UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Software Engineering Curriculum

Elen Tingas

Design of an Interactive Fitness Application Based on User's Emotional Requirements

Master's Thesis (30 ECTS)

Supervisor(s):
Ishaya Peni Gambo, PhD

Tartu 2022

Design of an Interactive Fitness Application Based on User's Emotional Requirements

Abstract:
Incorporating users' emotions within software engineering has evolved throughout time. In recent years various research studies started taking into account users' emotional needs and their integration into the requirements elicitation process. Addressing users' emotions at the early stages of software development has the potential for long-term benefits for the system. In this thesis work, we build a practical, interactive fitness application based on users' emotional requirements. We look at motivational models and their incorporation with emotional goals to form emotional goal models for our system. The emotional goal models are constructed from the Do/Be/Feel/Role data. We conduct a survey with 182 participants where we ask them to fill in answers for Do/Be/Feel/Role framework. We use the survey results to create emotional goal models for our system and make sure to address every emotional goal with a corresponding functional goal. The leaf-level of the emotional goal model hierarchy is then extracted into a set of user stories. We proceed with user stories to build a real world application using React JS framework. We evaluate the result of our practical application by asking users first to experience it and then fill in a System Usability Scale test. The resulting score we received from the SUS test was around 84.5, which is a positive score for software systems. It is concluded that incorporating users' emotional requirements in the early stages of development positively impacts its user experience and usability.

Keywords:
Requirements engineering, goal oriented modelling, fitness application, emotional requirements, Socio-technical system

CERCS: P170 Computer science, numerical analysis, systems, control

Interaktiivse treeningurakenduse kujundamine kasutaja emotsionaalsete nõuete alusel

Lühikokkuvõte:

Kasutaja emotsioonidega arvestamine on tarkvaratehnika valdkonnas aja jooksul arenenud. Viimastel aastatel on erinevad uuringud hakanud arvestma kasutaja emotsionaalsete vajadustega ja nende integreerimisega nõuete väljaselgitamise protsessi. Kasutaja emotsioonidega arvestamine tarkvaraarenduse varajases staadiumis võib tuua süsteemile pikaajalist kasu. Käesoleva magistritöö raames arendati praktiline ja interaktiivne treeningurakendus lähtudes kasutaja emotsionaalsetest nõuetest. Magistritöös vaadeldi motivatsioonimudeleid ja nende sidumist emotsionaalsete eesmärkidega, et luua emotsionaalseid eesmärgimudeleid. Emotsionaalsed eesmärgimudelid loodi Do/Be/Feel/Role andmete põhjal. Andmed koguti magistritöö raames korraldatud 182 osalejaga küsitluses, milles paluti osalejatel vastata Do/Be/Feel/Role raamistikus esitatud küsimustele.

Küsitluse vastuste põhjal loodi arendatud treeningurakendusele emotsionaalsed eesmärgimudelid ja tagati, et iga emotsionaalse eesmärgiga oleks seotud vastav funktsionaalne eesmärk. Emotsionaalse eesmärgimudeli hierarhia lehetasandi põhjal koostati kasutajalood. Koostatud kasutajalugude põhjal arendati React JS raamistikus tarkvararakendus. Seejärel viidi tarkvararakenduse kasutajate hulgas läbi küsitlus, paludes kasutajatel seda rakendust esmalt kogeda ja seejärel täita System Usability Scale test. System Usability Scale testi tulemus oli umbes 84,5, mis on tarkvarasüsteemile positiivne tulemus. Jõuti järeldusele, et kasutaja emotsionaalsete nõuetega arvestamine tarkvaraarenduse alguses avaldab positiivset mõju tarkvara kasutajakogemusele ja kasutatavusele.

Võtmesõnad:
Nõuete tehnika, eesmärgile orienteeritud modelleerimine, treeningurakendus, emotsionaalsed nõuded, sotsiotehniline süsteem

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

Contents

List of Figures

List of Tables

1 Introduction

Nowadays, people pay much attention to their physical and mental health. Since the beginning of time, humans have been trying to stay active to preserve their health and youth. Throughout their daily routines, most of them had to walk for most distances, since back then, people did not have the luxury of having electric scooters, electric bicycles, fast driving cars, etc. But the more technological advancements made our lives easier, the more we started finding ourselves in the position of constant immobility. For this reason, many people started attending gyms and physical activity workout sessions to keep their mobility high and remain healthy.

As we have observed during recent years, outbreaks of viral viruses and people's inability to leave their homes safely at all times caused them to remain in a closed space, which intuitively lessens physical activity. Many companies started creating physical fitness applications for PC browsers and mobile phones so that people could work out from the comfort of their homes [MHB+21, Fla16]. However, these types of applications and websites are still attaining popularity.

To the extent of our research, there has not been a lot of practical research done together with the users of the applications. The research we would be interested in is to understand the emotional requirements of users of such fitness applications and document them in a way so that we can elicit new requirements to enhance the engagement level of the application.

Emotions are a big part of how humans perceive the world and the events around them. The simplest things that affect human emotions can have a snowball effect of considerable consequences. For instance, not being able to enjoy your favorite morning drink at the start of the day can affect your mood throughout the entire workday. Such parallels can be made in people's dedication to their fitness activities. If it is a rainy and cold day, people are less motivated and thus less likely to attend their evening yoga class. Therefore, trying to proactively predict such emotions and anticipate them during a fitness application with the usage of certain features could significantly affect the user's experience.

1.1 Problem statement

The main motivation why we have chosen the topic of creating a fitness application based on emotional requirements is to measure the effectiveness of the research done before us practically. There has been theoretical research that introduced a method of eliciting emotional requirements together with other requirements. We would like to perform a qualitative study on how effective this method can be in the real world. The reason why we decided to focus on a fitness application is that physical activity that challenges our bodies' limits can have a significant impact on our mood [PBC15]. Often people rely on their motivation and emotion whenever going for a workout. Thus, if we

could propose a practical solution for incorporating users' emotional needs into their user experience, we could provide a higher possibility of the user finishing their workout and reaching their health goals. We would first like to create a minimum viable prototype of a fitness application website, on which we will perform surveys with multiple users and document their emotional experiences on each step of the way. Then we will analyze it and try to enhance the fitness application's features to meet users' emotional needs. Consequently, we would like to measure the effectiveness of our practical qualitative research by performing System Usability Scale test on our application.

RQ1: How can we elicit emotional requirements to design an effective fitness application?

RQ2: How can we implement an effective fitness application by incorporating user's emotional requirements?

RQ3: How to validate the approach(es) used in RQ2?

1.2 Research goal

The main goal of this research is to use the previously analyzed theory and requirements elicitation methods to create a practical real world application that will enhance the user experience and achieve their health goals. Also, we will perform a final testing round to determine how much incorporating emotional requirements helped the users with their workout routine experience. The overarching goal is to improve users' experience of online fitness websites and contribute to their overall health resolutions.

1.3 Research contributions

The following sums up the contributions of this thesis:

- Theoretically, we explore the state-of-the-art research done in the field of emotional requirements to stimulate and advance the cause of the research agenda in building socio-technical systems that are emotionally acceptable and usable.
- Methodologically, we employ a survey with participants to construct the emotional goal model for the socio-technical system we examine. We provided novel insights on how websites can be improved based on documenting and using real-life users' opinions and experiences of the app.
- Practically, we implement a fitness application based on the elicited emotional requirements that attempt to measure the effectiveness of previous research and approaches in developing socio-technical systems.

1.4 Thesis outline

Chapter 2 presents the research background, shows other existing approaches. Chapter 3 describes the methodology and approach used in answering the research questions. Chapter 4 presents the study results based on the methodologies used. Chapter 5 presents the discussion of the results. Lastly, the threats to validity, conclusions and future work are presented in Chapter 6 and Chapter 7 accordingly.

2 Background

The background of this research is mainly based on the method of eliciting and representing emotional requirements reported in [TSP+19]. Their research examines how emotions are defined and perceived in socio-technical systems. In other words, we need to clearly define how to elicit functional and quality requirements and emotional ones. There are multiple ways of looking into emotions to predict and calculate user outcomes. As observed in [TSP+19], emotions are described in the two most superficial features: valence and arousal. Arousal is the scale of calmness through excitement, and valence is the scale of unpleasantness through pleasant feelings. They look into emotions as complex entities that our brain constructs. [TSP+19] takes motivational modelling as the way to construct emotional requirements in socio-technical systems where we take into account the goals to be achieved. We define the do, be and feel type of goals and extract them from the stakeholders of the given system. The do and be types of goals represent the functional and quality requirements, respectively. On the other hand, feel types of goals represent the emotions that are constructed in the stakeholder's brain according to the theory of constructed emotion. In their work [TSP+19], Taveter et al. describe two main techniques for eliciting such requirements: interviews and workshops. We will mainly use goal models to visualize and document the outcomes of the results gained by our survey.

In their work [MPLL+15] Miller et al. add the concept of emotional goals, which capture the emotional needs of stakeholders in a socio-technical system. Hence, because of focus on the people they call such models people-oriented software engineering (POSE) models. The main reason they decided to add this additional concept is that, despite human emotions becoming important in design [Nor04] and human-computer interaction during recent years, this prevalence did not transfer to software engineering. There are only a few papers addressing the emotional feelings of the users in requirements engineering [BJvB02, CPCLSAGC11, CPHLGCSA10, RB05, TS08]. The aim is to make emotions first class citizens in the elicitation process of the requirements. In his book about emotional design [Nor04], Norman talks about the three levels of human brain, their effect on emotions and what designers can infer from them.

1. Visceral processing: fast judgements that are pre-conscious and automatic. This type of processing is fairly consistent in people and pre-programmed in their brains. It is related to the appearance of the product in terms of design.

2. Behavioural processing: the sub-conscious part of the brain that dictates everyday behaviour. It is also consistent and in terms of design reflects the effectiveness of use.

3. Reflective processing: the most conscious and reflective part of the brain processing. This is where all the emotions and feelings are generated. In terms of design, it is

about the core meaning of the product. Reflective processing varies from person to person unlike the first two levels that are mostly consistent.

The functional and non-functional requirements arise from behavioral processing. Whereas reflective processing is where the user perceives the meaning of the system, which bears long term impact on the user's understanding of the product. This is why bringing forward emotions into the first level of requirements engineering will allow us the opportunity to address the deep reflective processing of the stakeholders.

[MPLL+15] argues two reasons on why emotional requirements should be first class citizens.

- The very subjective nature of emotions proposes many challenges in its nature. However, if not explicitly considered during the requirements elicitation process, then following the development process will have strict deadlines and limitations, which will result into complete dismissal of user's emotional needs.

- Unless explicitly addressed, stakeholders are not likely to speak about their emotional needs. In usual experience stakeholders focus on the functional requirements, often times ignoring the soft issues like social and emotional aspects. In spite of this, stakeholder's perception depends heavily on their emotional reaction to the system.

2.1 Related work

There are multiple related works done by the researchers that take various socio-technical systems and measure the emotional requirements to be taken into account. One such example is the e-health care system that measures the emotions of the stakeholders while using the system [TSP+19]. The result showed that getting conflicting emotional needs from different stakeholders is possible. For instance, in the case of an electronic health care system, some patients did not have much knowledge about their current health conditions and needs. Thus they required more assistance from the doctors and preferred the doctors to make most of the health decisions for them. On the other hand, some patients who were familiar with their health conditions wanted more freedom in their healthcare-related data decisions.

Another example of related work is the research done by Mooses [Moo21], where they defined the emotional requirements for health applications for adults aged 35 to 40. In this research, they had 9 workshops where they looked into the do-be-feel types of goals that we discussed earlier. In our work, we will create the do-be-feel goal models and implement them into a real life application in order to qualify the results of this method.

There are also the works done by [CPHLGCSA10, CPCLSAGC11], where the researchers ask the users how they feel regarding certain requirements and document it

in an X and Y axis where X is the level of arousal and Y is the level of please. The data points where X is low and Y is high are considered the accepted requirements.

The works done by [ST10, TS08, Sut11] define soft issues that are represented by values and motivations. They determine the importance level of each one on the requirements and filter out the crucial to pay attention to. In the same way, [PPK+11] takes personal values into consideration and ties each one to a requirement that would be ignored in other ways.

2.2 Gap analysis

Much theoretical work was done in this research area. However, there is still a need to measure the improvement of using emotional requirements in designing the features of a system. By employing practical code and seeing the results of a real life application, we will be able to contribute to the research area of emotional requirements and emotional goal modelling. We will evaluate our results with multiple questionnaires with the users.

3 Methodology

This chapter describes the methodology and approach used in answering the research questions.

3.1 Approach to Answering Research Questions

3.1.1 Approach for RQ1

In order to elicit emotional requirements we plan to perform a survey with large number of people who will fill in a questionnaire about our fitness application. The reason we decided to do a survey is to be able to reach as many people as possible and get their valuable feedback. The main goal for the survey will be to extract do/be/feel/role data which stands for the following:

- Do - functional goals: (communicate, sustain well being, call for help)
- Be - quality goals: (secure, scalable, reliable)
- Feel - emotional goals: (helpful, purposeful, engaging)
- Role - stakeholder roles.

In order to make the survey and the idea of the fitness application not completely abstract and vague to the participants, we will present a visualized prototype with each main feature. The goal of having the initial prototype is to help the users communicate their ideas in the most precise way. The plan is that the participants will take a look at the prototype besides the initial possible features that will be listed and decide what to add to the prototype, what is not needed, what could be improved and communicate it to us via their response. In order to help the participants understand what we mean by asking "What should X feature make you feel like?" within our framework of research, we plan to put an initial list of feel emotions from the research work by [Moo21]. Since her work was focused to create an ideal requirements for fitness application, our results will only benefit from having the possible feel emotions. All of the options in the survey will be either as checklists or a free text input for the users to fill in by themselves.

After collecting the results of the survey, we will extract the do/be/feel/role lists and represent with in an emotional goal model. Emotional goal model is an enhancement from the motivational goal model, which emphasizes the integration of feel emotions. Traditional motivational goal models have the following components:

- Functional goal
- Quality goal

- Roles

For emotional goal models, we will enhance them to have the following components:

- Functional goal
- Quality goal
- Emotional goal
- Roles

The notation or elements for the emotional goal models can be seen on Figure 1. Functional goals are represented by parallelograms, emotional goals are heart shapes, quality goals are clouds and roles are human like figures.

The skeleton of a goal model is a hierarchy of functional goals presented as a tree. The hierarchical structure shows that the sub component is part of the top-level component. The root of the tree is at the top of the page representing the overall goal of the system.

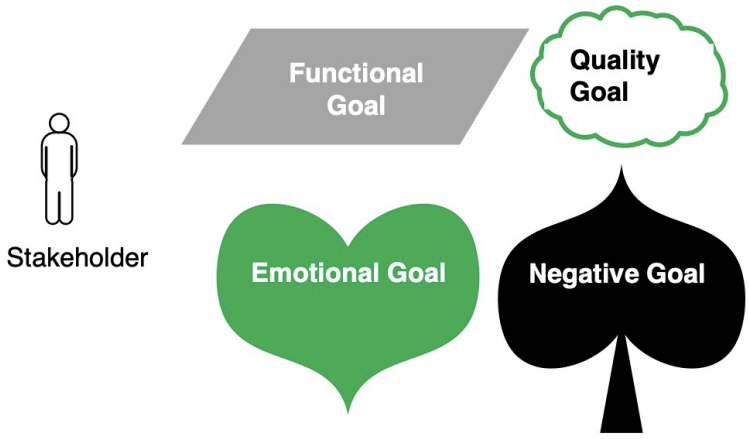

Figure 1. Notation for emotional goal model components. Source: [BLLS+19]

Functional goals in a goal model represent coarsely user requirements for the system to be designed. A functional goal is a representation of a functional requirement of the socio-technical system (in our case, the fitness application). A quality goal, as its name implies, is a non-functional or quality requirement of the system, representing a quality aspect of achieving given functional goal(s). An emotional goal is an emotional or affective requirement for the system to be designed, representing how achieving given functional goal(s) should feel for a human performing a certain stakeholder role.

Goal models resulting from survey can be easily turned into user requirements in the form of US (user stories). The general idea is that the socio-technical system to be created is first represented as a hierarchy of functional goals along with attached to them quality and emotional goals and roles. The goal model is then elaborated into low level goals until the leaf level is achieved [TNR+17]. The leaves of the goal tree are then further elaborated into user stories, that are simple artefacts for representing user requirements in agile software engineering [Coh04, PEM03]. A user story is a short, simple description of a feature described from the perspective of the person or role who desires the new capability, usually a user or customer of the system. The following is an example format we have used in this study for creating the US:

As a <user performing a certain role>, I need <to perform action> to support <achieving a certain goal>[<with the consideration of certain quality goal(s)>][<so that achieving certain emotional goal(s)>].

Please note that including in a user story quality goals to be considered and emotional goals to be achieved is optional.

3.1.2 Approach for RQ2

The approach for implementing a fitness application to incorporate user's emotional requirements is to first start a React JS project. React JS is one of the most popular and reliable frameworks for developing website applications. It is declarative and uses Components to assemble the overall website. The language that we will be using is TypeScript with React JS.

For state management purposes we will incorporate React Redux into our application. Redux is a popular state management system that enables us to manipulate the state from everywhere within the code, which makes the code easily scalable and manageable. For saving the user's data throughout multiple usages within the browser, we will use Redux Persist library that will let us the opportunity to rehydrate our state every time the user opens our website. This will support the features of saving one's constructed workouts, 30 day plans and other important data to be stored.

To provide responsive user interface and high quality design we will be using Material UI which ships us with a lot of ready made base level components, like Typography and Text areas. Since we will need a lot of media content to make sure the user is easily engaged with the upcoming workout moves, we will use a separate server to store all the media files regarding fitness workouts.

For quality fitness content, we will be using the workout moves and instructions that https://musclewiki.com/ provides. Once our application is ready to be deployed to a server, we will use Render to deploy it and use continuous deployment tools that they provide. This means every time a new piece of code is pushed to our GitHub server, our application will be automatically deployed.

3.1.3 Approach for RQ3

In order to evaluate the results from RQ2, that is the practical application that we have created, we will employ the System Usability Scale (SUS) testing method. The SUS method has been around for use for a long time and it has shown to be an effective evaluation questionnaire to deduct conclusions over a system's usability [Tho]. It was created by John Brooke in 1986 and has been proven to be a valid and reliable since his time. The questionnaire contains 10 questions and compared to other evaluation methods it is easy to be used online and it does not contain exhaustive long or open questions that usually bore or intimidate the user. The following Table 1 shows those ten questions.

The answer template to these questions is a Likert Scale with 5 points: the first point being "Strongly disagree" and 5th point being "Strongly agree". Please see Figure 2.

Calculating the SUS Usability score consists of multiple steps. Firstly, we subtract one from each odd numbered score. Then we subtract each even numbered score from 5. The reason we do these steps is that we want to normalize the result scores around 4, since the odd numbered questions are positively oriented and even numbered questions

Table 1. SUS questions

Number	Question
1	I think that I would like to use the fitness application frequently.
2	I found the fitness application unnecessarily complex.
3	I thought the fitness application was easy to use.
4	I think that I would need the support of a technical person to be able to use the fitness app.
5	I found the various functions in the fitness application were well integrated.
6	I thought there was too much inconsistency in the fitness application.
7	I would imagine that most people would learn to use the fitness application very quickly.
8	I found the fitness application very cumbersome to use.
9	I felt very confident using the fitness application.
10	I needed to learn a lot of things before I could get going with the fitness application.

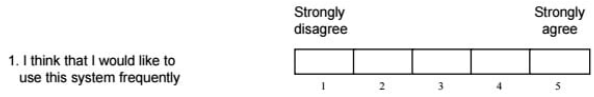

Figure 2. SUS template sample. Source: [Mei]

are negatively oriented. Then we take all the newly calculated points we gathered, sum them up and multiply by 2.5. The result is a number out of 100.

If the SUS score is equal to or above 80.3, then it's a really successful outcome and people enjoy the website a lot. If the score is around 68, which is the average SUS score for systems, then it is a normal, well-doing result. However, if the score is equal to or below 51, then the website needs multiple improvements and usability should to be prioritized as soon as possible.

Thus, our methodology will be to first ask real life users to experience our website and then fill in the SUS questionnaire.

4 Results

This chapter presents the study results based on the methodologies used.

4.1 Answering RQ1

4.1.1 Initial application prototype

For eliciting the emotional requirements our first steps were to create the initial prototype of the fitness application. We decided to create them with Miro prototypes. Overall, there are 5 main prototype pages reflecting the features of our system. The first is the feature of having a workout list page, as shown in Figure 3. The second is the feature of having a 30 Day Plan, as shown in Figure 4. The third is the feature of having a workout in progress, see Figure 5. The forth is the feature of having a feedback page, see Figure 6. Finally, the fifth is the feature of having a challenges list page, see Figure 7.

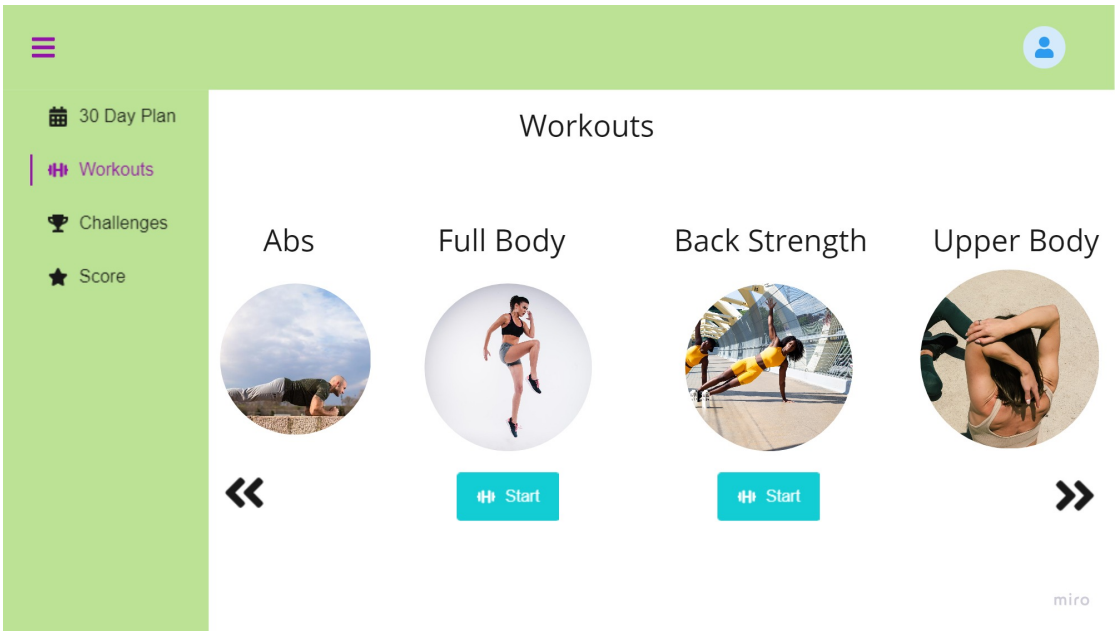

Figure 3. Feature of having a workouts list page

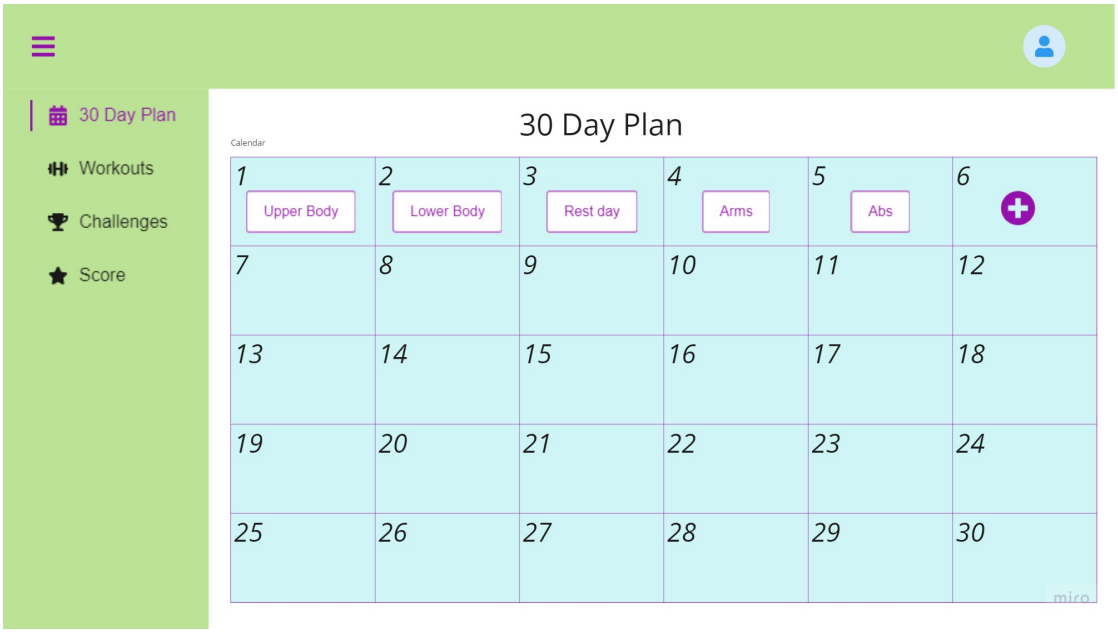

Figure 4. Feature of having a 30 Day Plan

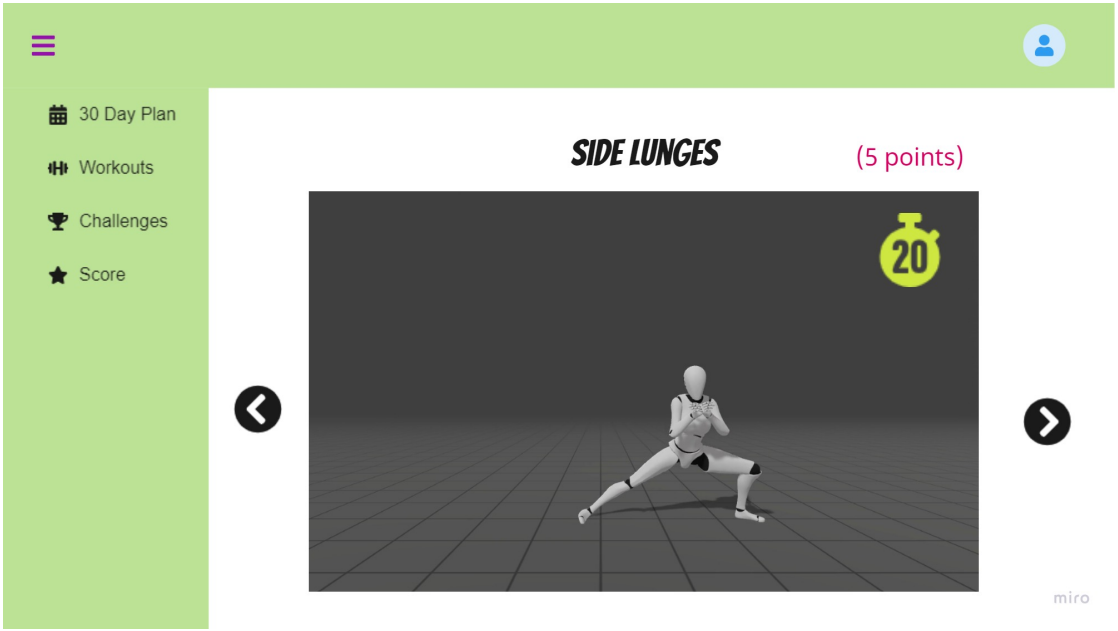

Figure 5. Feature of having a Workout in Progress

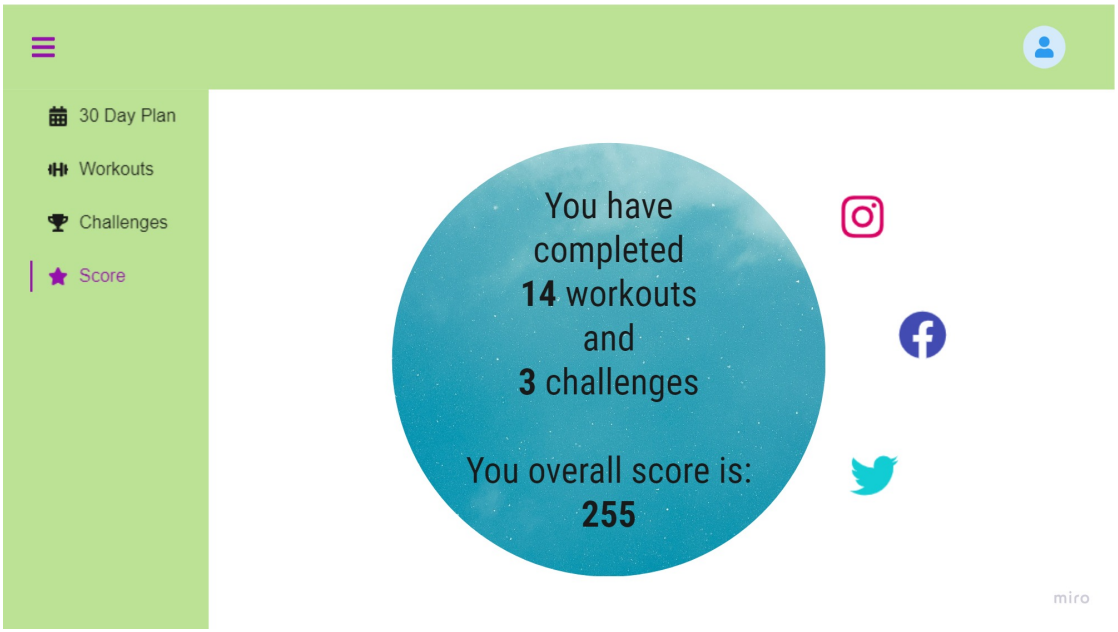

Figure 6. Feature of having a feedback page

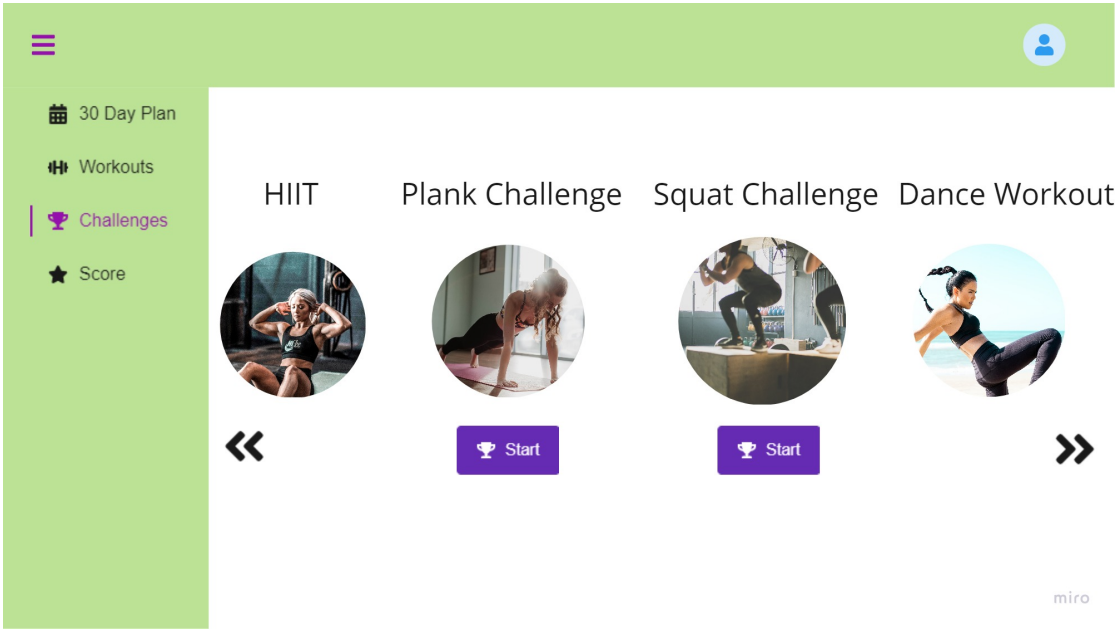

Figure 7. Feature of having a challenges list page

4.1.2 Survey with 182 participants

We conducted a survey with 182 participants to get the data corresponding to do/be/feel/-role framework. Initial questions were asked from the participants to determine their age, gender and country. We also asked them to pick a fitness level, which will serve the purpose of roles within our system. For sake of researching a broad community with various types of backgrounds and professions, we asked the users to optionally input their occupation. Here is the summary of the results of the initial get-to-know questions we asked our participants.

- Age - around 36% of the users were within the age range of 20 to 25, 21% were from 25 to 30 and around 15% were from 30 to 35, see Figure 8 for additional data.
- Gender - out of 182 participants 96 were male and 86 were female, see Figure 9.
- Fitness level - 51% of the responders reported to have intermediate fitness level, 43% of the responders choose the role of a beginner and the remaining 6% were experts. See Figure 10.
- Country - we have received responses from all over the world including Armenia, Nigeria, Estonia, USA, Pakistan, Iran, Germany, Brazil, etc.
- Occupation - we received in total 136 responses for the optional field of occupation and here are some of the responses that make sure our study reaches a community with different backgrounds and interests: software engineer, student, banker, lecturer, civil servant, virtual assistant, physical instructor, retiree, architect, financial analyst, logistics specialist, QA engineer, PhD student, marketing specialist, cybersecurity specialist, HR analyst, Lawyer, civil engineer, etc.

After the series of initial questions presented above, we started the main part of the survey where we ask the participants to take a deep look into each one of our potential features on the prototype and provide their input.

For each of the five features, we ask 3 main questions:

1. In your opinion, what should such feature DO?
2. In your opinion, what should such feature BE like?
3. In your opinion, what should such feature make you FEEL like?

Here are the results for every feature. Since we have a lot of answers with various ideas, we take into account the requirements and requests that address the emotional requirements stated by the participants.

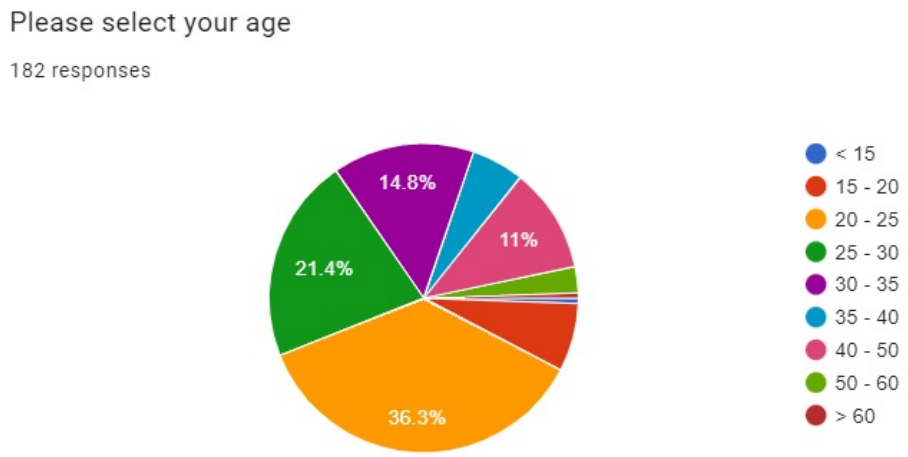

Figure 8. Age results

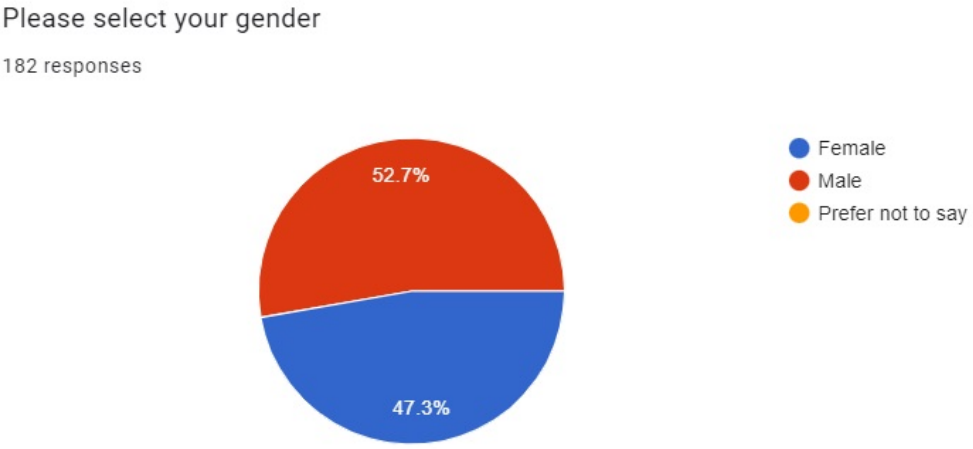

Figure 9. Gender results

Feature of having a workouts list page: (see Figure 11, 12, 13)

- what should such feature do?
  - Let me choose a desired workout plan
  - Show information about each workout's focus areas
  - Let me choose a random workout plan

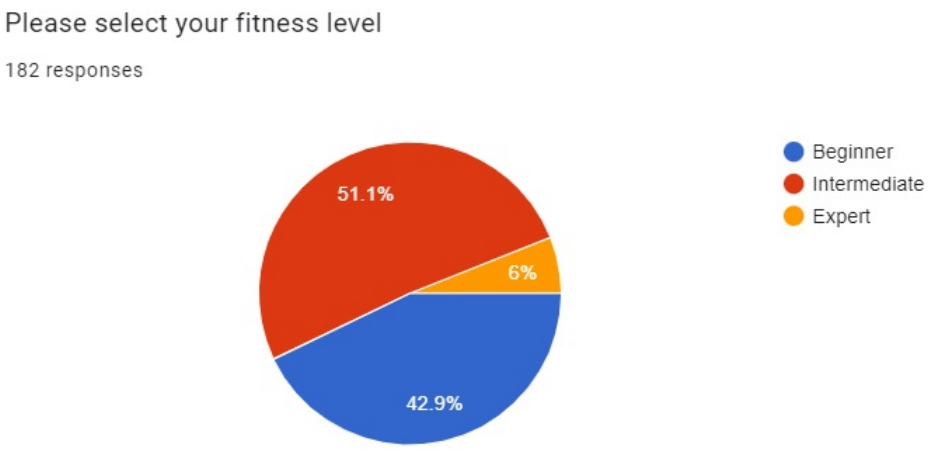

Figure 10. Level of fitness results

  - Show information about the intensity of the workout plans
  - Recommend workouts based on user's fitness level
  - Provide workout plans for different levels and fitness goals.
  - Amount of burned calories
  - State the advantages of each plan
- what should such feature be like?
  - Adaptive
  - Intuitive
  - Modern
  - Innovative
  - Self-explanatory
  - Descriptive, since I want information about each workout's focus areas
  - Interactive
  - Relevant to the needs of the body of each individual
  - Easy to understand content for people who don't know any sport related termins/abbreviations
  - Simple and clear
  - Attractive

- what should such feature make you feel like?
  - Motivated
  - Inspired
  - Informed
  - Special
  - Guided
  - Healthy
  - Confident
  - Encouraged

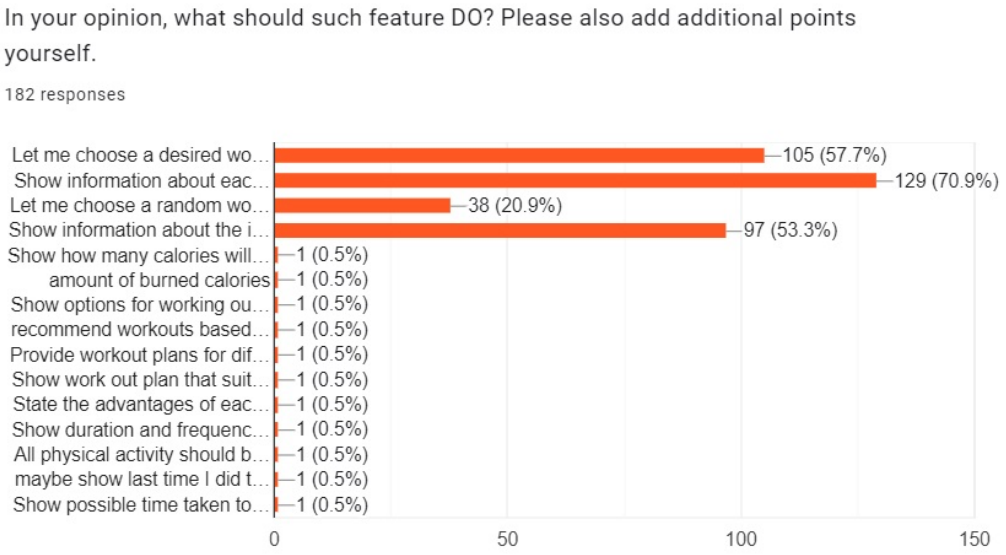

Figure 11. 1. Results for Do requirements

Feature of having a 30 Day Plan: (see Figure 14, 15, 16)

- what should such feature do?
  - Automatically generate a 30 day workout plan for me
  - Let me construct my own 30 day workout plan

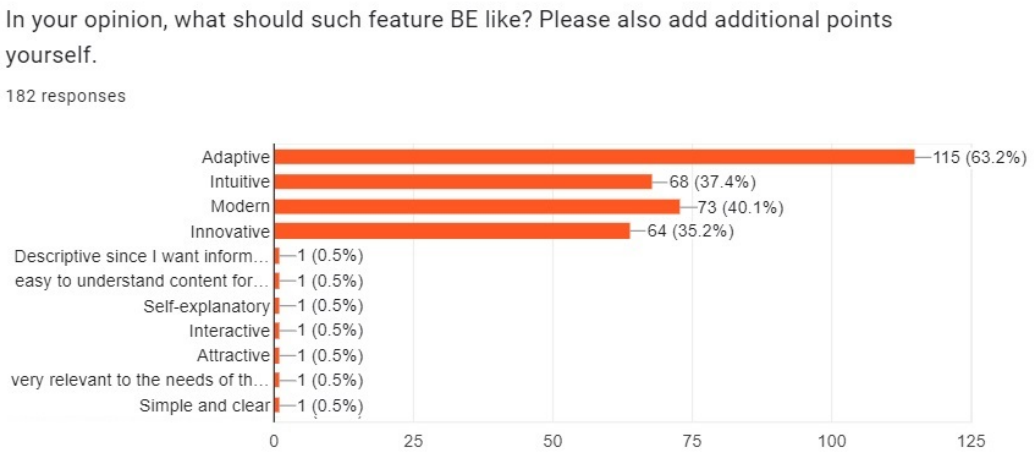

Figure 12. 1. Results for Be requirements

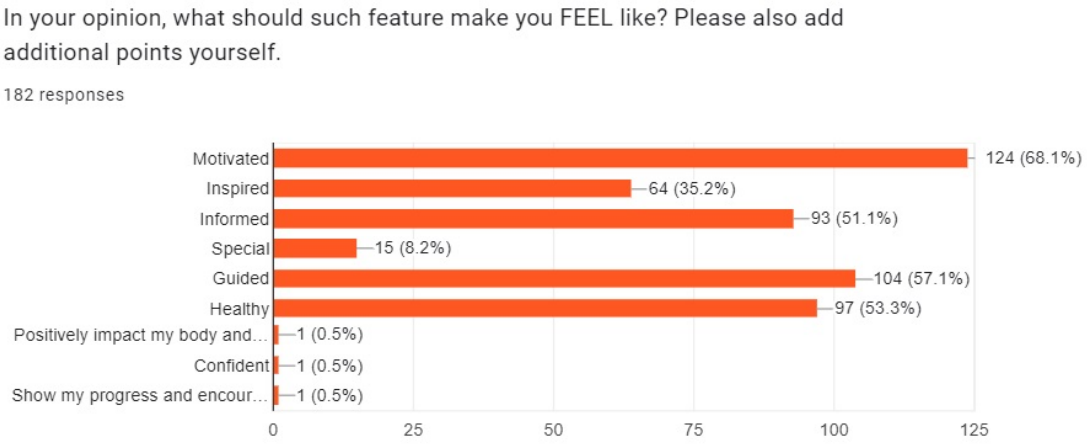

Figure 13. 1. Results for Feel requirements

  - Include meditation/stretch/rest days
  - Let me make some changes to the automatically generated 30 days workout plan
  - Show a progress bar of burnt calories/lost kgs that I will get in 30 days
  - Make the plan flexible for me to alter

- what should such feature be like?

  - Simple
  - Compatible
  - User friendly
  - Realistic
  - Challenging
  - Flexible
  - Modifiable
- what should such feature make you feel like?
  - Cared for
  - In control
  - Positive
  - Hopeful
  - Guided
  - Satisfied
  - Organized
  - Confident

Feature of having a Workout in Progress: (see Figure 17, 18, 19)

- what should such feature do?
  - Provide a timer for each workout move
  - Reward me a score based on the workout move
  - Let me have short breaks during the workout
  - Let me create my own workout routine
  - Show how many calories I burned doing the workout to keep me motivated
  - Display the correct way to do an exercise
  - I also like the feeling of having a timer for all activities. Instead of "15 leg raises" I'd prefer "30 seconds leg raises" I tend to loose count or get distracted
- what should such feature be like?
  - Safe

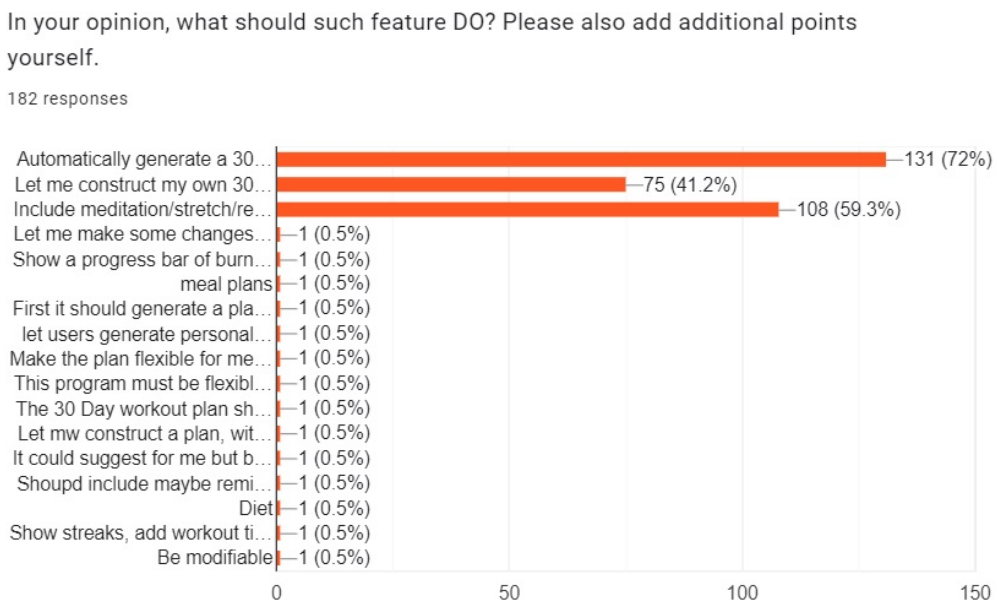

Figure 14. 2. Results for Do requirements

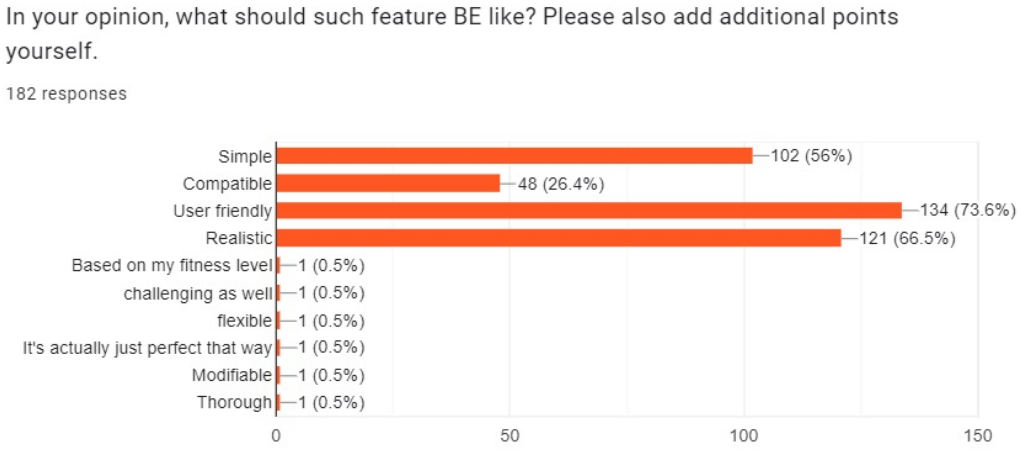

Figure 15. 2. Results for Be requirements

- Practical
- Playful
- Precise

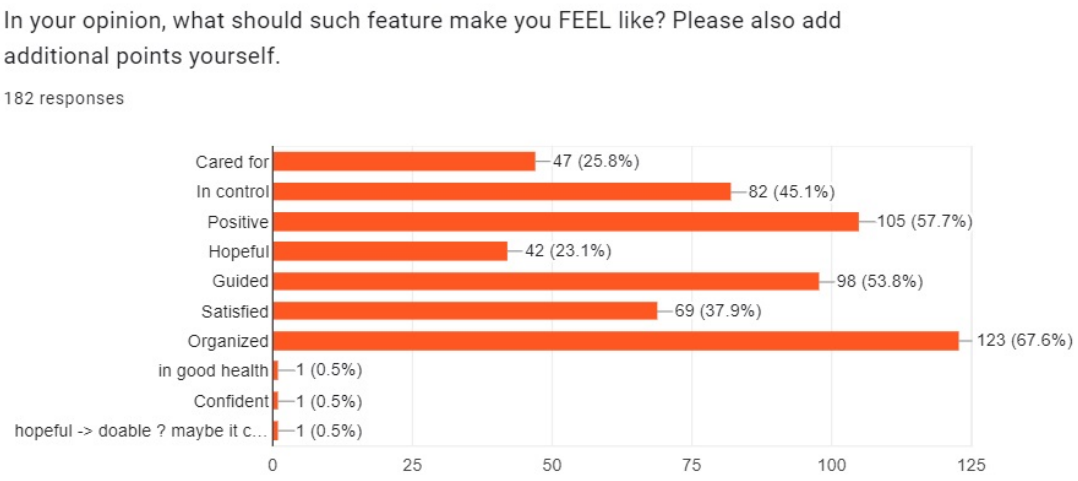

Figure 16. 2. Results for Feel requirements

  - Adaptive
  - Informative
  - Explanatory
  - Effective

- what should such feature make you feel like?
  - Success
  - Supported
  - Addicted
  - Motivated
  - Happy
  - Energetic
  - Guided
  - In control
  - Self-confident
  - Not exhausted
  - Not tired

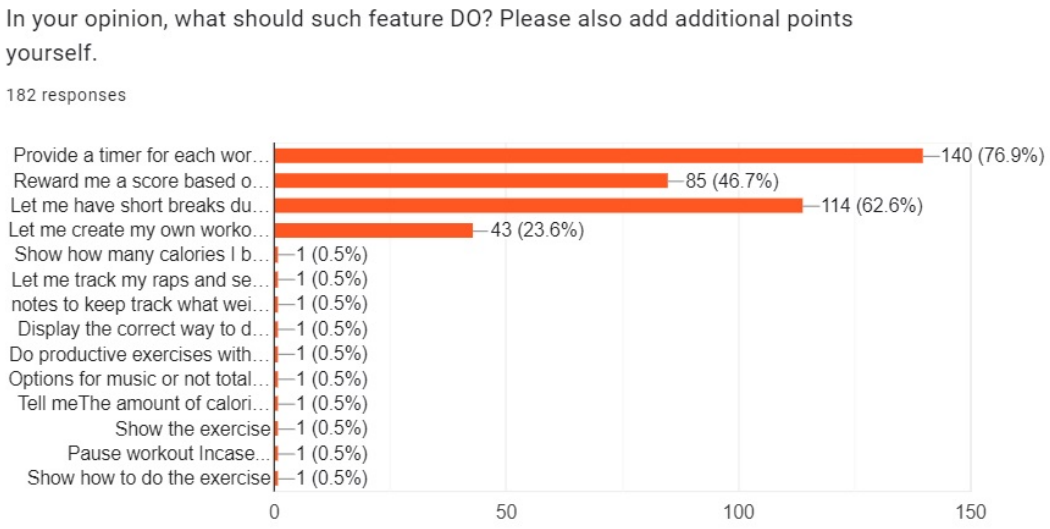

Figure 17. 3. Results for Do requirements

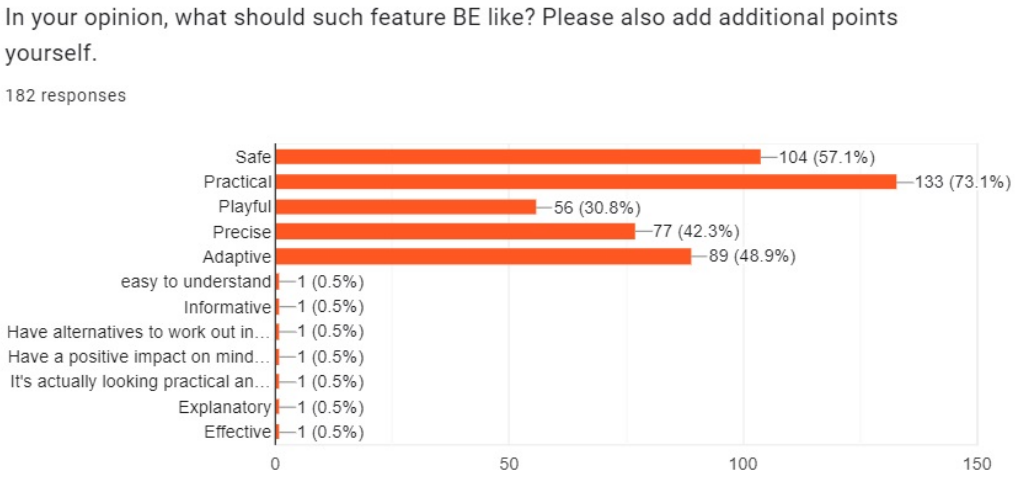

Figure 18. 3. Results for Be requirements

Feature of having a feedback page: (see Figure 20, 21, 22)

- what should such feature do?
  - Track the workouts I have completed
  - Track the challenges I have completed

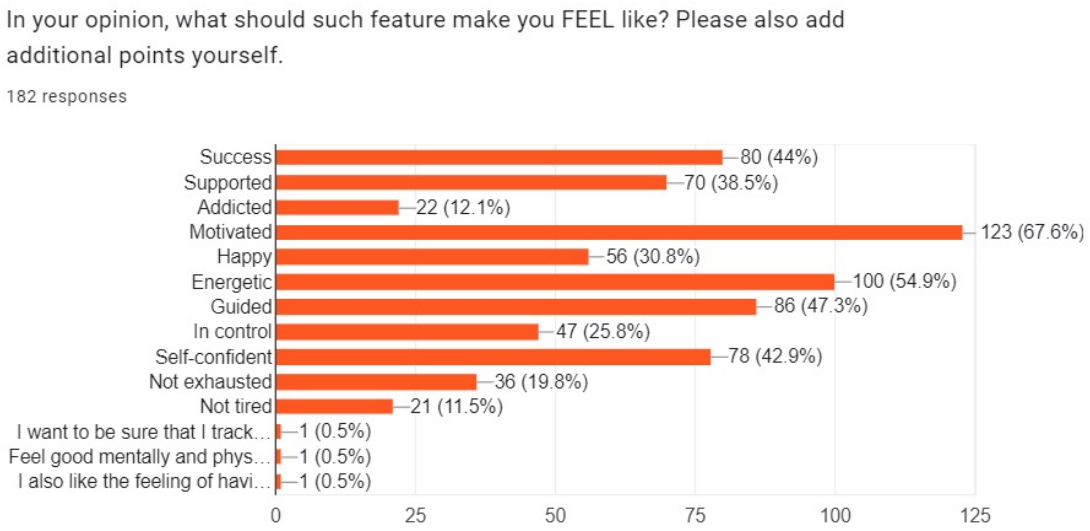

Figure 19. 3. Results for Feel requirements

  - Provide feedback about my overall score
  - Share my progress with friends in social media
  - Show how many calories were burned doing workouts and challenges
  - Share progress with friends, also give estimate of calories burned.

- what should such feature be like?
  - Reliable
  - Ads-free
  - Realistic
  - Insightful
  - Accurate
  - Relevant

- what should such feature make you feel like?
  - Hopeful
  - Victorious
  - Proud
  - Special

- Informed
- Supported
- Inspired
- Accomplished
- Encouraged

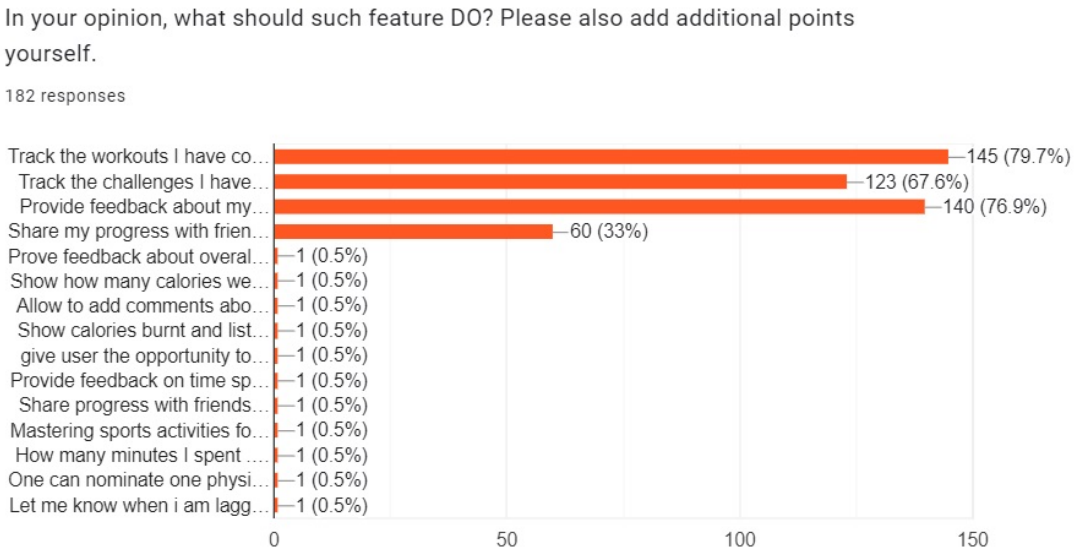

Figure 20. 4. Results for Do requirements

Feature of having a challenges list page: (see Figure 23, 24, 25)

- what should such feature do?
  - Let me choose a desired challenge to complete
  - Show information about each challenge's focus areas
  - Let me choose a random challenge
  - Reward me points for completing a challenge
- what should such feature be like?
  - Innovative

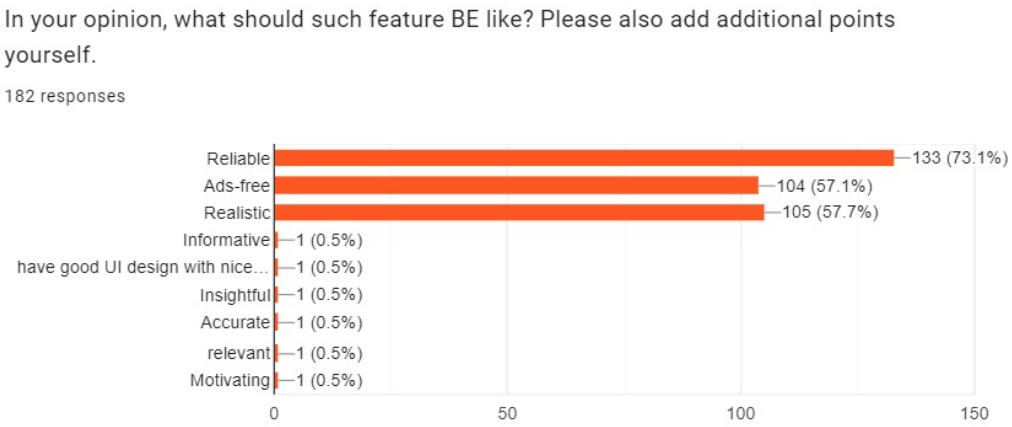

Figure 21. 4. Results for Be requirements

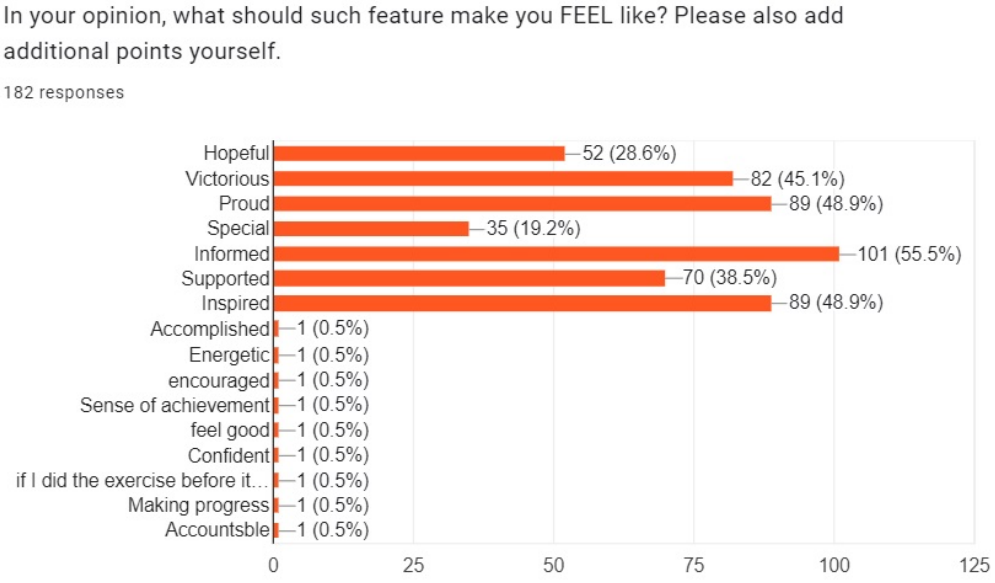

Figure 22. 4. Results for Feel requirements

- Up to date
- Reliable
- Informative
- Reasonable

  - Clear

- what should such feature make you feel like?
  - Amused
  - Excited
  - Special
  - Active
  - Victorious
  - Challenged
  - Competitive
  - Inspired
  - Curious

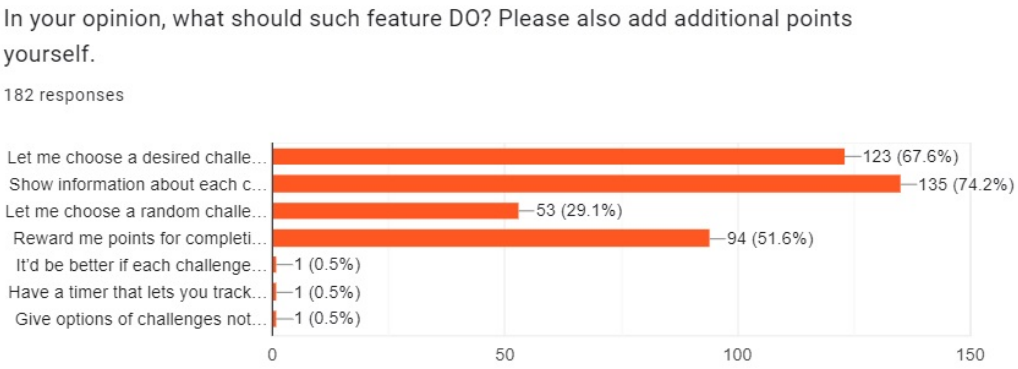

Figure 23. 5. Results for Do requirements

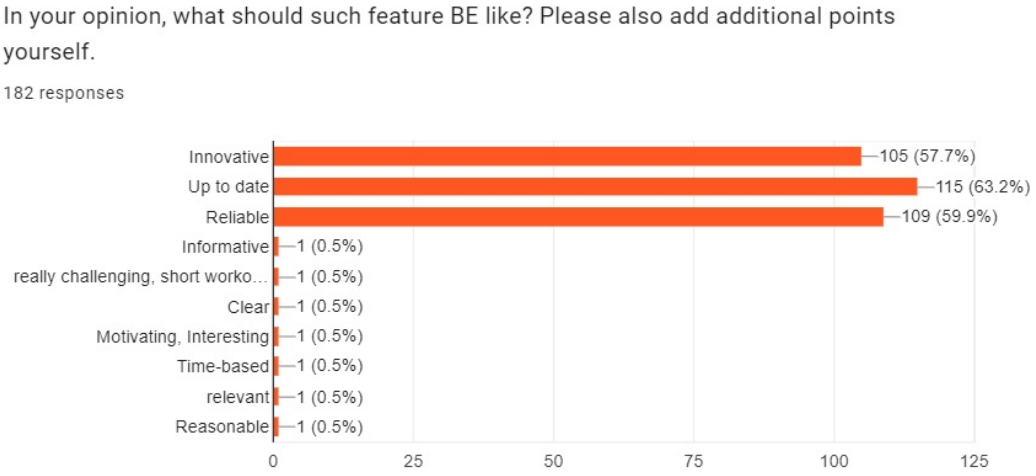

Figure 24. 5. Results for Be requirements

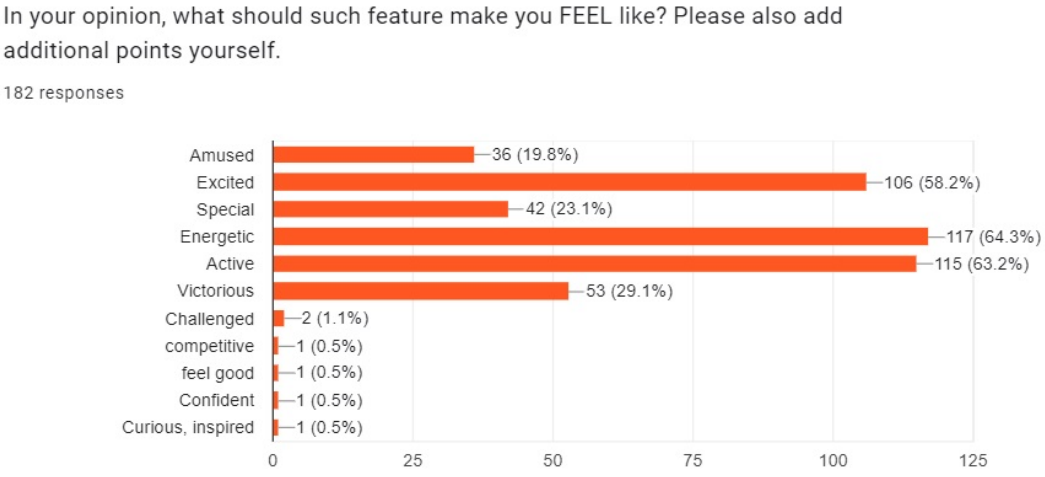

Figure 25. 5. Results for Feel requirements

4.1.3 Creation of emotional goal models

Now that we have collected the Do/Be/Feel/Role requirements list from our survey, we go ahead with the creation of our emotional goal models. In order to be able to analyze each goal in detail, we will first present an overview of the entire goal model and then further elaborate each part of it in a separate emotional goal model. Please see the overview of the goal model in Figure 26.

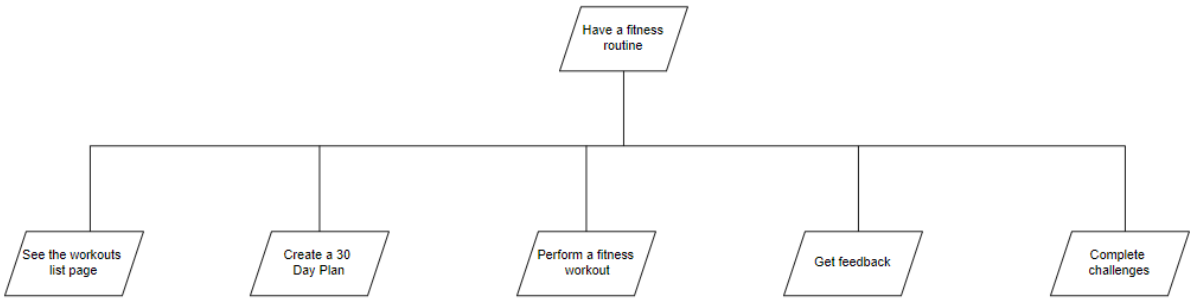

Figure 26. Overview of the system's goal model

We can now start to elaborate on each of the sub goals and present their corresponding emotional models. For seeing the workouts list page, we branch out seven sub goals that address every emotional requirement. For example, picking a random workout makes the user feel confident in themselves. Knowing information about workout's focus areas and the amount of burnt calories makes the user feel informed and guided correspondingly. Since the functional Do features that our participants suggested did not address the emotions of feeling motivated and healthy, we add an additional one ourselves. The sub goal we are referring to is reading about workout health benefits in order to address the emotional requirements of feeling motivated and healthy. Please see Figure 27 for the detailed emotional goal model.

For the sub goal of creating a 30 day plan we branched out five more sub goals. Having a 30 day plan automatically generated for the beginners makes them feel cared for and guided. On the other hand constructing their own plan for experts makes them feel in control. Similarly, customizing their plans makes users feel organized and having the ability to have rest days makes sure they feel satisfied and positive. The last sub goal which is having a progress bar for the burnt calories will make them feel hopeful and confident. Please see Figure 28 for the detailed emotional goal model.

The third sub goal of performing a fitness workout has seven branches. The sub goal itself needs to make the users feel happy and not tired throughout the workout. Seeing

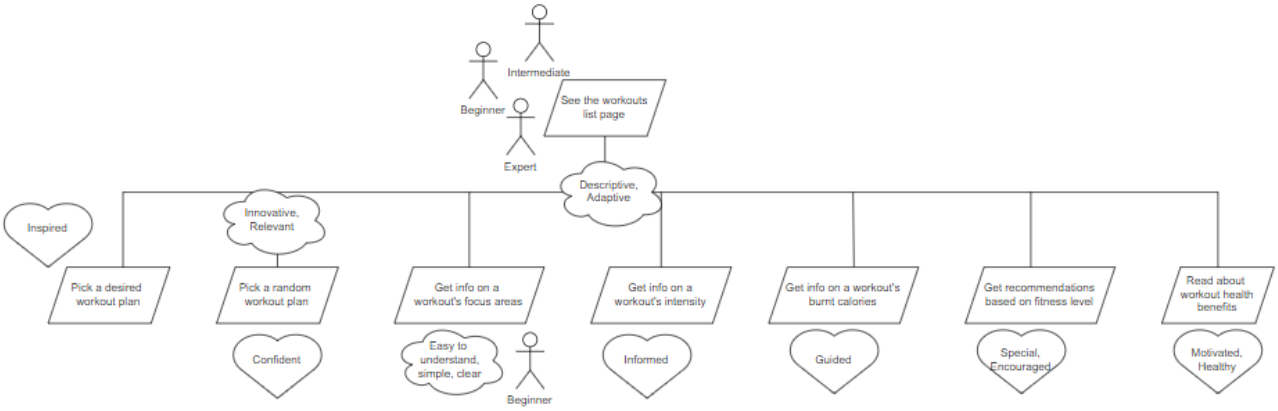

Figure 27. See the workouts list page

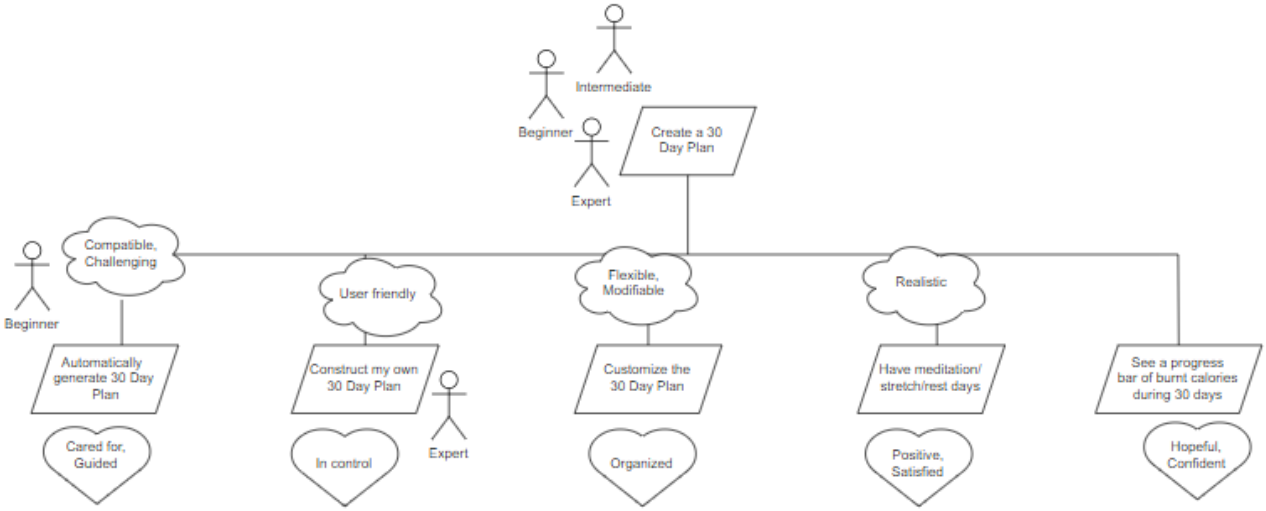

Figure 28. Create a 30 Day Plan

a timer for every workout move will make the user feel guided. Getting rewarded with a score for performing an exercise move will make them feel supported. The sub goal of having short breaks will address the emotions of not feeling exhausted and feeling energetic. Letting the experts create their own workout plan will make them feel in control. Seeing the amount of their burnt calories will make them feel successful and being able to see the correct way of performing an exercise will make them feel self-

confident. The last branch in the sub goal's emotional goal model is to see motivational messages throughout the workout. This functional goal was added by ourselves to address the emotions of feeling motivated and addicted to the workout, since the do requirements mentioned by the participants in the survey did not address the latter ones. Please see Figure 29 for the detailed emotional goal model.

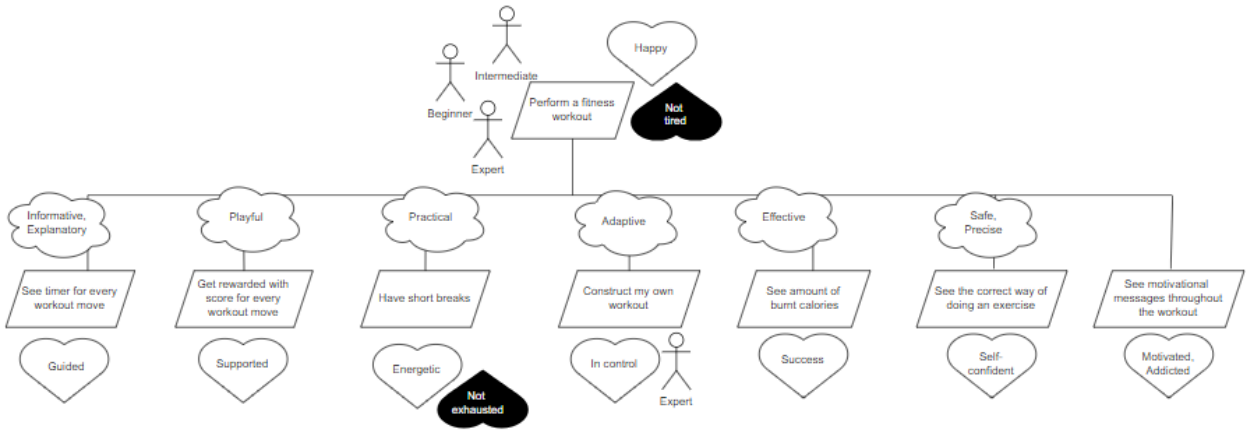

Figure 29. Perform a fitness workout

The sub goal of getting feedback has six branched out leaves. Tracking their completed workouts and challenges will make the users feel accomplished and special correspondingly. Showing the amount of total burnt calories and their overall score will make them feel informed, hopeful and victorious. Sharing progress with friends addresses the emotions of feeling proud and supported. Lastly, in order to address the emotions of feeling inspired and encouraged we added the functional goal of being able to immediately start a new workout right after seeing their feedback. Please see Figure 30 for the detailed emotional goal model.

The last sub goal of seeing the challenges list page has four branches. Picking a desired challenge from the list will make the users feel active and amused. Picking a random challenge will make sure they feel curious and excited. Getting information on a particular challenge's focus areas will make them feel inspired. Lastly, getting a score for completing every challenge will address the user's emotions of feeling competitive, victorious and special. Please see Figure 31 for the detailed emotional goal model.

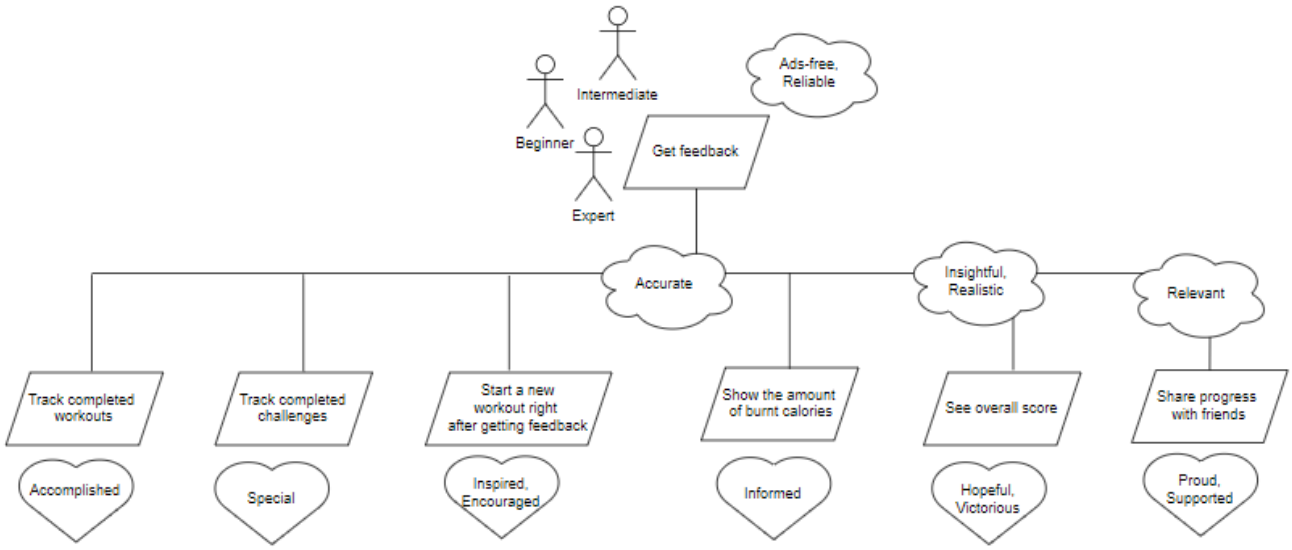

Figure 30. Get feedback

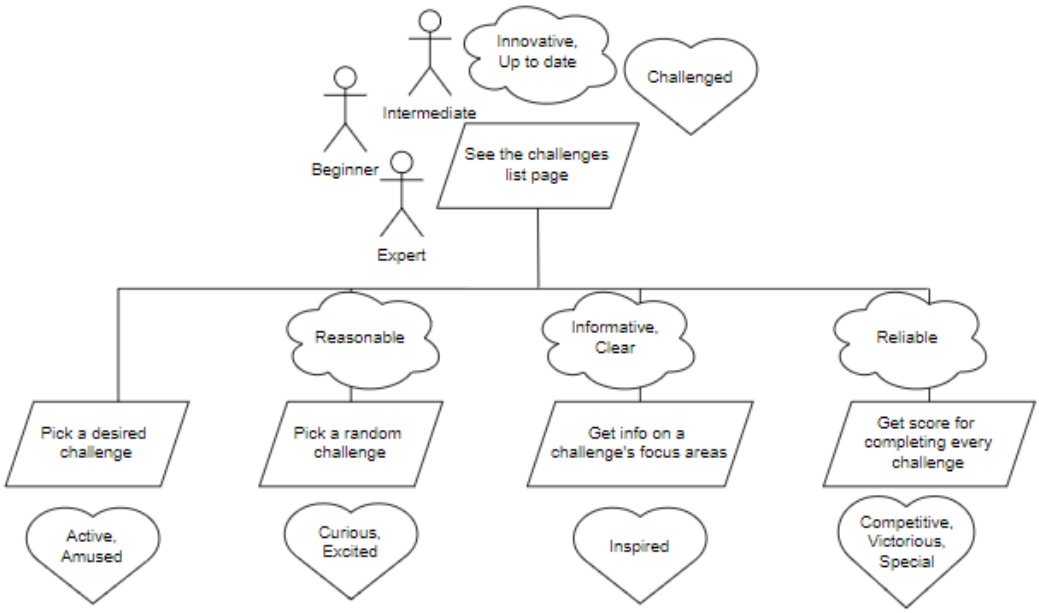

Figure 31. See the challenges list page

4.1.4 Extracting user stories

Since we already have our emotional goal models, we can extract the user stories from the leaves within the model hierarchy. For the role that refers to all three: beginner, intermediate and expert fitness level roles we will refer to as user. In other words:

User = Beginner, Intermediate, Expert

The user story format we described in our methodology addresses the option of including the quality and emotional goals inside the user story. However, since all of our emotional and quality goals are covered by the functional goals, as you can see in the emotional goal models of the previous section, it would be redundant to include them in user stories.

For the sub goal leaves: see the workouts list page (Figure 27), the following are the user stories.

1. As a user, I want to be able to pick a desired workout plan from the workouts list page.
2. As a user, I want to be able to pick a random workout plan from the workouts list page.
3. As a beginner, I want to get info on a workout's focus areas from the workouts list page.
4. As a user, I want to get info on a workout's intensity from the workouts list page.
5. As a user, I want to get info on a workout's burned calories from the workouts list page.
6. As a user, I want to get recommendations based on my fitness level from the workouts list page.
7. As a user, I want to be able to read about health benefits of having a workout.

For the sub goal leaves: create a 30 day plan (Figure 28), the following are the user stories.

1. As a beginner, I want to be able to automatically generate a 30 day plan.
2. As an expert, I want to be able to construct my own 30 day plan.
3. As a user, I want to be able to customize my 30 day plan.
4. As a user, I want to be able to include meditation/stretch/rest days in my 30 day plan.

5. As a user, I want to see a progress bar of burned calories during 30 days.

For the sub goal leaves: perform a fitness workout (Figure 29), the following are the user stories.

1. As a user, I want to see a timer for every workout move during my workout.
2. As a user, I want to get rewarded with a score for every workout move during my workout.
3. As a user, I want to have short breaks during my workout.
4. As an expert, I want to be able to construct my own workout.
5. As a user, I want to see the amount of burned calories during my workout.
6. As a user, I want to see the correct way of doing an exercise during my workout.
7. As a user, I want to see motivational messages throughout my workout.

For the sub goal leaves: get feedback (Figure 30), the following are the user stories.

1. As a user, I want to be able to track my completed workouts.
2. As a user, I want to be able to track my completed challenges.
3. As a user, I want to be able to start a new workout right after getting feedback.
4. As a user, I want to see the total amount of burnt calories in feedback page.
5. As a user, I want to see my overall score in feedback page.
6. As a user, I want to be able to share my progress with friends.

For the sub goal leaves: see the challenges list page (Figure 31), the following are the user stories.

1. As a user, I want to be able to pick a desired challenge from the challenges list page.
2. As a user, I want to be able to pick a random challenge from the challenges list page.
3. As a user, I want to get info on a challenge's focus areas from the challenges list page.
4. As a user, I want to get rewarded with a score for completing each challenge.

4.2 Answering RQ2

In this subsection we will present the results of the implementation of our fitness application. We present our code results which in fact show that we have covered and completed all of the user story requirements elicited in RQ2.

We have decided to name the app Motifit: and title combination of motivation and fitness to symbolize the emotional aspect that we will be taking into account.

The GitHub link[1] for Motifit project contains the source code.

4.2.1 Structure

To start the implementation of the Motifit fitness application we first elaborate the structure and the main components of the system.

Crucial system components include workouts and challenges list. See Figure 32 for the detailed view. A single workout has an id, title, description, role which is either Beginner, Intermediate or Expert, imagePath of the image that will be representing the workout in workouts list. The last property in a single workout is the workoutInProgress list. This list consists of the Move id's in a corresponding order of performing them. For simplicity reasons we will place move id as zero for the resting period. For example the Moves id list of [2, 3, 0, 5] means that the user will perform the 2nd move, then the 3rd move, then the user will rest, then the user will perform the 5th move.

The other important component is challenges list which is essentially an array of single challenge components. A single challenge has an id, title, description, score, imagePath and challengeInProgress list of move id's. The score of a challenge is the number that we will grant the user when the entire challenge is completed. Challenge in progress list represents the same logic as workout in progress list in Workout component.

Finally, we have the Move component of the system which is the granular representation in a fitness workout. It consists of an id, title, score, calories, instructions and imagePath. Please note that despite having moves id list in the challenge object, each challenge still has their own score. This is because unlike workout, doing a challenge will grant a specific amount of score only once the entire challenge is done. Also, the score of a challenge is not necessarily the sum of the scores of all moves in its moves list (unlike workout score). We will elaborate on this topic further in the subsection for each feature.

We also define the structure for creating a 30 Day Plan, which is essentially a list of thirty Day components. Please see Figure 33 A single Day has an id, isDone property showing if the workout of the day was completed or not and a workoutId, which points to the corresponding workout in the system.

Lastly, we elaborate on the structure of creating one's own custom workout plan. This consists of two main lists that are mutually exclusive: the chosen moves list and

[1] https://github.com/elentingas77/motifit

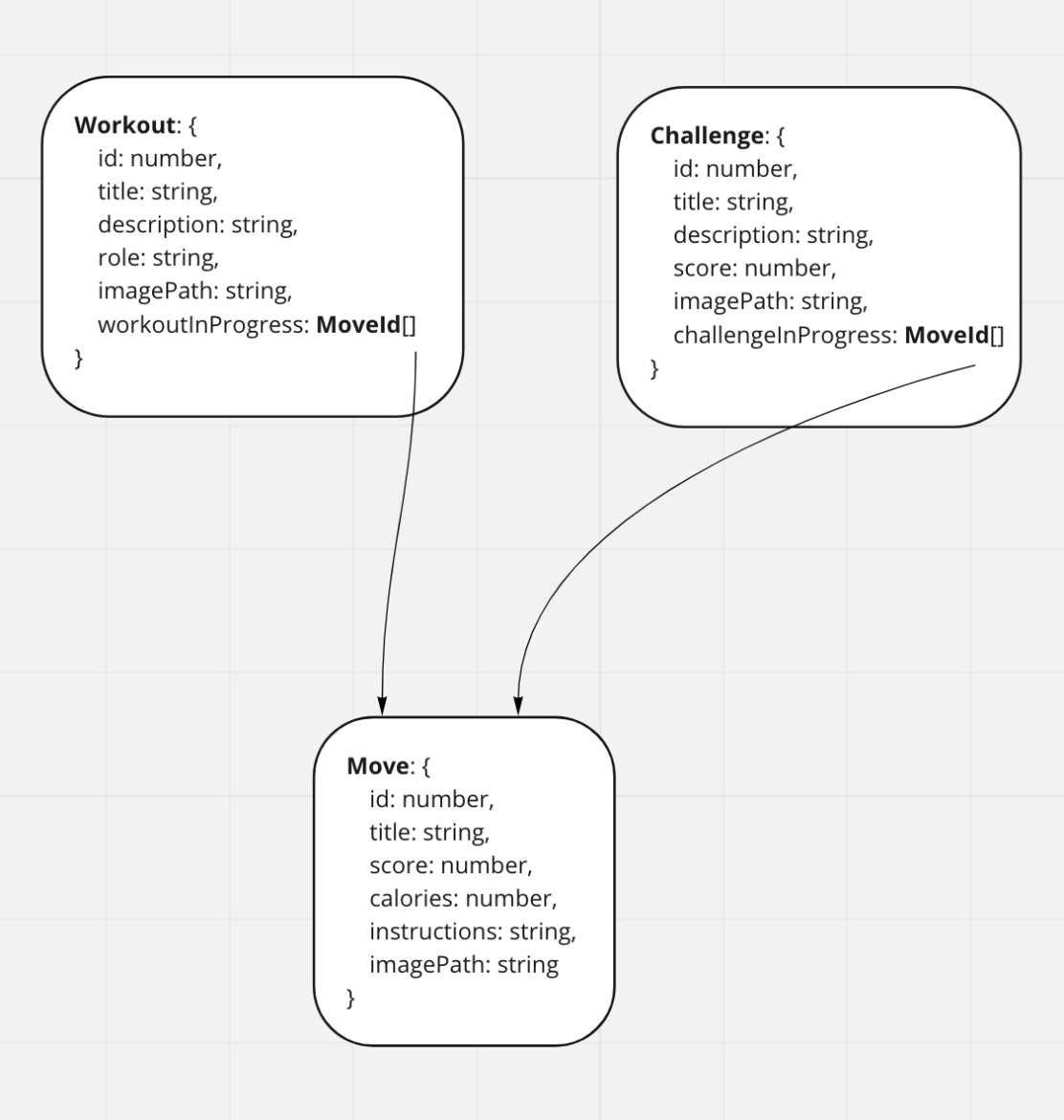

Figure 32. Structure of workouts and challenges

not chosen moves list. The combination of these lists is simply the entire moves set of the system. The chosen moves list includes the moves that the user selected and it has to have at least five items in order for the user to start their custom workout. Please see Figure 34 for details.

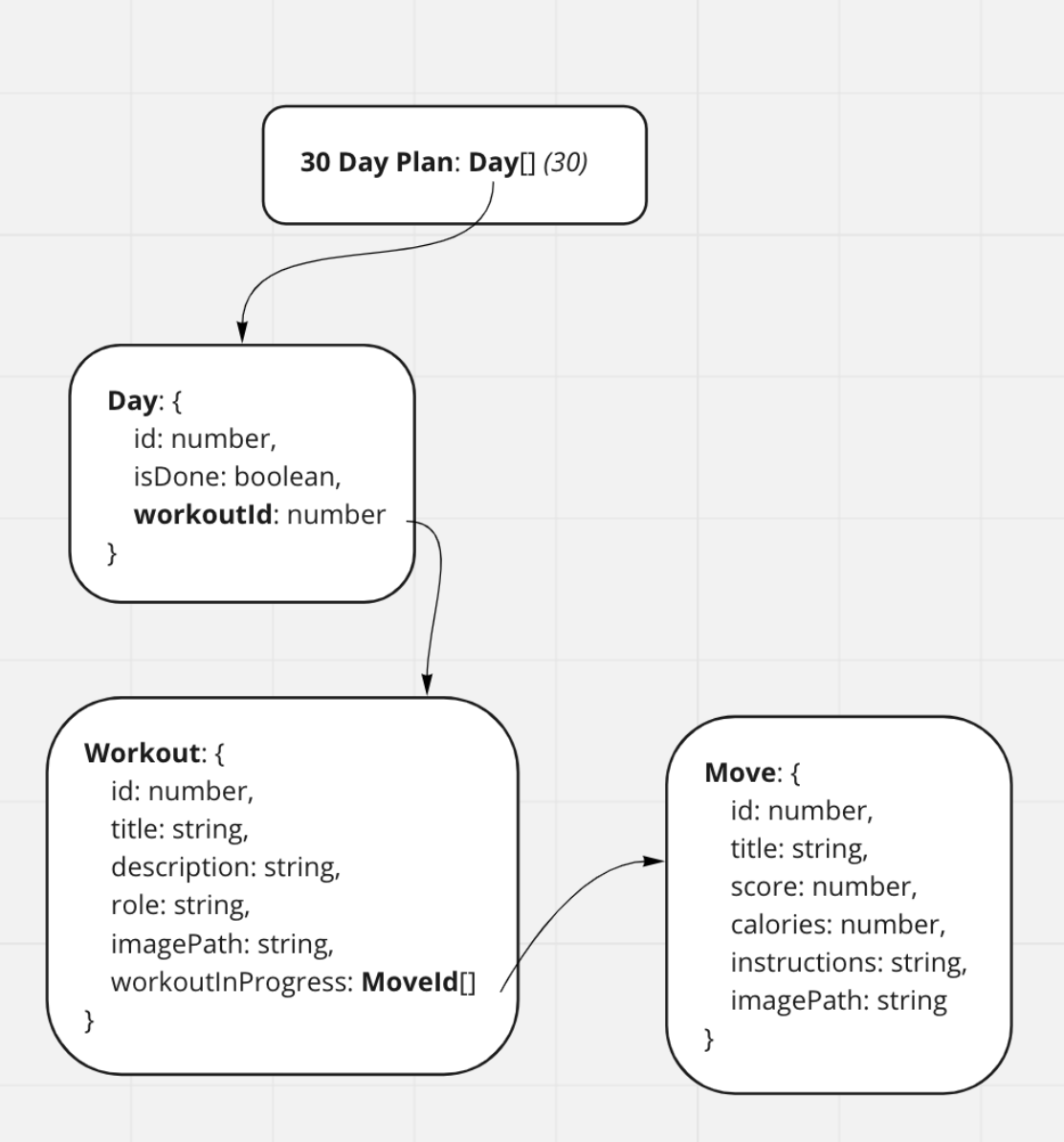

Figure 33. Structure of 30 Day Plan

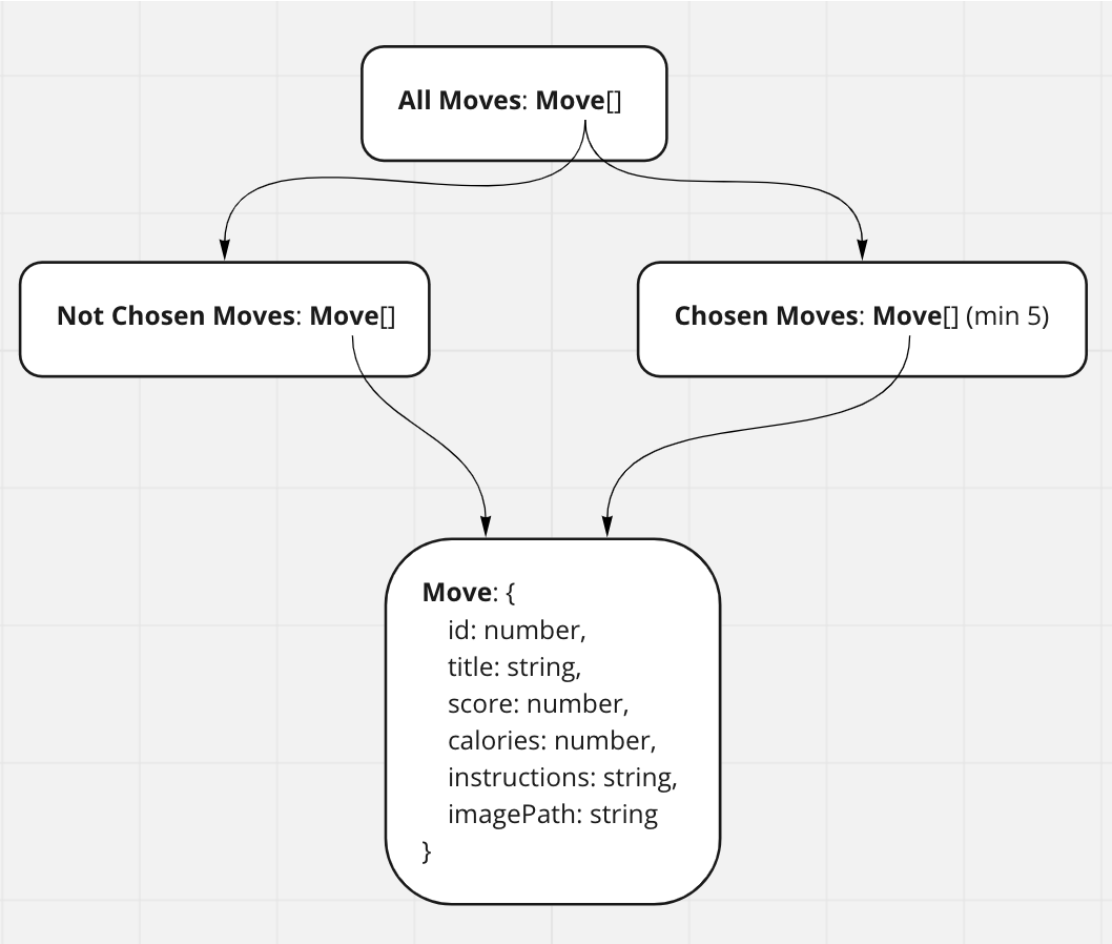

Figure 34. Structure of Construct My Workout

4.2.2 State management

For managing the state we have chosen the React Redux (https://react-redux.js.org/) library. It is a highly performant library that handles application state in a single object which consists of reducers. One of the main advantages of Redux is its scalability and easily access to the state from everywhere inside the app. Redux also solves issues of unnecessary rerendering of the state by only rerendering the components that have their immediate data manipulated. It has been around since React JS had only class components and also supports the hooks within React. For class components we would normally use the connect function to connect the Redux state with a single component making it a container. However, since the arrival of hooks, Redux also supports functions like useSelector and useDispatch and we can simply import the hook and use them within our components instead of having to connect every single component to the Redux state.

In order for our application to have consistent state, that is if the user closes the browser and comes back later, we want them so be able to access their previous data. Since the Redux state is initiated from the start every time the application is loaded, normally the user would lose their previous information in the state. We solve this problem with Redux Persist library (https://www.npmjs.com/package/redux-persist), which provides us the opportunity to save the redux state within the local storage and rehydrates the state whenever the user opens our application website again. In order for the user to have good UX feedback while Redux Persist library loads the state from local storage, we provide a default loading component to the PersistGate component. Figure 35 describes the details of the setup process.

```
ReactDOM.render(
  <React.StrictMode>
    <GlobalFonts />
    <Provider store={store}>
     <PersistGate loading={<ReactLoading type={"spinningBubbles"} color="#fff" />} persistor={persistor}>
      <I18nextProvider i18n={i18n}>
        <Router history={browserHistory}>
          <Home>
            <Routes />
          </Home>
        </Router>
      </I18nextProvider>
     </PersistGate>
    </Provider>
  </React.StrictMode>,
  document.getElementById('root')
)
```

Figure 35. React Redux and Redux Persist initialization

4.2.3 Routing

For the routing of the Motifit application, we use the react-router-dom library (https://reactrouter.com/). The main Routes we have inside our app are shown in Table 2.

Table 2. Motifit Routes

Title
/workouts
/30-day-plan
/challenges
/score
/construct-workout
/workout-in-progress/:id
/challenge-in-progress/:id
/ redirects to /workouts

On the top level, we have the Router component that accepts browser history in order to initialize. Figure 35 contains the code piece.

4.2.4 Fitness level roles

As we have stated before, we have three major user roles using our application; see Table 3.

Table 3. Motifit Roles

Roles
Beginner
Intermediate
Expert

Once the application is loaded, we grant the user a default role of a Beginner. This means that all the workout recommendations will first be showing beginner workouts. Also, the 30 Day Plan workouts that are available for the user to create, automatically receive, or customize are also workouts that are for the role of Beginner.

Figure 36 shows how we represent the feature in the app. Once the role is changed to a different one, we update the system to correspond the new data to the newly assigned role.

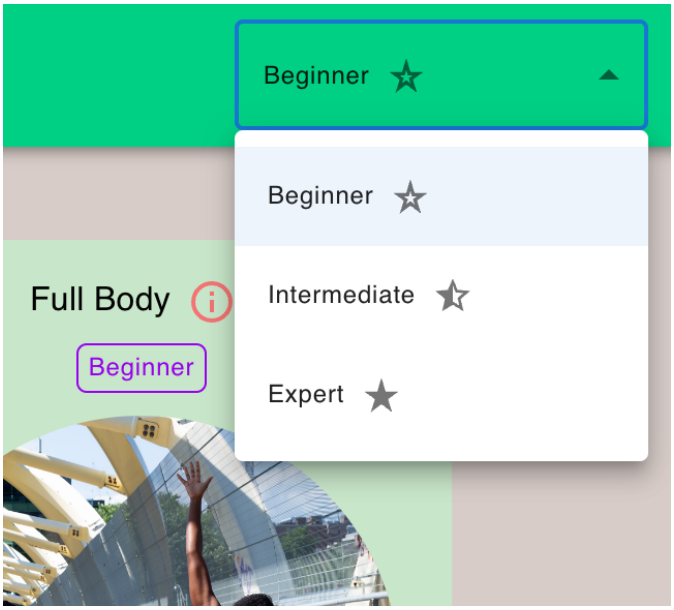

Figure 36. User Roles

4.2.5 Workouts

The workouts page is the route we land on when we open our website. See Figure 37. It contains a list of all the available workouts for the user. Depending on the role, it will first recommend the workouts with the user's role; after that, the list will show workouts for roles that are different from the user's one, in case the user wants to experiment with their fitness routine. A single item in the workouts list is a reusable React component that shows the information on the title, focus areas (description), and calories. The user can start a workout by clicking the purple Start button. Once the user hovers over the information icon, focus areas are displayed. For instance, the Strong Arms workout's focus areas are Deltoids, Biceps, Triceps, Forearms, and Shoulders, see Figure 38.

We have the feature to start a random workout, which is a button located right underneath the workout list carousel. To keep the user informed about the benefits of exercising, we display random facts about health and fitness, Figure 39. We have taken the points from the following source [Med].

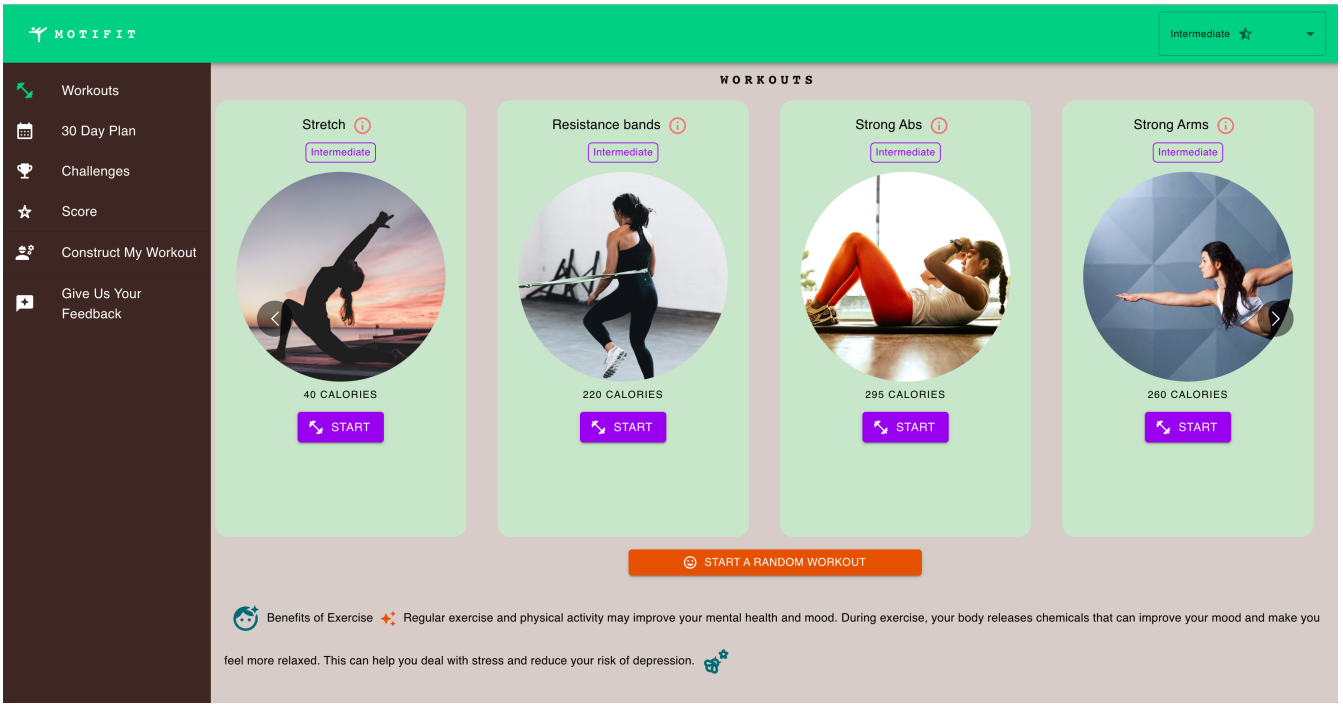

Figure 37. Workouts page

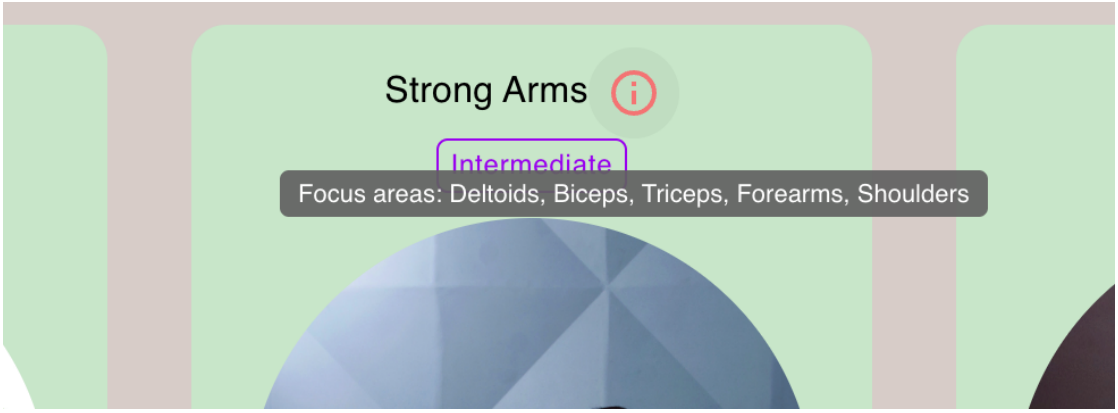

Figure 38. Focus areas

Figure 39. Benefits of exercise

4.2.6 30 Day Plan

The 30 Day Plan page allows users to generate an automatic plan or start one from scratch. Figure 40 shows the functionality within the app. The user can mark days as done or undone and customize their plan by selecting the dropdown workouts list and choosing the desired workout routine, see Figure 41.

For simplicity reasons, we incorporate rest days with workout plans by marking rest days to have workout id as zero in the day object. For example, if the day has a workoutId property as 0, we already know that it is a rest day.

Selecting the start from scratch button cleans the entire plan (Figure 42) and lets the user start constructing it according to their workout preferences (Figure 43). It is important to note that the available workouts for the 30 Day Plan correspond to the user's role. If the user is an Expert, then the workouts list will include routines for the Expert role.

We also introduce the feature of the calories burned progress bar, see Figure 40. This lets the users see their total amount of calories to burn within 30 days and shows the already burned amount of calories. It updates dynamically according to the number of workouts done or undone.

We should also note that we let the users themselves click the done button since after starting the workout, the user may decide to finish it early. If the user completes a workout early, we still incorporate it in their workout score (just with fewer scores and calories). However, the user may not want to mark a workout as done in this type of case, so we let them decide on the completion status. In other words, the 30 Day Plan is more of an organization and guidance tool that allows users to track their progress and be proud of it instead of having strict rules on when a workout is completed or not.

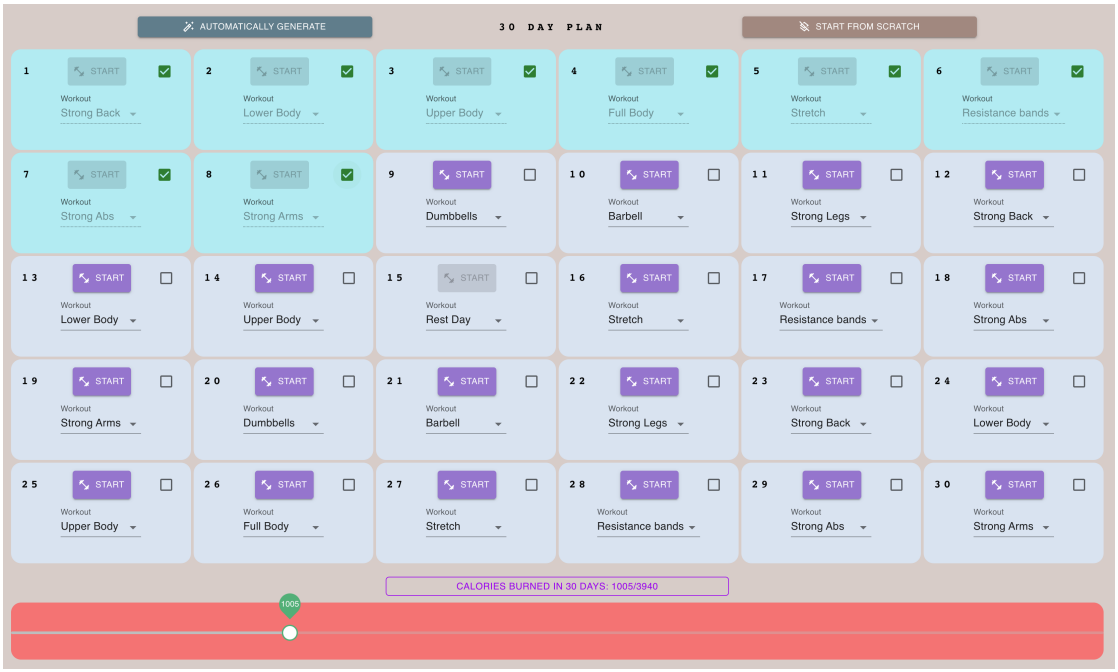

Figure 40. Automatically generated 30 Day Plan

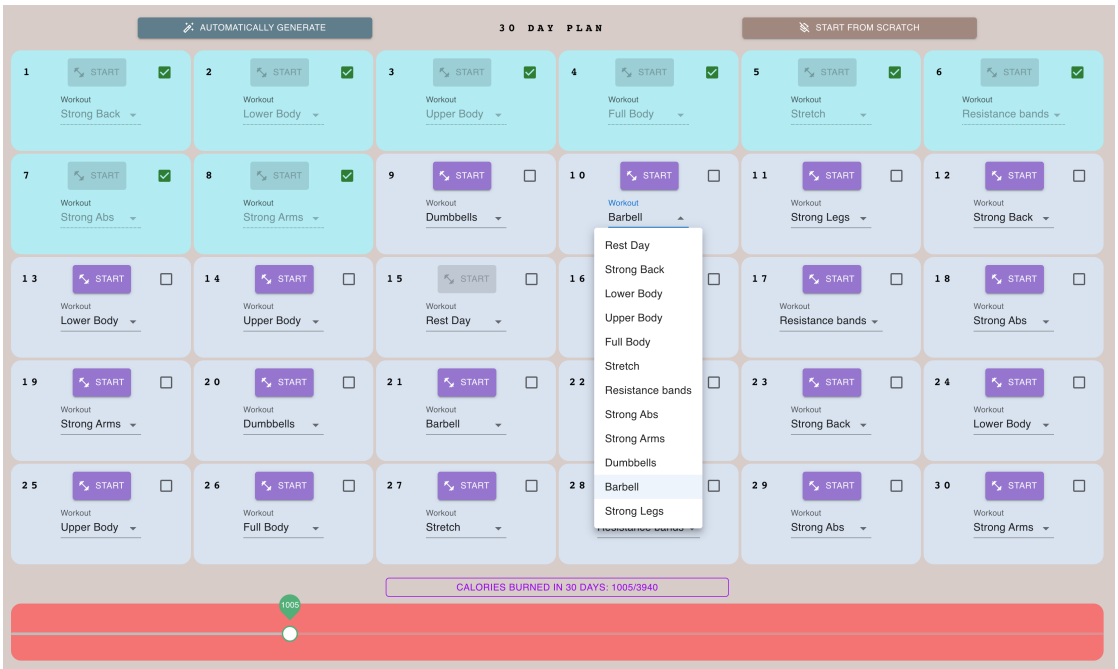

Figure 41. Customizing the automatically generated plan

Figure 42. Starting a plan from scratch

Figure 43. Editing the plan from scratch

4.2.7 Challenges

The challenges page shows a list of single challenges. A single challenge list item is a reusable React component that includes information about the challenge title and focuses on areas hovering over the information icon and its score.

We named the challenges after various Estonia cities to make them more enjoyable. The user can start a challenge by choosing the yellow Start button. The user can also start a random challenge from the button right below the challenges list if they feel adventurous.

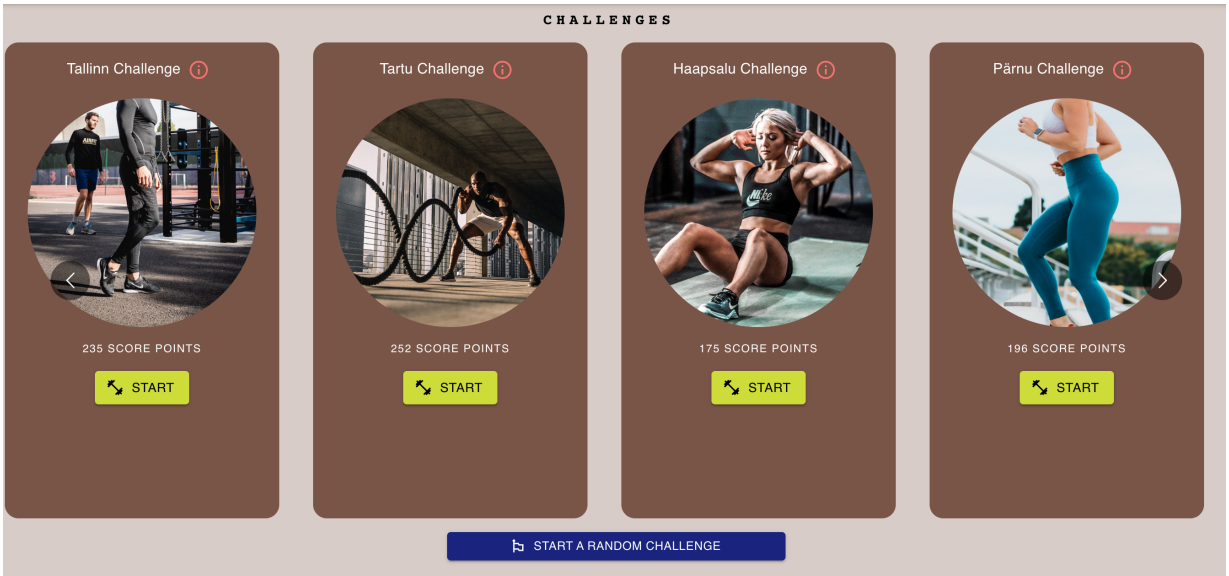

Figure 44. Challenges page

4.2.8 Workout and Challenge in progress

Once the user clicks the Start a workout button, they are navigated to the "/workout-in-progress/:id" page. The workout in progress page reads the incoming data on the given workout and already knows the number of moves, rest periods, calories, and scores for each move. It starts to show the moves in chronological order and accumulates scores and calories after every move is finished, see Figure 45. Each move is 120 seconds, i.e., 2 minutes. There is a timer that counts down the seconds for a move, and also there is shown the fitness level role for which a particular workout is meant. If the user wants to finish a workout early, we finalize the score and calories accumulated so far and add it to their overall score, redirecting them to their Score page. We would count it as a workout if at least one move were successfully finished. If the user continues their

workout journey until the end, then after the last workout move, we redirect them to their Score page information, which is freshly updated.

We use the workout instructions and media resources from the following source [Mus].

To make the experience interactive and interesting, we add a text-to-speech generator to our application so that the user does not have to come close to the website screen every time a move is switched. We use the SpeechSynthesisUtterance interface of Web Speech API [MDN]. Thus, whenever a particular move changes, the speech handler reads the instructions for the user.

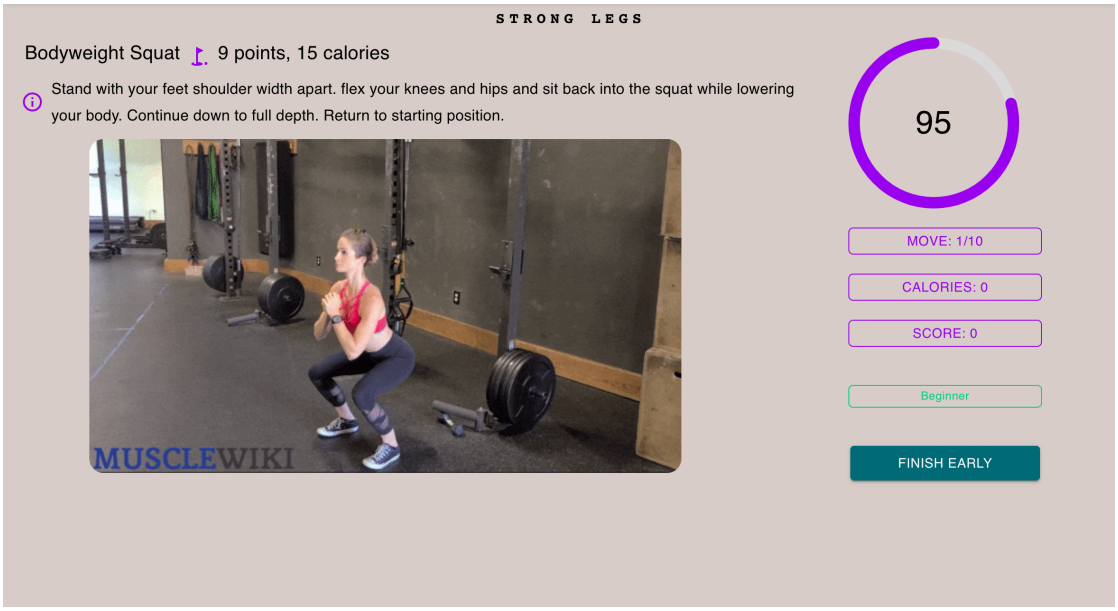

Figure 45. Start: workout in progress

Once a move is switched, and a new move is about to be displayed, right before reading the instructions, the speech handler first reads a random motivational message for the user to keep their emotional state positive. We also show a snack bar containing the message on the bottom left corner of the screen; see examples 46 47 48. The snack bar automatically disappears after a couple of seconds.

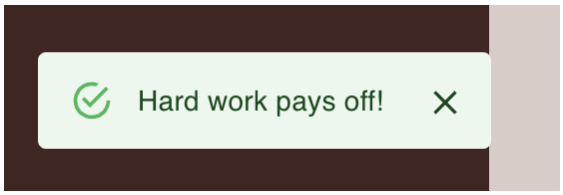

Figure 46. Motivational message example 1

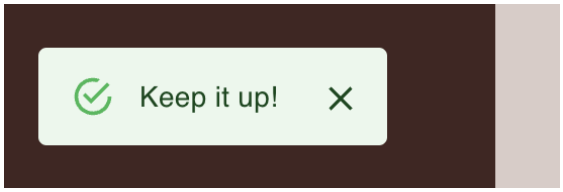

Figure 47. Motivational message example 2

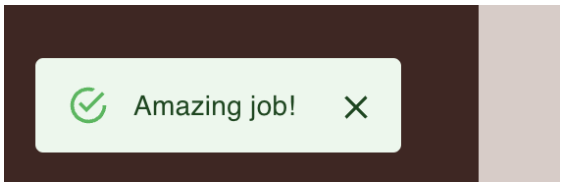

Figure 48. Motivational message example 3

The list of motivational messages that we arbitrarily show to the user can be seen in Table 4.

Table 4. Motivational messages

Motivational messages
'You are doing so great!'
'Amazing job!'
'Hard work pays off!'
'You are so brave!'
'Keep it up!'
'Do not give up!'
'Keep up the great work!'
'You are unstoppable!'

The rest period shows the timer and simply waits until the 120 second period is over to move on, see Figure 49. During rest time, the user is granted neither a score nor a calorie count.

To see a full example of a workout in progress, we can take a look at the following Figure 50. In this case, the user has reached the 7th move, is currently supposed to perform the "Crunches" move, already heard the speech handler read the motivational message to them (since the snack bar already has disappeared), and is now hearing the instructions on how to perform a crunch move. The workout is designated for a Beginner role. The user so far has accumulated 65 calories and 43 score points, which will later be

Figure 49. Rest time

reflected on their overall score.

It is also worth noting that for Beginners, the workouts have 10 moves; thus, the duration of such workouts is 20 minutes. For Intermediates, workouts have 20 moves, making the overall workout 40 minutes long. And for Experts, the workouts have 30 moves, making them 1 hour long, since every move's duration is 2 minutes.

The challenge in progress is quite similar to the workout in progress (see Figure 51), except during a challenge, we do not show motivational messages, not to distract the user from their goal and also to let them overcome the mental challenge of finishing the workout until the end. We also do not provide the score for each move since the score for a challenge is only received when the challenge is fully completed. Instead of the Finish early button, we show a Quit button, which redirects the user to the challenges list page. If the user finishes the entire challenge, we bump their completed number of challenges, update the score and the total calories, and redirect them to their Score page.

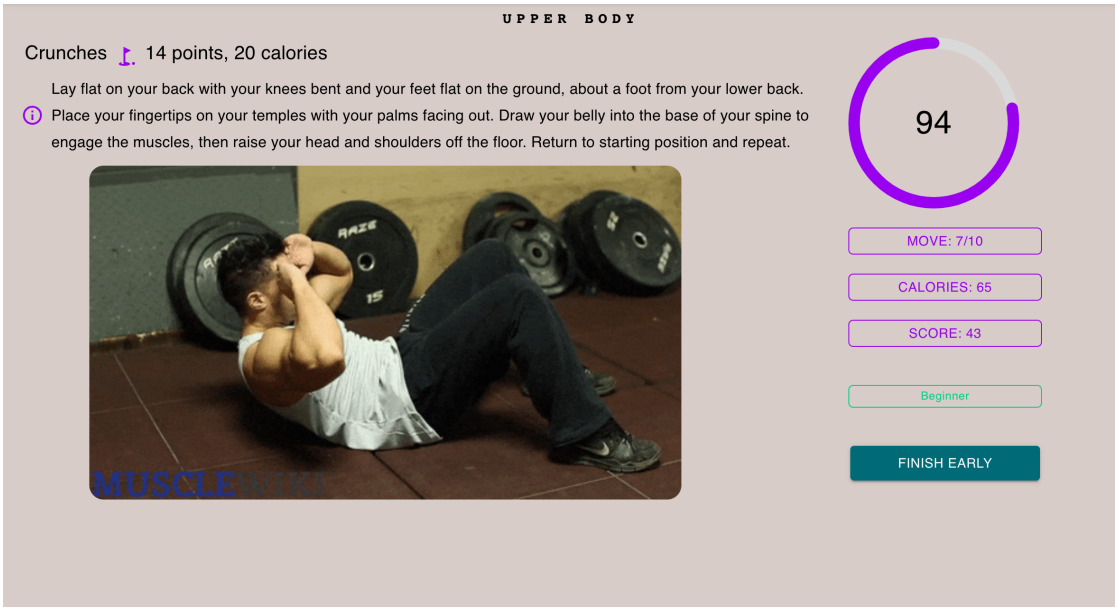

Figure 50. Workout In Progress

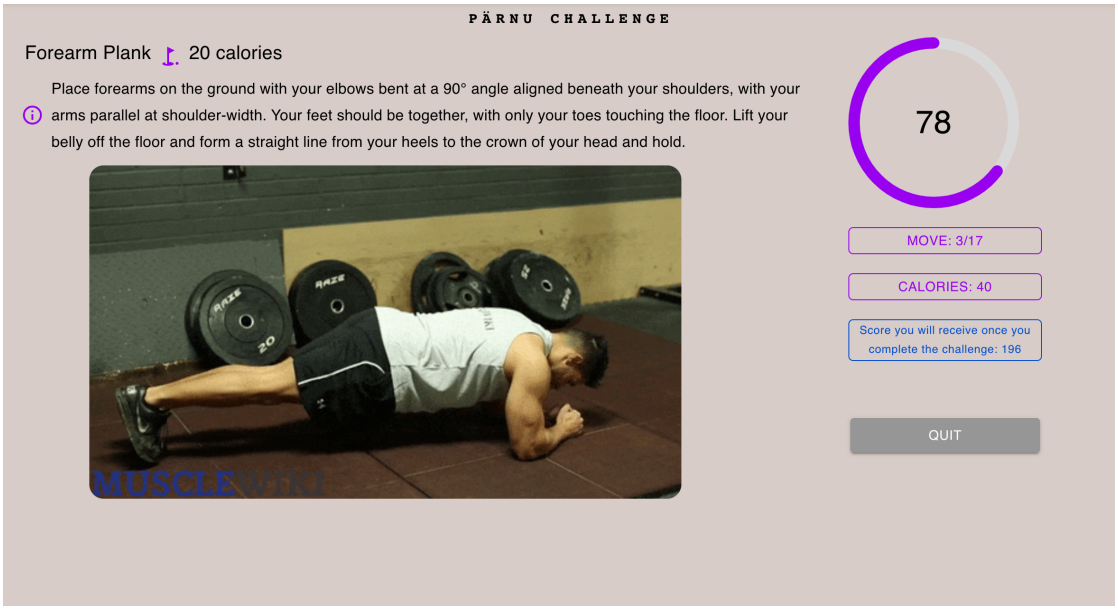

Figure 51. Challenge In Progress

4.2.9 Score

As mentioned above, the score statistics of the user are accumulated in the state and saved to be viewed anytime the user needs. We show information about the number of completed workouts, the number of completed challenges, the total calorie count, and the overall score, see Figure 52.

Figure 52. Score page

We let the user share their progress with friends in multiple social media platforms including Twitter (Figure 53), WhatsApp (Figure 54) and Tumblr (Figure 55).

In case the user gets excited by their score and is motivated to start another fitness workout, we show a button that lets them immediately choose and start a new one.

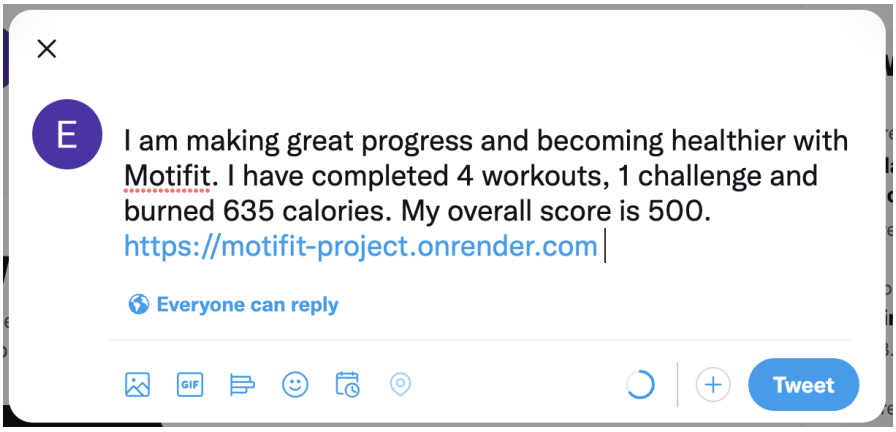

Figure 53. Share to Twitter

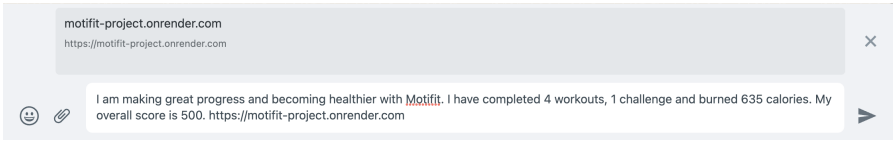

Figure 54. Share to WhatsApp

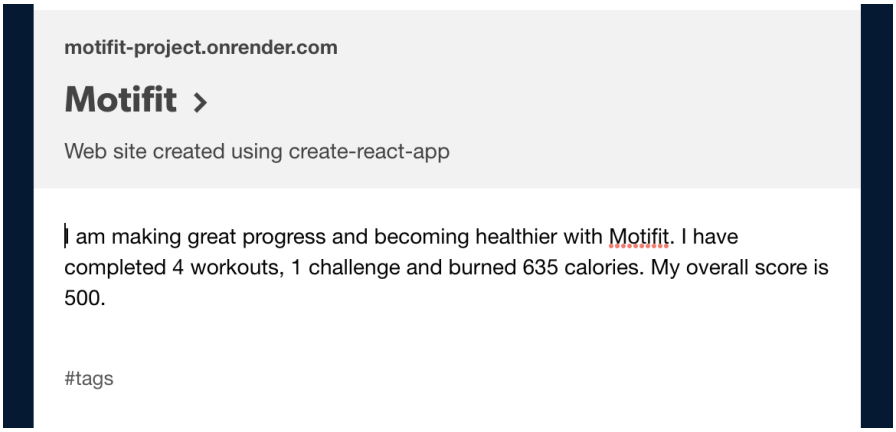

Figure 55. Share to Tumblr

4.2.10 Construct My Workout

The final requirement that we need to cover is the feature of letting the user construct their own workout. To achieve this, we separate two mutually exclusive lists. The first list contains all the available workout moves, and the second list contains the chosen workout moves by the user. We then let the user select desired workouts and transfer them to the chosen workout moves list, see Figure 56. To start the workout, the number of chosen moves must equal at least 5. Once there are 5 or more moves in the list, the Start workout button is enabled, and the user is navigated to the workout in progress page when clicking it, see Figure 57. The amount of score and calories collected by the custom workouts are also added to the overall feedback score.

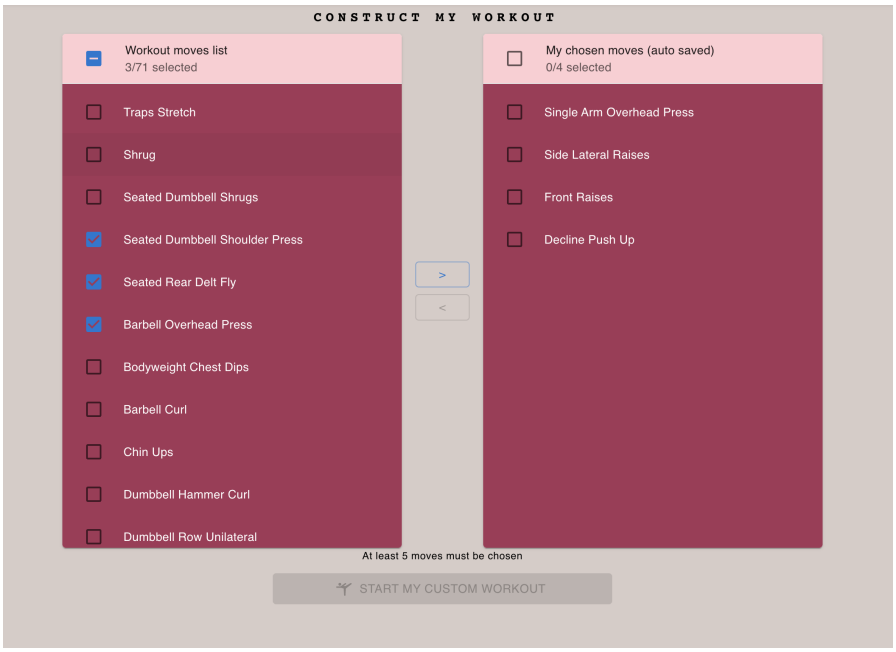

Figure 56. Construct My Workout, less than 5 moves

4.2.11 Running, testing and building

In order to process such a big number of media files for workout moves, we store the media resources on a separate server. Whenever a workout move needs to be displayed to the user, we use the remote server to refer to the media file paths.

The running, testing, and building steps of our app are covered in the Read Me file in the GitHub repository. The first step for all of them is to install the package dependencies with npm install or npm ci. We recommend using npm ci since it does not manipulate

Figure 57. Construct My Workout, user can start

the package.lock file, and the result of the install is always exactly the same. For running the app, we run npm start.

We have implemented various e2e tests with Cypress library to ensure that the significant features and resources are correctly loaded within the app. For running the e2e tests, we run npm run cypress:run. Please see the terminal result of the e2e tests in Figure 58.

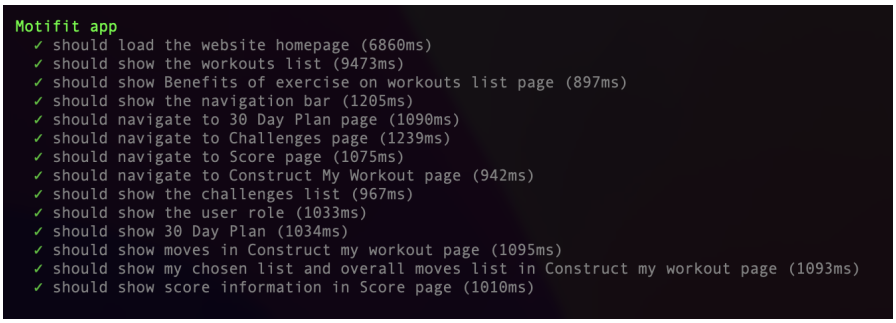

Figure 58. End to end tests

Finally, for building the application, we run npm run build, and it will generate a build folder within our repository. Once we commit and push the changed build folder

into the remote GitHub server, our app will be automatically deployed to https://motifit-project.onrender.com/ website.

4.3 Answering RQ3

To evaluate the results implemented in RQ2, we created a System Usability Scale (SUS) survey and attached the survey link to our website. Whenever a user clicks the Give Us Your Feedback button in Figure 59 from the navigation bar, a new window is opened with the SUS survey to be filled.

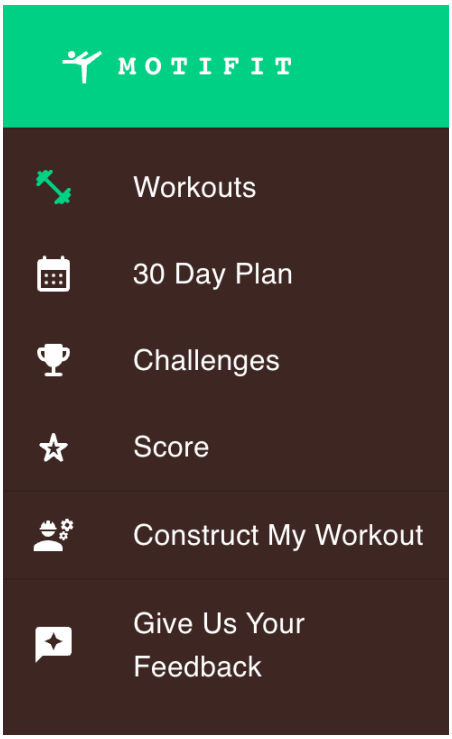

Figure 59. Send feedback button

These are the questions we asked the users in the survey, see Table 1 and Figure 61. The format of possible answers can be seen in Figure 60.

We have received 21 responses from the users, and the following Table 5 summarizes the score results from each one of them. Calculating the average of all the SUS scores, we get around 84.5, which is a positive System Usability Scale score for our interactive fitness application.

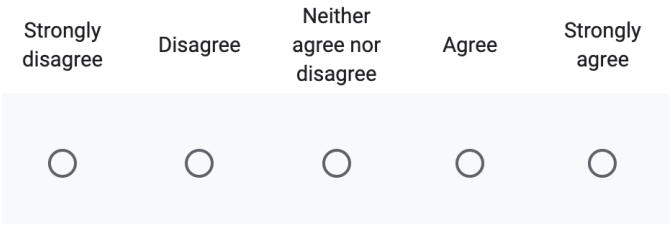

Figure 60. SUS answer format

Table 5. SUS results

Number	**Normalized sum**	**SUS score**
1	37	92.5
2	33	82.5
3	35	87.5
4	36	90
5	35	87.5
6	32	80
7	36	90
8	31	77.5
9	35	87.5
10	34	85
11	34	85
12	32	80
13	30	75
14	33	82.5
15	35	87.5
16	37	92.5
17	32	80
18	33	82.5
19	32	80
20	35	87.5
21	33	82.5

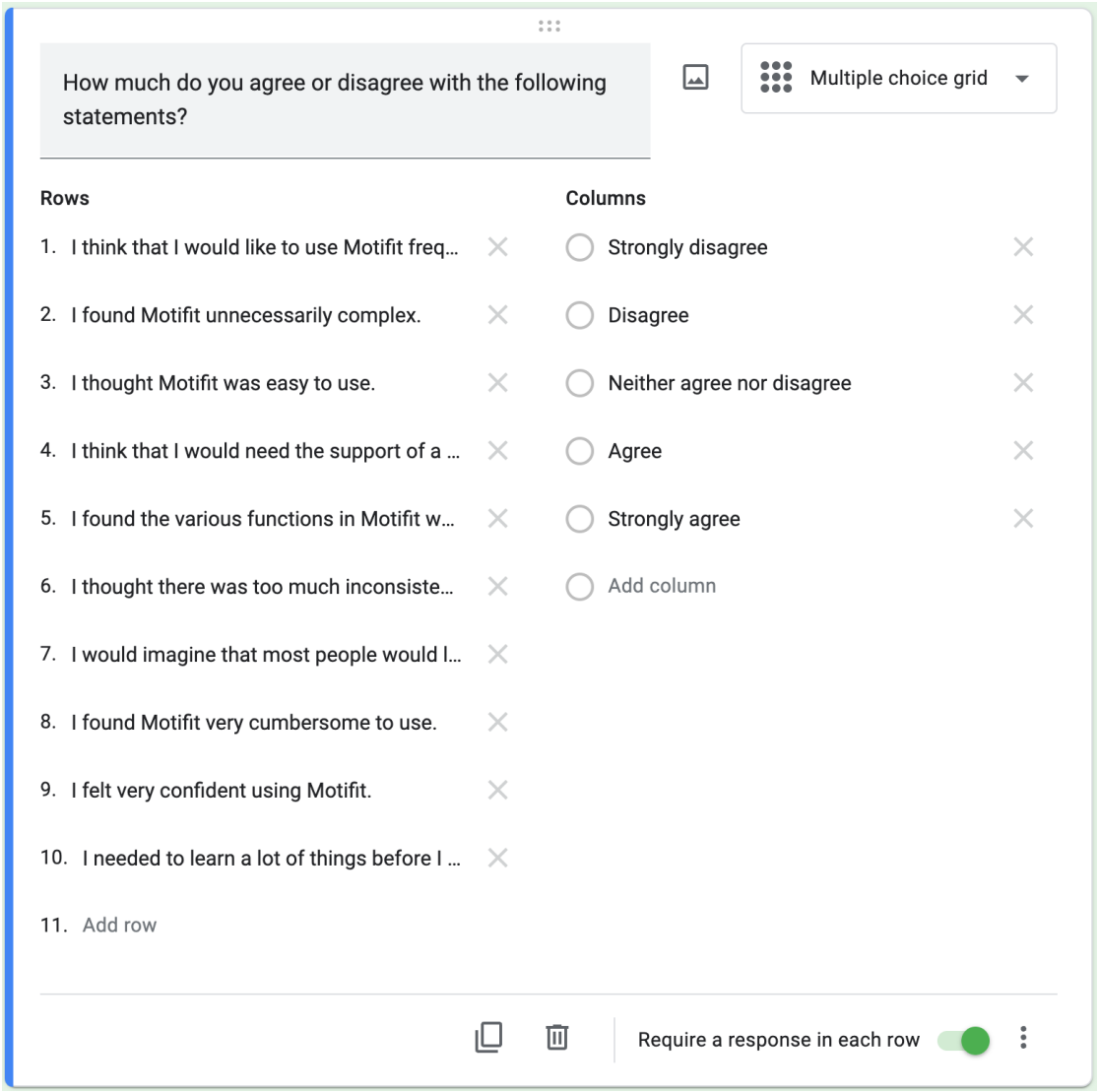

Figure 61. SUS Survey

5 Discussion

We have analyzed the research done behind the usage of emotional requirements in software engineering. We saw that there had been many indicators of how crucial it is for a socio-technical system to take into account a user's emotions—as abstract and subjective as emotions are, not incorporating them within the requirements elicitation process might lead to complete abandonment of the user's emotional needs. For that reason, in our work, we have used emotional goal models to document users' emotional requirements and address them by creating functional goals. After that, we extracted the user story requirements from the emotional goal model hierarchy leaves and documented them in our findings.

After eliciting the main requirements, we created a real-world practical application that real life users would use to evaluate the improvements and effects of our theoretical work. After completing the source code and successfully deploying the interactive fitness application that we named Motifit, we conducted a System Usability Scale survey to assess users' overall opinion of the application. The SUS score we received was high enough to verify the importance of emotional requirements in socio-technical systems and the importance of addressing them in the early process of requirements elicitation.

6 Threats to validity

It would be useful to note the threats to validity while acquiring the survey results for extracting Do/Be/Feel/Role data. Since we required input from many users, we organized the survey online. However, with online surveys, it is possible that some participants would need personal guidance about the context of a question to fill in the answer more accurately.

7 Conclusion and future work

To conclude our work, we have discussed the importance of incorporating users' emotional requirements within the early stages of the requirements elicitation process in software engineering. After that, we conducted an extensive survey with 182 participants and extracted the Do/Be/Feel/Role data from the results. Using the Do/Be/Feel/Role data, we constructed the emotional goal models and addressed every emotional goal with a new functional goal to make sure the real life application reflects the emotional needs. Then, we implemented a real-life application, an interactive fitness app we named Motifit. After deploying the source code, we decided to conduct a SUS survey where we asked the users first to experience our application and then fill in the questionnaire. We received the 84.5 SUS score, which is a positive result for a system. Thus, we conclude that incorporating

the user's emotional requirements in the early stages of software engineering improves the user's experience and long-term reflection on the system.

For future work, we would like our practical application to have the ability to sync within various browsers so that users can utilize their fitness data with multiple devices. It would be a significant improvement to incorporate health monitoring gadgets like smart watches into the functionality. In terms of theoretical future work, an analysis can be done to resolve conflicts arising throughout the requirements elicitation process. We have seen examples of conflicting emotional needs of the users, like Expert roles wishing to have full control over their workout routine while Beginner roles want to have automatic guidance from the system.

8 Acknowledgements

I would like to express gratitude to my supervisor Ishaya Peni Gambo for all his help and guidance throughout the whole process.

References

[BJvB02] Todd Bentley, Lorraine Johnston, and Karola von Baggo. Putting some emotion into requirements engineering. In *Proceedings of the 7th Australian workshop on requirements engineering*, pages 227–244. Citeseer, 2002.

[BLLS+19] Rachel Burrows, Antonio Lopez-Lorca, Leon Sterling, Tim Miller, Antonette Mendoza, and Sonja Pedell. Motivational modelling in software for homelessness: lessons from an industrial study. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 297–307. IEEE, 2019.

[Coh04] Mike Cohn. *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.

[CPCLSAGC11] Ricardo Colomo-Palacios, Cristina Casado-Lumbreras, Pedro Soto-Acosta, and Ángel García-Crespo. Using the affect grid to measure emotions in software requirements engineering. 2011.

[CPHLGCSA10] Ricardo Colomo-Palacios, Adrián Hernández-López, Ángel García-Crespo, and Pedro Soto-Acosta. A study of emotions in requirements engineering. In *World Summit on Knowledge Society*, pages 1–7. Springer, 2010.

[Fla16] LJ Flanders. *Cell Workout: At home, no equipment, bodyweight exercises and workout plans for your small space*. Hachette UK, 2016.

[MDN] MDN. Web speech api - web apis: Mdn.

[Med] MedlinePlus. Benefits of exercise.

[Mei] Jens Oliver Meiert. Revitalizing sus, the system usability scale · jens oliver meiert.

[MHB+21] Denys JC Matthies, Thorleif Harder, Franz Bretterbauer, Viktoria Ginter, and Horst Hellbrück. Fitfone: Tracking home workout in pandemic times. In *The 14th PErvasive Technologies Related to Assistive Environments Conference*, pages 272–276, 2021.

[Moo21] Kerli Mooses. An ideal physical activity app for adults-what should it be like? requirements of adult users. In *RESOSY@ APSEC*, 2021.

[MPLL+15] Tim Miller, Sonja Pedell, Antonio A Lopez-Lorca, Antonette Mendoza, Leon Sterling, and Alen Keirnan. Emotion-led modelling for people-oriented requirements engineering: the case study of emergency systems. *Journal of Systems and Software*, 105:54–71, 2015.

[Mus] Musclewiki. Musclewiki.

[Nor04] Donald A Norman. *Why we love (or hate) everyday things*. Perseus Books Group, 2004.

[PBC15] Vera Pedragosa, Rui Biscaia, and Abel Correia. The role of emotions on consumers' satisfaction within the fitness context. *Motriz: Revista de Educação Física*, 21:116–124, 2015.

[PEM03] Frauke Paetsch, Armin Eberlein, and Frank Maurer. Requirements engineering and agile software development. In *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, pages 308–313. IEEE, 2003.

[PPK+11] Rumyana Proynova, Barbara Paech, Sven H Koch, Andreas Wicht, and Thomas Wetter. Investigating the influence of personal values on requirements for health care information systems. In *Proceedings of the 3rd Workshop on Software Engineering in Health Care*, pages 48–55, 2011.

[RB05] Isabel Ramos and Daniel M Berry. Is emotion relevant to requirements engineering? *Requirements Engineering*, 10(3):238–242, 2005.

[ST10] Alistair Sutcliffe and Sarah Thew. Analysing" people" problems in requirements engineering. In *2010 ACM/IEEE 32nd International Conference on Software Engineering*, volume 2, pages 469–470. IEEE, 2010.

[Sut11] Alistair Sutcliffe. Emotional requirements engineering. In *2011 IEEE 19th International Requirements Engineering Conference*, pages 321–322. IEEE, 2011.

[Tho] Nathan Thomas. How to use the system usability scale (sus) to evaluate the usability of your website.

[TNR+17] Tanel Tenso, Alexander Horst Norta, Hannes Rootsi, Kuldar Taveter, and Irina Vorontsova. Enhancing requirements engineering in agile methodologies by agent-oriented goal models: Two empirical case

studies. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 268–275. IEEE, 2017.

[TS08] Sarah Thew and Alistair Sutcliffe. Investigating the role of'soft issues' in the re process. In *2008 16th IEEE International Requirements Engineering Conference*, pages 63–66. IEEE, 2008.

[TSP+19] Kuldar Taveter, Leon Sterling, Sonja Pedell, Rachel Burrows, and Eliise Marie Taveter. A method for eliciting and representing emotional requirements: Two case studies in e-healthcare. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 100–105. IEEE, 2019.

Appendix

I. Glossary

- POSE - people-oriented software engineering
- US - user stories
- SUS - system usability scale
- UI - user interface
- RQ - research question
- app - application
- UX - user experience
- API - application programming interface

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Elen Tingas**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Design of an Interactive Fitness Application based on user's Emotional Requirements,

   supervised by Ishaya Peni Gambo, PhD.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Elen Tingas
08/08/2022