



CHALMERS

UX ANALYSIS: A WORKOUT APPLICATION FOCUSING
ON ELEMENTS OF USER EXPERIENCE AND
GAMIFICATION

*Andreas Carlsson, Felix Engelbrektsson, Lisa Larsson,
Elin Nordström, Mickaela Södergren*

Chalmers University of Technology
Department of Computer Science and Engineering

Gothenburg, Sweden 2018

Abstract

The purpose of the thesis was to examine what aspects make up a compelling user experience by developing a gym workout application for mobile phones. By combining research and collaborating with test users, an application for Android and iOS was developed and evaluated. By doing several iterations of the application it was possible to find out what contributes to a compelling user experience and implement this in a real product. Gamification increases the user experience and has the potential to motivate users. User experience is a subjective topic which makes it difficult to define. Aspects to consider are standardization, a logical flow and visual effects. The project outcome was an application released to the app stores as well as valuable insights into how user experience can make an impact on the usability of an application.

Sammandrag

Syftet med projektet var att undersöka vilka aspekter som bidrar till en tilltalande användarupplevelse och implementera detta i en mobilapplikation med fokus på gymträning. Genom en kombination av efterforskning och samarbete med testanvändare utvecklades och utvärderades en mobilapplikation. Genom att göra flera iterationer av applikationen var det möjligt att undersöka vad som bidrar till en tilltalande användarupplevelse och implementera detta i en produkt. Användandet av spelmoment bidrar till en förbättrad användarupplevelse och har potentialen att motivera användare. Användarupplevelse är ett subjektivt ämne som är svårt att definiera. Aspekter att ha i åtanke är standardisering, ett logiskt flöde och visuella effekter. Värdefulla insikter kunde dras om hur stor påverkan användarupplevelsen kan ha på en applikation. Projektet resulterade i en mobilapplikation som släpptes på App Store och Google Play.

Contents

1	Introduction	1
1.1	Background	1
1.2	Purpose	1
1.3	Limitations	2
1.4	Summary of Method	2
2	Literature Review	3
2.1	User Experience	3
2.2	Gamification	3
3	Implementation	4
3.1	Early Stage Mockups	6
3.2	Core	10
3.2.1	React and React Native	10
3.2.2	Basic Workout Schedules	11
3.2.3	Backend and Database	11
3.2.4	API Design	11
3.2.5	User Authentication	13
3.2.6	Security	14
3.2.7	Backup Strategies	15
3.2.8	Server Monitoring	15
3.2.9	Development Tools	16
3.2.10	Deployment	17
3.2.11	Application State Management	18
3.3	Gamification	19
3.3.1	Achievements	20
3.3.2	Calculating achievements	20
3.4	User Experience	21
3.4.1	User Testing	22
3.4.2	Features and Statistics	24
3.4.3	Animations and UX	25
3.5	The Website	25
4	The Final Application	26
4.1	The Main Views	27
4.1.1	Log In/Create Account	27
4.1.2	Dashboard	28
4.1.3	Workout	30
4.1.4	Calendar	33
4.1.5	Schedules	33
4.1.6	Achievements	34
4.1.7	Settings	35
4.2	Navigation	36
4.3	Design	37

4.4	Gamification and Animations	37
5	User Experience of the Final Application	37
5.1	Feedback From Test Users	38
5.2	Measured Results	38
6	Societal and Ethical Aspects	39
6.1	Integrity	40
6.2	Ethical Challenges	40
7	Discussion	41
7.1	Analysis of The Resulting Product	41
7.2	Analysis of The Resulting User Experience	44
7.3	Research Methodology	45
7.4	Contribution	46
7.5	Future Direction and Possible Improvements	46
7.6	Required Competence	47
8	Conclusions	47
	References	49
A	Updated Icons	51
B	Privacy Policy	52
C	Terms & Conditions	54
D	Poster - Campus User Test	56
E	Password Reset Email	57
F	Mockups	58
G	Website	61

Dictionary

Backend

The server side part of the application, running on a centralized server and handling requests from the frontend.

Exergames

A term used for video games that are also a form of exercise.

Frontend

The part of the application visible to the user.

JSON

JavaScript Object Notation. A compact text based format for exchanging data.

JWT

JSON Web Token. Format for creating access tokens encrypted with a secret key.

mHealth

Abbreviation for mobile health, which is the usage of mobile phones in medical care.

Middleware

A function that interrupts the normal flow of a function and make changes to the input variables, or in case of a server, the request object.

Node

A runtime for executing JavaScript code on a server instead of in a browser.

REST

Representational State Transfer. A way architecture API's in a unified manner where the HTTP-verb in combination with the endpoint URL should be self-explanatory for the action performed.

TLS/SSL

Transport Layer Security / Secure Socket Layer. A security mechanism that ensures that data transmitted over a network are encrypted.

UI

User interface, the part of an application that users can see and interact with.

UX

User experience. The overall experience for someone when using a program or application.

1 Introduction

In this introductory section the background of the subject will be given with some of the previous research on how gamification might improve the user experience. This is followed by the purpose along with limitations and the project methodology.

1.1 Background

There are over 300.000 different health and fitness applications for mobile phones available on the market, many of them with their own brand or niche. This is a trending phenomenon known as mobile health application (mHealth), the use of smartphones and applications in medical care (Steinhubl, Muse, & Topol, 2013).

Although there exists many health and fitness applications sharing a similar scope, almost half of them (40%) has a limited success among the final users (less than 5.000 downloads) (IMS Institute for Healthcare informatics, 2015), for reasons related to considerable differences in terms of features, engagement and user-friendliness.

While there seem to be plenty of tools and services to help a person get started with their training and health goals, obesity in Sweden has increased over the recent years (Folkhälsomyndigheten, 2018). One of many possible reasons for that could be that people tend to lose the motivation to change their lifestyle habits, even with the help of mobile applications.

Gamification and user experience appear to be a way to keep motivating users. An example of this is the field experiment on how badges can increase user activity where they found a correlation on gamification and the user engagement (Hamari, 2017). Other health and fitness applications using gamification are “Zombies, run!”, where the user outruns zombies, and “Ingress”, where the user captures flags. These two are examples of what is called Exergaming, which also has proven quite effective in engaging users in their workout through gamification (Sun, 2013).

The majority of applications available on the market today does not seem to be appealing enough for the users to continue their usage, when considering the limited success among users. There is therefore a need to investigate the possible elements that makes an application appealing.

1.2 Purpose

The purpose of the project is to investigate how one could develop a mobile application for gym workouts that is appealing and perceived as easy to use. The project focuses particularly on the user experience and ways to enhance it with gamification.

The following questions are answered in this thesis:

- What previous research has been made on user experience and gamification in the means of motivating a user?
- Which are the elements in the application that seems to be less or more appealing to the user?

1.3 Limitations

The application developed in this project is focused on gym workouts. There are other applications out there that niche themselves towards diet, cardiovascular activities, yoga or a combination of them. To have a clear focus, this application's scope is gym workouts even though healthy meals and cardiovascular exercise also play a role in a person's general health.

1.4 Summary of Method

The task was to investigate how to make the user experience compelling and the application motivating. To determine this, a literature review of previously performed studies and an empirical investigation were conducted.

The literature review consisted of summarizing results from studies about user experience and gamification, as well as examining health and fitness applications that were currently available on the market. The results from the review served as a backbone when designing the foundations of the application.

The development of the application started at the very beginning of the project. The application was modified as the review and the empirical examinations were merged into the development. The implementation part of the project consisted mainly of the core functionality (i.e. the workout diary and schedules), gamification and the user experience.

The empirical investigation on the other hand depended on the feedback from test users. Their feedback on different versions of the application was used to improve it for the next version. Having test users participating in the process from the very start was intended to help in discovering issues and problems at an early stage, rather than at the end of the project.

In addition to the test group that followed the development of the application, a final test on a public site was performed towards the end of the project. The test users were not chosen in advance, instead they consisted of people passing by that were interested in testing the application. They were given a couple of tasks to perform, and the results of the test depended on their feedback and how well they completed the tasks.

2 Literature Review

To gain a better understanding of the field, it was important to carry out a literature review. Gamification and user experience were examined since they are the main areas of significance to the project.

2.1 User Experience

User Experience (UX), refers to how a user is affected by a product and focuses on understanding the users' needs, abilities and limitations. There are three perspectives of user experience (Turner, 2017). The first is that its main attribute is usability and accessibility. The second is that user experience is something that is difficult to analyze mainly because the perception of what is easy and desirable is highly subjective. Experiences are personal and of the moment, according to Turner. The third perspective is that it is a way to indicate the complexity of human psychology and a fresh way to apply this knowledge in technology.

A part of the user experience is about where the user expects to find different functionality, for example menus and titles. Some kind of standardization makes the experience easier to understand and less frustrating for the user (Dyson & Jennings, 2014). The user might also feel that the interface is cluttered when the functionality do not appearing where they are expected or when there are too many functionalities in focus at the same time (Dyson & Jennings, 2014).

2.2 Gamification

Deterding et al. defines gamification as “the use of game design elements in non-game context” (Deterding, Dixon, Khaled, & Nacke, 2011, p. 9). The goal is usually to engage the users, motivate learning and productivity, as well as usefulness and physical exercise. The key is to activate the person's natural desires of competing against others, gaining status and mastery by learning. Social aspects also play an important role (Deterding et al., 2011).

Usually gamification consists of rewards in the form of achievements and badges that are given to the user when fulfilling a streak or completing a challenge. It can also be competition and making progress visible to others, with high scores and leaderboards. Studies show that users generally use a product or service less over time and that gamification has a positive effect on preventing this (Hamari, 2017). This is causing users to be more engaged both in the amount of time they use the service and the effort they put in their usage.

It is no coincidence that millions of people around the world suddenly one day started walking down the streets to catch pokémons when the popular game “Pokémon Go!” was released (Hern, 2016). Although perhaps unintentional from the user's perspective, the game lead to a significant increase in physical

activity (Althoff, White, & Horvitz, 2016). Furthermore, according to Althoff et al., it was revealed that Pokémon Go! had achieved an increase across all user groups including the low activity populations.

In a study about the effects of game-based applications versus performance-based applications, it was discovered that game-based applications may help users dissociate from exercises more readily (Gillman & Bryan, 2016), which, for a user who does not enjoy exercising, might be more favorable.

Gamification has especially shown effectiveness in healthy and younger groups, but has also proven itself with people who has a good overall health, but does not exercise, as well as unhealthy people (Lee, Lee, & Lee, 2017). According to Lee et al., findings suggest that there were differences in the effectiveness of gamification for different kind of groups.

There are some disagreement on gamification. The term itself is broadly used, a habit that is criticized by game developers since it only uses a few components of a game. This adaption, when used for example in fitness applications, tend to erase the natural motivation and hype that a game normally provides (Lister, West, Cannon, Sax, & Brodegard, 2014). There is also an uncertainty regarding the use of gamification when only using game elements and not being an actual game. Many studies on health and gamification have targeted exergames and other games where you need to move and be physically active while playing (Lister et al., 2014), in comparison to non-game services.

3 Implementation

The developing process was roughly divided into three layers, visualized in Figure 1, that describe its focus and what would be the most influential features that would shape the application.

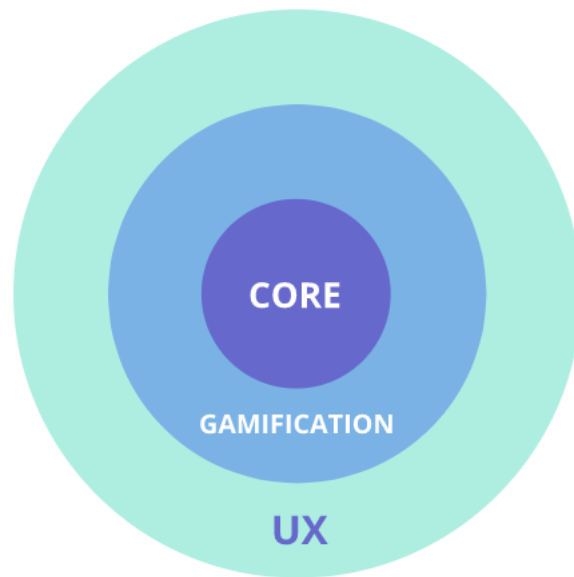


Figure 1: The three layers of the application. Created by author.

The **core layer** of the application included basic functionality like creating an account and logging in to the application. It also included the ability to create workout schedules, log an ongoing workout and view previous workouts. The main focus of the application was gym training, which meant the focus of all the core aspects were for working out in the gym.

The **gamification layer** of the application used techniques to motivate the users and inspire them to increase their performance. The gamification part of the application was built upon the core foundation by adding features to the application on top of the functionality. This was done in order to make the application more vivid and to give the user a feeling of accomplishment. With the application reacting to the user, it exhibited behaviors that told the user that their actions had an impact.

The **UX layer** of the application enhanced the user experience by adding subtle animations, thoughtful design choices, a logical flow and soft colors. This layer surrounded the entire application in a way that every part of the application somehow contributed to the total user experience.

During the stages of actual implementation, the different features of the application were administered during separate stages. This chapter goes more into depth of the implementation procedure of each feature, as well as what techniques were used.

3.1 Early Stage Mockups

Some mockups were created before and early into the development of the application, see Figure 3 to 8. At that point in time the most basic functionality had already been decided, but there was still some uncertainty as to what more functionality the application should have. More mockups of the early stage ideas can be seen in Appendix F. There was no specific method used to create the mockups. Instead, inspiration was taken from other applications when creating the design.

To make the application easy to survey, the overview of today's workouts was added, see Figure 3. By presenting today's workouts on the dashboard there is no need for the user to search for the most current information, which might decrease the number of taps for the user. The idea with the dashboard was to have the most important information collected in one place. At that point, there had been discussions about incorporating statistics and achievements into the application. Because of that, those features were added as sections to the dashboard as well.

The design of the application was meant to be light and spacious. To accommodate this, the dashboard was drawn with white background and plenty of space between the different sections. The color palette of the application was intended to be pastel and have a calm and fresh feeling. Because of that, shades of green, blue and purple were chosen, see Figure 2.

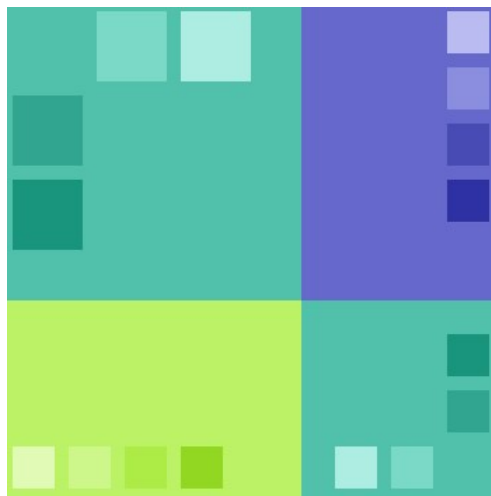


Figure 2: The color palette that was used in the entire design of the application. Created by author.

When it came to the actual workout views, the thought was to give them more emphasis by forgoing the white background and instead use the base colors in

the palette, see Figure 4 to 7. The hope was that by doing so it would give the users a fun feeling and would make them feel energized.

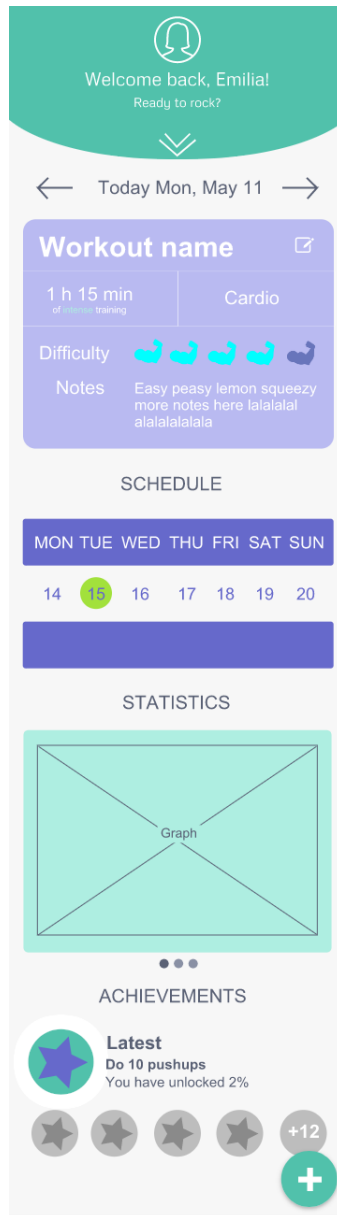


Figure 3: Early mockup of the dashboard. Created by author.

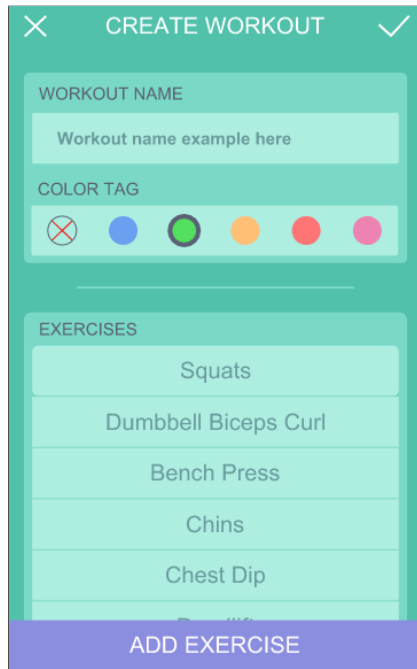


Figure 4: The first idea of viewing and editing a workout.
Created by author.

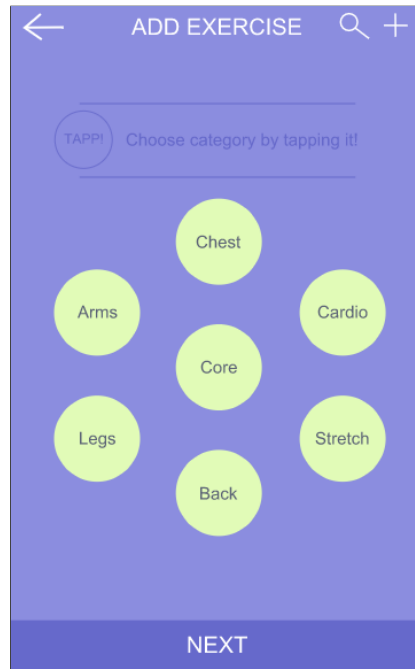


Figure 5: Initial mockups for choosing which exercise type to add.
Created by author.

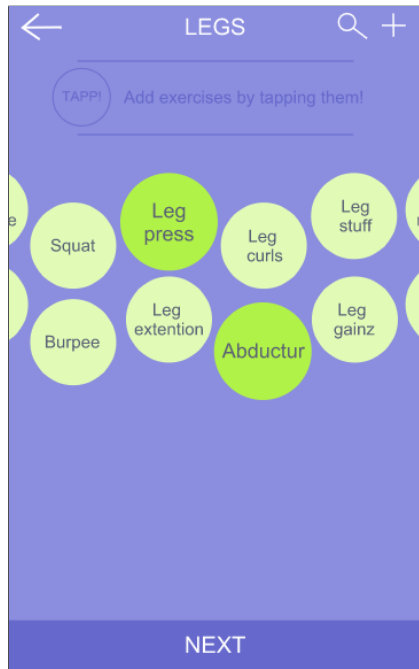


Figure 6: The view following the “select exercise type” view.
Created by author.

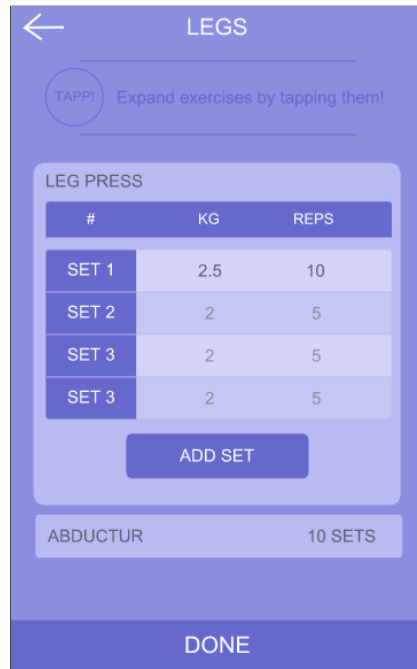


Figure 7: Early mockup for entering sets and repetitions.
Created by author.

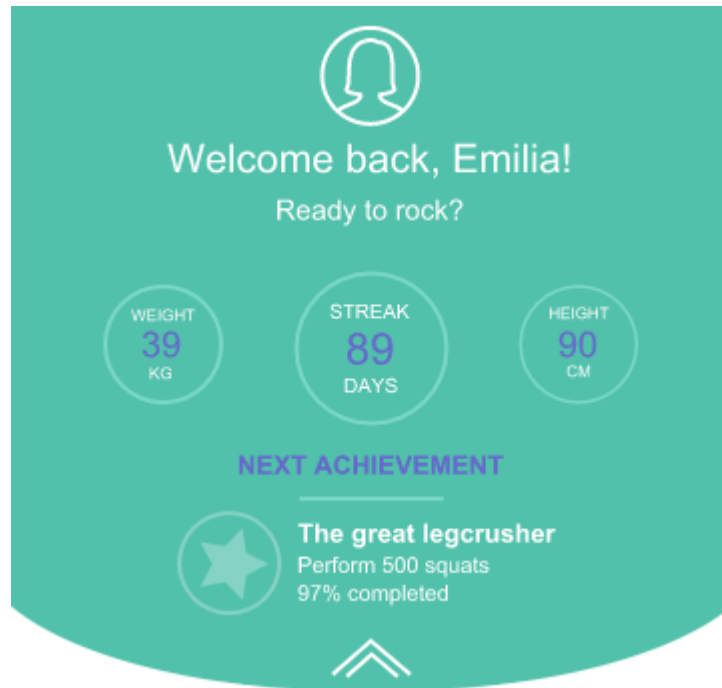


Figure 8: Header fully expanded in the dashboard view. Created by author.

3.2 Core

This section explains the implementation of the core functionality, starting with a brief explanation of a few concepts and then proceeding with the different parts of the core implementation.

3.2.1 React and React Native

React is a JavaScript library used for building user interfaces. The library permits large web applications to be developed without reloading the entire view. Instead, React creates an in-memory data structure cache, commonly referred to as the virtual document object model (DOM). When the application state changes, the new computed DOM is compared to the previous one and only the necessary things that changed are re-rendered. This makes view rendering much more effective and predictable compared to if the entire view was to be re-rendered.

React Native is the React architecture applied for developing against Android and iOS. The principle of React Native is the same as with React with the difference that it uses native views for a specific platform instead of the DOM element of an HTML document. It runs in a background process where

the JavaScript code is interpreted. This is done directly on the device. Communication with the Native platform is done via an asynchronous bridge.

3.2.2 Basic Workout Schedules

Implementing this in a simple and intuitive way proved a bit harder though. A few problems that has been present are whether or not to allow custom exercises, because if there is a defined set of exercises, that set has to be large and cover the majority of the users' needs, creating new sets for a given exercise without forcing the user to do too many actions or filling out unwanted fields. An attempt to solve the latter problem was to introduce a copy function which would copy a set and insert it with the same numbers.

3.2.3 Backend and Database

The backend of the application is hosted on a virtual private server located in Amsterdam. The operating system used is Ubuntu (version 17.10). This was chosen due to its good reputation and wide adaption within the industry.

During the development of the product, the entire server side stack was running on one single machine without any partitioning or load balancing. In a larger project this would not have been a great option since it introduces a single point of failure. It also became quite obvious during the project that as soon as something broke on the backend, the application was broken for everyone.

The choice of the actual server side technology was a bit more complex to make since there were a lot of good options. The group had experience from writing servers in both Python and JavaScript. Either one would had been a viable option so it was mostly a question of which one would be fastest to get up to speed with. In the end, JavaScript with the engine NodeJS together with the web framework ExpressJS was chosen.

For the database part of the backend, the main decision to make was to use either an object oriented database like MongoDB or to use a classical relational one like PostgreSQL. A relational SQL database and in particular PostgreSQL was chosen for the project. The reason for this was mainly that the group members had more experience with relational databases than with object oriented ones. Since the gamification part of the project would need some advanced queries it would be more effective to use a more established technology.

3.2.4 API Design

For the communication between the frontend (the mobile application) and the backend (the Linux server), an API had to be developed. The first step was to decide whether to go with the classical RESTful design or with the newer GraphQL implementation. There was some prior knowledge for both approaches

within the group and both of them had their own upsides and downsides. Since the RESTful design was the quickest to get up to speed with, this was implemented in the first version but was considered to be subject for change later on.

REST is more of a philosophy than an actual specification for API design. This gives some freedom, and responsibility, to the implementer to use the correct verbs and endpoints. The GET and POST endpoints turned out to be quite easy to design in a good fashion. What was more difficult was the PUT/PATCH endpoints. The reason for this was that a workout needs to be partly saved while the user is interacting with it. If the user for example changes the difficulty of a workout then this specific information needs to be communicated to the backend, together with the workout id and the user token. This is significantly less data than for an entire workout together with all of its exercises. Even though it in some sense breaks the RESTful philosophy it was decided to implement these kind of situations as subqueries. In the above example this was solved by sending a PATCH request to the endpoint `/api/workouts/difficulty/:id`.

In a mobile application it is convenient for the user if all of the data is synchronized with the backend without any need for pressing a save button. Communication asynchronously with the server in the background every time the application state changes is then necessary. Furthermore, a balance between too much communication and the risk of losing progress is needed when keeping the client and server synchronized.

```

export function addSetToExercise(exerciseId, reps, weight) {
  return dispatch => {
    AsyncStorage.getItem('jwt').then(jwt => {
      axios
        .post(
          `https://getpushapp.com/api/workouts/exercise/${exerciseId}`,
          {
            reps,
            weight
          },
          {
            headers: { Authorization: `Bearer ${jwt}` }
          }
        )
        .then(({ data }) => {
          dispatch({
            type: ADD_SET_TO_EXERCISE,
            payload: { id: data[0].id, reps, weight }
          });
        });
    });
  };
}

```

Figure 9: Redux action for making a request to the backend to add a set to a specific workout. Created by author.

For this application the balance was to save the application state as often as possible without causing too much network traffic. Button presses, like changing difficulty or color, immediately sends a request to the server while saving text input was performed when the component lost focus, i.e. keyboard was dismissed. This choice was made due to the relative small number of users. With a larger number, a different approach might have been more appropriate.

3.2.5 User Authentication

The very first part of the project was to set up a user authentication system where the users should be able to create an account, logging in to the application and, if needed, reset their password.

In the first version of the authentication system a token was created on the server side, stored in the database and sent to the user. This token was then sent to the backend with every request together with a unique identifier for the specific user. When a request from the client hit the backend a request was made to the database to fetch the token for the specific user. If the token sent with the request payload matched the token stored in the database, the request was processed. If no token was attached or if there was a mismatch, an error was sent back to the client.

This approach was simple to implement but had several drawbacks. Sending a request to the database to fetch user information for every request both increased the load on the server and the time it took for the client to get a response. This was not too noticeable in the beginning since so few requests were made but would have created a major bottleneck in the future. Another drawback that was more obvious to the team was that only one person could be logged in to a specific account simultaneously.

The entire log in system was refactored to make use of JSON Web Tokens instead. This way an object containing the user's id is created on the server each time the user logs in. The object is then encrypted with a private key and passed to the client in the response object.

The client receives the encrypted object, i.e. the token, and stores it locally on the user's device. Whenever a request is sent to the server the token is attached to the request in the authentication part of the header. The server decrypts the token and attaches the user id to the request object before it hits the actual endpoint. A middleware is used to interrupt the request and handle the authorization part of it.

The decryption of the token serves two purposes. First, if the decryption is successful we know that the request is made from a logged in user. Secondly, by attaching the user id to the request object we can make sure that users only get access to data they're authorized for. There is no need to do an extra round trip to the database to fetch user data since it is all stored in the encrypted token.

Using this new JWT authentication system made it possible to have several devices logged in on the same account at the same time. This helped a lot while developing the application and made it possible to collaborate in a more effective manner.

3.2.6 Security

Closely related to user authentication and authorization is the security of the application. Since the data that is provided by the user is considered as sensitive, especially when it comes to handling passwords, it is important to protect the system from any possible attacks.

All data between the server and client is transmitted encrypted using HTTPS with signed certificate. The certificate was provided by a service called LetsEncrypt which provides free TLS/SSL-certificates that are easy to install on your own server. Using TLS/SSL helps protect against man-in-the-middle attacks and also makes sure that the client are communicating with the intended receiver of requests.

3.2.7 Backup Strategies

Managing user data on the server side also requires safety in terms of data loss, so a backup is taken on a regular basis. To handle this a CRON job was set up which gets run twice every hour. The CRON job calls a shell script which serves two purposes. First it finds all backups older than 5 days and removes them. Then a connection to the database is established and a raw dump of the data is being made.

The database dumps are stored as SQL files on the same server in a designated backup folder which has restricted access. If a potential attacker somehow managed to gain access to the server they would have a difficult time getting access to the backup files.

In larger scale applications it would not be sufficient to store this data on the same machine since a system failure might not only destroy the database itself but also destroy the backups. A more robust way to handle this would be to store the backups on a separate media, preferably not at the same location as the database server.

The choice of running the backup script twice every hour and saving the backups for five days was done for the sake of keeping the load on the database server to a minimum and not letting the backups take up too much space. There is a trade-off for how often backups should be taken in the application. If done too often the load on the server will be too high and if done too seldom a restored backup will contain stale data.

One option that was considered but not implemented due to lack of time was to make use of write-ahead logging. When using write-ahead logging, every request that makes any change to the data (i.e. all queries except for pure selects) are saved in a separate file. Using this technique allows for backups to be taken more seldom since the backup in combination with the write-ahead log will make sure that the data is restorable at any given time.

3.2.8 Server Monitoring

Checking the state of the server and sending alerts when something fails is important for keeping the downtime as low as possible. Since the server side part of the application runs on a single machine, thus introducing a single point of failure, this is even more important.

To handle the monitoring, a service called Datadog was integrated with the backend server, see Figure 10. This is a real time monitoring service which is integrated into the core of the operating system and keeps track of important metrics such as network requests, system load and memory usage. It was also integrated with the PostgreSQL database management system so it would

be possible to monitor the state of the database anytime. In the system it is also possible to set up alerts to immediately give information to the system administrator if anything is not working properly.

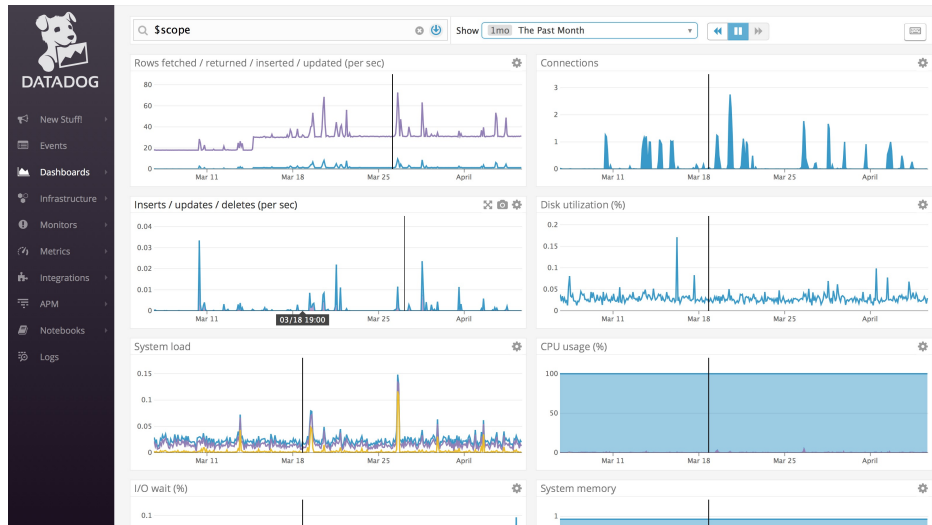


Figure 10: Monitoring dashboard for the database showing one month of usage. Graphs are visualizing rows fetched, inserts, updates and system load for the chosen time period. Created by author.

3.2.9 Development Tools

There are several tools helping with the development of the application, related to React Native and Redux. The most important of them is the stand-alone React Native Debugger which is built on Chrome Developer Tools. In the debugger it is possible to both see the output to the console as well as the current state of the Redux object and which actions are being dispatched.

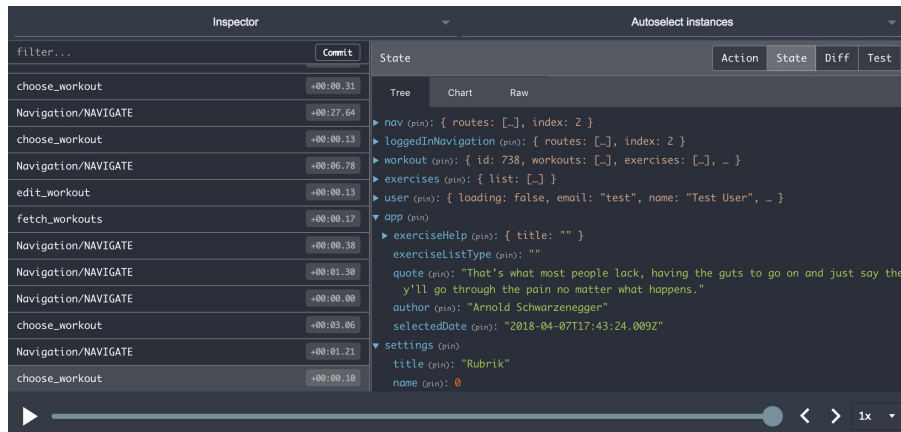


Figure 11: Redux Native Debugger showing the dispatched actions to the left and the calculated state to the right. The slider at the bottom makes it possible to perform time travel and is useful during debugging. By dragging the slider back and forth it is possible to replay actions and watch changes in the state tree. Created by author.

To be able to run the application without using a real device, simulators emulating Android and iOS was being used. This allowed for hot reloading of the application and a smoother development experience.

3.2.10 Deployment

The deployment process of the application needs to be as simple and fast as possible to make it easy to ship often. The frontend and backend are deployed separately since the backend gets updated on a more frequent basis compared to the frontend. The initial plan for the application was to only make five deployments of the frontend, once every second week in the beginning of the project and once every third week towards the end. This plan proved difficult to stick to since bug fixes often needed to be pushed as soon as they were ready.

To deploy the frontend a system called Expo was used. Using this system made it possible to continuously push new versions to test users without forcing them to download a new version from their application store. Towards the end of the project the application was deployed to both App Store and Google Play and the user's kept testing their version from there.

The backend was deployed more often and during development this was done several times a day. To enable this, a deploy script was written in BASH and linked as an alias on the server. Deploy of the backend demanded logging in to the server via SSH and running a command to start the deployment process. The deploy script starts with pulling down the latest version of the project from

the master-branch on GitHub. Then the Node package manager is used to install any new dependencies as well as the latest version of all the existing ones. It then stops the current running version of the backend in the process manager and boots up the new version.

3.2.11 Application State Management

Application state is a topic that might not need a lot of thought in the beginning, but as the application grows larger it quickly becomes a hurdle.

In React Native the most obvious approach is to store application state at the component level since it is already an integrated part of the framework. This is not a bad option if the state is only used in a few components that have a direct connection.

When the application grows, a lot of the state is not limited to a specific component but considered more global. For the application, the schedules

for a logged in user need to be available in several places in the application and may not be limited to one component.

The schedules need to be fetched from the backend at the moment the application boots up or in the case of user logging in. This introduced the problem of sharing state between components and at the same time avoid fetching too much data from the backend.

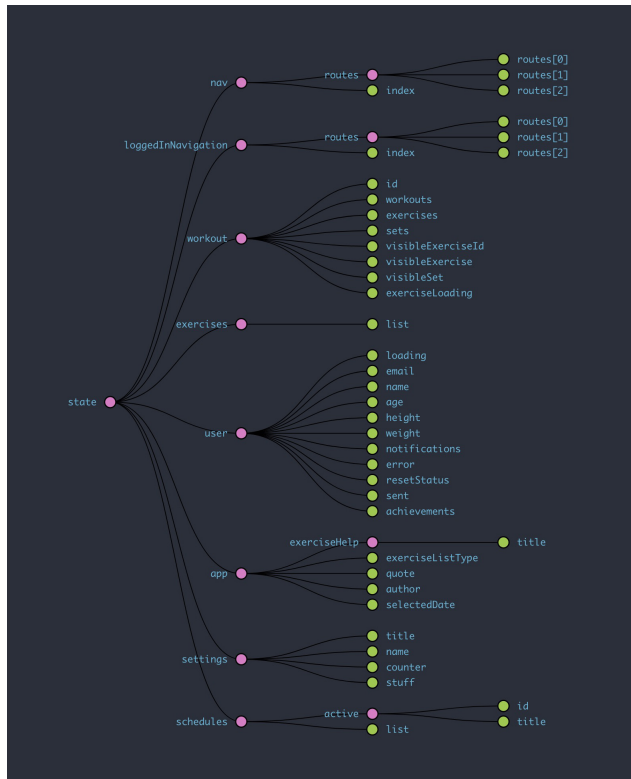


Figure 12: Illustration of the application state hierarchy. Created by author.

To solve this problem the design pattern called Flux was used. The specific implementation used in the application is a library called Redux. As described in section 3.3, Redux lets you define the state in a single store. A Redux store is a simple JavaScript object, globally available. The entire state of the application, including information about which view is currently active is stored in that object. To separate different sections of the store from each other and avoid name collisions, the store is separated into several different objects at the top level of the store object.

Each section of the store is handled by something called a reducer. A reducer consists of an initial state and a switch case statement. The initial state is what the state looks like before it has been altered by the application in any way. The switch case statement has one case for every action type that is relevant to that specific section of the store.

The actions that the reducers listen to are sent via action creator methods. An action is an object that has two keys, one for the action type and optionally one for the payload. The action type is a string that describes the action, and is what the reducers listen for. A payload may be attached to the action if it needs to pass on new data to the store.

When an action is being dispatched it passes through all of the reducers, but only the one listening to that specific action gets a chance to alter its state. The default case for every reducer is to return its previous state. When an action object hits a reducer that listens to it, that reducer is responsible for calculating the new state and returning it. The store object is seen as immutable. Altering the state therefore has to be done by allocating a new space in memory, not changing the data at the current location.

Handling asynchronous changes to the Redux state requires a little more thought than synchronous updates. For example, all actions that require communication with the backend are performed asynchronously. These actions work different in the way that instead of returning a simple object they are returning a function, which accepts another function called dispatch as its argument. Inside this function, any asynchronous code may be run. When it finishes, the dispatch function gets invoked with the action object as its only argument. In the case of a request to the backend, two calls are being made asynchronously. At first the user token is being fetched from local storage, then it is attached to the request made to the endpoint of choice. When the result comes back, dispatch is being invoked with the response from the backend as the action payload.

3.3 Gamification

Because of the evidences from the literature review that gamification does provide some form of motivation, unintentional or not, and the fact that it should

encourage both non-active users as well as active users, it was decided that it would be incorporated into the application.

3.3.1 Achievements

To make the application more entertaining, a variety of achievements have been used. These are gamification elements that are commonly used in applications in order to increase the user’s motivation (Glover, 2013). The idea behind this is to notify the users of their progress and triggering the feeling of being challenged.

Achievements are awarded to the users when small challenges in the application are completed. When completed, the user receives a badge, which can be seen in their *achievements* page. Every achievement has three different levels: bronze, silver and gold. Each achievement is a secret to the user until it has been fulfilled. The reason behind this was to create a fun and surprising feeling whenever a user receives one. Achievements are not only used to give the users more motivation to increase the amount of exercise, but also to encourage more diversity in the type of workout, and highlight potentially bad or harmful patterns.

One example of this is the “Spontaneous” badge, see Figure 13. This badge is earned when the user uses the “Quick start” function in the application.



Figure 13: “Spontaneous” badge for level silver. Created by author.

3.3.2 Calculating achievements

The calculation of achievements is done on the server side. A single endpoint is responsible for initiating the calculation, saving the result to the database and returning it to the client. Before deciding on the approach, the group discussed several different solutions. The most straightforward solution would have been

to calculate achievements each time a change is made to any of the workout related parts of the application. This would have included such things as setting a start and end time for a specific workout or adding an exercise to a workout. Even if this approach would be possible to implement, scaling it would be far too resource intensive. The next option was to split up raw calculations of achievements to several endpoints. An endpoint for a specific achievement would then only be called if there was a chance that a change in the state could lead to the user gaining an achievement. This option would be the least demanding in the sense of computing resources. Even though this approach did not make it into the implementation stage, it was foreseen that it would most likely introduce unwanted side effects and difficulty to keep the code clean and separated.

A middle way was to calculate all achievements at once and do it via only one request to the backend. Making a GET request to a specific endpoint together with the user's access token triggers a function in the backend. In that function, all achievements for that specific user are first deleted, then a list of promises are created and invoked. Each promise is responsible for a single achievement and is resolved when that particular achievement calculation is finished.

```
const calculateAchievements = (req, res, next, db) => {
  db
    .any('DELETE FROM achievements_user WHERE user_id = $1', [req.user.id])
    .then(() => {
      Promise.all([
        calculateNightOwl(req.user.id, db),
        calculateChickenLegs(req.user.id, db),
        calculateInstructionsRead(req.user.id, db),
        calculateSpontaneous(req.user.id, db)
      ]).then(() => {
        res.sendStatus(200);
      });
    });
};
```

Figure 14: Achievement calculation done with the Promise in JavaScript. Created by author.

This approach turned out to work. If the function is invoked when the user presses the achievement link in application, the achievements are most often returned without any noticeable delay for the user.

3.4 User Experience

The user experience part of an application is something that has to be considered early in the process and cannot be applied at the very end. It has to be weaved into the development.

3.4.1 User Testing

The target age of the users is around 20 to 30 years, because younger people are generally more accustomed to use mobile phones (Abeeel, Beullens, & Roe, 2013). Studies also indicate that they appreciate gamification more than older users (Bittner & Schipper, 2014). From that information, the project had enough ground to make the assumption that the target group should be young.

Since the main focus of this application has been the user experience, it has been important to get feedback from real users. To get feedback, test users have been used from an early stage of the project. Around 30 test users with an age ranging within the target age and with mixed experiences of training have been used. The mix of experienced users and beginners regarding gym workouts was chosen in order to receive feedback from users with different perspectives.

The feedback was collected by unstructured interviews with the test users, where the focus from the interview was the feedback that the user had. In order not to modify or influence the user's own ideas, the user did not receive specific questions but more general questions of their impressions and experiences of the application.

The intention was to release the application every other week in order to collect the feedback of every version from the test users. As mentioned previously in section 3.1.9, "Deployment", the application was deployed more often due to hot-fixes of bugs. The changes and upgrades made from the previous to the next version were based both on feedback from the users and on a plan created in the beginning of the project specifying what functionality should be included.

In order to get more extensive feedback on the final version of the application, an event was held on the campus of Johanneberg where the application was displayed. The testers were given a set of smaller tasks, such as creating an account, starting a workout and editing it. All the tasks are displayed in the list below and were performed in the order they appear from top to bottom in the list. The first sentence is the task that the user received, and the sentence in italics is what the tasks tested. The tasks represent the main functionality of the application and the user gets to test buttons and the layout of the different parts of the application.

1. Create an account. *Tests how easy the "Create account" screen is to understand.*
2. Start a workout and name it. *Tests the user experience of the start screen and the Edit Workout screen.*
3. Add a leg exercise and add a set. *Tests how comprehensive the user finds it to add and exercise and how easy it is to edit the exercise.*

4. Copy the set. *Tests the intuition of the copy set functionality.*
5. Delete the set. *Tests the intuition of the delete set functionality.*
6. Read the instructions of the exercise. *Tests the overview of the Edit Exercise view since the user is supposed to find the instructions icon.*
7. Set the difficulty of the workout and the end time. *Tests how easy and smooth the user can edit the properties of the workout.*
8. Check your calendar. *The user has to access the menu in order to find the calendar, which tests the intuition of the menu layout and placement on the screen.*
9. View your achievements. When did you achieve it? *The user has to go through the menu again and click on an achievement in order to see the details of it.*
10. Log out. *Tests how easy it is for the user to access the log out icon from the menu through the settings.*

Their performance during the tasks were observed and a lot of valuable feedback was received. The event resulted in a wide range of test users, from people who had barely set foot in the gym to people who go there on a regular basis. A lot of data was collected and over 100 people downloaded the application to their phones.



Figure 15: Poster from campus usability test. Created by author.



Figure 16: Test users on campus. Created by author.

3.4.2 Features and Statistics

A big part of the initial idea was to provide statistics and feedback to the user from their logged workouts in order to show the progress of the user's strength and possibly muscle growth. It was also meant to be used as a foundation for further gamification functionality in which challenges would be dynamically created based upon the previous results of the user. A few weeks in to the project this part was made a stretch goal as it was a lot to be implemented. It is still an important part of the project, but since the focus is on user experience there was not enough time to prioritize statistics.

A compromise was made and on the dashboard there is now a spider graph of the muscle groups that has been exercised in a workout. An example is shown in Figure 17.

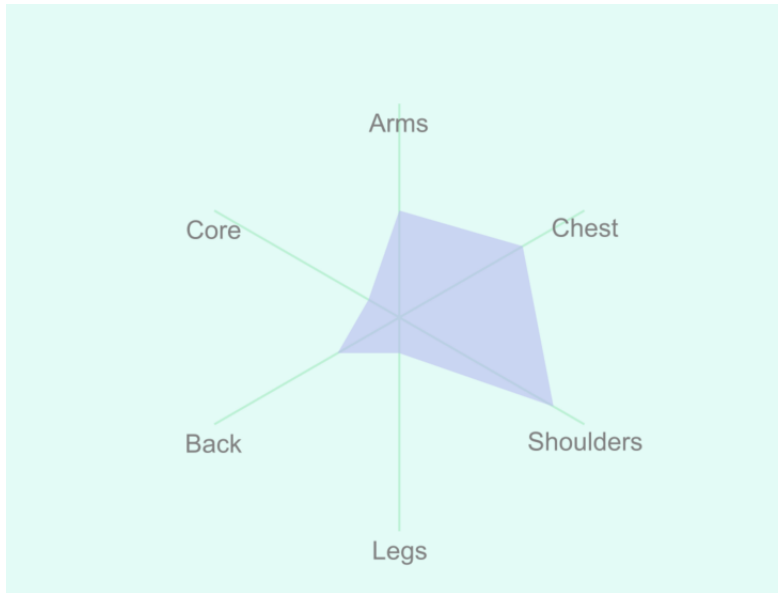


Figure 17: A spider graph representing muscles trained during a workout. Created by author.

3.4.3 Animations and UX

Making use of animations can enhance the user experience if done correctly. There are many small animations in play within the application in order to make user experience more pleasant. Some are more subtle while others are very distinct. Some examples of animations are the spinning achievements and the menu sliding out smoothly.

The UX design has had a primary focus which has been to make everything easy, intuitive and using as few taps as possible. At the same time the design needed to be coherent among all different views, both for making sure it aesthetically looks good as well as keeping the views intuitive.

3.5 The Website

A website for the project was needed for several reasons. The site was first created as a means of communication to the test users of the application. By using a website as a unified source of information of how to download the beta version was much more effective than using email or some other tool for communication.

The website was also necessary to give the users a place where they could reset their password. When a password reset request was made in the application, a link to the website with a predefined token was sent to the user's email. When

visiting the password reset page the token was validated on the backend and if a match was found the user was able to reset their password.

Later on in the project the website usability of the website was further extended by making it possible to log in to an admin page. On the admin page it was possible to add exercises and edit the description of them. Even though this functionality was only to be used by the administrators it made it much simpler to maintain the database of exercises. Before implementing this, all changes to the database was needed to be done by logging in the server and writing SQL queries.

The log in system for the website made use of the same credentials as for the application since the backend of the website was connected to the same database as the backend for the mobile application. When a administrator logged in to the website the JWT token was saved in local storage and was then attached to the header of every request. This prevented any unauthorized requests from being executed on the backend.

4 The Final Application

This part is about the resulting application and its layout, final design and functionality. The discussion whether or not something could have been done differently will be found in the discussion section.

The final product is the application, which is a gym workout diary where the user can create, log and view workouts and exercises they have done or plan to do. The name of the application is PushApp, referring to word play of doing a push up. Its logo is displayed in Figure 18. The application can be run on both iOS and Android and was launched on App Store and Google Play.



Figure 18: The logo of the application. The application name is PushApp. Created by author.

4.1 The Main Views

The application was divided into seven main views, where each view covers one core functionality. From the menu, see Figure 19, that opens when tapping in the top left corner in any of these main views, the user can maneuver to other views such as their calendar, profile settings, achievements, schedules and back to the dashboard view.

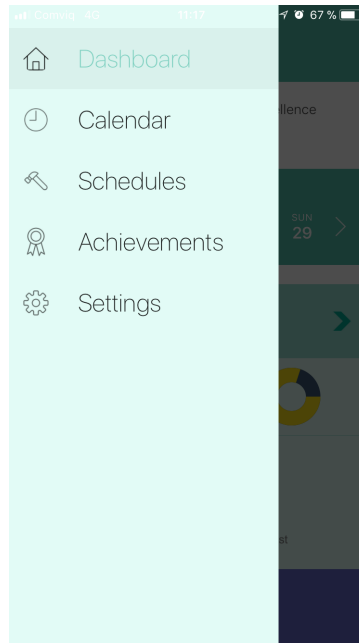
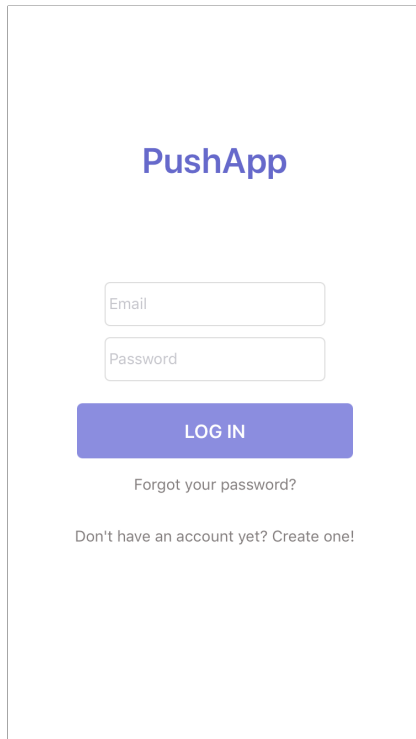


Figure 19: The side menu where the user can access the Dashboard, Calendar, Schedules, Achievements and Settings. Created by author.

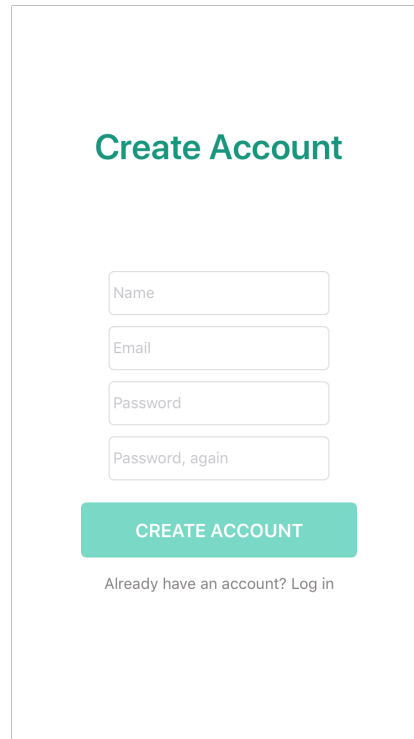
4.1.1 Log In/Create Account

The *Log in* view is the default view that shows on start, see Figure 20. The user can create an account by clicking on “Create account” beneath the “Log in” button and the *Create account* view shows, see Figure 21. Once logged in, the user stays logged in between sessions, until they tap the log out button that is found in the settings view. After logging in, the user is directed to the *Dashboard*, see Figure 22.



The image shows a mobile application login screen. At the top, the text "PushApp" is displayed in a blue font. Below this, there are two input fields: "Email" and "Password". A blue button labeled "LOG IN" is positioned below the input fields. Underneath the button, there is a link that says "Forgot your password?". At the bottom of the screen, there is a text prompt: "Don't have an account yet? Create one!".

Figure 20: The Log in view. This is the first page that the user sees when starting the application. Created by author.



The image shows a mobile application account creation screen. At the top, the text "Create Account" is displayed in a green font. Below this, there are four input fields: "Name", "Email", "Password", and "Password, again". A green button labeled "CREATE ACCOUNT" is positioned below the input fields. Underneath the button, there is a link that says "Already have an account? Log in".

Figure 21: The Create Account view. Once the account is created, the user is logged in. Created by author.

4.1.2 Dashboard

The *Dashboard* is the main view of the application. The user is greeted with a randomized motivational quote fetched from the backend and they can see and access a summary of the previous workout properties, see Figure 23. If no workouts have been completed yet, the user sees a welcome message instead of the workout summary, see Figure 22. The calendar strip allows the user to swipe to see previously logged workouts. The dashboard is also the view where the user can create new workouts.

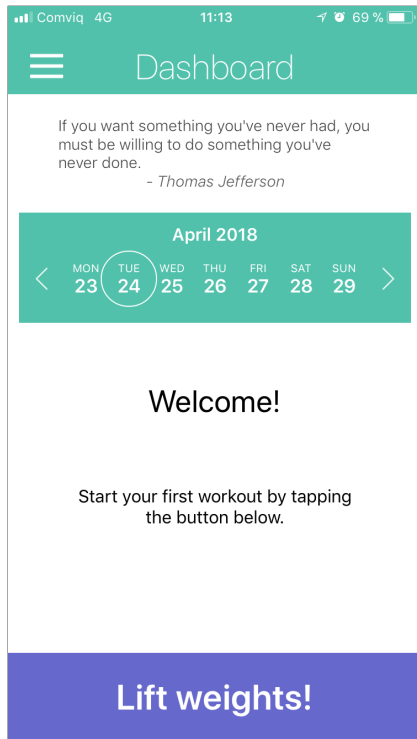


Figure 22: This shows the first time that the user logs in to PushApp. The Dashboard is currently empty, and some instructions on how to get started serve as a placeholder. Created by author.

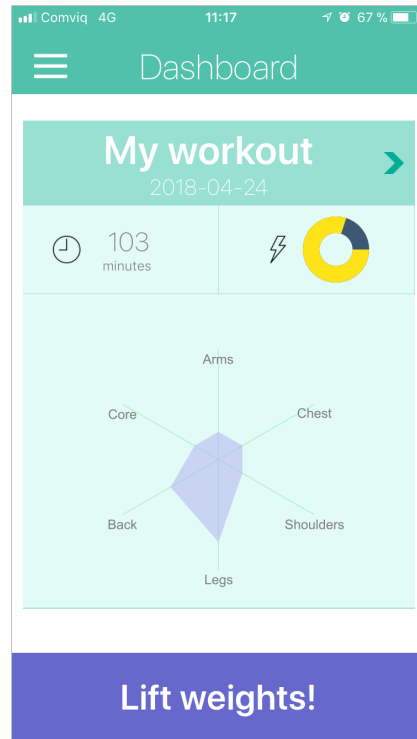


Figure 23: When the user has created and ended a workout, the Dashboard looks like this. It now shows a summary on the last workout that the user made. Created by author.

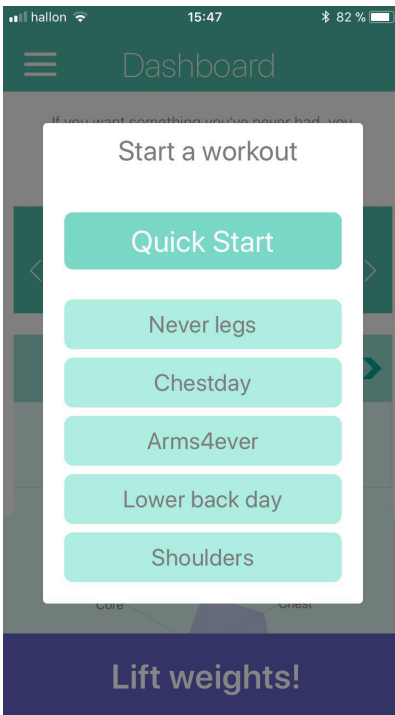


Figure 24: The pop up showed when tapping the “Lift Weights!” button. The user sees previously created workout templates and a “Quick start” button. Created by author.

The “Lift weights!” button is where the user can create workouts. When pressing this button, a pop up will show where the user can start a workout either by choosing one of their existing workout schedules or start a new empty workout by pressing “Quick Start”, see Figure 24. Both of these options will direct the user to a view where they can edit the workout, see Figure 25.

4.1.3 Workout

In the workout view, the user can edit a workout, see Figure 25. The properties of a workout are the name of the workout, the date, the start time and stop time, a list of added exercises, a color tag, a difficulty level and the user’s notes. All of these can be edited by the user. The color tag is used in the calendar to separate the different workouts from each other. The difficulty is displayed with images of five flexing arms where the scale goes from the first arm, “No sweat”, to the fifth arm, “Hellish”. The marked arms are highlighted and a short animation plays that makes the arms “pop”. The selected difficulty is displayed as a progress bar in the dashboard, see Figure 23, which fills more with every highlighted arm.

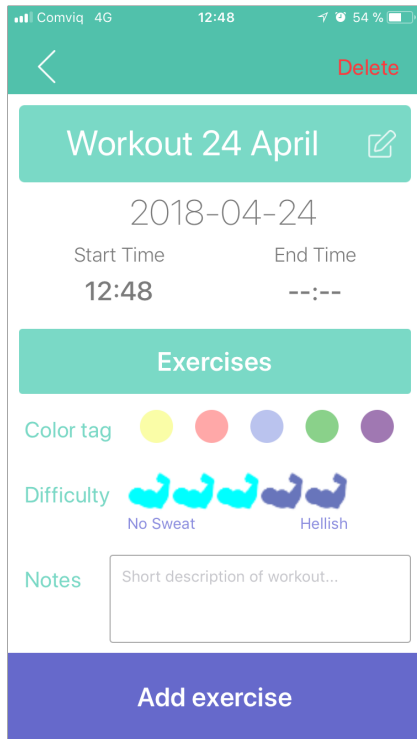


Figure 25: Create a new workout. This is the view that shows when the user taps “Quick start” or one of the workout schedules. Created by author.

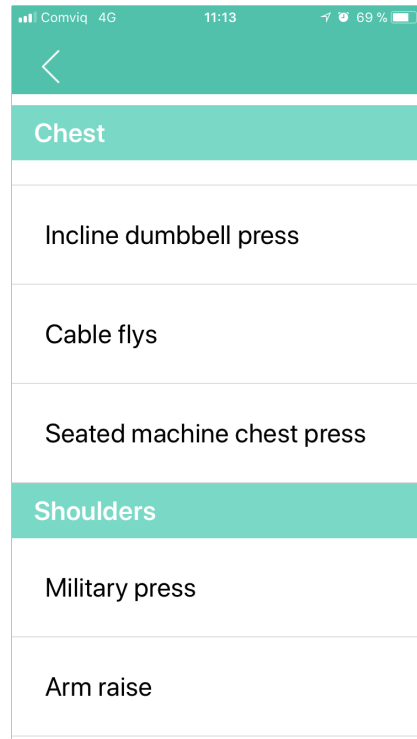


Figure 26: A part of the exercise list. The exercises are divided by which body part that is the main target of the exercise. Created by author.

When the user taps “Add exercise” a list of exercises shows, see Figure 26. In the list, the exercises are categorized by which body part they exercise. The user is not able to add any exercises to the workout that are not in this list.

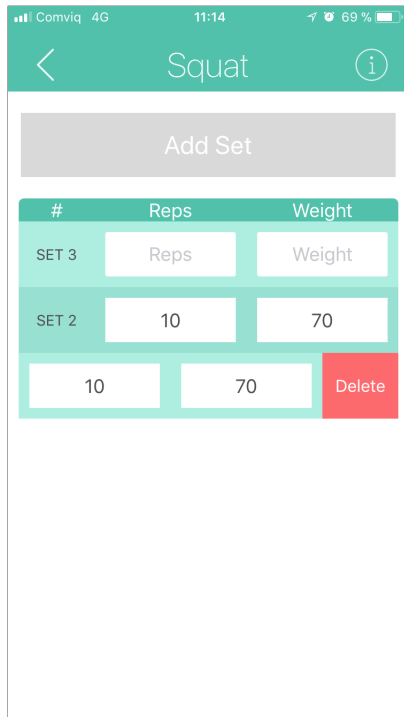


Figure 27: The edit exercise view. The user can swipe the added set to the left and the option of deleting the set is displayed. Created by author.

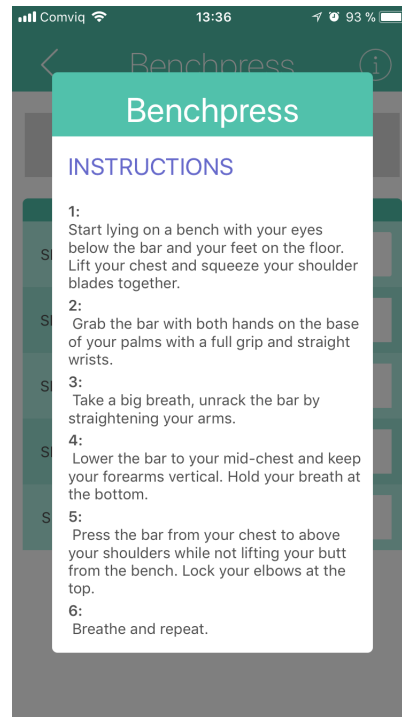


Figure 28: The instruction for an exercise. The user gets step by step instructions and tips on how to perform it. Created by author.

For each exercise the user can edit the number of sets and repetitions, as well as the weight in kilograms for each set. The sets can then be copied or deleted by swiping left or right respectively on the specific set, and more sets can be added by pressing the “Add set” button, see Figure 27. The user can also see instructions for the workout by pressing the information icon in the top right corner, which displays a pop up with the instructions, see Figure 28. The instruction is a numerated list, showing step by step how the exercise is performed. Some of the exercises that might be more difficult to the beginner also has a tip.

When a workout is done, the user enters the stop time and the total time of the workout is saved and showed in minutes on the dashboard. The information from the logged time, the difficulty and the categories of the exercises in the workout are used for the gamification part where the user can obtain different achievements. With the back arrow in the top left corner of the workout view, the user is directed back to the dashboard, again see Figure 23, where the latest workout is updated.

4.1.4 Calendar

The calendar is shown in Figure 29. The user can see an overview of previous workouts and the details of them by tapping the date they were carried out, see Figure 30. This is also where the color tags from when creating the workouts are used. These can be used to for example identify different types of workouts.

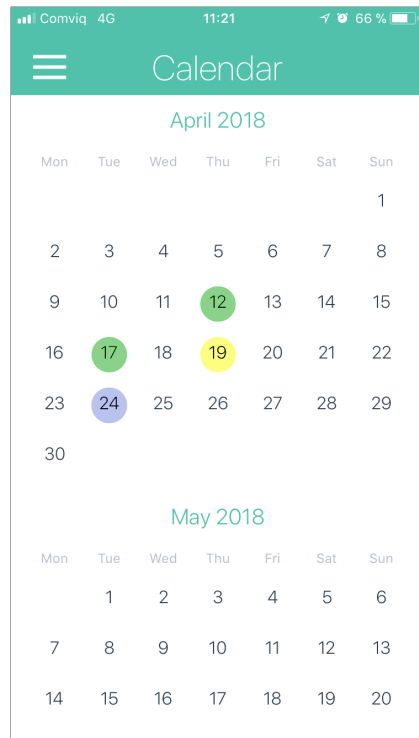


Figure 29: Calendar view. The colored dots are previous workouts. If the user has selected a particular color for the workout, that color is shown here.

Created by author.

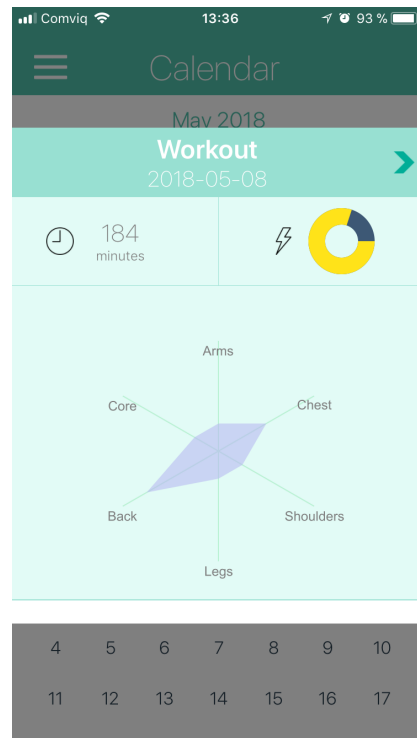


Figure 30: The details of a previous workout showing the duration, difficulty level with color showing difficulty as well as a spider graph visualizing targeted muscle groups.

Created by author.

4.1.5 Schedules

From the menu, the user can also go to their schedules page, see Figure 31. This is where the user can create their workout templates that they can choose from the “Lift Weights!” button in the Dashboard when starting a workout. The schedules can be named and the user can add exercises from the same list of exercises as shown before, see Figure 26. By swiping the exercise to the left, the user gets the option to delete it.

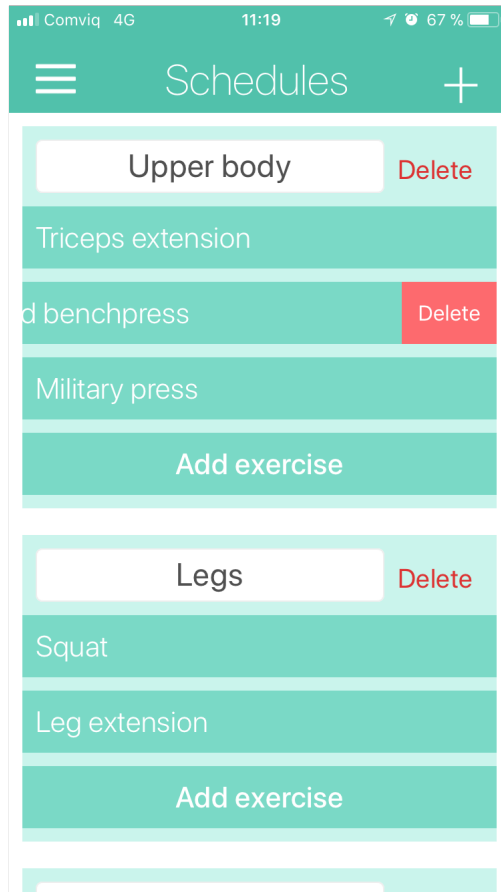


Figure 31: Schedules view, where the user's workout templates are created and edited. Created by author.

4.1.6 Achievements

When completing different achievements, the user can earn badges. These are displayed on the *Achievements* page, see Figure 32. When tapping on the badges, the application displays a popup with more detailed information about the specific achievement, see Figure 33.



Figure 32: The achievements view. This particular user has achieved all levels of all achievements. Created by author.

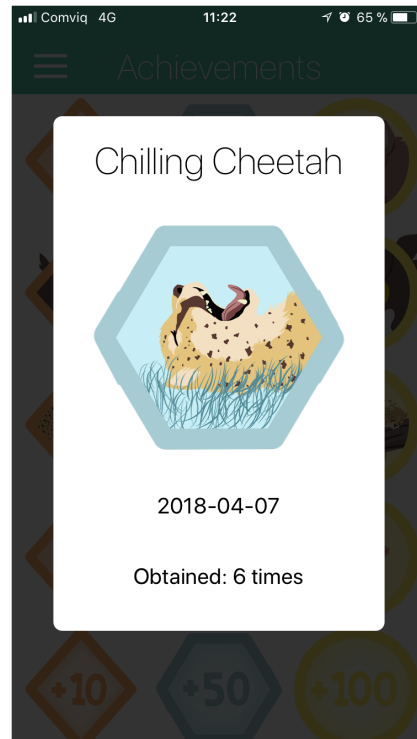


Figure 33: Description of the achievement “Chilling Cheetah”. The pop up is shown when the badge is tapped. Created by author.

4.1.7 Settings

The last thing in the menu is the settings, see Figure 35. Here, the user can update their account information and the name they want to display.

When pressing the “SAVE” button, the information is checked, and if there are any violations, a message is displayed and the corresponding text field is highlighted to make it easier for the user to understand the problem, see Figure 34. This happens if the name is empty, the email is not valid, the new password is less than six characters or the two passwords does not match.

With the button in the top right corner the user can log out from the application. This will take the user back to the log in view, see Figure 20.

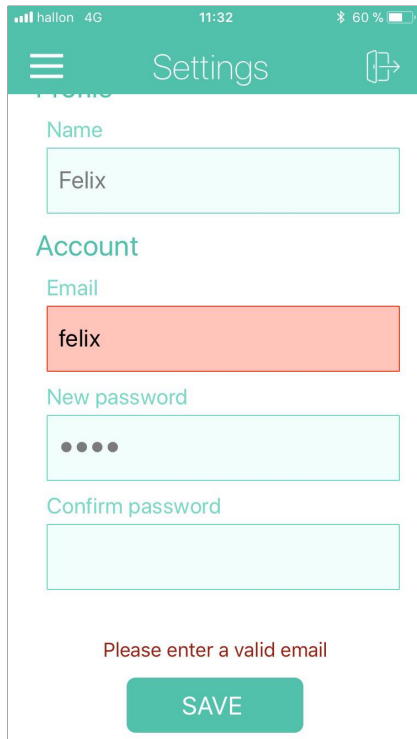


Figure 34: Feedback given to the user when the email entered does not have a valid format. Created by author.

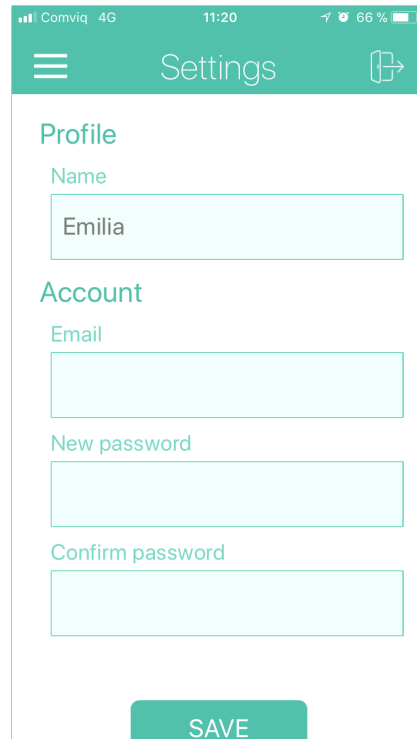


Figure 35: Profile settings, giving the user the ability to change name, email and password. Created by author.

4.2 Navigation

The navigation in the application has been divided into two major parts. The first one is the navigation before logging in which enables the user to log in, create an account and resetting the password. The second part handles all navigation inside the application once the user has logged in.

The latter part has been divided into two sub systems for navigation. It is common for mobile applications to use a menu bar in the bottom of the application allowing users to navigate quickly to the different views. In this application, the menu is more similar to the menus used in websites adapted for mobile phones. When a user taps the menu icon, a menu slides in from the left revealing the different navigation options, see figure 19 .

The items placed in this drawer menu was chosen because of the flow of the application, a user is only able to access the top level of a functionality from the drawer menu. The user may access sub views related the functionality of

the top level once inside there.

4.3 Design

The design mockups that were drawn in the beginning of the project provided a vision of how the application was going to look. In the end there are still some similarities to those sketches, but some things are different. A couple of important choices has remained constant from the start however, which includes the color scheme as seen in Figure 2. These colors help making the components clear and simple, leaving a lot of space inside the application. The functionality described in a previous section was layed out according to the idea of a light and bright impression of the application.

4.4 Gamification and Animations

The gamification part of the application consists mainly of achievements where the user is rewarded with badges when fulfilling certain requirements. Each achievement is connected to its own badge, for example the user can get the “Night owl” badge when exercising several times between 22 in the evening and 6 in the morning. These achievements have three different levels: bronze, silver and gold. They are rewarded when obtaining the achievement multiple times and the levels are shown in the achievements view.

The achievements currently available for the user are displayed in the list below. The badges seen in Figure 32 were designed to reflect the key property of the achievement although some are not visible in the picture.

- Night Owl - The user has worked out at night time.
- Chilling Cheetah - The user has not worked out for some time.
- Chicken Legs - The user has not trained their legs for some time.
- Badge Magnet - The user has collected several badges.
- Scholar - The user has read the instructions on some exercises.
- Spontaneous - The user has used quick start several times in a row.

5 User Experience of the Final Application

In this section the second part of the results will be covered, namely the user experience of the final application. This covers the test users and the feedback obtained from the test users and focuses on the actual results from the user experience and gamification research. The outcome from the motivational elements in the application and their effects on the test users will be further examined in the discussion section. The results from the user experience research part are split into the feedback received from the test users and data gathered from people testing the final version of the application.

5.1 Feedback From Test Users

During the course of the project, the test users provided feedback on the application regarding functionality and design. Some of the feedback is shown in the list below, where we chose the feedback that trended over many of the test users. This means that the list is a summary of things that several of the users pointed out. The feedback is from the test on campus and from the test group:

- Some difficulty to start a workout.
 - The functionality of the “Lift Weights!” button was not obvious or intuitive.
- Difficulty to copy a set.
 - A bug on the Android version caused the whole view to swipe instead of just the item.
 - The user is accustomed to tapping and holding to copy something which made the swipe gesture far from obvious.
- When changing the name of the workout, the user tries to tap the icon.
 - A flaw in the Android version makes the view scroll so much that the user could not see what they wrote.
- If the information entered when creating an account was incorrect, visual feedback was given which the users appreciated.
- When trying to add an exercise, it was difficult to find the button. It seemed like the title in the middle of the view should be tapped instead of the actual button. The users were confused when that did not work.
- The users wanted to see which achievements were possible to obtain. The animation of the badges was cool.
- The categories in the exercise list could have been organized in anatomical or alphabetical order so that it is easy to find the exercise quick. The user wanted to be able to add their own exercises.

As seen in the list, the main part of the feedback regards the design parts of the user experience such as the layout of buttons and icons. The use of non-standardized gestures, such as swiping to copy, made it less intuitive. Analysis of the feedback is discussed in Section 7.

5.2 Measured Results

In total we had 70 test users on the campus tests. The user experience results is measured in how much struggle a user had when performing a couple of actions in the application. The results have been summarized in Table 1 and divided into three categories: “No problem”, “Slight problems” and “Needed help”.

The task was labeled as “No problem” when the test user found the right functionality without hesitation and completed the task with no issues. The “Slight problems” category was when the test user tried a few times before completing the task but did not need help. The last category, “Needed help” was when the test user required an explanation and/or a hint in order to succeed. The numbers in the table represent the total amount of test users who completed the task in the category.

There is a tendency in the table that shows that most of the users had no problem when creating an account, setting the difficulty of a workout and viewing their achievements. On the opposing spectre many users struggled with copying the set and needed help. The tasks where most of the users had “Slight problems” were creating a workout and changing its name, add the exercises and checking the calendar. Overall the majority of the users ended up in the “No problems” or the “Slight problems” categories.

Task	Test Score		
	No problem	Slight problems	Needed help
Create account	56	7	4
Create workout	33	25	9
Change name of workout	26	14	17
Add leg exercises	7	16	6
Add reps and weight	22	38	7
Copy the set	6	26	36
Set difficulty	43	17	7
Check calendar	24	14	10
View Achievements	56	9	2
Log out	22	30	15

Table 1: Displays the results from the campus test. Some tasks does not contain data from all test users, because some users were not asked to perform all of the tasks due to bugs on Android.

6 Societal and Ethical Aspects

Since the user experience analysis in performed via a workout application, there are a couple of aspects that needs to be lifted. The application itself has to meet a couple of standards regarding personal information, but the user experience must also be regulated in order to not provoke negative behavior.

6.1 Integrity

The application needs to follow the laws and regulations concerning the personal data and integrity of the users. In May, 2018, a new law concerning personal data, named GDPR (General Data Protection Regulation) will replace PUL (Personuppgiftslagen).

Personal data concerns information that could identify a person, i.e. personal number, email, name, address etc. (Datainspektionen, 2018). In the application the email is collected along with a display name. The display name does not have to be the user's real name, but the email could possibly be used to identify the user.

Sometimes personal data could be considered as sensitive. Sensitive personal data is information such as the user's ethnic background, political opinions, sexual orientation, health etc. (Datainspektionen, 2017a). It is forbidden by law to collect that data, unless you have the user's explicit consent (Datainspektionen, 2017b). In the application a user's health could be deduced by accessing the data collected from workouts, achievements etc. Because of this, it will be considered as sensitive data.

To face the challenges concerning personal data, the users will have to sign an agreement that states what data is being collected and how it is used. This privacy policy, see appendix A, also says that by using the application, the user has agreed to the application collecting and storing that data. This agreement is signed in the process of creating an account. If the user does not sign the agreement, they will not be able to create an account in the application.

It is also important to make sure that the data that is collected is kept secure and private. This can be done by forcing users to choose strong passwords when they create their account (i.e. the password must contain at least a number, an uppercase letter and have to be at least 6 characters long), take measures against data breaches and have a protocol to follow if a data breach should occur. In this case the application checks that the password is at least 6 characters long.

6.2 Ethical Challenges

With the introduction of gamification into the application, challenges such as game addiction and health- and workout related diseases comes into play. For example, a user might experience unhealthy pressure to exercise due to prompts and notifications from the application. It is also relevant to consider that the effects of game addiction might cause the user to exercise in an excessive, unhealthy manner in an attempt to get a "better score" or to earn more achievements.

To counter this, the application has introduced achievements that encourages the user to rest in-between workouts. The application has also decided to skip notifications and prompts that reminds the user to exercise. This might seem counter-intuitive, since the goal of a workout application often is for the user to live a healthier life, but too much exercise can be as unhealthy as not exercising.

A user might feel the need or compulsion to get strong quicker. It is well known that building muscles is something that takes time. If the user then tries to lift weights that are too heavy or tries exercises that they are not familiar with, they might unintentionally harm themselves. This could lead to serious injuries. To counter the risk of harm, the application introduced the exercise instructions to every exercise in the application. Then the user is informed of how the exercise is carried out correctly.

Despite the ethical challenges mentioned above, they are believed to be outweighed by the possible advantages of the application. However, precautions have been taken in order to minimize the possible negative effects at least in some ways.

The application does not keep track of weight, age or height because it may be sensitive information for the user. They might experience negative emotions that could influence eating disorders or anxiety on their appearance.

7 Discussion

This section will analyze and discuss the results from the previous section. Firstly, the resulting product is evaluated and discussed. Following that discussion is the analysis of the user experience, which is where the test results and feedback is discussed. Finally, future directions and improvements will be discussed.

7.1 Analysis of The Resulting Product

The application was designed to be appealing to the eye and give the users a sensation of light and space. When deciding the color scheme, the goal was to provide colors that matched well together and that produced positive emotions associated with working out. The colors from the palette, see Figure 2, enhance certain emotive experiences such as calmness, freshness, joy, power and vitality (Meerwein, Rodeck, & Mahnke, 2007).

The initial design, see Figure 3 to 7, is different when compared to how the design actually turned out. The general idea is still there: there is white space to allow plenty of breathing room and many concept and ideas are still present, although a little bit modified. Taking the “Dashboard” part of the application as an example, one can see similarities between the mockups and the final de-

sign. For example, the actual application still has a calendar strip as well as today's date's workout.

Although there are still similarities, there are also differences. The reason behind this is lack of time as well as some development issues. Many ideas, like the statistics part of the application, had to be abandoned and dropped due to the time sensitive nature of the project. But in the beginning of the project, when the mockups were prototyped, the project was mainly concerned with what could be added rather than what could be done. This was a good way to brainstorm and come up with ideas for the application, even if every idea did not make it to the final application. Some of the ideas and designs did not make it due to them being too time consuming to actually implement. There were also times when features and design choices did not work out as intended when actually trying them out. Because of that, they were either dropped or modified. An example of this was the header in the dashboard view, see Figure 3. Although it was aesthetically pleasing, the functionality presented in the drop down header was no longer part of the application and was therefore altered to the static header, see Figure 8 and the header in Figure 22.

The functionality in the final version of the application were, according to the feedback in Section 5.1, said to be sufficient and fun. There is however still room for improvement and additions to the functionality. The application is focusing on gym workout, but not everyone is doing their workouts in the gym. Some people prefer cardiovascular training or sports, and in order to receive more helpful results these could be included in the application. If the types of workout the user can choose from matches their preferences, their motivation might be affected in a positive way. Other possible features to add to the application are diets, calorie counters and meal suggestion, all which can have a positive impact on a person's health. Many of the test users thought that it would be fun to be able to add their own exercises to the application. This could help the application to fit a broader audience, and in that way improve the user experience further. It is however important that the application has a focus. If we were to implement all these functions the application's broadness would increase, but on the other hand the broadness would perhaps diminish each functionality.

The calendar with the color tags was intended to be a good way for the user to see which workout they should do next, since different color tags could be used for different types of workouts. They could be a possible motivator since the dots in the calendar makes it easier to get an overview of workout habits. However, this is something that would require more research.

The gamification in the application consists only of achievements that the users receives when completing a certain task. From the beginning, the application was supposed to have more gamification features, such as being able to take part in challenges with friends and family, being challenged by the app itself

and viewing other people's progresses. Unfortunately many of the features had to be dropped. Had more gamification features been included, the result might have been different.

The randomized quotes on the Dashboard are meant to be inspirational and motivating to the user, see Figure 22. They serve as an addition to gamification and the graphical effects when it comes to enhancing the user experience. In the effort to make the application become more alive, there were plenty of other ideas discussed as well. One of them was personalized greeting phrases in addition to the quotes. These phrases would have been different depending on how the user interacted with the application. The reason for abandoning this idea was because the dashboard became rather cluttered which did not resonate with the design of the application.

The need for a website arose as the application was mirrored through another application called Expo, which was difficult for the users to set up properly. It started off as a guide in how to install the application and it did not take long before other usages were found. When the test users forgot their password we were able to change it through communicating directly with the database, however since this required active work from our side we decided it would be easier to automate it. Thus came the password resetting feature of the website.

The most helpful feature for the developers was the forms in which it was possible to enter new exercises and their descriptions. Even though it was time consuming constructing this feature, the scalability of it saved time. See Figure 36 for a view of how it looks.

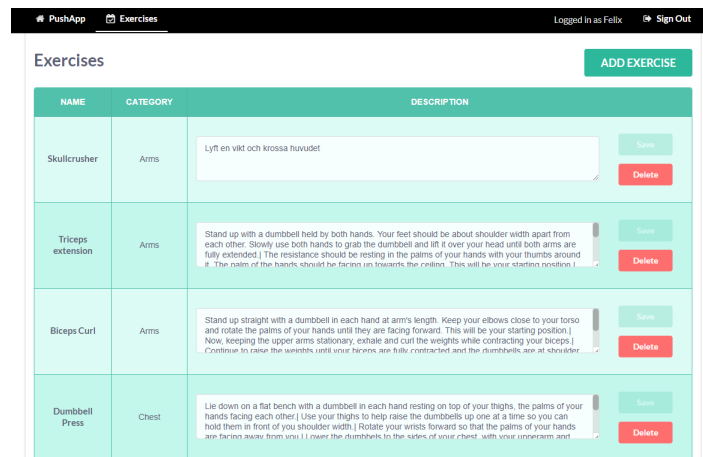


Figure 36: The view of the website where exercises are added or modified. Created by author.

7.2 Analysis of The Resulting User Experience

The results from the Table 1 indicate how intuitive the users found the different parts of the application. Something that might explain the results on the create account and log in views could be that they are both neat and only have one main functionality each, which is centered in the view. Creating an account and logging in is also something that the user has seen before in almost every application available on the market so that they know what to do immediately. It seemed that most of the users found it easy to understand the dashboard layout, even though some had slight problems with understanding the button. The reason for this could be that the title of the button was “Lift weights!” which might not have been intuitive to the users. The achievements view is accessed through the menu, which could suggest that if the menu is easy to understand, the task is simple.

The examples above state what the test users seemed to view as intuitive. An example of what the users found difficult was copying a set. A reason for this could be that the standard on Android and iPhone is to tap and hold the item that one wants to copy—not swiping it to the right. Another reason that could have affected the result is that when the Android users tried to swipe to copy, they could swipe to the next view—something that was not supposed to happen. If the Android bug had not been present the test users might have figured it out sooner.

Regarding the design, many of the users enjoyed the colors and the freshness. They also liked the graphical effects, such as the spiderweb graph on the dashboard and the animations on the popping difficulty arms. Overall it seems that these vivid and moving details of the design enhances the user experience and attracts the user’s attention. The special effects were fairly balanced according to the feedback. Many of the users agreed that the achievements improved the total experience of the application and would probably serve as a motivator, which was in line with the results from the studies in the literature review in Section 2.

Something that seemed to be less of a good experience for the user was the buttons that were placed in the bottom of the view, such as the “Lift Weights!” and the “Add exercise” buttons, see Figures 23 and 25. They thought that the label “Lift weights!” perhaps should have been “Start workout” and that the buttons should have been placed towards the middle of the view in order to attract focus to the button.

Regarding the gamification part, the users consistently thought that the badges were a fun part of the application that made it different from many other health and fitness application. Some suggestions of what might motivate the user more could be to display obtainable achievements. This might have raised the feeling of being challenged as well as explaining that there in fact are different levels of

a badge.

A couple of graphical details appeared to be confusing, such as the log out button. Some of the test users believed it was a button for sharing the experience on a social media, and thus did not understand where they could log out from the application. Another confusing part was when the user tried to edit the name of the workout. They were supposed to tap on the name, but some of them found it more intuitive to tap the icon, see Figure 25. This suggest that users expects icons to be interactive. Another aspect that could have had an impact on the result is that some of the icons were interactive while others were not. The learning outcome of this was that it is important to keep visual elements within the application consistent. The use of suitable icons in the correct places in this context seems to be something that improves the total user experience and usability.

To summarize the user experience part of the application there are some flaws and room for improvement. Overall the feedback from the users was mainly positive.

7.3 Research Methodology

As there was quite a short period of time to develop and evaluate the application we felt the need to involve test users at an early stage. In hindsight this might have been the wrong decision as the test users became involved in a stage where the product was not yet ready to be evaluated. There were plenty of bugs that were obvious and this got the majority of the focus from our test users who then reported already known bugs. The priority could instead have been to have a smooth running and bug free application with less functionality in order to receive feedback on those parts that were fully implemented and working.

In order to get more accurate results, it might have been appropriate to do a follow up test with a larger number of users to measure and reassure the results compared to the previous test. We would then also have received feedback on the components that were implemented or changed since the previous tests.

Another stretch goal was to implement additional gamification features such as challenges and titles linked to these challenges. The idea was that the challenges would be based on the user's previous performance and tailored so that the user only needed to make a relative increase in some area. This could have provided another dimension to the application and perhaps the competitive aspect could have been included as well. The intention would have been to encourage the user on an individual level.

7.4 Contribution

While this project does not offer any new insights to the research in question, it does provide an interesting point of view. By utilizing the knowledge and theories that are established and available, this project manages to provide a more hands on experience to the subject in question. By doing so, it is possible for others to build upon the findings and perhaps even provide new insights to the subject with help of this project.

7.5 Future Direction and Possible Improvements

For future research one thing to consider could be performing a pre study with both personas and test users to learn more about the user requirements. As mentioned before, the test users were not as involved as intended. A good way to have involved them would have been to conduct some usability evaluations with the mockups, which could have lead to more commitment from the users.

If a pre study was to be conducted, it would be beneficial to increase the number of test users and perhaps even conduct a major survey. Since this project only had around 30 regular test users, it was hard to justify the results from a user point of view. Because of that, a more significant number of test users might help future projects justify their results in a more reliable way.

The decision to use the same files when developing for both Android and iOS lead to problems later on in the project. Since 80% of the project members were developing mainly for iOS, the development for Android was not as prioritized. Some components and their properties that worked fine for iOS did not work for Android. Therefore, we needed specific code that did the same thing for the Android version. With this being the case, the neglect of Android led to a number of different bugs that were only found some time after a feature had been finished. This made it more problematic to solve them.

A better approach could have been to develop the Android and iOS versions separately to avoid confusion and increase performance. It would also have been easier to determine when a feature was implemented correctly if they had been separate.

Developing the applications in the native languages, Swift for iOS and Java for Android, would have increased the workload significantly. On the other hand, it would have increased the likelihood of having best practices used for each operating system instead of focusing on having the exact same user interface.

For the purpose of investigating what makes an application appealing, this project might not have needed to implement a complete working application. It could have sufficed to perform tests with paper prototypes and to develop

a model for the user interface. This would have saved some time that could have been used to focus more on the features of the application and developing different models and have test groups that received different models in order to have more precise and detailed results.

7.6 Required Competence

There are several competences needed to develop a mobile application. Some of these are listed below. The competence of the group was broad enough and sufficient for the project.

- **Design** - To create a good user interface some basic design skills are needed. It is good to have some knowledge of illustration programs such as Photoshop, Gravit Designer, Gimp etc. The most needed skill, however, is a natural instinct for good design. It is also good to have some knowledge about standardization within the system.
- **Frontend** - There are plenty of ways to implement the frontend and knowing at least one of them would decrease the time spent.
- **Backend** - The backend is used to expose the database layer via a unified API and make it possible for the frontend application to communicate with the server. This requires knowledge of server-side technologies such as building an HTTP server, developing an API and general knowledge about the operating system Linux.
- **Database** - Implementing the database layer requires familiarity with a query language such as SQL as well as knowledge of how to create a viable database schema.

8 Conclusions

The purpose of the project was to develop an appealing mobile application for gym workouts while focusing on user experience. Although the purpose might not contribute directly to society, the results of the project still provide an interesting perspective. It gives other researchers the possibility to further their own investigations.

The research questions answered in this report were:

- What previous research has been made on user experience and gamification in the means of motivating a user?
- Which are the elements in the application that seems to be less or more appealing to the user?

Regarding the first question, there have been several studies on gamification and user experience. According to the studies, gamification seems to have the

power to motivate users. This is especially true among the younger audience. Even though there are many studies on user experience, it is difficult to conclude what is considered good user experience. This is due to user experience being subjective and reflects the complexity of human psychology.

The results we have found regarding the second question comes from feedback from the test users along the way as well as from performing a final test with more users. The users associated the color choices with positive feelings, and the achievement badges as a fun part of the application. There was also positive feedback regarding the graphical effects and the simple navigation.

Users were confused about the interactive parts of the application, for example the untappable icons that were not consistent throughout the application. Another problem that most of the users experienced was the copy feature which used a rather unconventional swipe gesture. This was unclear to the majority of the users.

React Native, the chosen framework, gave the opportunity to develop for both Android and iOS simultaneously. Using this approach we were able to share a large part of the code base between the two systems. It was also possible to write the frontend application for both systems in the same language as the backend. This kept the learning curve at a manageable level.

The gamification parts of the application were appreciated by the test users but the actual usage was too small to see if it actually made any difference to their workout habits. Conclusions can be drawn from both interviews with the test users and the test on campus that a compelling user interface makes the application much more likely to be used on a regular basis.

Including test users in an early stage was a mistake that slowed down the development process and shifted the focus from developing a solid product to shipping hotfix of the code. The lack of a separate backend server for development made it difficult to introduce breaking changes to the API without disturbing the experience for the test users. A better approach could have been to always have two different backend servers, one to run the latest development version and one intended to be used in production.

For future research in the field an improvement to be made should be to increase the amount of test users in order to get more reliable data. It would also be a good idea to include more measurable variables.

The project outcome was an application released publicly to the App Store and Google Play. Valuable insights were gained on how user experience can make a difference for a mobile application.

References

- Abeele, M. V., Beullens, K., & Roe, K. (2013). Measuring mobile phone use: Gender, age and real usage level in relation to the accuracy and validity of self-reported mobile phone use. *Mobile Media & Communication*, 1(2), 213–236. doi:10.1177/2050157913477095
- Althoff, T., White, R. W., & Horvitz, E. (2016). Influence of pokémon go on physical activity: Study and implications. *Journal of Medical Internet Research*, 18(12), e315. doi:http://doi.org/10.2196/jmir.6759
- Bittner, J. V. & Schipper, J. (2014). Motivational effects and age differences of gamification in product advertising. *Journal of Consumer Marketing*, 31(5), 391–400. doi:10.1108/JCM-04-2014-0945
- Datainspektionen. (2017a). *Känsliga personuppgifter, uppgifter om brott och personnummer*. Retrieved May 7, 2018, from <https://www.datainspektionen.se/dataskyddsreformen/dataskyddsforordningen/kansliga-personuppgifter-uppgifter-om-brott-och-personnummer/>
- Datainspektionen. (2017b). *När känsliga personuppgifter får behandlas*. Retrieved May 7, 2018, from <https://www.datainspektionen.se/dataskyddsreformen/dataskyddsforordningen/kansliga-personuppgifter-uppgifter-om-brott-och-personnummer/nar-kansliga-personuppgifter-far-behandlas/>
- Datainspektionen. (2018, February 26). *Datainspektionen* [Video File]. Retrieved from <https://www.youtube.com/watch?v=4lVLRMpoq70>
- Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining "gamification". In *Proceedings of the 15th international academic mindtrek conference: Envisioning future media environments*, 2011. Tampere, Finland, pp. 9-15. doi:10.1145/2181037.2181040
- Dyson, M. C. & Jennings, E. M. (2014). Examining the Interfaces to E-journal Articles: What Do Users Expect? In *Design, user experience, and usability. user experience design for everyday life applications and services*, 2014. Heraklion, Crete, Greece, pp. 164-172. doi:10.1007/s12160-015-9730-3
- Folkhälsomyndigheten. (2018). *Övervikt och fetma*. Retrieved May 12, 2018, from <https://www.folkhalsomyndigheten.se/folkhalsorapportering-statistik/folkhalsans-utveckling/levnadsvanor/overvikt-och-fetma/>
- Gillman, A. S. & Bryan, A. D. (2016). Effects of Performance Versus Game-Based Mobile Applications on Response to Exercise. *Annals of Behavioral Medicine*, 50(1), 157–162. doi:10.1007/s12160-015-9730-3
- Glover, I. (2013). Play as you learn: Gamification as a technique for motivating learners. In *Edmedia: World conference on educational media and technology*, 2013. Victoria, Canada, pp. 1999-2008. Retrieved from <https://www.learntechlib.org/p/112246>
- Hamari, J. (2017). Do badges increase user activity? A field experiment on the effects of gamification. *Computers in Human Behavior*, 71, 469–478. doi:10.1016/J.CHB.2015.03.036
- Hern, A. (2016, July 12). Pokémon go becomes global craze as game overtakes twitter for us users. *The Guardian*. Retrieved May 7, 2018, from <https://www.theguardian.com/technology/2016/jul/12/pokemon-go-takes-over-twitter>

- [//www.theguardian.com/technology/2016/jul/12/pokemon-go-becomes-global-phenomenon-as-number-of-us-users-overtakes-twitter](http://www.theguardian.com/technology/2016/jul/12/pokemon-go-becomes-global-phenomenon-as-number-of-us-users-overtakes-twitter)
- IMS Institute for Healthcare informatics. (2015). *Patient adoption of mhealth*. Retrieved May 13, 2018, from <https://www.iqvia.com/-/media/iqvia/pdfs/institute-reports/patient-adoption-of-mhealth.pdf?la=en&hash=B3ACFA8ADDB143F29EAC0C33D533BC5D7AABD689>
- Lee, C., Lee, K., & Lee, D. (2017). Mobile healthcare applications and gamification for sustained health maintenance. *Sustainability*, 9(5), 772. doi:10.3390/su9050772
- Lister, C., West, J. H., Cannon, B., Sax, T., & Brodegard, D. (2014). Just a Fad? Gamification in Health and Fitness Apps. *JMIR Serious Games*, 2(2), e9. doi:10.2196/games.3413
- Meerwein, G., Rodeck, B., & Mahnke, F. H. (2007). *Color - communication in architectural space*. Berlin, Basel: DE GRUYTER.
- Steinhubl, S. R., Muse, E. D., & Topol, E. J. (2013). Can mobile health technologies transform health care? *JAMA*, 310(22), 2395–2396. doi:10.1001/jama.2013.281078
- Sun, H. (2013). Impact of exergames on physical activity and motivation in elementary school students: A follow-up study. *Journal of Sport and Health Science*, 2(3), 138–145. doi:<https://doi.org/10.1016/j.jshs.2013.02.003>
- Turner, P. (2017). *A psychology of user experience : Involvement, affect and aesthetics*. Cham: Springer International Publishing.

A Updated Icons

After the final user tests, the icons of the application were updated in order to fit better with the overall design and to be more intuitive for users. These can be seen in Figure 37.

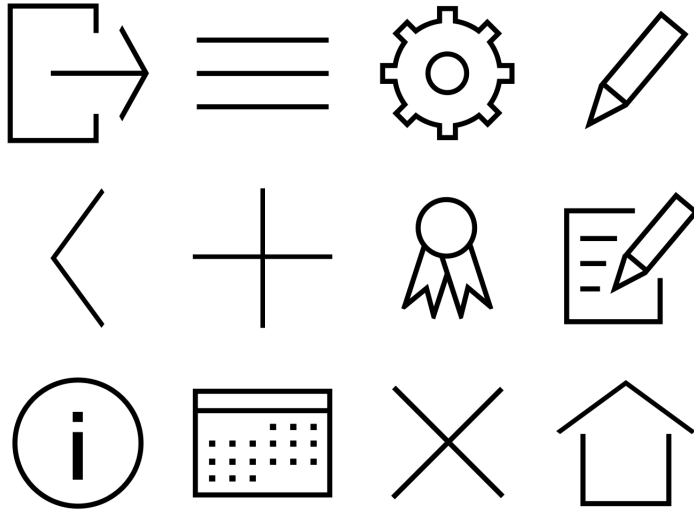


Figure 37: A new version of the applications icons, created after the final user tests. Created by author.

B Privacy Policy

We at GetPushApp.com built the Get PushApp app as a free app. This service is provided by GetPushApp.com at no cost and is intended for use as is.

This page is used to inform website visitors regarding our policies with the collection, use, and disclosure of personal information if anyone decided to use our service.

If you choose to use our service, then you agree to the collection and use of information in relation to this policy. The personal Information that we collect is used for providing and improving the service. We will not use or share your information with anyone except as described in this privacy policy.

The terms used in this privacy policy have the same meanings as in our terms and conditions, which is accessible at Get PushApp unless otherwise defined in this privacy policy.

Information Collection and Use

For a better experience, while using our service, we require you to provide us with certain personally identifiable information, including but not limited to email, first name and your logged workouts. The information that we request will be retained by us and used as following:

The email is used by the app as a means of identifying the users and allow them to sign in to their accounts. The email may also be used to send information to you, such as if you have forgotten your password etc. The first name is used to greet the user upon signing in, and is in no way used otherwise.

The collection of logged workouts is used to allow you as a user to track your progress.

The app does use third party services that may collect information used to identify you.

Link to privacy policy of third party service providers used by the app:

- Currently no third parties involved

The app does not share your personal information with any other third parties.

Security

We value your trust in providing us your personal information, thus we are striving to use commercially acceptable means of protecting it. But remember that no method of transmission over the internet, or method of electronic storage is 100% secure and reliable, and we cannot guarantee its absolute security.

All data is transferred securely to the server side by using

Changes to This Privacy Policy

We may update our privacy policy from time to time. Thus, you are advised to review this page periodically for any changes. We will notify you of any changes by posting the new privacy policy on this page. These changes are effective immediately after they are posted on this page.

Contact Us

If you have any questions or suggestions about our privacy policy, do not hesitate to contact us.

C Terms & Conditions

By downloading or using the app, these terms will automatically apply to you – you should make sure therefore that you read them carefully before using the app. You are not allowed to copy, or modify the app, any part of the app, or our trademarks in any way. You are not allowed to attempt to extract the source code of the app, and you also should not try to translate the app into other languages, or make derivative versions. The app itself, and all the trade marks, copyright, database rights and other intellectual property rights related to it, still belong to GetPushApp.com.

GetPushApp.com is committed to ensuring that the app is as useful and efficient as possible. For that reason, we reserve the right to make changes to the app or to charge for its services, at any time and for any reason. We will never charge you for the app or its services without making it very clear to you exactly what you're paying for.

The Get PushApp app stores and processes personal data that you have provided to us, in order to provide our Service. It is your responsibility to keep your phone and access to the app secure. We therefore recommend that you do not jailbreak or root your phone, which is the process of removing software restrictions and limitations imposed by the official operating system of your device. It could make your phone vulnerable to malware/viruses/malicious programs, compromise your phone's security features and it could mean that the Get PushApp app will not work properly or at all.

You should be aware that there are certain things that GetPushApp.com will not take responsibility for. Certain functions of the app will require the app to have an active internet connection. The connection can be Wi-Fi, or provided by your mobile network provider, but GetPushApp.com cannot take responsibility for the app not working at full functionality if you do not have access to Wi-Fi, and you do not have any of your data allowance left.

If you are using the app outside of an area with Wi-Fi, you should remember that your terms of the agreement with your mobile network provider will still apply. As a result, you may be charged by your mobile provider for the cost of data for the duration of the connection while accessing the app, or other third party charges. In using the app, you are accepting responsibility for any such charges, including roaming data charges if you use the app outside of your home territory (i.e. region or country) without turning off data roaming. If you are not the bill payer for the device on which you are using the app, please be aware that we assume that you have received permission from the bill payer for using the app.

Along the same lines, GetPushApp.com cannot always take responsibility for the way you use the app i.e. You need to make sure that your device stays

charged – if it runs out of battery and you can not turn it on to avail the Service, GetPushApp.com cannot accept responsibility.

With respect to GetPushApp.com’s responsibility for your use of the app, when you are using the app, it is important to bear in mind that although we endeavour to ensure that it is updated and correct at all times, we do rely on third parties to provide information to us so that we can make it available to you. GetPushApp.com accepts no liability for any loss, direct or indirect, you experience as a result of relying wholly on this functionality of the app.

At some point, we may wish to update the app. The app is currently available on Android and iOS – the requirements for both systems (and for any additional systems we decide to extend the availability of the app to) may change, and you will need to download the updates if you want to keep using the app. GetPushApp.com does not promise that it will always update the app so that it is relevant to you and/or works with the iOS/Android version that you have installed on your device. However, you promise to always accept updates to the application when offered to you. We may also wish to stop providing the app, and may terminate use of it at any time without giving notice of termination to you. Unless we tell you otherwise, upon any termination, (a) the rights and licenses granted to you in these terms will end; (b) you must stop using the app, and (if needed) delete it from your device.

Changes to This Terms and Conditions

We may update our Terms and Conditions from time to time. Thus, you are advised to review this page periodically for any changes. We will notify you of any changes by posting the new Terms and Conditions on this page. These changes are effective immediately after they are posted on this page.

Contact Us

If you have any questions or suggestions about our Terms and Conditions, do not hesitate to contact us.

D Poster - Campus User Test

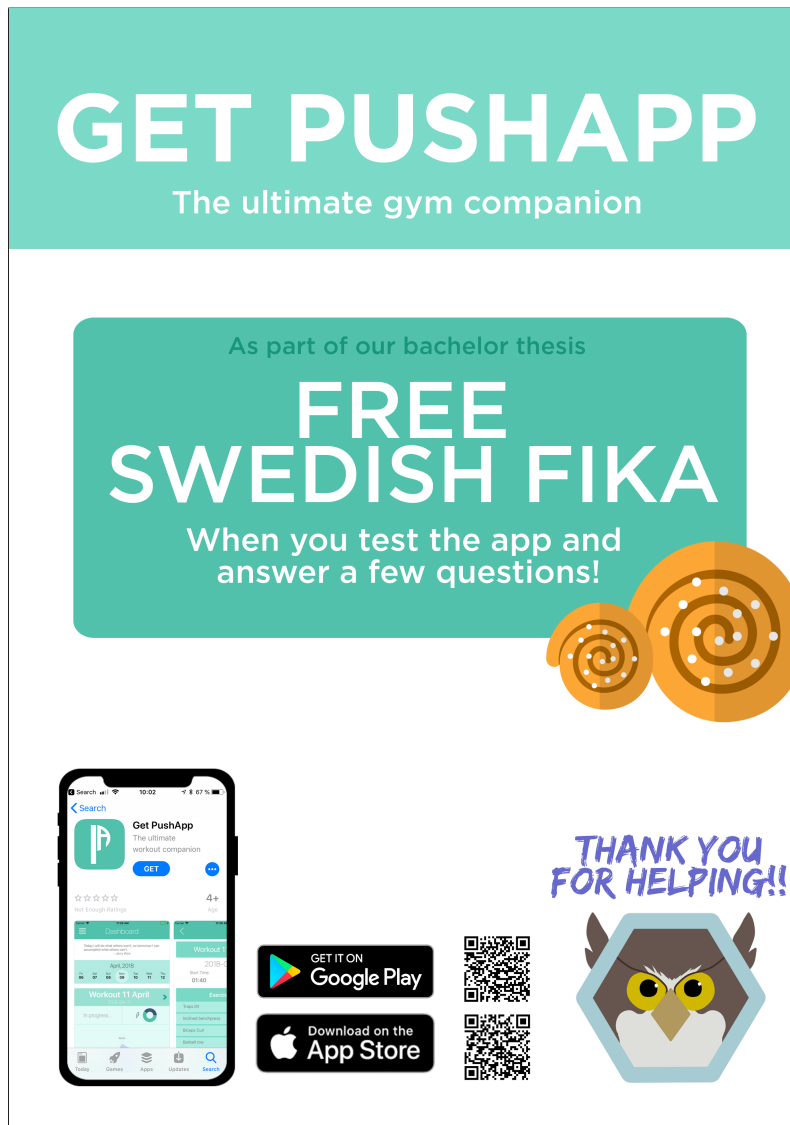


Figure 38: The poster used during the user tests on campus.
Created by author.

E Password Reset Email

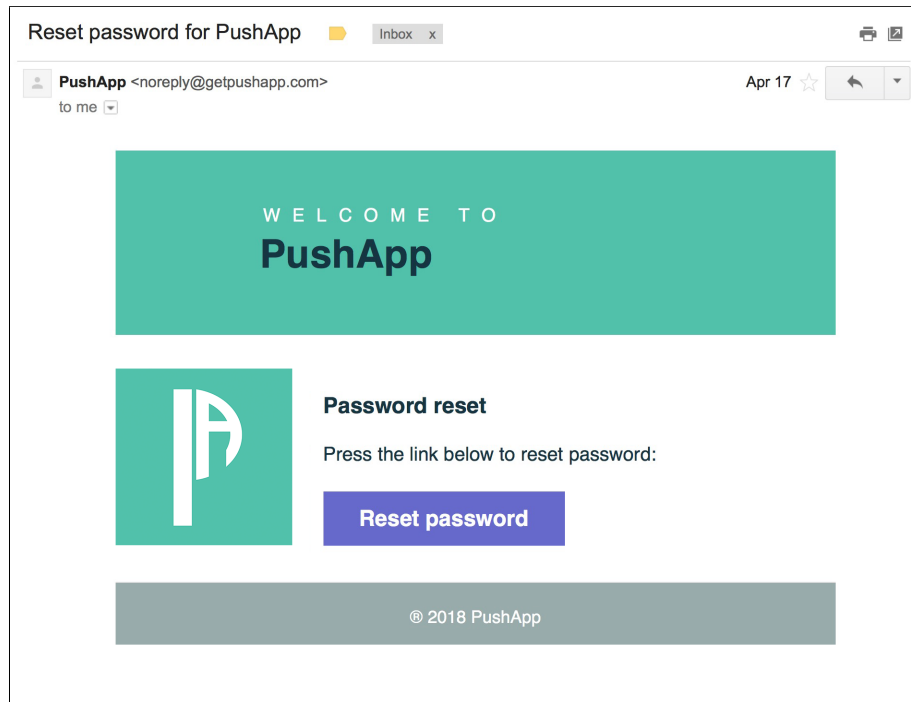


Figure 39: The email sent to the user when a password reset is requested. Created by author.

F Mockups

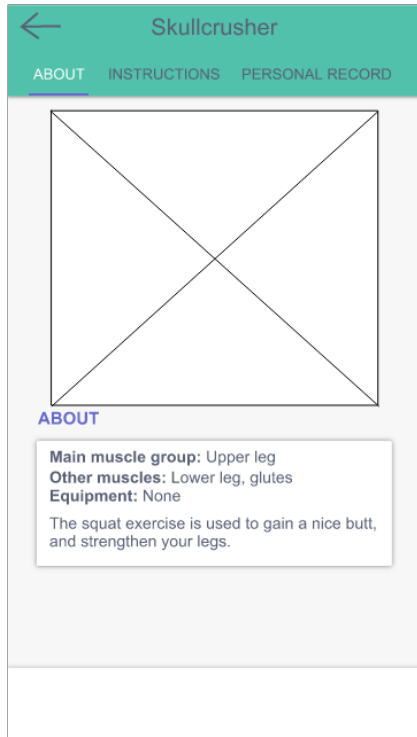


Figure 40: The mockup for the about view of an exercise.
Created by author.

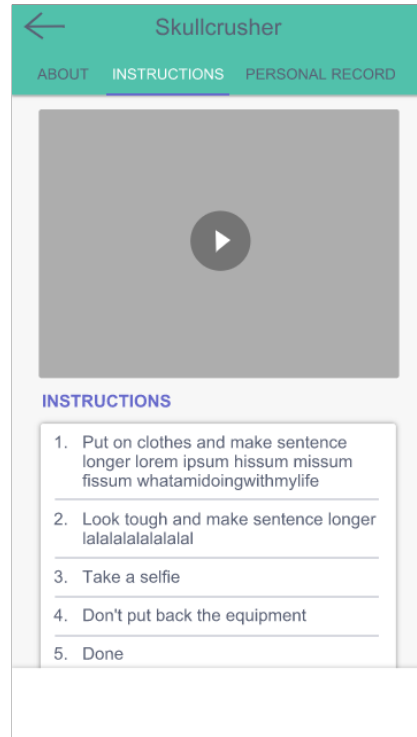


Figure 41: The mockup for the instructions view of an exercise.
Created by author.

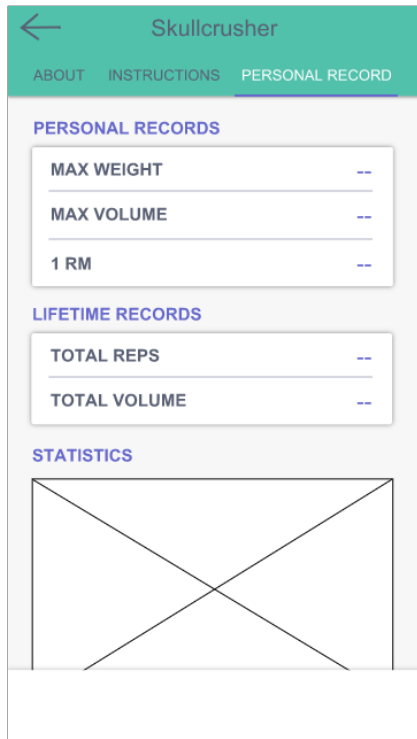


Figure 42: The mockup for the personal records view of an exercise. Created by author.



Figure 43: The mockup for the profile view. Created by author.

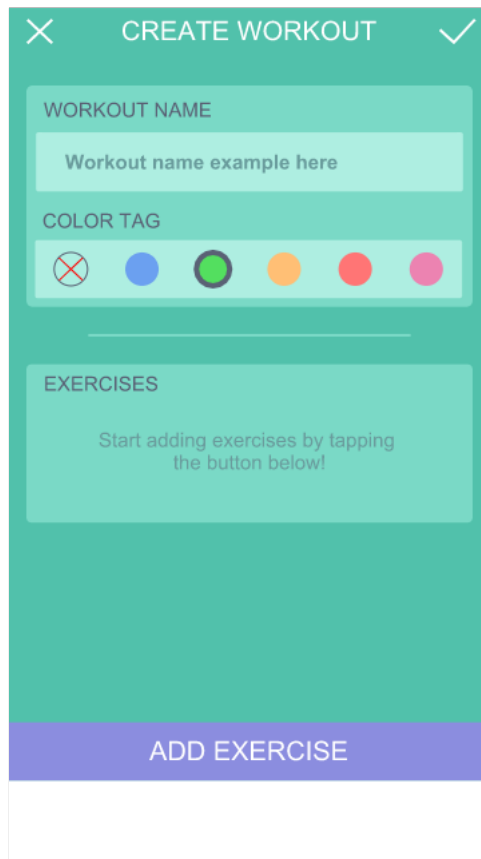


Figure 44: The mockup for the workout view where no exercises have been added yet.
Created by author.

G Website

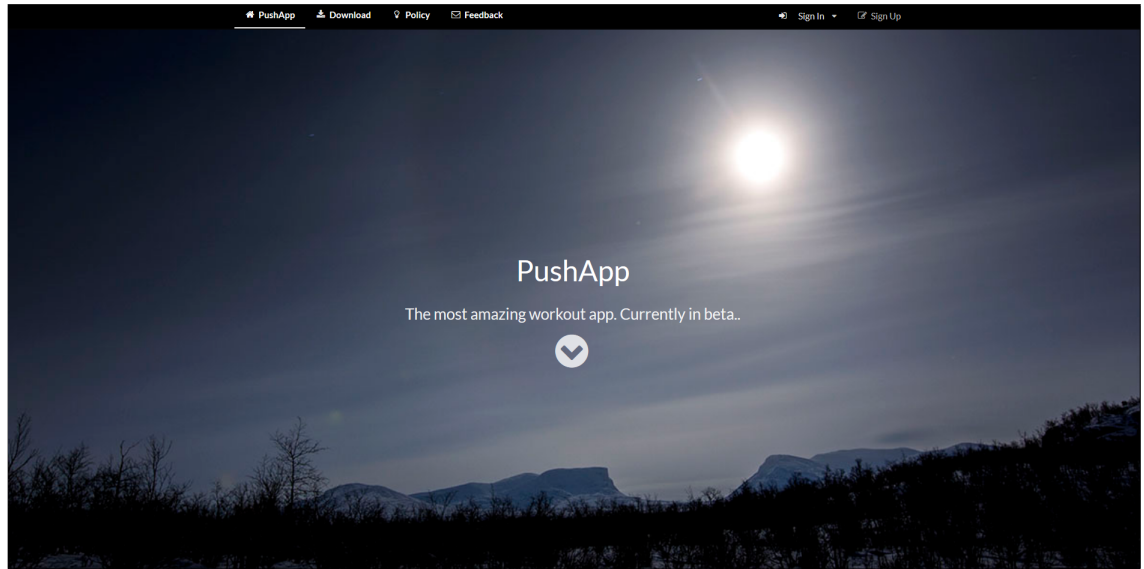


Figure 45: The website for the application, can be found at www.getpushapp.com. Created by author.