timeline should be created based on a tag. All events with the same tag should be visualized in timeline when the user wants it.	7	4	0.07
CR2-17	Create tags	As a [user] I want to [create different tags that can be used to tag events with]	7	2	0.07
CR2-16	Add several tags	As a [user] I want to [mark an event with several different tags]	7	1	0.07
CR2-19	Create a tag cloud	Create/use an algorithm for creating a tag cloud based on the amount of tag used	6	8	0.07
CR2-14	Facilitation	As a [facilitator] I want to [annotate facilitation in the timeline] (e.g questions, quiz, assignments)	5	20	0.06
CR2-24	Create new timeline from event	As a [user] I want to [extract an event to a new timeline]	5	2	0.05
CR2-15	Move an event	As a [user] I want to [extract an event to another existing timeline]	5	2	0.05

Continued on next page

Table 6.2 – continued from previous page

CR2-20	Create a new timeline from item	As a [user] I want to [extract a item to a new timeline]	5	2	0.05
CR2-23	Move an item	As a [user] I want to [extract a item to another existing timeline]	5	2	0.05
CR2-21	Separate tags	Items in an event should be able to be tagged individually	5	1	0.05
CR2-22	Remove tags	As a [user] I want to [be able to remove tags that i don't need anymore]	5	1	0.05
CR2-25	Edit tags	As a [user] I want to [be able to change tags later on so that it fits better]	5	1	0.05
CR2-26	Change tags	As a [user] I want to [change the tag on an event]	5	1	0.05
CR2-27	Integrate with Facebook	As a [user] I want to [see the Facebook status of my connected users in my timeline]	4	20	0.05
CR2-28	Integrate with Twitter	As a [user I want to [be able to follow ha twitter-hashtag with my timeline so that tweets automatically appears in my timeline	4	20	0.05
CR2-29	Integrate with SMS	As a [user] I want to [get SMS from group members into the timeline], so that I see the correspondence with the group members	4	15	0.05
CR2-18	Add items at a specific time	As a [user] I want to [be able to add an item at a specific time without adding it to an already existing event]	4	3	0.04

Chapter 7

Iteration 2 - Collaboration

In this chapter we will focus on expanding the application to be a collaborative tool, by facilitate sharing of the collected events. This involves implementing additional features to the application itself, as well as creating a common server infrastructure and Web services for the information exchange between users of the application.

7.1 General Adaptation to Support Collaboration

Before starting the work on the new server infrastructure and adaption of the client side, we had to extend our data models, as well as decide how the data exchange between the server and client was to happen.

7.1.1 Changes in the data Model

As the collaborative aspect was taken into consideration while designing the data model architecture in iteration 1, fields such as a sharing flag was already in place. We saw however that the data model had to be extended with two more models, users and group.

User and Group Models

To ease the sharing and collaboration in timelines between multiple users we introduced groups to the application. As the updated UML in Figure 7.1 shows, a group consists of one-to-many users. All shared timelines must be connected to a predefined group. The advantage of this solution as opposed to adding individual users to a shared timeline is the reuse of groups. A user might share several experiences with the same group, and having this group stored and synchronized in the application eases the creation of additional shared timelines. A group connected to a shared timeline will work in the same timeline, and users will see the content added by other users, see section 7.3.2 for more details. By showing the groups content in the same timeline we want to achieve *awareness*, as was shown beneficial in "Tools for Students Doing Mobile Fieldwork" [41], but in contrast to this mobile tool we are viewing the content captured by the group directly in the application.

7.1.2 Data Exchange

To support exchange of data objects between the server and the application an inter-operable Web service needed to be created. This Web service should provide the data using a structured language to code and decode data objects, as well as providing these data using an open protocol to transfer the data. The most common types of Web services are SOAP-based Web services and RESTFul Web services. SOAP-based Web services, also known as "big" Web services [11], provides their services through the Web Services Description Language (WSDL) and has been the standard way to implement Web services in bigger applications, typically using the SOA-architecture [1]. Through the increasing amount of smaller applications, such as mobile apps and Web sites using Asynchronous JavaScript with XML(AJAX) a more lightweight approach to Web services has evolved, called RESTful Web services. Oracle states that "Developing RESTful Web services is inexpensive and thus has a very low barrier for adoption" [11]. At the time of development Android didn't provide any SOAP library. The alternatives were therefore to write our own, or use an open source library, such as kSOAP[2]. For the purpose of our application, RESTful Web services suited our needs and was the simplest service to integrate with the Android platform.

[1] "SOA is a way of architecting the enterprise application as a set of cooperating services that all the enterprise users want"[31]

[2] KSOAP - `http://ksoap2.sourceforge.net/`

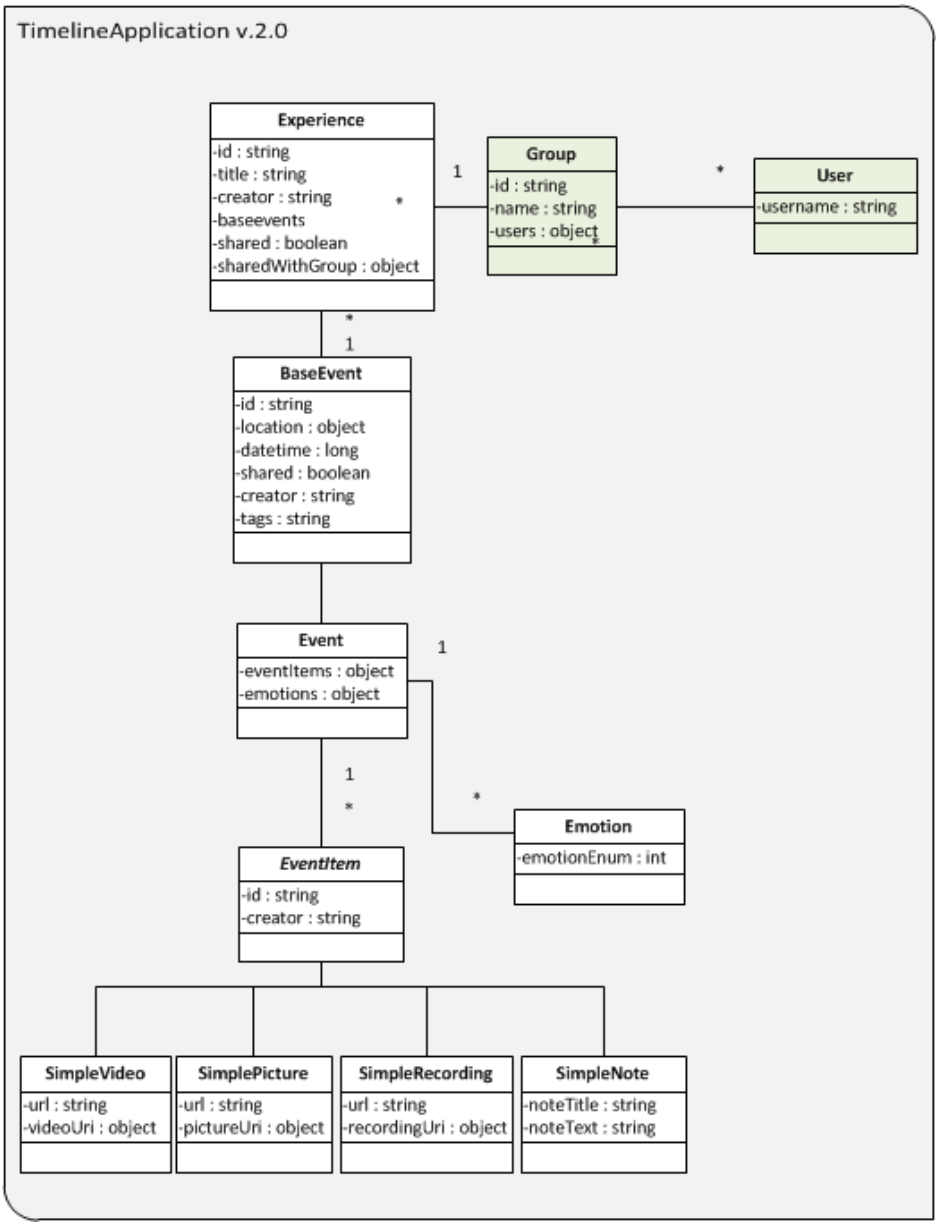


Figure 7.1: Client UML diagram in iteration 2. The User and Group model was added in iteration 2

To represent data there are typically two structured markup languages that are in common use, eXtensible Markup Language (XML) and JavaScript Object Notation(JSON). XML offers the richest, while JSON offers the lightest markup language. JSON has gained popularity in the same areas as RESTful Web services, in mobile and Web applications because of its low overhead. Especially in mobile applications, where the user can be accessing Internet using mobile networks where the user pays per byte received. These advantages, in addition to requirement WSR3-1 (Table 8.1), stated by the MIRROR Project, made JSON the choice of markup language for data object exchange.

7.2 Server Implementation

7.2.1 Server Infrastructure

To enable exchange of information across the clients running the application a central database and server providing the data was a requirement. A database and a server can in principle be set up on any computer connected to the Internet. The throughput of the server will be limited by the computing power of the server and the connection speed of the Internet link. Table 7.1 shows a few basic requirements we set to the server.

ID	Description
SR2-1	Connected to Internet 24/7
SR2-2	Reliable with an uptime $\approx$ 100%
SR2-3	Database for storage
SR2-4	Support Java 6 virtual machine (JVM)
SR2-5	Quick to set up
SR2-6	Free of charge

Table 7.1: Requirements to server

Alternative 1: Set up a computer to act as server

Setting up a computer as a server would require to acquire the necessary hardware and software to run an application server. This could be done with a relatively simple computer and Linux operating system. However, none

of us had any extensive experience with the setup, which could potentially break requirement SR 2-2 and SR 2-5. In addition came the time to acquire the necessary hardware.

Alternative 2: Cloud application hosting

As opposed to set up the computer by oneself, one can hand over this to a 3^{rd} party host, commonly named "The Cloud". Hosting on clustered hardware in the cloud offers a reliable server, dynamic scalability depending on the needs of the service, database management, backups and central administration of the server. There are several actors providing "Cloud Services" such as Amazon Web Services(AWS) EC2, Heroku, Rackspace Cloud, GoGrid, Google App Engine and many more. These companies offers different services in the cloud, supporting different platforms, and at different pricing schemes. The servers hosted in the cloud are typically accessed and set up through a Web interface, and a minimum of server experience is therefore required.

7.2.2 Google App Engine

Google App Engine offered an interesting free option:

> *"Run your Web apps on Google's infrastructure. Easy to build, easy to maintain, easy to scale. Google App Engine enables you to build and host Web apps on the same systems that power Google applications. App Engine offers fast development and deployment; simple administration, with no need to worry about hardware, patches or backups; and effortless scalability"* [26].

Advantages using Google App Engine:

- Known platform(Java).
- Same platform as Android. Meant easy implementation of the Java models from the Android application.
- Known development environment (Eclipse).
- Large development community for Google App Engine and Android integration.

- Enough free storage and computing power for development and testing of the application. An efficient application on a free account can use up to 1GB of storage and up to 5 million page views a month[3].
- Easy setup using our Google Accounts.
- Configurable, easy to remember URL under the ".appspot.com" domain. No need to set up and remember a static server IP-address.
- No need to set up a computer running the server, and maintain and monitor that the server was running and connected to Internet.
- Google App Engine handles backup routines.
- A Web interface with database viewer and log console available from any computer connected to Internet.

Google App Engine fulfilled all requirements stated in table 7.1, and a decision to use Google App Engine as server was made.

7.2.3 RESTful Web Services

To support information exchange between the server and the application, RESTful Web services were created. Representational State Transfer (REST) is an architectural style that specifies constraints, such as the uniform interface, that if applied to a Web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web [9]. The data and information on the server is referenced in REST as resources. These resources can be accessed using simple, well-defined operations, typically through HTTP.

> *"In the REST architecture style, clients and servers exchange representations of resources using a standardized interface and protocol. These principles encourages RESTful applications to be simple, lightweight, and have high performance.*
>
> *RESTful Web services typically map the four main HTTP methods to the operations they perform : create, retrieve, update, and delete"* [9].

Table 7.2 shows a mapping of HTTP methods to the operations they perform.

[3]Google App Engine - `http://code.google.com/appengine/whyappengine.html`

HTTP Method	Operations Performed
GET	Get a resource
POST	Create a resource and other operations, as it has no defined semantics
PUT	Create or update a resource
DELETE	Delete a resource

Table 7.2: Mapping HTTP Methods to Operations Performed [9]

Required Web Services

Before starting the implementation of the RESTful Web services a set of requirements of which services that were needed for our application to achieve collaboration and sharing between several devices was elaborated. The requirements in Table 7.3 were the ones we found as a minimum to achieve the desired functionality for user and group management as well as synchronization of timelines.

ID	Type	Description	Input	Output
WSR2-1	GET	Get all experiences of a user	username	List<`Experience`>
WSR2-2	PUT	Add all experiences from a user	List<`Experience`>	void
WSR2-2	PUT	Add an experience	`Experience`	void
WSR2-3	PUT	Add an event	`Event`	void
WSR2-4	GET	Get all groups	none	List<`Group`>
WSR2-5	GET	Get all groups a user is member of	username	List<`Group`>
WSR2-6	GET	Get a group	groupID	`Group`
WSR2-7	GET	Get all users	none	List<`User`>
WSR2-8	PUT	Register a group	`Group`	void
WSR2-9	DELETE	Remove a group	groupID	void
WSR2-10	PUT	Add a user to a group	groupID, username	void
WSR2-11	DELETE	Remove a user from a group	groupID, username	void confirmation
WSR2-12	GET	Get a user	username	`User`
WSR2-13	PUT	Register a user	`User`	void

Table 7.3: Required Web services. Objects are represented in JSON

Web Service Implementation

The technical details of the implementation of the Web services and an example of a Web service can be found in Appendix B.1 and B.2.

7.2.4 Using the RESTful Web Services

Since the REST architecture is based on the well defined and well known HTTP protocol standards, making the requests are straight forward, and supported by a wide range of platforms and applications, including the Android API. The code examples below shows the steps to register a group on the server.

```
1 public static void putGroupToGAE(final String jsonString){
2     final HttpHost targetHost = new HttpHost(Utilities.↩
          GOOGLE_APP_ENGINE_URL, 80,"http");
3     // Using PUT here
4     final HttpPut httpPut = new HttpPut("/rest/group/");
5     makeJSONHttpRequestContentTypeHeader(httpPut);
6     sendJSONTOGAEServer(jsonString, targetHost, httpPut);
7 }
```

Listing 7.1: Example of method using a RESTful service

```
1 {"id":"ec3d391b-0259-4f12-bdc6-715f273734ca",
2 "members":[{"username":"anderskri@gmail.com"}],
3 "name":"My group"}
```

Listing 7.2: The JSON with group information

The method in listing 7.1 takes a JSON String with the group information as parameter. The format of the JSON string is shown in listing 7.2.

In line 2 of Listing 7.1 a constant `org.apache.http.HttpHost` with the host information (standard HTTP port and protocol) is created. In line 4 a `org.apache.http.client.methods.HttpPut` object with the path of the RESTful Web service resource is created. This path is defined by the `@Path` annotation, see listing B.2. Then the methods calls the following helper methods:

```
1 private static void makeJSONHttpRequestContentTypeHeader(HttpRequestBase ↩
      httpRequest) {
2     // Make sure the server knows what kind of a response we will accept
3     httpRequest.addHeader("Accept", "application/json");
4     // Also be sure to tell the server what kind of content we are sending
5     httpRequest.addHeader("Content-Type", "application/json");
6 }
```

This method simply sets the HTTP headers to indicate to the RESTful service that it is a JSON we are sending, and a JSON we would like as response.

```
1  private static void sendJSONTOGAEServer(final String jsonString,final ↩
       HttpHost targetHost, final HttpPut httpPut) {
2  Runnable sendRunnable = new Runnable() {
3      public void run() {
4          try {
5          DefaultHttpClient httpClient = new DefaultHttpClient();
6
7          StringEntity entity = new StringEntity(jsonString, "UTF-8");
8          httpPut.setEntity(entity);
9
10         HttpResponse response = httpClient.execute(targetHost, httpPut);
11
12         Log.v("Put to GAE", response.getStatusLine().toString());
13         }
14         catch (Exception ex){
15         ex.printStackTrace();
16         }
17     }
18 };
19 Thread thread =  new Thread(null, sendRunnable, "putToGAE");
20 thread.start();
21 }
```

This method creates a `org.apache.http.impl.client.DefaultHttpClient`, sets its content to the JSON String with "UTF-8" encoding and executes the HTTP request on the host created in listing 7.1. To keep the application responsive and not block the user interface (UI) thread this is done in a separate thread than the UI.

7.3 Client side Adaptation

7.3.1 Communication with the Server

A new wrapper class for the server communication, `GoogleAppEngineHandler.java`, was created. This class contains static methods for getting and storing Timeline objects on the server. An example of a method in this class is shown in listing 7.3.

```
1  public static void addGroupToServer(Group groupToAdd){
2      Gson gson = new Gson();
3      String jsonString ="";
4
5      try {
6          jsonString = gson.toJson(groupToAdd, Group.class);
7      } catch (Exception e) {
8          Log.e("save", e.getMessage());
9      }
10
11     Log.i(TAG, "Saving group-JSON on Google App Engine: "+jsonString);
12     ServerUploader.putGroupToGAE(jsonString);
13 }
```

Listing 7.3: Example wrapper method

This method simply converts the object using GSON[4], a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object. The resulting JSON string is then passed to the method explained in listing 7.1.

Using the GSON library simplified the serializing and deserializing of the objects and JSON strings a lot. For most of the objects, the conversion went without problems, and was achieved with just a few lines of code, such as the example in listing 7.3. However, inherited objects, such as `Event.java` extending `BaseEvent.java` and the different items extending `EventItem.java` required some modifications. GSON supports the creation of custom serializers and deserializers, so the user can create or parse the JSON manually. As a workaround of the limitation of GSON we created custom serializers for `Experience` and `Event` and deserializers for `EventItem`. We also had to send the object as its base object, and including the class name as a property to correctly deserialize the object into the appropriate subclass. Listing 7.4 shows an example of such deserialization.

[4] `http://code.google.com/p/google-gson/`

```
protected static class EventItemDeserializer implements JsonDeserializer<EventItem> {
  public EventItem deserialize(JsonElement json, Type typeOfT, JsonDeserializationContext context)
      throws JsonParseException {

     String className = json.getAsJsonObject().get("className").getAsString();
     String id = json.getAsJsonObject().get("id").getAsString();
     Account creator = new Account(json.getAsJsonObject().get("creator").getAsString(), "com.google");
     EventItem ei;
    if(className.equals("SimplePicture")){
         String filename = json.getAsJsonObject().get("filename").getAsString();
        ei = new SimplePicture(id, creator, filename);
          return ei;
    }
    else if(className.equals("SimpleNote")){
        String noteTitle = json.getAsJsonObject().get("noteTitle").getAsString();
        String noteText = json.getAsJsonObject().get("noteText").getAsString();
        ei = new SimpleNote(id, noteTitle, noteText, creator);
          return ei;
    }
    else if(className.equals("SimpleRecording")){
         String filename = json.getAsJsonObject().get("filename").getAsString();
            ei = new SimpleRecording(id, creator, filename);
            return ei;
    }
    else if(className.equals("SimpleVideo")){
            String filename = json.getAsJsonObject().get("filename").getAsString();
            ei = new SimpleVideo(id, creator, filename);
            return ei;
    }
    else
          return null;

  }
}
```

Listing 7.4: Example of deserialization

If a new item is introduced or an item class is renamed, one will have to edit this deserializer as well. This is the downside of such a workaround, but as the purpose of this application is to provide a working prototype, we accepted this downside and used this approach.

7.3.2 Collaborative Features

Adding Support for Users and Groups

To enable sharing of content the users need to be identified. To make this seamless and easy to use we took advantage of the fact that there is a Google Account[5] registered on the Android device. The Android device requires a Google Account to be registered to be able to use many features of the device, such as Android Market[6]. Through the Account Manager in the Android API we are able to use the username(e-mail address) of the Google Account as our user identifier. The user is automatically registered to the Timeline application (if not already registered) when starting the application.

Users can select which users to add to the group using the group management view, accessible from the dashboard (Section 6.3.3). To make room for the group management at the dashboard, we had to remove the "my profile" options, as this did not serve any purpose, since the username is fetched automatically from the Google Account. For prototype purposes one can select from all other users registered to the Timeline application. The users are automatically registered to the group, and can start the collaboration instantaneously.

A user can at any time decide to leave a group. This will delete this user from the group but not the timelines gathered. These will still be available as private timelines and the user has to manually delete them. We wanted to assure that the user didn't delete his collected learning experience without knowing.

Selective Sharing

By default every event added to a shared timeline is not shared. Although we are stressing awareness in the timeline, we also want the user to reflect on what content to share. We want it to be a deliberate action to share an event with the rest of the group. The user should stop and think if this event contains information that contributes to the shared experience. This might also lead to interesting discussions if a group decides to discuss about the findings in the timeline. This was one of the issues discussed with the evaluator during the iteration 1 evaluation (Section 6.5.4).

[5]About Google Accounts: Basic information - `http://www.google.com/support/accounts/bin/answer.py?hl=en&answer=27439`

[6]Android Market - `https://market.android.com/apps/`

We also put the constraint on sharing events that once an event is shared, there is no way to "unshare" it again. If this had been allowed we would have had an inconsistent timeline. Imagine a scenario where two events have a connection where the last event happens based on the first. If the first event suddenly gets unshared, and thus removed from the shared timeline we are left with a "loose" event that is taken out of context.

The user is informed by this constraint on sharing when trying to share an event, see figure 7.2.

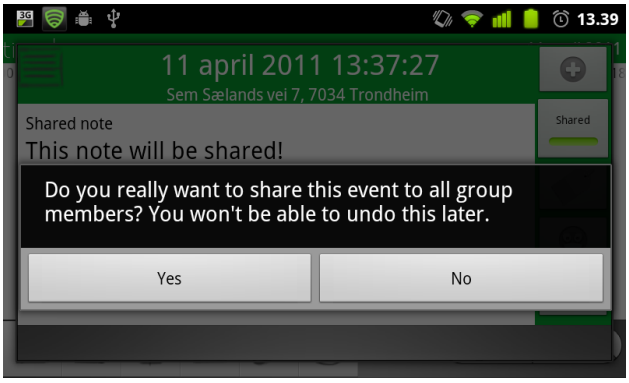


Figure 7.2: The user has to confirm sharing, as this action cannot be undone

Manual Synchronization - Automatic Push

A decision to make the synchronization manual was made, which implies that the user has to push the synchronization button to initiate a synchronization between the server and the client. The decision was made after weighting the value of having automatic synchronization against development time and effort. By using manual synchronization we still meet requirement CR2-2 (Table 6.2). To overcome some of the disadvantages of this solution we did however implement automatic push into some elements, meaning that events, items and emotions are immediately pushed, when the original event already is shared. This results in a server that always has the updated content, but the clients has to download manually using the synchronization mechanism to keep their clients updated. The synchronization mechanism will also upload any content that has not been uploaded due to disabled Internet connection. The synchronization feature was such a central feature of the application that it had to get a place on the dashboard. Since the "settings" button did not have any functionality connected, it was sacrificed.

7.3.3 Reflection Features

Based on the feedback we got in the evaluation of iteration 1, we decided to introduce some additional features that further could help in enhancing reflection.

Private Tagging

To allow the user to categorize and revisit similar events across several timelines, a tagging system was introduced in iteration 2. We see the tagging of events as an individual activity, where the users tag events with their own words, so that they will find their tagged events more easily. Each user will therefore have their own tagging database where previously used tags are stored.

We saw tagging as such a central feature of the application that the tag management was added as an item on the application dashboard, see section 6.3.3. From the tag management the user can add and remove tags and create new timelines based on events tagged with one or a number of selected tags. The latter was a result of the feedback received, in the evaluation of iteration 1, during a discussion about the tagging feature (Section 6.5.5).

E-mail Activity Report

During iteration 1 an idea(see table 6.1, idea 7) of presenting the timeline in a Web-based environment resulted in requirement CR2-5(Section 6.5.7). With the server implementation in place the foundation for such an feature was in place. However, the amount of time estimated for this task exceeded the available time. Therefore we decided to develop a solution that easily made it possible to present the timeline data on a desktop computer – by sending a users timelines to their registered e-mail(from their Google account). We call this a "Timeline activity report". The activity report includes all the users events, as well as others events shared in shared timelines. The activity report is intended to give the user a way to get an overview of ones events, and the possibility to view eg. images on a larger screen. By giving the user this option we provide an additional way of stepping back to their experiences.

At this point we got to utilize the advantages of Google App Engine – the good integration with Google mail servers. With just a few lines of code using JavaMail we were able to send out the activity report by calling a created

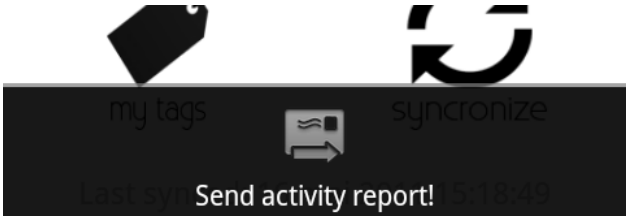


Figure 7.3: The "send activity report" option was added under the menu button at the dashboard

REST service. The formatting of the e-mail was done using HTML, and this requires that the receiver has a HTML enabled e-mail client. A snippet of a Timeline activity report is shown in Figure 7.4

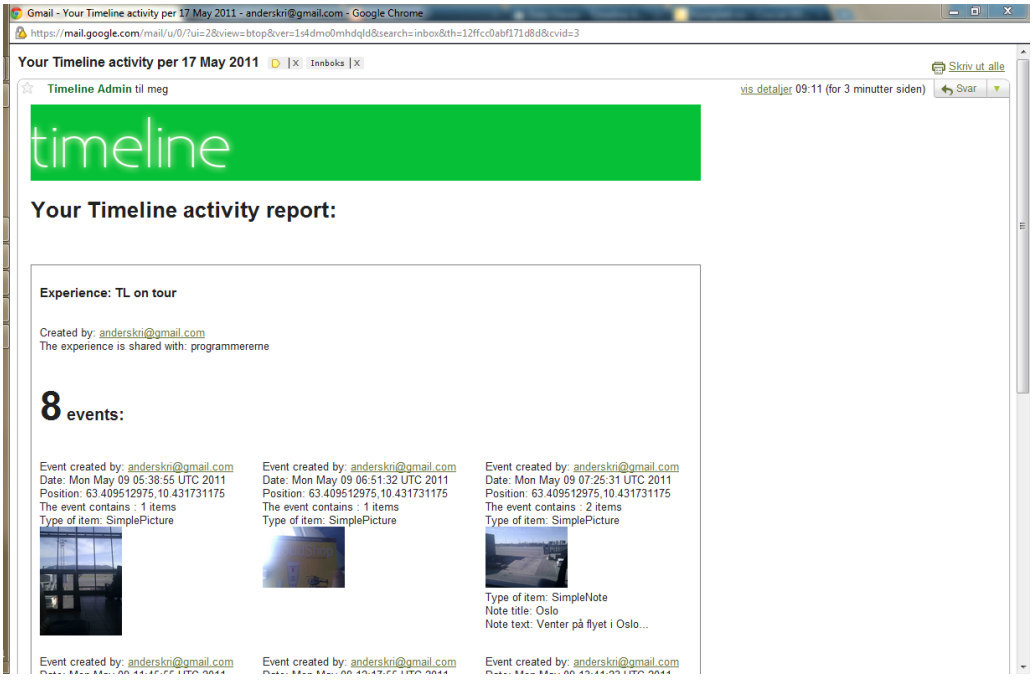


Figure 7.4: Snippet of Timeline activity report

Map View

Another interesting feature introduced in iteration 2 was the map view. This feature was proposed by the evaluator in iteration 1 (Section 6.5.4). While in a timeline, clicking the menu button will bring up the option to view the events in a map. In the map, annotation of the event will show up at the

location of creation. Figure 7.5 shows an example of a map with events at various locations. Clicking on the annotation will bring up the event view, in the same way as clicking in the timeline. The main motivation to add this functionality at this stage was to showcase how one could utilize attributes collected by the devices integrated features, such as the location.

The map view presents an alternative visualization when information is gathered in short period of time, and thus minimizes the importance of time. As long as the latitude and longitude coordinates are associated with the event and stored in the database, adding a Google Map view in an Android application was a straightforward task. Because of these two reasons we considered the map view to be valuable addition.

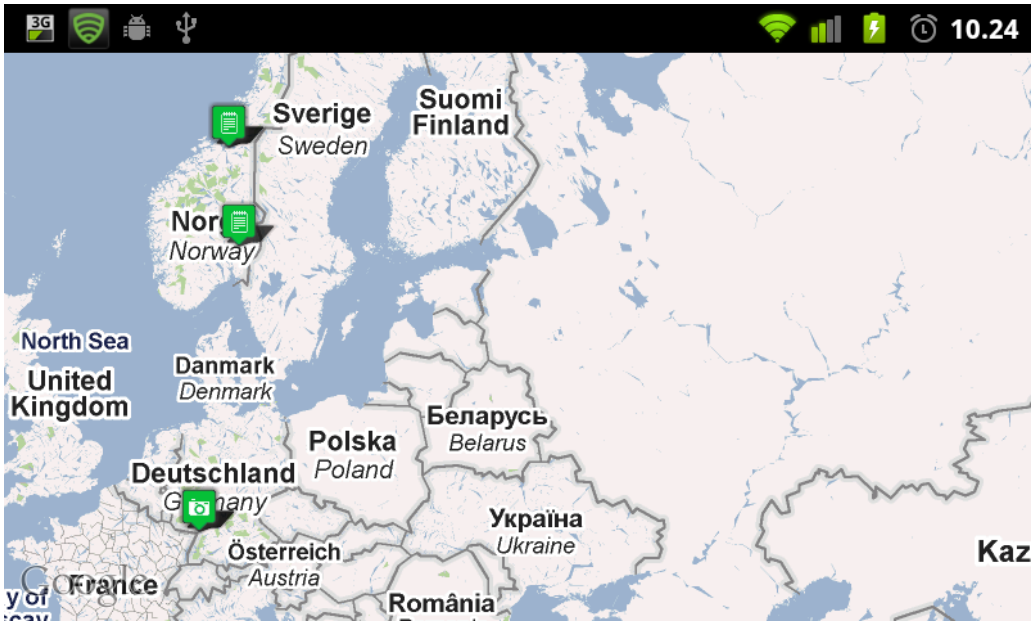


Figure 7.5: Map view showing events at the location of creation

7.3.4 Changes in user Interface

During iteration 2 a new graphical profile was introduced. We decided to go with a light theme, using white background and green as the primary color. The green color was already used in the dashboard and in the newly created logo(Figure 2.1). We decided to use this in headers and dialogs throughout the application. In addition we created a set of icons for the event annotations with a “sketch-feel”, as this was quick to create, and it underlines that this is a prototype, and an application in development.

Chapter 8

Iteration 3 - Integration

In this chapter the work performed to integrate the application with external systems will be described. We will look further into how other applications can use information collected by the Timeline application, and how the application itself can be extended to support integration with other systems.

During iteration 3 an integration between the Timeline application and an external application, MoodMap, for representing the average mood of a group was created. This chapter will highlight challenges, decisions and results of the integration between Timeline application and MoodMap.

8.1 Integration Overview

The integration that took place during the third iteration included 2 applications as information feeders, a cloud infrastructure, and 5 different visualizations of the same data gathered from the feeders. In this application sphere, the Timeline application was used both as an information feeder as well as a information visualizer. Also, the cloud infrastructure used for the integration purposes was an extended version of the same cloud infrastructure built by us during iteration 2 (See Section 7.2). The overview of the integration can be seen in figure 8.1.

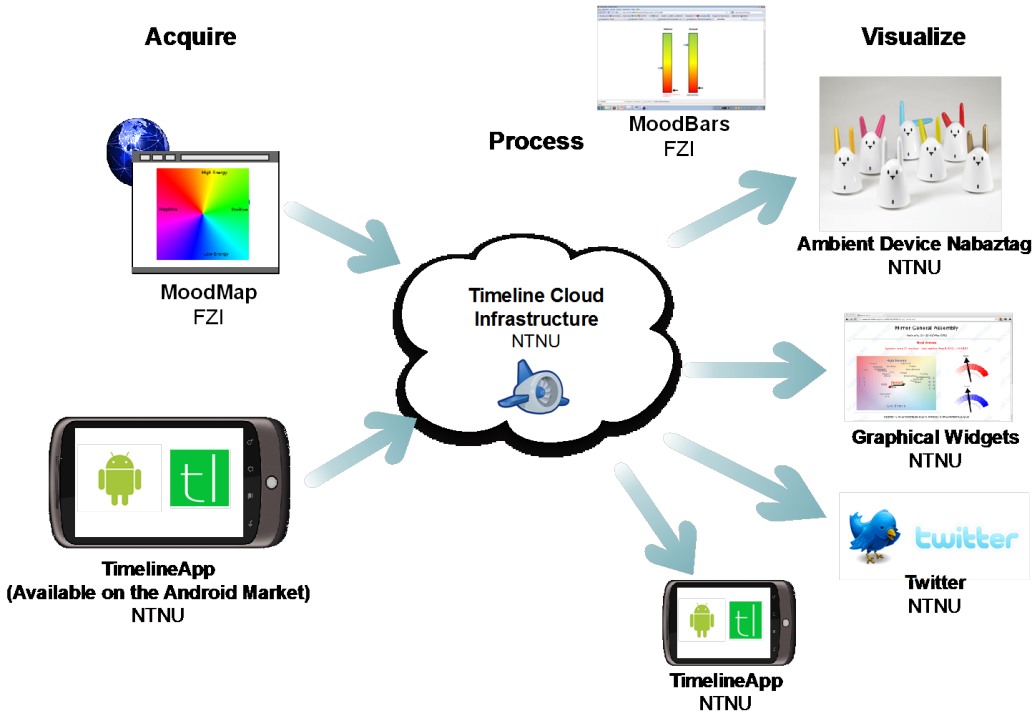


Figure 8.1: An overview of the integration performed during Iteration 3

8.2 The Purpose of the Integration

The purpose of this integration was to showcase how we could utilize the general approach we had taken to develop the application in the first and second iteration by extending it to support both interaction with other applications and new functionality. It was also to see how integration could give more value to users in terms of reflection when using the Timeline application together with other applications.

This integration prospect did not only give us the possibility to get a technical evaluation through extending our application and server for integration, thus showcasing flexibility. It also gave us the possibility to test the application in a real life setting with users familiar to the domain of learning and reflection. Access to users with such a high degree of expertise in our research domain was an important reason that made us choose to participate in this integration process.

The experiment gave us valuable feedback through informal interviews with users, usage observation and gathering of server data. The results we obtained are presented in Chapter 9.

In this section we will describe in more detail how the integration was performed and how we planned to demonstrate it to the users at the MIRROR general assembly (GA) in Karlsruhe, Germany, between the 10-12. of May.

8.2.1 Prototype Demonstration at the MIRROR GA

At the MIRROR Project GA, the objective was to show a demonstration of how different applications in the domain of reflection could work together to provide feedback to a work group based on the general energy levels and feelings of the group during work sessions. The general term of the energy levels and feelings during the demonstration was called mood. The subjects of the demonstration were the attendees of the GA. By using different applications, users could post their mood at anytime during the GA. Then, the mood average of the participants in the work group who posted their mood was calculated by our cloud infrastructure. Different applications then polled back these values to visualize the average mood of the group in several representations. All these visualizations could easily be seen by the users by being visible in the room where the work took place.

From this, the goal was to see if the general mood of the group could generate reflective thinking about the work performed during the sessions. When did the average mood of the group rise or drop - was it connected to the work being done, long sessions or heated discussions? Could the average mood create discussions about methods used in the sessions? Would the average mood give feedback to the coordinators of the meeting of how it was ran? All these where example of questions the demonstration could be able to answer. Even if these questions are not directly attached to the thesis, our work was important in this process by providing the users an application to post their mood and visualize the average mood. We also provided the infrastructure for calculating the average mood and making the integration between the applications possible.

8.2.2 Showcasing Flexibility

Another purpose of the integration was to showcase how our application could interact with other applications to demonstrate the flexibility of its infrastructure. The Timeline application was developed to be a general tool in the domain of reflective learning and through this demonstration we could evaluate how this generality provided purpose for easy integration with other

applications. In this case the usage of its data in several different applications focusing around the average mood presentations.

8.3 Requirements for the Integration

To perform the integration between the Timeline application and the external applications we composed a set of requirements in co-operation with the partners developing the other applications. The requirements are summarized in Tables 8.1, 8.2 and 8.3.

ID	Type	Description	Input	Output
WSR3-1	GET	Get current average mood of a timeline	ExperienceID	JSON with valence and arousal values
WSR3-2	PUT	Add a mood to a timeline	username, ExperienceID, valence, arousal	void

Table 8.1: Required web services to integrate with external partners

ID	Description
SR3-1	A sensible method for calculating mood average
SR3-2	Support for two-dimensional moods, with valence on x-axis and arousal on y-axis

Table 8.2: Requirements to the server for integration with external partners

ID	Description
CR3-1	A sensible representation of the two-dimensional mood
CR3-2	An option to add mood and visualize the mood in the timeline view
CR3-3	An option to get and visualize the current average mood in the timeline view

Table 8.3: Requirements to the client for integration with external partners

8.4 MoodMap as a Basis for Integration

Earlier in the development process we touched the notion of adding functionality of a mood map in our application. Here the users could post their mood and view a representation of the mood of the participants in the group. This was also discussed during the evaluation of the prototype in iteration 1 (See Section 6.5.6 for an idea of using the mood map in a collaborative timeline), making this an interesting feature for us to integrate in the application.

When we were proposed to work with the MIRROR Project to integrate our prototype with their applications, the use of a mood map was central. The purpose was to show how the MoodMap could be utilized in different applications, working together through integration, giving feedback to participants in group based work sessions. The MIRROR Project would provide some demonstrative applications both for acquiring and visualizing data, but they still needed an infrastructure for integration between the applications. This was proposed to be the cloud infrastructure we created in iteration 2.

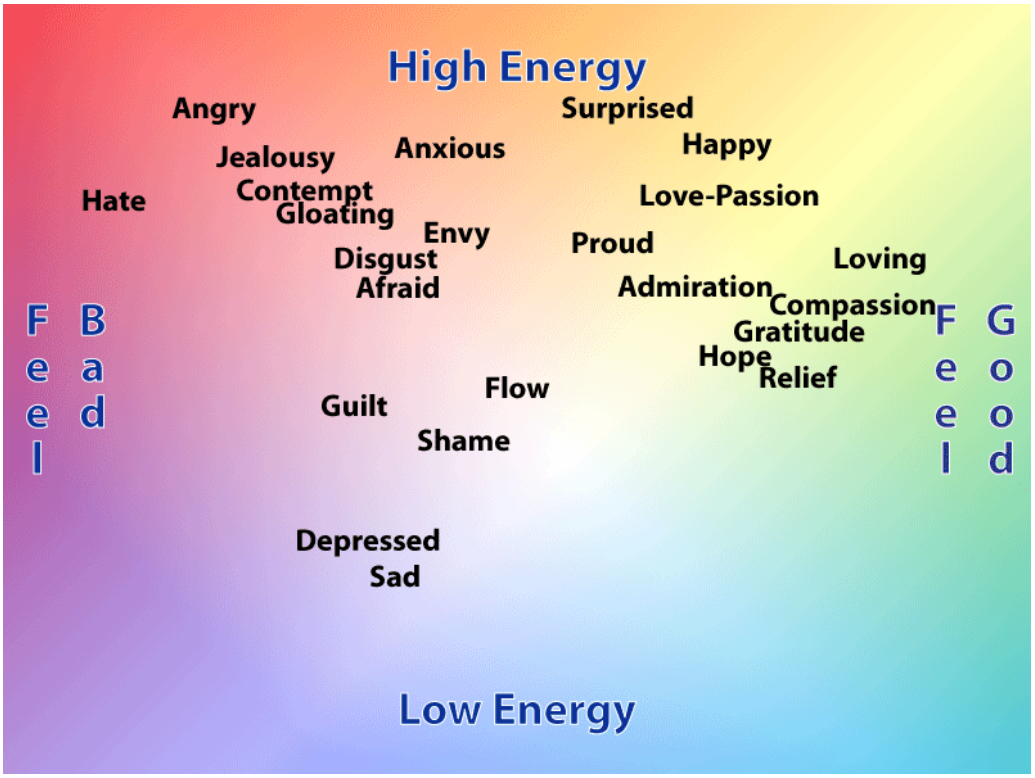


Figure 8.2: The MoodMap used as a basis for integrating the applications

A mood map visualizes emotions in a specter of different feelings. The

MoodMap used for this integration was provided by the MIRROR Project with a few different specifications. The first was that emotions are often evaluated as "feeling good" or "feeling bad", called the valence level of a feeling. The second was that emotions also tend to increase or decrease our arousal, or energy level. In Figure 8.2 a MoodMap illustrating this spectre is illustrated, and it was also this map the MIRROR Project provided us with.

8.4.1 Translating the MoodMap to Data

To transfer the attributes of the MoodMap to be represented as usable data for integration purposes, we mapped the MoodMap into a coordinate system. Then we used these coordinates on the MoodMap to extract data or add input data based on values. Two values were identified:

- The X value representing the valence level - ranging from 0.0 - 1.0
- The Y value representing the arousal level - ranging from 0.0 - 1.0

A specific place on the MoodMap would then be identified as MoodMap(x,y). For instance a X value of 0.7 and a Y value of 0.8 would hit somewhere around "Happy" on the MoodMap illustration, as presented in Figure 8.2.

8.4.2 Representing the MoodMap in the Timeline Application

As requirement CR3-1 stated, a representation of the two-dimensional mood had to be integrated into the Timeline application. On a small mobile screen, a detailed MoodMap as the one in figure 8.2 may not be the best option. The user interface guidelines for developing for the popular iPhone operating system(iOS) states that all targets on the screen that the user can interact with should be fingertip-sized. The recommended size of tappable elements is an area of about 44x44 points [24]. For a user to specify his or her mood on such a map, the most obvious solution would be to let the user click on the place in the MoodMap that is most suiting. On a small mobile screen however, this would be difficult for the user. Not only to hit the exact point that reflects the users mood, but also to understand what the MoodMap tries to show. Therefore, during a workshop with representatives from the MIRROR Project, we decided to create our own representation of the MoodMap in the Timeline application.

X and Y values and representation			
x(0,0-0,5), y(0,5-1.0)	 nervous	x(0,5-1,0), y(0,5-1.0)	 happy
x(0,5-1,0), y(0,0-0,5)	 sad	x(0,5-1,0), y(0,0-0,5)	 calm

Table 8.4: The mapping from a quadrant in the MoodMap to the mood icon and its meaning in the Timeline application.

We were already familiar with using emoticons from iteration 1 when we created the functionality for assessing events. Therefore, we decided to take a similar approach in this matter. We split the MoodMap into four quadrants, representing four different mood statuses and mapped those to four different emoticons that the user could choose from. Table 8.4 shows how this mapping was done.

8.5 External Applications Integrated in the Cloud

A total of six demo applications were integrated with the cloud infrastructure to either acquire or visualize data during the demonstration at the MIRROR GA. The Timeline application was one of these, working both as a provider and visualizer of data. We will in the next section present the applications we worked with and how they used our cloud infrastructure.

8.5.1 MoodMap - Created by FZI, Karlsruhe Germany

A Web based application allowing the user to click in the MoodMap in Figure 8.3. A click will give a value of valence and arousal between 0.0 and 1.0 and push this information to the server to serve as a part of the calculation the average mood value.

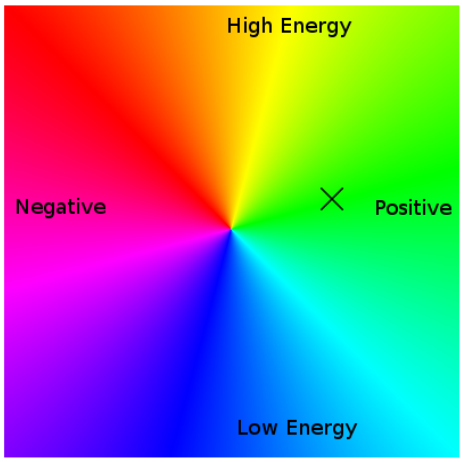


Figure 8.3: The users could click on the MoodMap in their Web browser to express their mood. This information was pushed to the server.

8.5.2 MoodBar - Created by FZI, Karlsruhe, Germany

Another Web based application that visualizes in two separate bars the last posted valence and arousal level compared to the average mood.

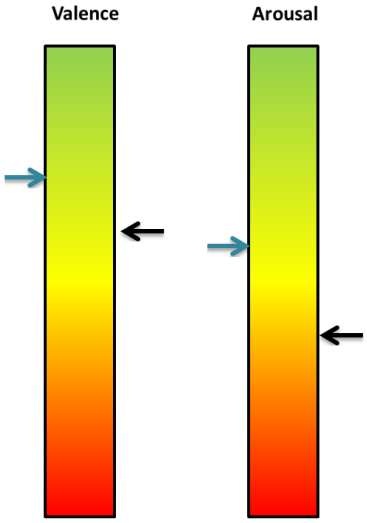


Figure 8.4: The MoodBar Web application created by FZI

8.5.3 Ambient Device Nabaztag - Configured by Simone Mora, NTNU

Nabaztag (Armenian for ”hare”) is a Wi-Fi enabled ambient electronic device shaped as a rabbit. The device can to a certain extent be programmed to visualize different kinds of data. The Nabaztag was configured to represent the average mood of the MIRROR GA by having its ears representing the valence level (ears standing up = positive valence, ears going down = negative valence) and the color on its belly representing the arousal level. The device was connected to the Internet and polled the server for the average mood at a set interval, visualizing it to the crowd in the room.

Figure 8.5: The Nabaztag device

8.5.4 Graphical MoodMap widget - Created by Simone Mora, NTNU

Web-based application visualizing the average mood as a point on the MoodMap. Also showing the arousal and valence level. Configured to poll the server for the average mood values each 15. second.

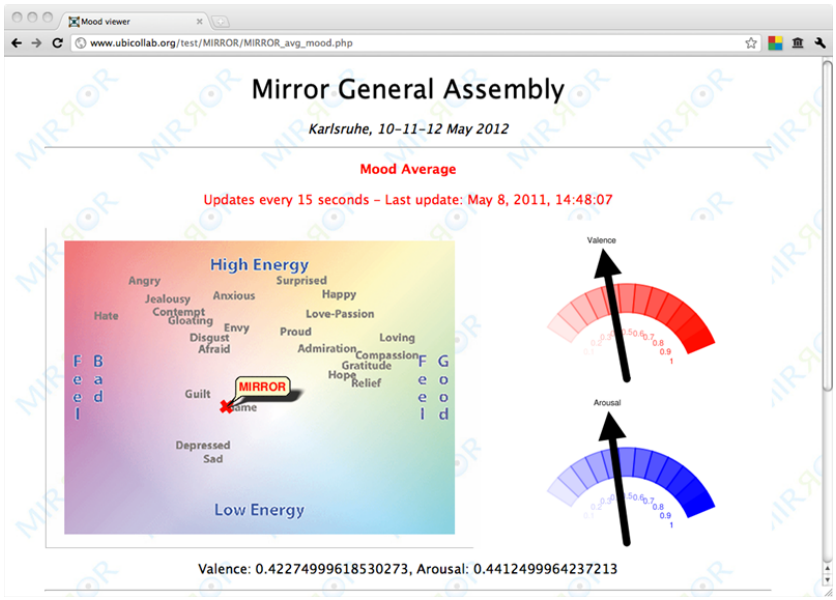


Figure 8.6: The graphical MoodMap widget visualizing the average mood of the MIRROR GA participants

8.5.5 Twitter

A Twitter account was created for the purpose of visualizing the average mood. It automatically tweeted the average mood of the MIRROR GA at an interval of 15 minutes, so followers of the account could get information about the mood.

8.6 Developing the Application for Integration

When we were to further develop the Timeline application to support the Mood functionality, we had to make some extensions to the original architecture. In addition, we had to create a visualization of this in the application as well as new interaction possibilities for the users. This work is described in this section.

8.6.1 Extending to Support MoodMap Functionality

To introduce the new mood functionality and satisfy requirement SR3-2 and SR3-3 (see Table 8.3), only a few additions to the Timeline application architecture had to be made. A new model, MoodEvent, was extended from the BaseEvent and used as the object for saving the information of a mood. Figure 8.7 shows how the new MoodEvent model interacts with the rest of the models. The MoodEvent object has the same attribute as a normal event, but in addition it has attributes concerning mood data. The data from the MoodMap was translated as described in section 8.4.2 and we created a new enumeration, Mood, to hold the data. The enumeration has four different types; Happy, Nervous, Calm and Sad - and the same values and icons as presented in Table 8.4.

8.6.2 Visualizing the MoodEvent

After the implementation of the MoodEvent, the users could add their mood to the timeline through a new quick action menu in the tool bar as shown in Figure 8.8. From this menu, the user could choose between one of the four mood types to represent their actual mood. This will appear as an annotation in the timeline at the time it was posted. The mood will be sent to the server, but will not be visible for the other users of the timeline.

To get the average mood of all the users in a shared timeline, the user can click on the menu button on their Android device and get the option of polling the average mood from the server and showing it in a dialog window on the screen. The visualization is created to correspond with the overall interface design of the application. Figure 8.9 shows how the user can get the average mood of the timeline from the server, as well the dialog window of the corresponding mood. This figure also shows how mood events are scattered throughout a week together with all the normal events.

8.6.3 Privacy of the MoodEvent

The normal events of a timeline are by default set to be private. This implies that the users explicitly needs to share an item for it being public. A shared timeline event will then be available for all the members of the group the timeline is connected to (See Section 7.3.2). For a MoodEvent however, this option is unavailable. By default, a MoodEvent inserted to your timeline

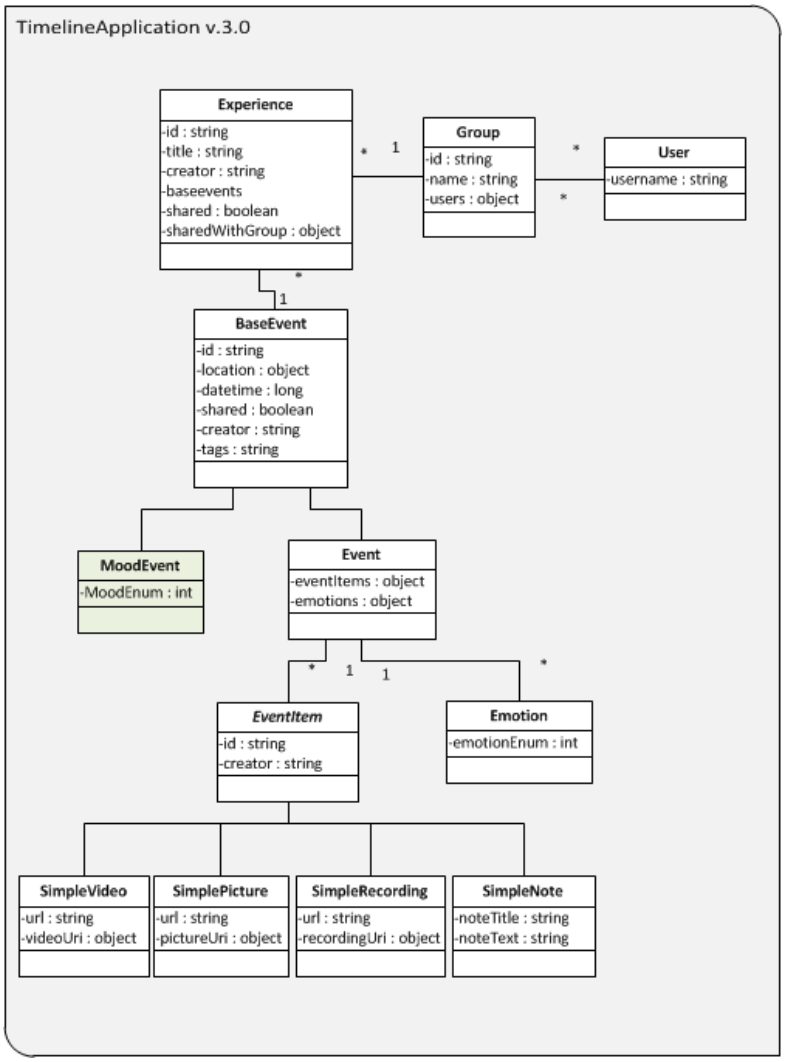


Figure 8.7: This UML diagram shows how the new MoodEvent model was introduced to the already existing Timeline application model structure

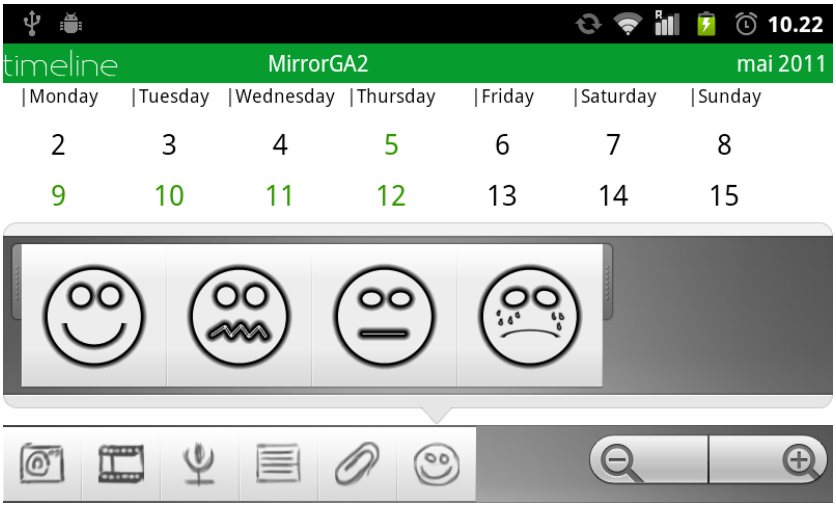


Figure 8.8: The user can choose from 4 different moods when inserting his or hers mood in a shared timeline

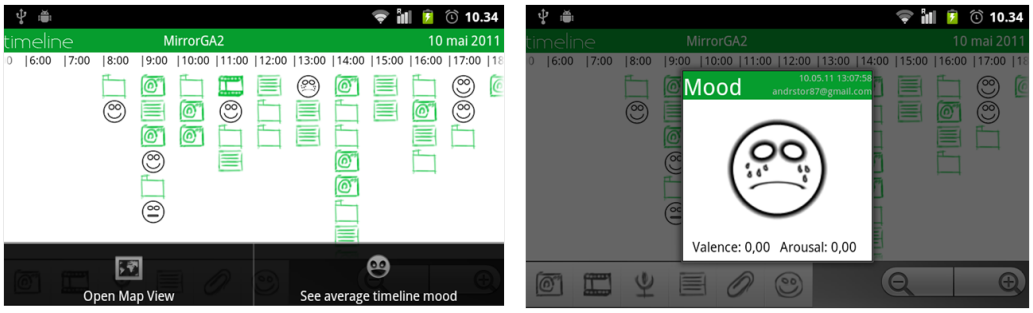


Figure 8.9: The user can get the average mood of a timeline (Left) and click on a mood event in the timeline to see a dialog window with information (Right)

will not be seen by other users, even if the timeline is shared. But still, the data of the MoodEvent posted is sent to the server as a basis for the average mood calculation. The average mood of a timeline is available for all the users, but the mood of a specific user is only visible to the user who posted it.

We chose this approach as it was a requirement that it should be possible at any time to get the current mood of the timeline group (Requirement WSR3-1 and CR3-3). To satisfy this requirement, a mood *has* to be shared at the time it is posted. This contradicts to the approach we use for the normal events, and shows one of the challenges of integrating different systems.

8.7 Extending the Cloud Infrastructure for Integration

Equivalent to the Timeline application we also had to make small adjustments to the cloud infrastructure to support the MoodMap integration.

8.7.1 Model Changes

To extend the cloud infrastructure to integrate the MoodMap of the external applications, the same model changes as the Timeline application had to be implemented. This included adding the two dimensions of the mood, valence and arousal to the MoodEvent.

8.7.2 New Web Services

To allow external applications to insert MoodEvents into the cloud database, we created a set of simplified web services. The only values the external applications were supposed to insert was the valence and arousal, see requirement WSR3-2 (Table 8.1). However, one of *our* requirements to store events in the database is that the ID of the experience to insert the event into, and the creator of the event, has to be known. We therefore ended up with a Web service taking in the experience ID, username, valence and arousal as parameters. The server then set up the rest of the fields required for an Event, such as ID and timestamp. As well as inserting MoodEvents to the database our cloud infrastructure was also supposed to provide services to

obtain the average mood of an experience (Requirement SR3-1). A simple algorithm computing the average of the last ten inserted MoodEvents was implemented. We chose to use the last ten MoodEvents to reflect the current trend. To get the average mood of a timeline, a web service taking the experience ID as parameter was created (Requirement WSR3-1).

Chapter 9

Final Evaluation

Figure 9.1: The evaluation was performed at the FZI Science Centre for Informatics

In this chapter we explain the evaluation we performed at FZI Forschungszentrum Informatik [1] in Karlsruhe for the MIRROR Project general assembly. We will present the result we gathered through informal interviews of users, logging of usage and data gathered from the server used during the evaluation. At last, we will discuss results in regards to our research questions.

[1] `http://www.fzi.de/`

9.1 Evaluation Settings

The evaluation was conducted during a three day long general assembly (GA) for the MIRROR Project. During the meeting the application was made available to all participants and we observed its usage. It was during the first two days we were present and managed to collect most of our feedback. About 40 members took part of the GA and it included long meeting sessions with discussions and presentations, smaller sessions with the group divided in smaller divisions and a guided tour of the science center where the meeting took place. In this section we will in detail describe the planning phase, the actual evaluation setting, how it was performed and how we worked with the feedback gathered.

9.1.1 Preparation

Before the trip, we released the application on the Android Market [2] to make it easy for the participants of the GA to obtain it. To make the participants aware of the application before the actual GA assembly, an e-mail was sent out to all the participants explaining the purpose of our application and evaluation as well as how and where they could acquire the application. The reasoning behind this was to not having to use a lot of time explaining how to use the application during the GA. In this e-mail we also attached a short user guide that could help the users get started using the application.

We also prepared a specific timeline for usage during the GA called "MirrorGA2" that was shared with the "Mirror" group. We invited all participants registered on the server to this group so that the timeline would be available for them. This was the timeline to be used for sharing of data and moods during the GA.

9.1.2 Setting for the Evaluation

The evaluation took place in a conference room where all the participants were seated in a horseshoe pattern (See Figure 9.2) with their faces facing the presentation area at the bottom of the horseshoe. The presentation area included two large presentation canvases (where all the presentation slides were presented), a smaller canvas showing the MoodMap widget (See

[2]The Android Market is Google's equivalent to the Apple app store for distributing and selling applications for Android devices - `https://market.android.com/`

Figure 9.2: The seating arrangement in the room where the evaluation took place

Figure 8.6), and on a table in front of these canvases, the Nabaztag rabbit was placed (See Figure 8.5). Everything was easily visible for all participants sitting in the horseshoe. In Figure 9.3 the presentation area can be seen.

At the beginning of the first day, we held a short presentation for the participants explaining how they could acquire and use the Timeline application to share their experiences and their mood during the GA. We also encouraged the users to play with the application as much as possible. This presentation was a part of a bigger presentation also explaining the concept of using the mood map to measure the average mood of the GA participants. During the first break we handed out a quick start guide to help the users get started with the application. This quick start guide can be found in Appendix C.

All the participants with an Android device could use the application, and we also brought 5 devices (4 HTC Desire HD [3] and 1 Samsung Galaxy Tab [4]) to give out in order to let users without an Android device participate. 25 users in total were invited to the "Mirror" group, meaning that they were the ones with access to the "MirrorGA2" timeline.

The first day lasted from 08:30 in the morning to 19:00 in the evening and included presentations, discussions, separated meetings, lunch and breaks, and a guided tour of the centre. The second day where shorter for us, because we left half way in, around 14:00, but for the rest of the participants also this

[3]HTC Desire HD Android Phone - `http://www.htc.com/europe/product/desirehd/overview.html`

[4]Samsung Galaxy Tab Android Tablet - `http://www.samsung.com/global/microsite/galaxytab/`

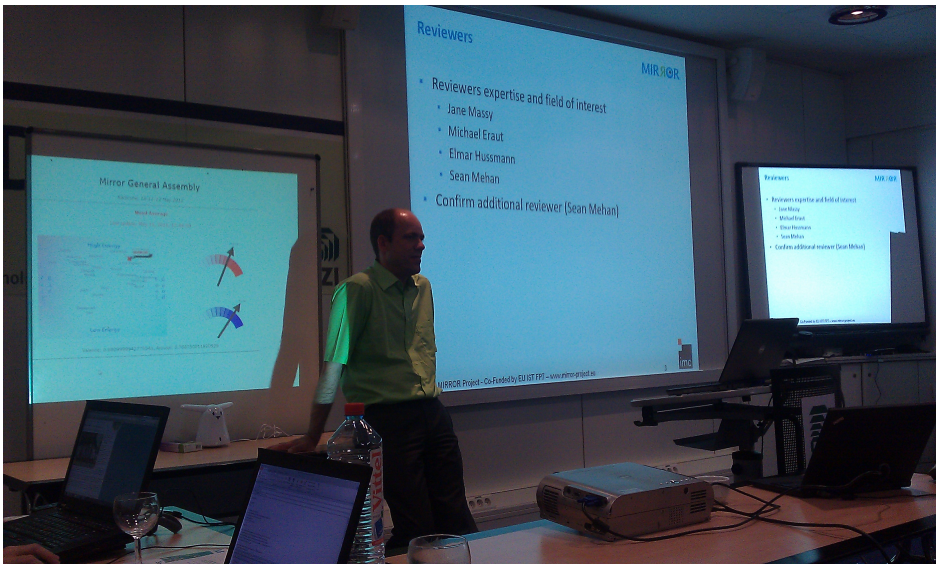


Figure 9.3: The presentation area. The Mood Map widget and the Nabaztag rabbit are visible to the left

day lasted from 08:30 to 19:00. When we went home, most of the technical devices were left behind so that the participants still could share content in the timeline also the last day, as well as having the MoodMap visible during the meeting. This made it possible for us to collect data from the last part of the GA, even if we weren't present.

9.1.3 Collection of Feedback and Data

We collected feedback and data from three different sources during the GA. When we had the opportunity, we did small informal interviews with the users, asking them questions about the application. As we also were a part of the "Mirror" group in our application, we had access to the "MirrorGA2" timeline and all its shared content. From observing when and what users shared in this timeline, and connecting this to the ongoing session at that time in the GA, we got valuable data. At last, we had access to the server where we could look at the logs and data storage to collect data about the actions performed and the activity of usage. This also let us see the activity in the sharing of moods, either posted from our application or one of the other applications available.

9.1.4 Users

As the MIRROR project revolves around reflective learning at work, the members are in one way or another associated with the domain. During the GA we talked with experts both in theoretical and technological areas of this domain. In other words, most of the users participating in our evaluation can be considered experts in the field of reflection in learning and this gave us an extra dimension when it came to feedback based around the concepts of the application. A problem with this is that these users may not represent the intended users of the application and could give a different kind of feedback. But as this is a prototype mainly based around concepts, and not (at least yet) intended for the public, we felt that this was the most valuable feedback we could get.

9.1.5 Analyzing the Data

After the GA, all the data we had collected was a collection of notes, server logs, database usage and the content of the "MirrorGA2" timeline. To get answers from all this data we had to analyze it. We did this by sorting all our notes and relevant log findings into different themes of feedback. Then we analyzed and triangulated this data by looking for similar results from different sources.

9.2 Results

In this section we will present the results we gathered from the feedback collected at the GA. These will be presented in different themes based on the type of feedback. Some may overlap a bit, but we try to keep the results sorted to these themes.

9.2.1 Mobility and Time

Even though the mobility in this setting was not the most important aspect of the mobile device, users still felt it was usable to have the possibility to collect data with the application. Few desktop applications supports pictures, videos and audio recordings as good as a mobile application can, because of the easily accessible hardware components that most new smart phones offers.

This is one of the main strengths of using a mobile device for collecting data, and one of the definitions of mobile learning, stated by O'Malley et.al [39].

A problem that was mentioned with the setting of our evaluation was that because of the short time-span, the attributes of the timeline as a representation may not be utilized when all events are collected in such a short period of time. People are not going back to see older events that they still have fresh in mind.

9.2.2 Usability

Overall, the users seemed satisfied with the usability of the application. For being just a prototype with a very limited focus on usability during the development (and zero usability tests), the amount of feedback we got on problems using the application due to usability issues were limited.

The most important feedback was that the application was hard to use on devices with smaller screens than the HTC Desires' [5] 3.7 inches, which we had been using during the development. Especially a person using a HTC Wildfire [6] with a 3.2 inch screen complained about a hard time opening the different events in a timeline because of the small screen he had.

Other small usability issues we got feedback on was to better visualize events in a timeline based on their attributes, for instance if they have been assessed, gotten additional content or updated. Some users expressed that it should be easier to see which events belongs to different users in a shared timeline. Also, some felt it was hard to locate some of the functionality that the application provided.

9.2.3 Technical

Server Scalability

During the evaluation in Karlsruhe we utilized the web interface Google App Engine offers. We could easily monitor the status of the server, the content in the database, incoming requests and error logs. What we experienced was

[5] HTC Desire Specifications: `http://www.htc.com/www/product/desire/specification.html`

[6] HTC Wildfire specifications: `http://www.htc.com/no/product/wildfire/specification.html`

that the continuous polling from the MoodMap widget every 15. seconds consumed a lot of CPU time. During the first hour of operation most of the free CPU time of 6.50 CPU hours was consumed. At this time we could really try out the scalability advantages of a cloud infrastructure. We simply changed our billing budget at Google App Engine to allow use of CPU time up to $6 per day. Within a couple of minutes this increased the daily limit to 66.50 CPU hours, which proved to be more than enough for the evaluation. Figure 9.4 shows the status of the Google App Engine towards the end of day one. Note the budget of CPU time that was changed during the session, and the CPU seconds graph that shows increasing activity throughout the day.

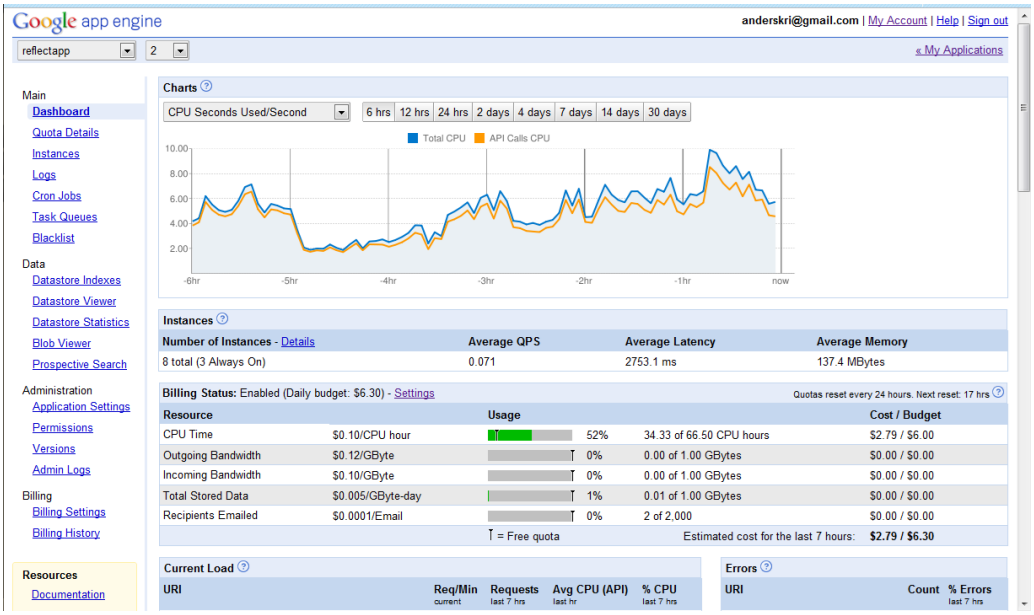


Figure 9.4: The status of Google App Engine during the first day of the evaluation

Synchronization

One of the feedback we got from several of the users was that they thought the synchronization of the timelines was too time consuming. As the number of shared events, especially picture events, increased, so did the time of the synchronization. Using the wi-fi connection offered at the location, a synchronization process took approximately 1 minute. Using a mobile 3G connection the process took considerably more time, and was unbearable for the users, which had to switch to the wi-fi connection.

The feedback from the users showed that most of the users accepted that the initial synchronization took some time, as a number events and media files had to be loaded. However, the subsequent, regularly synchronizations should be quick to perform. Some users also expressed a desire for a real time synchronization where events are automatically pushed into the current timeline. They were used to this behavior from e.g. the e-mail client on their devices.

Integration

An important goal while attending the GA was to have a talk with the developers that had used our cloud API to integrate their MoodMap applications. The developers told us that they had no problems using our web services. They had used both the insertion service and the getter service for the mood average, and said that it was convenient for them that we had created simplified services that only needed a JSON with valence and arousal. This way they did not have to know the model structure specifics of the Timeline application, and as one of the developers told us – *"it just works as we expect!"*.

Devices used During Evaluation

From the developers console in the Android Market we could get access to statistics about the different clients that had installed the application through Android Market. With over 310 different Android devices available in various shapes and sizes [5], it is a challenge to develop applications for Android. Add to the equation that there are several different Android versions to make it even more challenging. It was therefore interesting for us to test the application on as many devices as possible in order to evaluate the quality of the application, and earn valuable experience in developing applications for Android. The statistics from Android Market is shown in Figure 9.5 and 9.6. As we can see of Figure 9.5 the application was installed on 12 different devices, running 4 different versions of Android(see Figure 9.6). Feedback from users told us that the application was running as it should on all devices, but the user experience was different.

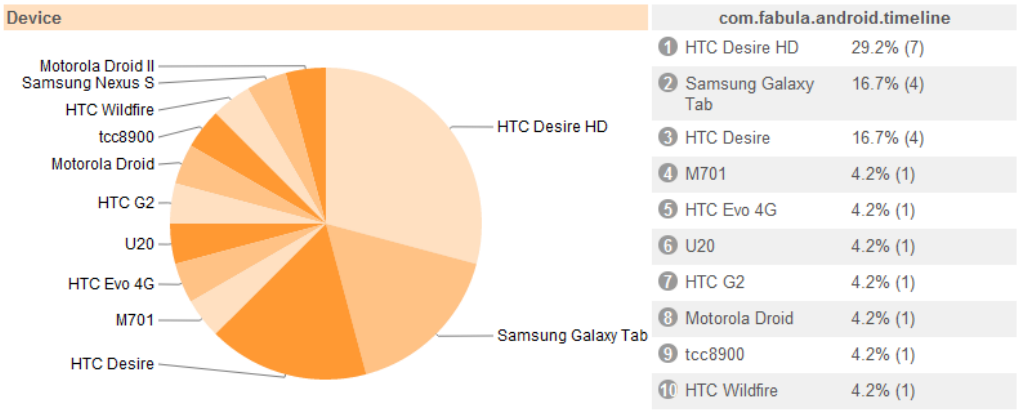


Figure 9.5: Sector diagram showing the distribution of the different Android devices used during the evaluation

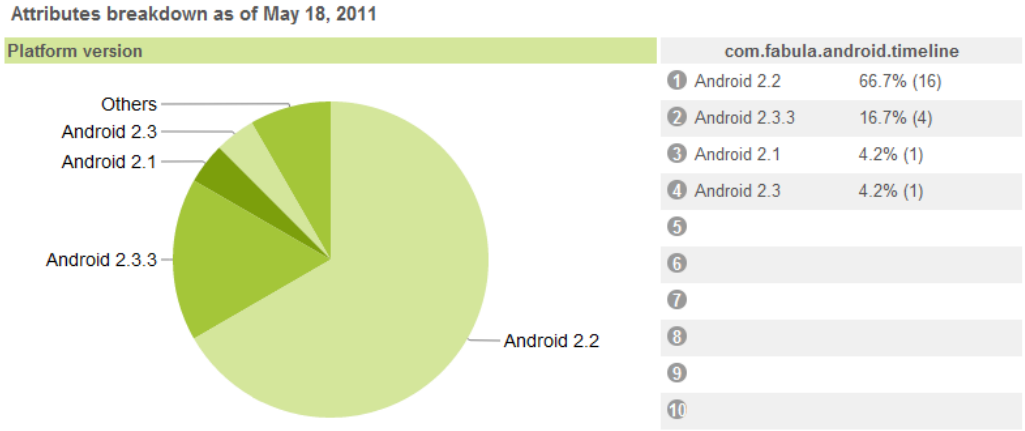


Figure 9.6: Sector diagram showing the distribution of the different Android versions used during the evaluation

9.2.4 Reflection

A lot of the feedback we got from the user in the domain of reflection was connected to the use of reflection triggers to enhance reflection in the application. Many users expressed that they felt involved in the process of building the timeline, and that it was fun to see the events that other users added. We can take from this that users felt it was interesting to see how the timeline evolved, and that they used it to be aware of what other users were doing or thinking about.

We observed that some users used the possibility of adding extra content to an event, and these were often an answer or an addition to the original content. For instance a comment to a picture. Assessment of events with emoticons was also happening, and we observed that these emoticons always were positive, maybe indicating that the users didn't feel comfortable enough with each other to be negative.

Many of the events shared were related to the working environment that the meeting took place in. Notes like "it's so warm in here" or "a long time since the last break now" shows that the users used the application to express their feelings towards things that affected the work being done. This is an interesting observation because such information could be used by the organizer of the meeting to make the work environment more efficient.

We got this feedback from a user that felt that the application was more suitable as a reflection journal:

> *"The visualization reminds me more about a journal than a tool specifically designed for reflection. The data you collect are typically used for reflection purposes and therefore I think this is a good tool/concept for creating the "database" of information that later may be used in reflection. But I am not sure if the application itself at its current state can enhance reflection."*

When asked to elaborate more on this statement, the user emphasized that the data collected by the application was suitable for reflection, but that she had troubles with finding the functionality that could help enhance reflection. When we showed her how to asses and comment an event, she rephrased that she meant that the visualization itself could be too little to enhance reflection, but by giving user the possibility to interact with the data, this in itself could trigger reflection.

The feedback we received on the mood map integration pointed to it being

appealing to many of the users, and it was easy for them to see the value of it. Several users mentioned that it would be interesting to have the average mood data available when working with people and that this could trigger reflection in group work. If we relate this to the reflection levels of Fleck and Fitzpatrick [19], we see that this functionality could take the reflection up to level 3, by exploring relationships between the average mood and events.

Some functionality were also proposed to enhance the support for reflection. Users requested a better visualization of which items in the timeline was their own. They felt it was hard to distinguish between the ones collected by "me" and those collected by others. A suggestion was to have the original content of an event separated from the content added later on, more like a comment on the original data. It was also mentioned that the application was so general that it needed some guidelines, either integrated in the technology or presented before usage. From the application itself it was hard for some users to know what kind of data to collect and share, and it was easy to forget to share items that could be used for reflection. This was also evident from looking at the "MirrorGA2" timeline after the GA. Some of the items shared to the timeline included reflective notes or pictures, but a lot was also more or less random images or notes. It seemed like some users were using the application more as a fun tool to take pictures or write messages with, while others were sharing more deliberate in the form of content that could be used for reflection (see Section 9.2.6 for identified item categories).

An interesting observation we made after we had left Karlsruhe, was that one of the users had added a note reflecting back on the last day: "although the timeline does not reflect this, it has been a day with lots of discussions. now the official part is over." The user used the timeline to look back at the experience and reflected on it, even after the evaluation was over.

9.2.5 Visualization

What we wanted to find out about the visualization was how users felt about the use of a timeline as representation of a learning experience. The feedback we got on this was very one-sided in the direction of *"yes, the timeline works as a representation of a learning experience"*. Users told us that it was easy to understand why different elements were placed as they were in the timeline and that it was nice to have their experiences sorted based on time. Users also commented that the representation of the MoodMap in the application was useful, as a smiley face easily maps to ones mood, and made it quick to add moods to the timeline.

9.2.6 Usage

In this section we will describe how the application was used based on data and our observations.

Getting the Users Started

As the application was published on Android Market, and the participants were noticed about this some days in advance, several people had already installed the application on their device. We observed that about 5 people started to use the application and contributing to the shared timeline by annotating and sharing content, before the actual evaluation had started. These people had already passed the familiarization process and knew and understood the basic concepts of the application when the evaluation started.

The rest of the users were introduced to the application at the beginning of the first day, and we had to perform some help duties throughout the evaluation to get some of these users started. As we were helping a user get connected with the shared group, we observed that someone had started a new group with the same name as the common shared group. There were several users that had been added to this group, and that indicates that some people had misunderstood the group concept.

Types of Items

The sector diagram in Figure 9.7 states clearly that there are two types of items that were mainly used, the note and the picture.

Item Categories

The users shared a lot of different items during the meetings. In addition to separating the items into types, we could from observation and collected data see that there are several categories of sharing. It seems like some users are more aware of how to use the application in a reflective manner - sharing notes with their thoughts about what is happening and questions they have. From this we can see that some users are on their way to what Fleck and Fitzpatrick call the second level of reflection [19]. Other users tended to use the application more like other more known social media applications are

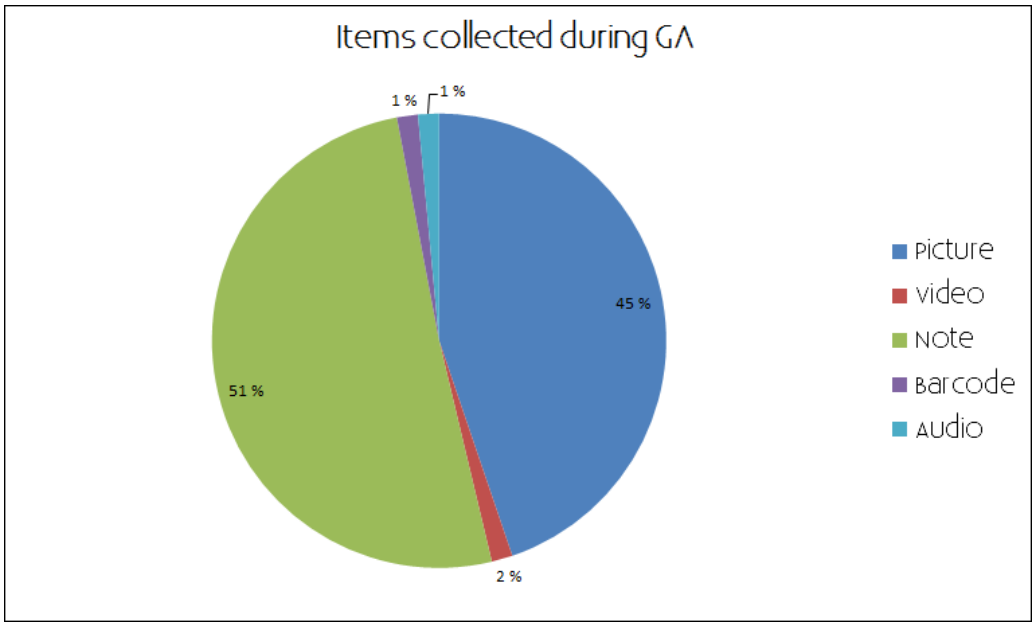


Figure 9.7: Sector diagram showing the distribution of the different item types created during the evaluation

used, for instance sharing status updates and descriptions, sending messages through the timeline and information they collect. Examples:

Bug Discoveries

> *"can't swipe timeline to the right although today is no longer in the future"*

Status updates

> *"We are now splitted into two tracks. I'm attending to the technical meeting."*

Descriptions

> *"These moodmaps are really nice"*

"Death by powerpoint"

"Storyboards, finding the relationship with user studies"

"clustering cards ..with suggested reflection activities"

Messages Between each Other

"Does the discussion of the deliverables affect the mood of the MIRROR group?"
"For sure it did!"

"The room is so warm. This could be a long day"
"Seems all the Laptops heat the room up ;)"
"Woolsweater was also very hot!"

Reflections

"Sounds like it is a lot of criticism to the coordination. Lets see how can reflect on it and make it better"

"I think I used to much time for my slides, however, it was interesting"

"Had to setup wifi since it took hours to sync via 3g.."

"Low arousal in the group now. Do people need lunch? Good valence though.:-)"

"Integration, we still need a lot more integration in this project"
"maybe there should be an automatic break alarm when energy gets below a specific level :)"

"although the timeline does not reflect this, it has been a day with lots of discussions. now the official part is over."

Questions

"The tool collects experiences, but does it lead to reflection?"

Information

"http://aal.fzi.de/" (Collected by scanning barcode)

9.2.7 Mood Sharing

To evaluate the mood sharing and the variation of the average mood, we created a graph of the average mood for each of the days in the GA. What we found interesting was that this graph is also in principle a timeline. However, looking at the graphs alone did not give us any useful information by itself. Yes, we saw that there was variations, and highs and lows, but not any information of *why* this was the case. We therefore overlaid events from our Timeline application on the particular day, at interesting areas of the graph, such as highs and lows, or sudden decrease or increase in mood (see Figure 9.8). This gave us interesting results both in terms of correlations between the mood and events, and of the usage of the application in general. In the following sections these findings will be elaborated. The references to the events referred to in the following sections are to be found in Picture 9.9.

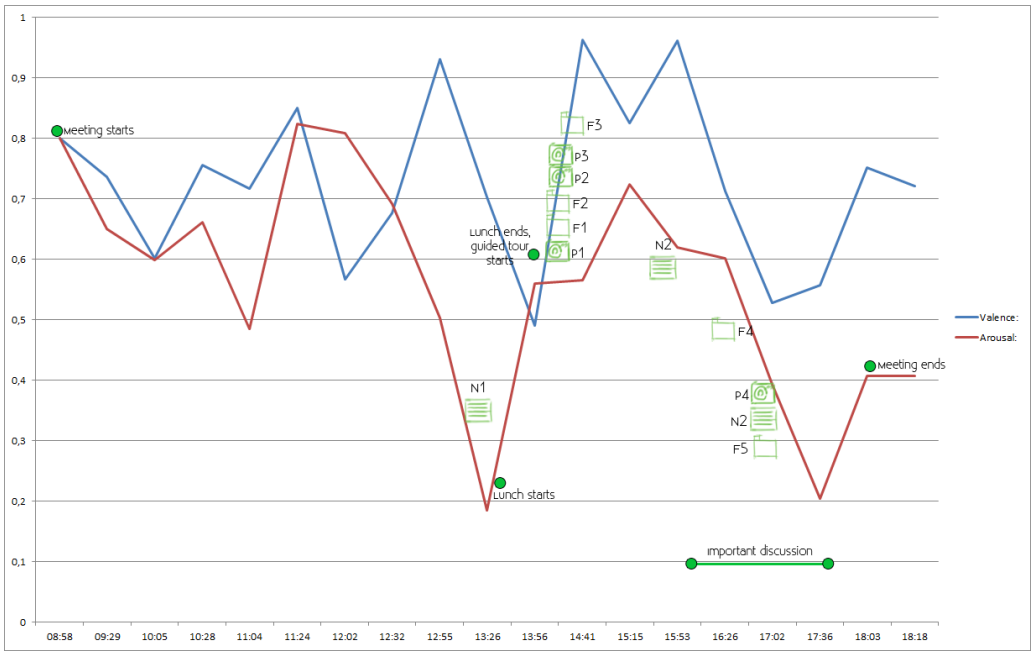


Figure 9.8: A mood graph with events from the Timeline application

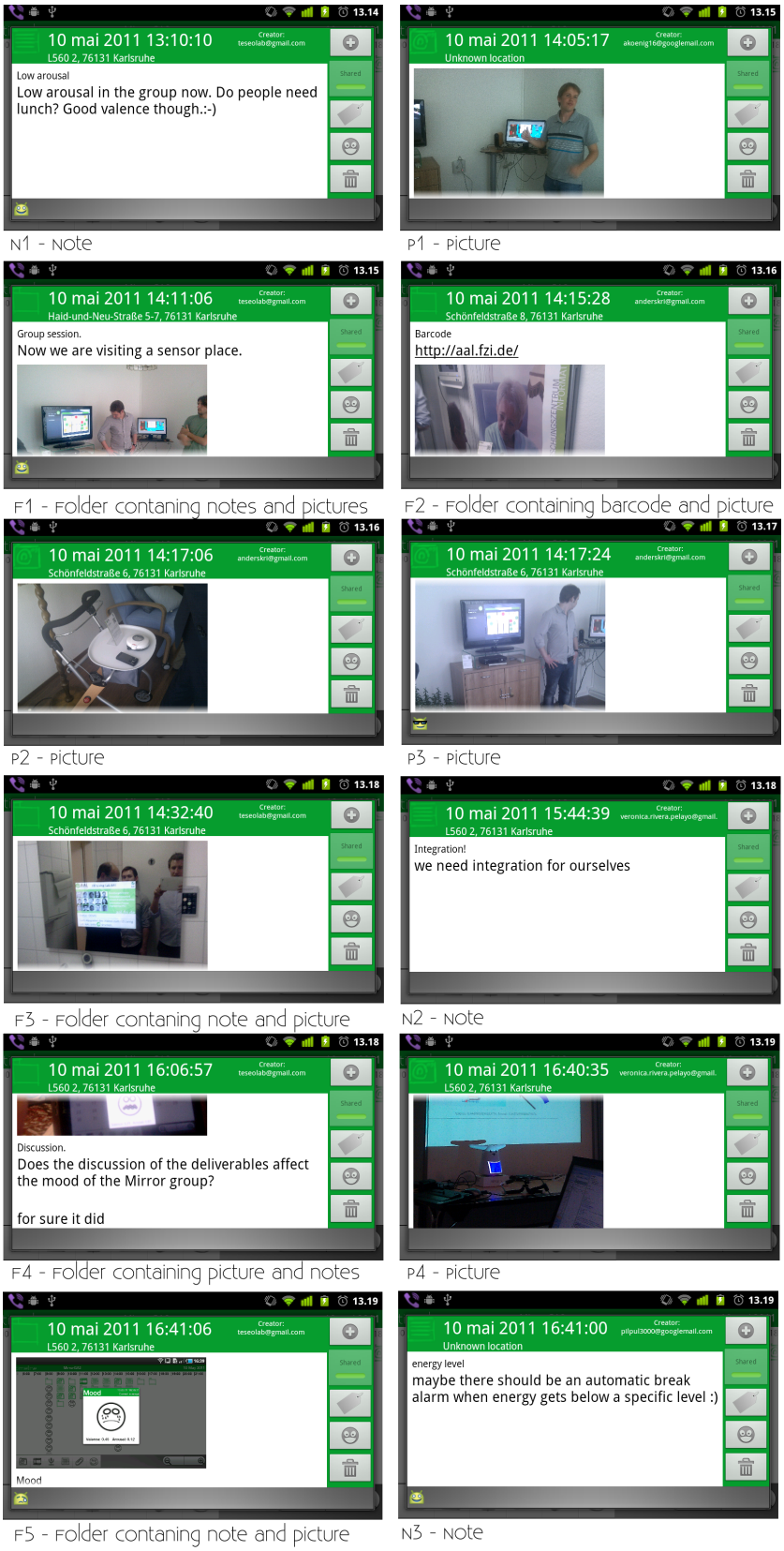


Figure 9.9: Screenshots of the events annotated in the graph

9.2.8 Events Explaining Mood

At approximately 13:30 we can see a low in arousal, while the valence still is high. This means that the group had a good feeling, but was low in energy. A user shared a note, N1, in the timeline some minutes before this arousal bottom states an observation of this drop in arousal, and asks a question if people need lunch. One person has also assessed the note with the emoticon "like". This might be to state an agreement with the creator of the note.

We can also see that moods were posted during the lunch, and by the end of the lunch the valence and arousal had evened out. What this is caused by couldn't the timeline explain, as no events were posted during the lunch.

After the lunch, a rise in the valence and arousal can be observed. In the same period we can also see a peak in number of events during an hour. Between 14:00 and 15:00 8 events, with a total of 15 items were collected. This was highest amount of events and items in one hour during our evaluation. This can be explained by a guided tour to a nearby facility, a living lab[7], with state-of-the-art sensors and technology. What we can observe from the events captured during the guided tour was that there were merely pictures events and folder events containing pictures and notes. One person also used the barcode functionality to add information from a barcode as a description to a picture, see Figure 9.9, event F2.

The next observation can be seen starting at note N2, at approximately 15:45. This note was added during the startup of a new session that eventually ended up in an important discussion of one of the main themes in the MIRROR project. During this discussion we can see that both the valence and arousal values decreased. The timeline has 4 events created during this session. Interestingly all the events are commenting the mood, not the discussion, or any specific events leading to the decreasing mood. The only event linking the mood to the discussion is event F4. But it only states that there is a connection between the mood and the current discussion. What we observed being there during this discussion was that the persons involved in the discussion were too engaged in the discussion, and did not have the time to create events or adding their mood using the tools provided. The persons contributing to the timeline and adding mood were the ones that didn't play an active role during the discussion, and that might felt a little bored as the session lasted over 2 hours without a break, and wanted to point this out by adding their mood and emphasizing the bad mood using the timeline.

[7] AAL FZI Living Lab - http://aal.fzi.de/

9.3 Discussion

In this section we will discuss the results and experiences gathered during the MIRROR GA evaluation, the expert evaluation in iteration 1 and the development process. We will discuss this against the research questions the Timeline application was developed for answering. Our main research question is general in terms of covering the whole aspect of capturing a learning experience collaboratively in a timeline. Our sub-research questions however, are more specific by trying to answer particular parts of the main research question. We will use the results gathered to answer all the sub research questions, before we summarize by answering the main research question.

9.3.1 Sub RQ-1

What functionality should a mobile collaborative tool support to help users share their learning experiences?

Sub research question 1 focused on answering if the functionality we developed for the Timeline application would support the users in sharing their learning experiences. We have identified six different aspects that is important to take under consideration when developing a tool for collaborative sharing. We will now discuss why we believe these are so important when answering RQ-1.

(1) System Captured Attributes.

By utilizing the embedded features in new mobile phones, a lot of information can be gathered. This can normally be attributes like time and location. What we learned from developing this tool is that these attributes can be valuable when collecting data even if they are not intended for usage. When we first started developing the application, time was the only attribute we regarded as important to store, but other attributes, like location, was so straight forward to collect as well. This was also included in our data and storage models. This can be done in the background and does not need to influence the usage of the application.

During the development process of the application, we understood that the requirements of the tool changed when experimenting with different scenarios. Therefore our advice is that when developing a general tool for supporting collaborative collection of data, as much metadata as possible about the data collected should be included. This will ease the development process because the tool will be much more flexible in handling different types of usage, and it will be easier for the developers to extend the application to support it. An example of this was when we, after iteration 1, were introduced to using the location of collected data as an alternative representation to time (See Section 6.5.4). As we already had collected the location attributes in our models, such a representation was easy to develop.

(2) User Collected Data

When we developed the functionality for collecting data our thought process was more or less; "the more the merrier". This is not always the case. After the application evaluation we understood that having support for so many data types creates issues. First of all, we observed that the user mostly used simple notes and pictures when sharing data (See Figure 9.7), even if all the collection functionality were located in the same tool bar. This can be because of the setting where there weren't many happenings that required the use of the other data types. Even when we went on the guided trip, the notes and pictures were still the dominant data types. It can also be because this is the data types regular users are more familiar with sharing in other domains, for instance SMS, MMS or Facebook. Unfortunately, we were not able to test the application in another setting, so we can't answer this with more than these hypothesizes.

Also, even with just notes and pictures - with just 25 people using the application at the same time - this puts a strain on the infrastructure. We observed that users were annoyed when having to wait around a minute for synchronizing their devices with the server. We can imagine, that with more extensive use of larger data types such as video and audio, this strain will just rise and will require, in addition to a optimized synchronizing algorithm, a server infrastructure that can be both expensive and complex. This can be a problem for light weight tools such as mobile applications when it comes to usability and user satisfaction, and may scare the users from actually using the tool.

(3) Synchronization

Synchronization of data between a server and devices can be done in two different ways, automatically or manually. Automatically meaning that everything should be synchronized in real time or at a set time interval, while manually means that the user should decide when to do the synchronization. We took the manual approach. Based on the feedback we got from the final evaluation, in a setting where the users do the collection of experiences in a short period of time, in close proximity to the rest of the users they are working with, an automatic approach is better. This is because in such a setting, you *know* that other people are contributing at the same time as you, and this triggers curiosity. With automatic synchronization this would not be an issue because all new updates in the timeline would be visible almost instantaneously. However, an issue with automatic synchronization is that the device needs an Internet connection for this functionality to work. In a setting where people work more distributed, over a longer period of time and without a stable Internet connection, the manual approach that we took might be more suiting.

Another thing to note, the MIRROR Project, whose goal is to develop applications in the domain of reflective learning listed "Offline mode" as one of the most important requirements of mobile collecting tools. The reasoning behind this is that reflection should be able to happen anywhere at anytime.

(4) Awareness

To foster collaborative sharing, it is important that the users are aware of environment that they are a part of [13]. Feedback on this issue pointed to a lack of support for this in our application. Users emphasized the importance of being notified of changes in the timeline, and felt it was difficult to keep awareness of the other users activity without this notification. It should be a prioritized requirement to foster awareness in a collaborative sharing tool. For instance, based on the feedback for our application; every time updates happens in a collaborative environment, participating users should be notified.

(5) Guidelines

We understood early during the evaluation that the usage of the application differed from user to user. As we were made of aware of during the first

iteration evaluation: "Users often tend to use functionality different than intended by the developers"(Section 6.5.3). This became apparent for us as well. Examples from the different types of usage is presented in Section 9.2.6.

The positive about a general tool for sharing learning experience is that it doesn't set any restrictions on what kind of content to be shared. On the negative side, it is very difficult to set a standard for how it should be used. For instance, our intention was for the user to share reflective thoughts and data. Even if we told them that this was the intention, much of the shared content did not correspond to this. Our results tells us that guidelines or training is needed, verbal or technical, if the tool is to be used in a restrictive manner. Either, the users have to agree upon usage of the tool, or the tool has to "force" the users into this type of usage.

(6) Collaborative Environment

A collaborative environment such as the shared timeline can create an environment where the users feel encouraged to contribute. This can have both a negative and positive effect on reflection. The negative part can be that users posts events without a reflective purpose, as stated in the previous section. This might be something that happened during our final evaluation where people posted status updates and general descriptions. On the contrary, a positive effect of feeling encouraged to contribute will be when the user posts meaningful and potentially reflective content. The latter might be subject to the need of guidelines to understand what content to contribute, as well as some experience using the tool.

By creating and sharing experiences in a collaborative tool, the group obtains awareness of the other users content. We have through the evaluation showed that this leads to feedback on ones contributions from other users. The feedback came either as assessments through emoticons or through adding additional items to the event. Based on this we believe that by letting the users interact on their shared learning experiences in a collaborative environment, the requirements of the reflective process presented by Boud et.al [2] will be met. The users are given a reason to *return to the experience* by having to look at the feedback received. This feedback gives new feelings towards the experience, thus covering *attending to feelings*. At last, the user will *re-evaluate the experience* based on the feedback received.

To summarize, we argue that a requirement of a collaborative tool for sharing learning experiences should be a collaborative environment that encourages

the users to contribute by letting the users interact with each other and for instance give comments or assessment to each others work.

9.3.2 Sub RQ-2

How does a timeline work as a representation of a learning experience on a mobile device?

Our feedback showed that the timeline was perceived as a good representation of a learning experience by the users as it was a familiar concept of representing events. Its strength is that it can show the evolution of a learning experience over time, and thus showing the processes from which the users or a group of users can obtain a learning outcome.

9.3.3 Sub RQ-3

How should data be presented to enhance learning by reflection?

The Timeline application presented the events as annotations differencing the events by the types of items(using icons) and if they were shared or not(using colors). Evaluations showed that this was good, but alone was not enough to enhance reflection. The concepts of events and items proved to be perceived as natural and there was no feedback indicating that this way of presenting happenings was illogical. Feedback received emphasized that the application lacked the ability to differentiate the events further, based on other criteria, such as assessment, tagging and relations.

The users also requested that they wanted to be able to control the presentation of the timeline to a greater extent. They wanted to filter out what they found less interesting for their learning experience, and sort what they found interesting in a way they found useful, for example emphasizing what is my contribution to the timeline, and what are the others'.

We tried to achieve the functionality described in the previous paragraph, using the tagging system. This was a feature we have no observations of peo-

ple using, and thus have no basis of evaluating. The reasons for it not being used might be that it was not explicit and intuitive enough, and that people should be able to do this directly on the data, by filtering and connecting events in the timeline.

9.3.4 Sub RQ-4

What kind of user interactions should the tool support to trigger reflective processes?

The final evaluation gave us a lot of interesting feedback on reflection triggers. An expert in the field of reflection stated that this is one of the most important aspect of an application such as the one we developed. Even though we had tried to integrate triggers in the sense of assessment and selective sharing, this was not explicit enough to trigger reflective processes.

Based on the feedback from the evaluation we have elaborated what kinds of other triggers that could have been integrated in the application.

System Triggers

A possible solution to integrate triggers into the application is to have some system triggers. By system triggers we mean that the system, based on certain events or at certain times, prompt the user to do an action on the timeline. Such an action could be:

- Ask for an assessment when another user shares an event in a shared timeline.
- Ask the user to share their mood at specific time intervals.
- When capturing a picture the system could ask if the picture describes the event well, or if an additional note is needed.

User Triggers

Triggers similar to the ones described in system triggers, except that these are triggers that the user can initiate. A kind of user trigger can for ex-

ample be the facilitated learning discussed in the evaluation of iteration 1, Section 6.5.5, idea number 6. The evaluator stated that this seemed a bit out of our scope. Retrospectively we see that this might have been a good trigger for reflection. Imagine a scenario where for example a facilitator can send out prompts for the following actions:

- Register your current mood.
- Assess at least 3 events.
- Asking the user to answer a question.

Integrate the Tool With Other Data Sources

During the work of analyzing the final evaluation (that actually is a reflective process on a learning experience itself) we saw that combining the timeline with the mood data collected with the MoodMap gave interesting results (See Section 9.2.8). By combining these to data sources we got more out of the timeline, and more out of the mood graph. This experience shows that integrating the tool with other systems might be a good trigger for reflection. An interesting feature to exemplify this would have been to be able to overlay the mood graph onto the timeline, and thus trigger reflection. This way the user will be able to see different data types in relation to each other, and data that make little sense by itself might give a new meaning. A specific point during the evaluation that could have utilized such a feature was when some participants observed a decreasing mood, and felt the need to express this on the timeline. With the graph integrated in the application the meeting coordinator could have reflected upon how the meeting was organized.

9.3.5 Main RQ

How to enhance reflection by capturing collaborative learning experiences in a timeline?

As previously stated, our main research question summarizes the sub-research questions answered in the previous sections. Based on these answers, as well as reviewing related literature and work, we argue that we have identified some characteristics that answers our main research question:

(1) A timeline is a good representation of a learning experience and functions as an overview of events that together leads to a learning outcome. Users finds it easy to map the representation of events collected in a timeline to the concept of a learning experience.

(2) Separation of data in the visualization is important. The timeline visualization itself is not a sufficient trigger for reflection. It is essential that data with different attributes are visualized in a separated manner so that users are able to draw connections or identify patterns that can lead to reflection. This is important in collaborative environments as individual users will be unable to comprehend the experiences shared by others without support.

(3) Technology itself can act as a scaffolding for reflection. However users has to agree on guidelines for usage that steers towards reflecting on their actions and interactions gathered with the technology.

(4) Consider the type of intended scenarios for the application when deciding the type of synchronization needed. The use of the application in a short period of time increases the demand of frequent synchronization. In such a scenario, a real time collaboration tool might cover the users demand better.

(5) The type of data supported for capturing a learning experience should correspond with the goals of the scenario. Small snapshots of an experience can be enough to enhance reflection. Data types that demands a lot from the infrastructure of a tool can also be a strain (e.g. videos). From our experience and results, it would be vise to limit the amount of data types supported to the actual needs in the scenario the application is developed for. Both in terms of usability and technical issues.

This thesis don't claim that this set of characteristics are complete guidelines for achieving reflection in a collaborative tool, but based on our results we argue that the implications of them are important. To identify additional characteristics or strengthen the viability of those identified, further evaluation has to be conducted. If the Timeline application should be used for such evaluations, other scenarios should be considered.

Chapter 10

Conclusion

10.1 Summary

In this thesis we have investigated how collaborative learning experiences collected with mobile devices and represented in a timeline can help reflection. We have developed a prototype application on the Android platform in order to look deeper into what kind of data that should be collected and how the data should be visualized and interacted with.

Today's mobile devices are powerful devices capable of capturing a variety of information. Our investigation of related work and existing applications showed that there are many applications developed for information collection. To differentiate from these applications, and support a central aspect of our research question, *reflection*, we wanted the users to collaborate and perform actions on the collected data in order to promote reflection and enhancing learning experiences. We identified 5 high level requirements that we wanted the application to support; collection of experiences, being in the environment, representation in a timeline, collaborating on experiences and re-visiting experiences. All these requirements were achieved in the Timeline application.

The functionality implemented was a result of taking advantage of the theoretical background, related work, evaluation and discussions with an expert in the area of reflection as well as co-operation with a EU project(MIRROR) that are conducting research in the area of reflection. The resulting prototype was evaluated over 3 days in a real user setting during a general assembly in MIRROR.

From this evaluation, together with the work done in background theory and related work, we have analyzed our results and identified some characteristics that should be considered when developing a tool for enhancing reflection on learning experiences in a collaborative environment.

10.2 Contributions

The work done in this thesis contributes to the areas stated in Section 1.1.1; collaboration technology, mobile technology and technology enhanced learning. Contributions are done in terms of the development of the Timeline application for Android. The experiences gathered both during the development as well as the evaluation of the application gave us a foundation to answer our research question. From these answers we identified characteristics that can go into future research and development in the aforementioned domains.

On the technical level we have created an application including the Android application and server implementation. We showcased how features of the Android system could be utilized in both the aspects of mobile learning referred to in Section 1.1.1.

The work on the cloud infrastructure lead to a nice acquaintance with Google App Engine, and our experiences with the integration with this kind of server infrastructure can be of useful for both possible further development of the Timeline application and for development of other collaborative applications.

Both the application and server source code have been released under the GNU Public License v.3.0 [1], and the repositories are located at Github:

- Timeline application: `https://github.com/andekr/Timeline-App`
- Timeline server: `https://github.com/andrstor/Timeline-Server`

In Section 10.4, Future Work, we have identified interesting features and improvements of how the work done in this thesis can be taken further.

[1] `http://www.gnu.org/licenses/gpl.html`

10.3 Reflection on our Work

The main work done in this thesis was the development of the Timeline application. This task challenged us to learn more about mobile application development on the Android platform. Also, we got to dive into the popular domain of cloud computing, more specifically the Google App Engine, when developing the server infrastructure for the collaborative aspect of the application. These two experiences has given us knowledge that we for certain will benefit from in our future work life. And not to forget, the development of this tool has been a fun and exciting experience for two software development enthusiasts.

Another aspect when working with this type of technology is the extreme speed of which the technology advances. During the course of this thesis several technological updates on the Android and Google App Engine, as well as the development tools has been developed, and we feel its a pity that the Timeline application couldn't benefit from those because of time limitations.

We first became aware of the MIRROR project during iteration 2, and had already developed the application for other scenarios then what MIRROR proposed. If we earlier in the thesis could have started communication with the MIRROR project, we would have been able to dedicate more time for specifically developing the tool for their proposed scenario. Instead, we were lucky enough to experience that our tool and infrastructure already were general enough to support the scenario, but with a little more specialization we may would have been able to get even better results.

Usability was never an prioritized part of the prototype we developed, but as the tool was tested with real life users, it became apparent that it has a large impact. When we look back and reflect, we think that earlier user involvement in the development would have given us insights on how to make the application more attractive and usable for users.

Over all, we can look back at the work done in this thesis and be happy with what we have achieved. We got to work with exiting technologies, an interesting assignment and was invited, based on interests in our thesis work, to the general assembly of a large EU-project to evaluate our tool. This feels like an achievement we can be proud of.

10.4 Future Work

The aim of this thesis has been to identify how we can enhance reflection in a collaborative tool by collecting learning experiences to timelines. As stated in previous sections, there have been some limitations in the evaluation of the tool we developed to answer our research questions.

Therefore, to further improve our answers, more evaluation would have to be conducted. Further work may involve testing the application over a longer period of time where a set of users utilizes the tool to share their learning experiences about the work they perform. Even if we feel that the evaluation we performed was enough to give us the answers we have provided - further evaluation could enhance the strength of these answers and maybe give new ones.

The MIRROR project is planning to extend the application and further adapt it to be used in the MIRROR testbeds.

10.4.1 Technical Improvements

Even though we were pleased of the state of the developed application there is always room for improvements. During the evaluation several technical improvements were proposed. In this section we will summarize the ones we feel would be natural to include in possible new versions of the Timeline application.

Synchronization and Security

Towards the end of our thesis an interesting speech was held during the annual Google IO [2]. The speech name "Android + App Engine: A Developer's Dream Combination" [3] caught our attention and the session introduced new improved development tools to integrate Android with Google App Engine using push technology, that would address one of the problems with the synchronization solution implemented in this thesis as well as adding Google Authentication to improve security.

[2] `http://www.google.com/events/io/2011/about.html`

Improved Visualization and Interaction in the Timeline

The feedback from the final evaluation pointed out that the separation and interaction on the events in the timeline view could be improved to even further promote reflection.

Integrate with more data Sources

By combining the application with sources for automatic gathering of data, such as sensors, additional information can have been gathered without interaction of the user, and give information that is not otherwise perceptible by humans [19]. This can be useful in settings where the user don't have time to gather information. We saw such a scenario during our evaluation, where the users involved in discussions seemed to forget the assessment of their moods. A sensor for e.g. heart rate would have been interesting to see integrated into the timeline.

Bibliography

[1] Smart history. http://smarthistory.org/timeline.html.

[2] *Understanding Adult Education and Training*, chapter Experience-based learning. Allen and Unwin, 1999.

[3] Brad Abrams. Google io session overview: Android + app engine: A developer's dream combination. http://bradabrams.com/2011/05/google-io-session-overview-android-app-engine-a-developers-dream-combination/.

[4] Roy Ballantyne and Jan Packer. Introducing a fifth pedagogy: experience-based strategies for facilitating learning in natural environments. *Environmental Educational Research*, 2009.

[5] Hugo Barra. The official google code blog - android: momentum, mobile and more at google i/o. http://googlecode.blogspot.com/2011/05/android-momentum-mobile-and-more-at.html.

[6] Morten Bohøj, Nikolaj Gandrup Borchorst, Niels Olof Bouvin, Susanne Bødker, and Pär-Ola Zander. Timeline collaboration. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 523–532, New York, NY, USA, 2010. ACM.

[7] D. Boud, R. Keogh, and D. Walker. *Reflection: turning experience into learning*. Kogan Page, 1985.

[8] A Brockbank and I McGill. *Facilitating reflective learning in higher education*. Buckingham: Society for Research in Higher Education and Open University Press, 1998.

[9] Oracle Corporation. Restful web services developer's guide - chapter 1 introduction to restful web services and jersey. http://download.oracle.com/docs/cd/E19776-01/820-4867/ggnyk/index.html.

[10] Oracle Corporation. Restful web services developer's guide - chapter 3 creating a restful resource class. http://download.oracle.com/docs/cd/E19776-01/820-4867/ggnxo/index.html.

[11] Oracle Corporation. Types of web services - the java ee 6 tutorial. http://download.oracle.com/javaee/6/tutorial/doc/giqsx.html.

[12] Oracle Corporation. Using jax-rs with jaxb. http://download.oracle.com/javaee/6/tutorial/doc/gkknj.html.

[13] Paul Dourish and Sara Bly. Portholes: supporting awareness in a distributed work group. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '92, pages 541–547, New York, NY, USA, 1992. ACM.

[14] Evernote. Evernote web page. http://www.evernote.com/.

[15] FABULA. About us - fabula - seamless networks for transforming the city into an arena for learning. http://www.fabula.idi.ntnu.no/?q=about.

[16] FABULA. Fabula project description. http://www.fabula.idi.ntnu.no/sites/default/files/FABULA_short.pdf.

[17] Facebook. Facebook web page. http://www.facebook.com/.

[18] Jerry Alan Fails, Allison Druin, and Mona Leigh Guha. Mobile collaboration: collaboratively reading and creating children's stories on mobile devices. In *IDC '10: Proceedings of the 9th International Conference on Interaction Design and Children*, pages 20–29, New York, NY, USA, 2010. ACM.

[19] Rowanne Fleck and Geraldine Fitzpatrick. Reflecting on reflection: framing a design landscape. In *Proceedings of the 22nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction*, OZCHI '10, pages 216–223, New York, NY, USA, 2010. ACM.

[20] Kavita A. Ganesan, Neelakantan Sundaresan, and Harshal Deo. Mining tag clouds and emoticons behind community feedback. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 1181–1182, New York, NY, USA, 2008. ACM.

[21] J. Grudin. Computer-supported cooperative work: history and focus. *Computer*, 27(5):19 –26, may. 1994.

[22] Michael J. Hannafin and Susan M. Land. The foundations and assumptions of technology-enhanced student-centered learning environments. *Instructional Science*, 1997.

[23] Jarle Moss Hildrum. Sharing tacit knowledge online: A case study of e-learning in cisco's network of system integrator partner firms. *Research Paper. Industry and Innovation. 16(2)*, pages 197–218, 2009.

[24] Apple Inc. Ios user interface guidelines, March 2011.

[25] Google Inc. Android menu design guidelines. `http://developer.android.com/guide/practices/ui_guidelines/menu_design.html`.

[26] Google Inc. Google app engine. `http://code.google.com/appengine/`.

[27] Google Inc. Google news timeline. `http://newstimeline.googlelabs.com/`.

[28] Google Inc. Google time map. `http://code.google.com/p/timemap/`.

[29] D.A. Kolb. *Experiential learning: experience as the source of learning and development*. Prentice Hall, Englewood Cliffs, NJ, 1984.

[30] Birgit Rognebakke Krogstie. *The work-reflection-learning cycle in software engineering student projects: Use of collaboration tools*. PhD thesis, Norwegian University of Science and Technology, NTNU, 2010.

[31] B.V Kumar, Prakash Narayan, and Tony Ng. *Implementing SOA using Java EE*. Prentice Hall, 2010.

[32] Wayne Liew. 10+1 mobile application development trends for 2010. About the trends for development of mobile technologies and applications.

[33] Xiaodong Lin, Cindy Hmelo, Charles Kinzer, and Teresa Secules. Designing technology to support reflection. *Educational Technology Research and Development*, 47:43–62, 1999. 10.1007/BF02299633.

[34] Mnemograph LLC. Timeglider: Web-based timeline software. `http://timeglider.com/`.

[35] Catherine McLoughlin and Ron Oliver. Maximising the language and learning link in computer learning environments. *British Journal of Educational Technology*, 29(2):125–136, 1998.

[36] Laura Naismith, Peter Lonsdale, Giasemi Vavoula, and Mike Sharples. Literature review in mobile technologies and learning. *Nesta Fu-*

turelab Series, 2009. http://www2.futurelab.org.uk/resources/documents/lit_reviews/Mobile_Review.pdf.

[37] Chris Nesladek, Richard Fulcher, and Virgil Dobjanschi. Twitter for android: A closer look at android's evolving ui patterns. http://android-developers.blogspot.com/2010/05/twitter-for-android-closer-look-at.html.

[38] Massachusetts Institute of Technology. Simile widgets. http://www.simile-widgets.org/timeline/.

[39] C. O'Malley, G. Vavoula, J.P. Glew, M. Sharples, and P. Lefrere. Guidelines for learning/teaching/tutoring in a mobile environment. Technical report, MOBIlearn/UoN, 2003.

[40] Jennifer Preece, Yvonne Rogers, and Helen Sharp. *Interaction Design: beyond human-computer interaction*. John Wiley and Sons, 2002.

[41] Mattias Rost and Lars Erik Holmquist. Tools for students doing mobile fieldwork. In *Proceedings of the Fifth IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education*, pages 74–81. IEEE Computer Society, 2008.

[42] Marlene Scardamalia and Carl Bereiter. Higher levels of agency for children in knowledge building: A challenge for the design of new knowledge media. *Journal of the Learning Sciences*, 1991.

[43] Kjeld Schmidt and Liam Bannon. Taking cscw seriously: Supporting articulation work. *Computer Supported Cooperative Work*, 1:7–40, 1992.

[44] Jane K. Seale and Alan J. Cann. Reflection on-line or off-line: the role of learning technologies in encouraging students to reflect. *Computers and Education*, 34(3-4):309 – 320, 2000.

[45] Mike Sharples, Dan Corlett, and Oliver Westmancott. The design and implementation of a mobile learning resource. *Personal Ubiquitous Comput.*, 6:220–234, January 2002.

[46] G. Stahl, T. Koschmann, and D Suthers. Computer-supported collaborative learning: An historical perspective. *In R. K. Sawyer (Ed.), Cambridge handbook of the learning sciences*, 2006.

[47] K. Strampel and R. Oliver. Using technology to foster reflection in higher education. *In ICT: Providing choices for learners and learning. Proceedings Ascilite Singapore 2007*, 2007.

[48] A.L. Strauss. *Continual permutations of action.* Communication and social order. Aldine de Gruyter, 1993.

[49] Twitter. Twitter web page. `http://www.twitter.com/`.

[50] Mark J. Weal, Danius T. Michaelides, Mark K. Thompson, and David C. DeRoure. The ambient wood journals: replaying the experience. In *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, HYPERTEXT '03, pages 20–27, New York, NY, USA, 2003. ACM.

[51] Peta Wyeth and Ian MacColl. Noising around: investigations in mobile learning. In *IDC '10: Proceedings of the 9th International Conference on Interaction Design and Children*, pages 147–155, New York, NY, USA, 2010. ACM.

[52] Ron Yeh, Chunyuan Liao, Scott Klemmer, François Guimbretière, Brian Lee, Boyko Kakaradov, Jeannie Stamberger, and Andreas Paepcke. Butterflynet: a mobile capture and access system for field biology research. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 571–580, New York, NY, USA, 2006. ACM.

Appendices

Appendix A

Expert Evaluation Walkthrough

This is the presentation used during the expert evaluation of iteration 1. The presentation includes an overview of the state of the application, as well as slides to promote discussions(in Norwegian).

Masteroppgaven

"From learning experiences to shared learning trajectories"

- How can mobile experiences in learning be supported by tools for enhancing reflection?
  - How can the reflection process create new or enhance existing learning trajectories from different learning experiences?
  - How can collaborative tools help learners share their learning trajectories?
  - Will the timeline on the mobile device work as a representation of a learning trajectory when emphasizing on reflection?

timeline

Ekspertevaluering

timeline

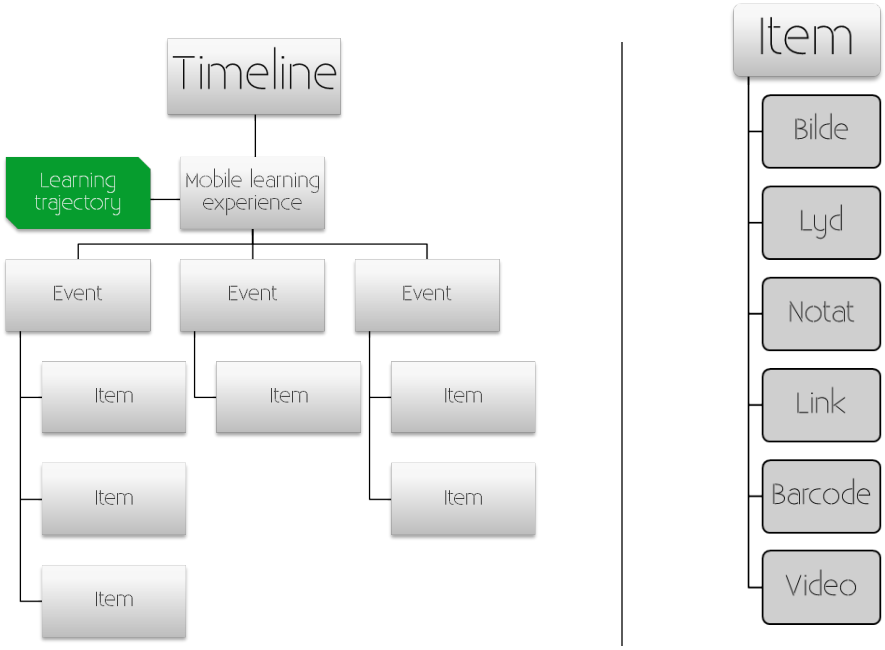
Konsepter

Tidslinje - representerer en "experience"
Experience: En experience representerer et tidsrom med hendelser som potensielt kan gi et læringsutbytte. Tidslinjen er en visuell representasjon av en experience.

Event - representerer en hendelse i "experiencen"
Events er elementer som experiencen består av. En event kan inneholde en eller flere items. Hvor mange items som kobles på en event avhenger av hva som passer situasjonen best. Et bilde kan si mye, men et kort notat som beskriver situasjonen, følelser, ideer og tanker kan øke læringsutbyttet ved å støtte refleksjon bedre.

timeline

Konsepter



timeline

Designvalg

Tidslinje
- Horisontal, kalenderlook - "Usynlig tidslinje"
- Ulike tidsenheter (zoom)

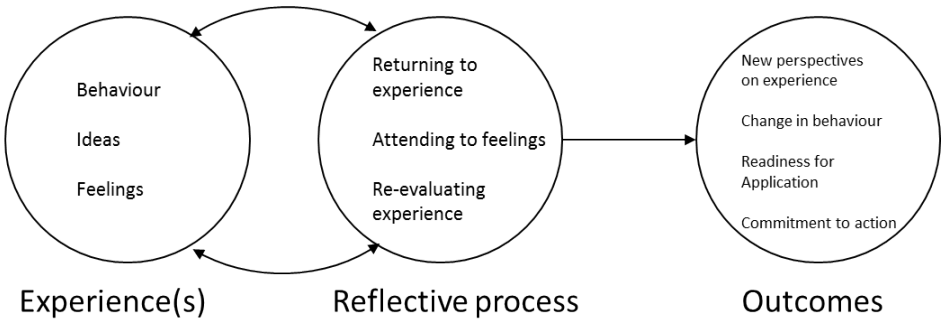
Event
- Brukerstyrt: video, bilder, notater, lyd, barcode, "følelser"
- Systemstyrt: tid

Koblinger
- Flere typer data kan samles i en event

Vurderinger
- Hver event kan vurderes individuelt

timeline

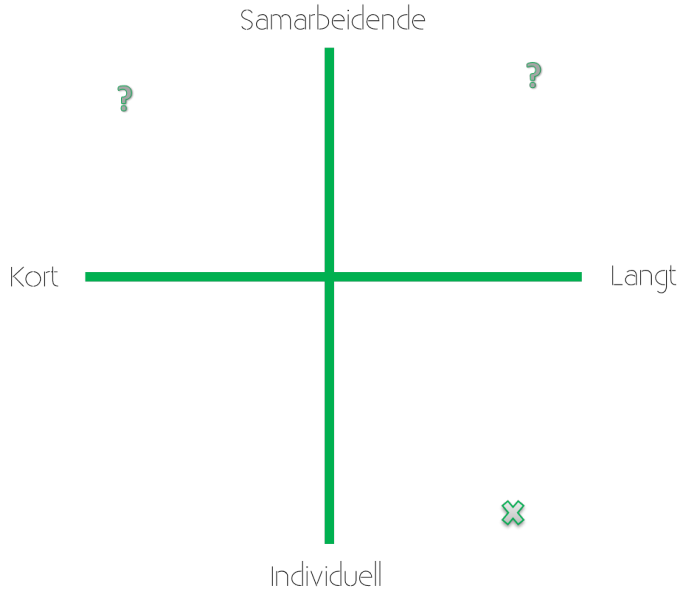
Teori



Boud et al. 1985

timeline

Flere dimensjoner for bruk av appen



timeline

Scenario

Bruk av applikasjonen i et skoleprosjekt over et lengre tidsrom

En gruppe i faget Eksperter i Team skal iløpet av et semester jobbe med å lage seg et bilde av byggenæringen i Trondheim, ved og blant annet se på smarte løsninger som kan være med på å spare både økonomi og miljø.

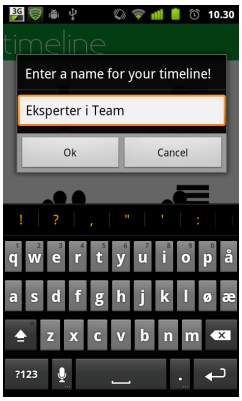
I den sammenheng ønsker gruppen å benytte Tidslinjeapplikasjonen for å samle informasjon, samt for å få en oversikt over fremgangen i prosjektet. Dette håper gruppen kan være med å belyse ulike utfordringer gruppen har hatt i prosessen.

På de neste slidene vil vi belyse mulighetene applikasjonen gir gjennom å vise hvordan den kan brukes i et slik type prosjekt.

timeline

Opprettelse av prosjekttidslinje

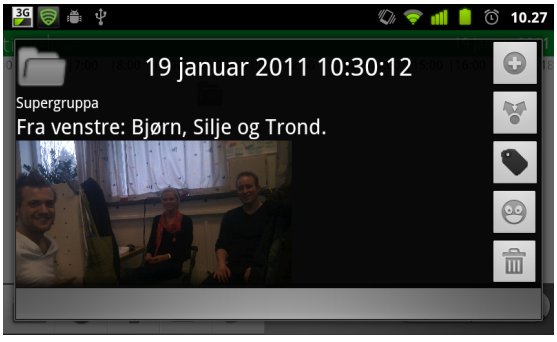
Karl oppretter en tidslinje som skal følge gruppen fra hans standpunkt gjennom prosjektperioden.



timeline

Første dag 1: Gruppen

Første dag i semesteret tar Karl et bilde av gruppen sin.



timeline

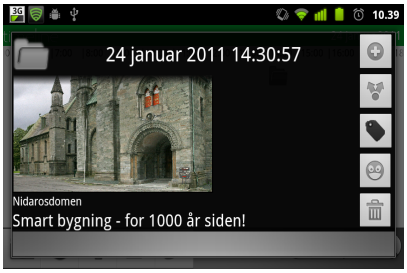
Senere: Litteratursøk

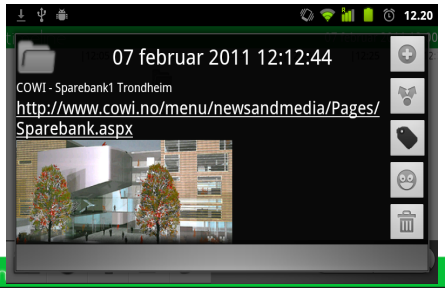
I begynnelsen av prosjektperioden er fokuset på litteratursøk. Karl legger til funnene sine i applikasjonen

timeline

Utflukter

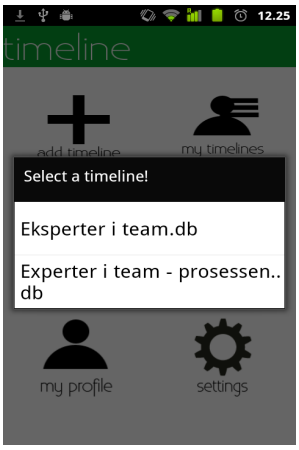
I løpet av semesteret drar gruppen på flere utflukter. Karl bruker også applikasjonen når han på fritiden ser bygninger som kan være relevante for prosjektet.

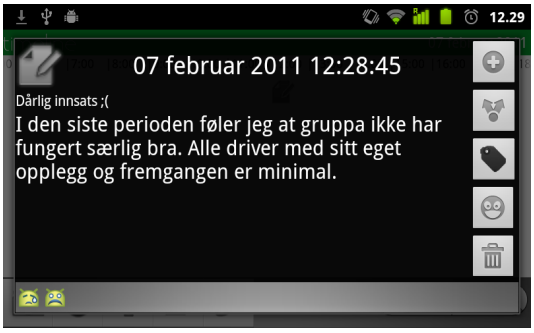




Personlige Tanker?

Karl bruker også applikasjonen til å skrive ned tanker rundt arbeidsprosessen i prosjektet.

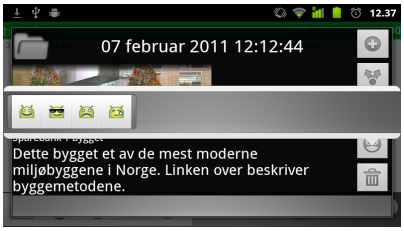


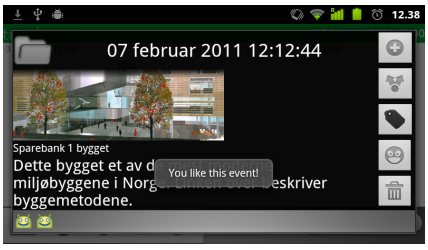


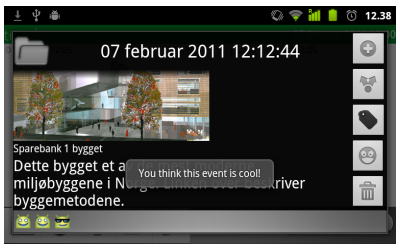
timeline

Vurderinger

Ettersom mer og mer data blir samlet inn, går Karl gjennom tidslinjen sin og vurderer de dataene han har.



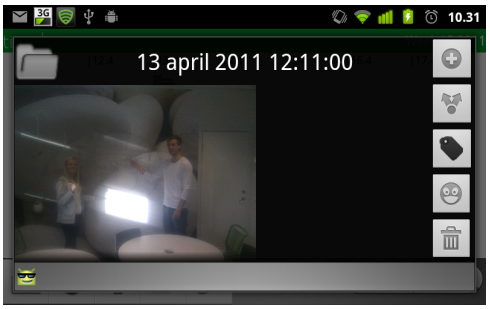


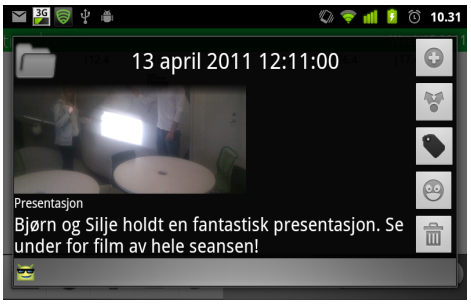


timeline

Presentasjon

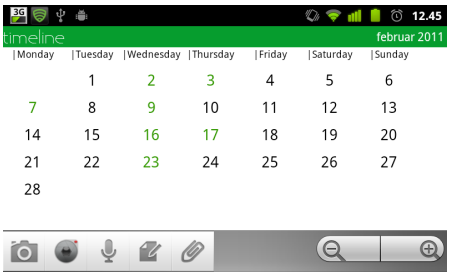
På slutten av semesteret holder gruppen en presentasjon av arbeidet sitt. Karl tar opp presentasjonen og tilbakemeldinger med applikasjonen.

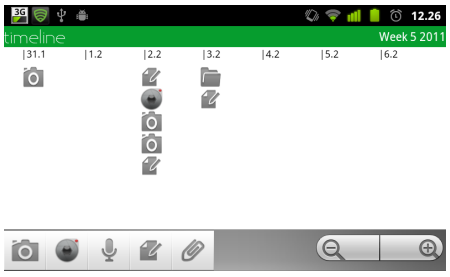


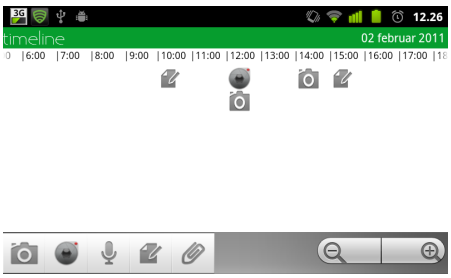


timeline

Finne tilbake til hendelser







timeline

Hvordan bidrar applikasjonen til arbeidet?

- Hvordan støttes refleksjon nå?

I slutten av semesteret skal gruppen skrive rapport om hva de har jobbet med. I tillegg til data og informasjon om emnet som er samlet med applikasjonen, så kan gruppen bruke tidslinjen til å beskrive hvordan arbeidet har gått frem.

Når var det mest fremgang? Var det noen perioder med konflikter?
Hvilke oppdagelser hadde stor betydning for prosjektet?

Første versjon av applikasjonen støtter refleksjon med:

- Innsamling av enkel informasjon
  - Items
  - Vurdering med emotions
- Tilbakeblikk på informasjon
- Fleksibilitet med tanke på tidsaspekt?

timeline

Veien videre

- Hvordan forbedre støtte for refleksjon?

timeline

Idéer

timeline

1.Kombinere tidslinjer og events

- Gi brukerne mulighet til å slå sammen forskjellige tidslinjer, eller "dra ut" en tidslinje fra en annen.
  - Velge en mengde items og starte en ny tidslinje med alle disse inkludert
  - Starte en ny tidslinje med utgangspunkt i et event/item
- Gjør det lettere å identifisere nye "learning trajectories"

timeline

2.Distribuerte (delte) tidslinjer

- Flere brukere skal kunne jobbe sammen på samme tidslinje samtidig
  - Fremme samarbeid
  - Gi tilbakemeldinger i "realtime"
  - Dele tidslinjer mellom hverandre
  - Vurdere innholdet i hverandres tidslinjer
- Alternativer - Hva er best for refleksjon?:
  - Tidslinjen dannes av en gruppe
    - "creator" er gruppen
  - Tidslinjen dannes av individer
    - Spesifisere at "creator" er individ

timeline

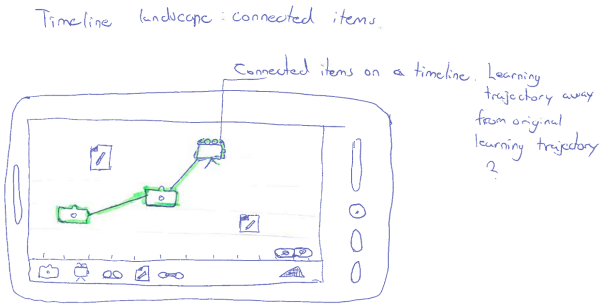
3.Tagging av experience-events i tidslinjer

- Gi brukerne mulighet til å tagge experience-events i tidslinjen med forskjellige tags som kan "sortere"/koble events innad i en tidslinje såvel som mellom forskjellige tidslinjer
  - Systemdefinerte vs brukerdefinerte tags?
  - Lage tidslinjer basert på tags (extract en tidslinje av tagga experience-events)
  - Gi events identifikatorer basert på tags (fargeomriss, spesielle ikoner e.l.)

timeline

4.Linke items sammen i tidslinje

- Gi brukeren mulighet til å vise en link mellom ulike events i en tidslinje, basert på tags.
  - events med samme tag kan linkes sammen grafisk for å gi brukeren en oversikt over hva som kan relateres.
  - *Eks:*



timeline

5.Identifisere experience-events basert på vurderinger

- Gi experience-events med vurderinger et eget utseende basert på vurderingene som er gitt.
  - Skille experience-events basert på vurderinger/tilbakemeldinger
  - Eks: Rød - viktig, Grønn - relevant, gul - dårlig etc.

timeline

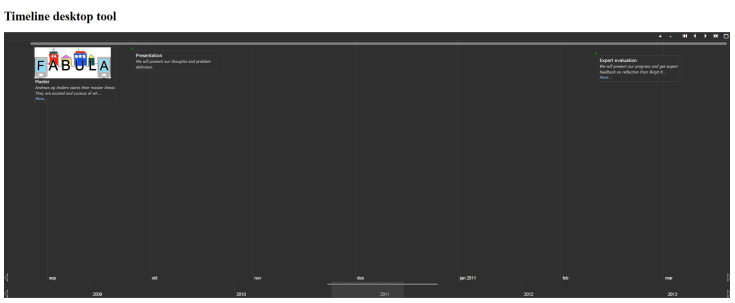
6.Fasilitert læring

- Støtte for "admin-modus" der ekstern person/system kan legge inn annotasjoner i annen brukers tidslinje
  - Lærer: legge inn læringsmål, spørsmål, quizzer, mood report, oppgaver i tidslinjen - styrt læring
  - Fasilitator: fasilitere i tidslinjen, refleksjonsspørsmål, observasjoner - styrt refleksjon

timeline

7.Deling av tidslinje til andre medier

- Gi brukerne mulighet til å eksportere sine tidslinjer til et annet medie - typisk en webressurs
  - Presentere tidslinje i en webbrowser
    - Deling av ressurser
    - Presentasjon av tidslinje



timeline

8.Tettere integrasjon med telefon/andre media

- Støtte for SMS rett inn i tidslinje
- Støtte for å følge twitter/facebook feeds i tidslinjen
- Støtte for å følge blogger

- Integrere med andre systemer spesifikke for visse prosjekter
  - Dropbox
  - Versjonskontroll

Appendix B

Web Services

B.1 Implementation

The RESTful Web services were implemented using JAX-RS (Java™ API for RESTful Web Services, JSR-311)[1] and Sun's reference implementation for JAX-RS, Jersey[2]. The different services for acting on the resources were created using the `@GET`, `@PUT` and `@DELETE` annotations of JAX-RS. The services accepts XML and JSON. Java Architecture for XML Binding (JAXB)[3] is used to automatically map Java classes to XML or JSON, and vice verca. JAX-RS has built-in support for JAXB. "The JAXB annotations within a RESTful Web service indicate the XML structure that should be generated from that code" [12]. The model classes are therefore annotated with the JAXB annotations `@XmlRootElement` and `@XmlAccessorType (XmlAccessType.PROPERTY)`, indicating the root of the XML/JSON and that the properties of the class should be included as elements.

These services were then exposed to the Web by deploying the services as a Java Servlet[4] on the server using Jersey ServetContainer.

[1] http://jsr311.java.net/
[2] http://jersey.java.net/
[3] http://jaxb.java.net/
[4] http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Servlets.html

B.2 Example of a RESTful Web Service

This section describes the construction of a RESTful Web service, using one of our resources as an example. Listing B.1 shows the complete resource. Subsequent listings are parts from this resource. For a full reference of all the services, see Appendix B.3.

```
1  @PUT
2  @Consumes({"application/json", "application/xml"})
3  @Produces({"application/json", "application/xml"})
4  @Path("/experience/")
5  public Experience putExperience(Experience experience) {
6      PersistenceManager pm = PMF.get().getPersistenceManager();
7      try {
8          pm.makePersistent(experience);
9      } finally {
10         pm.close();
11     }
12
13     return experience;
14 }
```

Listing B.1: The full resource

```
@PUT
```

This line describes that this method will process HTTP PUT requests.

```
@Consumes({"application/json", "application/xml"})
@Produces({"application/json", "application/xml"})
```

The `@Consumes` annotation is used to specify the MIME media types sent from the client that a resource can consume [10]. This method accepts XML or JSON as content of the HTTP request. Correspondingly the `@Produces` annotation describes the MIME media types this method produces and can deliver to the client.

The client describes the MIME types of the content and response by setting the header fields `Content-type:` [5] and `Accept:`.

```
@Path("/experience/")
```

Listing B.2: The @Path annotation

[5] http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

The `@Path` annotation's value is a relative URI path indicating where the Java class will be hosted. One can also embed variables in the URIs to make a URI path template [10]. For example, in the corresponding `@GET` resource, the username of the user requesting the experiences is passed to the application as a variable in the URI, like this, `/experience/{username}`.

```
public Experience putExperience(Experience experience) {
    PersistenceManager pm = PMF.get().getPersistenceManager();
    try {
        pm.makePersistent(experience);
    } finally {
        pm.close();
    }

    return experience;
}
```

This is the method that execute the actual request. The `Experience`-object passed as parameter is the object automatically mapped from XML or JSON by JAXB.

B.3 Complete set of Web Services

This chapter includes a JavaDoc of the complete set of Web services developed in the thesis.

Package Class Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD

FRAMES NO FRAMES All Classes
DETAIL: FIELD | CONSTR | METHOD

com.fabula.timeline.service.rest.impl

Class TimelineResource

```
java.lang.Object
  └ com.fabula.timeline.service.rest.impl.TimelineResource
```

```
public class TimelineResource
extends java.lang.Object
```

REST services for Fabula Timeline Android Application.

Author:
andekr, andrstor

Constructor Summary

`TimelineResource()`

Method Summary

com.fabula.timeline.service.model.Group	`addUserToGroup(java.lang.String groupid, java.lang.String username)` PUT Adds a user to a group **URL: /group/{groupid}/user/{username}/** TODO: Register user if not registered
com.fabula.timeline.service.model.Experiences	`GetAllExperiences()` GET Gets all experiences in server **URL: /experiences/** Gets all the experiences stored.
java.lang.String	`getAverageMoodByID(java.lang.String experienceid)` GET Gets the average mood of an experience by ID **URL: /mood/id/{experienceid}/** Produces a string with the average mood as integer.
java.lang.String	`getAverageMoodByName(java.lang.String title)` GET Gets the average mood of an experience by title Note: Not as accurate as ID, and is case sensitive **URL: /mood/title/{title}/**
java.lang.String	`getEvents()` GET Get events as String (Mainly for testing purposes) **URL: /events/** Produces toString of all events as String
com.fabula.timeline.service.model.Experiences	`GetExperiences(java.lang.String username)` GET Gets all experiences of a user **URL: /experiences/{username}** Takes username of the user to get experiences Produces the all experiences shared with the user as JSON or XML
com.fabula.timeline.service.model.Group	`getGroup(java.lang.String groupid)` GET Gets the group with the id sent as parameter **URL: /group/{groupid}** Produces JSON or XML with group information.
com.fabula.timeline.service.model.Groups	`getGroups()` GET Gets all the groups **URL: /groups/** Produces JSON or XML with list of all groups
com.fabula.timeline.service.model.Groups	`getGroupsOfUser(java.lang.String username)` GET

	Gets the groups the user with the username given as parameter **URL: /groups/{username}** Produces JSON or XML with list of groups the user is member of.
com.fabula.timeline.service.model.User	**getUser**(java.lang.String username) GET Gets the user with the username sent as parameter **URL: /user/{username}** Produces JSON or XML with user information.
com.fabula.timeline.service.model.Users	**getUsers**() GET Gets all the users **URL: /users/** Produces JSON or XML with all users.
java.lang.String	**isUserInGroup**(java.lang.String groupid, java.lang.String username) GET Gets if user is member in a group **URL: /group/{groupid}/{username}/**
com.fabula.timeline.service.model.Event	**putAverageMoodByID**(java.lang.String id, java.lang.String username, int mood) PUT Inserts a mood to an experience only by parameters, and no content needed Uses experience id to identify experience **URL: /mood/id/{id}/{username}/{mood}/**
com.fabula.timeline.service.model.Event	**putAverageMoodByName**(java.lang.String title, java.lang.String username, int mood) PUT Inserts a mood to an experience only by parameters, and no content needed Uses experience title to identify experience **URL: /mood/title/{title}/{username}/{mood}/**
com.fabula.timeline.service.model.Event	**putEvent**(com.fabula.timeline.service.model.Event event) PUT Store or update one Event **URL: /event/** Consumes a complete event as JSON or XML and stores/updates in Google App Engine.
com.fabula.timeline.service.model.Experience	**putExperience**(com.fabula.timeline.service.model.Experience experience) PUT Store or update one Experience **URL: /experience/** Consumes a complete experience as JSON or XML and stores/updates in Google App Engine.
com.fabula.timeline.service.model.Experiences	**putExperiences**(com.fabula.timeline.service.model.Experiences experiences) PUT Stores/updates all experiences in server **URL: /experiences/** Stores all the experiences sent in.
com.fabula.timeline.service.model.Group	**putGroup**(com.fabula.timeline.service.model.Group group) PUT Registers a group **URL: /group/** Consumes a JSON or XML with the group information.
com.fabula.timeline.service.model.User	**putUser**(com.fabula.timeline.service.model.User user) PUT Registers a user **URL: /user/** Consumes a JSON or XML with the user information.
java.lang.String	**removeGroup**(java.lang.String groupid) DELETE Deletes a group **URL: /group/{groupid}** Takes a group id as parameter.
java.lang.String	**removeUserFromGroup**(java.lang.String groupid, java.lang.String username) DELETE Removes a user form a group **URL: /group/{groupid}/user/{username}/**

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

TimelineResource

```
public TimelineResource()
```

Method Detail

putEvent

```
public com.fabula.timeline.service.model.Event putEvent(com.fabula.timeline.service.model.Event event)
```

PUT
Store or update one Event
URL: /event/
Consumes a complete event as JSON or XML and stores/updates in Google App Engine. Requires that the "father-experience" is already registered. Produces the registered event as JSON or XML
Example of JSON:

```
{ "className":"Event", "creator":"anderskri@gmail.com", "eventItems":[ { "className":"SimpleNote",
"creator":"anderskri@gmail.com", "id":"b44f63a4-a17e-468d-bc4f-149430a8aa20", "noteText":"This is the content of an
example note.", "noteTitle":"This is an example of a note." } ], "experienceid":"13b9b0d2-c8cd-4436-9be3-bddef9ebd5a2",
"id":"d270281e-5579-429d-8570-5de901d52a0f", "latitude":63.41652922, "longitude":10.40268466,
"datetimemillis":1301904788593, "shared":true, "mood":0, "average":false }
```

---

getEvents

```
public java.lang.String getEvents()
```

GET
Get events as String (Mainly for testing purposes)
URL: /events/
Produces toString of all events as String

---

putExperience

```
public com.fabula.timeline.service.model.Experience putExperience(com.fabula.timeline.service.model.Experience experience)
```

PUT
Store or update one Experience
URL: /experience/
Consumes a complete experience as JSON or XML and stores/updates in Google App Engine. Also stores all children Produces the registered experience as JSON or XML
Example of JSON:

```
{ "creator":"anderskri@gmail.com", "events":[ { "className":"MoodEvent", "creator":"anderskri@gmail.com",
"datetimemillis":"1301556003782", "experienceid":"4cfb71dc-30f6-47e5-829a-a21e08e568b9", "id":"43799bf7-53fa-4acc-8045-
0d8a0f29ea8b", "latitude":"63.41652448", "longitude":"10.4027416", "mood":"1", "shared":"true" }, {
"className":"MoodEvent", "creator":"andrstor87@gmail.com", "datetimemillis":"1301556030401", "experienceid":"4cfb71dc-
30f6-47e5-829a-a21e08e568b9", "id":"9689012d-520b-4bf1-b19f-06c1072f9f8b", "latitude":"63.4165383",
"longitude":"10.40269734", "mood":"-2", "shared":"true" }, { "className":"Event", "creator":"andrstor87@gmail.com",
"datetimemillis":"1301556205024", "emotionList":[ { "emotionType":"LIKE", "emotionid":"22d08b27-0266-4561-8a16-
96b86b099aa9" } ], "eventItems":[ { "className":"SimplePicture", "creator":"andrstor87@gmail.com", "filename":"-
1779601815.jpg", "id":"91e0b5ba-a384-40b4-88ac-edecf8c28e91" }, { "className":"SimpleNote",
"creator":"andrstor87@gmail.com", "id":"9b3dd9a7-ac7e-44e2-a62d-e6191c2a3aff", "noteText":"The most productive in the
world.", "noteTitle":"My workplace" } ], "experienceid":"4cfb71dc-30f6-47e5-829a-a21e08e568b9", "id":"b05348b2-d827-
4192-b153-c283ac251586", "latitude":"63.41654401", "longitude":"10.4027174", "mood":"0", "shared":"true" }, {
"className":"Event", "creator":"anderskri@gmail.com", "datetimemillis":"1301564716999", "eventItems":[ {
"className":"SimpleNote", "creator":"test@timelineapp.no", "id":"5c58f9ae-3f6a-4a79-b09e-a3ad9b58d26b",
"noteText":"Finally Fabula is on Timeline!", "noteTitle":"Fabula on Timeline" } ], "experienceid":"4cfb71dc-30f6-47e5-
829a-a21e08e568b9", "id":"adf77be1-fc8d-4a5b-b465-16c4d09a68f6", "latitude":"63.41654447", "longitude":"10.402711",
"mood":"0", "shared":"true" }, { "className":"Event", "creator":"test@timelineapp.no", "datetimemillis":"1301567596334",
"eventItems":[ { "className":"SimplePicture", "creator":"test@timelineapp.no", "filename":"1689419062.mp4",
"id":"5fc3463d-31ef-42a4-8b5e-3ae5a9a58f57" } ], "experienceid":"4cfb71dc-30f6-47e5-829a-a21e08e568b9", "id":"76d46599-
eda8-487d-a829-b5517bc6ab43", "latitude":"0.0", "longitude":"0.0", "mood":"0", "shared":"true" } ], "id":"4cfb71dc-30f6-
47e5-829a-a21e08e568b9", "shared":"true", "sharingGroup":"4bbd7541-5c49-45e5-87c9-d75ac1e3a36a", "title":"Fabula" }
```

GetExperiences

```
public com.fabula.timeline.service.model.Experiences GetExperiences(java.lang.String username)
```

GET
Gets all experiences of a user
URL: /experiences/{username}
Takes username of the user to get experiences Produces the all experiences shared with the user as JSON or XML

GetAllExperiences

```
public com.fabula.timeline.service.model.Experiences GetAllExperiences()
```

GET
Gets all experiences in server
URL: /experiences/

Gets all the experiences stored. Mainly for testing and administrative purposes. Produces the all experiences as JSON or XML

---

putExperiences

```
public com.fabula.timeline.service.model.Experiences putExperiences(com.fabula.timeline.service.model.Experiences experiences)
```

PUT
Stores/updates all experiences in server **URL: /experiences/**
Stores all the experiences sent in. Mainly for testing and administrative purposes. Produces the all experiences as JSON or XML

---

putUser

```
public com.fabula.timeline.service.model.User putUser(com.fabula.timeline.service.model.User user)
```

PUT
Registers a user
URL: /user/
Consumes a JSON or XML with the user information. Currently only a username. Produces the JSON or XML of the registered user.
Example JSON:
`{"username":"yourusername@mail.com"}`

---

getUser

```
public com.fabula.timeline.service.model.User getUser(java.lang.String username)
```

GET
Gets the user with the username sent as parameter
URL: /user/{username}
Produces JSON or XML with user information.

Parameters:
`username` - of user to get
Returns:
`User` the user with the username sent as parameter

---

getUsers

```
public com.fabula.timeline.service.model.Users getUsers()
```

GET
Gets all the users
URL: /users/ Produces JSON or XML with all users.
Example of returning JSON:
`{"users":[{"username":"user1@gmail.com"},{"username":"user2@gmail.com"},{"username":"test@timelineapp.no"}]}`

Returns:
`Users` the users

---

getGroup

```
public com.fabula.timeline.service.model.Group getGroup(java.lang.String groupid)
```

GET
Gets the group with the id sent as parameter
URL: /group/{groupid}
Produces JSON or XML with group information.

Parameters:
`groupid` - of group to get
Returns:
`Group` the group with the id sent as parameter

---

putGroup

```
public com.fabula.timeline.service.model.Group putGroup(com.fabula.timeline.service.model.Group group)
```

PUT
Registers a group
URL: /group/
Consumes a JSON or XML with the group information. The JSON should contain at least one member(the creator). Produces the JSON or XML of the registered group.

---

removeGroup

```
public java.lang.String removeGroup(java.lang.String groupid)
```

DELETE
Deletes a group
URL: /group/{groupid}
Takes a group id as parameter. Deletes the group with the given id. Returns a boolean as String of the status of the delete. Not implemented yet.

Parameters:
`groupid` - group id
Returns:
String true if successful. Not implemented

---

getGroupsOfUser

```
public com.fabula.timeline.service.model.Groups getGroupsOfUser(java.lang.String username)
```

GET
Gets the groups the user with the username given as parameter
URL: /groups/{username}
Produces JSON or XML with list of groups the user is member of.

Parameters:
`username` - of user to get groups from.
Returns:
`Groups` the user is member of.

---

getGroups

```
public com.fabula.timeline.service.model.Groups getGroups()
```

GET
Gets all the groups
URL: /groups/
Produces JSON or XML with list of all groups

Returns:
`Groups` the user is member of.

---

isUserInGroup

```
public java.lang.String isUserInGroup(java.lang.String groupid,
                                       java.lang.String username)
```

GET
Gets if user is member in a group
URL: /group/{groupid}/{username}/

Parameters:
`groupid` -
`username` -
Returns:
String true if user is in group

---

addUserToGroup

```
public com.fabula.timeline.service.model.Group addUserToGroup(java.lang.String groupid,
                                                               java.lang.String username)
```

PUT
Adds a user to a group
URL: /group/{groupid}/user/{username}/
TODO: Register user if not registered

Parameters:
`groupid` - Id of group to add user to
`username` - username of already registered user
Returns:
JSON or XML with the group

---

removeUserFromGroup

```
public java.lang.String removeUserFromGroup(java.lang.String groupid,
                                             java.lang.String username)
```

DELETE
Removes a user form a group
URL: /group/{groupid}/user/{username}/

Parameters:
`groupid` - id of group to remove user from
`username` - of user to remove
Returns:
String true if user is removed from group

---

getAverageMoodByID

```
public java.lang.String getAverageMoodByID(java.lang.String experienceid)
```

GET
Gets the average mood of an experience by ID
URL: /mood/id/{experienceid}/
Produces a string with the average mood as integer.

Parameters:
`experienceid` - Experience id of the experience to get average mood from.

---

getAverageMoodByName

```
public java.lang.String getAverageMoodByName(java.lang.String title)
```

GET
Gets the average mood of an experience by title
Note: Not as accurate as ID, and is case sensitive
URL: /mood/title/{title}/

Parameters:
`title` - title of the experience to get average mood from.

---

putAverageMoodByName

```
public com.fabula.timeline.service.model.Event putAverageMoodByName(java.lang.String title,
                                                                     java.lang.String username,
                                                                     int mood)
```

PUT
Inserts a mood to an experience only by parameters, and no content needed
Uses experience title to identify experience
URL: /mood/title/{title}/{username}/{mood}/

Parameters:
`title` - of experience to insert mood
`username` - of user to set as creator
`mood` - to insert. int from -2 to 2. Very sad to very happy.
Returns:
the created `Event`

---

putAverageMoodByID

```
public com.fabula.timeline.service.model.Event putAverageMoodByID(java.lang.String id,
                                                                   java.lang.String username,
                                                                   int mood)
```

PUT
Inserts a mood to an experience only by parameters, and no content needed
Uses experience id to identify experience
URL: /mood/id/{id}/{username}/{mood}/

Parameters:
`id` - of experience to insert mood
`username` - of user to set as creator
`mood` - to insert. int from -2 to 2. Very sad to very happy.
Returns:
the created `Event`

Appendix C

Timeline Application Quick Start Guide

This is the quick start guide of the Timeline application handed out to users at the Mirror GA.

TimelineApplication

Timeline quick start guide

timeline

What is this:

This application allows you to share your experiences in learning, work or social life with your friends, co-workers and family!

Choose what to share, and what to keep private!

Reflect on your work by collecting short notes, pictures and videos. Go back and revisit your experiences directly in your timeline or send them to your e-mail.

And much more...

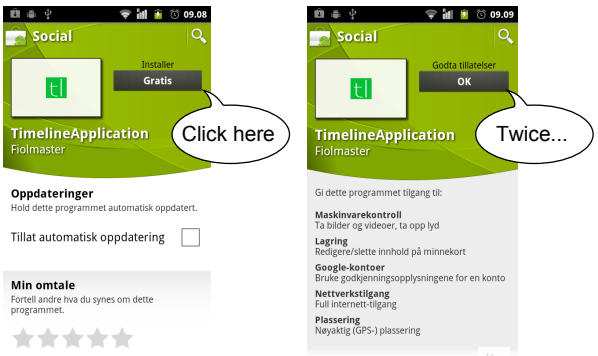
timeline

1. Installation from Android Market

Two download options:

1.1 Scan this barcode

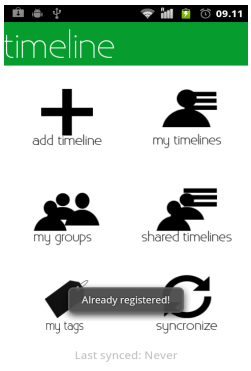
1.2 Visit Android Market: http://bit.ly/timelineapp



timeline

2. Start the application

2.1 The application will register a user with your Google account:



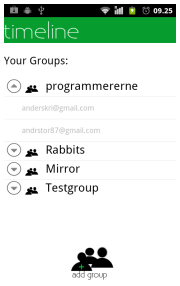
Tip!
The application uses standard android navigation:

- Back button - takes you one level back
- Menu button - gives additional options to the current screen
- Long click - gives additional options to the pressed item

timeline

3. Getting started

To be collaborative you need someone to collaborate with.
This is managed by **groups.**



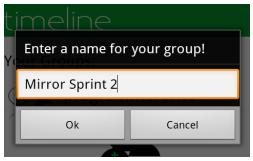
Two ways to become member of group:
3.1 Create a new group, and add other users to the group
3.2 Someone else adds you as member of a group

timeline

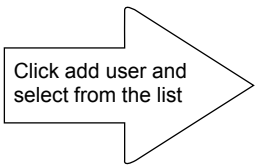
3.1 Create a new group and add users

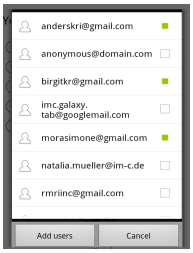
3.1.1 Click on the **add group** button:

3.1.2 Enter a name for your group:



3.1.3 Add new user(s) to your group by long-clicking the group name:

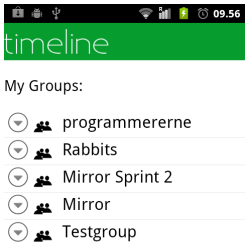




timeline

3.2 Getting added to a group

If someone adds you like showed in the previous slide, you will be automatically added to the group and it will show in "**my groups**"



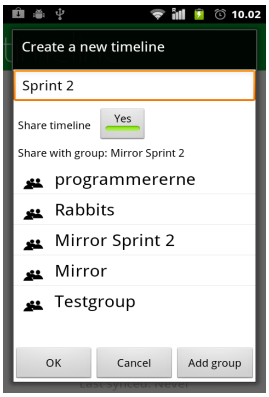
We will add all users in the GA to the "Mirror"-group that will be used during the GA.

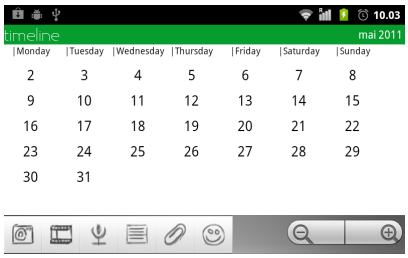
timeline

4. Create a new timeline

4.1 Click on the **"add timeline"** button:

4.2 Give your timeline a name and select a group to share it with:

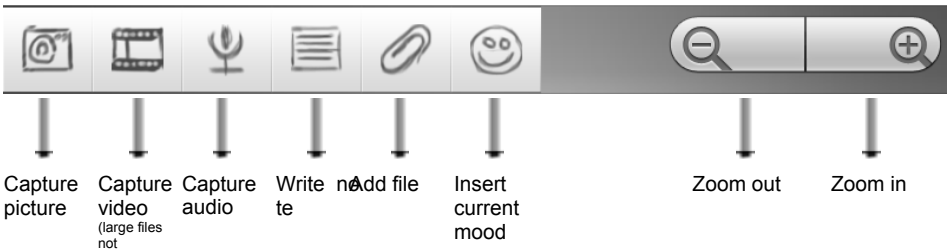




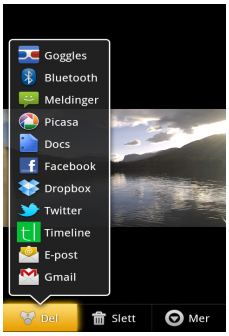
timeline

5 Adding content

5.1 Using toolbar in the timeline



5.2 You can also add content from other applications (example from gallery):



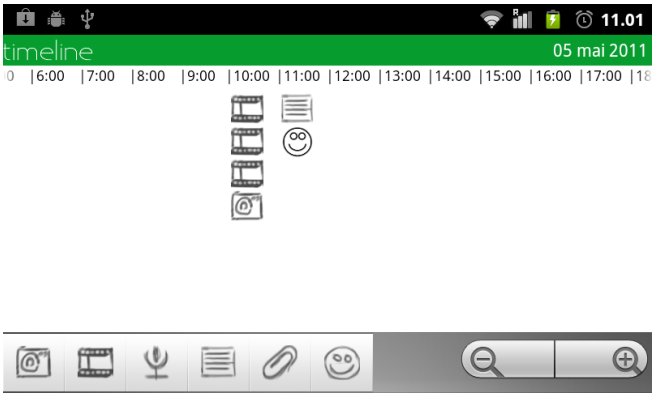
timeline

6. Viewing content

Open toolbar: you can add more items to this content.

timeline

7. View all content in timeline



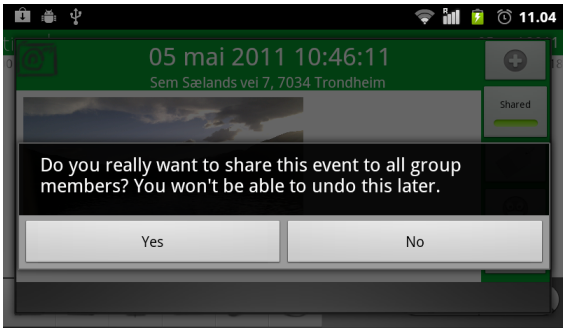
All your content will show in your timeline at the time of creation

timeline

8. Sharing

8.1 Sharing your content:

Click on the share button when viewing content



All your content will by default be set to ***Not shared***

timeline

8. Sharing

8.2 Getting all content shared with you:

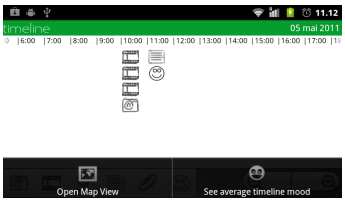
Click on the "**synchronize**" button:

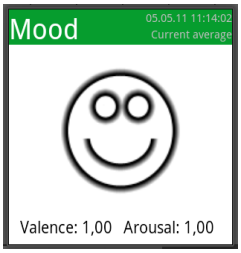
All timelines and content shared with you will be synchronized to your client.

timeline

9. Average timeline mood

In a shared timeline you can see the average mood of the timeline by clicking the **menu** button while viewing the timeline and select "**see average timeline mood**":





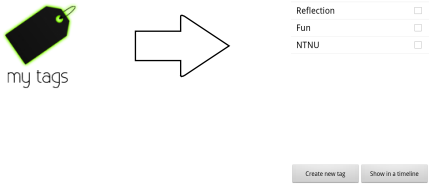
This relies on users sharing their mood in the timeline

timeline

10. Extra features

10.1 See all content in a timeline in a mapview:

10.2 Use tags to organize your content:



timeline

10. Extra features

10.3 Send activity report:

From the dashboard (main menu screen) click the menu button on the phone and select "send activity report". You will then receive an e-mail with a report of all the activity in the timelines you are a part of:

timeline

Summary for use of the app in GA

1. Download and install the application

2. Synchronize - you should now have the "MirrorGA2" timeline and "Mirror" group.
   If not, talk with us and we will add you to the group

3. Add and share content!

You are also free to create your own groups and timelines.
Have fun using the application!

Just ask if anything is unclear or you experience any problems.

timeline

Any questions?

Useful links:

Quick start guide (this): http://bit.ly/timelineqs

Link to Android Market: http://bit.ly/timelineapp

timeline