

Tor-Martin Holen

Creating a Social Media App for Goal Accomplishment

Master's thesis in Applied Computer Science
Supervisor: Mariusz Nowostawski
June 2019

Tor-Martin Holen

Creating a Social Media App for Goal Accomplishment

The Social Media App, SetGoals

Master's thesis in Applied Computer Science
Supervisor: Mariusz Nowostawski
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science





Norwegian University of
Science and Technology

Creating a Social Media App for Goal Accomplishment

Tor-Martin Holen

01.06.19

Master's Thesis
30 ECTS

Supervisor: Mariusz Nowostawski

Abstract

The work covers persuasive technology and how this can be used to create a social media application for goal accomplishment. The work attempts to discover if personality traits assessed through the BFI-44 affects how the users behave in the application in the form of content they produce. However, the work lacked a large active user group. The results do also covers advertising results obtained from campaigns with Google Ads and Facebook Ads. In total the application received 102 registered account, but only 18 can be considered active users. The prototype is created with Flutter, Dart and Firebase. Social Media Application, SetGoals, is open sourced, and anybody is free to fork it and use it for non-commercial gains.

Contents

Contents	i
List of Figures	iv
List of Tables	v
List of Listings	1
1 Introduction	1
1.1 Related works	2
1.1.1 VivoSpace	4
1.1.2 ActiveShare	4
1.1.3 EcoIsland	5
1.1.4 BinCam	5
1.1.5 Goal Tracking	5
1.1.6 Papers	5
2 Methodology	7
2.1 Literature Search	7
2.2 Data collection	7
2.3 Classifying Users	8
2.3.1 Bartle’s taxonomy	8
2.3.2 Myers-Briggs Type Indicator	8
2.3.3 BrainHex	9
2.3.4 Persuasive Potential Questionnaire	9
2.3.5 Five Factor Model	9
2.4 Data processing	10
2.5 Plan A	10
2.5.1 User Recruitment	10
2.5.2 Risks of Plan A	11
2.6 Plan B - The contingency plan	11
2.7 Theory	11
2.7.1 Storyboards or in-app-test	12
2.7.2 Socially Oriented - Persuasive Strategies	12
2.8 Hypotheses and Data Logging	12
2.8.1 User Retention	13
2.8.2 Posts and goals	13
2.8.3 User-set Reminders	14
2.8.4 Privacy	14

2.9	Flutter and Dart	14
2.9.1	Quirks and Features of Dart	15
2.10	Firebase	16
2.10.1	Firebase Authentication	16
2.10.2	Cloud Firestore	17
2.10.3	Cloud Storage	18
2.10.4	Cloud Functions	19
2.10.5	Firebase Cloud Messaging	20
2.11	The Prototype	20
2.11.1	The Goal List	22
2.11.2	Login Process	22
2.11.3	Sumarizing the application	23
3	Results	30
3.1	Test Account	30
3.2	User Recruitment from own Network	30
3.3	User Recruitment from Advertisement	30
3.3.1	Jodel Advertised Post	30
3.3.2	Google Ads	30
3.3.3	Facebook Ads	32
3.3.4	Result of Advertising	32
3.3.5	Analytics	36
3.3.6	App Installs and Deletes	36
3.3.7	Differences in the advertisements	37
3.4	User Feedback	37
3.4.1	Questionnaire	37
3.4.2	In-app feedback	39
3.5	User Behaviour	39
3.5.1	Implemented Persuasive Strategies	39
3.6	Processing data	40
3.7	Economical Aspects	41
3.8	Architecture	42
3.9	Optimizing for Mobile	42
4	Discussion	50
4.1	Technical Discussion and Application Design	50
4.1.1	The Home Feed	50
4.1.2	The Goal List	50
4.1.3	The Map	50
4.1.4	User Recruitment	50
4.1.5	Usability	50
4.1.6	Architecture	51

4.1.7	Packages	52
4.1.8	Technical Debt	52
4.1.9	Privacy	52
4.1.10	Ways to improve the data collection	52
4.2	Flutter	53
4.3	Legal Aspects	53
4.3.1	GDPR	53
4.3.2	App Review	53
4.4	The app	54
4.4.1	Updating the app	54
4.5	Ethical Aspects	55
4.5.1	Publication bias	55
4.6	Persuasive Technology Techniques (PTT)	55
4.7	Open Source	55
4.8	Complying with GDPR and NSD	55
4.9	Loss of users	56
4.10	External Factors	56
4.11	Login/Notification bug	57
4.12	Bug: Wrong Image When Posting	57
5	Conclusion	58
5.1	Main Contributions	58
5.2	Reflections	58
5.3	Further Work	59
5.3.1	Persuasive Software Features	59
5.3.2	User Testing	60
5.3.3	Sharing	60
5.4	Deprecation	60
5.5	Bad internet connections	60
5.6	Limitations of this work	60
	Appendices	61
A	Designing the questionnaire	62
A.1	Original Questionnaire	62
A.2	Revised Questionnaire	62
B	Content Analysis	64
	Bibliography	71

List of Figures

1	Flutter’s Layered Architecture	17
2	A technology overview.	21
3	Prototype: Goal List Feature	22
4	Prototype: Map Feature	23
5	Prototype: Create Goal Feature	24
6	Prototype: Login Screens	25
7	Prototype: Reminder Feature	27
8	Prototype Overview: Startup and Authentication Flow	28
9	Application Overview	29
10	Advertised Post on Jodel	31
11	Google Ads: Overview	32
12	iOS Installs	33
13	Difference Between Active Users: Android vs iOS	36
14	Installs and Uninstalls	36
15	Overview of the analytics data from Apple’s App Store Connect	37
16	Old and New Login Screens	44
17	The usefulness of the map feature	45
18	Shows the users who submitted content of a certain type (N)	45
19	Word Cloud: Text From Goals	45
20	A overview of the registrations in the application.	46
21	Correlation Analysis of Personality and Actions Taken	47
22	A box-plot of the BFI responses in the application	48
23	App Running on Different Phones	49

List of Tables

1	European Android Campaign	33
2	First World-Wide Android Campaign	34
3	Second World-Wide Android Campaign	35
4	Persuasive techniques present in the application.	40

List of Listings

2.1	Basic Flutter App: Running a Flutter App	15
2.2	Basic Flutter App: Scaffolds	16
2.3	Firestore Query Example	18
2.4	Cloud Firestore Rule Example	19
2.5	Cloud Function Example	20
2.6	Populating goal feed	26

1 Introduction

Today there exist a plethora of social media applications. These usually provide a specific set of services to their users. The most apparent example is Facebook, which is a social network for everybody. Some others are Linked In, Twitter and Instagram.

However, do these social networks help the users themselves, or are they just a waste of time? Regardless of the answer, people often have goals they want to achieve. Yet, distraction is everywhere and without a solid habit, it is easy to forget to about a personal goal. Ironically distractions quite often come in the form of notifications from social media, or various push notifications of minor importance.

The thesis aimed to create a social media application for general purpose goal accomplishment. Interestingly, literature has stated that social media approaches have shown promise in regards to app-based interventions[1], which is partially why I decided to take this approach. The other reason is that cross-platform development and cloud technology are interesting domains in rapid development, of particular interest Flutter and Firebase. This and the application will be presented in the Methods Chapter 2 at Section 2.11.

This thesis' purpose was to explore how to design and create a social media application to improve the users intent to achieve their goals. More precisely how the implementations of various software features affect the user. Results will be presented in regards to this, but user engagement was a problem in the application so the results do not have the best sample size. On the other side, the thesis will present recruitment data from various recruitment methods such as advertisement and local recruitment.

Interestingly, there are ethical issues of how these software features are implemented, such as faulty implementations can result in an opposite effect to what was desired[2]. One such case is defined by the term upward social comparison. This occurs when an individual does a self-comparison with another individual who achieves higher results or quantities of something. This can be good but it can also be bad, really bad. A common example is drinking, let us say "Adam" at 65kg compares himself with "Bob" at 100kg. Adam sees Bob drink 7 beers, and decides "if Bob drinks that much why can not I do that?". This would be a negative example of upward social comparison. F.ex an application trying to prevent binge drinking would have to consider effects such as these into account and ensure the application does not accidentally trigger unwanted behaviour in their users.

Another interesting example of such effects, while this was not an app-based intervention, but a sponsored campaign that targeted the parents of smoking teens. The purpose of the campaign was to try to get the teens to quit smoking. Unfortunately, the campaign had the opposite effect on teens. Interestingly, the campaign was sponsored by a smoking company.

The field of persuasive technology (PT), is a field within Human-Computer Interaction. The goal of PT is to persuade the user of persuasive systems into performing the desired behaviour [3, 4]. This field was highly relevant to discover potentially decent features to include in a social media application.

Do note that the term persuasion applies to different contexts, however, it is computer-mediated persuasion this thesis will mainly focus on. The term persuasion have existed from before computers became household items, and it is not limited to computer systems, research exists on human-to-human persuasion.[5]

People are different, what is persuasive to one person may not be persuasive at all to another person. Thus a lot of research has gone into categorizing traits in people that correlates with positive results to certain persuasive techniques.[6, 7] Interventions are quite related as well. PT has stated that interventions are often started without being having a theoretical framework about what works and not. Regardless of how easy it is to get a be persuaded to change something depends. First, if the individual is trying to quit doing something it may be harder due to existing habits, such as smoking and drinking. If no habits exist then it is somewhat easier, as long as the user is slowly building a habit. Finally, there is the reinforcement of a habit which is the easiest.

Literature states that the user should find select their own goal, however, I argue in addition to letting the user create their own goals, that providing them with a set of predefined local challenges, may promote social engagement in the application, as reflected by the social suggestion strategies[3].

1.1 Related works

Do note that some of what I have written here is from or based upon the related work section from my Research Project Planning report (IMT4205) and my Advanced Project Work Report(IMT4894).

Computer software is developed for various purposes. Among them are software solutions that try to promote behaviour change in their users. To facilitate behaviour change by using software there is a field known as persuasive software design[8]. The primary purpose is to assist users of persuasive software by changing their behaviour due to their own free will, i.e. no coercion or deception. [3] This is of relevance to the hypothesis as it is a field with.

In particular, the literature suggests various ways of persuading users, however, the type of persuasion that is of relevance is human-computer persuasion. A key aspect for achieving is motivation, this is consistently mentioned in multiple works [3, 9], various models and theories focus on it such as the Elaboration Likelihood Model (ELM) and Self-determination theory (SDT). ELM defines two routes to persuasion a central route and a peripheral route, the central route consists of arguments to persuade the user through understanding and users that can be persuaded through this kind of persuasion is motivated. The peripheral route is for users using cues for evaluating information, i.e. they do not have motivation or time to evaluate the actual arguments being presented. [8].

SDT is a theory about motivation devised by Deci and Ryan, it focuses particularly on motivation types as it is frequently used in many fields, some of the key points in this theory is that intrinsic motivation is the best way for a person to be motivated when he must do a task. The opposite is

extrinsic motivation which may come from feeling obligated to do something due to external factors, such as living up to expectations, threats, or incentives. In addition, there is three basic needs, competence, autonomy, and relatedness. [10]. This including computer science, quite frequently within gamification. Gamification is relevant as it is a way to increase the engagement with the application through badges, points and many other engaging features.

Various fields are relevant in terms of helping users achieve their goals, there is gamification, social influence, serious games, and as previously mentioned persuasion. Studies have shown that users with different personalities respond differently to different features. One example is [11] where they discovered that players with certain characteristics remained players of Pokemon Go longer. Other methods have been developed for classifying users such as the Bartle Test [12], which characterizes users as achievers, explorers, and achievers, although this is a criticised method. This will be discussed in the Section 2.3, as it is central to how the application is assessed.

Various researches have studied similar applications to what I am making, some examples are Pokemon Go and Foursquare. Pokemon Go ¹ is an exergame, a game that facilitates users to move around in their local community and catch digital Pokemon. In a study about Pokemon Go, a group of researchers researched if various personality traits influenced who kept using Pokemon Go over time and who stopped. They used the Big Five Personality model to asses differences in their study [11]. It is interesting to see where Pokemon Go will go in the future, as it is in rapid development. And it seems to have been a game which many papers from many backgrounds have studied. [13] Where it will go with 5G will be interesting [14].

With the Foursquare application there exists two versions, but the one of interest is the Foursquare Swarm application ² since it is the gamified version of the app. It appears there was one app in 2014 that was split into two apps, one focusing just searching for new places, and the Foursquare Swarm application with more gamified features. Anyway, Foursquare Swarm helps its users keep track of places they have been to and assists in finding new places. A study of Foursquare was done in 2013 [15], note that at the time there was only one Foursquare app and an application may change vastly over five years. The study focused on interviewing active Foursquare users, finding out which features were detrimental to them. In particular two features distinguished themselves, these were the majorship and badge system features. Majorships are earned by checking in at a place enough times during a 60-day period, however, it gets harder the more users there are checking in at the place, thus competition elements arise. Moreover, the badge system made users explore new areas to obtain badges, such as visiting museums they never would have gone to without the app [15]. However, this study shows the features liked with highly active users so the results may not be representative for all users.

Another study researched if gamification could influence logging behaviour, they researched if gamification features could increase participants tendency to log their TV-viewing habits. Their findings further contribute to the badge feature, however, they highlight that this is less effective with the older population, and more prominent with younger audiences. [16]

¹<https://play.google.com/store/apps/details?id=com.nianticlabs.pokemongo>

²<https://play.google.com/store/apps/details?id=com.foursquare.robin>

It appears most applications, within persuasive software for behavioural change is exercise or health applications, one example of health improvement is a study [17] where they tried to prevent sedentary behaviour in people working in offices by notifying them when they had been sitting still too long. Sedentary behaviour is associated with an increased risk of various health issues.

Some examples of persuasive exercise applications are Strava³, and Garmin Connect⁴ are examples of applications utilizing various techniques to persuade their user to exercise. Such as Strava may send you notifications of friends activities when it has been some time since you last used the application. In addition, there are competitive elements such as leaderboard features. Garmin Connect is heavy on badges. Both applications have social features such as news feed, with like and comment functionality.

In addition, some persuasive design patterns are presented in [8] social learning and facilitation, competition, cooperation, and recognition. In addition, they look at chains and social elements that can be worth looking further into.

In the context of gamification, one should be wary of how it is used, as it can reduce the user's motivation in the long run, however, this is very context specific, and it depends on the features used. In one case in a classroom setting students were exposed to leaderboards and badges during a course and this caused them to lose motivation and ultimately get a worse grade than the study's control group. [9]

In regards to social features, there have been found links between the type of usage users use social media. Active usage is when the user actively engages activities such as posting and commenting on content, and then there is passive usage where the user only views what his friends have posted. The latter group has been found to have worse mental wellness after engaging in social media usage. There are also negative sides such as upwards social comparison where the user compares the "perfect life" of his friends to his life. Content posted on social media is skewed towards positive events. Of particular interest is the type of content that has been found to promote well-being in users, which is targeted messages from persons with close ties. Which is one on one communication that has been shown to improve the users' well-being. Interestingly likes have very little meaning. [18]

1.1.1 VivoSpace

VivoSpace were a social-media prototype developed for healthy behaviour change, targeting diet and physical activity specifically[19]. This is perhaps the most similar application to what will be presented in this thesis. The study used the ABC framework. The application had implementations of a quite similar set of features to what is present in my application.

1.1.2 ActiveShare

A project that in many ways resembles my own, however, their system was designed for only physical exercise. Interestingly they had used challenges, such as where one user could challenge a friend.

³<https://play.google.com/store/apps/details?id=com.strava&hl=en>

⁴<https://play.google.com/store/apps/details?id=com.garmin.android.apps.connectmobile>

The users of the challenge feature liked it. However, few challenges were sent, the authors think it could be due to the user's busy schedules. The authors derived the information from interviews. Regardless, this is supportive that challenges sent by friends could have potential.[20]

1.1.3 EcoIsland

EcoIsland is a research project that was created to help participants reduce their CO2 emissions. Some persuasive systems draw on social psychology like EcoIsland[21], albeit the work is a bit old and have some limitations. Their results indicate that social facilitation and conforming behaviour can have an impact in persuasive systems.

1.1.4 BinCam

Perhaps of particular interest is BinCam[22] a persuasive system that attempts to engage and motivate its testers into recycling more and better. This work highlights some crucial points with persuasive systems such as

1.1.5 Goal Tracking

As one of the main features implemented in the application is goal tracking and logging, some related works were investigated. F.ex[23] investigated how to support users in their goals to make them both challenging and achievable as this is what literature on goal accomplishment states is a necessity for them to be completed.

1.1.6 Papers

The paper "RightOnTime: The Role of Timing and Unobtrusiveness"[24] Researched if sending reminders were less intrusive at random times or by a time set by the user. They found a correlation that the time set by the user was less obtrusive. However, they did not have statistical significance for their hypotheses.

A paper named "More than Sex: The Role of Femininity and Masculinity in the Design of Personalized Persuasive Games" investigated how gender identity affected some persuasive techniques. This differed from other works by investigating Persuasive Technology by looking into Gender Identity as this study does.[25] Masculinity and Femininity are keywords here. Feminine people are more susceptible to customization, personalization, praise, reward, self-monitoring, simulation, suggestion and competition. Competition and comparison are more persuasive to men as opposed to females.

Another paper "Persuasive Patterns in Q&A Social Networks" [26] investigated how Stack Overflow could be interpreted as a persuasive application. The paper does provide a good explanation of the history of models used in persuasive technology. The article has a list of the persuasive technology techniques specified in the PSD model. It evaluates the Primary Task Support, Dialog Support, and Social Support features found in Stack Overflow. The study is salami sliced stating a future study will evaluate the Credibility Support category of the PSD model. The paper is an evaluation of the persuasive features used in Stack Overflow, but it does not use test or evaluates which users respond well to which persuasive strategies.

The paper "Investigation of Social Predictors of Competitive Behaviour in Persuasive Technology" [27] validates that reward is more effective for males as opposed to females. Figures out some relationships between persuasive strategies, such as competition is predicted by reward and social comparison. Social learning, on the other hand, is not predicted by reward nor social comparison. Some of the limitations are that it was limited to Canadians, and used a slightly less known model the Persuadability Inventory Model.

A paper "Personalized Persuasive Technology – Development and Validation of Scales for Measuring Persuadability" [28] presents another model for evaluating the relationship between users and persuasive techniques. However, it is limited to these persuasive techniques: "rewards, competition, social comparison, trustworthiness, simulation and social learning". The goal of this paper was to create a psychometric inventory for evaluating persuadability, where there was no existing inventory from before.

Then "Harnessing the Science of Persuasion"[5], is more of old work. The article has some citing issues it is old, uncommon style, finding the work cited work would take time. The article presents six persuasive techniques used in human-to-human persuasion. Such as it is not targeted at computer-mediated persuasion. Nonetheless, the techniques are Liking, Reciprocity, Social Proof, Consistency, Authority, and Scarcity. But these show some similarities to the techniques presented by Oinas-Kukonen et al.[4]

"Normative social influence in Persuasive Technology"[29] presents how normative social influence can affect how a person does on a goal. The paper investigates the comparing role, show himself how he does in relation to a goal, such as the number of phone calls to a specific person as opposed to another person. Another approach would be the mediating role, show friends how he does on a goal, thus having peer pressure. It is possible to persuade by comparing to other people to get a participant to do more. The paper did not prove this, but using social ties as a comparison to get the participant to do more and using a large number of people to get the participant to do more showed promise.

Finally the paper "Persuasive Backfiring: When Behavior Change Interventions Trigger Unintended Negative Outcomes"[2] presents what can happen if persuasion goes wrong, it focuses on the negative sides of behaviour intervention. Which they state may be ignored or overlooked in research endeavours. These works are also less frequently published, which is a problem as if it was published it could prevent other people from repeating the mistake.

2 Methodology

2.1 Literature Search

In the beginning, literature was found with the Web of Science. Of particular interest are behavioural change systems, and persuasive systems. These terms are somewhat overlapping but yield different works when searched. Not all literature found were relevant for my domain, such as e-commerce literature. The literature of interest works using systems persuading the user to change habits or behaviour in a positive way for the user or society. It is worth noting this work does not concern itself with persuasion to influence opinions of users in a social media context, examples such as election meddling through social media applications (SMA).

A conference called Persuasive contained many related works ¹. In the first conference paper, a work titled "Captology: A critical review"[30] summarized the field back in 2006.

For the technical aspects about Flutter and Dart articles from medium are sometimes used, this is the publication venue commonly used by those involved by the development of the UI-Toolkit Flutter and/or the programming language, Dart. In addition, conferences have provided decent information about flutter and dart. Most recently, the conference Google I/O 2019 provided many talks about Flutter, Dart and Firebase. Most notably there was the release of a technical preview for Flutter for the web. At Mobile World Congress 2019 (MWC19) flutter also made an appearance. In December at Flutter Live 2018, the framework transgressed released its first stable release. At DartConf 18 the framework first started gaining traction. Note that Flutter, Dart, and Firebase will be elaborated on later in this chapter.

Please note that some of the literature used may also be from works I found last semester and used in my papers. I decided to get a head start on my prototype last semester, I structured the work for the four courses I was taking to revolve around the creation of a social media application. The works can be found in this footnote ². Shortly summarized these revolved around the cross-platform framework Flutter, map-based challenges, a research project proposal, and a reward system. Most of the literature used is from the last two works.

2.2 Data collection

The independent variable for the experiment is the user's personality type. The dependent variables are the Persuasive Software Features' impact on Users' engagement, motivation, accomplishments, and activity level.

¹Persuasive Conference Papers: <https://link.springer.com/conference/persuasive>

²Previous Semesters Reports <https://drive.google.com/drive/u/0/folders/12tLTuw0J3Jo9Ky3v7e7-hF4ShMf18sbS>

2.3 Classifying Users

The goal of this thesis is to discover application features that facilitate goal accomplishment among users of different personality types. Therefore a model had to be chosen to have a valid ground for making a comparison.

Last semester I looked particularly into Bartle's Taxonomy of Player Types, however, this model has some shortcomings, so I performed a more in-depth review of potential models suitable for the project. I evaluated a few different models for classifying the users, the ones that were once considered for this project were: Bartle's Taxonomy of Player Types, the Myers-Briggs Type Indicator (MBTI), the Five Factor Model (FFM), and BrainHex. I had four criteria for choosing a model to classify the users.

1. The model's applicability in similar works.
2. Number of questions and time taken to get a result.
3. Difficulty of Submitting the result to the application.
4. Ease of using the result submitted in statistical analysis.

2.3.1 Bartle's taxonomy

Bartle's taxonomy classifies according to three different categories, Achiever, Explorer, Killer, and Socializer. In the last semester, I investigated Bartle's Taxonomy of Player Types. However, this model is not suited as it is based on Bartle's own experience with MUD games[12]. Additionally, literature has criticized it as not being properly testable and possibly containing flaws in the model. Emphasis on the fact that Bartle's Taxonomy is a qualitative model[31]. The model only requires the user to answer 30 questions with two options for the answer, where the questions are in the "would you rather" format (or a similar format). The model does not present the same questions each time it is taken (See <http://matthewbarr.co.uk/bartle/>). The result is given with a percentage of how your results match each category, summed up the categories total at 200%. The result can be generalized to a four-letter shorthand, such as "AESK", which would mean achiever has the highest result, and killer the lowest. Using the abbreviation it is easy to use in the application. Using the shorthand it should be easy to compare users based on the primary category they fall into. The advantages with the method are that if only the category with the highest result is used fewer testers will be needed to have a decent sample size.

2.3.2 Myers-Briggs Type Indicator

Please read criteria three first. Myers-Briggs is used in Yet the major downside with this method is that it is less accurate for people in with results close to 50% in the categories. In addition, one other shortcoming with the method is that people often get different results after retaking the test. The questions are presented in a 7-point Likert Scale from agreeing to disagree. The test consists of 54 questions (See <https://www.16personalities.com/free-personality-test>). Four categories with two possible values for each category. The first category extroversion or introversion, the second sensing or intuition, the third thinking or feeling, the fourth judging or perceiving. Each category is scored with a percentile anything below 50% gives the first value in the categories,

anything above gives the second. Thus there are 16 different combinations. 16 different categorizations provide difficulty, however, it should be possible to analyze individual categories in the method. However, the problem that the model turns a percentile scale into a binary value for each category does bring up potential problems with people who score quite close to the 50 percentile.

2.3.3 BrainHex

BrainHex consists of seven different classes, i.e., Achiever, Conqueror, Daredevil, Mastermind, Seeker, Socializer, and Survivor.[31] It has also been validated to be appropriate in The result is abbreviated as Main class-Subclass, f.ex Achiever-Conqueror. This would be relatively easy to analyze.

The user would have to be redirected to another site to take the quiz to get the result which is not optimal. The quiz consists of three pages of questions. The first page consists of 10 personal questions about the user including gaming habits. The second page has 21 Likert-scale questions. The last page consists of uniquely placing seven statements on a scale from 1–7. However it is mostly used in game contexts, and some of the questions may seem off-topic for the application.

2.3.4 Persuasive Potential Questionnaire

Another model is the Persuasive Potential Questionnaire, this model is a bit special as it seeks to unify the way of researching persuasive systems[32]. The study that presented is from 2016, however, it has not received much attention only 10 works have cited it (as of 08.03.19).

2.3.5 Five Factor Model

The five-factor model rates a user on five different factors as the name implies. These factors are Openness to Experience, Conscientiousness, Extraversion, Agreeableness, and Natural Reactions. During my investigation of the Five-Factor Model, I came across two appropriate variants. One with 44 questions, the original BFI-44 model[33, 34, 35]. Another shorter version, the BFI-10 model[36]. Initially, the later seemed more prominent however it is not recommended to use this model unless there is a severe time constraint in obtaining the results. The BFI-10 model is based on the BFI-44 model, and given its reduced number of questions, it is not as accurate, hence why it is not the de facto method.

To obtain the results, I designed a questionnaire widget to quickly receive the results from the testers. The widget only shows one question at a time, thus it requires no scrolling from the user, but only a single input to a Likert scale to progress. Back and forward actions are provided in the case of an input error.

The five-factor model is frequently used in later works in Persuasive technology. Normally the model uses 44 questions (BFI-44) to evaluate the user. However, this may be time-consuming. The BFI model has been used in recent works such as [37] they used the BFI-10, where they investigated both personality and mobility type as factors affecting which persuasion strategies were most effective for the user. The statistical analysis of BFI may be a bit harder than the BrainHex, but it appears to be more state-of-the-art and appears more suited for the context, as it doesn't ask game-related questions, but personality questions.

The final choice

The BFI-44 was deemed the most suited method for the In addition, it has been used by previous works in the field before.

2.4 Data processing

In a related paper [37] they used a methodology that was suitable for the BFI test. In the paper, they use a model called BFI-10 to quickly obtain a measure of the users' personality in just ten questions. The main difference is that they used data based on storyboards instead of an app test, and have a different set of persuasive strategies. In the paper, they use PLS-SEM to analyze the data. The paper explains what analysis methods they used before performing the analysis, Exploratory Factor Analysis (EFA), Kaiser-Meyer-Olkin (KMO), Principal Component Analysis (PCA)[37].

2.5 Plan A

Note, this plan happened. The main plan was to let users test the application over a period of time with their own devices. This has implications with NSD and GDPR (General Data Protection Regulation) as the application uses the cloud service Firebase. This could allow for the users to actually experience the persuasive features first hand.

After using the application the participants will be asked to answer questions about what features they preferred. To asses, the persuasive software features in-app questions could be administered to the users. In-app questions have advantages, such as no need to use another service, notifications to remind the user to answer the questions. The main downside is that it takes some time to implement the solution

2.5.1 User Recruitment

To test these users must be recruited to the application, which is a big risk. Consider this Plan A. However, if plan A should fail an alternative approach will be taken. Investigating users' preferences with Goal accomplishment applications, creating a conceptual application according to the findings from the user studies, and

The recruitment of users was done in multiple ways. Initially, I recruited users through my own social network, this proved efficient, however, it feels a little invasive. Also an attempt to recruit users I did not know personally Posts were put up on LinkedIn and Instagram, however, this form of user recruitment did not yield considerable results. Interestingly some people liked the posts but did not download the application. I also contacted some individuals who had expressed an interest in my app last semester, and people whom I had helped previously.

Additionally, I attempted to recruit users through a quiz-event where I was the quiz master. Pamphlets advertising the app were placed on the tables along with a few pieces of Twist confectionery. The confectionery was placed there as literature has shown that if you give somebody something they will be more likely to repay a favour. Unfortunately, in this setting, this did not seem to be the case, as the confectionery were more popular. A total of 2 users out of the approximately 50 people who were at the event registered in the application. It is possible that people did not perceive that

the confectionery and pamphlets were from me, due to the theme of the quiz covered "godteri, flesk og delikatesser" (translated to candy, pork, and delicacies).

2.5.2 Risks of Plan A

As this plan have a lot of risks related to personal data. A formal request was sent to NSD (Norwegian Center for Research Data) to get approval for the project. NTNU has an agreement with NSD that they are the responsible institution for approving projects that process personally identifiable information.

The main risks as stated previously is that it may not be approved by NSD and GDPR may be a problem. The workload of implementing plan A is massive. The users may lose interest in the app, and not answer any questions. User recruitment is costly. Responses given online may not be as accurate.

2.6 Plan B - The contingency plan

A note, plan B did not happen. Plan B consists of only using social media features with a test account. The testers will be handed a device allowing them to interact with the application until they feel they have used it enough. Afterwards, they will answer some questions anonymously in the app, the app will ask them about their age-range and their gender. According to a conversation with an NSD representative, this should be fine as long as the data can't be traced back to who answered what. Advantageously, I did not need to request written consent from the testers nor submit another application with NSD for this approach. To honour and ensure the anonymity of the questionnaire the results will be stored locally on one of my three phones that work, I will not look at the data until after I have sufficient testers. In addition, the questions answered will be placed at a random line in a CSV file, thus further anonymising the data by eliminating the order it was submitted. For plan B only local recruitment was applicable, the easiest way to recruit testers were to recruit testers from around campus, or to use my personal network to obtain testers.

2.7 Theory

There is a bunch of works that have made interesting discoveries of various factors contributing to the acceptance or rejection of persuasive software techniques. In one study[25], the authors looked into how sex, femininity, and masculinity affected persuasive software features. Let me emphasize that the authors investigated femininity and masculinity independently of the sex. They found that males found competition and comparison more persuasive, these strategies were rated lower by females.

However much of the literature warns that the domain of their studies may impact the validity of their results. One example domain from[25] were the promotion of physical activity through video game design. Then again other studies have studied domains such as sustainability (f.ex power usage, and recycling[22])

2.7.1 Storyboards or in-app-test

Something that was apparent with much of the literature in the field was that quite frequently storyboards were used instead of an in-app-test. The advantages of doing so are that considerably less work is needed to gain results about the persuasive features, as development is a costly and lengthy endeavour. However, factors like the user experience of using the features cannot be properly measured with storyboards as they are evaluating a concept in each storyboard with users opinions and not their practical experience. Test persons with storyboards may think they will like a concept, but they may have a different opinion if they actually tested it out in an application. Thus in-app-testing was decided on as a way to test the persuasive techniques, even though the development workload were quite high.

The existing literature on storyboards gave a good impression on what was interesting to implement in the application, in particular interest were the articles by Rita Orji which gave insight to what would be the best strategy for developing a one-size-fits-all solution.

2.7.2 Socially Oriented - Persuasive Strategies

Of particular interest is the strategies, Competition, Comparison and Cooperation. These strategies work well in social settings and worked well to implement in the application. These strategies have both strengths and weaknesses, which may come into focus depending on how they are implemented, as study[7] highlighted.

In particular, competition may make something more interesting and give a feeling of accomplishment. However, it may backfire and affect everything from self-esteem, confidence, privacy, motivation and more[7], which is why it is important to implement such features in a way that minimizes the potential negative consequences. One key point from[7] is to turn competition into self-competition, such as the user competes against himself. This avoids the issues of a flawed competition where one participant have an obvious advantage which may demotivate or negatively affect another participant. Otherwise multiple winning criteria should be specified to avoid a single winner, but instead designs for a more nuanced competition, with multiple winners.

A paper by B.J. Fogg, a leading expert on the field, proposed a conceptual framework for how to implement persuasion into the software. He argues that motivation, effort, and triggers have a major effect on how effective the persuasion will be.[38]

2.8 Hypotheses and Data Logging

The works using the persuasive systems design model have found that certain software features may not appeal to everyone. Some works recommend a one size fits all solution for persuasive systems design. Particularity systems consisting of utilizing personalization, goal setting, suggestions. These should be accompanied by strategies such as competition, comparison and rewards to make the application more motivating.[6] However not all of these fit into my application, in particular, personalisation is not implemented in the application, the app offers user customization instead as this was easier to implement. There were also time constraints. Unfortunately, if the findings in[6] are applicable this may indicate that some users high in "openness to experience" may dislike

this feature. The paper only evaluated 10 strategies in this paper, so some of the strategies I have implemented is outside what this paper tested.

The storyboards technique the papers used seem to avoid certain issues. For example, they ask for the users' opinions towards a persuasive feature they have just have been exposed to, thus how do storyboards account for the users' actual willingness using the software. What a person thinks about an abstract idea may be completely different than how the person behaves around an actual implementation. Though many factors play a part in an implementation; Is the usability good, do the users like the feature, and is it working as intended? Of particular interest is when an intervention backfires, and have the opposite effect than what it was designed to do.[2] Take the case of a

In addition to the hypotheses below, feedback will be accepted from the users. This could be a good source to find flaws in the design of the application.

2.8.1 User Retention

Hypotheses

- Do users who continue to use the app differ in some personality traits than those who do not continue using it.
- Do users who have high openness to experience stop using the application faster than others?

The motivation here is to have something to discuss if the user retention rate is low in the application.

2.8.2 Posts and goals

Hypotheses

- Do seeing posts from friends trigger upwards or downwards social comparison?
- Are some users with different personality traits more likely to progress in their goals?
- Will some users prefer to post about their goal instead of logging data towards it?

Part of the reason behind making the data-logging private is that the users cant compare themselves to others as easily. They only get an indicator of how their friends are doing on their goals by the progress indicator on the post. The reasoning behind this implementation is that the literature states that self-comparison and self-competition are better than comparison and competition with friends. Since tension can arise in the friendship, and people may be subject to negative social comparison or positive social comparison.[7] Do note that both positive and negative social comparison can be bad for a person experiencing it. Imagine a person named Jack, he sees that he had been drinking less than a friend, who is a lot bigger than Jack. Jack makes a positive social comparison thinking he is not doing so bad since his friend drinks more than him. Jack than feels this level of drinking is an acceptable level and reduces his effort in trying to drink less.

This will be assessed through a mix of in-app questionnaires and analysis of Firestore data.

2.8.3 User-set Reminders

Hypotheses

- Do users who set reminders become more likely to log data in the application, than those who do not?
- Do users who set reminders and those that do not differ in personality traits?
- Do users who set a reminder and do not log data cancel their reminders?

This will be assessed by sampling the database for the number of users who have set reminders for some of their active goals and through questions.

2.8.4 Privacy

Hypotheses

- Are users who are more concerned with their privacy less likely to publicly engage in the app, but more likely to have private goals?
- Does the privacy level in the application suit the users?

These hypotheses will be assessed through questions administered in the app about privacy.

2.9 Flutter and Dart

Before introducing the prototype it is essential to introduce both Flutter and Dart. A common problem with native mobile development is that usually both an iOS application and Android application must be developed. This can be a costly and time-consuming process.

The application is made in Flutter a cross-platform framework, that allows the developer to have a single codebase for their front-end. Flutter's programming language is Dart and it is often referred to as a UI-toolkit. The thesis will use the framework and UI-toolkit interchangeably.

The interesting thing about flutter is how it handles creating the UI. Flutter avoids many problems other cross-platform frameworks have. Other cross-platform frameworks often expose an API that contains methods for creating native widgets, but this is prone to some issues. Among those issues are the deprecation and slow updates, i.e. when the native environments updates can take a long time before the changes are reflected in the API. Instead of relying on the native components for creating the UI, flutter uses its own rendering engine based on Skia. Skia is the rendering engine used in the browser Google Chrome.

It should be noted that Flutter allows you to access native views if required, but it is not recommended since it impacts performance negatively. There are a few cases where Flutter may require a native view, this is done through *view inlining*. One example is accessing google maps this is done through view inlining.

In flutter there are two types of widgets, *StatelessWidgets* and *StatefulWidgets*. The layout is defined by using functional composition inside the build method found in both of these widget types. Instead of using an imperative style for defining UI, Flutter uses a declarative programming style to define the UI.

The framework has a layered architecture as shown in Figure 1.

```

1 void main() => runApp(MyApp());
2
3 class MyApp extends StatelessWidget {
4   @override
5   Widget build(BuildContext context) {
6     return MaterialApp(
7       title: 'Widget Example',
8       theme: ThemeData(
9         primarySwatch: Colors.blue,
10      ),
11      home: BasicWidgetExample("Hello World!", title: 'Minimal Material App'),
12    );
13  }
14 }

```

Listing 2.1: A minimalistic example of how to create and run a flutter application with dart. *BasicWidgetExample* is a stateless widget defined in listing 2.2

Flutter can be both Ahead of Time (AoT) compiled and Just in Time (JiT) compiled, which is beneficial for both the development and release of an application. When a flutter app is JiT compiled it runs in a Dart virtual machine (VM). This provides benefits when developing Flutter apps. For example, Hot Reload updates the app with new code changes with the state preserved within seconds. If new objects are introduced into the code this won't work properly. When this happens a Hot Restart can be performed instead, this action is also quite quick and is often done well within a few seconds. When the application is AoT compiled the compilers compile the app without dynamic type checking, and does not run it in the Dart VM anymore. In addition, Flutter is quickly being developed to bring support for both web and various desktop environments.[39]

As always it is good practice to separate UI from application logic, for this several architectural patterns are supported. Such as Flex, Redux or Bloc. The latter is the one used in the application and was a pattern introduced simultaneously with the framework. Flex and Redux are patterns frequently used in React applications, they have also been frequently used in Flutter. Packages to aid with using these patterns exist for all (see <https://pub.dartlang.org/packages/>).

2.9.1 Quirks and Features of Dart

Dart is a language with a lot of similarities to both Java, C#, and Javascript. makes writing layout in Flutter easier and less verbose. Some examples; Dart has made the "new" keyword optional. Dart supports both named and required parameters. Especially when designing layouts named parameters are frequently used. Empirically, this feels like less of a cognitive load as opposed to using only required parameters, and more can be inferred by only looking at the code. Dart allows trailing commas after parameters, which is convenient when creating layouts in Flutter.

In Dart 2.0 and up the datatype *double* no longer requires a decimal value a *int* works (ex: 3.45, 6, -3 are all good), and that is two characters saved. Plus in release 2.3 of Dart new syntax and

```

1 class BasicWidgetExample extends StatelessWidget {
2   final String requiredText;
3   final String title;
4
5   const BasicWidgetExample(this.requiredText,
6     {Key key, this.title = "No Title Set!"})
7     : super(key: key);
8
9   @override
10  Widget build(BuildContext context) {
11    return Scaffold(
12      appBar: AppBar(
13        title: Text(title),
14      ),
15      body: Center(
16        child: Column(
17          mainAxisAlignment: MainAxisAlignment.center,
18          children: <Widget>[
19            Text(requiredText),
20          ],
21        ),
22      ),
23      // Scaffold can have more parameters such as:
24      // floatingActionButton, drawer, and bottomNavigationBar
25    );
26  }
27 }

```

Listing 2.2: Shows how to create a minimal application with a scaffold showing a appBar with a title and a hello world message.

operators will make working with collections more easy[40]. Among these features is the spread operator which can be used in collections to unpack another collection in the collection where it is used.

Both Flutter and Dart are Open-Source projects by Google, and updates to the programming language Dart regularly happens with Flutter in mind.

2.10 Firebase

The prototype has been developed with Firebase's services such as Firebase Authentication, Cloud Firestore, Cloud Storage, Cloud Functions, and Firebase Cloud Messaging.

2.10.1 Firebase Authentication

Firebase Authentication provides OAuth 2.0 login to the application. In addition, it offers the possibility to use various sign-in providers as the login method. As of May 2019, it offers login with

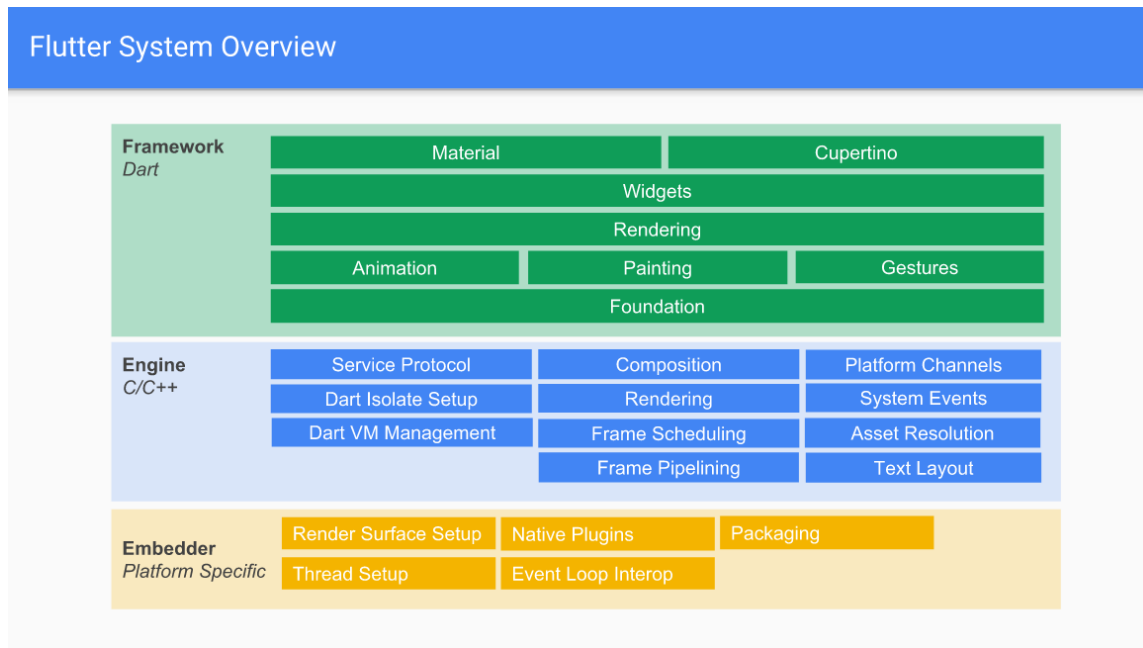


Figure 1: Flutter’s Layered Architecture. The image is [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/) licensed and obtained from Flutter.dev (<https://flutter.dev/docs/resources/technical-overview>)

Email/Password, Phone, Google, Play Games, Game Center (Beta), Facebook, Twitter, GitHub, Yahoo, Microsoft and Anonymous login. Not all of these were available at the start of the project, however, there is some work associated with adding support for each of these in the application. Thus only a subset was chosen, initially just Google and Facebook login. However, at the very start, when I was asking a few people to try the app they addressed concerns that they did not like to login with social media providers. Email/Password support was implemented in 8 hours the day thereafter.

2.10.2 Cloud Firestore

Cloud Firestore is a NoSQL database. Some advantages with this are that no infrastructure needs to be managed as opposed to applications with a SQL database. F.ex Firestore scales horizontally, adding more machines to cope with the size and demands of the database. This means that there is no downtime when scaling the database. Contrary to how it is with SQL databases, as these need to be transferred to bigger and more powerful machines as they grow, subjecting applications to downtime.

Data modelling in Firestore is however different from data modelling in SQL databases. Data denormalization is frowned upon in SQL databases, but in NoSQL databases, this is crucial to the performance to the database as it can reduce the number of queries needed to accomplish a task.

Firestore does impose a few limits on the documents in the database. Documents have a size

limit of 1MB, in addition, it can not have more fields than 40000 indexes[41]. More information can be found in the Firestore's documentation on usage and limits[42]. Firestore indexes everything in the documents to provide excellent read speeds, through a binary search algorithm. The write speed is a little bit slower, but reads are usually more frequent than writes.

Queries are quite easy. Do note that the syntax of a query varies slightly depending on the language, such as in if it is written in Dart for clients on Flutter, or in TypeScript for the Cloud Functions. Equality searches can get quite specific when creating a query, Listing 2.3 shows an example query used in the prototype to obtain the users' active reminders.

```

1  static Stream<QuerySnapshot> getReminderStream(String goalId) {
2    return collectionAccount()
3      .document(_account.accountId)
4      .collection("reminders")
5      .where("goalId", isEqualTo: goalId)
6      .where("canceled", isEqualTo: false)
7      .orderBy("timeToRemind", descending: true)
8      .snapshots();
9  }

```

Listing 2.3: A example query with Firestore, using equality searches on the client application with Dart.

However, there are some downsides with this approach, the queries can not have "and", "or" or "not equal to" operations in them when querying a single field. This is because queries must return a set of documents between a start index and an end index. It is not possible with the binary search algorithm used behind the scenes to return multiple sets of documents in one query. If this is needed, the results of multiple queries must be processed instead.

In addition, Firestore can create composite indexes if there is a need for more advanced queries. A composite index is may be needed for certain queries, in an example in[43] Todd Kerpelman specifies zip-codes with restaurant rating. This would allow users to find restaurants within a zip-code and with a rating using relational operators (<, <=, >=, >). Composite indexes are created when needed and automatically managed by Firestore[43]. The data in Firestore is secured by security rules. These defines who are allowed to access what in the database. An example of such rules is displayed in Listing 2.4.

2.10.3 Cloud Storage

As image resources are quite large these would be unsuited for storage in Cloud Firestore, as documents are limited to 1MB. Fortunately, Cloud Storage exists. Cloud Storage is a simple filesystem where various files can be stored, commonly used for images. Cloud Storage is secured in the same way as Cloud Firestore with security rules, so there is some relevance in Listing 2.4. The difference is the service it targets, and some syntax differences with how file paths are targeted contrasting to collections and documents in firestore.

```

1 service cloud.firestore {
2   match /databases/{database}/documents {
3     function signedIn(){
4       return request.auth.uid != null;
5     }
6     function resourceOwner(ownerId){
7       return request.auth.uid == ownerId;
8     }
9     function friendOfAccount(accountId){
10      return
11      ↪ exists(/databases/{database}/documents/account/{accountId}/friends/{request.auth.uid})
12    }
13    match /account/{accountId}/posts/{postId} {
14      allow read: if resourceOwner(accountId) || friendOfAccount(accountId) &&
15      ↪ resource.data.goal.public == true;
16      allow write, update, delete: if resourceOwner(accountId);
17
18      match /comments/{commentId}{
19        allow read, create: if friendOfAccount(accountId) ||
20        ↪ resourceOwner(accountId);
21        allow update, delete: if request.resource.data.account.accountId ==
22        ↪ request.auth.uid;
23      }
24    }
25  }
26 }

```

Listing 2.4: An example of Cloud Firestore rules. This is a limited example, the rules used in the prototype spans 144 lines.

2.10.4 Cloud Functions

Cloud functions are triggers that happen in response to a event. Some examples are when CRUD operations happen on documents, when a URL receives an https call, or when a new user is created. The benefit of using Cloud Functions is that you can have a serverless app. Cloud functions are written in either Node.js, Python, or Go.[\[44\]](#) For Node.js everything from Node.js 6,8 and 10 can be used. Node.js 6 is deprecated, and Node.js 10 is in beta. It should be noted during the project's lifespan Node 6 was the default environment, and Node 8 was in beta. Node 6 was chosen so this project have functions written in Node 6, and they have not yet been updated even though the environment was deprecated the 17.04.2019. As the deadline for updating them are the 22.04.2020 this was not prioritized.

Listing 2.5 shows a Node.js 6 function when a challenge is created in Firestore, an event triggers that executes the Cloud Function.

```
1 export function onNewChallenge() {
2   return functions
3     .region('europe-west1')
4     .firestore
5     .document('/challenge/{challengeId}')
6     .onCreate(async (snapshot, context) => {
7       const data = snapshot.data();
8       if (data !== undefined) {
9         const accountId = data.creatorId;
10        await rewardPoints(accountId, PointType.Creativity, 5, null);
11      }
12    })
13  }
14 }
```

Listing 2.5: An example of a Cloud function in Node.js 6

2.10.5 Firebase Cloud Messaging

Firebase Cloud Messaging is a service used to send push notifications to Android, iOS and web. In an application with Firebase Cloud Messaging enabled it is possible to send messages to the users a few different ways. One way is from the Firebase Console, a graphical UI for inspecting the different services' data and looking at analytics. Another is from the Cloud Functions when sending messages three protocols can be used. However, the HTTP v1 API is recommended as the other two are legacy protocols. The data of notification of this protocol is represented with JSON. It is identified with a name field, a data field, and have four different fields for specifying the content of the messages. The notification field is a field that can be used to get the notification for Android, iOS and Web, but it has fewer options, lowest common denominator. Then there are the android, apns and webpush fields which are used to target Android, iOS, and web respectively, with more customization.[45] Lastly, there are union fields (token, topic, and condition) which decides what the notification should target, such as a single device, users who are assigned to a topic, or some condition. In Flutter the messages are received through the Firebase Messaging plugin. The setup of this plugin requires the app to set up listeners for what should happen when a notification is clicked when the app is running, in the background and when the app is terminated.

2.11 The Prototype

The prototype is a continuation of some work I did last semester. The application that will be demonstrated has been published on Google's Play Store for Android devices and Apples App Store for iOS devices. An overview of the technologies and services used can be seen in Figure 2. The technology overview showcases the deployment targets, frontend framework and the external services. The architecture is serverless, with firebase's services managing the database and providing the possibility to run cloud functions on certain events, and more, as previously presented. The fig-

ure has been made with Flutter, the frontend framework. Thus the application could reach a bigger target audience.

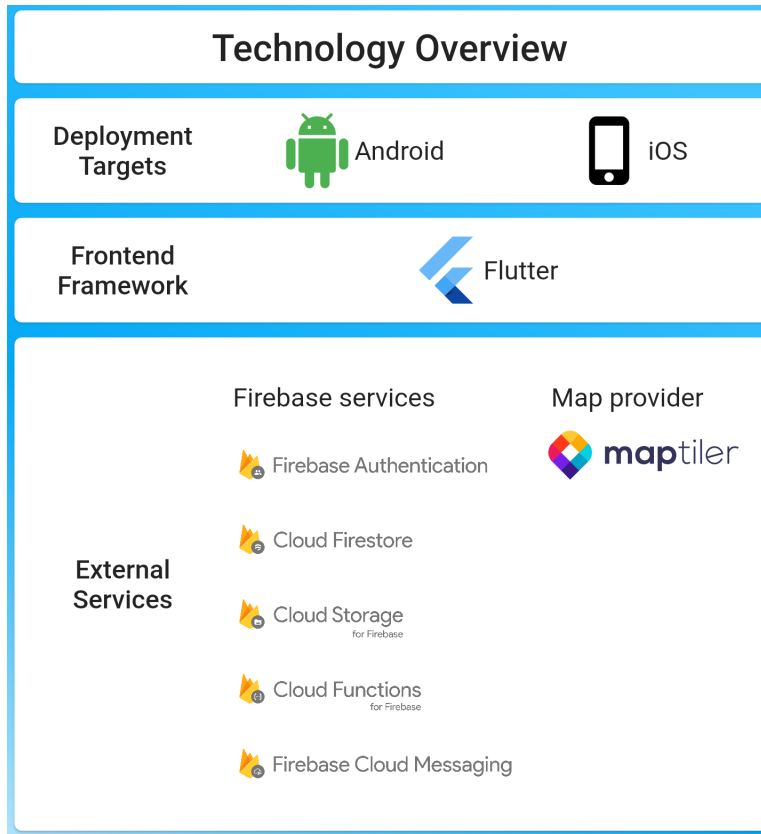


Figure 2: A technology overview.

The implementation is based upon theory from academic literature. There were works stating that social media interventions were deemed worth investigating further and some found positive results. Thus last semester the project that was started as a social media application for goal accomplishment.

The main routes in the prototype is a home feed, goal list, map, and profile page. Bottom navigation is used to navigate between the four main screens, this can be seen in multiple figures, such as Figure 3. The home feed show posts the friends of the users have posted to the application. The goal list shows the user what he or she is working on. The map is a suggestive feature that presents local challenges in the area if other users have submitted something. The profile page shows various stats, like how many goals the user has created and completed along with some points of three different types,

2.11.1 The Goal List

The goal list feature is where the user can manage his or her goals, this feature is shown in Figure 3. The main widgets used to create the screen is ListView, ExpansionTile and TabBar. The ListView is populated with the user’s data from Firestore. More precisely, the ListView listens for a Stream of QuerySnapshot, and the DocumentSnapshots contained within the QuerySnapshot are mapped into a Goal data class through deserialization, which is used to populate the ListView through a custom GoalTile widget. See Listing 2.6 for how this is done in Flutter and Dart. The screen shows the user’s goals, with the possibility to filter goals based upon state. If a post or log entry has been made towards a goal, it is defined as active, if not it is new. If the user has created a post stating it is completed or failed it gets categorised as those categories respectively.

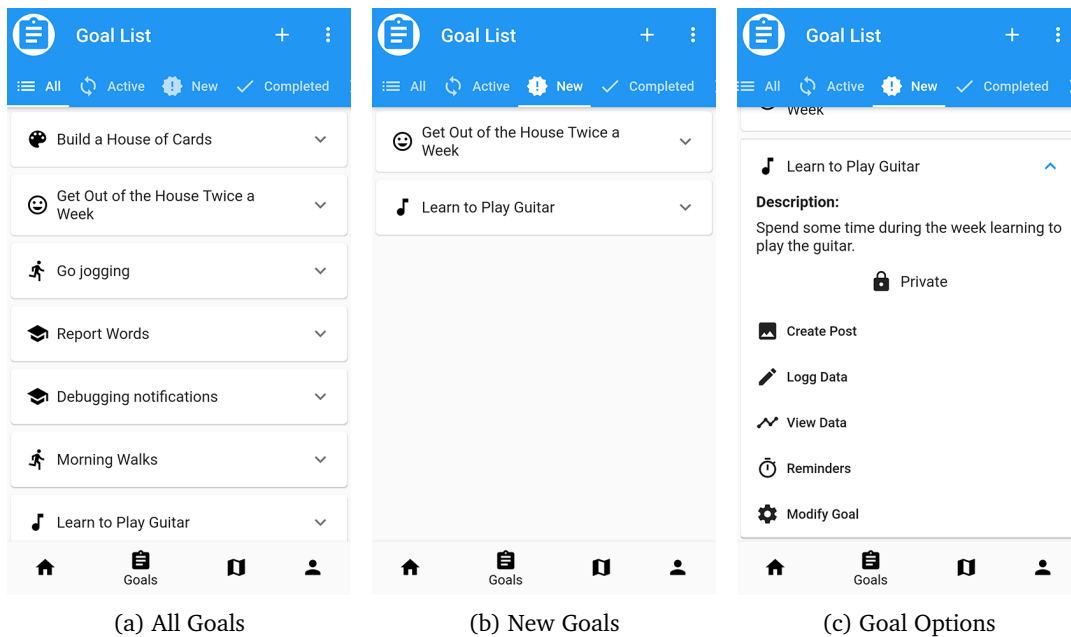


Figure 3: The goal list feature.

2.11.2 Login Process

When the application starts the app checks if a previous FirebaseAuth has signed-in to the app if so the authentication token is validated. If the token is still valid the user will see the app’s home page, otherwise, the user is redirected to the login. Likewise, if no previous FirebaseAuth was found. The user can sign in with either Facebook, Google or Email. When the login succeeds the user is redirected back to the authentication listener which then displays the home page, for as long as the authentication token is valid. This is demonstrated in Figure 8.

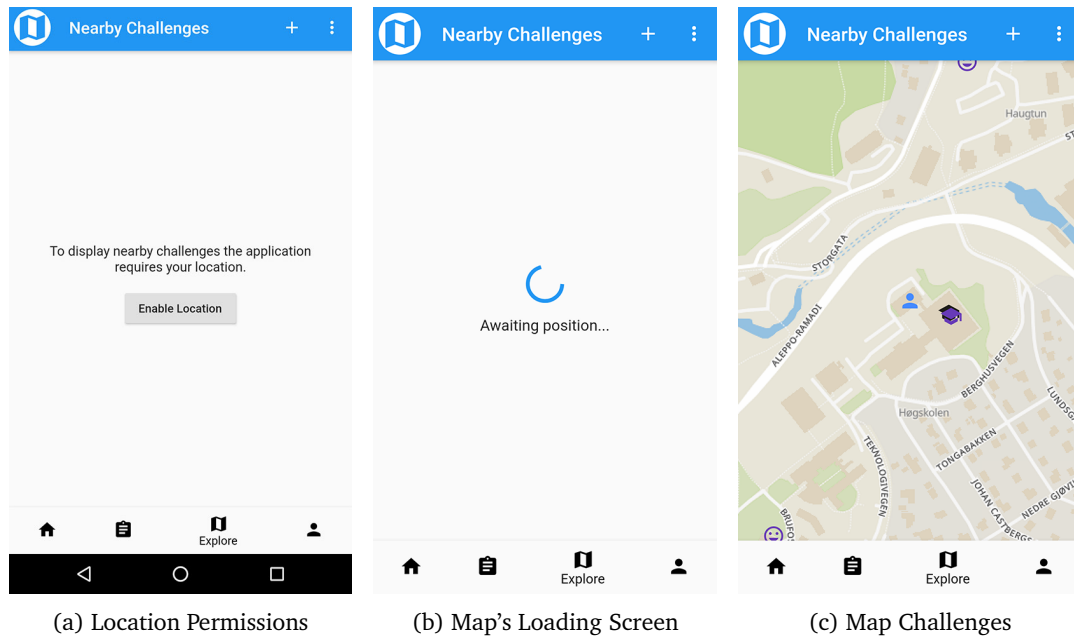


Figure 4: A map feature was included to provide a way to share challenges with the local community the app was used in. This feature was slightly inspired by Pokémon Go, where users can interact with their environment.

2.11.3 Sumarizing the application

In the application there are too many widgets possible to navigate to so I created an illustration that shows most of the possible routes, see Figure 9 for an application overview.

The project contains 199 dart files and 19373 lines of code. Even though Flutter allows projects to access native functionality this project have almost no native code. The exception is a small snippet that was needed to configure local notifications for the reminder functionality on Android and iOS. In addition to some configuration for Android and iOS. On Android the build.gradle files had to be modified. For example, to use firebase services the root gradle file needs the dependency classpath 'com.google.gms:google-services:4.2.0'. The app/build.gradle the file did specify how to use release builds.

Regardless, some of the dependencies in the project, the plugins, they do utilize native code. Which is why this project can avoid resorting to native code in the first place. A small example of plugins using native code is "Flutter Local Notifications", "Flutter Image Compress", "exif" and "Image Picker".

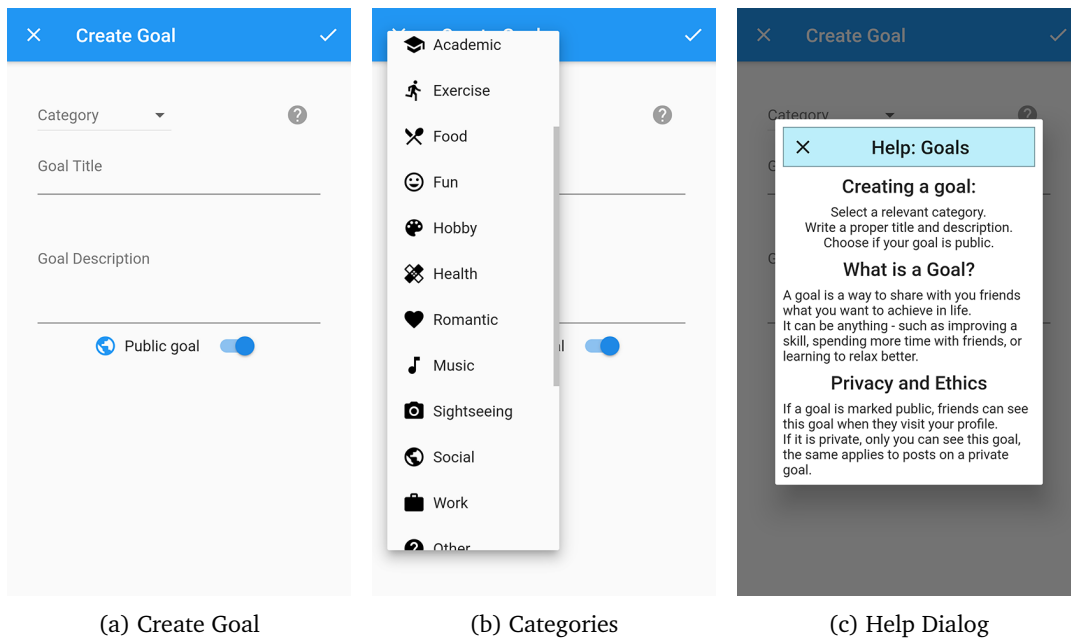


Figure 5: The goal creation screen in the app, as shown in 5a. 5b lists the categories the goal can be classified as; Academic, Exercise, Food, Fun, Hobby, Health, Romantic, Music, Sightseeing, Social, Work, and Other. 5c shows the help dialog that is available by pressing the help icon.

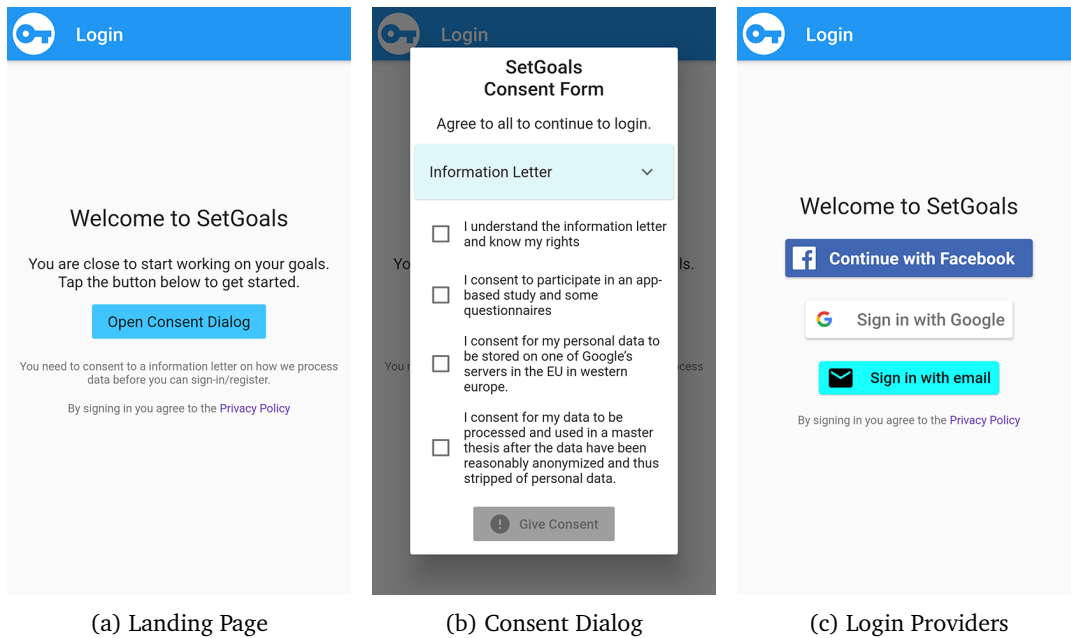


Figure 6: The login screens used in the application. Here 6a shows the landing screen introducing the users to the app. Secondly, after pressing the consent button they see the consent dialogue in 6b, presenting and getting consent on an information letter to the participants is a requirement by the NSD. Finally, 6c presents the login options the user. Do note that a slightly older version of these screens was used for some of the advertisements. Section 3.3.7 elaborates on the older versions.

```
1 @override
2 Widget build(BuildContext context) {
3   return StreamBuilder<QuerySnapshot>(
4     stream: stream,
5     builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
6       // ... (Omitted code for when snapshot is null or empty)
7       return ListView(
8         children: snapshot.data.documents.map((DocumentSnapshot document) {
9           print(document.data);
10          Goal goal = mySerializers.deserializeWith(Goal.serializer, document.data);
11          return GoalTile(
12            goal: goal,
13            showOptions: showOptions,
14            onTap: () => onTap(goal, document),
15            onLongTap: () => onLongTap(goal, document),
16          );
17        }).toList(),
18      );
19    },
20  );
21 }
```

Listing 2.6: Code for how the ListTile in the goal feed is populated. Slightly modified excerpt from: `/lib/widgets/util/view_goals.dart`

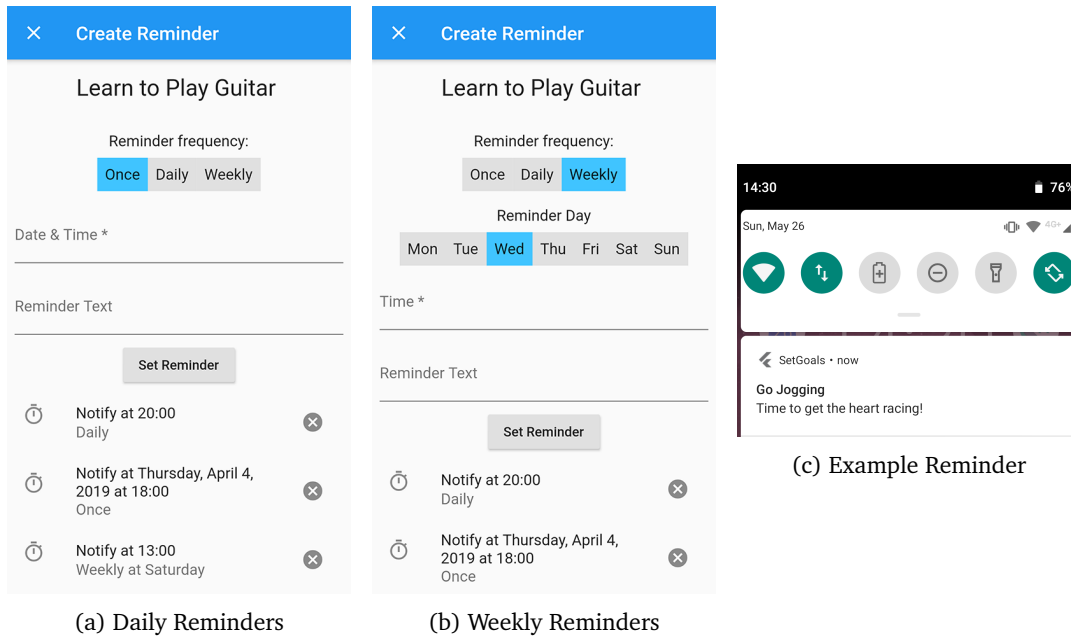
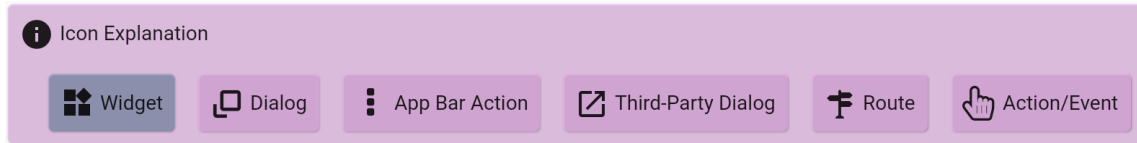
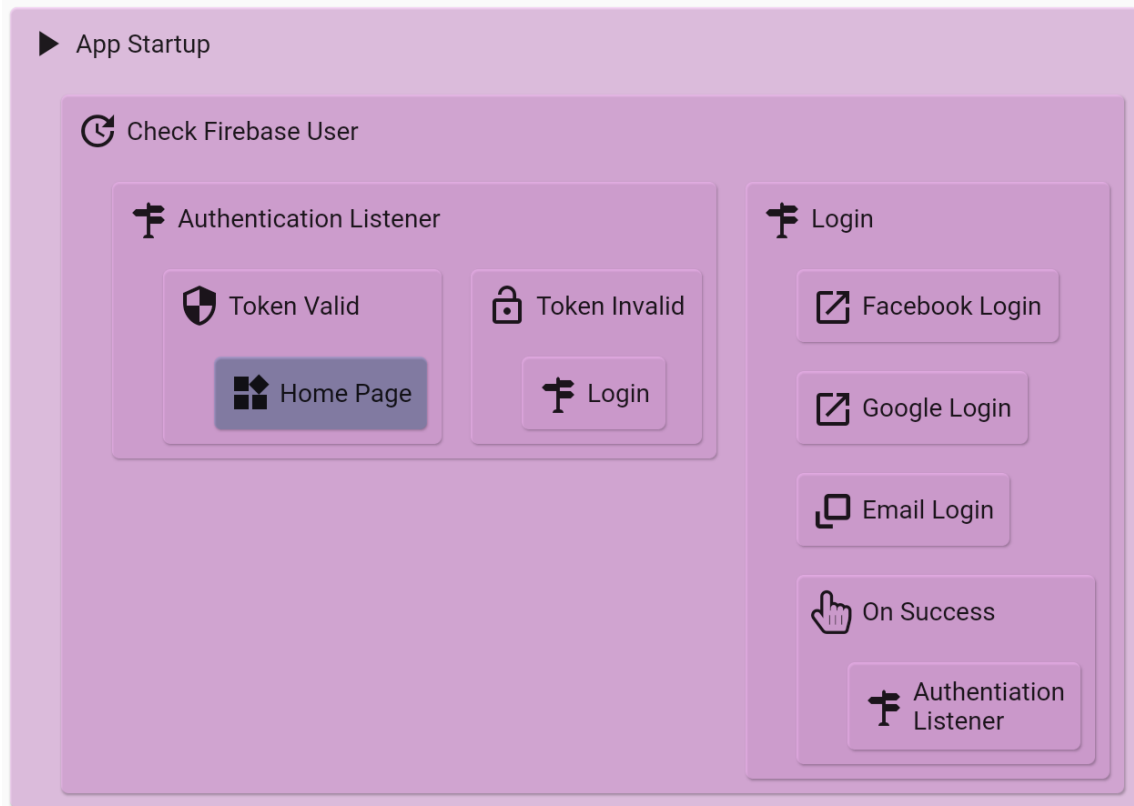


Figure 7: The reminder feature in the app. 7a illustrates how a daily reminder is set, the user have to specify the date and time. When the text field is tapped, a date-picker and time-picker sequentially appear. For 7b with weekly reminders, the date-picker is not used, instead, a custom widget where a weekday can be selected is used instead. The reminder notifications are handled by a plugin called Flutter Local Notifications, as the name implies these notifications are handled by the device.

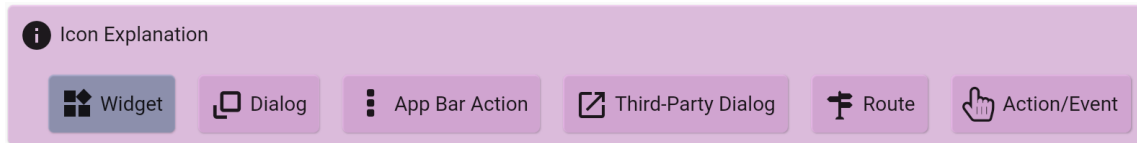


(a) Illustrates the meaning of the icons used in the sub-figures below. Note, a route is also a widget. A few other icons are used as well but these are mostly self-explanatory.

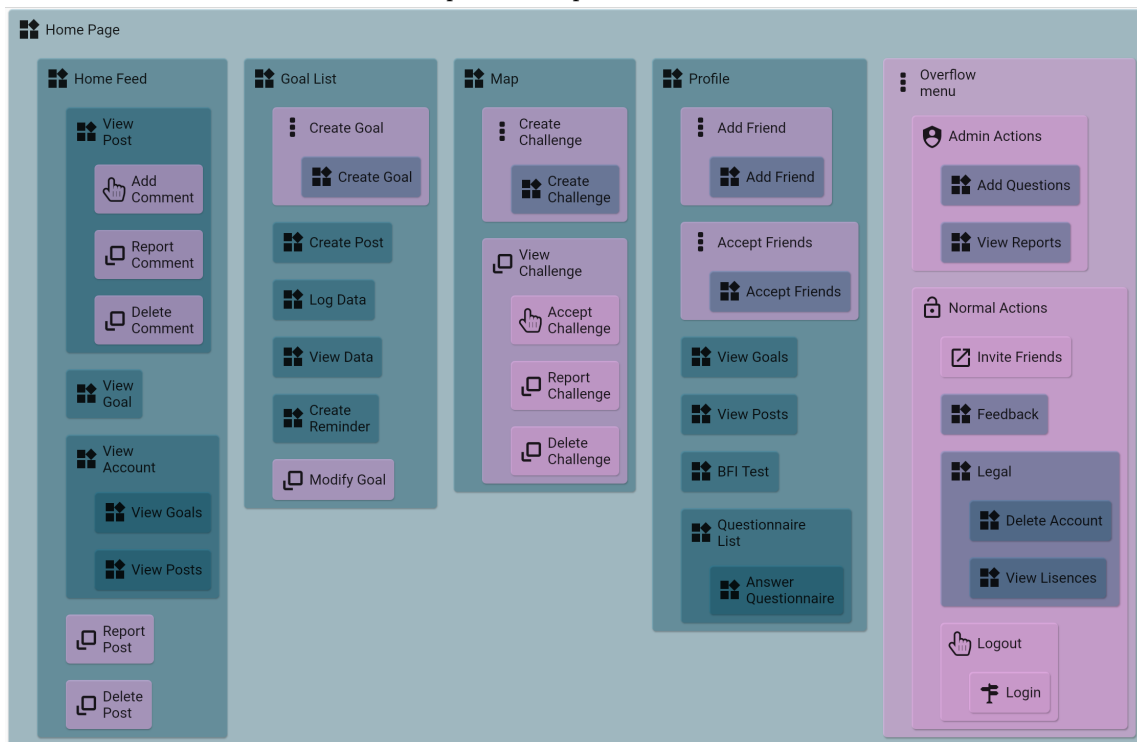


(b) An high-level view of the application's start-up and authentication procedure.

Figure 8: Application's Startup and authentication flow



(a) Icon explanation repeated for convenience.



(b) A high level overview of the different components/features of the application.

Figure 9: Application Overview

3 Results

The results will cover the user recruitment and advertising results through local recruitment, Google Ads and Facebook Ads. In addition, there is a section presenting the user feedback that was obtained through an in-app questionnaire. Later I present an analysis of the users' behaviours and the correlation found between their actions in the app. There will also be some mentioning of architecture in this section. Finally, there are some sections about the economic aspects of the app, optimization for mobile, and architecture.

3.1 Test Account

Before recruitment some test accounts was created, quite many in fact. This was done to ensure the protocol when a user wanted to delete his account worked, a lot of test accounts had to be registered and deleted. There exist a few test accounts in the Firestore database, but these test accounts have been removed from the dataset that is analysed. This includes my own account in the application.

3.2 User Recruitment from own Network

Initially, I recruited some users from my own network, through personal messaging, Linked-In, and Instagram. Approximately 16 users were recruited this way, and five users were recruited from some of these 16 users A few other recruitments happened in other situations as well. Two users registered during app review facebook and apple, one each, thus not real accounts There were a few cases when random users found the app when no advertisements were running, but this happened rarely.

3.3 User Recruitment from Advertisement

3.3.1 Jodel Advertised Post

First Jodel is a social media application that is anonymous and shows anonymous messages from people within approximately a 10 km radius. An advertised post was published on this platform for 99kr, which is shown in Figure 10. This was an inefficient way of advertising as only a few users registered, and the cost per user was quite high.

3.3.2 Google Ads

The application was advertised on Google Ads (Formerly: Google Adwords), a voucher was used to obtain 750 NOK in ad credits on the service. Four different campaigns were launched with various success, these are shown in Figure 11. Do note that the conversions from the iOS campaign are not shown in the panel since I had not set up conversion tracking from the iOS app properly, however,



Figure 10: The advertised post on Jodel (in Norwegian). Translated the post states: "In relation to my master thesis I am searching for participants for a mobile app. Do you want to try it out?". The post did get a score of 13 the sum of the upvotes and downvotes, but subtract two from the score due to self-promotion, so 11. However, only a couple of users installed it from this campaign.

Figure 15 gives some indication of the conversions, and Figure 12 Note that I will use the term conversions and this will be synonymous with app installs.

To begin with, a campaign with a worldwide audience was launched targeting Android devices, with a Cost-per-install (CPI) at 3.8 NOK. Later I launched an iOS campaign this also had a worldwide audience, at the same CPI. Unfortunately, this ran into problems with the availability of Firebase in China. I speculate that the Android campaigns explicitly did not target China, as Google Ads probably registered this. I attempted to create a second android campaign this time in Norway, but the impressions it made during two days ranged from 4-6 so this campaign was cancelled after two days. I assume this happened due to the low target CPI at 3.8 NOK. A second Android campaign was launched this time with the audience set to Europe. This time the CPI was set slightly higher at 6.5 NOK.

A third Android campaign was launched, the second worldwide Android advertisement. This was done to assess if the login screen had any effect on the number of registrations, the update is shown in Figure 16. There was a change in the login screen to make it more reputable with more official looking login buttons. However, this campaign ended up receiving a higher click-through rate and the number of conversions while it had made fewer impressions. I assume this was due to Google had been able to better target my app to users who would find this application interesting. Thus other factors may have played a role, so this was not analysed.

Campaign ↓	Conversions	Cost / conv.	Conv. rate	Cost	Clicks	Impr.	CTR	Avg. CPC
📱 iOS World-wide 29-30/04/19	0.00	NOK0.00	0.00%	NOK210.75	614	19,404	3.16%	NOK0.34
📱 Android World-wide 24-30/04/19	115.00	NOK3.27	9.70%	NOK376.52	1,186	67,277	1.76%	NOK0.32
📱 Android World-wide 23-26/05/19	158.00	NOK2.35	15.98%	NOK372.09	989	40,931	2.42%	NOK0.38
📱 Android Norway 05-06/05/19	0.00	NOK0.00	0.00%	NOK0.00	0	6	0.00%	—
📱 Android Europe 06-09/05/19	89.00	NOK4.53	11.14%	NOK402.96	799	64,526	1.24%	NOK0.50
Total: App campaigns ⓘ	362.00	NOK3.76	10.09%	NOK1,362.32	3,588	192,144	1.87%	NOK0.38

Figure 11: An overview of the different campaigns launched within Google Ads

3.3.3 Facebook Ads

Finally, a Facebook advert was launched the format of choice differed from what was used on Google. Facebook ads rely more on pictures, slideshows, and videos. So the format of choice for this ad was a video for 15 seconds. The ad was created with a template provided by Facebook. Interestingly, Facebook advises against images with a lot of text, as these may be shown less frequent or in the worst case not at all. This provided a small conundrum for my app since most of the app's screens are text heavy. The number of problems increased when screenshots of the application looked awful in the video template. The solution may be a bit odd as the images in the video highlight the phones the app is running on rather than just the app itself. The phones used in the ad is Motorola Moto X4, Nexus 5X, and iPhone SE. The ad only targets Android users, yet showcasing an iPhone in the ad may be a good thing to show that the app exists for multiple platforms. The ad created can be found here¹.

The Facebook ad reached a 5861 people, and made 6442 impressions, resulting in 14 conversions. The cost per conversion was 4.86 NOK, and a total of 67.97 NOK was spent. The reason not more was spent was due to facebook stopped displaying the ad, as they deemed the quality too low. Facebook does not make money if the ad is not clicked, as I am paying per click, not per impression. In terms of 3-second views the add received 1,137 of those, and the view percentage of the ad was 21.69%. The ad lasts 15 seconds.

3.3.4 Result of Advertising

The results of the European Android campaign is shown in Table 1. The results of the first world-wide Android campaign is shown in Table 2. The results of the second world-wide Android campaign is shown in Table 3.

¹Facebook Advert: <https://www.facebook.com/watch/?v=2206818989395891>

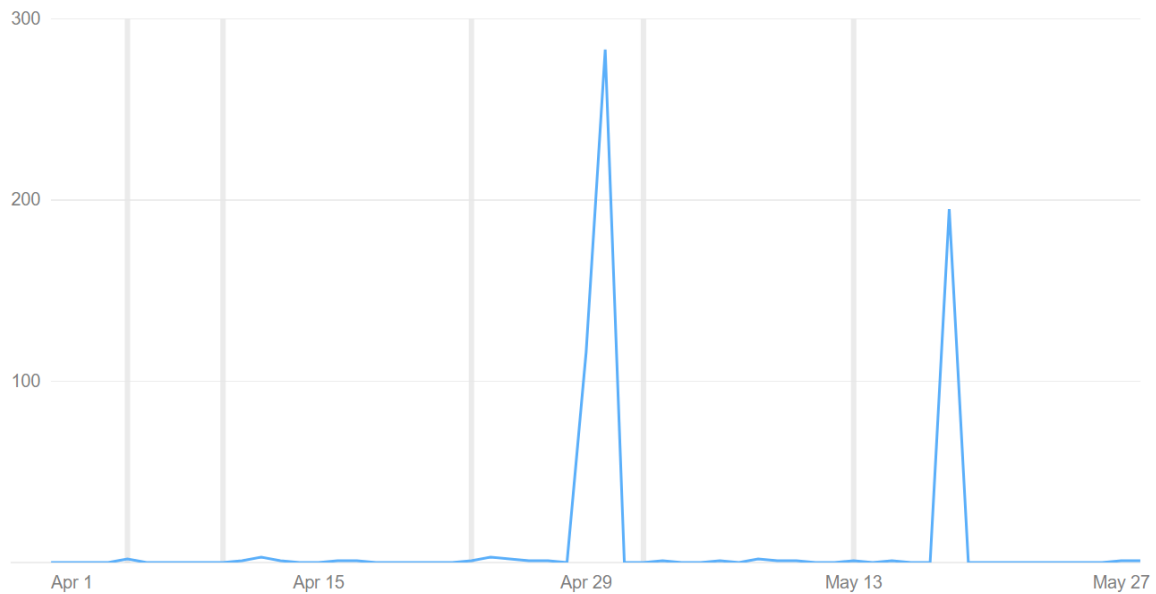


Figure 12: The number of installs on iOS. The first spike is the iOS advertisement from Google Ads which mostly targeted China. I do not know what the spike at the 18th of may is caused by, but it is 195 conversions events from the United States. Even though there are a lot of conversion events there were barely any installs from these dates. Apparently, Firebase is blocked in China. The gray lines show the dates an update was released.

Table 1: The European Android Campaign. Countries where at least one conversion event happened. Sorted on conversions, high to low. Do note a column have been added to the table with USD per conversion, this is calculated from the NOK value with the 90-day average exchange rate from xe.com at the 20.05.19, 1 NOK = 0.11604 USD. Abbreviations: Conv = Conversion(s), Impr = Impressions, CTR = "Click-through rate", CVR = "Conversion rate", AvgCPC = "Average cost-per-click".

Locations	Conv	Clicks	Impr	CTR (%)	CVR (%)	AvgCPC (NOK)	Cost (NOK)	NOK/Conv	USD/Conv
Ukraine	16	79	6211	1.27	20.25	1	78.98	4.94	0.57
Bulgaria	11	40	2540	1.57	27.50	0.9	35.85	3.26	0.38
Romania	9	47	4172	1.13	19.15	0.73	34.09	3.79	0.44
Kosovo	9	115	7961	1.44	7.83	0.28	31.69	3.52	0.41
Serbia	8	157	11254	1.4	5.10	0.26	41.28	5.16	0.6
Albania	6	52	5502	0.95	11.54	0.45	23.42	3.9	0.45
Macedonia (FYROM)	5	105	8678	1.21	4.76	0.3	31.73	6.35	0.74
Belarus	4	18	1864	0.97	22.22	0.87	15.62	3.9	0.45
Bosnia and Herzegovina	4	68	5033	1.35	5.88	0.27	18.24	4.56	0.53
Poland	3	8	413	1.94	37.50	0.82	6.59	2.2	0.26
Portugal	3	13	1252	1.04	23.08	0.8	10.36	3.45	0.4
Slovenia	2	5	435	1.15	40.00	1.71	8.57	4.29	0.5
Greece	2	13	1588	0.82	15.38	0.57	7.41	3.71	0.43
Ukraine	1	5	207	2.42	20.00	0.25	1.23	1.23	0.14
Slovakia	1	1	198	0.51	100.00	1.46	1.46	1.46	0.17
Montenegro	1	23	2009	1.14	4.35	0.5	11.58	11.58	1.34
Hungary	1	7	486	1.44	14.29	1.51	10.55	10.55	1.22
Spain	1	6	280	2.14	16.67	1.1	6.57	6.57	0.76
Moldova	1	6	809	0.74	16.67	0.95	5.7	5.7	0.66
Lithuania	1	5	241	2.07	20.00	1.17	5.87	5.87	0.68

Table 2: First world-wide Android Campaign. Countries where at least one conversion event happened. Sorted on conversions, high to low. Do note a column have been added to the table with USD per conversion, this is calculated from the NOK value with the 90-day average exchange rate from xe.com at the 20.05.19, 1 NOK = 0.11604 USD. Abbreviations: Conv = Conversion(s), Impr = Impressions, CTR = "Click-through rate", CVR = "Conversion rate", AvgCPC = "Average cost-per-click".

Locations	Conv	Clicks	Impr	CTR (%)	CVR (%)	AvgCPC (NOK)	Cost (NOK)	NOK/Conv	USD/Conv
India	20	150	6409	2.34	13.33	0.44	66.27	3.31	0.38
Iran	9	94	5153	1.82	9.68	0.36	33.92	3.76	0.44
Algeria	6	103	6665	1.55	5.83	0.21	21.43	3.57	0.41
Egypt	6	80	5299	1.51	7.50	0.26	20.42	3.4	0.39
Bangladesh	5	29	921	3.15	17.24	0.56	16.37	3.27	0.38
Dominican Republic	5	20	1039	1.92	25.00	0.32	6.5	1.3	0.15
Morocco	5	70	4069	1.72	7.14	0.35	24.31	4.86	0.56
Haiti	4	24	894	2.68	16.67	0.26	6.29	1.57	0.18
Zambia	4	12	343	3.5	33.33	0.48	5.76	1.44	0.17
Senegal	3	35	1050	3.33	8.57	0.23	7.93	2.64	0.31
Libya	3	18	855	2.11	16.67	0.17	3.11	1.04	0.12
Mali	3	15	314	4.78	20.00	0.31	4.6	1.53	0.18
Cameroon	2	13	455	2.86	15.38	0.49	6.37	3.18	0.37
Sri Lanka	2	7	166	4.22	28.57	0.22	1.54	0.77	0.09
Benin	2	5	388	1.29	40.00	0.13	0.67	0.33	0.04
Bolivia	2	14	697	2.01	14.29	0.28	3.91	1.95	0.23
Lebanon	2	21	1702	1.23	9.52	0.26	5.38	2.69	0.31
Yemen	2	11	400	2.75	18.18	0.35	3.81	1.91	0.22
Colombia	2	8	442	1.81	25.00	0.58	4.61	2.3	0.27
Tanzania	2	10	808	1.24	20.00	0.43	4.27	2.14	0.25
Iraq	2	56	4410	1.27	3.57	0.2	11.41	5.71	0.66
Albania	2	6	134	4.48	33.33	0.45	2.69	1.35	0.16
Togo	2	8	249	3.21	25.00	0.49	3.92	1.96	0.23
Belize	1	1	75	1.33	100.00	0.29	0.29	0.29	0.03
Uruguay	1	1	36	2.78	100.00	0.74	0.74	0.74	0.09
Myanmar (Burma)	1	3	73	4.11	33.33	0.48	1.44	1.44	0.17
The Gambia	1	5	161	3.11	20.00	0.14	0.71	0.71	0.08
Democratic Republic of the Congo	1	5	163	3.07	20.00	0.79	3.93	3.93	0.46
Paraguay	1	5	222	2.25	20.00	0.48	2.42	2.42	0.28
Sudan	1	8	287	2.79	12.50	0.39	3.1	3.1	0.36
Burundi	1	4	34	11.76	25.00	0.1	0.39	0.39	0.05
Cambodia	1	10	129	7.75	10.00	0.26	2.62	2.62	0.3
Burkina Faso	1	8	193	4.15	12.50	0.21	1.7	1.7	0.2
Rwanda	1	1	111	0.9	100.00	1.37	1.37	1.37	0.16
Tunisia	1	20	1367	1.46	5.00	0.3	5.92	5.92	0.69
Pakistan	1	56	5545	1.01	1.79	0.32	17.88	17.88	2.07
Cuba	1	4	139	2.88	25.00	0.29	1.16	1.16	0.13
Niger	1	2	156	1.28	50.00	0.19	0.38	0.38	0.04
Guinea	1	8	227	3.52	12.50	0.24	1.91	1.91	0.22
Peru	1	11	274	4.01	9.09	0.29	3.16	3.16	0.37
Brazil	1	15	489	3.07	6.67	0.43	6.45	6.45	0.75
Ecuador	1	13	506	2.57	7.69	0.18	2.28	2.28	0.26

Table 3: Second world-wide Android Campaign. Countries where at least one conversion event happened. Sorted on conversions, high to low. Do note a column have been added to the table with USD per conversion, this is calculated from the NOK value with the 90-day average exchange rate from xe.com at the 20.05.19, 1 NOK = 0.11604 USD. Abbreviations: Conv = Conversion(s), Impr = Impressions, CTR = "Click-through rate", CVR = "Conversion rate", AvgCPC = "Average cost-per-click".

Locations	Conv	Clicks	Impr	CTR (%)	CVR (%)	AvgCPC (NOK)	Cost (NOK)	NOK/Conv	USD/Conv
India	58	252	9605	2.62	23.20	0.53	134.67	2.31	0.27
Iraq	11	81	3660	2.21	13.58	0.23	18.36	1.67	0.19
Algeria	9	60	2855	2.1	15.00	0.26	15.57	1.73	0.2
Bangladesh	7	42	1110	3.78	16.67	0.42	17.77	2.54	0.29
Iran	7	36	1412	2.55	19.44	0.36	12.82	1.83	0.21
Colombia	5	11	356	3.09	45.45	0.91	10.05	2.01	0.23
Pakistan	4	55	3672	1.5	7.27	0.29	15.93	3.98	0.46
Ecuador	4	7	215	3.26	57.14	0.72	5.01	1.25	0.15
Bolivia	3	7	178	3.93	42.86	0.77	5.42	1.81	0.21
Syria	3	7	246	2.85	42.86	0.37	2.58	0.86	0.1
Morocco	3	48	1548	3.1	6.25	0.2	9.42	3.14	0.36
Egypt	3	72	2364	3.05	4.17	0.22	15.89	5.3	0.62
Benin	2	6	143	4.2	33.33	0.46	2.76	1.38	0.16
Ethiopia	2	10	198	5.05	20.00	0.24	2.43	1.22	0.14
Chile	2	3	81	3.7	66.67	1.24	3.73	1.86	0.22
Lebanon	2	5	429	1.17	40.00	0.42	2.1	1.05	0.12
Mongolia	2	4	80	5	50.00	0.35	1.4	0.7	0.08
Sierra Leone	2	4	151	2.65	50.00	0.32	1.3	0.65	0.08
Nepal	2	8	303	2.64	25.00	0.31	2.45	1.22	0.14
Niger	1	5	72	6.94	20.00	0.3	1.49	1.49	0.17
Macedonia (FYROM)	1	1	111	0.9	100.00	0.21	0.21	0.21	0.02
Angola	1	1	73	1.37	100.00	0.51	0.51	0.51	0.06
Moldova	1	1	4	25	100.00	0.3	0.3	0.3	0.03
Myanmar (Burma)	1	4	87	4.6	25.00	0.32	1.28	1.28	0.15
Jordan	1	10	388	2.58	10.00	0.1	1.04	1.04	0.12
Belarus	1	4	13	30.77	25.00	0.42	1.69	1.69	0.2
Nigeria	1	5	287	1.74	20.00	0.81	4.04	4.04	0.47
Kyrgyzstan	1	2	44	4.55	50.00	0.24	0.48	0.48	0.06
Sudan	1	14	342	4.09	7.14	0.44	6.1	6.1	0.71
Jamaica	1	2	71	2.82	50.00	0.71	1.41	1.41	0.16
Tunisia	1	13	642	2.02	7.69	0.24	3.06	3.06	0.36
Laos	1	2	53	3.77	50.00	0.44	0.88	0.88	0.1
Cuba	1	1	69	1.45	100.00	0.66	0.66	0.66	0.08
Ghana	1	4	233	1.72	25.00	0.48	1.92	1.92	0.22
Venezuela	1	8	253	3.16	12.50	0.15	1.2	1.2	0.14
Zambia	1	4	121	3.31	25.00	0.81	3.24	3.24	0.38
Guinea	1	2	94	2.13	50.00	0.29	0.58	0.58	0.07
Cote d'Ivoire	1	3	77	3.9	33.33	0.52	1.57	1.57	0.18
Papua New Guinea	1	1	22	4.55	100.00	0.44	0.44	0.44	0.05
Kenya	1	1	24	4.17	100.00	0.48	0.48	0.48	0.06
Mali	1	7	195	3.59	14.29	0.16	1.1	1.1	0.13
Afghanistan	1	6	164	3.66	16.67	0.51	3.04	3.04	0.35
Palestine	1	3	58	5.17	33.33	0.48	1.45	1.45	0.17
Brazil	1	12	588	2.04	8.33	0.59	7.09	7.09	0.82
Albania	1	2	73	2.74	50.00	0.4	0.79	0.79	0.09
Vietnam	1	1	52	1.92	100.00	1.2	1.2	1.2	0.14

3.3.5 Analytics

The project has access to quite a few analytics frameworks. There is one on Firebase, Facebook, Google Play Console, and one on Apple's App store

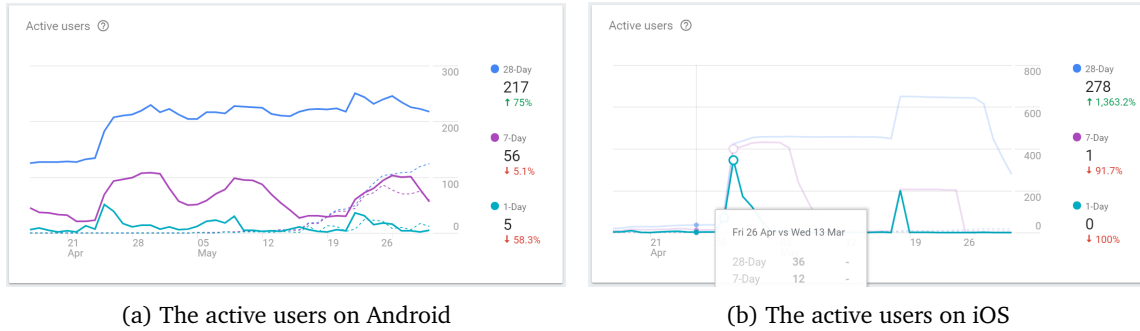


Figure 13: Shows the differences between the active users on Android and iOS. Note there was a spike of 201 iOS users on the 18. May, which does not make sense as no advertising campaign was held for iOS at this date.

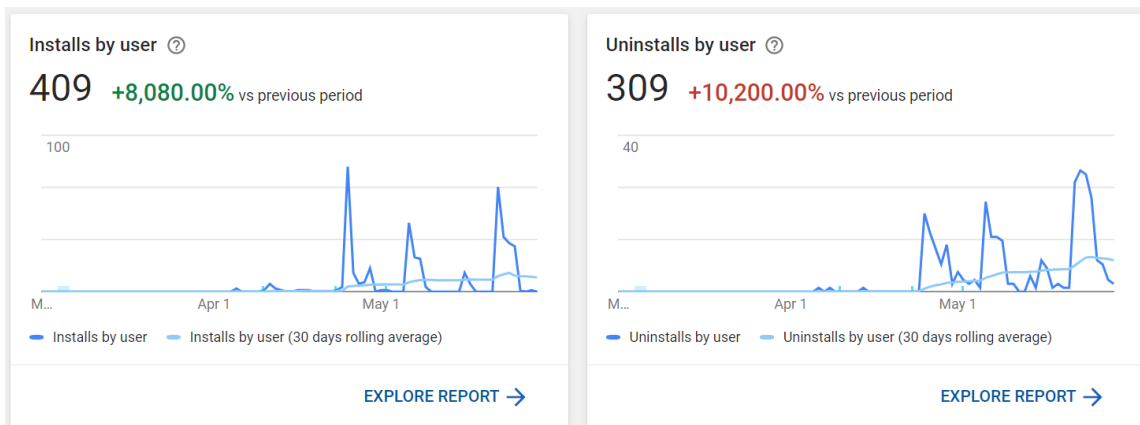


Figure 14: The number of installs and uninstalls for android from the Google Play Console. The Figure shows data from the 1st of March to the 29th of May.

3.3.6 App Installs and Deletes

The application received quite a lot of install. Unfortunately, the number of registrations was far lower. It is possible that the consent dialogue at the beginning affected the number of people who registered. As the process requires the users to check 4 checkboxes, this repels them from the application. This conundrum was a bit unfortunate as I was required by NSD to have a form like that before I could start collecting user data. Do note that the data from Apple is from opt-in users only. Which apple states is 33% of the users who installed SetGoals.

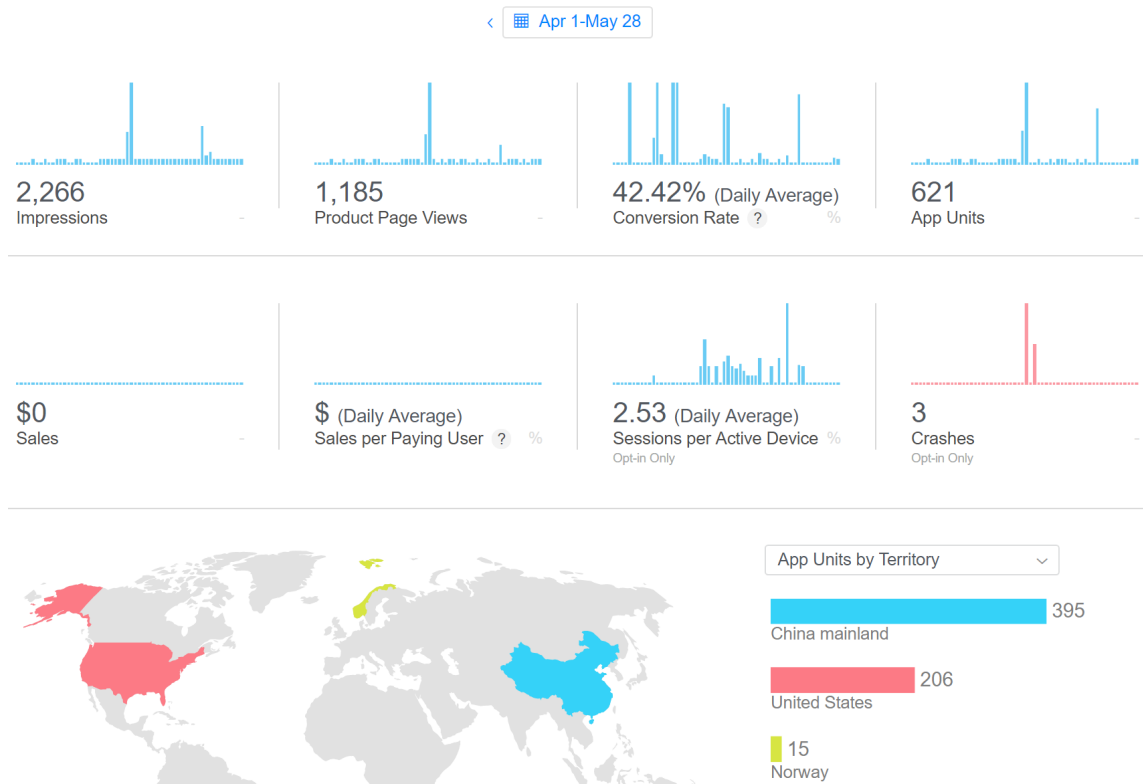


Figure 15: Overview of the analytics data from Apple’s App Store Connect

3.3.7 Differences in the advertisements

The login screens differed slightly in the advertisements that were launched on Google and Facebook. In essence, the difference was that the buttons looked unofficial in the previous version, and more official in the later, as seen in Figure 16.

3.4 User Feedback

3.4.1 Questionnaire

In total 20 questions were issued inside the application, out of these 7 users responded. The questions can be found in Appendix A. In short, they can be categorised as general questions, social questions, goal logging questions, map challenge questions, privacy questions, and a couple of feedback questions. Most questions were administered with a text field where the user could answer. In some cases, the questions used a five-point Likert scale (alone or in addition to the text field), and where deemed necessary a photo was attached to the question to clarify the question.

Most participants were in their mid-twenties. All the participants were users who were recruited by me or through a friend. 4 males and 3 females participated.

When asked "Did you feel the application's social features (posts and comments) were useful for your progress?" six users indicated no or maybe, with one outlier. Four users pointed out that they did not have enough friends in the application. Getting people to connect with friends appears to be an issue. Thus they did not get the desired effect out of it.

The question "Did seeing posts from friends make you want to compete or collaborate with them towards a goal?" received a similar response with four users pointing out the problem with friends, in addition to a one yes, maybe, and no.

When users received the question "Did you find the point system motivating, engaging, or competitive?", it received two positive responses, two unserious responses, one no, and two responses indicating they were inactive users of the application.

In regards to goal logging, the users were asked "What did you and did not like about the view data feature?", an image was shown with this question showing the check-ins. One user indicated a problem with being unable to log past activities that the user forgot to log. Two users did not know it existed, two barely used it, one user was neutral, and two users liked it.

When asked about "How did you feel about having to log the data on the same day you did your activity? (Not being able to add entries for previous days)" Apart from one user not being aware of the feature, six users either didn't like it or indicated they would prefer more control to log their goals. one user also indicated the possibility of cheating one by just adding multiple entries during a day. And another user stated users would probably not cheat themselves with more control.

For continued use users were asked, "Would you continue to use this application, why or why not?". This was answered through a Likert scale with two yes, three no and two maybe. However, the additional info from the text responses two responses emphasised the need for more friends or indicated a lack of content in the app, but these responses were positive to the app.

Over to the reminders, "How did you feel about receiving the reminders, did they help you to work towards your goal?". Two yes, two maybe, two have not used them, and one blank answer.

In regards to the map; "Did you use the map feature, why and why not?". Six users said they did not use the map feature, due to the lack of challenges nearby and the fact that the map cannot be zoomed out further. Two users were positive to the concept. It appears there may have been confusion around the location used in the app as well. Figure 17, shows the users opinion of the usefulness of the map

Do you think the map feature could be improved, if so do you have any suggestions? Two users stated they wanted to zoom out and more content and one wanted it to be more aesthetically pleasing. The rest were unsure.

"Which of these login methods are you comfortable using: Email, Facebook, Google, or other methods?" two users indicated other methods, five mentioned email, three mentioned Google, and one mentioned facebook. Note, multiple methods were provided in the text.

"Would you be comfortable with other people seeing your goal's logged data, if so who?" One indicated no one but themselves, two indicated people they have chosen themselves, and three indicated friends and one indicated friends of friends.

"Who are you be comfortable sharing your personal goals too? (not private goals)" This yielded

four for friends, one for family, one for friends of friends, and two it depends.

"Did you encounter any errors in the application?" Here the users encountered a few different problems. There was an error with users not receiving points when they should in the beginning, this was fixed. The questionnaire was bugged was one report, however, I think this is due to using an outdated version of the application, so the app incorrectly displayed the questions. One case was a problem with the friend requests, this may have been a network error, but I am unsure. Another report concerned searching for peoples names which were inconsistent. This was due to the name is collected from the social media providers or whatever the user types in the name field in the registration form for email accounts. That was problematic as user tended to only write their first names. In addition to queries to firestore requiring the name to be an exact match, this made the searching for friends extra difficult.

The biggest error was when users clicked notifications when the application started from a terminated state, which resulted in a red screen with an error message. In addition, a few users had a problem with running the app when they first installed it on Android 6.0 and earlier, due to an error in the location plugin. This was conveniently fixed in the plugin 6 hours before I discovered the code that had the blame.

The last question in the app was a way to submit more feedback if they wanted to, or in the feedback form in the application, and it also served as a reminder to consent to the prize draw rules if they wanted to participate in the prize draw. No extra feedback was submitted here, however.

3.4.2 In-app feedback

Some of the users mentioned it was hard to remember to log their goals.

One participant elaborated on how the application could be better. The suggestion was to make the application calendar based as opposed to the list based design it currently has. The core idea meant having a calendar where the user could click on days to see their goals, along with posts from friends and reminders. Sort of like a calendar application, but with social features and specific days for their goals.

In addition, there were reports when users did encounter something they thought was an error.

3.5 User Behaviour

3.5.1 Implemented Persuasive Strategies

I would argue that the prototype have implementations for the persuasive strategies shown in Table 4.

Modify the table descriptions to be more accurate, and reflect my applications implementation how it is included.

There seems to be a trend with the users recruited from the ads. Most of them did no actions in the application, I assume they just explored it and left it. Figure 18 shows the number of actions the users took, the values marked valid means the number of users who did add the content type specified in the columns. As the figure shows there were a lot of missing values, which shows the lack of engagement with users who just registered.

Table 4: Persuasive techniques present in the application.

Strategy Type	Persuasive Strategy	Description
Primary Task Support	Self-Monitoring	Lets the user monitor some condition himself, and input the results.
	Reward	Rewards the user in some way for doing positive actions.
Dialog Support	Reminders	Remind the user to continue with some goal.
	Suggestion	Propose similar activities or goals for the user.
System Credibility Support	Surface Credibility	Appear like a reputable application.
	Authority	Showcase state of the art technology built by reputable companies, which is used in the application.
Social Support	Social Support	Let the user receive support from his social network.
	Social Comparison	What happens when users compares themselves to others in the application.
	Normative influence	The influence other users have on others.
	Recognition	Getting recognized for your accomplishments in a social context.

For the content the users submitted through goals, a word cloud was made to give some indication as to what they used it for, Figure 19 shows this. It is apparent that many users used it in a school context as most of the active participants were students. However, some users appear to have accepted a challenge named "Devoted Schoolday", which is the cause of those words appearing in the word cloud.

For the persuasiveness of the application, I analysed the personality traits, along with aggregated data, such as the goals, posts, reminders, and data such as the time the user was created and the users choice of login provider. Figure 21 shows bivariate correlations between the data. There is not a lot of relevant correlations between the personality traits due to the dataset being small. However there are some, but some discretion should be shown due to the sample size.

Four significant factors were discovered in regards to personality traits. The number of goals a user has is negatively correlated with extroversion and positively correlated with conscientiousness. The number of public goals is negatively correlated to openness, which seems a bit contra-intuitive. Users who signed in with Google is positively correlated to openness.

There were some interesting significant factors that were not related to personality traits as well. Creation time is negatively correlated with the number of friends, which makes sense given the advertising happened after local recruitment. The choice of login providers for both email and facebook have significant correlations with creation time. Users who signed in with Facebook is positively correlated with creation time, which means more users who registered later in the app used facebook login. Users who signed in with Facebook is also negatively correlated with the number of friends. Users who signed in with Email is negatively correlated with creation time, this indicated that the users from the local recruitment preferred email as their login method. More on this in the discussion.

3.6 Processing data

There were a few options for how to process the data stored in Firestore, such as BigQuery, SPSS, Python, and R. BigQuery was eliminated however since it seemed hard to get into. In addition, the

size of my dataset may be a bit small compared to what this service is usually used for,

I ended up using Cloud Functions to extract the data into a CSV format, so I could further process it in R. Writing the Cloud Function proved rather tedious with multiple nested Promise.all() statements in the ".then()" call. The reason for this is that the Firestore database structure I created for the users is a nested structure. F.ex: obtaining the log entries it is required to obtain the user documents, then the goal documents, and then finally the logs. The advantage of this approach is that the Cloud Function can be rerun at a later time to generate updated CSV data. This is an expensive approach if the project would have had considerably more active users, more on this in section 3.7. Note that this process could have simplified this process would have been to create conversion events in analytics for what I wanted to track.

R proved decent for merging the different CSV files into a single file with aggregated data. Some further analysis was done in R, but it proved time-consuming, so some statistics were done in SPSS instead, with the dataset created in R. I used SPSS to obtain a table showing correlation with a bivariate analysis to detect correlations between the user's content, personality, sign-in date and choice of login provider.

3.7 Economical Aspects

The primary cost associated with the app is three folded. A few development costs, then there are the more important service costs and advertisement costs. Also, I had a few costs with motivating participants for answering the questionnaires as I held a contest, where five participants could win gift cards. Although, limited to Norway due to complicated laws in various countries. Development costs consist of the Apple Developer licence, which is a yearly fee to be allowed to publish on the app store and a one time fee to Google for publishing on Google Play at 25\$. The main service costs are the Firebase services and MapTiler cloud. Firebase provides the database, authentication, file-storage, cloud messaging functionality and much more. MapTiler provides the map tile used in the map. Advertising costs stemmed mainly from Google ads.

I will disregard the development costs and mainly look into the service costs and advertisement costs. Thus I will evaluate what user retention rate would be necessary to make the application profitable. And look a little bit into what commercial models would be appropriate for the application. Such as free (as it is now, but with ads), commercial, freemium, and free trial + subscription-based. Donation-based contribution-based.

In a panel discussion at MWC19, leading industry experts discussed business models for suitable for applications Of particular interest is the donation-based model used by Wikimedia.[46] Some of the most important is to establish trust and value-exchange in application to make it profitable [47]. I would argue that as it stands now this has to be a free application, as it does not properly engage users and most users only registers and leaves, and if users had to pay for it I think they would want a refund quite quickly. The current problems that have to be fixed are the engagement aspect.

Firebase has a quite complex pricing system, but you only pay for what you use (See: <https://firebase.google.com/pricing/>). There is a free tier, a fixed plan at 25\$ and pay as you go

plan, blaze. In the free plan in Firestore, 50000 reads and 20000 writes are free each day. Let us engage in a hypothetical scenario about pricing in Firestore. I refer to the goal list in the application a simple ListView that shows the user's goals. This is a stream that shows as many documents as the user have goals each time he visits this page. The users in the app tended to have between 1 and 6 goals, but let's say hypothetically that the average user has 4 goals. In addition, let us say each user visits this screen on average 5 times a day. This would mean that the user uses 20 document reads to accomplish this. Let us say the user also visits the feed and sees posts from friends five times a day. I set the hypothetical average to 10 posts, so 50 reads. Then there is the map where all the challenges are currently loaded when opened. This does not scale well and is not optimal at all, so every time the user is accessing the map 22 document reads occur. On a side note, this could be fixed with specifying in the query what latitude and longitude it should be restricted to. The map gets visited 5 times a day as well, so 110 reads. So during a day let's say the use the average user uses 180 reads, this would mean the free tier would support 277 users with this usage pattern.

3.8 Architecture

As Flutter is a brand new Framework/UI-Toolkit it is relevant to elaborate on the results and experiences using it for designing and architecting the application. For the architecture of the application, the pattern of choice was the bloc pattern, which was introduced with flutter. It basically consists of using a StreamBuilder where you specify the initial data, and the stream it should listen to. The stream is provided by the bloc, and the bloc defines methods to update the state. Thus the business logic is handled in the bloc and the StatefulWidget is only responsible for the UI. It should be noted that in this project I choose to use the built-in Stream, StreamController and Sink functionality. This package is called RxDart² When I was using the bloc pattern it was quite convenient to update the state of the application. Good practice in flutter is to create new widgets you can reuse instead of creating everything from scratch (from the default widgets provided).

3.9 Optimizing for Mobile

Development for different devices requires implementing layouts that look good screens of different sizes. My main device for development is a Motorola Moto X4, at 5.2". For iOS, I have a 4" iPhone SE. I assume very few people who use social media apps have smaller devices than this, therefore my lower limit for screen support is at 4" and the upper limit is not set. Additionally, the aspect ratio of the device should be either 16:9 or 18:9, the later is preferable for larger devices. However, I emphasize that the application is intended for phones and not tablets. One could argue that anything over 6.5" and/or aspect ratios approaching 1:1 will make the application look awkward, but still technically usable. In the near future applications may need to take into account foldable devices as well. Yet, it is quite easy to make a layout consistent across differently sized devices of all shapes with flutter, since it can easily make be modified with info from MediaQuery in Flutter [48].

In Figure 23 three different emulators are shown side by side, this illustration highlights the

²RxDart: <https://pub.dev/packages/rxdart>

necessity to have different devices in mind even though the application is only designed for mobile. Particularly, the iPhone is the problem child refusing to show content properly when it gets too cramped. However, in the larger phones, there may be a lot of empty space.

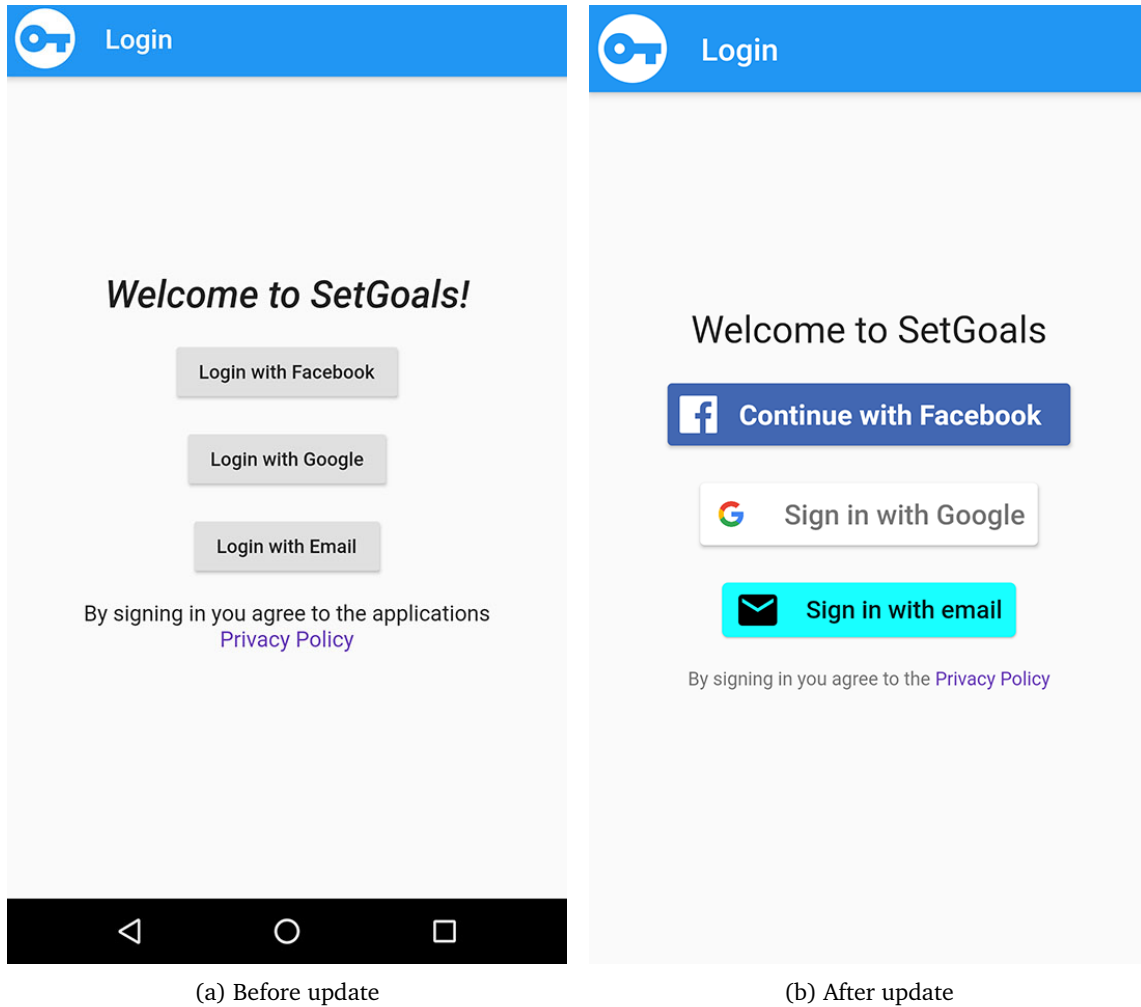


Figure 16: Shows the old and new designs of the login screens. Note the screenshots are from different devices, hence some small differences.

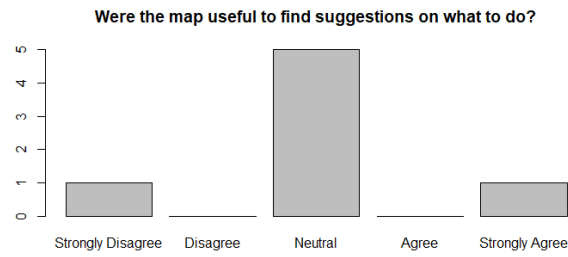


Figure 17: The usefulness of the map feature

Statistics

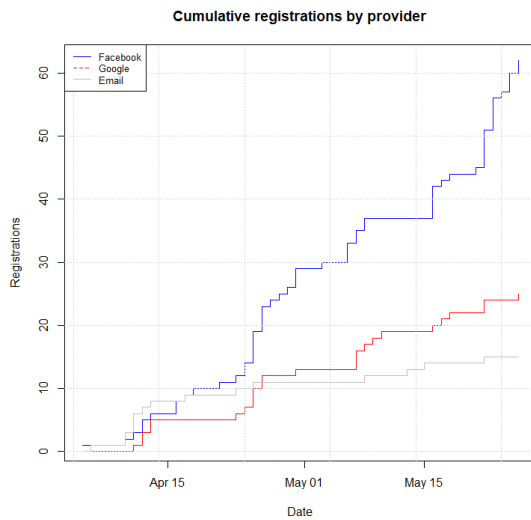
		friends	goals	publicGoals	privateGoals	goalLogs	reminders	posts	comments
N	Valid	16	23	17	9	8	4	11	2
	Missing	86	79	85	93	94	98	91	100
Mode		1	1 ^a	2	1	1	1 ^a	1	1 ^a
Range		2	5	3	2	16	5	8	1
Minimum		1	1	1	1	1	1	1	1
Maximum		3	6	4	3	17	6	9	2

a. Multiple modes exist. The smallest value is shown

Figure 18: Shows the users who submitted content of a certain type (N)



Figure 19: A word cloud of the most used words from the goal titles and descriptions



(a) The number of registrations over time.

		Provider			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Facebook	62	60.8	60.8	60.8
	Google	25	24.5	24.5	85.3
	Email	15	14.7	14.7	100.0
Total		102	100.0	100.0	

(b) The total number of registrations by provider.

Figure 20: A overview of the registrations in the application.

Correlations

		openness	conscientiousness	extroversion	agreeableness	neuroticism	goals	publicGoals	privateGoals	goalLogs	reminders	posts	friends	creationTime	facebook	google	email
openness	Pearson Correlation	1	-.306	.447	.333	-.021	-.380	-.675*	-.444	.629	.837	-.517	.086	-.212	-.300	.474	-.182
	Sig. (2-tailed)		.216	.063	.176	.934	.200	.023	.453	.371	.369	.235	.734	.397	.226	.047	.469
	N	18	18	18	18	18	13	11	5	4	3	7	18	18	18	18	18
conscientiousness	Pearson Correlation	-.306	1	-.290	.421	-.445	.674*	.520	.759	-.703	-.836	.837	.126	-.398	-.195	-.070	.247
	Sig. (2-tailed)	.216		.243	.082	.064	.011	.101	.137	.297	.369	.019	.617	.102	.439	.782	.324
	N	18	18	18	18	18	13	11	5	4	3	7	18	18	18	18	18
extroversion	Pearson Correlation	.447	-.290	1	.222	-.063	-.582*	-.599	-.246	.106	.611	-.667	-.270	-.029	.128	.155	-.268
	Sig. (2-tailed)	.063	.243		.375	.805	.037	.051	.690	.894	.581	.102	.278	.910	.612	.540	.283
	N	18	18	18	18	18	13	11	5	4	3	7	18	18	18	18	18
agreeableness	Pearson Correlation	.333	.421	.222	1	-.484*	.171	-.032	.504	-.127	-.283	.083	.246	-.446	-.081	.322	-.237
	Sig. (2-tailed)	.176	.082	.375		.042	.576	.925	.387	.873	.818	.860	.326	.064	.748	.192	.344
	N	18	18	18	18	18	13	11	5	4	3	7	18	18	18	18	18
neuroticism	Pearson Correlation	-.021	-.445	-.063	-.484*	1	-.441	-.484	-.782	-.464	.714	-.073	.150	.044	.247	-.316	.078
	Sig. (2-tailed)	.934	.064	.805	.042		.132	.131	.118	.536	.494	.876	.553	.861	.322	.202	.758
	N	18	18	18	18	18	13	11	5	4	3	7	18	18	18	18	18
goals	Pearson Correlation	-.380	.674*	-.582*	.171	-.441	1	.822**	.682*	.228	-.275	.773*	.217	-.113	-.335	.011	.340
	Sig. (2-tailed)	.200	.011	.037	.576	.132		.000	.043	.587	.725	.015	.320	.606	.118	.960	.113
	N	13	13	13	13	13	23	17	9	8	4	9	23	23	23	23	23
publicGoals	Pearson Correlation	-.675*	.520	-.599	-.032	-.484*	.822**	1	.500	-.080	-.655	.415	.100	-.231	-.242	.059	.203
	Sig. (2-tailed)	.023	.101	.051	.925	.131	.000		.667	.898	.546	.306	.703	.372	.349	.822	.436
	N	11	11	11	11	11	17	17	3	5	3	8	17	17	17	17	17
privateGoals	Pearson Correlation	-.444	.759	-.246	.504	-.782	.682*	.500	1	-.300	.0	1.000**	-.218	.504	-.598	.0	.598
	Sig. (2-tailed)	.453	.137	.690	.387	.118	.043	.667		.624	.	.000	.573	.166	.089	.000	.089
	N	5	5	5	5	5	9	3	9	5	2	3	9	9	9	9	9
goalLogs	Pearson Correlation	.629	-.703	.106	-.127	-.464	.228	-.080	-.300	1	-1.000**	-.993	.148	-.412	-.592	.488	.195
	Sig. (2-tailed)	.371	.297	.894	.873	.536	.587	.898	.624		.	.073	.727	.311	.122	.220	.643
	N	4	4	4	4	4	8	5	5	8	2	3	8	8	8	8	8
reminders	Pearson Correlation	.837	-.836	.611	-.283	.714	-.275	-.655	.0	-1.000**	1	.0	.921	-.100	-.376	-.676	.911
	Sig. (2-tailed)	.369	.369	.581	.818	.494	.725	.546079	.900	.624	.324	.089
	N	3	3	3	3	3	4	3	2	2	4	1	4	4	4	4	4
posts	Pearson Correlation	-.517	.837*	-.667	.083	-.073	.773*	.415	1.000**	-.993	.0	1	.246	-.220	-.163	-.217	.483
	Sig. (2-tailed)	.235	.019	.102	.860	.876	.015	.306	.000	.073	.	.	.466	.515	.633	.522	.133
	N	7	7	7	7	7	9	8	3	3	1	11	11	11	11	11	11
friends	Pearson Correlation	.086	.126	-.270	.246	.150	.217	.100	-.218	.148	.921	.246	1	-.455**	-.286**	.167	.192
	Sig. (2-tailed)	.734	.617	.278	.326	.553	.320	.703	.573	.727	.079	.466	.000	.004	.094	.053	.053
	N	18	18	18	18	18	23	17	9	8	4	11	102	102	102	102	102
creationTime	Pearson Correlation	-.212	-.398	-.029	-.446	.044	-.113	-.231	.504	-.412	-.100	-.220	-.455**	1	.240*	-.030	-.295**
	Sig. (2-tailed)	.397	.102	.910	.064	.861	.606	.372	.166	.311	.900	.515	.000		.015	.767	.003
	N	18	18	18	18	18	23	17	9	8	4	11	102	102	102	102	102
facebook	Pearson Correlation	-.300	-.195	.128	-.081	.247	-.335	-.242	-.598	-.592	-.376	-.163	-.286**	.240*	1	-.709**	-.517**
	Sig. (2-tailed)	.226	.439	.612	.748	.322	.118	.349	.089	.122	.624	.633	.004	.015		.000	.000
	N	18	18	18	18	18	23	17	9	8	4	11	102	102	102	102	102
google	Pearson Correlation	.474*	-.070	.155	.322	-.316	.011	.059	.0	.488	-.676	-.217	.167	-.030	-.709**	1	-.237*
	Sig. (2-tailed)	.047	.782	.540	.192	.202	.960	.822	.000	.220	.324	.522	.094	.767	.000		.017
	N	18	18	18	18	18	23	17	9	8	4	11	102	102	102	102	102
email	Pearson Correlation	-.182	.247	-.268	-.237	.078	.340	.203	.598	.195	.911	.483	.192	-.295**	-.517**	-.237*	1
	Sig. (2-tailed)	.469	.324	.283	.344	.758	.113	.436	.089	.643	.089	.133	.053	.003	.000		
	N	18	18	18	18	18	23	17	9	8	4	11	102	102	102	102	102

*. Correlation is significant at the 0.05 level (2-tailed).
 **. Correlation is significant at the 0.01 level (2-tailed).
 c. Cannot be computed because at least one of the variables is constant.

Figure 21: This table is too big to show in its entirety. Please see the Appendix B for a better view or inspect a image at <https://drive.google.com/open?id=1e000dQh8sbxERJGSC1nUqgwlUntXiDDT> Shows the correlation between personality traits and the actions taken by the users who answered the BFI, as well as the time the users were created, and their choice of login provider.

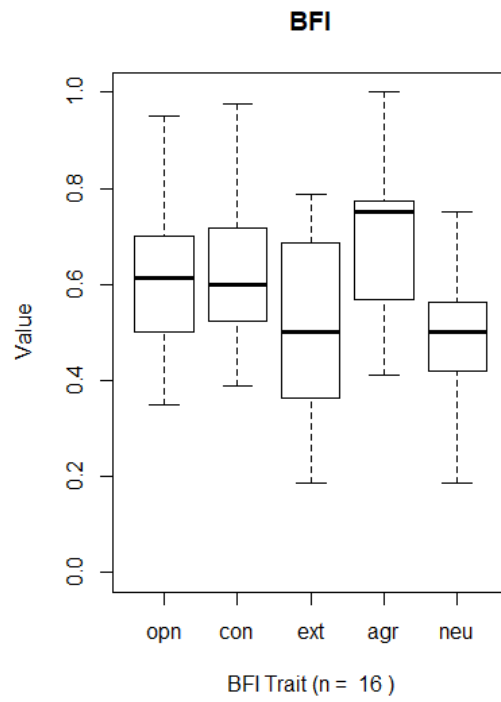


Figure 22: A box-plot of the BFI responses in the application

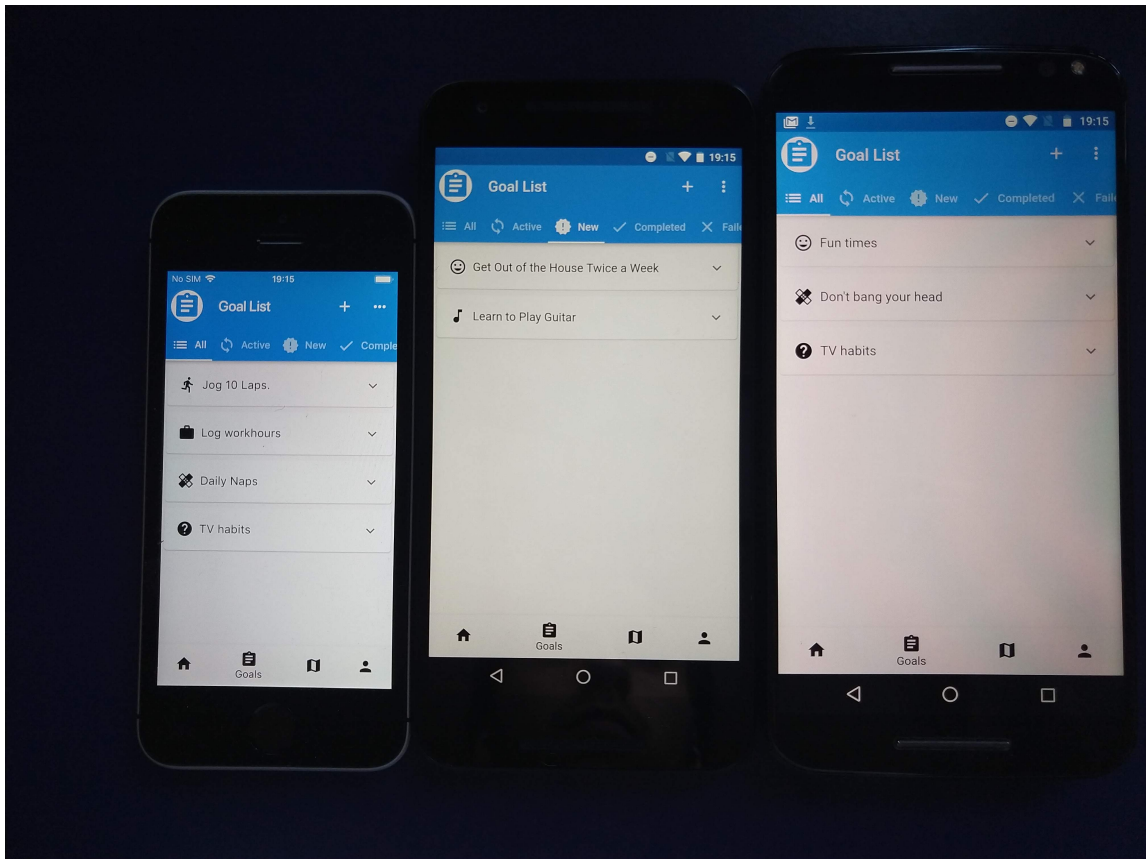


Figure 23: Illustration of the app on different screen sizes. Left: iPhone SE 4.0", middle: LG Nexus 5X 5.2" , right: Motorola Moto X Style 5.7"

4 Discussion

4.1 Technical Discussion and Application Design

4.1.1 The Home Feed

Why do I not show the users buttons that lead to add a friend or share the app functionality instead of a message? When the users arrive in the app what they see is an empty message. A better design might have been to use buttons to take users directly to the features that potentially could help them get content in their feed such as adding friends, or a goal's create post option.

Missing like functionality results in less recognition from their friends. In addition, the like would be a good hook to get the user back to into the application. There is some literature that likes may not positively impact users in an application. The cited study was done with Facebook though, which is common for literature about social media.

4.1.2 The Goal List

The major comments the goal list received was about the design, participants did not like the design, and another comment stated it would make more sense having a separate screen for the goals when they were clicked as opposed to the dropdown functionality provided by the ExpansionTile.

4.1.3 The Map

Why is not the map liked? It should have a lot of additional features such as status indicator on the challenges, is it done, active or new. It should have likes and comments. Likes would be a good status indicator which could have been used for another potential feature. Filtering to show popular challenges.

In addition, it could have been interesting to show an image of the place where the challenge is located in the widget's header. This would have to be user submitted though. Public feeds of who have participated in them posted only with the consent of course.

4.1.4 User Recruitment

Something I wonder is how the results would have been if I had used a service like [FindParticipants.com](https://www.findparticipants.com) to recruit experiment participants instead. Yet this was not an option when the site was discovered as it appears the US version too have an organisation like NSD, it is known as IRB where projects must be reviewed, and that was too much of a risk.

4.1.5 Usability

There is the question of whether it was negative or positive that the users could not log activities from previous days. It was thought this could be a motivator for having them commit to the app, but this may be hard without a properly established habit. The questionnaire showed that no user was positive to not being able to create logs for previous days.

There is the question of whether students are sceptical of using another student's social media app, so this may be a factor to consider when thinking about the validity of the correlations presented in the results. Do the users recruited from the advertisement, differ from the users recruited through personal ties? It appears users from advertisement are far less active in the application than the users from personal recruitment.

A three-step registration process may have been bothersome for some users. In hindsight, it might have been better to drop the "open consent form" button and instantly show the consent form in a screen instead of a dialogue.

I may have overcomplicated the process with the checkboxes, perhaps a check all option should have been available. The checkboxes were included due to the explicit checkboxes on NSD's forms that must be checked to agree to a research project. However, when visiting websites it is often enough to just agree with the click of a button to how their data is being used, with the option to modify the default preferences. If the project was not a research project it could probably have been done in a more similar fashion to reduce the steps needed to progress to the login, and still be a legal way of doing it. Informed consent is crucial to follow the GDPR [49].

Some users did not like that there was not an option to post without attaching an image. Introducing this feature will result in previous versions showing an error if these posts appear. A scheduler release is the best solution to prevent error on one platform. The feed is quite useless when a red error message appears down the rest of it.

There are more ways to improve the application one way is to enable more zooming on the map. This restriction was put on the application since the map API costs 20\$ if it exceeds the free tier of 100000 map tile requests a month. Given the approximately 19 participants that were active managed to use up from 9100 requests through April with this restriction. The number of requests for May is only 3950 requests. I think it would quickly exceed the limit inflicting an additional cost on the project. The Map could probably be replaced with Google maps, as no official package for Google Maps was available when the project started. It appears to be cheaper with Google Maps, as they offer 200\$ in monthly credit on the platform

The usability of the application is something I wonder about is bad at times. Since users stated they did not like the design in the questionnaire, I should have taken more time to make the app more appealing visually. Created a Theme for the application in Flutter, so it differentiates itself.

At Google I/O 2019 the Flutter team showcased how material design could be used to create visually unique applications with flutter as well as customizable with different screen sizes, device types as well as making them responsive keyboards and screen readers [48, 39]. So there are possibilities if enough time is invested in making the application aesthetically pleasing.

4.1.6 Architecture

As previously stated the architecture used for the application was the bloc pattern. There are a few different approaches, to what to use as streams in Flutter as there exists a package to make it compatible with the Reactive X API. RxDart's Observable is an extension of Flutter's Stream, and RxDart's StreamController is extensions of StreamControllers.[50]

From Google I/O 2019 the Flutter team recommended a pattern called Providers to beginners as since they received feedback that the bloc pattern was quite hard when developers were also learning FLutter [51]. Yet any architecture developers know is better than no clear architecture, if a developer knows Flux, Redux or Bloc, then those are great options [51]. Alternative patterns were considered in the early phases of the project, such as Flux and Redux since they are quite common in React. In Flutter, there is also a pattern called Scoped model which is quite similar to the inherited widgets. It differs in some ways, such as it takes a model instead of the variables inherited widgets takes, and it has more options.

4.1.7 Packages

In total, the project have 29 dependencies and 4 development dependencies, in total 33 dependencies (or packages). Usually, there was no problem with these but at times there were issues with the dependencies that use native code (more correctly plugins than packages)

4.1.8 Technical Debt

As with any project, the technical debt in a project rises over time. The worst code in the project is located in the profile page.

4.1.9 Privacy

It does seem odd that there was a preference for users in the local recruitment to use email login. It would be interesting to know why. To speculate; Was it because users feared what information I would receive about them through such a login? Or is it an indication that student within computer science and usability is more reserved towards social media providers? Or something else entirely perhaps. On a side note, phone authentication could be added to the system without cost for the first 10000 logins per month.

It would be interesting to see if more users would be active if they did not have to register to begin with. Firebase does have a login provider called Anonymous login which is useful for letting users try out the app without officially registering in the application. A user registered this way can be converted into a regular user if the user decides to connect the account with another login provider method.

4.1.10 Ways to improve the data collection

There is no secret that the data collection could have been improved, such as adding relevant analytics events for the application and researching user engagement. In addition, if the problems with the user engagement are solved and users continue to use the application as regular users then A/B testing could be an option for obtaining better measures for what works and what does not work in the app. A/B testing allows for presenting the app with small changes to different user groups.

4.2 Flutter

Programming of layouts with Flutter is quite different from f. ex. layout in Android such as LinearLayouts, RelativeLayout and ConstraintLayout. On an empirical note, to be able to define a layout with code feels like a much faster and better developer experience than switching between layout files and the code that utilizes those files.

4.3 Legal Aspects

The data one obtains from using social media providers can be small or huge. Facebook allows users to request scopes, some of which requires approval through an app review and some that don't. Google does provide basic data. The data obtained was the name and the email of the user, two indicators that count as personal information.

4.3.1 GDPR

Mishandling this information can have dire consequences, as GDPR can impose major fines for breaches to the regulation. Having to comply with the regulation takes time and creates conundrums that have to be solved in the database solution used. Some of the most notable were implementing functionality so the user can delete his or her account so that all traces of personally identifiable information is removed from the database. As the solution used a root collection for accounts, and each account is paginated to have subcollection such as posts, feed, goals, friends and reminders. Note the feed and friends collection, these are collections that will have personally identifiable information about other people. Thus, if these were to delete their account this information must be deleted from the database to comply with GDPR. This was implemented by storing references about the content a user created in other people's account subcollections. These references were stored in a collection in the account document of the content creator.

4.3.2 App Review

In regard to preventing poor quality, illegal activities and security issues, often apps must undergo reviews. This happened a few times for SetGoals. The first time Apple reviewed it there was a backlash due to poor app store presence. This was fixed with the next submission and the first version was released. When publishing an update it was rejected due to the prize draw, it had to specify official rules for the contest. This included how long the contest was open for submission, when winners will be announced and where. Users could still do the questionnaires and not participate there was a checkbox the users had to click in the rules to be eligible. This makes sense, as it is a measure to prevent fake contests and scams from appearing in the app store. After redoing the rules the release went fine through the review. Then a third release went without problems, where the questionnaire system was improved. The last was a minor change to the review was quick.

In contrast with Apple, Google only had a one-time automated review, that was done in 30 minutes when the Android version was first released in December 2018. For Facebook, there is no review unless, they detect something that violates their policies, or you request a scope that requires a review- In the latter case, a request for a scope must have a reason for doing so, it will

not be granted without a clear necessity for it, such as friendlist. SetGoals did undergo a review with Facebook due to Facebook detected an unofficial SDK was used to access the app.

4.4 The app

Some of the advantages of the framework Flutter is that both the Android and iOS app stay true to the native look and feel of their OS. For example, the action overflow icon changes depending on the OS. The scroll overflow effect does the same, thus no users should feel completely foreign in the app, Flutter does attempt to stay true to the look and feel of the native application's platform.

4.4.1 Updating the app

Releasing app updates can take a while. On the Android's Play Store it is quite quick, but on Apple, every new update must go through a review. In total, the app failed 3 app reviews with Apple. The review is to ensure the application follows Apple's policies and to keep a certain quality in their app store. This was failed once for the application's first release, due to a poor app store presence. Later after the app was released, the review failed due to I forgot running "flutter build ios" before archiving a build. This resulted in the app only displaying a white screen to the reviewer, which is a good thing that the review picked up, otherwise a broken build would have been published. The white screen was a result the release version being improperly configured, resulting in this error message being printed to the console: "Could not find an option named "track-widget-creation" ¹ The last time it failed was due to the incentive for users to participate in the research project, a prize draw. I had to more clearly specify the rules of the prize draw and specify that Apple did not sponsor the prize draw.

This can introduce problems, such as when new features in the update would result in errors in the previous version. One example was when changing the posts on the home feed to allow posts without an image. The previous version required an image and gave an error when no image was attached.

In addition, the app had a policy warning with Facebook for the android app, for using an unofficial SDK for the login. The app uses a plugin that simplifies the login process for facebook login, and this plugin offers an option for forcing the facebook login to use web views for login. This was what caused the policy warning to be triggered, I had a week to comply with the warning if not restrictions would be enforced to my app.

The release process could have been simplified if I had used Continuous Integration / Continuous Delivery (CI/CD) services. A CI/CD service called Code Magic is specifically developed for Flutter. Code Magic provides automatic code signing which would have eliminated the need for a Mac to develop for iOS .[52] This service was at the conference Flutter Live was held in December 2018 so it is quite new for Flutter.

¹<https://github.com/flutter/flutter/issues/26279>

4.5 Ethical Aspects

First of all, what could possibly go wrong with an application like this? That is a lot, if a persuasive application does not work as intended it can have an unintended side-effect, and these effects may have a long halftime effect[30].

If the users are not aware of the persuasion is it then manipulation and coercion instead?[30]

Persuasion may be an interesting dilemma in terms of ethics

Some sources have highlighted various issues with it. Take[53] they created a concept for a surreal (slightly dystopian) persuasive exercise application. Illustrating the dangers of just looking at metrics, thus not seeing the user for a user but just a quantity of data. In addition to what does the system design promote? Does the user get dependent on the system for performing the behaviour, or does the user develop behaviours independent of the system?

What[53] suggest seem slightly related to what[54] presents in their work in terms of how group peer sessions work in addiction treatment.

4.5.1 Publication bias

There is a publication bias in the domain where mostly positive results are published. Negative results, on the other hand, seem to come up short. Some factors[2] mentioned contributed to this was the fear of being ridiculed, or not being funded in future endeavours. Negative results are helpful for new research endeavours since it helps the researchers avoid the pitfalls of the experiments that reported negative results.

4.6 Persuasive Technology Techniques (PTT)

As I probably have mentioned there is many different PTT, in Oinas-Kukkonen and Harjumma's paper they presented a collection of various techniques used within the field. In particular, they categorized them as primary task support, dialogue support, system credibility support, and social support.

4.7 Open Source

Open sourcing code is a good way to share code with others and allows others to contribute or continue someone else's work. I have decided to open source this application under a non-commercial license. Since then other researchers, flutter developers, or other people who are interested in the application can check it out, by forking it. Researchers in persuasive technology could possibly use it if they see fit. The application is available on my GitHub Account here: <https://github.com/TMHolen95/SetGoals>.

4.8 Complying with GDPR and NSD

GDPR endorses privacy by design. For the experiment, I had to submit an application to NSD, a necessity when processing personally identifiable data. The app does handle the email addresses of the users, and name of the users.

Of particular importance is to make the user understand what joining an experiment or signing

in to an application involves. In a normal research experiment that involves personally identifiable data the user usually signs an information letter. However, this would have been less than ideal for this experiment. Instead, the information letter was implemented as a dialogue. The dialogue consisted of all the information a normal information letter contained, and four checkboxes the user had to check. After checking the boxes first then the user could log in to the application.

The consent could easily be withdrawn in the applications legal section. As it is a security essential operation firebase did not allow this operation without a fresh login, a descriptive error message was presented to the user in this case. This action would remove all trace of their personal data from the database, which quite conveniently were executed from a cloud function. Additionally, the legal section had information on how to proceed if the user wanted to correct personal data, or request a copy of his/her data. In essence, this was simplified to send me an email, GDPR requires me to reply within 30 days.

The user has the right to withdraw his consent at any time. However, this shall not be of a hindrance for research. Imagine a research project processing a person data, and the person withdraws his consent. If the project were to redo all their calculations due to consent withdrawal, the workload is unreasonable. However, the data from this user may not be used in the datasets of new calculations. Shortly summarized withdrawal of consent affects datasets, not existing results already processed from the datasets.

Optimising for these challenges presents some challenges. Think about caching, which involves storing some data locally on a device. Caching does lead to more responsive and better user experience. However, if a user invokes his right to be forgotten then caching may raise some ethical dilemmas. Such as, when the database deletes the images there may still be copies in other user's caches. The images may not be shown to these users, but the images could still exist on their devices. If someone tampered with the source code then they could in theory still retrieve these images.

It is hard to do anything from the database in this scenario. Any possible solution must be implemented in the application. One option is to invalidate the cached data after a certain time has passed. This would not solve the original problem, but it would lessen the chance of illicit retrieval.

4.9 Loss of users

The loss of users before registration may have been caused by unofficial login buttons.

Something that seems apparent is that social features are not as easy to get people to use.

4.10 External Factors

There were a lot of external factors to comply with within this project such as Apple, Facebook, and NSD & GDPR. App reviews took time both for Apple and that one case with Facebook. In the start, it took quite a while to get the green light from NSD, but as the project followed the essentials of GDPR it was good to go in the end. Implementing the required functionality for GDPR took time, most time consuming was the right to be forgotten. Interestingly, no user ever sent me a mail to request their data corrected or receive a copy of their data. However, these are requirements of the

law, the GDPR [49], so they should be respected.

4.11 Login/Notification bug

After the initial release of the application a bug was discovered with the login system, multiple users encountered this error. Login worked perfectly in normal circumstances, but when the app was terminated and a user clicked a notification then the bug became apparent. Notifications from Firebase messaging have three different listeners `onMessage`, `onResume`, and `onLaunch`. `onMessage` is triggered when the app is running, and `onResume` is triggered when the app is in the background. For `onLaunch` the application starts from scratch, unfortunately, there were some race conditions that resulted in a null pointer. The fun part with this bug was that the stack trace was not available like it is with a normal "flutter run". In addition, Flutter does not print in LogCat, it does so to a run window in the editor (Android Studio, IntelliJ, or Visual Studio Code) or the terminal it is run from. When terminating the app you no longer receive this information, so initially, it was very hard to debug.

Fortunately, it became easier when I discovered that the logs could be accessed when running another Flutter app. This was not perfect as the logs usually were from a slightly older build than the last one run. I got the latest code changes live by uninstalling the app and then running "flutter clean" from the terminal followed by "flutter run" to install the debug version of the app. Then I could terminate the app, run "flutter run" on the other app, followed by clicking on a notification from firebase messaging. This obtained the stack trace and print statements from the app in the other project's run window.

The fix was to acquire the previous `FirebaseUser` before the app was run in Dart's main method. I had previously implemented a loading-page inside the app before, this widget decided where to route the user. Authenticated users to the `HomePage` and unauthenticated users to the `Login` screens.

4.12 Bug: Wrong Image When Posting

At a point, there was a bug when posting in the app. When a new post arrived in the feed the image from the post below it would occupy the new post. This was solved by attaching the posts and images a `UniqueKey`. In Flutter, keys are important when adding new elements to lists or moving the position of widgets. Without them, unexpected behaviour may occur, since the framework sees the correct widget type in the widget tree and assumes everything is okay.

5 Conclusion

5.1 Main Contributions

An argument can be made for the difficulties with app-based interventions, these range from how to engage the app's audience into forming habits. It can be stated that this app failed to do so. The project does, however, present advertisement data, along with user retention data. The project highlighted a problem with keeping users, so it can serve as an example. The data sample obtained was poor. This made it hard to answer the original hypotheses, but some correlations between personality and the content the users produced were found. Intriguingly a preference for email login was discovered in the users recruited through local recruitment early in the project, compared to the users' recruited through advertisement.

In addition, many features were hardly used such as user-set reminders, map challenges, and comment on peoples posts. Both reminders and comments were features which would notify a user through a notification, but these measures were not sufficient. Although, the suggestion I can provide to other Social Media Goal Applications is to ensure that there are enough features that remind the users that the application exists, thus find provide better ways of getting the users into habits of using the application.

Regardless the application has been open-sourced under a non-commercial licence. This means that other people can continue the work if they so desire. This will require renaming the applications package name on Android and iOS, as these are a unique identifier on the respective stores. In addition, a firebase project must be created, and some setup is required for connecting the projects. On Android, google-services.json must be replaced to the project's "android/app" folder, on iOS in the folder "ios/Runner" a similar file must be replaced named GoogleService-Info.plist. The procedure for adding an Android app and iOS app in the firebase documentation is of relevance. In addition, the Cloud Function project must be set up properly, through the Firebase CLI.

5.2 Reflections

The project is vast both in terms of development, and research domains. A social media application, that is not a small application, it consists of many parts that communicate with each other. the friend system, the feed, the commenting system, the profile system, the map, in addition to redirecting the user to pages where

In hindsight, the application did never receive a dedicated colour theme that set it apart from other applications, which potentially could be an interesting factor to address. Ironically in other applications, I have created this has been something of the first I have addressed, as it is a quite fun task.

The design of the application, would it have been better with a *Drawer* instead of a *Bottom-*

NavigationBar. As it would avoid some of the problems with the profile page for example. The profile page is quite messy with some unrelated content, such as the BFI-test and questionnaires. This could have been solved with a profile section in the top of the drawer, and the BFI-test and questionnaires could have had their own sections in the menu further below. In addition, it could have been possible to have a deeper navigation tree readily available.

However, it is worth noting that the application uses material design something that is uncommon on iOS. This may have affected the users on iOS negatively as the app may have felt foreign to what they are used to. To solve this Flutter provides a Cupertino package that could have been used, yet if this were to be used in addition it would have afflicted an extra workload on the project. An alternative option would have been to differentiate the application better, in other words, create a consistent style and branding of the application.

I would not recommend having such a wide topic for a master thesis. The reason for this is that it is development heavy, and requires a lot of mental context switching, between the different environments. Such as moving from some of these tasks to another Flutter, Cloud Functions, Firestore, Security Rules, managing releases, preparing for reviews, creating ads, and managing ads.

It would have been preferable to have it as a group project instead, so people can specialise in certain tasks.

However I have learned a lot like cloud technology with firebase's services, better grasp of futures and promises along with async and await. A new frontend-framework Flutter, and experience with maps.

Some of the areas the topic touched were advertising, habit formation, and user retention.

I could have behaved differently in the application myself.

What I feel most positive of from the project, that is how interesting it has been to learn Flutter and to use it in practice. It has been fun to reach a level where I feel comfortable with answering other peoples questions on stack overflow. Which - also served as a good way to learn new aspects of the framework, or discover widgets I was not aware of from before. Even though I have

Yet one is never fully educated in a subject or technology, it seems like there is always something new to learn. For example, while learning Firebase Cloud Functions.

Lastly, if you're going to publish an app that targets China, do not use Firebase as it is blocked.

5.3 Further Work

5.3.1 Persuasive Software Features

Application development takes time and to implement persuasive software features takes time. The application's context is social media and general purpose goal tracking. Thus the context is quite wide and software features can be implemented differently. Therefore, it is hard to say whether the results can be generalized to other domains. Future works should also check if the results are reproducible.

Another topic is validating additional features. The list proposed by Oinas-Kukkonen is vast and many of them are relevant for this application. Yet a time is a limiting factor, as only a few could be

implemented in the project's scope. Therefore other projects should

5.3.2 User Testing

Next, there is the possibility of performing a bigger user test and try to discern more nuances in the data.

5.3.3 Sharing

A feature in firebase that could improve the app substantially is dynamic links. A dynamic link could contain information to take a user to a specific place in an application. The great thing about dynamic links is that they persist through the installation process. It could allow the users to be redirected to anywhere in the application.

5.4 Deprecation

It is apparent that mobile and cloud technology progresses fast. During the lifespan of the project, both Flutter and Firestore went from beta technology goes into production. In addition, various plugins had breaking changes and for Cloud Functions the runtime environment got deprecated. In the earlier phases of Flutter, various first-party packages stated that breaking changes could occur. This happened, however only one time that impacted this project. The Firebase Authentication plugin introduced some breaking changes that changed how the sign-in was handled.

The default node runtime to the Google Cloud Functions started out at Node 6, with node 8 in beta. Later Node 6 was deprecated, and Node 8 became the default runtime, with node 10 support in beta. Which means that the functions in the project need to be converted into Node 8 or Node 10 functions. The changes need to be in place within 2020 as support will be removed at this point. In addition, there is a recent requirement that 64-bit APK's must also be uploaded through Google Play, the benefit of this is a reduced size for the APK. Universal APK's are larger as they include support for more CPU architectures.

5.5 Bad internet connections

A problem with the login occurred when users were on an unstable internet connection. Step one the user is registering they sign in with google, facebook or email. Step two they have completed registration and they return to the app, but the state does not update properly. When this problem happened, the users were stuck seeing a circular progress indicator.

5.6 Limitations of this work

The limitations of this works are that the main data in regards to personality and the content produced in the application consists of users who either were convenience sampled through local recruitment or were recruited by a participant on from the aforementioned group. Also, users may have been motivated to answering the questionnaire through the prize draw.

Appendices

A Designing the questionnaire

A.1 Original Questionnaire

Initially, 7 questionnaires were drafted with a total of 88 questions, like the BFI these were thought to be answered by a Likert scale. After being reviewed by my supervisor this was too many questions, and there could have occurred problems with getting users to answer all of them. An update was issued to the application to enable the system to administer questions containing all or a subset of an image, a question-text, a Likert-scale, and a free-text-form.

A.2 Revised Questionnaire

The following is the text the users encountered through the tutorial questionnaire: Question 1: Thanks for participating! There are two methods for answering a question, Likert scales and text fields. Both can be present, but if only a Likert scale is shown it will automatically proceed to the next question when tapped. Ready to proceed?

Question 2: The text fields have to be filled in with some text before you can progress when done tap the top-right arrow to go to the next question. If you do not want to provide an answer, feel free to type something random to proceed or explain why you don't want to answer. Want to continue?

Question 3: Here an image is attached to the question. The image shows the submit button, and a "Thank You!" message which means the questionnaire has been successfully submitted. You will use the button in the next question. Ready for the next question?

Question 4: A submit button will appear, when tapping the top-right arrow when you have answered the question. Ready to submit?

General Questions

- What is your age?
- How were you recruited to this application?
- What is your gender?
- Do you currently reside within Norway?

Social Questions

- Did you feel the application's social features (posts and comments) were useful for your progress?
- Did seeing posts from friends make you want to compete or collaborate with them towards a goal?
- Did you find the point system motivating, engaging, or competitive?

Goal and Logging

- What did you and did not like about the goal logging feature?
- What did you and did not like about the view data feature?
- How did you feel about having to log the data on the same day you did your activity?
- Would you continue to use this application, why or why not?
- How did you feel about receiving the reminders, did they help you to work towards your goal?

Map challenges

- Did you use the map feature, why and why not?
- Were the map useful to find suggestions to what to do?
- Do you think the map feature could be improved, if so do you have any suggestions?

Privacy Questions

- Which of these login methods are you comfortable using: Email, Facebook, Google, or other methods?
- Would you be comfortable with other people seeing your goal's logged data, if so who?
- Who are you comfortable with sharing your personal goals too? (not private goals)

Feedback

- Did you encounter any errors in the application?
- If you have any more feedback please provide it through the feedback dialogue. Thanks for answering these questions, if all questionnaires are checked and you have consented to the contest rules on the profile page you are eligible for participating in the prize draw. Remember to submit, thanks again. (show image of feedback dialogue)

B Content Analysis

On the following pages a pdf is imported from SPSS which shows the correlations between the personality, content, registration time, and choice of login provider. Refer to Section [3.5](#) for a summary.

Correlations

		openness	conscientiousness	extroversion	agreeableness
openness	Pearson Correlation	1	-.306	.447	.333
	Sig. (2-tailed)		.216	.063	.176
	N	18	18	18	18
conscientiousness	Pearson Correlation	-.306	1	-.290	.421
	Sig. (2-tailed)	.216		.243	.082
	N	18	18	18	18
extroversion	Pearson Correlation	.447	-.290	1	.222
	Sig. (2-tailed)	.063	.243		.375
	N	18	18	18	18
agreeableness	Pearson Correlation	.333	.421	.222	1
	Sig. (2-tailed)	.176	.082	.375	
	N	18	18	18	18
neuroticism	Pearson Correlation	-.021	-.445	-.063	-.484*
	Sig. (2-tailed)	.934	.064	.805	.042
	N	18	18	18	18
goals	Pearson Correlation	-.380	.674*	-.582*	.171
	Sig. (2-tailed)	.200	.011	.037	.576
	N	13	13	13	13
publicGoals	Pearson Correlation	-.675*	.520	-.599	-.032
	Sig. (2-tailed)	.023	.101	.051	.925
	N	11	11	11	11
privateGoals	Pearson Correlation	-.444	.759	-.246	.504
	Sig. (2-tailed)	.453	.137	.690	.387
	N	5	5	5	5
goalLogs	Pearson Correlation	.629	-.703	.106	-.127
	Sig. (2-tailed)	.371	.297	.894	.873
	N	4	4	4	4
reminders	Pearson Correlation	.837	-.836	.611	-.283
	Sig. (2-tailed)	.369	.369	.581	.818
	N	3	3	3	3
posts	Pearson Correlation	-.517	.837*	-.667	.083
	Sig. (2-tailed)	.235	.019	.102	.860
	N	7	7	7	7
friends	Pearson Correlation	.086	.126	-.270	.246
	Sig. (2-tailed)	.734	.617	.278	.326
	N	18	18	18	18

Correlations

		neuroticism	goals	publicGoals	privateGoals
openness	Pearson Correlation	-.021	-.380	-.675*	-.444
	Sig. (2-tailed)	.934	.200	.023	.453
	N	18	13	11	5
conscientiousness	Pearson Correlation	-.445	.674*	.520	.759
	Sig. (2-tailed)	.064	.011	.101	.137
	N	18	13	11	5
extroversion	Pearson Correlation	-.063	-.582*	-.599	-.246
	Sig. (2-tailed)	.805	.037	.051	.690
	N	18	13	11	5
agreeableness	Pearson Correlation	-.484*	.171	-.032	.504
	Sig. (2-tailed)	.042	.576	.925	.387
	N	18	13	11	5
neuroticism	Pearson Correlation	1	-.441	-.484	-.782
	Sig. (2-tailed)		.132	.131	.118
	N	18	13	11	5
goals	Pearson Correlation	-.441	1	.822**	.682*
	Sig. (2-tailed)	.132		.000	.043
	N	13	23	17	9
publicGoals	Pearson Correlation	-.484	.822**	1	.500
	Sig. (2-tailed)	.131	.000		.667
	N	11	17	17	3
privateGoals	Pearson Correlation	-.782	.682*	.500	1
	Sig. (2-tailed)	.118	.043	.667	
	N	5	9	3	9
goalLogs	Pearson Correlation	-.464	.228	-.080	-.300
	Sig. (2-tailed)	.536	.587	.898	.624
	N	4	8	5	5
reminders	Pearson Correlation	.714	-.275	-.655	. ^c
	Sig. (2-tailed)	.494	.725	.546	.
	N	3	4	3	2
posts	Pearson Correlation	-.073	.773*	.415	1.000**
	Sig. (2-tailed)	.876	.015	.306	.000
	N	7	9	8	3
friends	Pearson Correlation	.150	.217	.100	-.218
	Sig. (2-tailed)	.553	.320	.703	.573
	N	18	23	17	9

Correlations

		goalLogs	reminders	posts	friends
openness	Pearson Correlation	.629	.837	-.517	.086
	Sig. (2-tailed)	.371	.369	.235	.734
	N	4	3	7	18
conscientiousness	Pearson Correlation	-.703	-.836	.837*	.126
	Sig. (2-tailed)	.297	.369	.019	.617
	N	4	3	7	18
extroversion	Pearson Correlation	.106	.611	-.667	-.270
	Sig. (2-tailed)	.894	.581	.102	.278
	N	4	3	7	18
agreeableness	Pearson Correlation	-.127	-.283	.083	.246
	Sig. (2-tailed)	.873	.818	.860	.326
	N	4	3	7	18
neuroticism	Pearson Correlation	-.464	.714	-.073	.150
	Sig. (2-tailed)	.536	.494	.876	.553
	N	4	3	7	18
goals	Pearson Correlation	.228	-.275	.773*	.217
	Sig. (2-tailed)	.587	.725	.015	.320
	N	8	4	9	23
publicGoals	Pearson Correlation	-.080	-.655	.415	.100
	Sig. (2-tailed)	.898	.546	.306	.703
	N	5	3	8	17
privateGoals	Pearson Correlation	-.300	. ^c	1.000**	-.218
	Sig. (2-tailed)	.624	.	.000	.573
	N	5	2	3	9
goalLogs	Pearson Correlation	1	-1.000**	-.993	.148
	Sig. (2-tailed)	.	.	.073	.727
	N	8	2	3	8
reminders	Pearson Correlation	-1.000**	1	. ^c	.921
	Sig. (2-tailed)079
	N	2	4	1	4
posts	Pearson Correlation	-.993	. ^c	1	.246
	Sig. (2-tailed)	.073	.	.	.466
	N	3	1	11	11
friends	Pearson Correlation	.148	.921	.246	1
	Sig. (2-tailed)	.727	.079	.466	.
	N	8	4	11	102

Correlations

		creationTime	facebook	google	email
openness	Pearson Correlation	-.212	-.300	.474*	-.182
	Sig. (2-tailed)	.397	.226	.047	.469
	N	18	18	18	18
conscientiousness	Pearson Correlation	-.398	-.195	-.070	.247
	Sig. (2-tailed)	.102	.439	.782	.324
	N	18	18	18	18
extroversion	Pearson Correlation	-.029	.128	.155	-.268
	Sig. (2-tailed)	.910	.612	.540	.283
	N	18	18	18	18
agreeableness	Pearson Correlation	-.446	-.081	.322	-.237
	Sig. (2-tailed)	.064	.748	.192	.344
	N	18	18	18	18
neuroticism	Pearson Correlation	.044	.247	-.316	.078
	Sig. (2-tailed)	.861	.322	.202	.758
	N	18	18	18	18
goals	Pearson Correlation	-.113	-.335	.011	.340
	Sig. (2-tailed)	.606	.118	.960	.113
	N	23	23	23	23
publicGoals	Pearson Correlation	-.231	-.242	.059	.203
	Sig. (2-tailed)	.372	.349	.822	.436
	N	17	17	17	17
privateGoals	Pearson Correlation	.504	-.598	. ^c	.598
	Sig. (2-tailed)	.166	.089	.000	.089
	N	9	9	9	9
goalLogs	Pearson Correlation	-.412	-.592	.488	.195
	Sig. (2-tailed)	.311	.122	.220	.643
	N	8	8	8	8
reminders	Pearson Correlation	-.100	-.376	-.676	.911
	Sig. (2-tailed)	.900	.624	.324	.089
	N	4	4	4	4
posts	Pearson Correlation	-.220	-.163	-.217	.483
	Sig. (2-tailed)	.515	.633	.522	.133
	N	11	11	11	11
friends	Pearson Correlation	-.455**	-.286**	.167	.192
	Sig. (2-tailed)	.000	.004	.094	.053
	N	102	102	102	102

Correlations

		openness	conscientiousness	extroversion	agreeableness
creationTime	Pearson Correlation	-.212	-.398	-.029	-.446
	Sig. (2-tailed)	.397	.102	.910	.064
	N	18	18	18	18
facebook	Pearson Correlation	-.300	-.195	.128	-.081
	Sig. (2-tailed)	.226	.439	.612	.748
	N	18	18	18	18
google	Pearson Correlation	.474 [*]	-.070	.155	.322
	Sig. (2-tailed)	.047	.782	.540	.192
	N	18	18	18	18
email	Pearson Correlation	-.182	.247	-.268	-.237
	Sig. (2-tailed)	.469	.324	.283	.344
	N	18	18	18	18

Correlations

		neuroticism	goals	publicGoals	privateGoals
creationTime	Pearson Correlation	.044	-.113	-.231	.504
	Sig. (2-tailed)	.861	.606	.372	.166
	N	18	23	17	9
facebook	Pearson Correlation	.247	-.335	-.242	-.598
	Sig. (2-tailed)	.322	.118	.349	.089
	N	18	23	17	9
google	Pearson Correlation	-.316	.011	.059	. ^c
	Sig. (2-tailed)	.202	.960	.822	.000
	N	18	23	17	9
email	Pearson Correlation	.078	.340	.203	.598
	Sig. (2-tailed)	.758	.113	.436	.089
	N	18	23	17	9

Correlations

		goalLogs	reminders	posts	friends
creationTime	Pearson Correlation	-.412	-.100	-.220	-.455**
	Sig. (2-tailed)	.311	.900	.515	.000
	N	8	4	11	102
facebook	Pearson Correlation	-.592	-.376	-.163	-.286**
	Sig. (2-tailed)	.122	.624	.633	.004
	N	8	4	11	102
google	Pearson Correlation	.488	-.676	-.217	.167
	Sig. (2-tailed)	.220	.324	.522	.094
	N	8	4	11	102
email	Pearson Correlation	.195	.911	.483	.192
	Sig. (2-tailed)	.643	.089	.133	.053
	N	8	4	11	102

Correlations

		creationTime	facebook	google	email
creationTime	Pearson Correlation	1	.240*	-.030	-.295**
	Sig. (2-tailed)		.015	.767	.003
	N	102	102	102	102
facebook	Pearson Correlation	.240*	1	-.709**	-.517**
	Sig. (2-tailed)	.015		.000	.000
	N	102	102	102	102
google	Pearson Correlation	-.030	-.709**	1	-.237*
	Sig. (2-tailed)	.767	.000		.017
	N	102	102	102	102
email	Pearson Correlation	-.295**	-.517**	-.237*	1
	Sig. (2-tailed)	.003	.000	.017	
	N	102	102	102	102

*. Correlation is significant at the 0.05 level (2-tailed).

**.. Correlation is significant at the 0.01 level (2-tailed).

c. Cannot be computed because at least one of the variables is constant.

Bibliography

- [1] Hamari, J. & Koivisto, J. 2015. “working out for likes”: An empirical study on social influence in exercise gamification. *Computers in Human Behavior*, 50, 333–347.
- [2] Stibe, A. & Cugelman, B. 2016. Persuasive backfiring: when behavior change interventions trigger unintended negative outcomes. In *International Conference on Persuasive Technology*, 65–77. Springer.
- [3] Oinas-Kukkonen, H. & Harjumaa, M. 2008. Towards deeper understanding of persuasion in software and information systems. In *Proceedings of the First International Conference on Advances in Computer-Human Interaction, ACHI '08*, 200–205, Washington, DC, USA. IEEE Computer Society. URL: <http://dx.doi.org/10.1109/ACHI.2008.31>, doi:10.1109/ACHI.2008.31.
- [4] Oinas-Kukkonen, H. & Harjumaa, M. 2008. A systematic framework for designing and evaluating persuasive systems. In *International conference on persuasive technology*, 164–176. Springer.
- [5] Cialdini, R. B. 2001. Harnessing the science of persuasion. *Harvard business review*, 79(9), 72–81.
- [6] Orji, R., Nacke, L. E., & Di Marco, C. 2017. Towards personality-driven persuasive health games and gamified systems. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17*, 1015–1027, New York, NY, USA. ACM. URL: <http://doi.acm.org/10.1145/3025453.3025577>, doi:10.1145/3025453.3025577.
- [7] Orji, R. 2017. Why are persuasive strategies effective? exploring the strengths and weaknesses of socially-oriented persuasive strategies. In *International Conference on Persuasive Technology*, 253–266. Springer.
- [8] Oduor, M., Alahäivälä, T., & Oinas-Kukkonen, H. Oct 2014. Persuasive software design patterns for social influence. *Personal and Ubiquitous Computing*, 18(7), 1689–1704. URL: <https://doi.org/10.1007/s00779-014-0778-z>, doi:10.1007/s00779-014-0778-z.
- [9] Hanus, M. D. & Fox, J. 2015. Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Computers & Education*, 80, 152–161.

- [10] Deci, E. L. & Ryan, R. M. 2008. Self-determination theory: A macrotheory of human motivation, development, and health. *Canadian psychology/Psychologie canadienne*, 49(3), 182.
- [11] Mattheiss, E., Hochleitner, C., Busch, M., Orji, R., & Tscheligi, M. 2017. Deconstructing pokémon go – an empirical study on player personality characteristics. In *Persuasive Technology: Development and Implementation of Personalized Technologies to Change Attitudes and Behaviors*, de Vries, P. W., Oinas-Kukkonen, H., Siemons, L., Beerlage-de Jong, N., & van Gemert-Pijnen, L., eds, 83–94, Cham. Springer International Publishing.
- [12] Bartle, R. 1996. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD research*, 1(1), 19.
- [13] Althoff, T., White, W. R., & Horvitz, E. Dec 2016. Influence of pokémon go on physical activity: Study and implications. *J Med Internet Res*, 18(12), e315. URL: <http://www.jmir.org/2016/12/e315/>, doi:10.2196/jmir.6759.
- [14] Hanke, J. 02 2019. John hanke’s mobile world congress 2019 keynote: The biggest paradigm shift in mobile computing. Visited: 15/05/19. URL: <https://www.youtube.com/watch?v=Zs-T6HNPIkc>.
- [15] Frith, J. 2013. Turning life into a game: Foursquare, gamification, and personal mobility. *Mobile Media & Communication*, 1(2), 248–262.
- [16] Link, M. W., Lai, J., & Bristol, K. 2014. Not so fun? the challenges of applying gamification to smartphone measurement. In *Design, User Experience, and Usability. User Experience Design Practice*, Marcus, A., ed, 319–327, Cham. Springer International Publishing.
- [17] van Dantzig, S., Geleijnse, G., & van Halteren, A. T. Aug 2013. Toward a persuasive mobile application to reduce sedentary behavior. *Personal and Ubiquitous Computing*, 17(6), 1237–1246. URL: <https://doi.org/10.1007/s00779-012-0588-0>, doi:10.1007/s00779-012-0588-0.
- [18] Burke, M. & Kraut, R. E. 2016. The relationship between facebook use and well-being depends on communication type and tie strength. *Journal of Computer-Mediated Communication*, 21(4), 265–281.
- [19] Kamal, N. & Fels, S. 2012. Determining the determinants of health behaviour change through an online social network. In *International Conference on Persuasive Technology*, 1–12. Springer.
- [20] Fialho, A. T., van den Heuvel, H., Shahab, Q., Liu, Q., Li, L., Saini, P., Lacroix, J., & Markopoulos, P. 2009. Activeshare: sharing challenges to increase physical activities. In *CHI’09 Extended Abstracts on Human Factors in Computing Systems*, 4159–4164. ACM.
- [21] Takayama, C., Lehdonvirta, V., Shiraishi, M., Washio, Y., Kimura, H., & Nakajima, T. 2009. Ecoisland: A system for persuading users to reduce co2 emissions. In *Future Dependable Distributed Systems, 2009 Software Technologies for*, 59–63. IEEE.

- [22] Comber, R., Thieme, A., Rafiev, A., Taylor, N., Kr ä mer, N., & Olivier, P. 2013. Bincam: Designing for engagement with facebook for behavior change. In *IFIP Conference on Human-Computer Interaction*, 99–115. Springer.
- [23] Herrmann, K., Ziegler, J. u. r., & Dogan ü n, A. u. l. 2016. Supporting users in setting effective goals in activity tracking. In *International Conference on Persuasive Technology*, 15–26. Springer.
- [24] Tikka, P. & Oinas-Kukkonen, H. 2016. Rightontime: The role of timing and unobtrusiveness in behavior change support systems. In *International Conference on Persuasive Technology*, 327–338. Springer.
- [25] Busch, M., Mattheiss, E., Reisinger, M., Orji, R., Fr ö hlich, P., & Tscheligi, M. 2016. More than sex: The role of femininity and masculinity in the design of personalized persuasive games. In *International Conference on Persuasive Technology*, 219–229. Springer.
- [26] Adaji, I. & Vassileva, J. 2016. Persuasive patterns in q&a social networks. In *International Conference on Persuasive Technology*, 189–196. Springer.
- [27] Oyibo, K. & Vassileva, J. 2017. Investigation of social predictors of competitive behavior in persuasive technology. In *International Conference on Persuasive Technology*, 279–291. Springer.
- [28] Busch, M., Schrammel, J., & Tscheligi, M. 2013. Personalized persuasive technology—development and validation of scales for measuring persuadability. In *International Conference on Persuasive Technology*, 33–38. Springer.
- [29] Waardenburg, T., Winkel, R., & Lamers, M. H. 2012. Normative social influence in persuasive technology: Intensity versus effectiveness. In *Persuasive Technology. Design for Health and Safety*, Bang, M. & Ragnemalm, E. L., eds, 145–156, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [30] Atkinson, B. M. 2006. Captology: A critical review. In *International conference on persuasive technology*, 171–182. Springer.
- [31] Nacke, L. E., Bateman, C., & Mandryk, R. L. 2014. Brainhex: A neurobiological gamer typology survey. *Entertainment computing*, 5(1), 55–62.
- [32] Meschtscherjakov, A., G ä rtner, M., Mirnig, A., R ö del, C., & Tscheligi, M. 2016. The persuasive potential questionnaire (ppq): challenges, drawbacks, and lessons learned. In *International Conference on Persuasive Technology*, 162–175. Springer.
- [33] John, O. P., Donahue, E. M., & Kentle, R. L. 1991. The big five inventory—versions 4a and 54.

- [34] John, O. P., Naumann, L. P., & Soto, C. J. 2008. Paradigm shift to the integrative big five trait taxonomy. *Handbook of personality: Theory and research*, 3(2), 114–158.
- [35] Benet-Martinez, V. & John, O. P. 1998. Los cinco grandes across cultures and ethnic groups: Multitrait-multimethod analyses of the big five in spanish and english. *Journal of personality and social psychology*, 75(3), 729.
- [36] Rammstedt, B. & John, O. P. 2007. Measuring personality in one minute or less: A 10-item short version of the big five inventory in english and german. *Journal of Research in Personality*, 41(1), 203 – 212. URL: <http://www.sciencedirect.com/science/article/pii/S0092656606000195>, doi:<https://doi.org/10.1016/j.jrp.2006.02.001>.
- [37] Anagnostopoulou, E., Magoutas, B., Bothos, E., Schrammel, J., Orji, R., & Mentzas, G. 2017. Exploring the links between persuasion, personality and mobility types in personalized mobility applications. In *International conference on persuasive technology*, 107–118. Springer.
- [38] Fogg, B. J. 2009. A behavior model for persuasive design. In *Proceedings of the 4th international Conference on Persuasive Technology*, 40. ACM.
- [39] Fortuna, E. & Shack, E. 05 2019. Beyond mobile: Building flutter apps for ios, android, chrome os, and web (google i/o'19). (Visited: 17/05/19 10:45). URL: <https://www.youtube.com/watch?v=IyFZznAk69U>.
- [40] Nystrom, B. March 2019. Making dart a better language for ui. [Online; posted 19-March-2019]. URL: <https://medium.com/dartlang/making-dart-a-better-language-for-ui-f1ccaf9f546c>.
- [41] Kerpelman, T. 05 2019. Cloud firestore data modeling (google i/o'19). (Visited: 19/05/19 17:54). URL: <https://www.youtube.com/watch?v=lW7DwV2jST0>.
- [42] Firebase (Google Developers). Usage and limits. (Visited: 30.05.19). URL: <https://firebase.google.com/docs/firestore/quotas>.
- [43] Kerpelman, T. 05 2018. How do queries work in cloud firestore? | get to know cloud firestore #2. (Visited: 21/05/19 10:06). URL: https://www.youtube.com/watch?v=0fux_4c94FI.
- [44] Firebase (Google Developers). Writing cloud functions. (Visited: 30.05.19 20:33). URL: <https://cloud.google.com/functions/docs/writing/>.
- [45] Firebase (Google Developers). Rest resource: projects.messages. (Visited: 30.05.19 20:08). URL: <https://firebase.google.com/docs/reference/fcm/rest/v1/projects.messages>.
- [46] Mobile Ecosystem Forum. 04 2019. Future of business models panel - mwc 2019. (Visited: 15/05/19 20:11). URL: <https://www.youtube.com/watch?v=3TgsLZscpC4>.

- [47] Forum, M. E. 03 2019. Future of privacy panel - mef at mwc19. (Visited: 15/05/19 22:07). URL: https://www.youtube.com/watch?v=my_ipPdonNk.
- [48] Larche, W. & Robledo, A. 05 2019. Beyond mobile: Material design, adaptable uis, and flutter (google i/o'19). (Visited: 16/05/19 10:39). URL: <https://www.youtube.com/watch?v=YSULAJf6R6M&list=PLjxrf2q8roU2no7yR0rcQSVtwbYyxAGZV>.
- [49] The European Parliament And The Council Of The European Union. 04 2016. General data protection regulation. <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679>.
- [50] Boelens, D. 09 2018. Reactive programming—streams—bloc. [Online; posted 01-09-2018]. URL: <https://medium.com/flutter-community/reactive-programming-streams-bloc-6f0d2bd2d248>.
- [51] Hracek, F. & Sullivan, M. 05 2019. Pragmatic state management in flutter (google i/o'19). (Visited: 17/05/19 11:57). URL: https://www.youtube.com/watch?v=d_m5csmrf7I&list=PLjxrf2q8roU2no7yR0rcQSVtwbYyxAGZV&index=3.
- [52] Ariva, H. 01 2019. How to sign flutter apps for ios automatically without a mac. [Online; posted 31-01-2019]. URL: <https://medium.com/flutter-community/how-to-sign-flutter-apps-for-ios-automatically-without-a-mac-a2dc9cfa5a6c>.
- [53] Purpura, S., Schwanda, V., Williams, K., Stubler, W., & Sengers, P. 2011. Fit4life: the design of a persuasive technology promoting healthy behavior and ideal weight. In *Proceedings of the SIGCHI conference on human factors in computing systems*, 423–432. ACM.
- [54] Alrobai, A., Dogan, H., Phalp, K., & Ali, R. 2018. Building online platforms for peer support groups as a persuasive behavior change technique. In *International Conference on Persuasive Technology*, 70–83. Springer.

