NTNU
Norwegian University of
Science and Technology

Automatic self-evaluation system for novice Python developers

Sindre Haneset Nygård

Master of Science in Computer Science
Submission date: June 2016
Supervisor: Guttorm Sindre, IDI

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

Abstract

Students who enroll in university introductory programming courses often have very different backgrounds. Some have never written a single line of code in their entire life, while others have been programming for years. Having students with so different background can be a challenge for course organizers and teaching assistants. A system that can help the students self-evaluate their own assignments and skill can save a lot of time for the teaching assistants, time they can spend to help students in need. The system should also be able to provide assignments with adjusted difficulty to each individual student. This way, both students with a lot of programming experience as well as novice programmers can get challenging assignments. This project seeks out to uncover the requirements for such a system.

To identify the requirements, a questionnaire was sent out to students attending the object oriented programming course, and a prototype was made and tested by a small test group. A heuristic function to adjust the assignment difficulty by advancing a skill level based on the number of correct answers in a row was tested, and turned out to work very well. This project also looked at what aspects can increase the students motivation to do more assignments. Another element that was investigated was how to make the system as easy as possible to use, to encourage students with less programming experience to do more assignments.

A leveling system, achievements and a system for adaptive assingment difficulty turned out to be a great combination to have students with very different background work on assignments in the same system.

Sammendrag

Studenter som starter i introduksjonsfag på universitetet har ofte en veldig forskjellig bakgrunn. Noen har aldri skrevet en eneste linje kode, mens andre har programmert i flere år. Å ha studenter med så forskjellige bakgrunner kan være en utfordring for både for fagkoordinatorer, så vel som student assistenter. Et system som kan hjelpe en student å evaluere sitt eget kunnskapsnivå, og automatisk rette oppgaver kan frigjøre masse tid for student assistenter som heller kan bruke tiden til å hjelpe studenter som trenger hjelp med en oppgave. Et slikt system kan også tilby oppgaver med vanskelighetsgrad basert på hver enkelts students ferdighetsnivå. Slik kan både studenter som er gode til å programmere og studenter som sliter, få utfordrene oppgaver. Dette prosjektet skal prøve å finne ut hvilke krav som må stilles til et slikt system.

For å identifisere kravene ble en spørreundersøkelse sent ut til studenter som tok faget Object Orientert Programmering. Det ble også laget en prototype som ble testet av en mindre gruppe studenter. Dette prototypen fikk også funksjonalitet som kunne tilpasse vanskelighetsgraden på oppgavene basert på studenten som brukte systemet. Denne funksjonen fungerte veldig bra, og ble godt mottatt. Prosjektet forsøkte også å finne ut hvilke faktorer som kunne bidra til å motivere studentene til å gjøre flere programmeringsoppgaver. Et annet aspekt som ble undersøkt var hvordan systemet kunne bli laget enklest mulig, for ikke å virke skummelt for studenter som ikke er så veldig gode til å programmere.

Et nivåsystem, et system for utdeling av bragder, og tilpassning av vanskelighetsgraden på oppgavene viste seg å være en veldig god kombinasjon for å få studenter med varierende bakgrunnskunnskaper til å kunne gjøre oppgaver i samme system.

Acknowledgement

This work is a report that was done for my masters thesis in the spring of 2016 at the Norwegian University of Science and Technology. I have learned a lot from this project, and I hope that this work will benefit future students attending TDT4110 to learn python programming.

First and foremost I would like to thank my supervisor, Guttorm Sindre, for guiding me through this project. And also a big thanks to everyone who responded to the questionnaire and tested the prototype. I would also like to thank my friends and family for their support throughout my years here at NTNU, especially Marianne Smørgrav.

Trondheim, June 2016

Sindre H. Nygård

Contents

List of Figures

Chapter 1

Introduction

Students who enroll in university introductory programming courses often have very different backgrounds. Some have never written a single line of code in their entire life, while others have been programming for years. Because of this, the skill and knowledge level of the students vary greatly. A system that can motivate the students to do more assignments to increase their programming skills could help mitigate this issue.

1.1 Background

Every study program at the Norwegian University of Science and Technology has a list of compulsory classes that the students have to complete in order to graduate from their respective program. One class that often appear on the lists the technology programs are Introduction to Information Technology, TDT4110. Usually the students take the class in their first semester and therefore have very different background knowledge when it comes to programming. Some students have taken programming classes in high school, some have been programming as a hobby for many year or gained programming skills or experience through work or other activities. However, there may be students taking this class who have never written a single line of code before attending this class.

Students attending programs that closely relate to programming, such as Computer Science and Informatics, might have a high motivation to gain good programming skills. While students in some other programs might feel that the course is just a compulsory course they have to get through and their interest and attention really lies elsewhere.

TDT4110 has a set of compulsory exercises where the students need to solve half of the tasks for each exercise. This system works well for average and stronger students who can debug and finish the tasks, but the weaker students often end up stuck without making much progress while waiting for help from a teaching assistant. This can potentially be demotivating and the student can end up copying the tasks from a friend without learning very much or anything at all. The the following assignment can end up even harder to finish, because the student may not have learned the concepts that was needed to progress further in the course.

1.2 Goal

Having presented the problems with students having diverse background skills in programming, can a system to mitigate this difference be created? What should the requirements for such a system be? This project seeks to establish a set of requirements for an informal, low-threshold system that can help students with any kind of programming experience. For students with no experience at all to students with many years of experience, this system seeks improve their skills and provide feedback on what topics they need to work on. The system should motivate and encourage the student to continue to work on and improve their programming skills. At the same time, it should be secure from malicious malicious users who seeks to destroy or compromise the system.

By using gamification, can the system motivate students to do programming assignments more often? By giving assignments that are adjusted to the skill level of the individual student, can the system encourage students to do more assignments?

Through questionnaires and the development of prototypes, this report will attempt to answer the following research questions:

1. RQ1: What are the requirements for a system that allows a student to self-evaluate their programming skills and get feedback on what topics they need to improve?

2. RQ2: How can a self-assessment system provide a student with programming tasks that fit the individual students skill level?

1.3 Related work

There exists a number of web sites and systems that are designed to teach python, and computer programming in general. This section will look at how a few of these sites and systems are designed, what their concepts are and how they differ from this projects system.

1.3.1 QuizPack/QuizGuide

The University of Pittsburgh provides quizzes on various topics in their programming courses. In 2001, the university introduced QuizPACK. This is a system that automatically generates questions from parameterized templates [1].

The templates allows the system to output a large number for exercise questions for self-assessment of programming knowledge. Evaluations of QuizPACK showed that the use of this self-assessment system was a really powerful learning tool that increased the programming skill of the students [1]. Evaluation of QuizPACK also showed that students were very positive to the use of parameterized questions and that it could improve the students programming skill if QuizPACK is used on a regular basis [2].

In 2004, Brusilovsky et al. created an extension to QuizPACK, QuizGuide, which uses adaptive annotation to help guide the students to quizzes about relevant topics or areas that are important and requires further work [3]. Adaptive annotation augments the links with dynamic comments, in contrast to adaptive ordering, which orders a list of link by putting more relevant links closer to the top [4]. QuizGuide proved to be a valuable extension of QuizPACK as the students completed more questions in a larger variety of topics, which resulted in an increase of the students knowledge at the end of the course [3].

1.3.2 Codecademy

Codecademy offers complete courses in many languages and a variety of other technologies and frameworks. From JavaScript and Python, to Structured Query Language (SQL) and Git [5]. Codecademy has a very course like approach, where you solve a set list of problems in a linear manner. They give you an occasional quiz where you can test yourself to see how well you do. They also offer a pro package where you can pay a monthly subscription to get access to exclusive quizzes, lessons and projects [6].

While Codecademy focuses more on the complete course experience, with lessons accompanied by exercises, it does not adjust the assignments and quizzes given to the skill level of the student. The system described and developed in this project aims to complement an existing lesson and exercise course. It does so by offering dynamic assignments that are adjusted to the students skill level.

1.3.3 Code wars

Another web site that aims to increase a programmers skill is Code wars [7]. Code Wars is probably one of the more similar web sites to this projects system. The main focus is make users do assignments, or “Katas”. Kata is a Japanese word for a system of individual training exercises used in martial arts [8]. As the user completes katas, the user earns ranks and gain honors. The whole site is community driven and free. All the katas are created and evaluated by the community [9]. Each kata has a forum where users can ask for help and hints on how to solve that specific kata. After the user solves the kata, the user gets access to see how other users solved that kata.

There are a lot of similarities between Code wars and this system. However, there are a few key differences. Code wars does not automatically determine the users skill level, so the users has to find the right level of difficulty on their own for each topic. Achievements is also a feature that Code wars does not have.

1.3.4 Project Euler

A web site that caters more to people interested in math, but still offers a lot of programming exercises is Project Euler. Their problems range in difficulty, but are not annotated with a difficulty level [10]. The only way to get a feeling on the difficulty is to look at how many have solved that problem, compared to other problems.

1.3.5 Kattis

Kattis is an automated assessment system developed at KTH Royal Institute of Technology in Stockholm, Sweden. Initially it was designed to automatically assess programming exercises and save teachers and teaching assistants hours or work, so they could focus on other more important tasks. It was first used in the Programming and Problem Solving Under Pressure (Popup) course in 2005 where the students are solving a large amount of programming problems. By automating the assessment, the program verification could be done more thorough and efficient. In 2006 it was introduced to the Advanced Algorithm course and was extended to provide more advanced feedback to the students [11].

After years of experience and feedback from students, Kattis is now very flexible and can be used for many purposes where assessing programs is necessary. It is used in major international programming contests like the ACM International Collegiate Programming Contest, as well as being used for recruiters to assess the programming skill of potential new employees [11, 12]. It also allows for anyone to sign up and solve problems from past competitions. It features a high score and ranking system, and can hook up users with potential employers [13].

Enström et al. observes that while 80% of the students attending courses where Kattis is used had a positive attitude towards it, some students criticized Kattis for not providing enough feedback. However, they are claiming that the underlying reason for this is that students and teachers have a different view on what the requirements for correctness in a program is [11]. While students are willing to accept that some edge cases are treated incorrectly, the teacher believes that all legal input to a program should be treated correctly [14]. Therefore students blame the feedback from the system when their programs fail

[11].

Kattis also introduces an element of gamification from the fact that it times the program submissions and ranks the students based on this time it takes their program to complete. This can cause the students to enter a “Nintendo mode” where they will try to outperform themselves or other students [15]. However, this causes the students to spend more time optimizing and improving their program, which is an important factor when learning [16].

Chapter 2

Method

This chapter will describe the process of collecting the data needed to answer the research questions presented in chapter 1. One research methodology that is commonly used when doing research in information systems is the Design Science. Section 2.1 will describe the idea and the concept behind design science and the reason why this methodology is a good fit for this project. Section 2.2 will look at the process of gathering the data used to support claims and conclusions made in this report.

2.1 Design Science

Technology is often a result of intelligent design, not just a random occurring natural phenomenon. Usually it solves a problem or support and in human activities [17]. The key to understanding Information Systems (IS) and their implications is to realize that an IS is a man-made object, rather than being a natural occurring phenomenon [17]. Design science is an idea that was introduced in Science of the Artificial by Simon [18, 19]. The traditional research methodology in natural science seeks to understand and explain how a phenomenon works and understand reality. However, in the design science research methodology the scientists both create and study the artifact [17].

Design science research methodology has two steps: construction and evaluation [19]. The process is often repeated, where the artifact created in the construction step is improved after the evaluation step and can be evaluated again [19]. This iterative approach makes it possible to fine-tune the artifact to solve the problem in the best possible way.

The construction phase is usually initiated by some problem that needs a solution. The initial design of the artifact can be based on knowledge that was previously obtained. The activities involved in the creation of this artifact depends on what kind of artifact is being constructed. For example an algorithm could require a mathematical proof that it is sound and valid, or an IS would require some software development activity [20]. After the construction phase is completed, the artifact is tested and evaluated on how well it solves the problem that initiated the construction in the beginning. For an IS, this can for example be to test the prototype created on actual users or in the intended operating environment.

Based on this, design science research seems to be a suitable method for answering the questions that were stated in the introduction chapter. Chapter 3 will highlight the construction step, while chapter 4 will cover the evaluation step of the methodology.

2.2 Data collection

To collect the data needed to evaluate the artifact created for this project, a mix of quantitative and qualitative research was conducted. Quantitative research considers a large sample. The data are usually on a numerical format. The greatest advantage for this method is that it can target a broad representative sample [21]. The opposition to quantitative research is qualitative research. This method considers a smaller sample target, but generates more in depths data for each of the data entities [22].

First a quantitative questionnaire with simple multiple choice questions was sent out to a large group of potential users of the system. This questionnaire will be discussed in section 2.2.1. After the prototype was created, a small group of students tested the prototype while being observed, and and was interviewed after the prototype testing to get feedback of their view of the prototype. The feedback provided by the prototype testing was used to make improvements. The new prototype was then tested in the same manner as the first one. The prototype testing will be described and discussed in section 2.2.2.

2.2.1 First questionnaire

The first questionnaire was a simple multiple choice form that was sent out to get a reference point on what features and functionality was important to focus on during the initial development cycle.

As the final product of this project is aimed at novice python developers, it was important to reach out to students who recently learned or was in the process of learning python programming. Students attending the course Introduction to Information Technology, TDT4110, would be a natural target for this survey. Because this introductory class offers an introduction to python programming. However, this course is only offered in the fall semester, while this survey took place during the spring semester. It turns out that a lot of the students attending this introductory class progress into the Object-oriented programming course that the university offers in the spring semester, TDT4100. Therefore, this course was selected as the main source of respondents. The survey was also published through other channels, including social media. The object-oriented programming course turn out to be the main source for recruitment for the survey. As the goal of the survey was to get as much input as possible, the questionnaire was designed to be easy to answer and take about 2-3 minutes to complete.

When learning a new language, or any other skill, the key to proficiency is repetition and usage. Practice makes the master. This has also been shown to be the case in other systems designed to teach and improve knowledge about computer programming, such as in QuizPACK and Kattis [1, 11]. Therefore, a lot of the questions in the survey examined how to make students do more assignments more often. After a period of one week, the questionnaire got almost 130 responses with data that was very valuable for the initial development of the system. The results of the questionnaire will be further discussed in Chapter 4. The questionnaire is attached in its original language, Norwegian, in appendix B.1.

2.2.2 Prototype testing

After a working prototype was created it was time for prototype testing. The purpose of the prototype testing was to get feedback on what the implemented features needed to improve and who the user experience of the system was. Two rounds of prototype testing was conducted. One for each iteration of the prototype.

The initial questionnaire discussed in section 2.2.1 also included an optional field at the end, where students interested in participating in prototype testing of the system could leave their contact information. All of the prototype testers for the first prototype test phase were recruited from this group. The potential testers were contacted individually, and a time and place for the testing was agreed upon. All of the prototype tests in this round were conducted over one week. During the prototype testing the test person was observed while using the system for 30 minutes. After the test was completed the tester was interviewed about the experience of using the system to get some final feedback that might not have been uncovered during the test.

After analyzing the feedback from the first prototype test phase, a new iteration of the system was developed. As most of the feedback on the first prototype was concerning the graphical user interface, this new version focused on improving this, rather than introduce new features, but with a few exceptions. This new iteration of the system was prototype tested in the same way as the initial version. However, due to lack of response from the testers that signed up with the initial questionnaire, the testers for this second prototype was mainly recruited from personal networks, therefore the two test groups did not overlap.

To make the prototype testing more realistic, it was important that the testers had access to realistic assignments. Therefore 50 assignments were created for the first test round. They were a mix three different topics: conditional control structures, iterative control structures and functions. They were also mixed into three different difficulty levels. The easiest assignments were on the “fill in the missing parts” format. Listing 1 shows an easy sample assignment that was created for the testing, with the assignment text “*Complete the for loop to print all the elements in `lst`*”.

```python
lst = [6, 5, 4]
for item in lst:
    # Write your code here
```

Listing 1: Print all the elements in `lst`

The next level assignment for loop structure could for example be to construct the entire loop. Listing 2 shows how a medium sample assignment could

```
lst = ['blue', 'red', 'yellow']
# Write your code here
```

Listing 2: Print all the elements in `lst`

look like. A harder assignment example was the assignment "*Create a function f that prints all even numbers less than argument num*". Listing 3 shows the code provided for this assignment.

```
def f(num):
    # Your code here

f(10)
```

Listing 3: Print all even numbers less than the argument

There are no limit to how hard the assignments could have been made, but the time estimation given for the prototyping suggested that not many, if any at all, would reach a higher level. After feedback from the first prototype testing, the levels were made shorter, therefore a few of the testers did reach the top level before the testing was completed during the second prototype testing.

The first prototype testing received some feedback that the system had too few assignments, and that the same assignments kept showing up. To fix this for the second prototype testing, an additional 30 assignments were created. In the end, the system was tested by 9 different people. The results of both the prototype tests will be discussed in Chapter 4.

Chapter 3

Development and technologies

This chapter will discuss the development process, challenges that arose during the development and what technologies and frameworks that were used to get the prototype working. Section 3.1 will discuss the development process for the project. Section 3.2 will look at what technologies and frameworks were used for this system, and why they were chosen. At the end of this chapter section 3.3 will look at the challenges that arose during the development of this system.

3.1 Development

To build a prototype for this system, an iterative approach was selected. This allows for a prototype to be constructed. After completion of the this prototype, it can be tested and evaluated. Insight gathered from this testing can then be used to improve the old prototype by improving the user experience, fix bugs and implement newly discovered requirements. This development technique fits well with the design science research method discussed in section 2.1. While the code was checked in to the code revision system git, no effort was put into following some git development methodology such as GitFlow [23]. This decision was made, because the development team was only a single developer and there was no requirement for a stable production environment for the prototypes.

3.2 Technologies

As will be discussed in chapter 5, the system will be a web based application. This decision was made to ensure the availability across multiple platforms without needing to install any additional software. There are a lot of frameworks available for all kinds of programming languages to get a web applications up and running in a short time [24].

When starting a new software project, it is important to consider what programming language and technologies that can solve the problem in the best possible way. Python is a programming language with a lot of available libraries and a large collection of available web application frameworks [25]. Python has also proved itself to be reliable and capable of being used as the main framework in a lot of major web services like YouTube.com, Instagram.com and reddit.com [26, 27]. In addition to this, the system is, after all, about learning python programming, and the developer of the system already had extensive background knowledge and experience with python programming. Therefore, python was chosen as the programming language to use for the back end part of this systems.

Another consideration to make was what technology should form the basis for the front end part of the system, the part of the system that the user is actually interacting with. JavaScript is the most widely supported browser programming language and is supported by all major modern web browsers [28]. Just as with Python, JavaScript has a lot of available frameworks and libraries that offers rich functionality. Therefore, JavaScript was selected as the main programming language for the front end, along with HyperText Markup Language (HTML) and Cascading Style Sheets (CSS) for markup and styling.

Choosing a good framework can help the developer greatly by abstracting away tasks that have already been solved by other developers. A use case where this is especially useful is data models and interacting with database management systems. Unless the project has very specific needs, which this project does not have, a good framework for database management can make the system agnostic to what database is used for storing the data models and information. Therefore, as discussed in section 3.2.1, there is no need to spend much time selecting a database. PostgreSQL is an open source object-relational SQL database system that is reliable and easy to set up, therefore it was used as the database during the prototype testing [29]. However, it can easily be changed out with other alternatives like MySQL, Oracle and more.

The following sections will consider different frameworks and justify why they were chosen above other alternatives.

3.2.1 Django

As discussed earlier, Python has a lot of frameworks for web development available. These frameworks are often divided into two categories, microframework and macroframework. A microframework comes with only the most essential functionality. For a web framework, this is often only the bare minimum needed to get the service up and running. Usually this types of frameworks also include a router. A router is used to determine what functions should be called for each Uniform Resource Locator (URL). For any other desired functionality, the developer has to come up with a solution on his own. This can either be implement it, or to find a library with the features already implemented.

One of the most commonly used web microframeworks for Python is Flask. In addition to the router, Flask also includes a templating framework for creating dynamic HTML markup [30]. One thing that Flask, and most other microframeworks, does not have is an Object-Relational Mapper (ORM). An ORM can abstract away all the direct interaction that the developer might have to do with a relational database. The ORM will act as a middleman between the database and the developer, and the developer only needs to interact with objects instead of database queries. For example, a relational database can only store information in tables. Often an object consists of data from multiple tables. The ORM will get the data from the various tables and put them into an object that the developer has defined. A very popular ORM framework that is often used together with Flask is SQLAlchemy. SQLAlchemy is database agnostic, which means that it offers support for a wide variety of databases that can be swapped out with little to no modification of the code [31]. The advantage of not having all the bells and whistles in a framework is that the developer can leave out elements that are not really needed, so that the code becomes faster and less complex. It is also a lot easier to learn how the framework works, because it is much smaller.

The alternative to a microframework is a macroframework. This is a complete package that includes everything that is usually needed, from router and ORM, to a template framework and form generators, in addition to much more.

One such framework is Django. It is a very commonly used framework, and is being used by major software companies world wide, including Instagram.com, Pintrest.com and NationalGeographic.com.

As it is a macroframework, it does come with a lot of functionality out of the box, like user authentication, an ORM system, a template system for HTML and a lot of security features and mechanisms [32]. In addition to this, it can generate forms based on models and if has a very powerful admin panel that easily lets site administrators add, view, change and delete data stored in the models, without writing any code other than the models themselves. The ORM that is included with Django is, as SQLAlchemy, database agnostic, which means that the underlying database can easily be swapped out, and it comes with build in protection against SQL injections.

Because Django comes as a complete package, and most of its features are useful and likely to be used in this project, this would be the chosen web application framework for this system.

3.2.2 Brython

To verify that the code is correct, or that it even runs, it is necessary to run and test the code. As this is a web application, there are generally 2 ways of running the code, server side or client side. The most reliable way, that would provide the easiest methods to run reliable testing, is to verify the program server side. However running user generated scripts on the server could potentially pose a major security risks.

Code wars, as discussed in section 1.3.3, solves this problem by running each script in its own Docker container [33]. Docker wraps a complete file system in a container that shares the same operating system kernel, but have similar resource isolation as regular virtual machines [34]. This allows each container to offer the same security as virtual machines in terms of resource isolation, but has the ability to start in seconds. This also makes sure the testing environment always stays the same. Using this technique makes it possible to do more extensive testing of the scripts to make sure they are valid. However it also requires more resources and takes a little longer to run the tests.

The other way of verifying the code is to run it client side. In terms of security this is lower risk as the program is only run in the client browser. However, running python directly in the browser is not a trivial task. It is possible to do by using browser plugins like Microsoft Silverlight, but this plugin is only available for Windows and OS X and the open source alternative for Linux is no longer being maintained [35, 36]. Another possibility is to transpile the Python script into JavaScript, or ECMAScript 5 which is the formal basis for JavaScript. JavaScript is supported by all modern browsers and requires no additional plugins that the user must install and maintain [28]. There exists multiple transplilers from Python to JavaScript like Transcrypt, Skulpt or Brython. Transcrypt has to run on the server side so it involves the risk of having the code run on server side [37]. Skulpt transpiles on the fly and runs in the browser, however it only transpiles Python 2.x [38]. Brython offers much of the same functionality as Skulpt, but it transpiles Python 3 on the fly and runs it in the browser, which satisfies the requirements to run the python code in the browser without having the user to install and maintain additional plugins [39].

Brython aims to implement all the functionality that Python offers. However, there are some limitations that must be considered when running Python code in the browser. In particular file input/output. Brython implements reading files by using Ajax to load files from the server, but it does not implement writing to files [40]. For the use case of this project it is possible to implement the write file functionality by sending a POST request to the server who can then check if the content is as expected.

3.2.3 Other libraries

In addition to the previous libraries discussed earlier, a number of frameworks were used in a lesser degree, to support the system. This section briefly discuss some of them.

Bootstrap

When building a new web site from scratch a framework that can provide basic HTML and CSS pattern can help speed up the development process. There exists many frameworks that can aid the developer with this. Two very popular frameworks are Bootstrap and ZURB Foundation. Bootstrap was created by engineers at Twitter to make it easier to maintain their internal tools and to speed up the process of making prototypes. ZURB Foundation was released by the web design company ZURB. They are both open source and was initially released within a month of each other in 2011. They do offer more or less the same functionality, like an easy to setup grid system for element placement, navigational bars, responsive elements to handle various screen sizes and much more [41, 42]. They also ensure the same behavior across all commonly used modern browsers and scale the content on the web page to fit into any kind of screen, from mobile phones to large desktop screens.

In the end, for most use cases, the decision of which is the better comes down to personal experience and preferences. As the developer for this project already has extensive knowledge about Bootstrap, it was chosen as the design framework for this project.

jQuery

jQuery is a utility framework for JavaScript. It helps the developer deal with anything from manipulating Document Object Models (DOM) in HTML, and handling events and asynchronous JavaScript and XML (AJAX) calls [43]. There are not that any JavaScript utility frameworks that can compete with jQuery when it comes to usage and popularity [44]. As jQuery is also a requirement for using Bootstrap it was an easy choice to use jQuery as the main JavaScript utility framework.

Ace

The code editor is perhaps one of the most important elements in the whole system. Especially since the main target users for this system is novice programmers, it is important to have an editor that is easy to use and not too complex. There are a few in browser JavaScript implemented code editors available. Two of the most popular and widely used browser code editors are Ace and CodeMirror. They both provide much of the same features and are both open source. They offer syntax highlighting, auto indenting, which is very important when programming python, line numbering and other features that are commonly used in code editors [45, 46].

As they are both more or less equal, from a users perspective, the choice for which one to use came down to which one was easier to implement and configure. Ace worked perfectly out of the box, and the only setting was to make it use python highlighting and indentation. Ace is, among other web sites, used in the Github browser editor, and in Codecademy discussed in section 1.3.2 [47].

3.3 Challenges

As with any system development projects, some challenges always arise. This section will discuss a few challenges that arose and how they were dealt with.

Achievement system

One part of the system that was surprisingly tricky to implement was the achievements feature. One of the problems was to come up with a system for verifying whether or nor not a user should unlock a new achievement, that was as generic as possible. Another challenge was to know or foresee what statistical data or triggers that would be needed.

In the end, this project was not about creating the achievements system. Therefore, the solution was, although not pretty from an architectural level, to use a combination of creating method plugins for each achievement type, and a fixture with the database objects corresponding to each of the achievement verification methods. The other problem was solved by logging anything and everything that could be logged.

Brython file IO

Initially Brython had some problems with the python keyword "`with`". However, this turned out to be a known bug in the framework and was solved by upgrading to a newer version of Brython, available from their Github.com repository.

Another challenge with Brython, as discussed in section 3.2.2, is that because it only runs in the browser, it has no access to a regular file system where it can read and write from files. For assignments that required the user to read from files, this issue was solved by having a few static default files served from the web server, and the path to these files were given in the assignment text. This was possible because Brython implements an AJAX call to this location whenever it is instructed to read from a file. The more tricky problem was writing to files. Brython does not implement this behavior, but a proof of concept prototype was developed and briefly tested to send a POST request back to the server with the written data. The server then responded with an HTTP 200 OK Response. The data was discarded and no validation was done. No assignments given during the prototype testing involved writing to files.

Chapter 4

Results

The questionnaire and prototype testing yielded a lot of interesting results. This chapter will give a brief summary of the data that was gathered for this project. Section 4.1 will review the results from the first questionnaire that was described in section 2.2.1. Section 4.2 will review the results from both prototype testing sessions that was described in section 2.2.2. Chapter 5 will provide a more in depth discussion on the more interesting findings and their implications.

4.1 First questionnaire

As discussed in section 2.2.1, the main target group of this survey was students who attended the TDT4110 Introduction to Information Technology course. However, this was not an absolute requirement. Of the 125 respondents to the questionnaire, only one person had not attended that course. Of those 124 respondents, 54.8% had no programming experience prior to attending the TDT4110 course.

Question 1B asked the respondents, who had already said they had at least some degree of programming experience, how much programming experience

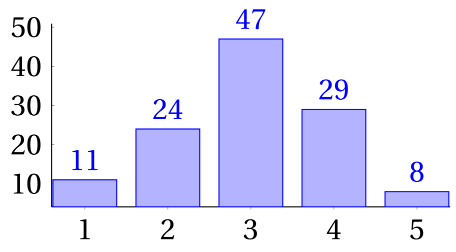


Figure 4.1: Question 1B: How much programming experience to you have?

they personally felt they had. They were given a scale from 1 to 5, where 1 was little and 5 was much experience. The survey was conducted after they had completed TDT4110, so all respondents would have at least some experience with programming. As can be seen from figure 4.1 the response had quite a normal distribution with only a slight overweight of respondents saying they had more experience.

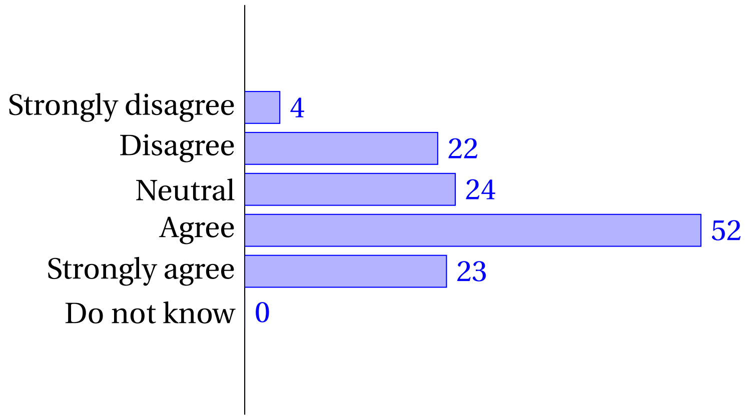


Figure 4.2: Question 3A: I am good at programming

In question 3A, the survey asked the respondents how much they agreed to the statement "*I am good at programming*". This question was almost the same as question 1A, however, stating a question in a different way can yield different results. From figure 4.2 it can be seen that the distribution is not entirely the same as question 1A from figure 4.1. The respondents without previous experience actually rated themselves on average 0.5 points higher, if converting the alternative “Strongly disagree” to a score or 1, and “Strongly agree” to 5, while those respondents with previous experience rated themselves on average 0.66 points higher.

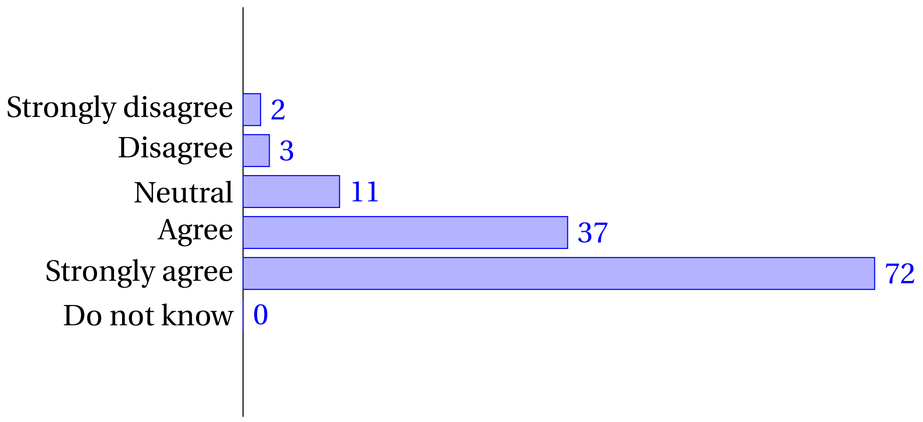


Figure 4.3: Question 3B: I would to more assignments if the difficulty are adjusted to my skill level

Question 3B examined the effect of having the difficulty level of each assignment adjust to the approximate skill level of the student currently solving that assignment. It asked the respondents how much the agreed to the statement “*I would do more assignments if the difficulty are adjusted to my skill level*”. Figure 4.3 shows the results of this question, and more than 87% say they would do more assignments if the difficulty level of the assignment is adjusted for their individual skill level.

Figure 4.4 shows a summary of question 3C, which asked how much the respondents agreed to the statement “*I would do assignments more often if they are short*”. In total 68% answered that they were more likely to attempt assignments more often if they were short. The idea behind this question is that the commitment to do shorter assignments are smaller, thus lowering the threshold for doing programming assignments in between other activities.

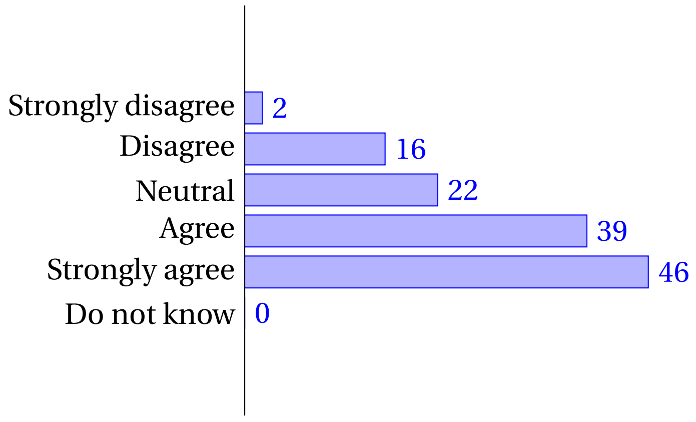


Figure 4.4: Question 3C: I would do assignments more often if they are short

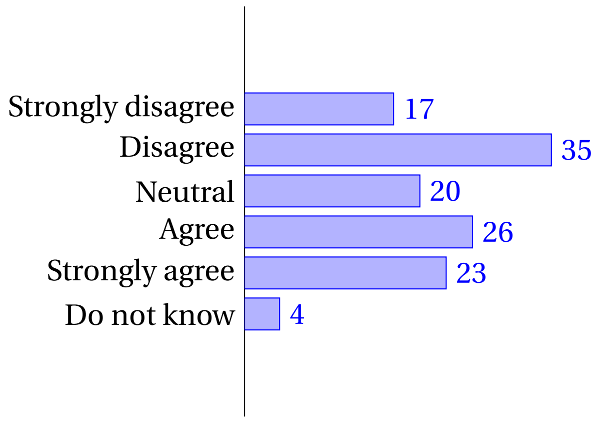


Figure 4.5: Question 3D: I would do more assignments, if they are available on mobile platforms

Question 3D was the statement "*I would do more assignments when I have extra time, for example when riding the bus, if the assignments are available on mobile platforms*". The intention was to figure out if there was a need for mobile support. In contrast to the two previous questions, the answers to this question were more mixed. Figure 4.5 shows that there is no common trend or agreement among the respondents, with 41.6% disagreed to the statement and 39.2% agreed to the statement.

Question 4 asked the students about their view on how they felt the best

17 (13.6%)	Short assignments
10 (8%)	Long assignments
69 (55.2%)	Mix of short and long assignments
29 (23.2%)	User chooses long or short assignments

Table 4.1: Question 4: Best way to learn programming is to do

way to learn programming was for them. "*The best way to learn programming is to do:*". Table 4.1 shows the results from this question. Not surprisingly, more than 75% of the students suggested that a mix of short and long assignments was the optimal way to learn programming.

9 (7.2%)	only a single topic of my own choosing
40 (32%)	a random mix of topics
41 (32.8%)	to choose one or more topics I want to improve
35 (28%)	the system to choose the topics I need to improve

Table 4.2: Question 5: When I do assignments I want

The purpose of question 5 was to figure out if the students wanted to concentrate their effort on a single topic at a time, or if a mix of multiple topics mixed into the quiz was more desirable. The statement was "*When I do assignments, I want...:* Most of the students, 92.8%, wanted some kind of mix. However, they were almost evenly split into three groups in the view of how the topics should be mixed. Table 4.2 lists the responses to the different options.

The next series of questions was about motivation, and how to motivate students to attempt more assignments. This was important to know, because the amount of effort put into problem solving, the greater the educational value will be [16]. The respondents were asked to rate how much they agreed with the following statements.

The statement in question 6A was "*I am more motivated to do more assignments if I can compete with myself by beating my old score*". Figure 4.6 shows that the majority of the respondents would be more motivated by a scoring sys-

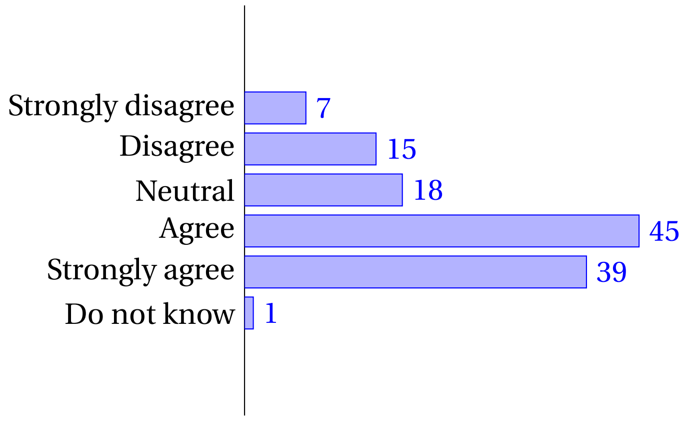


Figure 4.6: Question 6A: I get more motivated if I can compete with myself

tem where the student could compete against him or herself.

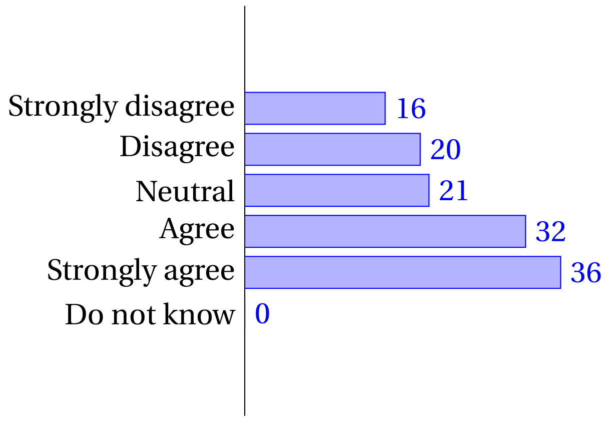


Figure 4.7: Question 6B: I get more motivated if I can compete against other students

Question 6B was in the same category as 6A, but instead of competing against one self, the question was focused around competing against other students. *"I am more motivated to do more assignments if I can compete with my friends".* The respondents were also motivated to compete against their friends, although more students liked the idea of competing against one self. Figure 4.7 shows the answers for this questions.

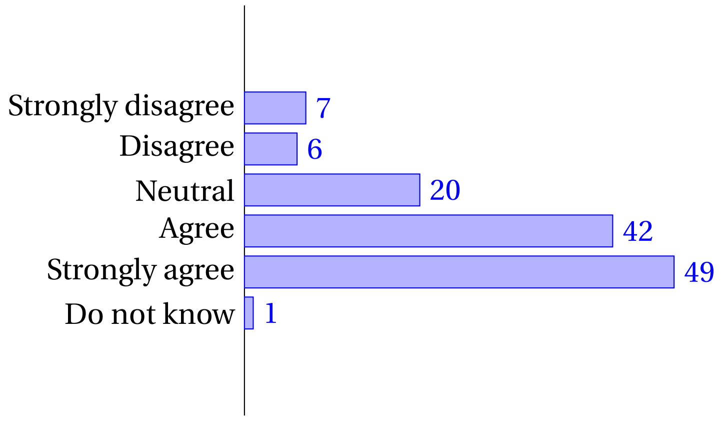


Figure 4.8: Question 6C: I am more motivated to do more assignments if I can earn badges and achievements

Question 6C asked the respondents how much they agreed to the statement *"I am more motivated to do more assignments if I can earn badges and achievements"*. This turned out to be the biggest source of motivation of the suggested methods. Figure 4.8 shows that more than 70% of the respondents would be more motivated by earning achievements and badges.

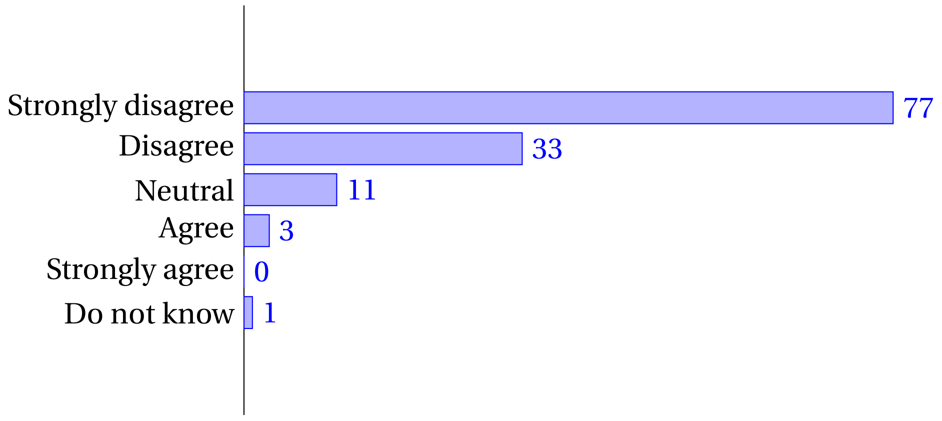


Figure 4.9: Question 6D: I am more motivated to do more assignments if I can share my progress and score on social media platforms

The last motivational question was 6D. It had the statement *"I am more motivated to do more assignments if I can share my progress and score on so-*

cial media platforms". A staggering 88% of the respondents disagreed with the statement, and more than 60% of all the students strongly disagreed. Figure 4.9 shows a clear agreement among the respondents that they do not get motivated by integrating the system with social platforms.

The last question, question 7, in the survey was an open ended question: *"I have other suggestions to a system for adaptive learning of (python) programming"*. Some people suggested that the system could be a competitor to other existing systems created to teach programming, like Codecademy and Project Euler, that were both discussed in section 1.1. Others suggested that the system should lead the users in a direction where the final product could be some larger system or a game. As mentioned in section 2.2.1, the target group for the survey was students attending the Object Oriented Programming Course at NTNU. That course has a game development project, which might have been an inspiration to some of the respondents. A number of respondents also supported the idea of short assignments and achievements. A wiki or links to references was also suggested. Another thing that was pointed out was the frustration of being stuck on an assignment for a very long time, without getting anywhere nearer of solving the assignment. This comment had a very strong support of the systems ability to adjust the difficulty level the assignments to the skill level of the user.

In the end, the survey invited respondents to leave their contact information to be contacted about prototype testing of the system. Quite many signed up as prototype testers, but as will be revealed later, in section 4.2.1, only a few volunteered when the prototype was ready for testing.

4.2 Prototype testing

As mentioned in section 2.2.2, both of the 2 prototype versions was tested. First the initial version was tested by a group of students from the questionnaire. After being improved based on the feedback provided during the first test, the second version was tested by a another group of testers. Both tests provided a lot of valuable data for this project.

4.2.1 First prototype testing

Despite a lot of prototype testing sign ups from the questionnaire, in the end only 4 students agreed to test the prototype. The testers had various level of programming experience, from barely any to a few years. This was very good, because it gave the possibility to test how the difficulty level would adjust to the different testers. Perhaps the most important discovery was that all of them learned something new from testing the system.

A very popular feature was the achievements. All the testers agreed that this was a fun feature that added excitement to the system. Some of the testers even cheered when they got an achievement. Another positive point the testers made was that there was so easy to get started. There was no need to install any applications or setup any Integrated Development Environment (IDE). It was simply just to log in and start writing code.

One problem that became apparent was after the user ran the code, it the code did not have the right solution. The user would start hitting backspace to erase the code that caused the problem. However, the focus was not automatically put back to the code input field, so instead, the user was sent back to the previous page. Another issue that was uncovered was that each level was too long. Therefore, heuristic function for level up needed some adjustments. A feature that some of the testers requested was the ability to reset the code that was provided for the assignment. If they changed to provided code and made any errors, they wanted the ability to change the code back to its original state.

As the system was going to provide a dynamic difficulty setting for each user, the each of the assignments has a difficulty level. The original though was not to disclose this level number to the users, however it was displayed as a tiny digit in the corner of the screen for debugging purposes. Several of the testers picked up this number and suggested that it should be more visible as they like to see how they progressed and saw it as a source of motivation.

In the end, the testers were very satisfied with the system and thought the idea of such a system was very good, and could add great value to the TDT4110 course. One of the testers even asked to get log in credentials for the system to continue the work later that day, after the prototype testing was completed. This tester only made it half way through the first level during the test session. The day after, the tester reported that all the assignments were completed.

4.2.2 Second prototype testing

Because the response from the questionnaire respondents were so low, the prototype testers for the second prototype testing phase was recruited from personal networks.

The achievement feature was still a very popular feature in the second prototype testing phase. The testers were also satisfied with the adjusted heuristics for gaining levels.

One thing that was addressed during the second prototype test phase was that the hints could be more visible. Especially if the user seemed to be stuck on a problem. This could be detected by for example counting the number of times the user tried to run the code. A way to make the hints more visible was suggested to make it pop out if the user ran the code with errors 4 times. Or the system could scroll down to the hint and highlight it.

The testers did not get the opportunity to spend much time trying out the new feature of adaptive annotation for selecting what topics the quiz should include. The reason for this was that the test session was only about 30 minutes, and the testers spent most of the time solving problems. However they liked the colored frame on the topics that were under their average level, and though it was a nice way to highlight topics that could require further work.

Chapter 5

Discussion

What would be the requirements for a self-evaluation system where students can get feedback on what programming topics they need to work more on? This chapter will discuss the reasons for how and why the system was made and what impacts the prototype testing and questionnaires had on discovering the requirements for such a system.

Section 5.1 will discuss some general ideas and internal mechanisms and functions, and reasons why they were implemented the way they were. Section 5.2 will discuss the user interface and experience. The discussions will be based on the results from the questionnaires and prototype testing that was conducted for this system.

5.1 General

From the questionnaire, it was clear that there were a mix of students having programming experience before attending TDT4110 and students who did not have any experience. There was a slight overweight of those having no computer programming experience at all. When there is such a big spread of the students skill level, it can be hard to provide good assignments that are challenging for everyone. Not too easy for some, yet not too hard for others. This can suggest that there is a need for a system that can provide challenging assignments to both students with extensive programming knowledge as well as to students who are completely new to computer programming. This was also supported in the questionnaire. Only 4% of the respondents said that they would not do any more assignments even if the difficulty level was adjusted to their own skill level. It was also noted in the open ended question at the end, how frustrating it is to be stuck on an assignment because it is too hard, with no option to progress to the next assignment.

As discussed, an important feature of the system is to be able to provide each student with assignments that have a difficulty level that is challenging enough without being impossible to solve. A heuristic that was implemented to solve this problem was to count the number of correct consecutive submissions in a row. Whenever a give number of correct consecutive submissions was registered, the system would increase the assignment level of that specific topic. For the first prototype testing, the number was five correct submissions in a row. However, feedback suggested that the levels were too long. Therefore, this number was lowered to three in the second prototype test. This seemed to be a better number, and the testers in the second prototype test phase was satisfied with the adjusted heuristic.

The prototype testing only focused on each assignment having only a single topic, because the assignments were only at basic level. However, as the assignments get harder and more complex, the assignments are more likely to incorporate multiple topics at the same time. The heuristic function for determining the skill level should be able to handle this and award skill levels for other included topics as well. A thing that needs to be considered is how to qualify a user for a given assignment with multiple topics. The difficulty for the various topics within a single assignment may vary.

The issue with multiple topics within the same assignment is also relevant to consider if the system shall accommodate longer assignments. The prototype testing only had short topics, but there is nothing in the way of having longer assignments as well. The questionnaire suggested that the students wanted a mix of long and short assignments.

Many systems that are created today integrates with one or more social media platforms. There are a few use cases where this system could benefit from integrating with a social media platform. One possible use case could be the ability to ask for help. If the user is stuck on a problem, it could make a post to the network, asking for assistance. It could ask the network for tips and tricks on how to improve the solution for an assignment. Another use case could be to showcase achievements that the user earns, or to challenge others to solve a given assignment. However, the questionnaire made it clear that integration with a social network is not something that should be prioritized.

5.2 The design and User Experience

The most important part of the system may be how the user experiences the system. The better the experience is, the more time the students will spend using the system. Because this is a learning system, the more time they spend using the system, the more they learn.

5.2.1 Design

After the user logs in, the home view is a dashboard. The dashboard offers personal statistics as well as options for configuring the next quiz.

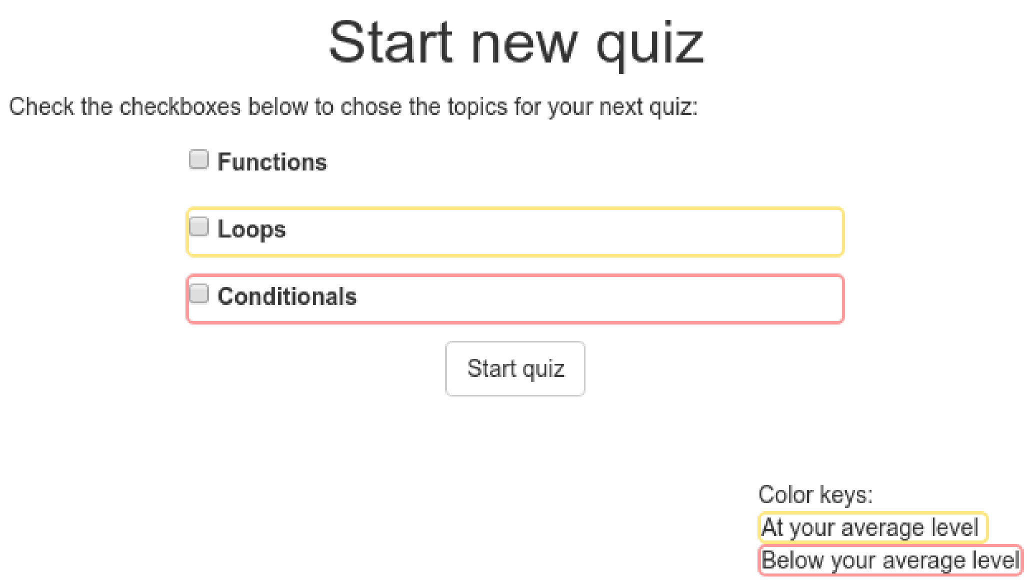
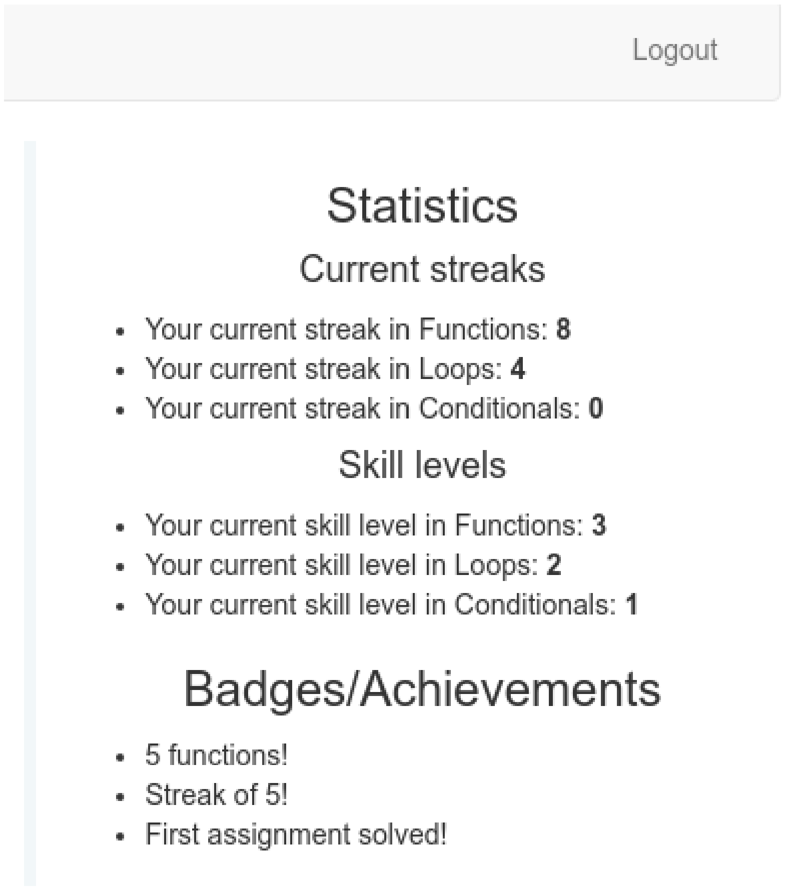


Figure 5.1: The view for configuring a new quiz

Figure 5.1 shows the view the user is presented with when configuring a new quiz. The student can choose one or more topics that the quiz should include. The list of topic options is highlighting topics that the student could need to do more work on. The way this is done is by calculating the average level a user has on all the available topics. Depending on this average level, the system highlights topics that the student is currently at the same level as, or a lower level. QuizGuide, discussed in section 1.3.1, showed that using this kind of adaptive annotation is a very effective way of guiding the students to quizzes that should be selected. This feature was one of the few that was introduced in between the two prototype test phases, and was therefore not considered during the first prototype test. During the second prototype test, the testers did not spend much time looking at the dashboard, but they all agreed that the idea would be very beneficial when configuring new quizzes, when the concept behind it was explained.



Statistics

Current streaks

- Your current streak in Functions: **8**
- Your current streak in Loops: **4**
- Your current streak in Conditionals: **0**

Skill levels

- Your current skill level in Functions: **3**
- Your current skill level in Loops: **2**
- Your current skill level in Conditionals: **1**

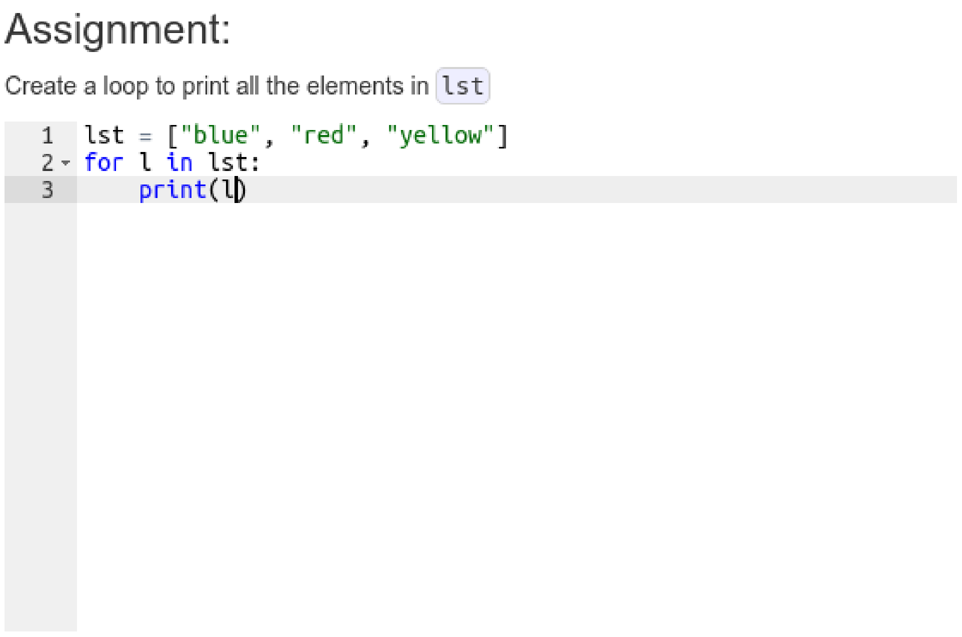
Badges/Achievements

- 5 functions!
- Streak of 5!
- First assignment solved!

Figure 5.2: This view displays the statistics for the current user

The other part of the dashboard is the statistics section. Figure 5.2 shows how statistics are displayed to the user. It was uncovered by the questionnaire and the prototype testing, that achievements and skill level can be a good source of motivation to do more assignments. Therefore, a summary of the stats were displayed on the users dashboard. It was not a highly commented on feature during the prototype testing. A reason for this could be that it has rather limited value in the beginning, before a sufficient amount of statistical data has been gathered, or any achievements has been earned. In addition, the testers spent most of their time solving assignments. It was noted, however, that it was interesting to see a summary of the different skill levels. In the future, when more topics and levels are added this could provide valuable feedback for the user on what topics need extra attention.

The streak statistics were meant to be a way for the student to compete against one self, or other students. The students could see how many correct assignments they could in a row, and compete to have the highest number. However, none of the testers seemed to pick up on that. The results from the questionnaire suggested that this would be a good source of motivation to do more assignments. As stated before, it might have had too little statistical data to be of any interest to the testers, and the time spent on the statistics page were very short. The effects of this feature could be enhanced by adding more graphical elements, to draw attention towards it. It could also be displayed in multiple locations, for example while doing assignments.

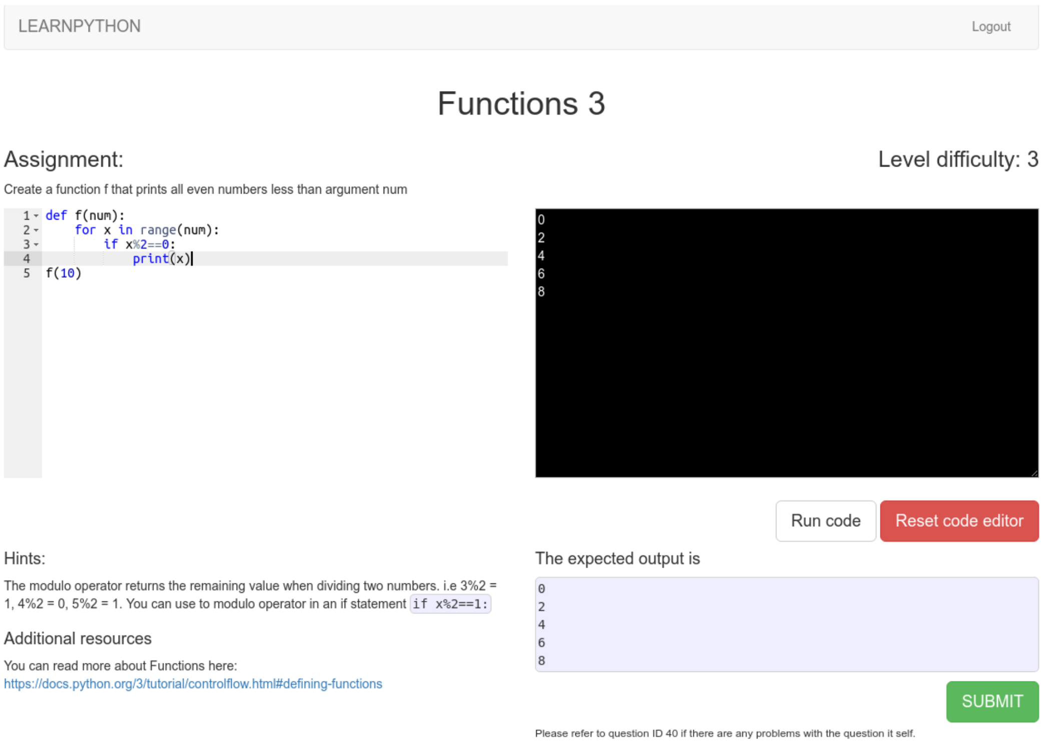

Assignment:

Create a loop to print all the elements in `lst`

```
lst = ["blue", "red", "yellow"]
for l in lst:
    print(l)
```

Figure 5.3: The ACE code editor

Figure 5.3 show the ACE code editor implemented for the system. It received feedback that it was very easy to use. There were no need to configure or install anything. The reason for making it easy to use was to make the system have as low threshold as possible for the users. Installing and configuring an IDE can be a daunting task for someone who are just getting in to python programming. The drawback of having such a simple IDE with no configuration options available for the users, is that the users can not set it up as they want. They can not change the color scheme, install advanced plugins, or even save the code. However, the assignments given by this system is not meant to be huge projects with multiple files and huge classes, so there would not be any need for all the bells and whistles that usually comes with an IDE. Figure 5.4 shows the assignment solving view of the second prototype.



Functions 3

Assignment:

Level difficulty: 3

Create a function f that prints all even numbers less than argument num

```
def f(num):
    for x in range(num):
        if x%2==0:
            print(x)
f(10)
```

```
0
2
4
6
8
```

Hints:

The modulo operator returns the remaining value when dividing two numbers. i.e 3%2 = 1, 4%2 = 0, 5%2 = 1. You can use to modulo operator in an if statement `if x%2==1:`

Additional resources

You can read more about Functions here: https://docs.python.org/3/tutorial/controlflow.html#defining-functions

The expected output is

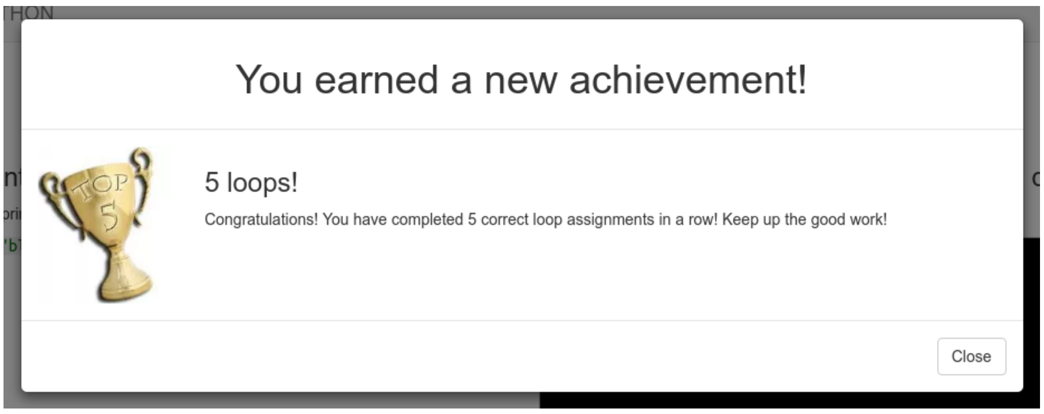
```
0
2
4
6
8
```

Please refer to question ID 40 if there are any problems with the question it self.

Figure 5.4: The assignment view for the second prototype

5.2.2 User Experience

An important aspect to consider when creating a new web application is the user experience. This factor is especially important, because the success of the system potentially could be measured by how much time the users are spending solving assignments. If the user has a good time using the system, he is more likely to come back and spend more time.


You earned a new achievement!

5 loops!

Congratulations! You have completed 5 correct loop assignments in a row! Keep up the good work!

Figure 5.5: The modal informing the user that a new achievement has been unlocked

One feature that most definitely was a hit among the prototype testers was the achievement system. This is a reward given to the student when a set of given requirements are satisfied. The questionnaire showed that more than 72% of the respondents had a positive attitude towards achievements. This was confirmed in the prototype testing where some of the testers even cheered when a new achievement was unlocked. Figure 5.5 shows the modal that pops up when the user unlocks the achievement of completing five assignments concerning loops in a row. To create a sense of relief and take a break from the static programming setting the achievement modal was made to slide in from the top with graphics in a surprising fashion. While the short term effect was very positive, the prototype testing did not examine the effect of wear out in a long term perspective. Over time, the excitement of getting a new achievement might be reduced when getting a new achievement turns into a routine. Therefore, the use of achievement should be used in a careful manner.

One thing that the testers complained about was getting the same assignment multiple times. As the system is already tracking what assignments a user has already solved, the system could easily filter out those assignments. However, when the system gets a much larger number of assignments, the chances for getting the same assignment multiple times in the same session decreases. If the system filters out assignments that the user already solved, the system runs the risk of running out of available assignments for a specific topic at a specific level.

Chapter 6

Conclusion and future work

It is now time to conclude this research project. A lot has been learned, and new questions has been discovered. Section 6.1 will summarize the project and attempt to answer the research questions posted in the introduction of this project. Section 6.2 will present new questions and problems that were discovered during the development of this system.

6.1 Conclusions

This system is something that can bring great value to students starting out on their journey to learn computer programming. Even though the skill level of the prototype testers varied a lot, from next to nothing to a couple of years, everyone walked away with more knowledge than they had before the test started. Section 1.2 posted two research questions that this project sought to answer:

1. RQ1: What are the requirements for a system that allows a student to self-evaluate their programming skills and get feedback on what topics they need to improve?

2. RQ2: How can a self-assessment system provide a student with programming tasks that fit the individual students skill level?

The following sections will provide a conclusion to each of the questions.

6.1.1 Research Question 1

The students who might benefit the most from this system are those who have to previous programming experience are struggling with the assignments. Therefore, this is a system that must be easy to use and most of the users must be able to use the system without special training. It must also be available across multiple platforms and must not require installation of any special software. To be able to track the performance and skill level of a student, and to allow administrators to add assignments, the system must have a function to authenticate the users and administrators.

Because the programming experience of the users are very variable, the system must also be able to provide easy assignments to struggling students, as well as assignments that can challenge students that are very good at programming. This adaptive assignment system is described in section 6.1.2. The system must provide feedback to the user to inform him about the status of the code, if the code submitted was wrong or correct. If the user is stuck on an assignment, the system should provide hints and recommend a place where the user can find additional information and documentation on the topic. If the user is unable to solve a problem, the user can move on to the next questions. The system must also inform the user about what topics require further work. This shall be done by using adaptive annotation to highlight those topics when the user is selecting what topics the next quiz will include. The system must also inform the user of what the current skill level of the various available topics are.

The propose of the system is to help students improve their knowledge of python programming. Therefore, most users using the system should gain some knowledge after using the system. Because of this, the system shall encourage the students to solve as many assignments as possible, and a majority of the users shall want to use the system again, after the first time use. This shall be done by awarding achievements. All unlocked achievements shall be displayed to the user. However, the number of achievements must be limited, to prevent the encouraging effect from being worn out immediately. Another way to keep the users interested in continuing to use the system is to always provide new assignments. Therefore, it must be possible for the system administrators to add new assignments, with hints and links to documentation, to the system. As the system is still in an early development stage, it is important to focus on maintainability and extendability while developing the system.

Table 6.1 summarizes the non-functional requirements for the system, while table 6.2 lists a summary of all the functional requirements.

ID	Non-functional requirement
NF1	99% of the users must be able to use the system without special training
NF2	The system must not require installation of additional software
NF3	90% of the users must want to use the system again, after the first time
NF4	90% of the users must learn something after using the system for 30 minutes
NF5	The system shall be cross platform compatible

Table 6.1: Summary of non-functional requirements

ID	Functional requirement
FR1	The system shall have a user/administrator authentication system
FR2	The system shall display the users current skill level for all topics
FR3	The system shall have adaptive annotation on topics that the users needs to improve
FR4	The system shall provide assignments with difficulty adjusted to the users skill level
FR5	The system shall show the user where additional documentation for a given assignment can be found
FR6	The system shall offer hints if the user is stuck with an assignment
FR7	The system shall verify and display if a user submission is correct or not
FR8	The system shall allow a user to continue to next question even if the code submitted contains errors
FR9	The system shall award achievements to the user
FR10	The system shall present a users earned achievements
FR11	The system shall allow system administrators to add new assignments with hints and links to documentation

Table 6.2: Summary of functional requirements

6.1.2 Research Question 2

To solve this problem, each assignment was annotated with a topic and a difficulty level and a user had a skill level for each topic. As the user solves assignments correctly, the user gains a higher skill level for a specified topic. As the user gains higher skill levels, the system picks out assignments with higher difficulty level equal to the skill level of the corresponding topic.

This solution to provide adaptive difficulty proved to be a good way, and the feedback suggested that such a system would be very much appreciated. The first draft required too much time to increase the skill level. A revised version of the heuristic function decreased the time needed to gain a skill level, by changing the number of correct assignments in a row to 3. This change in leveling time received positive feedback.

6.2 Future work

This system has a lot of potential for future improvements, the following sections briefly discuss a number of possible future work.

Further testing

While two prototypes of the system was tested by two smaller groups of testers, the system could benefit from more extensive testing and feature specific testing. One element to study is to see if the new system is more efficient than the old traditional exercise system currently used in TDT4110. Another element to investigate is the format of the assignments. Are shorter or longer assignments more efficient for learning, or should there be a mixture? When should *"fill in the missing part"* assignments be used and not used? The effect of some of the features could be tested. For example to award one group assignments, while another group does not get any.

Gamification

The system already offers a few mechanisms that originates from gamification theory, like achievements and a scoring system. Are there other elements that can be introduced to create even more motivation to complete more assignments and immersion in the system? The adaptive assignment difficulty system is also related to the level system, where the player achieves higher levels after gaining experience. As the level increases, so does the difficulty in the game.

Mass production of assignments

One issue that arose during the prototype testing was the number of assignments. Some of the testers complained that they kept seeing the same assignment over and over. After solving the same assignments several times, the learning effect fades away. However, creating a lot of assignments requires a lot of time and resources. Of the systems presented in section 1.3, only QuizPACK offered a solution for generating a large number of assignments out of a smaller number of template assignments. Is it possible to integrate parameterized problems into the system, or are there an even better way to create a huge amount of assignments? Code wars uses the community to create assignments. Is using community submitted assignments a viable solution in the educational setting of this system?

More sophisticated verification

The current verification of code is works well from a security perspective. However, is it possible to do more elaborate testing of programs to ensure correctness, and integrate the tests with automatic assignment generation?

More advanced heuristics for level advancement

The heuristics for selecting assignment difficulty worked great, after some adjustment, with assignments that was centered around a single topic. Is it possible to modify the heuristic to work on assignments that incorporates any number of topics?

Flexible dynamic achievements

Making a system for achievements that is both flexible, dynamic and easy to extend turned out to be quite a hard task. Therefore, to make the system more extendable, the system needs a way to make easily create new achievements.

Bibliography

[1] P. Brusilovsky and S. Sosnovsky, “Individualized exercises for self-assessment of programming knowledge: An evaluation of quizpack,” *Journal on Educational Resources in Computing (JERIC)*, vol. 5, no. 3, p. 6, 2005.

[2] S. Sosnovsky, O. Shcherbinina, and P. Brusilovsky, “Web-based parameterized questions as a tool for learning,” in *Proc. of World Conference on E-Learning, E-Learn*, 2003, pp. 7–11.

[3] P. Brusilovsky, S. Sosnovsky, and O. Shcherbinina, “Quizguide: Increasing the educational value of individualized self-assessment quizzes with adaptive navigation support,” in *Proceedings of E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2004*, J. Nall and R. Robson, Eds. Washington, DC, USA: Association for the Advancement of Computing in Education (AACE), 2004, pp. 1806–1813.

[4] P. Brusilovsky and L. Pesin, “Visual annotation of links in adaptive hypermedia,” in *Conference Companion on Human Factors in Computing Systems*, ser. CHI ’95, 1995, pp. 222–223.

[5] Codecademy, “Learn to code | codecademy,” 2015, [Online] https://www.codecademy.com/ [Accessed: 15- May- 2016].

[6] ——, “Introducing codecademy pro,” 2015, [Online] www.codecademy.com/pro [Accessed: 15- May- 2016].

[7] Codewars, “Train with programming challenges/kata | codewars,” 2015, [Online] www.codewars.com [Accessed: 15- May- 2016].

[8] C. Soanes and A. Stevenson, *Oxford Dictionary of English*. Oxford University Press, 2005.

[9] Codewars, “About | codewars,” 2015, [Online] www.codewars.com/about [Accessed: 15- May- 2016].

[10] ProjectEuler, “About - project euler,” 2015, [Online] https://projecteuler.net/ [Accessed: 15- May- 2016].

[11] E. Enström, G. Kreitz, F. Niemelä, P. Söderman, and V. Kann, “Five years with kattis—using an automated assessment system in teaching,” in *Frontiers in Education Conference (FIE), 2011*. IEEE, 2011, pp. T3J–1.

[12] Kattis, “Kattis,” 2016, [Online] www.kattis.com [Accessed: 20- May- 2016].

[13] ——, “Kattis,” 2016, [Online] open.kattis.com [Accessed: 20- May- 2016].

[14] Y. B.-D. Kolikant and M. Mussai, ““so my program doesn’t run!” definition, origins, and practical expressions of students’ (mis)conceptions of correctness,” *Computer Science Education*, vol. 18, no. 2, pp. 135–151, 2008.

[15] L. P. Rieber and D. Noah, “Games, simulations, and visual metaphors in education: antagonism between enjoyment and learning,” *Educational Media International*, vol. 45, no. 2, pp. 77–92, 2008.

[16] A. W. Chickering and Z. F. Gamson, “Seven principles for good practice in undergraduate education.” *AAHE bulletin*, vol. 3, p. 7, 1987.

[17] S. T. March and G. F. Smith, "Design and natural science research on information technology," *Decision support systems*, vol. 15, no. 4, pp. 251–266, 1995.

[18] H. A. Simon, *The sciences of the artificial.* MIT press, 1996.

[19] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[20] V. Vaishnavi and W. Kuechler, "Design research in information systems," 2004.

[21] S. Dahlum, "Kvantitativ analyse – store norske leksikon," 2014, [Online] https://snl.no/kvantitativ_analyse [Accessed: 24- May- 2016].

[22] U. Malt, "Kvalitativ – store norske leksikon," 2015, [Online] https://snl.no/kvalitativ [Accessed: 24- May- 2016].

[23] A. Dwaraki, S. Seetharaman, S. Natarajan, and T. Wolf, "Gitflow: flow revision management for software-defined networks," in *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research.* ACM, 2015, p. 6.

[24] I. Gilfillan, "Comparison of web frameworks," 2016, [Online] https://en.wikipedia.org/wiki/Comparison_of_web_frameworks [Accessed: 27- May- 2016].

[25] Python, "Webframeworks | python wiki," 2016, [Online] https://wiki.python.org/moin/WebFrameworks [Accessed: 27- May- 2016].

[26] F. Vieira, "What are some websites developed using python?" 2015, [Online] https://www.quora.com/

What-are-some-websites-developed-using-Python [Accessed: 27- May-2016].

[27] reddit, “Reddit: the code that powers reddit.com,” 2016, [Online] https://github.com/reddit/reddit [Accessed: 27- May- 2016].

[28] J. Zaytsev, “Ecmascript 5 compatibility table,” 2016, [Online] http://kangax.github.io/compat-table/es5/ [Accessed: 14- May- 2016].

[29] postgreSQL, “Postgresql: About,” 2016, [Online] https://www.postgresql.org/about/ [Accessed: 27- May- 2016].

[30] Flask, “Flask: A microframework based on werkzeug, jinja2 and good intentions,” 2016, [Online] https://github.com/pallets/flask [Accessed: 27-May- 2016].

[31] sqlalchemy, “Engine configuration — sqlalchemy 1.1 documentation,” 2016, [Online] http://docs.sqlalchemy.org/en/latest/core/engines.html [Accessed: 27- May- 2016].

[32] Django, “Django overview | django,” 2016, [Online] https://www.djangoproject.com/start/overview/ [Accessed: 27- May- 2016].

[33] Codewars, “This project contains the cli tool that allows codewars.com and strive.co to execute unsafe code remotely in multiple languages.” 2016, [Online] https://github.com/Codewars/codewars-runner-cli [Accessed: 14- May- 2016].

[34] Docker, “What is docker?” 2016, [Online] https://www.docker.com/what-docker [Accessed: 14- May- 2016].

[35] Python, “Webbrowserprogramming - python wiki,” 2016, [Online] https://wiki.python.org/moin/WebBrowserProgramming [Accessed: 14- May- 2016].

[36] Moonlight, “Moonlight,” 2015, [Online] http://www.mono-project.com/docs/web/moonlight/ [Accessed: 14- May- 2016].

[37] J. de Hooge, “Lean and mean python to javascript transpiler, featuring multiple inheritance and small downloads,” 2016, [Online] https://github.com/JdeH/Transcrypt [Accessed: 14- May- 2016].

[38] B. Miller, “Skulpt is a javascript implementation of the python programming language,” 2016, [Online] https://github.com/skulpt/skulpt [Accessed: 14- May- 2016].

[39] P. Quentel, “Brython,” 2015, [Online] http://brython.info [Accessed: 14- May- 2016].

[40] ——, “Brython documentation,” 2015, [Online] http://brython.info/static_doc/en/syntax.html [Accessed: 14- May- 2016].

[41] Bootstrap, “Bootstrap: The world’s most popular mobile-first and responsive front-end framework,” 2015, [Online] http://getbootstrap.com/ [Accessed: 14- May- 2016].

[42] Foundation, “Foundation | the most advanced responsive front-end framework in the world.” 2015, [Online] http://foundation.zurb.com/ [Accessed: 14- May- 2016].

[43] jQuery, “jquery,” 2016, [Online] https://jquery.com/ [Accessed: 14- May- 2016].

[44] w3Techs, "Usage statistics and market share of javascript libraries for websites, may 2016," 2016, [Online] https://w3techs.com/technologies/overview/javascript_library/all [Accessed: 14- May- 2016].

[45] Ace, "Ace (ajax.org cloud9 editor)," 2016, [Online] https://github.com/ajaxorg/ace [Accessed: 14- May- 2016].

[46] CodeMirror, "Codemirror," 2016, [Online] http://codemirror.net/index.html [Accessed: 14- May- 2016].

[47] Ace, "Ace - the high performance code editor for the web," 2015, [Online] https://ace.c9.io/#nav=production [Accessed: 14- May- 2016].

Appendix A

Acronyms

AJAX	JavaScript and XML
CSS	Cascading Style Sheets
DOM	Document Object Models
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IS	Information System
NTNU	Norges teknisk-naturvitenskapelige universitet
ORM	Object-relational mapping
SQL	Structured Query Language
URL	Uniform Resource Locator

Appendix B

Questionnaires

B.1 First questionnaire

As the native language for most of the target respondents are norwegian, the questions were designed and asked in norwegian.

1. Har du erfaring med programmering generelt?
   Alternativer: Ja/Nei
   (a) Hvis ja: Hvor mye erfaring:
   Alternativer: 1(Litt) - 5(mye)
2. Har du hatt ITGK?
   Alternativer: Ja/Nei
   (a) Hvis ja: Hadde du tidligere programmeringserfaring?
   Alternativer: Ja/Nei
3. Hvor mye er du enig i følgende utsagn:
   Alternativer: Helt uenig/Litt uenig/Hverken eller/Litt enig/Helt enig/Vet ikke

(a) Jeg er ganske god til å programmere

(b) Jeg vil gjøre flere oppgaver hvis de er tilpasset mitt kunnskapsnivå (Ikke for vanskelige/lette)

(c) Jeg vil gjøre oppgaver oftere dersom de er korte

(d) Jeg vil gjøre programmeringsoppgaver når det passer meg best, f.eks når jeg sitter på bussen o.l, dersom de er tilgjengelige på mobil plattform.

4. Jeg lærer best å programmere ved å gjøre:
Alternativer: Velg en

    (a) korte oppgaver

    (b) lange oppgaver

    (c) en blanding av korte og lange oppgaver

    (d) velge selv om jeg vil ha lange eller korte oppgaver

5. Når jeg gjør oppgaver ønsker jeg:
Alternativer: Velg en

    (a) kun oppgaver med et eget bestemt emne (Løkker, if-else struktur o.l)

    (b) oppgaver med tilfeldig blanding av emner

    (c) å kunne velge enten ett emne eller en egen balnding av emner

    (d) at systemet skal finne ut hvilke oppgaver jeg trenger å jobbe mer med

6. Jeg blir motivert til å gjøre flere oppgaver dersom jeg kan:
Alternativer: Helt uenig/Litt uenig/Hverken eller/Litt enig/Helt enig/Vet ikke

(a) konkurrere med poeng mot meg selv

(b) konkurrere med poeng mot mine venner

(c) låse opp bragder (achievements/badges)

(d) dele mine fremskritt på sosiale medier

7. Jeg har andre forslag til et system for tilpasset læring av (python) programmering:

   Alternativer: Tekstbox