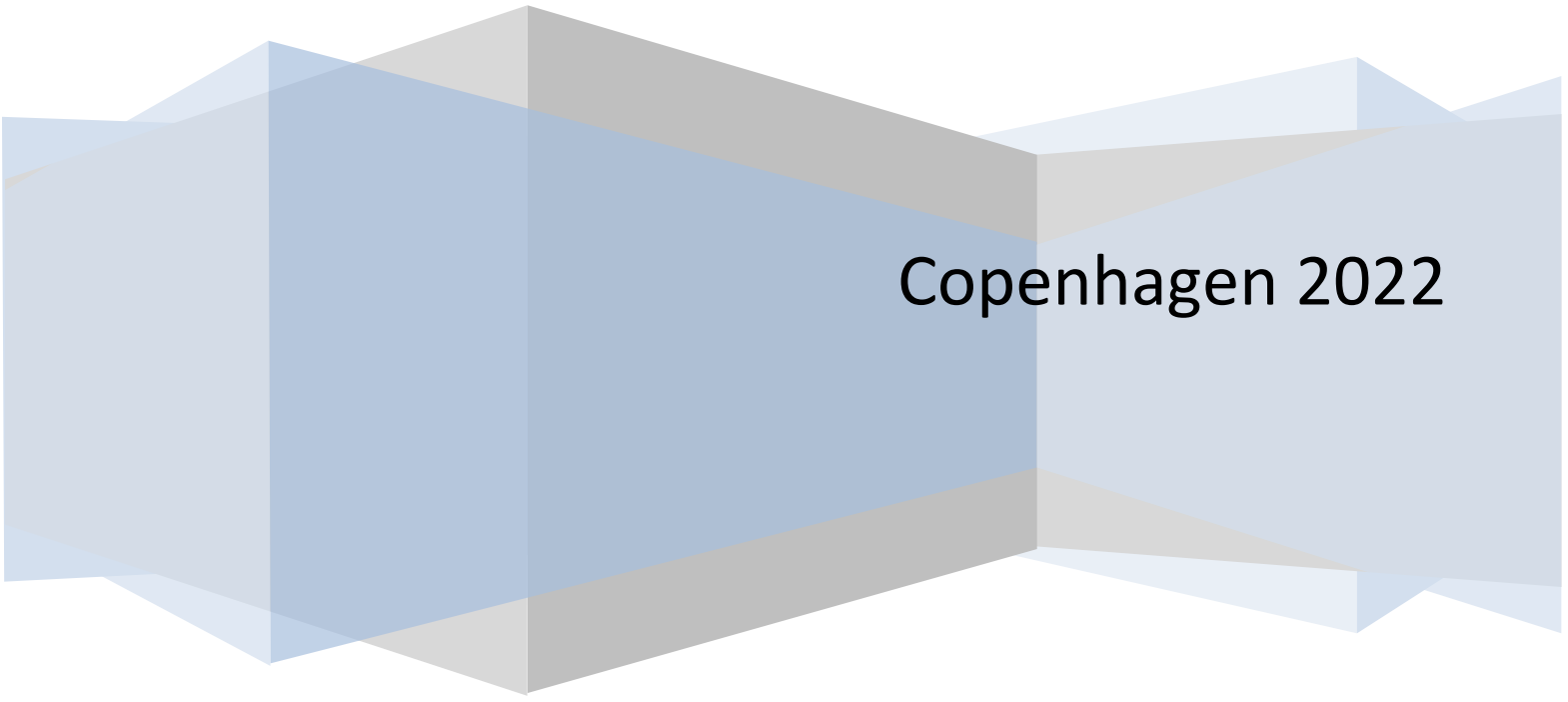
Fitness Buddy Finder Case

Neilos Kotsiopoulos – 20896 – neko@itu.dk
Andreas Elsberg – 20929 – elsb@itu.dk
Eduardo Trindade – 20890 – edtr@itu.dk
Marcus Holst Garner – 20934 – mahg@itu.dk
Nikoline Stensbæk Vejby – 20920 – nvej@itu.dk

Project Supervisor: Zachery Russel Kuntz
Course Manager: Elda Paja

Case Project in Software Engineering



Copenhagen 2022

Table of Contents

TABLES OF PICTURES, TABLES AND FIGURES

Abstract

As the importance of mental and physical wellbeing is increasing rapidly in the fast paced and globalized society we are currently living in, there is an increasing need for people to steam off, keep healthy and find ways to keep up with daily life. Loneliness and lack of motivation to exercise tend to lead people to have health issues and depression marks. A software, matchmaking peoples' needs in terms of combining their exercising schedule could aid in having both a healthier lifestyle and boosting the social life of the user. With a fitness buddy that matches your exercise needs it would be easier to create a habit around exercising regularly, and thereby reap the benefits of living a healthy life.

I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 1 of 16
{Explore your Case}

Author	Date	Version	Status	Description
NK	08/09	1.1.2	DR	Software Qualities
AE	28/09	1.1.3	DR	Explore your case
AE	28/09	1.1.4	DR	Software functions
ET	29/09	1.1.5	DR	User case stories
NSV	29/09	1.1.6	DR	Software qualities
MHG	29/09	1.2.1	DR	Revised software qualities and functions
ATM	29/09	1.1.1	RW	Reviewed
ATM	14/12	1.1.1	FV	Reviewed

Copenhagen 2022

.

1) Explore your Case

Problem:
You might often find yourself going to the gym alone, even though the gym is packed with people. The same applies in several exercise environments such as a park or an open basketball/football court. We know that working out with other people has positive effects both on your mental state and on the outcome of your workout, and that working out together makes you more motivated and less prone to skipping your workouts. Your mental health depends heavily on your physical status and vice versa, therefore exercise does not have to be explicitly a physical activity but could also be a mindfulness related activity (Steinhilber, 2007).

Literature review:
We set out to find research to back up our hypothesis, that exercising in groups can greatly increase the desired health benefits of exercising in groups. The article “The health benefits of working out with a crowd" describes numerous benefits of group exercise (Steinhilber, 2007).
The article describes how commitment to an exercise routine is positively impacted by participating in group exercise, rather than exercising alone. A study referenced in the article found that 95% of individuals who started a weight-loss program with friends completed the program, while only 76 of those who started a similar program alone were able to complete it.
Furthermore, research shows that working out with experienced practitioners of a given form of exercise can improve workout time and intensity by up to 200 percent.
It is well known that it can be difficult to maintain a workout routine alone and that working out with a friend, or a mentor is highly motivating. A study conducted by the International Journal of Stress management, as described in the article, shows that people who exercise with a friend expressed that they felt calmer afterwards compared to those who exercised alone (Plante, 2001).
There are many more benefits related to group exercise described in the article, and it might prove an interesting read, for those wanting to look further into the health benefits of group exercising.

The idea:
The idea will build upon existing apps described by fit bottomed girls (Jenn 2015). Most of the apps described in this article are focused on a single form of exercise, such as running, while our app idea is centered around building networks for any kind of exercise.
We want to set our app apart from existing apps by also including the aspects of mental well-being connected to exercising as part of a group. Our app will be able to make

suggestions for easily accessible forms of exercise as well as matching users together with similar exercise preferences. The goal is to make exercising fun and engaging for the user.

1.1) Software Qualities

The app should be designed to contain the following primary qualities:

- User Experience: The whole project idea depends on current user needs. The user input is important and should be reviewed on a regular basis. The applications' usefulness, usability and desirability will define the whole project's success. The user experience is focused on building a network of exercise partners, so that the user can read the benefits of group exercise as described above.
- Usability: The app must have a high focus on usability to attract a wide audience, that can easily invoke the functions of the product. In order for people to stay committed to their exercise routines through this app, it should be so easily accessible, so that users can find and maintain good partners without much effort.
- Security: The users of this app will store a lot of personal information in the app, to get the best experience. Therefore, the security level of the app needs to be a high priority to protect our customers' personal data, and to line up with international standards for GDPR laws.
- Adaptability: Adaptability is vital when building and expanding the app to get as many users and functions in the app as possible. This means that there is a need for the app to be flexible in terms of providing different services and functionalities based on the type of sport, exercise habits and what's available to users when they exercise etc.
- Scalability: Scalability is important to consider because our user base is basically everyone doing some form of exercise. Therefore, we are anticipating an increasing number of users accessing our software as it gains popularity. There is more value in our product if we have more users because the product is dependent on having many users to connect with other users and create a better user experience.

Table 1: The Seven qualities of Wildy Desirable Software, Mike Gualtieri(2011)

The Seven Qualities Of Extraordinary Software

Quality	What it means
1. User experience	Users' perceptions of an application's usefulness, usability, and desirability based on the sum of all direct and indirect interactions.
2. Availability	An application's readiness to perform its functions when needed
3. Performance	The speed with which an application performs a function that meets business requirements and user expectations
4. Scalability	An application's ability to handle increasing or decreasing volumes of transactions, services, and data
5. Adaptability	The ease with which an application's functionality can be changed or extended
6. Security	Mitigating the risk of attack and ensuring confidentiality, integrity, authentication, authorization, and nonrepudiation
7. Economy	Minimizing the cost to build, operate, and change an app without compromising its business value or any of the other six design qualities

58115 Source: Forrester Research, Inc.

1.2) Software functions

- Finding other people who wish to physically exercise or do mindfulness exercises based on geographical location. Thereby matching two-users based on several parameters.
- Setting up parameters for your exercises such as:
  - Previous experience.
  - Goal for the day.
  - What kind of exercise users are planning to do.
  - Time of day for the workout (calendar included in the platform)
  - Many more can be added.
- Track the user's performance during the activities and show a history of their evolution.

1.3) Monetizing the idea and why this project could be developed in real life

- The software could be expanded by making personal trainers schedule paid workouts in the app. Thereby giving them a platform with access to a variety of gym-goers, while also giving regular members the possibility of connecting with a professional.
- Premium memberships to personal trainers and free membership to regular users. A third subscription could be created for specific gym members.
- Healthy lifestyle related apps.
- Free and premium features

- Include a preference for discovering new activities/hobbies/sports in the application's preferences section.
- Partner with running clubs.
- Free = general database where the user can choose between users that are not necessarily specified to the user's specific needs (it would be more time-consuming to find a match, but not impossible)
- Premium = Put in more details about the user's preferences in a workout and fitness buddy, and the app would only show matches for this specific user's specific preferences (e.g., what gender the user's buddy should have, where they live, when and how they work out, their experience/goals with exercising etc.)

I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 2 of 16
{Project Kick-Off}

Author	Date	Version	Status	Description
AE	08/09	2.1.1	DR	Project Kick-off
ATM	29/09	2.1.1	RW	Reviewed
ATM	14/12	2.1.1	FV	Reviewed

Copenhagen 2022

2) Project Kick-off

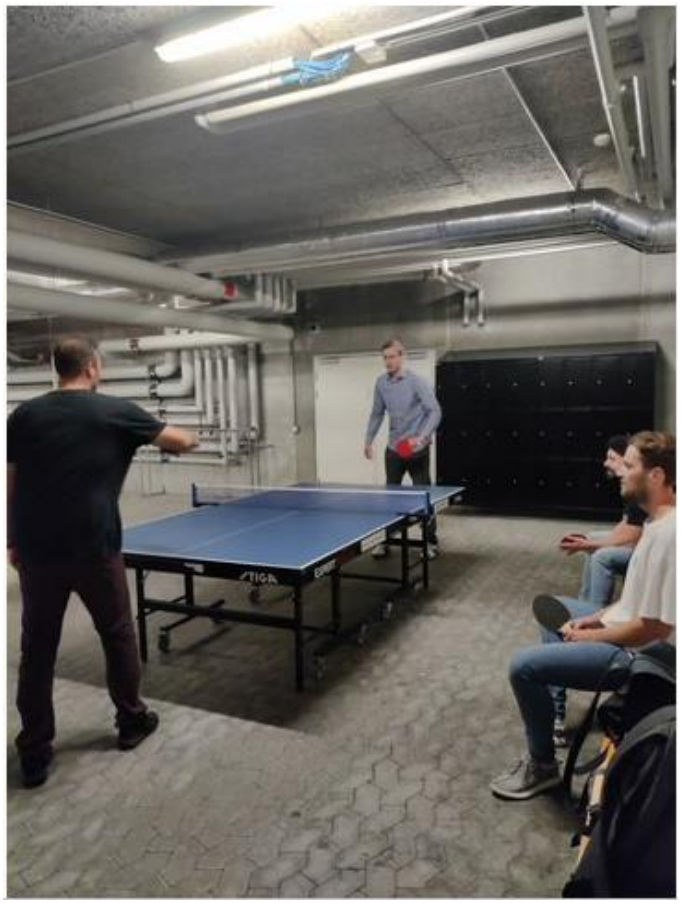
Task: Ask each team member to think of one specific and unique non-university related thing about themselves and share these.

Activity ideas:

- Bowling
- Ice cream at Ismageriet
- Ping pong in the basement of ITU

We went with the idea of Ping pong in the basement of ITU, mostly because it is in the spirit of our project, to bring people together by doing an exercise-based activity.
We played the game "around the table" which requires everyone to participate at the same time (See picture below).
Kicking the project off with several games of ping-pong did indeed bring us closer together. We tested our hypothesis that working out together is motivating, and at the end of the day, we came out as a more homogeneous group than before.

Picture 1:Team members playing ping pong as part of the Project kick-off activity



I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 3 of 16
{Team Protocol Discussion}

Author	Date	Version	Status	Description
ATM	01/09	3.1.1	DR	Team Protocol
AE	03/09	3.1.2	DR	Formatting changes
NSV	29/09	3.1.3	DR	Rewrote from notes to consistent bullet points
ATM	29/09	3.1.3	RW	Reviewed
NSV	10/10	3.2.1	DR	Rewrote text from bullet form to complete text
MHG	10/10	3.2.2	DR	Tone of communication
ATM	13/10	3.2.2	RW	Reviewed
ATM	14/12	3.2.2	FV	Reviewed

Copenhagen 2022

3) Team protocol discussion

We initiated our team protocol through a Q&A discussion inspired by the exercise description during the first week of the course.

Based on this discussion we came up with different rules and guidelines that we will be utilizing throughout this project.

3.1) Previous experiences

From our previous experiences with working in a project, we all agreed that it's first and foremost important to listen to one another and keep a respectful tone at all times. This will ensure that everyone participating in the project feels welcome and comfortable to pitch ideas.

Several of us have also had bad experiences with project members not putting in the work and effort to evenly distribute the work and make a good project, which has transpired strong frustrations. Likewise, it's been difficult to work with project members who's had a "my way or no way" mentality, which is why we agreed that we should always listen to and respect each other's ideas, even if we have a difference in opinion.

Furthermore, we also agreed that it's vital that everyone in the project is interested in doing a good job and to align expectations before project start, in order to have the best possible approach and outcome. To enforce this, we will take notes and do a summary of agreements at every meeting, so that everyone will always have a clear idea of what's been discussed and how to proceed.

3.2) Schedule and meeting structure

Initially we decided to allocate Thursdays from 9-18 to work on our project, and then plan additional meetings if needed.

However, we quickly found out that it was hard to keep focus and do all the work in a single day, and therefore decided to revise our schedule. In order to follow the scrum method more closely, we resolved to meet twice a week on Thursdays from 12:15 to 15:00 and Mondays from 13:00 to 14:00 instead. The reasoning behind this decision was to have more room to discuss while having a more agile project approach by meeting more than once a week.

During every meeting we will start out by checking in with all project members to see if anyone has any questions or things that need to be discussed.

Finally we will assign tasks for the next meeting, where we will also make sure that everyone has a clear understanding of what is expected for each of the delegated tasks.

Our general approach to our meetings and the project in general, is also to make sure that everyone is following all steps of the design process, and to manage everything through a Trello board. Most importantly, we also decided that everyone always need to be available to the group by checking the group chat and providing help if needed, so no one will get stuck on their own.

3.3) Tone of communication

As stated previously in the project protocol, it's first and foremost important that all members are always respectful, mindful and helpful to one another. When feedback is given, this should be given in a constructive way, while listening to and respecting each other's thoughts and ideas.

If any team members remain silent and or are not participating sufficiently in the project, it is also up to the remaining group members to ask for their input and create a safe space where everyone feels comfortable to share their thoughts. Likewise, it's up to the silent group member to speak up and communicate if they have a hard time grasping a concept or feel like they're having a hard time and need help.

3.4) Conflicts

If any conflicts arise during the project, it's important to always respect and listen to all group members' ideas and try to solve the problem through reasoning and compromise.
If this doesn't work, a democratic vote will be held to settle the dispute, and whatever option has the most votes will be chosen without further questioning.

3.5) Impact of the software process model

At first, we discussed using the XP software process model, but later decided to go with the scrum method which will be elaborated further in document 6.

I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 2 4 of 16
{Roles and Responsibilities}

Author	Date	Version	Status	Description
NK	17/09	4.1.1	DR	Role's description and rotation
NK/MHG	22/09	4.1.2	DR	Initial role connection plan
NSV	28/09	4.1.3	DR	Revised and rotated roles
ATM	29/09	4.1.3	RW	Reviewed
NK	10/10	4.2.1	DR	Revised roles according to Scrum method, revised role rotation and role connection chart
NK	14/10	4.2.2	DR	Revised Role Rotation
ATM	14/10	4.2.2	RW	Reviewed
ATM	14/12	4.2.2	FV	Reviewed

Copenhagen 2022

4) Roles and Responsibilities

Before we chose our project development method, we discussed the possible roles and responsibilities based on the traditional project management approach and came up with the following:

4.1) Roles

1. Project manager
2. UX/App user
3. Chairman
4. Mediator
5. Software Architect (UML)

4.2) Role Responsibilities

1. Project manager
    a. Update the schedule
    b. Management of Trello board
    c. Make sure that all project partners are aligned and meeting deadlines etc.
    d. Define frameworks and tools that will be used in the project
2. UX/App user
    a. User feedback from stakeholders
    b. Making sure that the app/platform has a smooth user journey/experience
    c. Design the application's screens
    d. Design the application's features
3. Chairman
    a. Setting up meetings with the supervisor
    b. Manage documents and take notes
    c. Propose the schedule's subjects for the next meeting
4. Mediator
    a. Contact possible stakeholders and bring suggestions and definitions to the technical team
    b. Search for early adopters
    c. Monitor the application's development and orientate the technical team to comply with the stakeholders' expectations
    d. Get feedback from the stakeholders periodically (together with UX/App)

5. Software Architect (UML)
    a. Design the application's classes and its iterations using UML
    b. Link the designed screens with the classes and its methods ("virtual" Front and Backend)

4.3) Role rotation

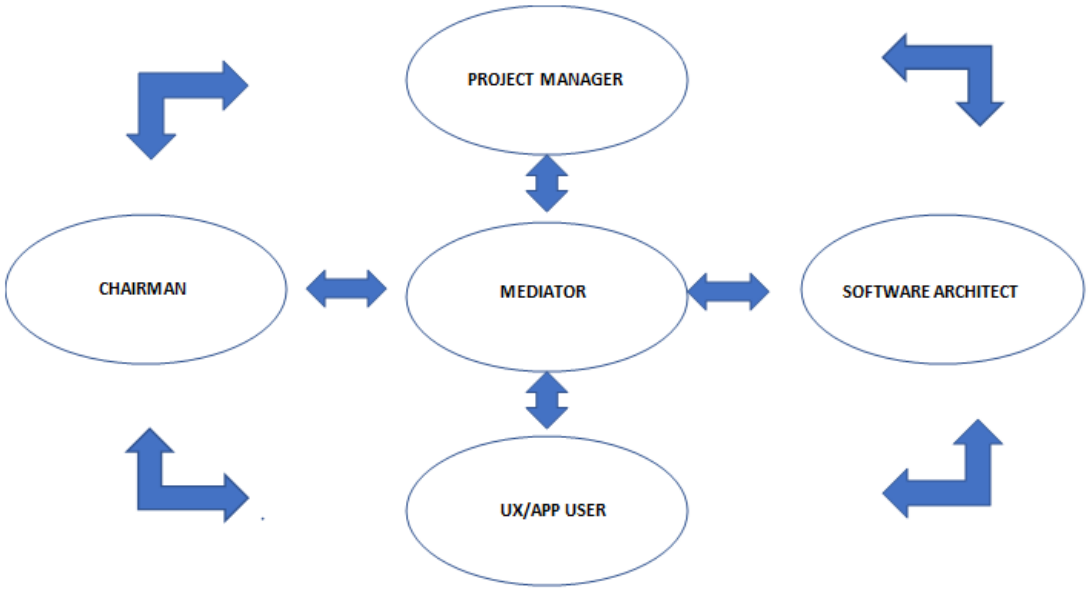
All roles will rotate every other week, for everyone in the group to gain experience within each role.

At the end of every other week there will be a discussion on how the experience with each role was and new roles will be assigned for the following week.

Table 2: Initial Rotation plan

Person/week	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
Andreas	2/5	2/5													
Eduardo	2/5	2/5													
Marcus	4	4													
Neilos	3	3													
Nikoline	1	1													

Figure 1: Initial role connection chart



4.4) Revised Roles and organizational chart

After further discussion and considerations towards our case, the group decided to approach the project using the Scrum Method instead of XP, and the initial roles and responsibilities were revised as follows:

Roles

1. Scrum Master (SM)
2. Chairman (C)
3. Product Owner (PM)
4. Development Team (DT)

Role responsibilities

1. Scrum Master
    a. Update the schedule
    b. Manage the Trello board
    c. Coach the team and make sure that all partners are aligned and meeting deadlines etc.
    d. Define the project framework and tools that will be used in the project
2. Chairman
    a. Setting up meetings with the supervisor
    b. Manage documents and take notes
    c. Propose the schedule's subjects for the next meeting
3. Product Owner
    a. Manage and prioritize the project backlog
    b. Define/accept the user case stories
    c. Set the requirements of the product
    d. Ensure good communication between the client and the project team
4. Development Team
    a. Develop the product features
    b. Test and execute the product features

Detailed explanation of the Roles and roles responsibilities within the Scrum method, can be found on Document 6 {Software Process Model} / Ch. 6.2 {Scrum method Roles and Responsibilities}.

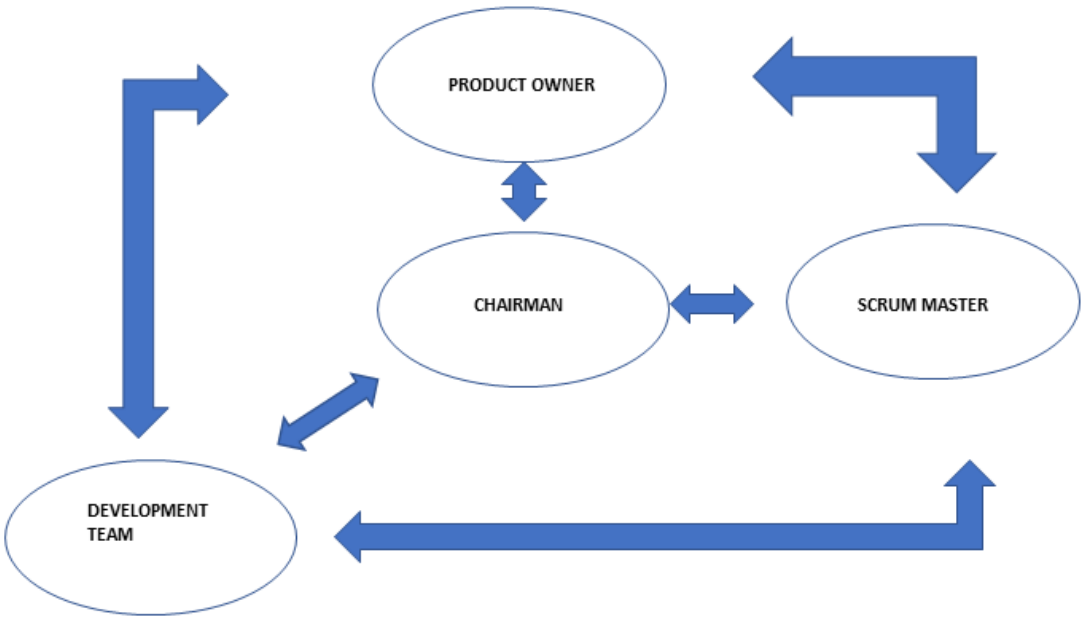
Table 3: Revised Rotation plan according to the new roles (revised 19.09.22)

Person/week	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
Andreas	UX/SA	UX/SA	DT	DT	SM	SM	DT	DT	DT	C/DT	C/DT	C/DT	DT	PO	DT
Eduardo	UX/SA	UX/SA	DT	DT	DT	DT	C/SM	C /SM	PO	DT	DT	SM	DT	DT	DT
Marcus	M	M	DT/C	DT/C	DT	DT	DT	PO	C/DT	DT	DT	SM	PO	DT	DT
Neilos	C	C	SM	SM	DT	PO	PO	DT	DT	SM	SM	DT	C/DT	C/DT	SM
Nikoline	PM	PM	DT	DT	DT/C	C/DT	DT	DT	DT	PO	PO	DT	SM	SM	C/PO
Zach			PO	PO	PO	SV	SV	SV	SV	SV	SV	SV	SV	SV	SV

The new roles as per our initial plan are distributed on a rotation basis. So, every team member can have the chance to perform within the holistic spectrum of the Scrum method. Since none of us have real life experience in working with Scrum, the whole procedure is based on a "learn-by-doing" process and each team member is helping one another once they change their roles.

Initially as per our agreement with our supervisor, the Product Owner role was assigned to the supervisor (as per the course's guidelines), but given the comments on Tollgate 1 assignment, we were advised to assign the P.O. role to one of the team members. So, from week 41 and ongoing the P.O. position will be rotating between the team members.

Figure 2: Revised team organizational chart (revised 19.09.22)



I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 53 of 16
{Risk Analysis}

Author	Date	Version	Status	Description
ATM	15/09	5.1.1	DR	Project and Product Risks
MHG	17/09	5.1.2	DR	Revised risks
NK	26/09	5.1.3	DR	Risk Mitigation and Analysis – Risk evaluation and counter steps
ATM	29/09	5.1.3	RW	Reviewed
NK	26/11	5.2.1	DR	Updating Risk 2, Added Risk 10,11,12
NK	10/12	5.2.2	DR	Added risk 13
ATM	14/12	5.2.2	FV	Reviewed

Copenhagen 2022

5) Risk Analysis

5.1) Project Risks

a. Staff turnover
    i. Quiet Quitting
b. Scheduling meetings
    i. Matching everyone's schedules to fit meeting times
c. Personality types
    i. How people like to work on a project
    ii. How personality types can clash and create certain group dynamics
d. Time management
    i. Predict the time spent with tasks
    ii. Recognize the difficulty level of each task
    iii. Underestimate the time to conclude an activity
    iv. Missing Deadlines

5.2) Product Risks

a. Entry to the market
    i. Get the first users - companies/ professionals / people adhesion
    ii. Community scalability
b. Product competition
    i. Similar products on the market
    ii. Established companies developing a better product because they have more experience/resources
c. Software tool doesn't perform as expected
d. Document inconsistent
e. Not meeting Mandatory Requirements

5.3) Both

a. Requirements change
    i. More/more advanced requirements than first anticipated
b. Size underestimate
    i. Underestimating the size of the system/software

5.4) Risk Mitigation and Analysis

Table 4: Risk Evaluation

Probability							
	6						
	5						
	4			Risk 10	Risk 7	Risk 1, Risk 4	Risk 8, Risk 11
	3	Risk 6		Risk 9			Risk 3
	2					Risk 13	
	1		Risk 5	Risk 2			Risk 12
		1	2	3	4	5	6
Effect							

Table 5: Risk evaluation and counter-steps

Risk ID	Description	Probability	Effect	Level	Strategy/ Action
Risk 1	Product competition	4	5	Moderate/High	Try to be first mover.
Risk 2	Staff turnover	1	3	Low	Role-redistribution
Risk 3	Entry to market	3	6	Moderate/High	Create more exposure
Risk 4	Project size underestimating	4	5	Moderate/High	More collaboration between group members. Time Buffer.
Risk 5	Scheduling Meetings	1	2	Low	Online meetings.
Risk 6	Personality Types	3	1	Low	Find ways for better communication and understand our classmates
Risk 7	Time Management	4	4	Moderate/High	Decrease our sprint times. Ask coaching from the more experienced member of the team
Risk 8	Software tool failure	4	6	High	Restructuring and learning from experience
Risk 9	Requirements Change	3	3	Moderate	Update as per requirements. Restructure and reallocate resources
Risk 10	Quiet quitting	4	3	Moderate	Set up new communication

					channels, Allocation of work
Risk 11	Missing Deadlines	3	6	High	Prioritize the project, help members that are behind
Risk 12	Not meeting mandatory requirements	1	6	Low/Moderate	Seek guidance and advice from teachers, Supervisors, TA's, fellow students
Risk 13	Document inconsistent	2	5	Moderate	Prioritize the editing of the document by using clear language

Table 5: Risk evaluation and counter-steps is analyzing the Risk Evaluation of table 4, providing counter-step examples to reduce the project's exposure to any specific risk.

Risks are divided into Product and Project risks. However, there are risks that combine both aspects. They are prioritized as Hight to Low risk and effectiveness as per Table 4: Risk Evaluation. Risks 1, 3, 4, 7 and 8 are prioritized as High to Moderate risks and need to be re-evaluated regularly. Risks 2, 5, 6 and 9 are prioritized as Low to Moderate risks considering their effectiveness to the overall project's success and that there is more flexibility on these. Risk 10 is a Moderate Level Risk but through the project it was generally noticed that this risk is higher than the Staff Turnover that was demoted to Low Level. Risk 11 is prioritized as High as we noticed that while on the second Tollgate, we were not able to fully deliver our Documents, and have not entered that Risk on the initial analysis. Risk 12 has a high effect, but we believe that we will meet the requirements, and have therefore prioritized it as Low/Moderate. Finally, Risk 13 is prioritized as Moderate and was identified during the Quality Assurance plan development.

I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 64 of 16
{Software Process Model}

Author	Date	Version	Status	Description
NSV/MHG	19/09	6.1.1	DR	XP-programming roles and advantages
NK	20/09	6.1.2	DR	Scrum method, roles and responsibilities
AE	29/09	6.1.3	DR	Advantages of using scrum
ATM	29/09	6.2.1	DR	Reasoning for picking the scrum-method
MHG	29/09	6.2.2	DR	Updated advantages of using scrum
NSV	29/09	6.2.3	DR	Revised text
ATM	29/09	6.2.3	RW	Reviewed
ATM	14/12	6.2.3	FV	Reviewed

Copenhagen 2022

6) Software Process Model

In this section we will provide a description of both the XP and the Scrum method as we initially chose to go with the XP method, but later decided to change to the Scrum method after elaborate comparison and discussions.
Our reasoning behind these choices will be clarified at the end of this document.

6.1) Extreme Programming method (Extreme programming 2013)

The extreme programming method is one of the most radical approaches to agile methodology, and mainly builds on continuously getting feedback from both the customer and the fellow team members on the project. The main points of relying so heavily on a continuous loop of feedback, is to anticipate and catch as many errors and challenges in the project as possible before delivering the product, but also to thrive in mistakes and learn from them in order to deliver the best possible result.

The method uses pair programming and continuous integration, where small and frequent batches are thoroughly tested and verified before implementation. This enables mistakes to be detected and eliminated effectively, while creating a stable system.

In order to make it as easy as possible to integrate new batches into the system and adapt swiftly to the constant changes during the project, it is vital that all components of the project are kept as simple as possible.

Advantages of using extreme programming

The advantages of using extreme programming as an agile project method is that the customer is highly involved in the project throughout the whole process and will thereby ensure a product that lives up to their requirements. Likewise, the system as a whole will be more stable due to error avoidance and continuous integration, while it is easy to make changes and undertake new requirements along the way as a consequence of the simple project design.

Disadvantages of using extreme programming

The disadvantage of using extreme programming is that it requires an extensive amount of participation and involvement from the customer, which might not always be possible. The customer could e.g., lack sufficient knowledge or interest in the project or could simply be short on time and resources, which would impede how the project could thrive on continuous customer feedback.
Furthermore, projects using the extreme programming method is often associated with higher costs and require more time and self-discipline, since the project hasn't been specifically mapped out from the beginning and a lot of changes are made underway.

Table 6: XP Principles (Sommerville 2016)

Principle or practice	Description
Collective ownership	The pairs of developers work on all areas of the system, so that no islands of expertise develop and all the developers take responsibility for all of the code. Anyone can change anything.
Continuous integration	As soon as the work on a task is complete, it is integrated into the whole system. After any such integration, all the unit tests in the system must pass.
Incremental planning	Requirements are recorded on "story cards," and the stories to be included in a release are determined by the time available and their relative priority. The developers break these stories into development "tasks." See Figures 3.5 and 3.6.
On-site customer	A representative of the end-user of the system (the Customer) should be available full time for the use of the XP team. In an extreme programming process, the customer is a member of the development team and is responsible for bringing system requirements to the team for implementation.
Pair programming	Developers work in pairs, checking each other's work and providing the support to always do a good job.
Refactoring	All developers are expected to refactor the code continuously as soon as potential code improvements are found. This keeps the code simple and maintainable.
Simple design	Enough design is carried out to meet the current requirements and no more.
Small releases	The minimal useful set of functionality that provides business value is developed first. Releases of the system are frequent and incrementally add functionality to the first release.
Sustainable pace	Large amounts of overtime are not considered acceptable, as the net effect is often to reduce code quality and medium-term productivity.
Test first development	An automated unit test framework is used to write tests for a new piece of functionality before that functionality itself is implemented.

6.1.1) XP-Roles

The customer:

This role is the businessman of the team, responsible for making all the business decisions, regarding the product. This role decides which features should be included in the product, and what they should accomplish as a whole. This role also sets the acceptance criteria for the product, keeps an eye on the budget, and decides what next to deliver promised features on time and within the budget. (Wiki.c2.com, 2006)

The developer:

This role is responsible for realizing the needs identified by the customer. Everyone on the team, except the customer and other secondary roles, is considered a developer. Because different products typically require different skills, and most members of the team have the "Developer" role. This role definition is kept short.

The Tracker (This role is optional):

This role keeps track of relevant metrics of progress for the team and identifies the areas where the product needs improvement. Typical metrics that the development teams track includes velocity, reasons for changes in velocity, overtime, and test results of the products features. Not all XP-development teams use a tracker. The tracker is a developer who also takes on the role of being the teams tracker.

The Coach:

If the development team isn't that familiar with the XP-method, it might be a good idea to hire an external consultant to function as a coach or use someone from another development team in the organization, who is experienced with the XP-method. The coach functions as a mentor and helps other team members with the XP-practices and maintains discipline within the team.

Manager (This role is optional):

This role oversees meetings, makes sure the team sticks to the meeting's agenda, keeps track of the meetings results, and shares notes with the tracker. This role also functions as the caretaker of the team, brings snacks and food to everyone. This role is also responsible for personnel actions.

Tester (This role is optional):

This role is in charge of functional testing of the product. This role graphs results and informs the other team members if test results decline. This role alerts everyone when the team is in big trouble, for whatever the reason.

6.2) Scrum method

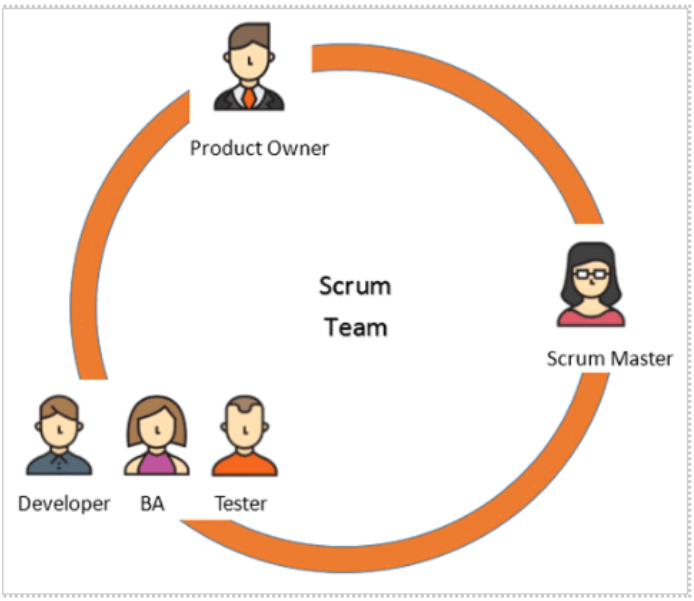
Table 7: Scrum Principles (Sommerville2016)

Scrum term	Definition
Development team	A self-organizing group of software developers, which should be no more than seven people. They are responsible for developing the software and other essential project documents.
Potentially shippable product increment	The software increment that is delivered from a sprint. The idea is that this should be "potentially shippable," which means that it is in a finished state and no further work, such as testing, is needed to incorporate it into the final product. In practice, this is not always achievable.
Product backlog	This is a list of "to do" items that the Scrum team must tackle. They may be feature definitions for the software, software requirements, user stories, or descriptions of supplementary tasks that are needed, such as architecture definition or user documentation.
Product owner	An individual (or possibly a small group) whose job is to identify product features or requirements, prioritize these for development, and continuously review the product backlog to ensure that the project continues to meet critical business needs. The Product Owner can be a customer but might also be a product manager in a software company or other stakeholder representative.
Scrum	A daily meeting of the Scrum team that reviews progress and prioritizes work to be done that day. Ideally, this should be a short face-to-face meeting that includes the whole team.
ScrumMaster	The ScrumMaster is responsible for ensuring that the Scrum process is followed and guides the team in the effective use of Scrum. He or she is responsible for interfacing with the rest of the company and for ensuring that the Scrum team is not diverted by outside interference. The Scrum developers are adamant that the ScrumMaster should not be thought of as a project manager. Others, however, may not always find it easy to see the difference.
Sprint	A development iteration. Sprints are usually 2 to 4 weeks long.
Velocity	An estimate of how much product backlog effort a team can cover in a single sprint. Understanding a team's velocity helps them estimate what can be covered in a sprint and provides a basis for measuring improving performance.

6.2.1) SCRUM method Roles & responsibilities

According to literature, the recommended Scrum team size is from 5 to 11 members, including the Scrum Master and the Product Owner (Coursera, 2022)

Figure 3: Scrum Team Organizational Chart (SoftwareTestingHelp, 2022)



The accountabilities of each role could be analysed as follows:

6.2.1.1) Product Owner

The Product Owner is the voice of the customer/stakeholders and hence is responsible for bridging the gap between the development team and stakeholders. The product owner manages the gap in such a way which would maximize the value of the product being built.

The Product Owner is set to be involved throughout the Sprint Activities and the development efforts and plays a very crucial role in the success of a product. The main responsibility is to maximize the value of the product by challenging the team, meeting stakeholders' and customers' demands, but also knowing when to say no.

To achieve these aims, product owners perform the following activities in Scrum:

- Set product goals
- Ensure that the product backlog is stocked with clearly expressed items that help reach the product goal
- Order the backlog
- Regularly update and refine the backlog together with the developers
- Collect feedback from users, customers, and stakeholders on the product increments
- Agree on sprint goals with the developers.

In addition to the activities above, product owners also carry out product discovery and strategy work. This includes creating and updating a product strategy and roadmap and measuring how much value the product creates using Key Performance Indicators (KPIs).

6.2.1.2) Scrum Master (resources.scrumalliance.org, 2022)

Even though the Scrum Master is not the Project Manager, the Scrum Master is the person who is responsible for facilitating/coaching the Development Team and the Product Owner to work on the day-to-day development activities (Hoffmann J, 2022).

The Scrum Master is the person to contact when there is an issue, when a team-member is facing difficulties in understanding the processes that need to be followed or even if coaching is needed. Both training and motivation of their team members rely on the responsibilities of the Scrum Master, while ensuring that Scrum Values and Principles are always fully understood.

To achieve these aims, Scrum Masters perform the following activities:

- Coach: Facilitate meetings, conversations, and improvements.
  i. Ensure that the goals, scope, and product domain are understood by everyone on the scrum team as well as possible.
  ii. Find techniques for effective product backlog management.
  iii. Help the scrum team understand product backlog items.
  iv. Ensure the product owner knows how to arrange the product backlog to maximize value.
  v. Facilitate scrum events as requested or needed.

- Protector: Run interference so the team can remain focused.
  i. Remove impediments to the team's progress.
  ii. Coach the team in organizational environments in which scrum is not yet fully adopted and understood.

- Leader: Lead without authority and put the team first.
  i. Coach the developers in self-organization and cross-functionality.
  ii. Leading and coaching the organization in its scrum adoption.

- Advocate: Reinforce agile principles throughout the organization.
  i. Understand and practice agility.
  ii. Plan scrum implementations within the organization.
  iii. Help employees and stakeholders learn more about scrum and enact it.
  iv. Work with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization.

6.2.1.3) The Development Team

The development Team is formed from professionals to do the hands-on work, such as Developers, Designers, Data Analysts and Testers. Ideally the Development Team consists of 3 to 9 members, but in the case of a small team, the Product Owner and The Scrum Master are also developing along with the other developers (Coursera, 2022).

Not all the Development team members always share the same responsibilities. According to their specialization the members might take separate roles all of which aim to achieve the end goals of the Scrum Team. But they do share common characteristics such as that they are:

- A Scrum Team Member
- Cross-functional
- Self-efficient
- Self-organized
- Skilled
- Committed and available
- Accountable

6.3) Advantages of using Scrum

Scrum has many advantages in the processes of developing software. Among these are: Quicker release of usable products to users and customers, higher quality, higher productivity of team members, lower overall costs, more adaptability regarding requirement changes and higher user satisfaction for the finished product (agilest.org, 2022).

Working in a scrum environment offers great flexibility because of the principles of simultaneous development, which means that both coding and testing is done throughout the entire process.

Scrum offers high adaptability because the scope of the project can vary. This is explicitly true due to the concept of design sprints, wherein project members work on smaller parts of the project in approximately 2-4 weeklong time periods.

Working with scrum typically means that tasks are prioritized by order of importance. For every design sprint, the importance of tasks to be done are evaluated on a scale, where the most important tasks take precedence in the corresponding design sprint.

The Scrum method uses the principles of pair programming, in which programmers pair up to produce better and more readable code, which in turn makes the program more adaptable to future design specification changes. This also means that employee satisfaction rises, as team members are seldom working on something entirely alone.

By factoring together all the benefits of working with scrum, we can increase the chances of delivering a satisfactory product at the end of the design process. This benefits everyone, from stakeholders, employees and users of the product.

6.3.1) The Scrum Sprint Workflow

For each sprint in the scrum method, you plan just enough to get started on a feature in the product (Sommerville, 2016). Then you build the feature that was planned, and then you test and review that feature. The idea is at the end of each sprint you should end up with a potentially shippable feature. The product owner can also decide if the feature still isn't quite ready yet after the sprint.

The scrum method also consists of a daily short meeting, with all roles involved to ensure that you are on track. Meaning the team members inform each other what they have completed since yesterday's meeting, what they are working on now and if they need assistance in some areas. The daily meeting is very short, and some companies prefer to have a standing meeting, to keep it that way.

The planning around the sprint consists of the product backlog, sprint planning, sprint backlog, sprint review and retrospective (Ibid.). The product backlog is where the product owner brings together a list of the best ideas for features that could be a part of the product. The product owner then picks out a feature to present to the team. Sprint planning is where the entire team picks out the top user stories to determine what should be a part of the next sprint. The outcome of the sprint planning is the sprint backlog, and the sprint backlog is the short list of user stories that have been picked out for the next sprint. After the sprint itself comes the sprint review. Where the team presents their output to the product owner. The last part is the retrospective, where the team members discuss what they can improve on for the next sprint.

6.3.2) The reasoning behind Scrum

We initially chose the XP-method because we were working on an app with a lot of different requirements, that would most likely change overtime. To handle the different expectations of the product and mitigating the risk of the project in general, we chose the XP-method. Since the XP-method was designed to handle development projects with changing requirements and managing risk during the project (extremeprogramming.org 2022). XP-method also made sense for us because we are going to develop an app that combines certain features that haven't really been seen before. Which automatically increases the risk of the project.

After we had some time to dwell on our choice of a software process model, we realized that the XP-method is so closely related to developing, and since we won't

produce any actual code on this project, the scrum method would make more sense, in terms of making our project as realistic as possible. Even though in terms of group size, a 5-member group would be more normal for a XP-development team (agilealliance.org 2022). But we chose to go with the scrum-method because it contains more project management aspects that we would be able to use throughout our project to make it as realistic as possible (Sommerville 2016). The project management parts of scrum we could use in our project would be the product backlog, sprint planning, sprint backlog, sprint review and the retrospective part of the scrum method.

Another reason we chose Scrum as our software process model, is because the Scrum-method is a good fit regarding the software qualities we want our product to possess. We chose the software qualities user experience, usability, security, adaptability, and scalability. For the software qualities user experience and usability, we repeatedly need user feedback from our users to improve the product, and the agile scrum method allows us to do updates for our features, or even implement new ones, rather quickly, because the scrum sprint cycle is relatively short, compared to a plan-driven method. We are making an app with a lot of different features, and we are expecting to add new features regularly in the future. Therefore, adaptability is a key software quality for us, and here we can also take advantage of the flexibility of the scrum method, to develop shippable features or updates at a fast rate. Same goes for scalability and security. For our software qualities in general, the scrum-method gives us the opportunity to react fast to our customer's needs, by having short repetitive work cycles that gives us the opportunity to respond to user feedback quicker, because we can consider new user stories, every 2 to 4 weeks. We are thereby taking advantage of the short amount of time you spent planning in Scrum.

The biggest resemblance between our project and scrum is the way we use sprints. This works well for us because we are not that experienced and need to break things down a bit more than you normally would, but specifically it also matches the way we get a small new project assignment each week due to our course structure. Scrum also fits our case/situation very well since it's also very dependent on the collective work and experience of the project members. Since neither of us has a lot of experience with this, it also serves us well to work this way and rely more on each other. With scrum being an agile approach, it's important for us to be adaptable because we don't have a lot of experience, but also because it's kind of a "pretend product", which means that it's harder for us to test things out in a realistic way. Our chosen collaboration tool, the Trello table, is being used as our product/project backlog which is also one of the key parts of the scrum method. We get the bigger tasks/project areas we need to finish every week, and then in our own backlog we break it down into smaller pieces so it's easier to manage and delegate.

I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 75 of 16
{Project Plan and Estimation}

Author	Date	Version	Status	Description
ET	22/09	7.1.1	DR	Planning structure
ET	22/09	7.1.2	DR	Task Tables and Gantt Chart
ET	26/09	7.1.3	DR	Planning context and description
NSV	29/09	7.1.4	DR	Revising text
ATM	29/09	7.1.4	RW	Reviewed
ET	09/10	7.2.1	DR	Changes after tollgate 1 feedback
ET	11/10	7.2.2	DR	Revision and small changes in the planning text
ATM	14/10	7.2.2	RW	Reviewed
ET	11/12	7.2.3	DR	Update Planning Content
ATM	14/12	7.2.3	FV	Reviewed

Copenhagen 2022

7) Project Plan and Estimation

The project scheduling plan approach was divided into the following major parts: task identification, task efforts estimation, and documentation. In the project scheduling phase described in section 7.1, our group identified all the main tasks, their derived subtasks, and the roles involved in each of them.

Furthermore, we discussed the connections between different tasks to find and expose all the dependencies between them as presented in section 7.2. During both previous activities, the planning process was documented in two different forms: a Trello board and a Gantt chart. An explanation of how it was documented and created could be found in section 7.3.

7.1) Tasks Identification and Roles Identifiers

As a first step in planning our project, all the members worked together to list the main tasks of the Software Engineering course. In addition, we listed some of the task expectations for the app development. Based on that, to simplify complex and lengthy tasks in a way to fit in the sprints of our chosen software process model SCRUM, we broke it into several smaller tasks to be assigned to the team members.

Focusing on the sprints, our approach was to think in an explicit way to make the roles for each task clear when assigned. Therefore, as shown in picture 2, to link the roles and tasks, role tags were created according to our Roles and Responsibilities analysis on Document 4. In this way, we will be able to plan each sprint and monitors the progress in a visual form as well.

Picture 2: Roles tags to be used in the sprints.

7.2) Tasks time estimation

After identifying the tasks and splitting them up into small tasks, we estimated the efforts for each one using the planning poker process but slightly modified. The planning poker is a gamified technique for task time estimation which consists of a consensus through the project team. For that, it is normally used cards face-down and after a brief task presentation by the moderator (chairman), which doesn't participate in the estimative, everybody turns their cards at the same time and presents the number that in their own opinion represents the complexity and time efforts for that task.

Instead of using cards our group wrote the Fibonacci numbers on the computer at the same time and presented them after an introductory presentation of each task and previous discussion. The Fibonacci numbers were chosen to represent the time effort once it had well-known steps between them [1, 2, 3, 5, 8, 13, 21, …] – the sum of the two previous numbers. Therefore, using that sequence it is possible to identify and separate the tasks' complexities once it has big steps, making it clear that the next number is much more complex than the previous one.

In our case, the moderator participated in the time estimation once we get to a consensus that all the group members needed to vote and estimate due to the lack of experience in that activity. During the planning poker, the activity kept going until the numbers could not get any closer. When discrepant numbers appeared, the members with the highest and smallest number talked about their different approaches to the task. Then, we repeated the process for all the identified tasks.

Table 8 presents all the main tasks tracked by our team for the first toll gate submission and some extra ones regarding the project development, explained in section 7.1. This table contains the schedule presentation data for the macro tasks with 4 main pieces of information: identified task, time effort (person-day), duration, and dependencies. For each task, it was estimated the time effort and duration based on the table presented in appendix A which, instead of having the duration, has the Fibonacci number from the planning poker.

Thus, the time effort was estimated for each one of the smaller tasks (Appendix A) based on the planning poker that together compose the macro and final task presented in table 8. In other words, the efforts of the smaller tasks were summed to have the final effort presented in table 8. For example, table 9 contains the subtasks extracted from appendix A. All the tasks presented contain a reference of 8.7 which appoints to Project Plan and Estimation in table 8 and it's possible to notice that the total sum of the efforts results in the amount of person-day indicated in table 8. In this

way, we could also estimate the duration but with an additional step which was the course's deadlines.

Table 8: Identified main tasks for Tollgate 1 and Project Development phases

Task		Person-day	Duration	Dependency
1	Tollgate 1	359	35	1 – 14, 18
2	Explore your case	40	7	14
3	Project Kick-Off	20	7	-
4	Project Team Protocol	20	7	-
5	Process Model Initial Discussion	10	7	2
6	Roles and Responsibilities	31	21	4, 5
7	Project Plan and Estimation	41	21	6
8	Risk Analysis	33	21	2, 5, 6
9	Configuration Management	37	14	-
10	Research XP vs Scrum	24	7	-
11	Software Process Model	53	14	2, 10
12	Combine Documentation	6	7	1 – 9
13	Roles Definition - Process Model	20	7	6, 11
14	User case stories	15	7	-
15	Project Requirements Document	70	21	-
16	Use Case Discuss and Document	36	21	2
17	Rich Pictures	24	14	16
18	Class Diagram over Problem Domain	60	28	2,17
19	Use Case Diagram and Scenarios	51	42	19
20	Mockup Design	117	42	14 – 19
21	Mockup Workshop Preparation	20	21	20
22	Testing Cases and Structure	35	21	15
23	Quality Plan	20	21	22
24	Tollgate 2	378	42	15-21
25	Review for Resubmission TG1	60	14	1
26	Review for Resubmission TG2	60	14	24
27	Review the Project Document	25	21	1-23

Table 9: Subtasks for Tollgate 1 and Project Development phases - example from appendix A

Subtask	Person-day	Fibonacci nº	Task
Schedule - Gantt Chart – Trello	16	5	8.7
Initial planning	9	3	8.7
Task estimations - Planning Poker	15	5	8.7
Final meeting and agreement	5	2	8.7
Work breakdown structure	8	3	8.7
Create project schedule	6	2	8.7
Assign estimations/due dates	2	1	8.7

The person-day represents the total time multiplied by the number of people that will be enrolled in the task, which also was discussed by our group in the planning poker. We planned to use a focus factor of 70% to estimate all the tasks' velocity during the

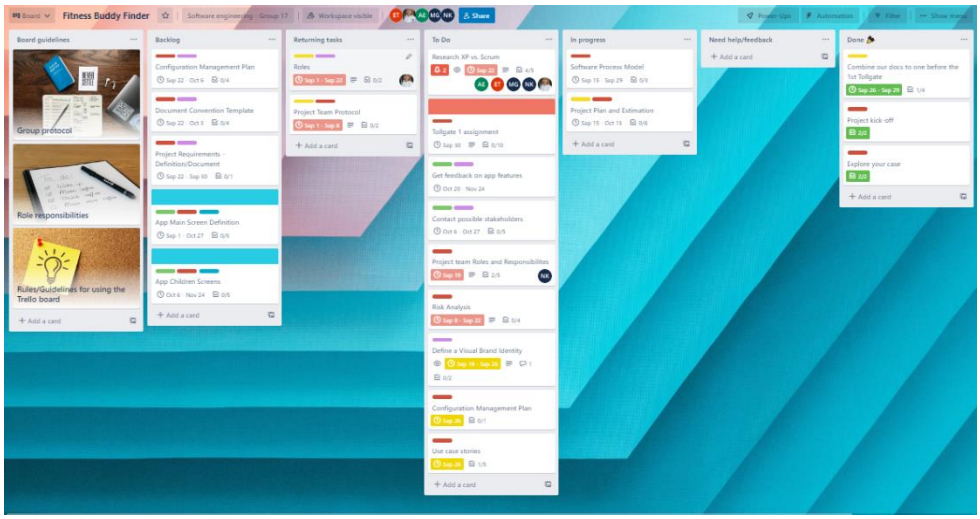
sprints. The focus factor will multiply the time effort to get a real-time that will be spent in the sprint. This value that will be used as a focus factor comes from the class recommendation due to our lack of experience in this subject. The last information in table 8 is the dependencies between the tasks. It also was identified during the planning meetings and represents how the tasks are connected and their prerequisites as well. Once this kind of table contains too much information and it is hard to check and track the progress, we used tools to implement a visual approach which was created based on the info presented in tables 8 and Appendix A. It is explained in the next section (7.3).

7.3) Planning structure and documentation

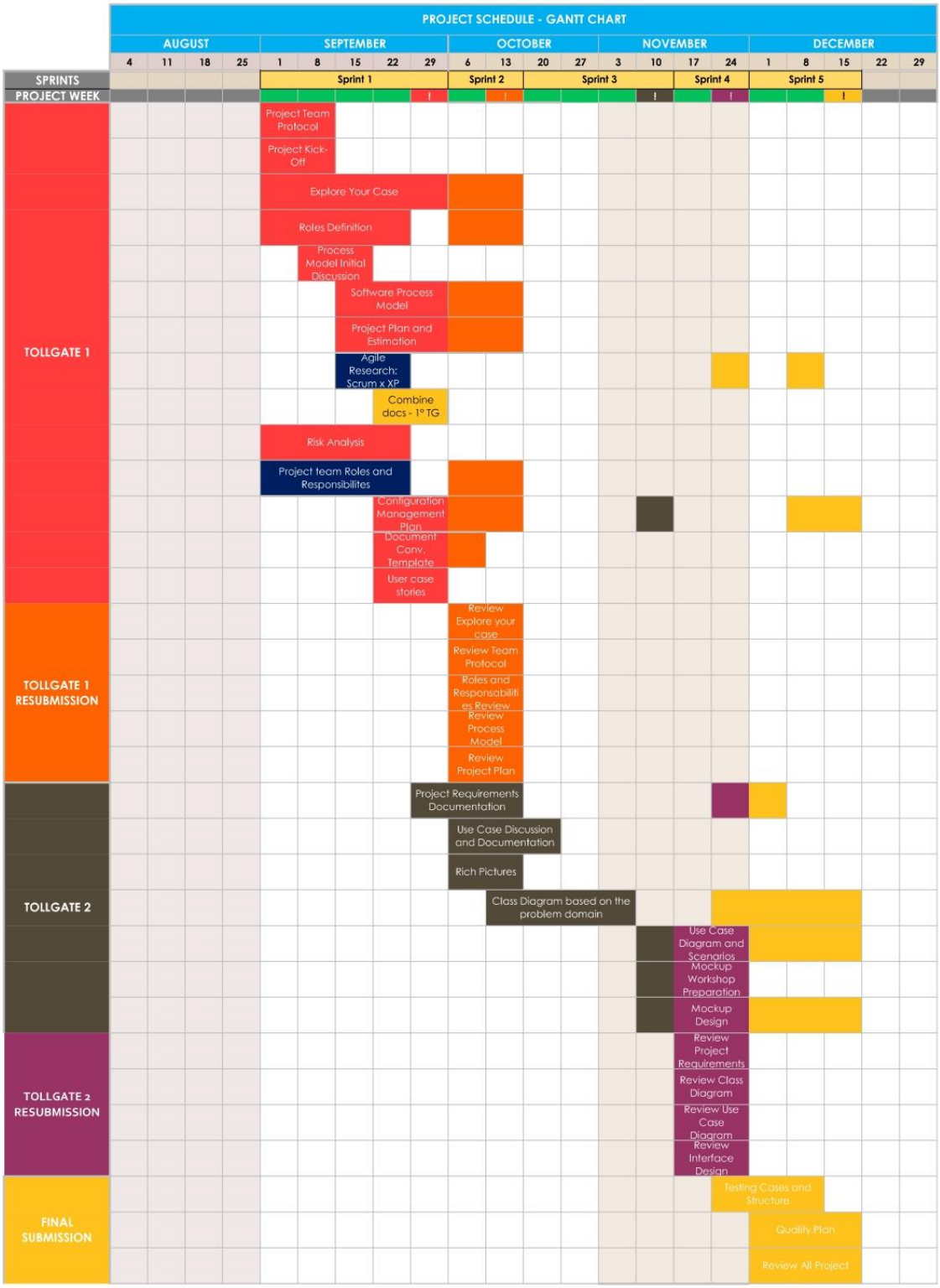
The tool chosen to be used as a task control plan and verification board is Trello (picture 3). Yet, in our tasks control tool, we fixed the first row of important information for the group project with clear information about the role's responsibilities, group protocol, and some guidelines to use the board. All of this is to make it easier for the rotation of roles during the project and to avoid misunderstanding.

As documentation of our project scheduling plan, we created a schedule presentation highlighting the duration, efforts, and dependencies for each task (table 8). The content of this table and their subtasks presented in appendix A were moved to the Trello dashboard. Furthermore, we created and Gantt chart (picture 4) to get an overview of the tasks (as bars throughout the weeks) to be aware of the deadlines, and help the team properly allocate working hours during the sprints in the project development. Also, we already modelled our board tasks in a format to apply the SCRUM method. The current rows are backlog, returning tasks, to do, in progress, need help/feedback, and done.

Picture 3: Trello Board



Picture 4: Project Gantt Chart



I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 86 of 16
{Project Configuration Plan}

Author	Date	Version	Status	Description
NK	24/09	8.1.1	DR	Project Configuration Plan
ET	26/09	8.1.2	DR	Update Project Configuration Plan
NK	27/09	8.1.3	DR	Document Control – Configuration Items – Front Page – Merging of Documents
AE/MHG	29/09	8.1.4	DR	Revision of the configuration plan
AE/MHG	29/09	8.1.5	DR	Revising configuration items
AE	29/09	8.1.6		Identifying configuration items
ATM	29/09	8.1.7	RW	Reviewed
NK	10/10	8.2.1	DR	Revise Document Control
ATM	14/10	8.2.1	RW	Reviewed
NK	21/10	8.2.2	DR	Revise Configuration Plan
NK	11/11	8.3.1	DR	Document control and version definition
NK	10/12	8.3.2	DR	Update Configuration Plan
ATM	14/12	8.3.2	FV	Reviewed

Copenhagen 2022

8) Project Configuration Plan

8.1) Naming convention:

The documents' names will be structured based on several metadata approaching the working status and its content. Furthermore, to create a standardized file name to enable the group's members to know all the important information needed before opening it and to track progress. All the tags will be explained in the following section. Every tag of the document will be separated by underscore.

8.2) Definitions and naming:

1. **Versions & Iterations:**

Format**: Dn.Vn.In,** where:
Dn is the number of the document
Vn is the number of the current version
In is the number of the latest change/update of the document

The Version and Iteration section can be found in the Configuration Management Table at the front page of every document as described below. Changes or updates that have been made are mentioned in the Description of the same table.

Each document's version starts from No 1 is updated at the beginning of a new sprint – see Document 7 – Project plan and estimation for Sprint time definition. The version change indicates that the document is approved.

Example, Version 9.1.2 means:

Document ID: 9

Version 1: the version we are currently working, without any major retro change

Latest update: It's the second update/addition done in the current Version

2. **Author:**

Table 10: Names and Initials

Names	Initials / Tags
Neilos Kotsiopoulos	**NK**
Andreas Elsberg	**AE**
Eduardo Trindade	**ET**
Marcus Holst Garner	**MHG**
Nikoline Stensbæk Vejby	**NSV**
All Team Members	**ATM**

3. **Storage Space:**
   Microsoft Teams SharePoint – Tag: MTSP
   Backup: Dropbox – Tag: DP

4. **Working status:**
   Draft – Tag: DR
   Reviewed – Tag: RW
   Final version – Tag: FV

5. **Figure naming:**
   Pictures:
   Representing Pictures and Snapshots added to the merged Document.
   Each picture is numbered by the ascending order that is appeared in the document followed by a small description of what they depict. A table of contents of Pictures can be found at the beginning of the merged Document for ease access.

   Tables:
   Representing Tables added to the merged Document.
   Each table is numbered by the ascending order that is appeared in the document followed by a small description of what they depict. A table of contents of Tables can be found at the beginning of the merged Document for ease access.

   Figures:
   Representing Figures added to the merged Document.
   Each table is numbered by the ascending order that is appeared in the document followed by a small description of what they depict. A table of contents of Figures can be found at the beginning of the merged Document for ease access.

8.3) Document Control

Every new task is initially done in a separate document, that the responsible team member creates. The team member then proposes the first draft of the document to the group, and if changes are required, then a new version is created, and the process is repeated until the document has been approved and is ready to be added to the merged document. Once the first Draft is merged to the combined document, then changes can be done to the combined document, by following the version control described above, in order to keep track of them.

The merged document consists of all the documents that have been approved by ATM.

All documents are represented in their front Page with the following table, to show the key information about the different iterations of the document, and how many versions it has been through. Each document starts with its first Version, i.e.

Document 1 first version would be 1.1.1. Then, each team member that updates the documents must update the version by increasing the number by 1 (thus iteration is not used). The specific point that the team member has made an alteration is specified in the Description section.

All our Documents are saved in our MTSP.

Table 11: Initial Version - Configuration Management Table

Author		Date	Post	Revision	

Table 12: Current/Final Version – Configuration Management Plan:

Author	Date	Version	Status	Description

8.4) Identifying configuration items

Configuration items, as defined by Lars Bendix, can be described as very important artefacts. Once an artefact has been identified as a configuration item, we must make sure that it is properly stored and backed up, so that it never disappears, and that if changes are made to it, you can always go back to an earlier iteration. Configuration items are all parts of the product, and we cannot succeed without them (Bendix, 2019).

Normally you could identify things such as source code for a program as configuration items. Since we are not doing any actual code in this project, we have identified our configuration items as all finalized documents that are to be evaluated at each toll-gate. And ultimately every document that goes into our final submission.

We are also working with documents that are not considered configuration items, such as meeting minutes, mails, drafts and agendas for the supervisor meetings.

Handling of Diagrams, Tables and Interfaces that designed by using a designated platform has been achieved by sharing the link of each one in the specified (according to the task) Card on the Group's Trello Project Management Home page:

https://trello.com/fitness-buddy-finder

A list with the Platforms used and their link to the group's homepage can be found below:

Table 13: Platform table

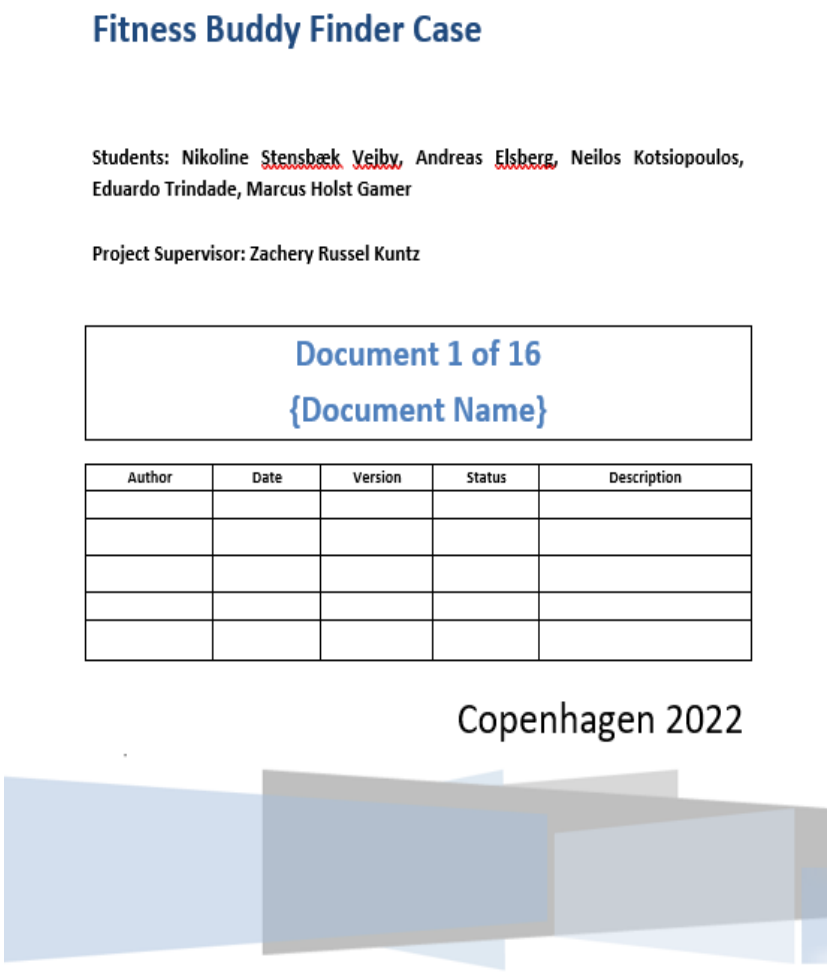
Platform Name:	Group's Homepage link:	Description:
Trello	https://trello.com/fitness-buddy-finder	Task Manager Workspace
Miro	https://miro.com/app/dashboard/	Board and Table Design
Lucid chart	https://lucid.app/documents/shared	UML Diagrams
Figma	https://www.figma.com/file/	Interface Design

8.5) Front Page

The Front Page of each document is depicted at the beginning of every Document and is as like the image to the right.

It includes all the Configuration items, the essential information about the project and our names.

It also includes the Version Control of each document, which is the method we used from the beginning to keep track of our documents individually and/or combined, the Status of the Document and The Description of the latest change.



Picture 5: Front Page Template

8.6) Documents merged into Project Charter

Table 14: Merged Documents

Document	Document Name	Responsible
1	Explore your Case	ATM
2	Project Kick-off	AE
3	Team Protocol discussion	AE, NSV, MHG
4	Roles and Responsibilities	NK, MHG, NSV
5	Risk Analysis	ATM
6	Software process model	ATM, NSV, MHG, AE, NK
7	Project Plan and Estimation	ET, NSV
8	Project Configuration Plan	ATM, NK, AE, MHG, ET
9	Engineering Requirements	ATM, NK, MHG
10	Explore the Use Context	NK, ET, AE
11	Rich Pictures	MHG, NSV, ET, NK
12	Analysis Class Diagram over Problem Domain	ATM, NSV, MHG
13	Use Case diagram and Scenarios	NSV, ET, NK, MHG
14	Interface Design using Mockups	ATM
15	Quality Assurance Plan	ATM
16	Project Reflection	

8.7) Documents merged into initial Analysis

Document	Document Name	Responsible
Abstract	DRAFT -- Fitness Buddy Finder Case – Abstract.docx	NK, NSV
1	Fitness Buddy finder Case – Abstract.docx	AE, ATM
2	Group Protocol SE17.docx	AE, ATM
3	Project kickoff.docx	AE, ATM
4	Roles and Responsibilities.docx	ATM, NK, NSV
5	Risk Analysis.docx	ATM
6	Software Process Model.docx	MHG, NSV, AE, NK
7	Project Plan and Estimation.docx	ET
7	Project Schedule V1.xlx	ET
7	Initial Gantt Chart V1.pdf	ET
8	Document Convention and Configuration Plan Template V1.docx	NK, AE
9	Observation Case Analysis V1.docx	NK
10	Use Case Diagram V1.png	NK
11	Use Case Diagram V2.png	NK
12	Revised Use Case Diagram V1.png	NK
13	Use Case Diagram V3.png	NK
14	Workshop Notes V1.jpg	NK
15	Workshop Notes V2.jpg	NK
16	Communication Tools Testing V1	NK
17	V1 of review and inspection.docx	MHG
18	V1 of reflection.docx	MHG

Table 15: Documents merged into initial analysis

As we have created the combined document on an early stage in this project, the documents to be merged into initial analysis were reduced as we were moving forward on the project and additions and changes were made directly to the combined document, following the agreed document control method.

In the end of each one of the five sprints in this project, each Completed Document along with the merged one were saved in a different file and named according to the submission requirements.

All the Documents can be found in the group's MTSP.

8.8) Configuration Control (Change Management)

For any change request to our product, The Fitness Buddy App. Every single change request (CR) must go through an evaluation process before it is finally determined whether the change request will be implemented or not (Bendix 2019).

Each CR will be posted in a special section on our Trello Board, which in our case will serve as our Change Control Board (CCB). Then the group member who posted the CR will inform the other group members and set up a meeting or address the change to the next forthcoming meeting. Based on all information available the group will discuss whether it will be a good idea or not to implement the CR. In the end the product owner has the final say as to whether the CR will be included in the product or not.

To align our change management with our chosen process model, scrum, we would introduce the CR in the weekly scrum meeting and decide if the CR is urgent in order to call for another meeting within the day or if it can wait until the next Sprint review, which is where CR issues are discussed. In the latter case, all CR proposals are analyzed as to whether they will be included in the next Sprint period or not. Task planning and management is updated in accordance with new CRs.

I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 9 of 16
{Engineering Requirements}

Author	Date	Version	Status	Description
NSV	13/10	9.1.1	DR	Writing initial text description and setting up tables with requirements
ATM	13/10	9.1.2	DR	Picking most important software requirements
NK	24/10	9.1.3	DR	Updating tables with user stories
MHG	24/11	9.1.4	DR	Refactoring user stories
ATM	14/10	9.1.4	RW	Reviewed
MHG	2/12	9.2.1	DR	Updating User Stories according to Mockups
NSV	15/14	9.2.2	RR	Revised text
ATM	14/12	9.2.3	FV	Reviewed

Copenhagen 2022

9) Engineering Requirements

We have identified several functional and non-functional requirements and mapped out the most important ones in the following tables. Each requirement has been given an ID and is described through a user story.

Since we are developing an app with a lot of different aspects to it, we are automatically also facing demanding engineering requirements. The "Fitness Buddy Finder" app is supposed to bring people together for several healthy activities, and contain a social, fun, rewarding and tracker component to it. For that reason, the entire team got together to discuss what would be most essential for the app to provide the best possible experience for future users and customers. We are aware that a lot of additional requirements could be useful for this application, but we have tried to narrow it down.

We divided our engineering requirements into functional and non-functional requirements. The non-functional requirements, based on if the requirement was describing the app's overall operational capabilities and performance, and functional requirements if the requirement describes a specific function or interaction.

9.1) Functional Requirements

Table 16: Functional Requirements

ID	Name	User story
1.1	Match making	I want to be able to find someone to work out with based on the given criteria, but I want to be able to decide on my own whether I want to become buddies with someone once a match has been provided. And I also would like to know about the most popular activities to make it easier to find a fitness buddy.
1.2	Access to user information	I want to be able to view basic and unpersonal user information on all users in the system, so I can decide whether they would be a good buddy or not, e.g., when a match is provided. After the match I want to know information of the person that I will exercise with in more specific details, especially if only the two of us will participate in an activity.
1.3	Calendar Management	As I am using my mobile Calendar app to organize my schedule, I want to automatically have my workout timeslots covered once I confirm my participation in an exercise activity. It would be nice if I could see with who I am working out with in my Calendar, as well. As I am planning a lot of group activities, I want to have the option to notify the fitness buddies that want to participate and register them automatically.

1.4	Measure stress level	As I am facing high levels of stress during the day, I want to have metrics of my stress level in my fitness app.
1.5	Progress tracker	I want to keep track of my progress both in terms of physical exercises but also my stress levels and in all the important health information.
1.6	Gamification	I am very competitive, and rewarding is a strong motivational factor for me. I want to get badges and rewards, once I achieve my goal, in order to keep up.
1.7	Recommendations	As I am not exercising regularly, I want to get recommendations and new ideas from my fitness app. Ideally, I would like to get instructions and videos of how to execute specific exercises correctly and healthy living tips.
1.8	Communication tool	I want to be able to text my fitness buddies about the workout session, the exercises, re-scheduling, or other required information.
1.9	Too much information on the Buddy Criteria	I want a clear and easy way to set the criteria for my future fitness buddy, while still having a lot of options to choose from, regarding my preferred activities and how many times a week a would like to meet up with my fitness buddy.
1.10	Improve usability by adding a main page	It would be nice to have a Main Page where you could easily navigate around the whole app in a quick and easy way. So, I don't have to click through a lot of different menus to get to the page I'm looking for.
1.11	Easy access to calendar	Since I need to do a bit of planning to fit my workouts or mental health activities into my daily life, it is important to me that the app has an easy access to the calendar function.
1.12	LGBTQ+ friendly options when creating profile and inputting criteria	If I am going to use the "Fitness Buddy Finder" app it is important to me that the app is LGBTQ+ friendly in the way that all gender options are available when creating a profile and inputting fitness buddy criteria

9.2) Non-functional Requirements

Table 17: Non-Functional Requirements

ID	Name	Description
2.1	Secure database	I want to be sure that the information that I provide are securely managed and stored and that I can delete them whenever I wish.
2.2	GDPR	I want to know that all the data that I provide are handled in accordance with the Danish and E.U. laws and regulations.
2.3	User friendly/Simplicity	I want an application that will make my life easier and will enjoy using in order to organize workout routines and finding people to exercise together, without having to spend a lot of time to get familiar with the app.
2.4	Tutorials	I want to have access to documentation regarding the use of the platform and health related information.

I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 10 of 16
{Explore the Use Context}

Author	Date	Version	Status	Description
NK	23/10	10.1.1	DR	Use Case Observation
ET/AE	07/11	10.1.2	DR	Social context - questionnaire
NK	02/11	10.1.3	DR	Research Analysis
AE	02/11	10.1.3	DR	Technical Context
ATM	10/11	10.1.3	RW	Reviewed
ATM	14/12	10.2.1	FV	Reviewed

Copenhagen 2022

10) Explore the use context

The use context is split into two areas, social and technical context. For the social context we want to do a questionnaire on how people perceive exercise and the health benefits related to it. While also finding out if and how people are already using some software to support their exercise habits.

For the Technical context, we will be doing an analysis of existing apps on the market, to find out how we can differentiate our app from those already on the market.

10.1) Technical context - Exploring existing apps within our domain:

We have investigated existing apps on the market that are all somewhat inside the domain of our fitness Buddy finder app.

The article "5 apps to help you find your perfect workout buddy" (Jenn, 2015) describes apps that are highly specialized within the domain of exercising. Two of the apps focus on running, one is targeted at professional athletes, while another focuses on biking. The first app described, Jaha, has the same outline that our app promises. It is a tool to match with other people in any user-defined exercise context. Unfortunately, this app does not seem to exist anymore, which leaves us with little information as to why this app was not a success.

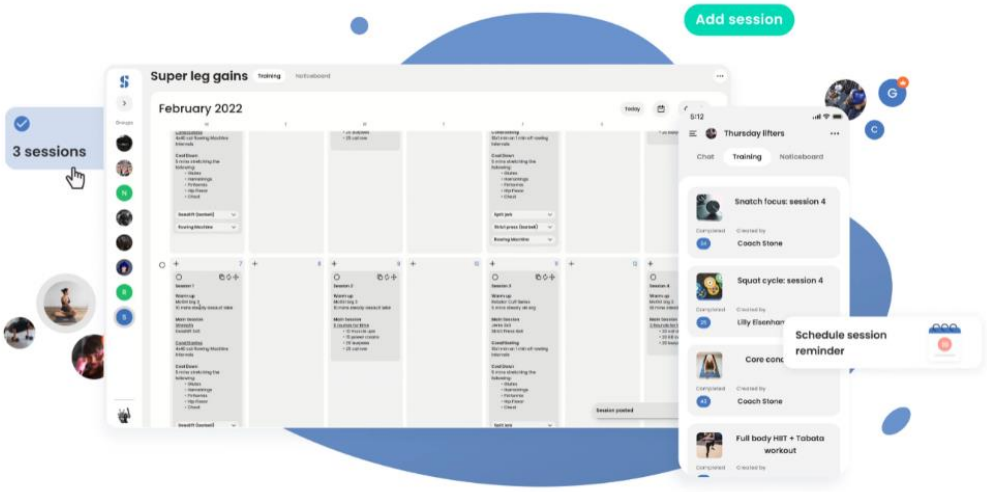
The common factor for these apps is the focus on the social aspect of doing exercise. They all operate on the premise that working out in an organized community can increase motivation for exercising and lead to better results in the process.

Based on this single article, we learned that a selection of the existing apps on the market are more specialized than our proposed software, which try to include any exercise as defined by the users.

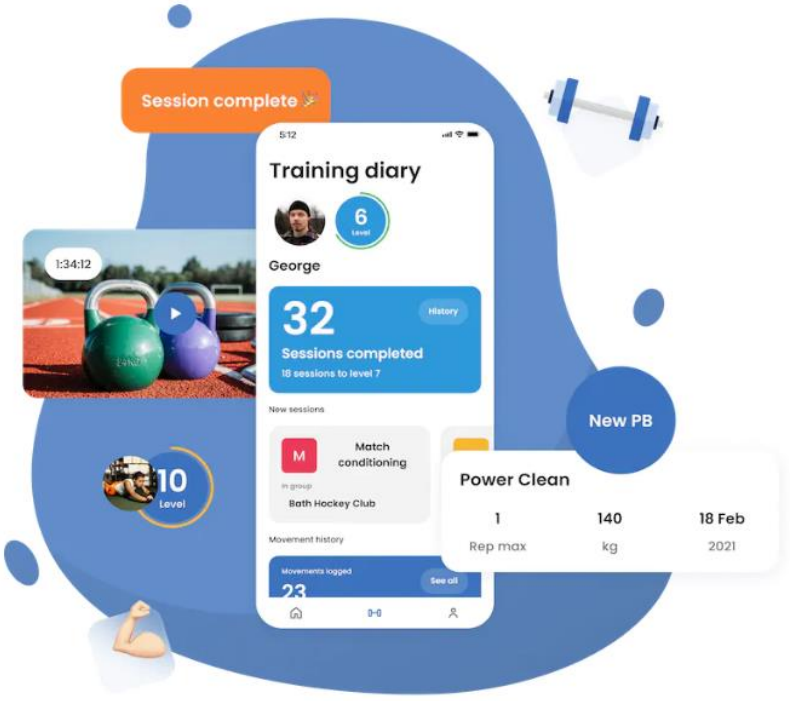
The article "5 social apps to exercise with friends and meet fitness goals together" (Paktar, 2022) contains 5 apps that focus on building networks within the domain of exercising. Some of these apps are more general in terms of types of exercise, and they can be set up to work with any exercise. The focus here is on building communities and sharing ideas and workout regimes with other users.

The most interesting app in our context is the app, Squaddy (https://www.squaddy.app/), which has many traits in common with the fitness Buddy finder we are proposing. The app is built around three main systems:

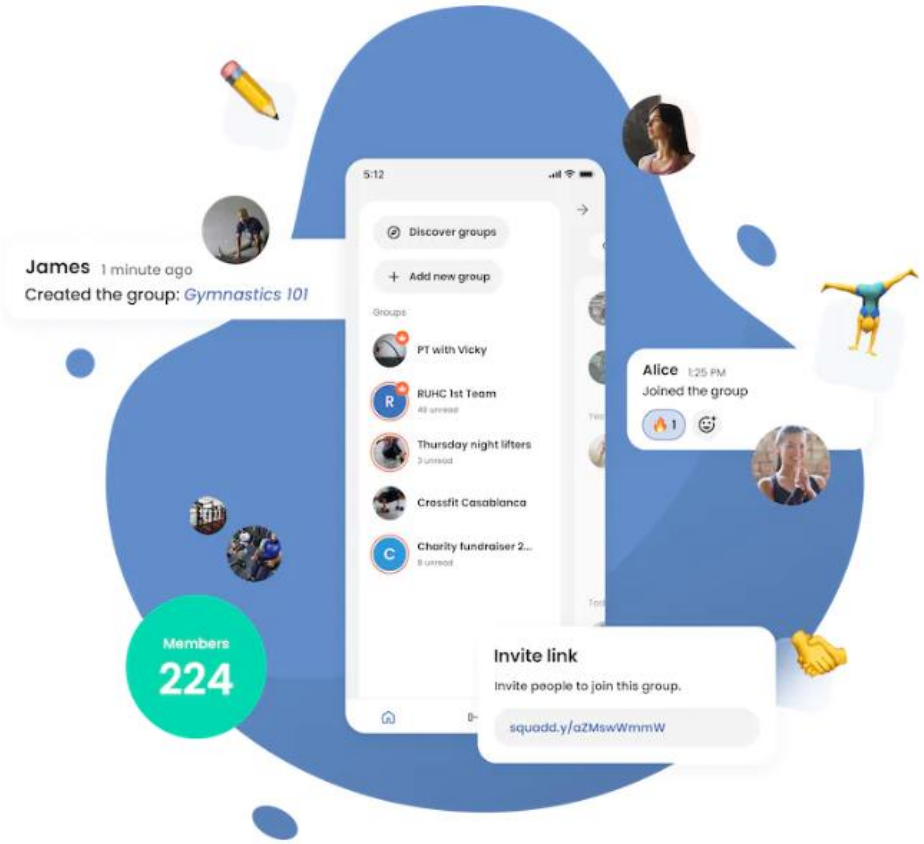
1. A calendar to manage exercise sessions, in which you can set up user defined sessions for other users to see and sign up for. The calendar is a tool for users to keep track of their weekly exercise schedule, hopefully making it easier for the users to incorporate exercise into their daily schedules.



2. A system to track personal progress across all the user's exercise sessions. The progress tracker's purpose is to keep the user motivated by tracking and showing progress made in previous sessions.



3. A group/chat system, in which users can set up groups to communicate about exercise sessions. This System is the connector between users, as they can communicate workout schedules, set up appointments and keep users committed to their work out schedules through bonding with exercise partners.



The Squaddy app does a lot of the same things as the proposed fitness Buddy Finder, but it still has a slight focus on the more conventional workout types, such as running, working out at the gym, CrossFit, and gymnastics.

10.2) Use context – takeaways from the technical context

The Fitness Buddy Finder app has many similarities with existing apps on the market, especially the Squaddy app. Among the primary features of our software are the calendar system, the chat system, a profile page, and a progress tracker, which are all common features of systems within the subject domain.

The primary similarities of the apps described in the articles is the focus on turning exercising into a social activity, rather than something you do alone. The most common features among the subject software are the calendar, group/chat function and progress tracker. The calendar has the purpose of helping people organize exercise

sessions together with other users, supported by the group and chat functions, that can enable people to connect and communicate outside exercise sessions. The progress tracker is in many cases utilized as a motivational tool for the users, where they can track and keep up with the actual benefits of using this kind of software.

The Fitness Buddy Finder software will distinguish itself from existing apps by focusing on a combination of physical fitness and mental health, combined with a social approach to exercising. Furthermore, we are developing a criteria-based match function, that can automatically match people together based on their desired activities within both physical and mental health related activities. The Fitness Buddy Finder app focuses on helping the user build strong communities for exercising, especially within the more.

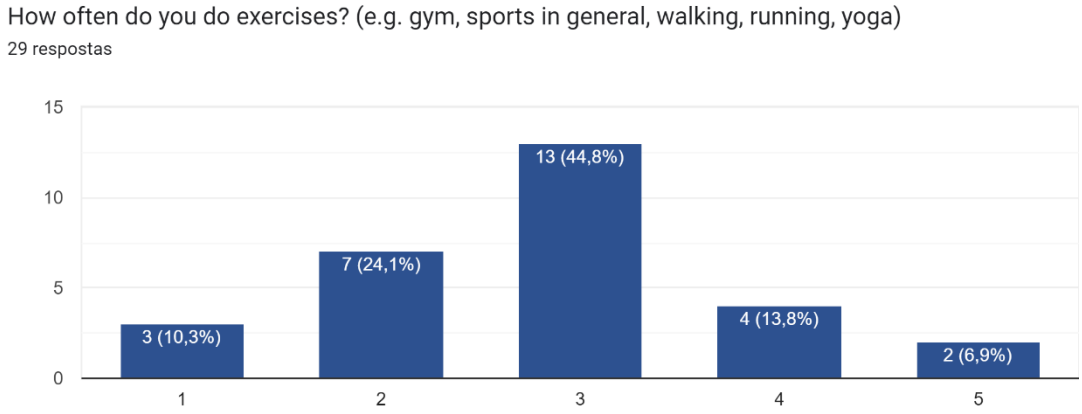
10.3.1) Social context – Questionnaire

To approach the social use context a questionnaire was created. The questions inside it had the objective to give us insights about what people think about this kind of application such as if they would use it, if they already use some fitness app and the most important what people think about exercising together, how often they do it and if they know the benefits to have an exercise buddy. The number of participants was 29.

It was possible to notice that the majority of the people who answer the questionnaire do some exercise regularly but would like to do more. Also, most of them are open minded to exercise with a new buddy and know the benefits of doing exercises together. Some of the people use fitness apps, but most of them don't and we can see it as an opportunity. Furthermore, almost 50% do exercises only alone but observing this result with the question about if they would like to exercise with more people, we can notice that there is an opportunity to connect people. Thus, the insights gained from this questionnaire were good and has shown us that it is possible to create an application with the fitness buddy match purpose.

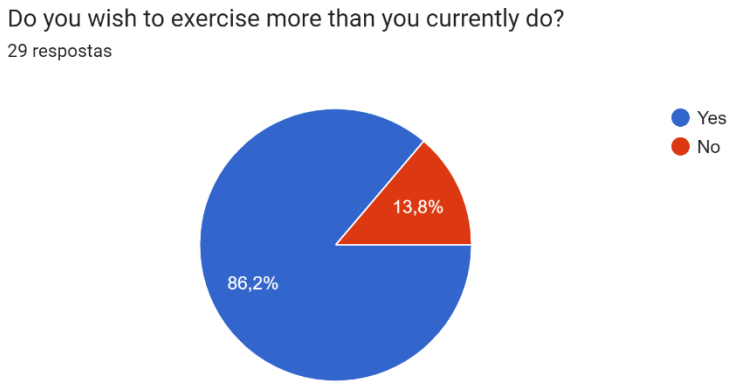
Questionnaire available at the following link: Fitness Buddy Questionnaire

Question 1:

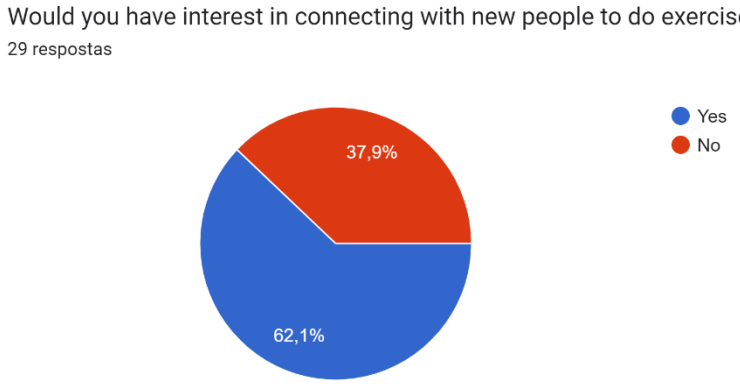


Where 1 is never and 5 is everyday

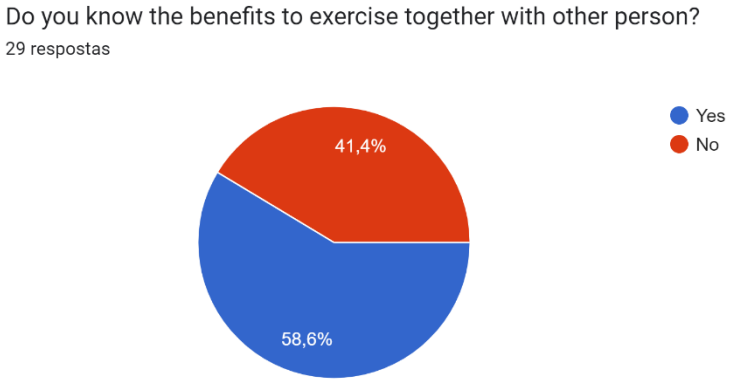
Question 2:



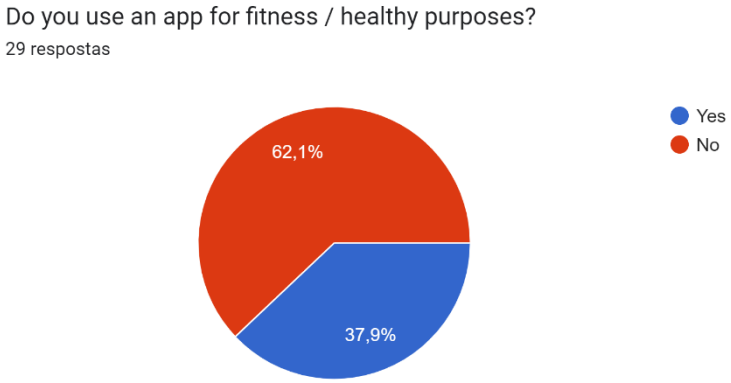
Question 3:



Question 4:



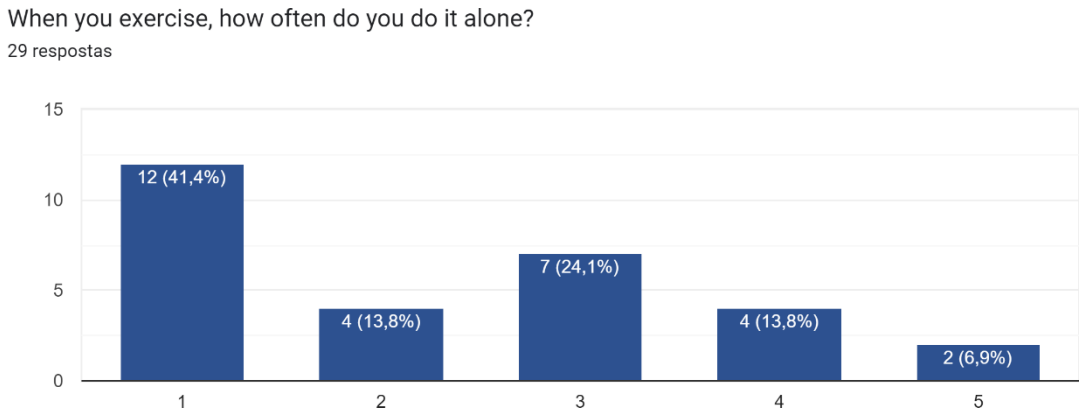
Question 5:



Question 6: If you do, which one?

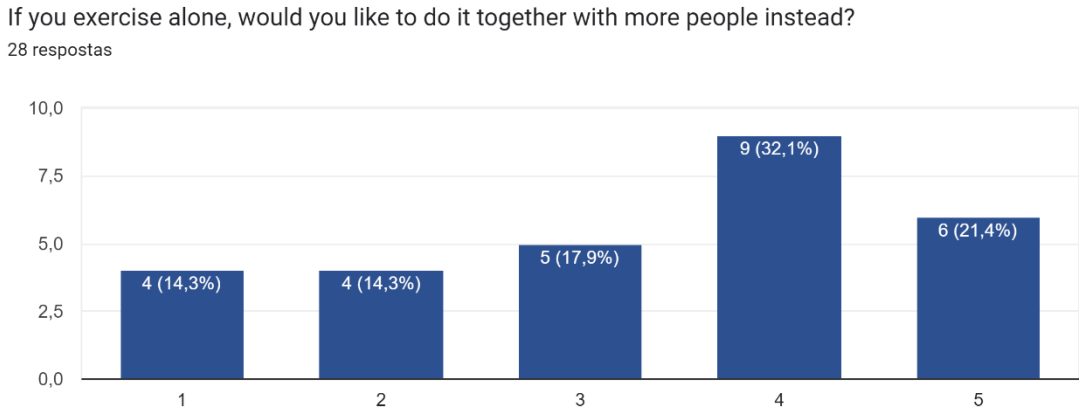
1. Garmin
2. Fitnotes, Huawei health
3. One for my Garmin watch, tracking my running/walking trips, and sometimes other exercises.
4. Zenit
5. I use Ring Fit Adventure for Nintendo Switch. Not really an app though
6. Strava
7. fitness pal
8. Apple Watch for tracking steps etc.
9. Pediometer
10. Samsung Health

Question 7:



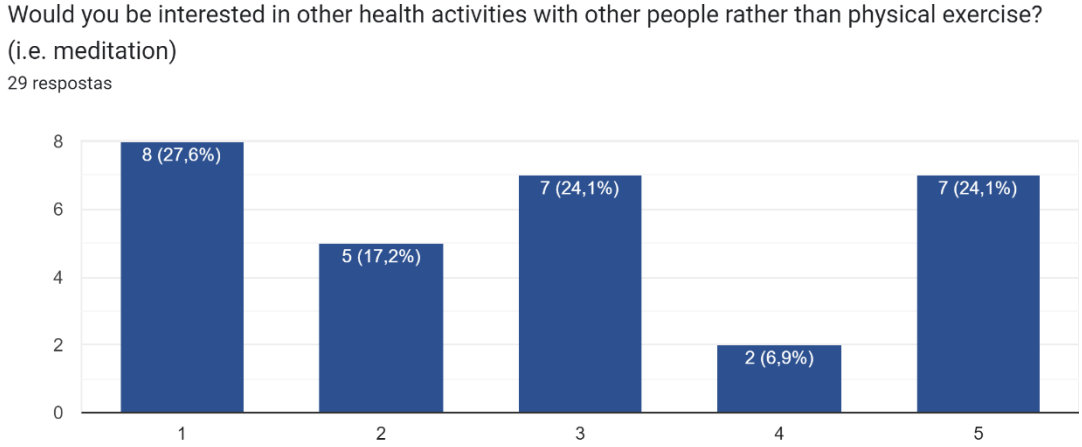
Where 1 is alone and 5 is always exercise together with people

Question 8:



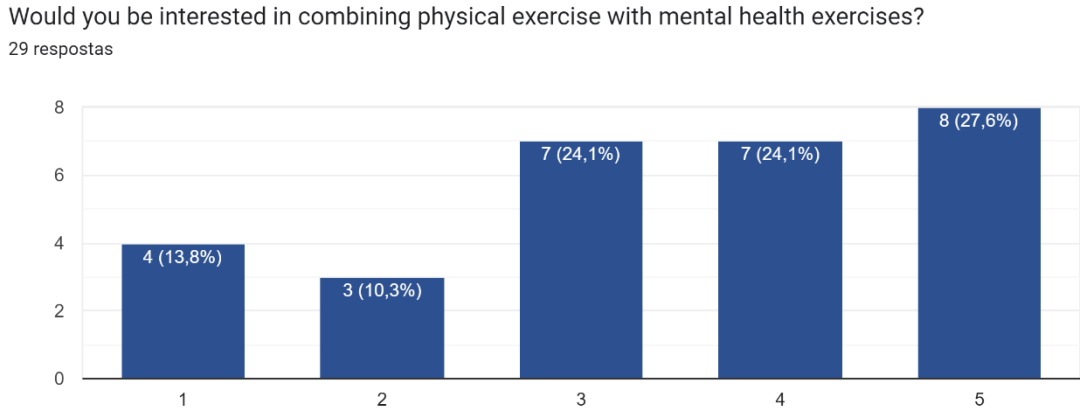
Where 1 is “no, I would not want to” and 5 is “yes, I would love to”

Question 9:



Where 1 is not at all and 5 is yes

Question 10:



Where 1 is not at all and 5 is yes

Question 11: Have you noticed any positive effects in your mental health when you exercise regularly?

Answers:

1. Yes
2. Yes, always. Happier, better sleep, more energy etc.
3. Definitely
4. It can be uplifting if you had a good practice with good teammates.
5. Yup, I feel better with myself
6. I really need to exercise after SE!
7. Yes, it is absolutely well documented that it benefits your mental health. I do a lot of exercising to clear my mind and have fun.
8. yes
9. yea
10. yes. Better sleep, better confidence, better focus in school.
11. Yes, a greater sense of accomplishment.
12. Less stress
13. Yes. Higher energy, self believe etc.
14. Yes, makes me feel great both physically and mentally :)
15. Yea
16. I become happier and more positive in general
17. Yes.
18. Yes, I have more energy in periods I have time for exercising
19. Yes, I feel a lot stronger mentally when exercising!
20. The only exercise I do is biking to different places and going for a walk now and then
21. I have a surplus of energy and sleep better
22. Yes, I keep up positivity easier and I tend to have more energy
23. Yes

10.3.2) Social context - Observation case analysis

As our second social research, we have chosen the observation case analysis, as we believe that our application idea is built on patterns that people develop during their exercise routines and in some cases the absence of them. Thus, the observation has been recorded over a structure of different patterns that have been observed which will be used for the development of the application's features. The parameters of the observation research are set as follows:

- Observation was performed throughout week 42
- Each observation duration had a duration of maximum 2 hours
- Data have been collected in different times within the week, within each gym's opening hours
- The research was conducted in two of biggest gym chains in Denmark

Pattern 1: It is quite rare to see people coming in groups in the gym to exercise. It was only three times over the observation period that some members join the gym in pairs to exercise together.

Pattern 2: Even if members were not coming to the gym with their friends, it seems that a lot of people have or make friends in the gym, or at least know each other and interact while exercising. The efficiency in exercising seems to be increased for people that met friends while exercising.

Pattern 3: Even in the group sessions, members were coming alone but at the end of their session some of them remained in the gym. Most of the ones that remained, were people that were hanging out together after the group session.

Pattern 4: Even if people seemed to work out individually, once they were contacted by another person, they were showing full interest and their working out efficiency seemed to be increased.

Pattern 5: In general, members are working out alone by listening to their own music. As long as someone asks them something (i.e., if they are doing an exercise correctly), they seem more than happy to stop their individual program and help.

Pattern 6: People were engaging a lot with their mobiles (i.e. posting photos and stories on Instagram, Facebook, TikTok). Once they had to deal with human contact (i.e. a trainer asking them something or another member asking to use the machine) then there was less engagement with their phones.

Pattern 7: Members that haven't talked to anyone during their stay in the gym tend to stay less time than people who interact with other members.

Pattern 8: Sometimes the group sessions had more people than the gym itself (members that were following individual working out schedules using the machines).

Observation time log:

Week 42							
16.10.22	17.10.22	18.10.22	19.10.22	20.10.22	21.10.22	22.10.22	23.10.22
09:00-11:00	17:00-19:00	15:00-17:00	18:00-20:00	18:00-20:00	17:00-19:00	09:00-11:00	12:00-14:00
SATS	FW	FW	SATS	SATS	SATS	FW	FW

SATS: SATS Østerbro Nygårdsvej 5, 2100 København
FW: Æbeløgade 4, 2100 København

10.4) Results of Social context Research

Analysis of the results is showing that are Software Qualities (see Document 1 – Chapter 1.1) are well defined and that the application features are well designed in principal but need to be re-evaluated with continuous feedback.

According to the answers to Questions 1-2-3, the participants expressed their wish to exercise more than they currently do, and the majority would be interested in connecting with other people in order to exercise together.

Even though according to the questionnaire most of the participants tend to exercise alone (a fact that it verified by the Observation Case analysis as well – Pattern 1), they would prefer to exercise with others, if they had the option – Question 7 & 8. Pattern 4 confirms the above answers, as people seemed to enjoy interaction while exercising alone.

Moreover, according to the questionnaire's answers, only few are using an app to track their exercising activity that are not unified and not providing any social connection. Mainly these apps are focusing on automatic data recording like step counting etc. Only 10 out of the 29 participants are using some kind of application regarding their workout, out of the 26 participant that responded that are exercising – Question 3. It is important to be noted on that point, that according to Pattern 6 from Observation case analysis, people tend to spend a lot of time on their mobile phones even when they are working out, but as it seems not in fitness related apps.

Combination of mental exercising with physical exercises seems not to be very appealing to our participants, thus the feature needs to be revised, either towards its usability or towards a better explanation of what is meant by this question – Question 10.

Pattern 5 of the Observation case is pointing that people are already engaged to their mobile phones while exercising. One of the major reasons for that is listening to their own music through their mobiles. Thus, a music feature might increase the engagement of the app users.

I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 11 of 16
{Rich Pictures}

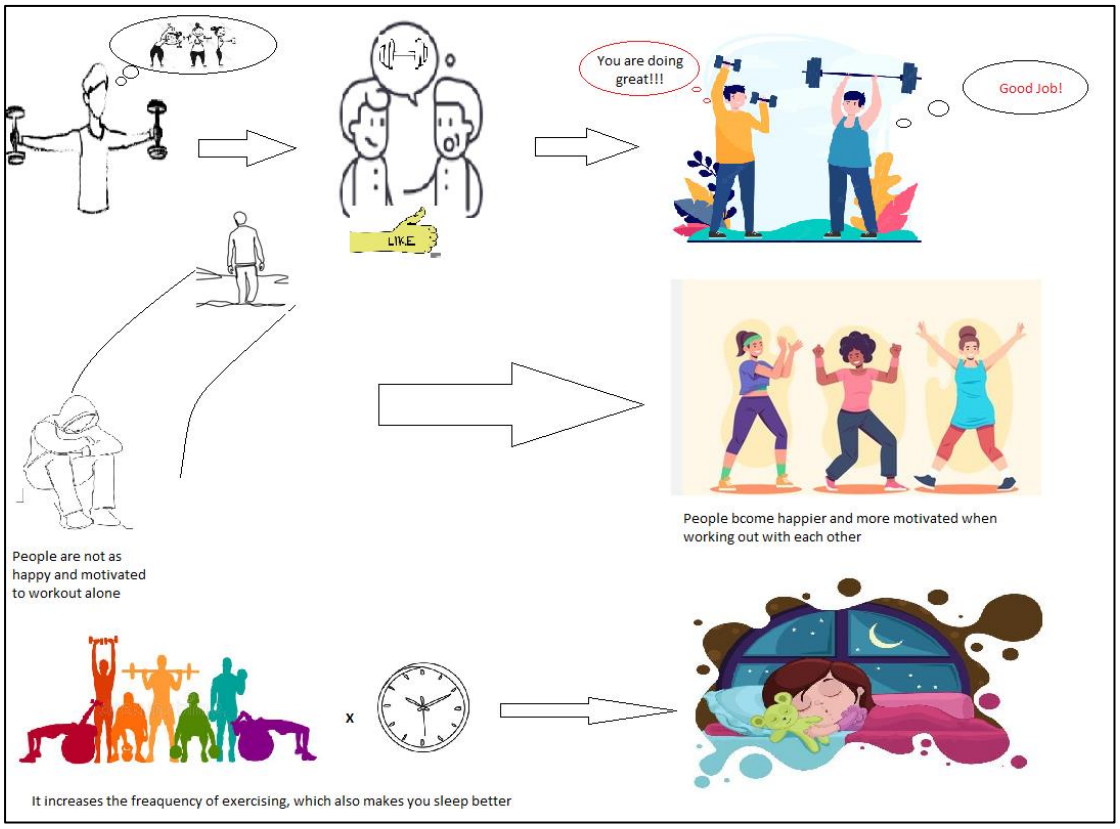
Author	Date	Version	Status	Description
MHG	09/11	11.1.1	DR	First draft of rich picture 1
NSV	09/11	11.1.2	DR	Insert pictures
ET	09/11	11.1.3	DR	Insert picture and write descriptions
NK&NSV	10/11	11.1.4	DR	Insert Picture and description
ATM	11/11	11.1.4	RW	Reviewed
ATM	11/11	11.1.4	FV	Reviewed

Copenhagen 2022

11) Rich Pictures

11.1) Rich Picture: Literature Research

Picture 6: Rich Picture 1

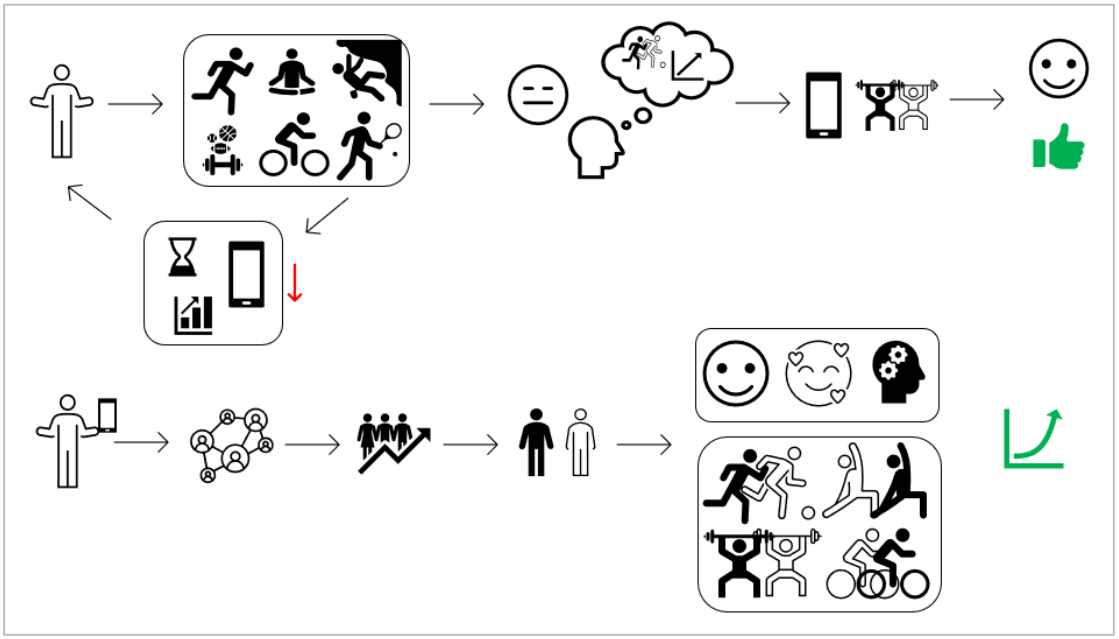


The first rich picture represents our findings of our technical research on literature that already exists. Our findings (as described on Document 10) indicate that people tend to get more motivated when they are working out in groups, rather than when working out alone. Even small gestures like a verbal prompting can create a welcoming environment to exercise more and feel better. It is quite interesting that our findings on the social surveys support the above (see Rich Pictures and descriptions below).

The mental rewards of working out are multiplying when performing exercises with other people as socializing in a major part of group activities. It is also well depicted that exercising with others increase the frequency of exercising activities. All the above lead to the reduction of the stress level and better sleep.

11.2) Rich Picture: social context questionnaire

Picture 7:Rich Picture 2



The above rich picture represents the insights that we got based on the survey answers. Therefore, using the description presented in the last chapter we draw it identifying the main points of the survey and how the users and possible customers feel about functionalities and the main purpose of the fitness buddy application. In other words, we produced this rich picture based on the answers that we got approaching the well-thought key points questions.

There are two paths in the picture, one to represent the user that currently do activities alone and one representing the same user using our application. In the first case we tried to trace a regular profile of people who do exercises, mostly alone. Once this case was the majority in our survey, we could connect this kind of user with two main things:

1- Those people do not have a strong accession to apps to track their progress or any other kind of apps linked with social activities. This was represented by the red down arrow and the box with a mobile and some data progress and time images. The purpose here is to show that this kind of application with only the progress track functionality is not commonly used by people who do exercises regularly.

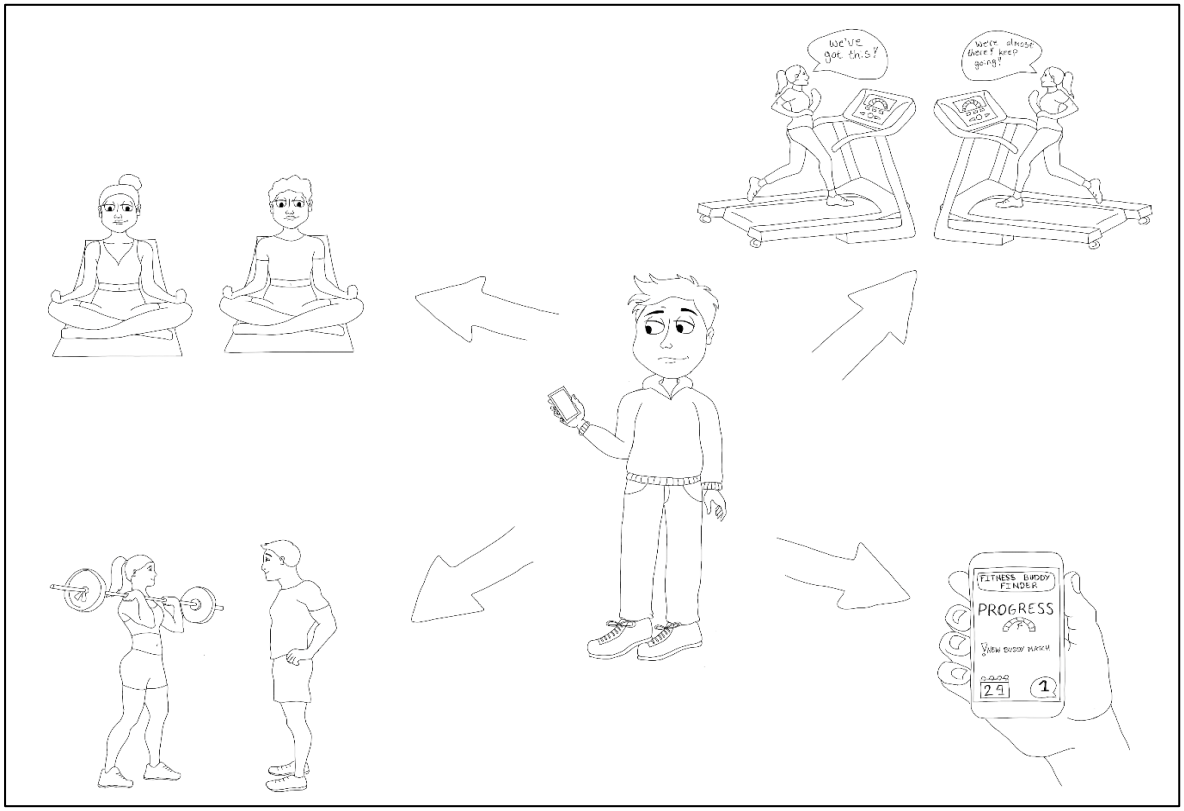
2- Moving forward, based on the answers, we could notice that people who practice exercises regularly have a deep motivation and a wish to improve and increase the frequency of the exercises during the week. Also, those people would like to find someone to do activities together. Most of the people, are open minded to at least try to meet new people

to exercise together and thinks that it is a really good idea to maintain a good practice and improve their motivation to exercise. To represent it we draw the next steps representing the people desire for increase the frequency of the exercises and, also to find a buddy. After that, following the diagram flow we represented the probably high assession of our app based on the survey.

The second path start with the same user but using our application. It shows the path starting for a matchmaking with a new buddy. Then, after the match they can exercise together, improving the performance and motivation, avoiding skip the exercises. Therefore, this flow is to show how to solve the problem of the first one.

11.3) Rich picture: observation case analysis

Picture 8: Rich Picture 3



The rich picture illustration demonstrates the functionalities and values of using the platform.

In the middle of the picture, we see a user of the platform, holding his phone with the fitness buddy app open. From the user, different functions of the app are demonstrated with arrows:

1. A phone showing a simple mockup of the fitness buddy finder app. In the bottom right corner a chat icon with a new message is found, where the user can chat with existing fitness buddies. To the left of the chat icon a calendar icon is shown, where the user will be able to see their planned exercises. Furthermore, a notification will show up on the screen when a new fitness buddy match has been found, while the user will be able to see their exercise progress based on their activity.

2. Various kinds of ways to exercise with a fitness buddy, including doing yoga together, running in fitness and doing weightlifting. What all the exercise sessions have in common is that the users are happy when exercising together, and that they are able to cheer and motivate each other to do better.

I.T. University of Copenhagen, MSc in Software Design
Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 12 of 16

{Analysis Class Diagram over Problem Domain}

Author	Date	Version	Status	Description
ATM	3/11	12.1.1	DR	Making class event table
NSV	7/11	12.1.2	DR	Making initial class diagram
NSV	9/11	12.1.3	DR	Revising class diagram
NSV	10/11	12.1.4	DR	Finishing diagram and adding text explanations
MHG	10/11	12.1.5	DR	Added explanatory text to the Event Class Table
ATM	10/11	12.1.5	RW	Reviewed
NSV	21/11	12.2.1	DR	Revised class event table and updated description accordingly
NK	21/11	12.2.2	DR	Revised Use Case Diagram
ATM	24/11	12.2.2	RW	Reviewed
ATM	08/12	12.2.2	DR	Revision of class event table and class diagram
NSV	14/12	12.3.1	FV	Final revision of class event table and class diagram, and updated text accordingly
ATM	14/12	12.2.2	FV	Reviewed

Copenhagen 2022

12) Analysis Class Diagram over problem Domain

12.1) Class event table

The problem domain that the class event table and the class diagram has been built on, can be described as helping people to build better habits and motivation around exercising, through social bonds.

Table 18: Class-Event Table

Class → / Events ↓	User	Match Making	Activity	Progress Tracker	Calendar
Fitness buddy Criteria is set	X				
Initiate buddy search	X	X			
Two user profiles are matched	X	X			
Match is accepted by users	X	X			
Exercise goals are set	X			X	
An activity is created	X		X		X
Calendar is updated automatically based on created activities			X		X
The progress Tracker is updated automatically when an activity has been completed			X	X	
A challenge between Fitness Buddies is created	X			X	
A badge is rewarded to a user when goals and challenges have been completed			X	X	
A user is notified of new activity recommendations	X				

The class event table shows how the different classes interact with the different modules of the app. To display this, we have organized our classes to consist of “User”, “Activity”, “Progress tracker”, “Calendar” and “Match making”.

The class “User” represents the different types of users/customers related to the problem domain, and is the class that interacts with the different modules of the app the most, which is to be expected as the class represents the users.

The super class “Activity” doesn’t have any subclasses, it represents all the different activities the users can try out, whether it is a physical workout or mental health exercises.

The ”Progress Tracker” will track the exercise progress for each user, to give feedback and motivation on how the user is developing and keeping up with their goals and habits. A progress tracker can both be set up for physical exercises, mental health exercises and create overall goals for the user and their fitness buddies.

The ”Calendar” class will also be a part of the platform and contain all the scheduled exercises for each user. Furthermore, the calendar will be able to store public events

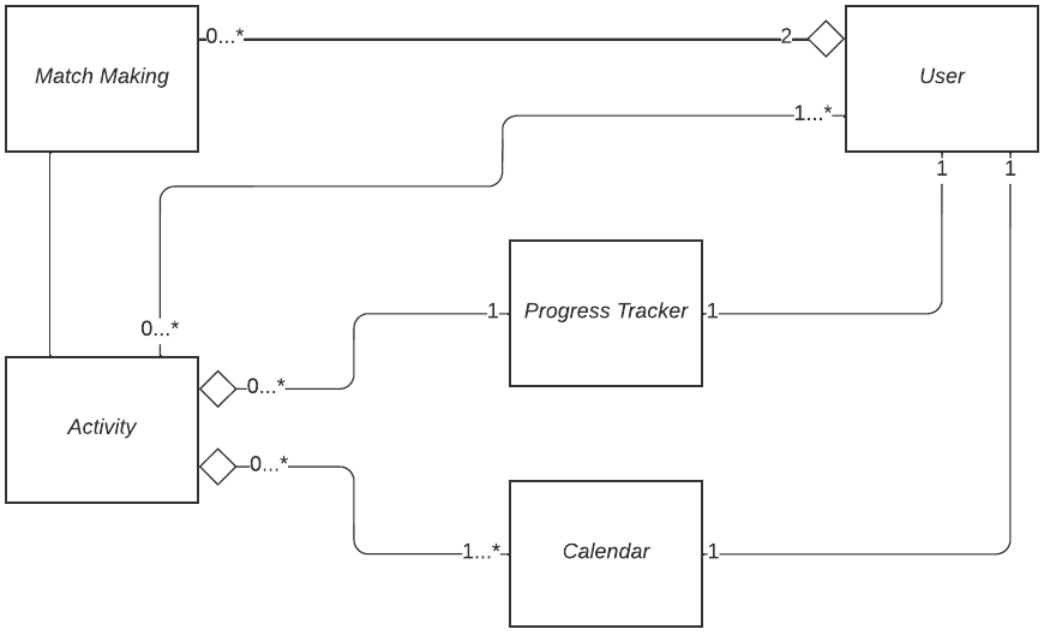
for multiple users and push all scheduled workouts and events to a users' personal calendar.

Lastly, the match making class will contain the primary function of the problem domain, namely matching users to become fitness buddies. A match will be made based on given criteria from the user profiles and will alert two users when a match has been found. Each user then has the ability to approve or reject the match, and afterwards chat with each other and view the other users' full profile, if a match was accepted by both users.

The Class event diagram has been made by using the Miro Online Whiteboard for Collaboration tool (https://miro.com/app/).

12.2) Class diagram

Figure 4: Class Diagram



The Class diagram has been designed by using the Lucid online Visual Collaboration Suite (https://lucid.app/).

The focus of the diagram is to depict the process and relations between classes in order to solve the problem domain. This entails the match making of users on the platform, to enforce the motivation of exercising through social bonds.

The user class represents the highest generalization of the diagram and will initialize most processes and events in the problem domain. When a user has started the match making process and have been matched up with another user, they will then

be able to schedule exercises together. After each exercise have been created or completed, the exercise will be added to or updated in both the calendar and the progress tracker connected to each user.

To show that the creation of an activity is not only dependent on the match making process and can be made by a user at any time, both individually or with a fitness buddy, an association between the user and activity has also been made outside of the match making class.

I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

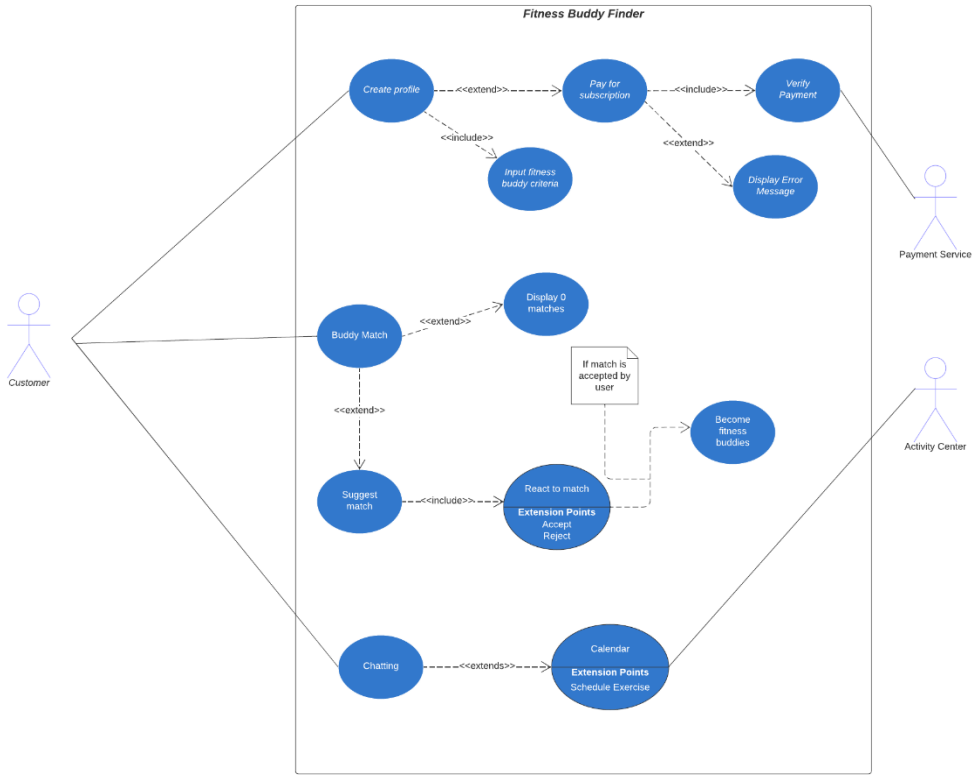
Document 13 of 16
{Use Case Diagram and Scenarios}

Author	Date	Version	Status	Description
NSV & ET	3/11	13.1.1	DR	Making use case diagram
NK	9/11	13.1.2	DR	Scenarios 1 & 2
NK/MHG	10/11	13.1.3	DR	Actors Table and Descriptions
ATM	10/11	13.1.3	RW	Reviewed
NK	18/11	13.2.1	DR	Matching Use case diagram and Actors diagram. Explanation why Class Event Table is slightly different
ATM	24/11	13.2.1	RW	Reviewed
NK	06/12	13.3.1	DR	Revision of Use case diagrams, Actor's table and Scenario 2
ATM	14/12	13.3.1	FV	Reviewed

Copenhagen 2022

13) Use Case diagram and Scenarios

Figure 5: Use Case Diagram



Use Case: The use case diagram represents the main use cases of the system which are identified as Creating a profile, Buddy match and Chat function.
Objects: Customer, Activity Center, Payment Service.
Function: Create profile, Buddy match, Chatting
The Use Case diagram has been designed by using the Lucid online Visual Collaboration Suite (https://lucid.app/).

13.1) Scenarios

13.1.1) Scenario 1

The customer is creating a profile and setting up their exercising settings according to their exercising schedule and level. Physical characteristics can be added as well. The customer is selecting their preferences on what type of exercises they like, which ones they prefer to perform alone and which ones they prefer to perform with company. They also select the characteristics of their ideal fitness buddy. The customer can upload a photo of themselves to support the social aspect of the app.

13.1.2) Scenario 2

The customer is a professional trainer that wants to expand their network to attract more people in the yoga group sessions they organize. The customer is paying the monthly subscription via the Payment Service so to can have more features like organizing group activities, create group chats and create group events on Calendar. The customer is selecting the payment method and the Payment Service is performing the banking details verification. Then the customer is subscribed for the selected period.

13.1.3) Scenario 3

The customer has set up search criteria to find a buddy to do yoga with. The costumer starts the matching function and is automatically matched with four different customers and an activity center, that offers yoga classes for beginners. After the match, the customer can choose which one(s) they wish to connect with. After having chosen to try out the activity center, the user gets access to the exercise sessions held by the activity center through the calendar function. The user chats with the staff of the activity center to check that they have classes for beginners, and then signs up for a beginner's yoga class at the activity center.

13.2) Description of Actors

Table 19: Actors table

ACTORS					
USE CASES		**Customer**	**Payment Service**	**Activity Center**	
Create Profile		✓		✓	
Match Making		✓			
Chat Function		✓		✓	
Subscription Payment		✓	✓		

13.2.1) Customer:

Goal:

A person that owns a profile but wants to find people to exercise with. The customer wants to match with other members to arrange group activities (indoors or outdoors) or activities regarding mental health.

Characteristics:

The systems' users include many customers, with different experience and profile

preferences. They might be gym members, non-gym members or even professional trainers that want to expand their network.

Examples:
A customer that is not a gym member is feeling insecure about their body, but they do not exercise regularly. However, they believe that if they have a buddy to motivate them, their weekly exercise activity will increase.

A customer that is a gym member is spending a lot of time in the gym during the week but is missing human contact, as they have noticed that when they exercise with other people, the process is more fun, and the results are better.

13.2.3) Activity Center:

Goal:
An activity center is represented by a person that owns a profile and is managing a gym or an indoors facility. The gym manager aims to attract more members to the gym, either as individuals or in group activities.

Characteristics:
The systems' users include many gym managers in different locations.

Examples:

Gym manager A wants to increase the revenue of their branch by introducing new

Pilates and Tai Chi group sessions and tries to advertise these new activities to attract new members. Gym manager A knows that the posters in the gym are not enough and wants to reach new potential members via the internet.

Gym manager B has just finished an upgrade to some of the gym's equipment and wants to find ways to attract new customers to avoid an increase to the current membership, so to cover the cost of the upgrade.

13.2.3) Payment Service:

Goal:
The Payment service is the automatic system that is responsible for banking services and verification of the payment details of the customer. The goal is the banking verification to be performed smoothly, secure and to be reliable.

Characteristics:
The payment service is outsourced. Maintenance and troubleshooting are not performed by our team.

Examples:

A customer which is a person wants to buy a monthly subscription. After selecting the monthly package that fit their needs its choses to pay by using their debit card. Then the customer is transferred to the Payment's service input information page, where the user enters their payment information. Then, the payment service is verifying the payment details and either approves the transaction or rejects it. The customer is receiving a notification regarding the output of the transaction.

I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 14 of 16
{Interface Design using Mockups}

Author	Date	Version	Status	Description
NSV & ET & AE	10/11	14.1.1	DR	Designing mockups
ATM	10/11	14.1.1	RW	Reviewed
NK	19/11	14.1.2	DR	Initial thoughts on workshop and Interface Design
MHG/ET	19/11	14.1.3	DR	Fitness buddy Criteria mockup
AE	20/11	14.1.3	DR	Description of each mockup screen
ATM	24/11	14.1.3	RW	Reviewed
NK	29/11	14.1.4	DR	Conducting the workshop
AE	29/11	14.3.3	DR	Results of the workshop
ET	12/12	14.3.4	DR	Workshop design
ET	12/12	14.4.1	DR	Mockup screen changes after workshop
ATM	14/12	14.4.1	FV	Reviewed

Copenhagen 2022

14) Interface Design using Mockups

14.1) Initial thoughts on Mockup workshop

In order to create the Mockup workshop session, the team decided to concentrate on developing the first given scenario from the "Use Case diagram and Scenarios" Document 13. The selection was made mainly because we consider the above the one of the most important aspects of the application design, as it is the first interaction of the user with the app. Moreover, we consider extremely important to explore what the users are requesting in the process of registering and sharing their personal information.

Scenario 1 – profile creation – will be explored in two different ways in the Mockup workshop:

- From the perspective of a customer as a gym member user, as described exactly on Scenario 1 – 13.1.1
- From the perspective of a customer that is professionally certified to train

The workshop as per the above, will be focused on the exploration of profile creation but there will be space for further development related to the other scenarios as well, given the fact that the participants can develop further ideas and notes on the case.

14.2) Interface design

14.2.1) Font

Even though font might be not considered the most important part of Interface design, we believe that it can make a major difference in the feelings that user is experiencing while using our app. We are aiming for a simple but not such a common font, so to create a warming environment for our users but on the same time to provide the "new vibe" feeling.

The main Font that we have decided to use for our application is "Fire Sans". We believe that depicts what is described above and will be tested as well during the Mockup workshop.

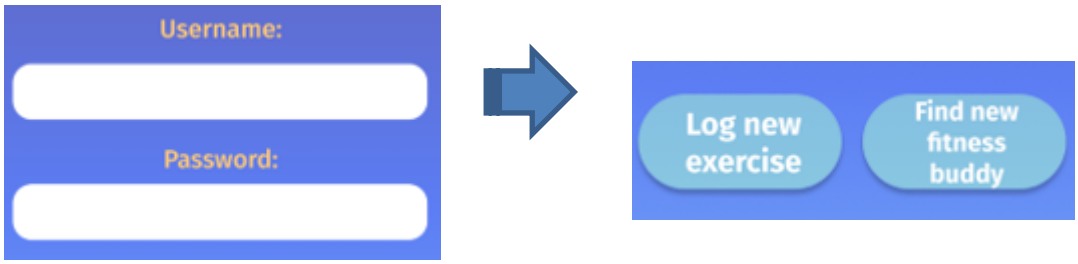
Picture 9: Fira Sans Outline

14.2.2) Buttons

Just like the Fonts described above, the buttons consist of another important component of the designing process. We have used different Background color (see below), corner ratio and shadows, in the different interfaces. Our aim was to create and maintain a warming and easy to use environment, while the user is navigating in the app.

For example, in while the "Log In" page the buttons have a white background, corner ration of 10 and no Shadow, in the "Profile Page", the buttons design change to Light blue background, a corner ratio of 30 and shadowing. We believe that the above is creating the targeted welcoming environment in the "Profile Page". The differences will be clearer below in the Page Interfaces section.

Picture 10: Different Design in Buttons



14.2.3) Colors

Probably the most difficult part of the designing process was to choose the "correct" colors of the Interfaces, in order to achieve our targets regarding the user-experience and usability qualities of our project.

Even though exercising is an intense activity that is normally depicted by colors of the "Reddish" spectrum, we were aiming on creating a warm environment which will produce the sense of relaxation to the user navigating through the application. Thus, we have used a gradient background from purple to blue (for specifications please see the

color pallet below). The fond color is mainly white or yellow, in order to achieve high contrast and an easy-to-read environment. More active colors like Red or Orange can be spotted either when an exercise is overdue or when the user is utilizing a function – such as finding a new Fitness Buddy. In any case, we have decided to use more pale tones of the active colors, so not to create a stressful sense to the users (i.e., if an exercise session is overdue or not performed will be colorized Red, but we do not want to stress the user for not making to achieve their target, but to motivate them to continue their journey). Moreover, we believe that the gradient change of pale colors in the background is relaxing for the eye but at the same time adds a new "modern" feature that is not so common in the existing social applications.

Picture 11: Background Colour Palette

Color	Color Name	Hex Code #RRGGBB	Decimal Code R,G,B
	corn flower blue	#6495ED	(100,149,237)
	deep sky blue	#00BFFF	(0,191,255)
	dodger blue	#1E90FF	(30,144,255)
	light blue	#ADD8E6	(173,216,230)
	sky blue	#87CEEB	(135,206,235)
	light sky blue	#87CEFA	(135,206,250)
	midnight blue	#191970	(25,25,112)
	navy	#000080	(0,0,128)
	dark blue	#00008B	(0,0,139)
	medium blue	#0000CD	(0,0,205)
	blue	#0000FF	(0,0,255)
	royal blue	#4169E1	(65,105,225)
	blue violet	#8A2BE2	(138,43,226)
	indigo	#4B0082	(75,0,130)
	dark slate blue	#483D8B	(72,61,139)
	slate blue	#6A5ACD	(106,90,205)
	medium slate blue	#7B68EE	(123,104,238)
	medium purple	#9370DB	(147,112,219)

Picture 12: Example of pale tones of Red used

Color	Color Name	Hex Code #RRGGBB	Decimal Code R,G,B
	tomato	#FF6347	(255,99,71)
	coral	#FF7F50	(255,127,80)
	indian red	#CD5C5C	(205,92,92)
	light coral	#F08080	(240,128,128)
	dark salmon	#E9967A	(233,150,122)
	salmon	#FA8072	(250,128,114)
	light salmon	#FFA07A	(255,160,122)
	orange red	#FF4500	(255,69,0)
	dark orange	#FF8C00	(255,140,0)
	orange	#FFA500	(255,165,0)

14.2.4) Easy Access

In order to increase the User Experience satisfaction, we were aiming to minimize the effort of the user to navigate through the app and achieve their target. Thus, a lot the designing process was focused on which buttons are the most useful ones to display, so the user can smoothly perform an action.

For example, in the log in page, we have added the option to login by use of the user's fingerprint and there is also a "Help me!" button that can instruct the user on how to navigate through the app. The buttons that we have selected for further navigation are part of the Mockup examination and will have to re-evaluated according to the results.

14.2.4) Page Interfaces

Login screen

The login screen has all the standard functionality that most apps have. You can log in by using your username and password or via fingerprint.

If the user does not already have an account, they can be taken to the account creation page, and if they have any unanswered questions as to why or how they should use this app, then they can press the Help me button, and go to the support page, where they can also see frequently asked questions, to help them get started with using the Fitness Buddy Finder app. **After workshop:** we added a background for the Fitness Buddy Logo.

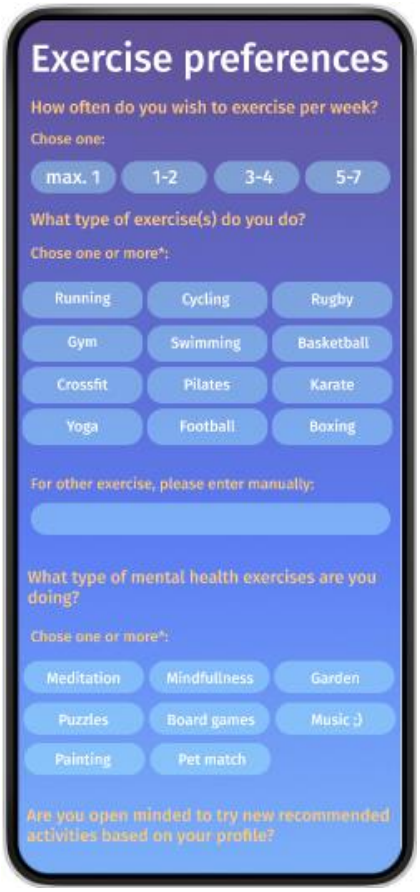
Create account

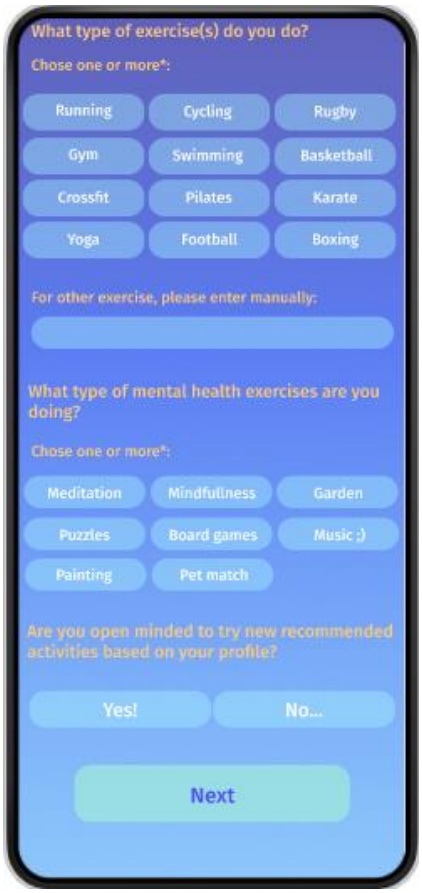
The create account page has six fields that must be answered to proceed. The most important ones are age, and city as they will be used to locate and match a user with appropriate exercise partners. The matcher will use both age and location as parameters for searching for appropriate exercise partners.

The profile picture is used for the profile page, where every user is required to have a picture.

The user can then proceed to exercise preferences, where additional parameters for the matcher can be set.

After workshop: added a user's profile description field. Ideally, it will expand when clicked to make easier to the user write in.





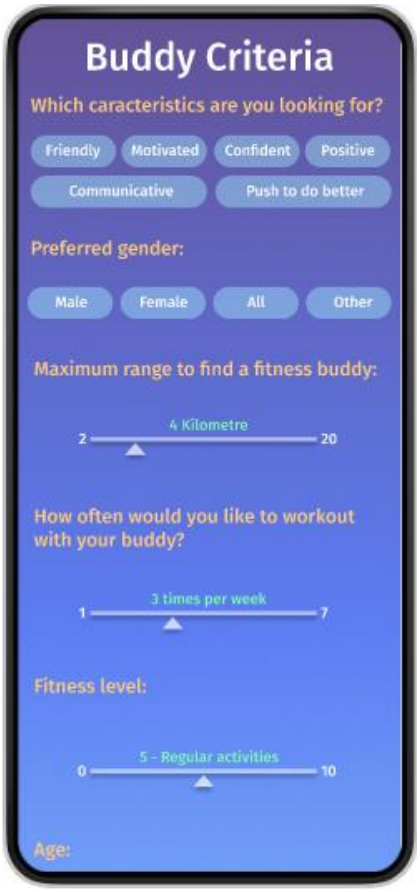
Exercise preferences

The exercise preferences page is where the user can set their initial preferences for getting started exercising with the fitness buddy finder app. The preferences can be changed later, as the users' preferences change.

When the user chooses which types of exercise, they wish to find a buddy for, they can also input their own type of exercise. It is important to the DNA of the fitness buddy finder app, that the user can find a buddy for any kind of activity.

The preferences are split into two main categories: regular exercise and mental health related activities. This is done to support the emphasis on both categories having an equal focus in the fitness buddy finder app.

After workshop: all the fields had its size increased to a better user experience when filling the information. A scroll functionality was added. All fields are available in one page. Both images in the left side are from the same page.



Buddy criteria

The criteria page is where a user can set additional parameters for their matching process.

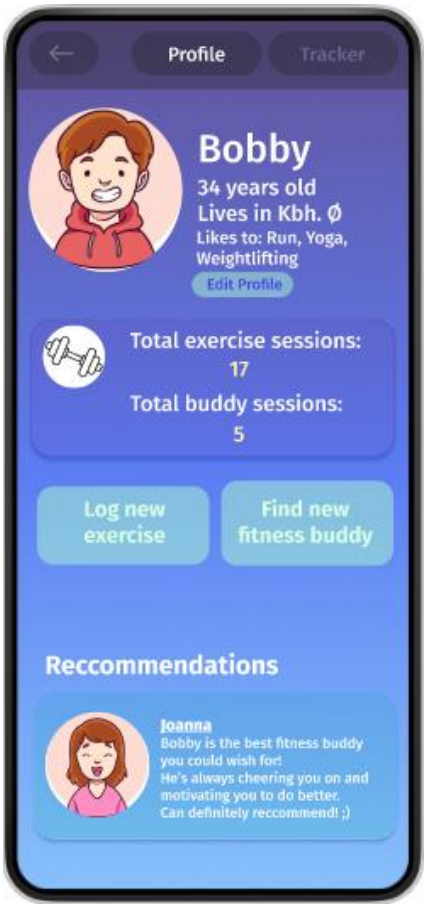
The characteristics matches users based on perceived personality traits, so that users with similar traits have a higher priority in the matcher.

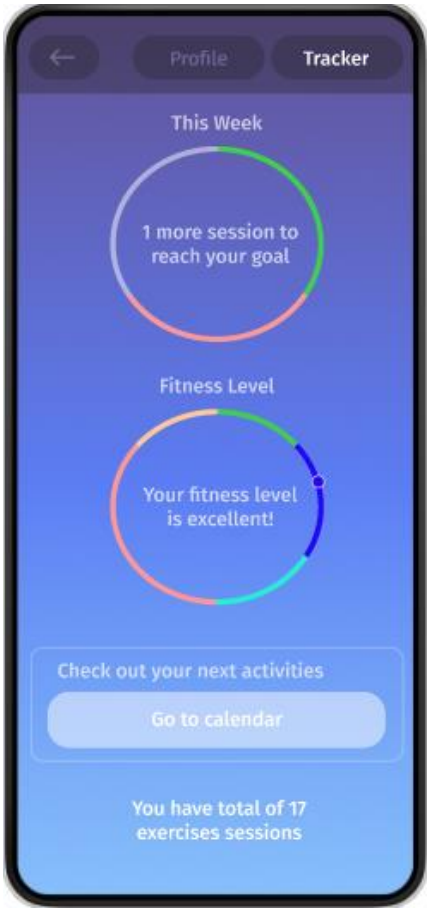
Maximum range tells the matcher how far you are willing to travel to find a fitness buddy.

The workouts per week slider, indicates how many times a week you would prefer to work out.

All these criteria are flexible, meaning that they help the matcher determine the best possible match between users, but does not exclude users from matching.

After workshop: all the fields had its size increased to a better user experience when filling the information. A scroll functionality was added. All fields are available in one page. Both images in the left side are from the same page. More gender options added.





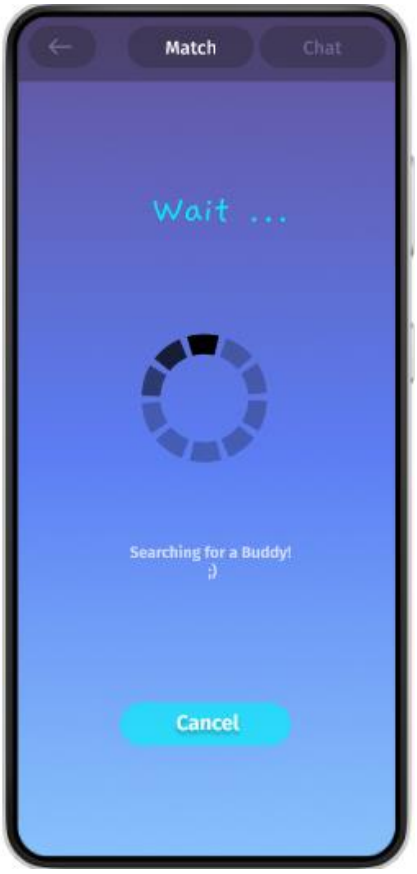
Profile page

The profile page is where you can review how you set up your preferences for exercising. As in the example, we can see that Bobby has set up, in which area he lives in for the purpose of finding relevant fitness buddies for his area. He has also stated his preferred activities so that the Buddy Finder can match him with people who have similar exercise preferences.

This page also features the latest and most relevant recommendations that a user gets from their Exercise Buddies. The recommendations are a motivating factor for the user, and a way for people to get more involved with their fitness buddies through good feedback. The recommendations are also used to assess whether a potential fitness Buddy is relevant before you accept the match and start doing exercise sessions with them. **After workshop:** added edit profile button.

Progress tracker

The progress tracker page is designed to give the user a good overview of how they are progressing with their exercise schedule, and to see the distribution of mental health-related exercises and more physically demanding activities such as a gym workout. The activities are color-coded coded; such that mental health activities are green, and more strength-related activities are red. Exercises that are not yet done and logged in the progress tracker will remain a faded color until it has been logged as completed. The progress circle is a visual representation of how your weekly exercises are distributed between the two main categories. The page shows how many exercise sessions the user has in total and how many they are missing for the current week, to reach their predefined goal. **After workshop**: instead of showing the next events, we changed it to hold only user track information, giving some data about the fitness level as well (gamification). The events information can be accessed in calendar.



~~Matching page~~ – During matchmaking page

The buddy finder page is straightforward, the user starts the matching process by pressing the button, and then the matcher searches for other users with similar preferences regarding location, types of exercise, and experience. When the search is complete, they get one or more matches, and they can go to their profile page and read recommendations, check exercise preferences, experience and any other information that is relevant to them.

Before they start searching, they can specify if they want to find buddies for regular exercise or mental health related exercises.

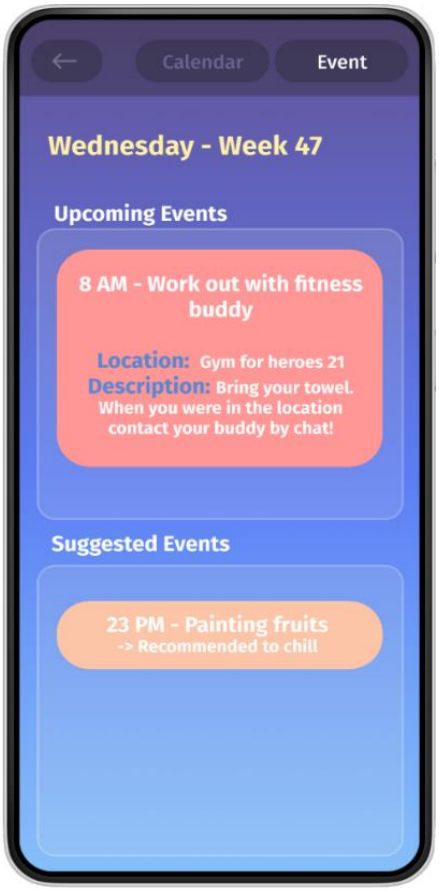
After workshop: this screen was divided in 3 other screens: before, after and during matchmaking.

Chat page

The chat page is a communication tool, where a user can chat with any of their accepted fitness buddies. The chat is used to coordinate with your buddies, on which exercise sessions to join and to keep each other updated on progress.

Activity centers and gym managers can also use the chat function to create group chats, with the purpose of giving information about upcoming sessions.

Users can press the Send Location button, to share their location if they are hosting or participating in an open exercise session.



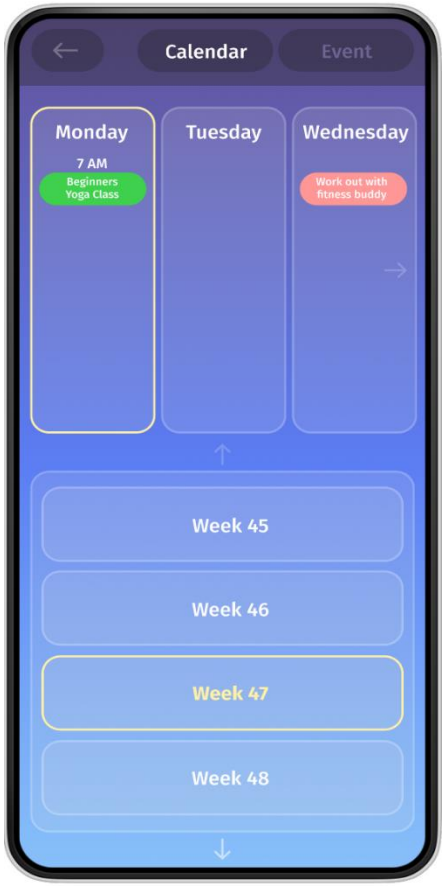
Event page

The events page is an extension of the calendar page.

The user can click on an event in the calendar to be taken to the event page. Here they can see details about upcoming events for that day, such as location, a small description of the event and even notes such as items to bring for the event.

The suggested event shows relevant events based on a user's exercise preferences. The events suggested will be somewhat related to the types of exercise, that a user has set as parameters on their preferences page.

The suggested events have the purpose of expanding the users' possibilities regarding exercising at a specific time of day, or if they are limited within the types of exercise they prefer.

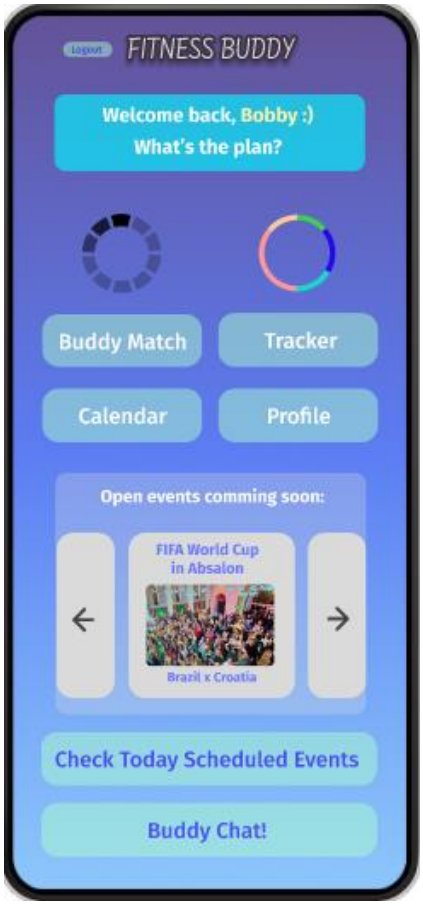


Calendar

The calendar is automatically updated whenever the user joins an exercise session.

The calendar is a tool for the user to keep an overview of when they have exercise sessions and to give a quick overview of the distribution between regular exercise sessions and mental health related exercise sessions.

A user can use the calendar to plan and keep track of past exercises, in order to gain an overview of how many exercise sessions they have in different periods.



After workshop: Main page

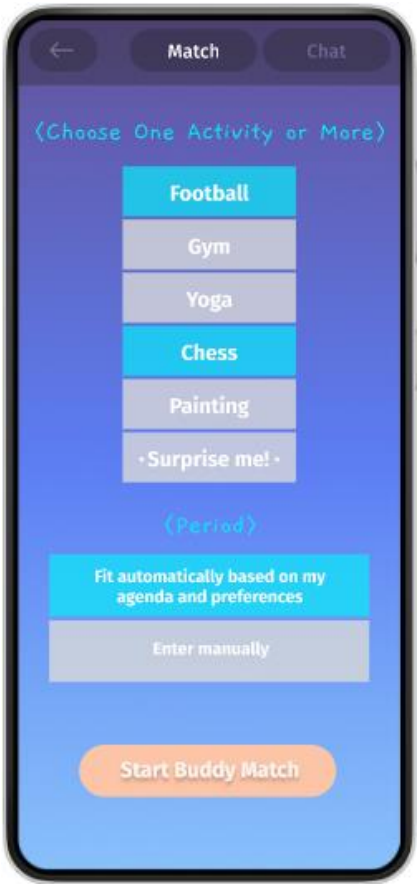
The main page has links to every key page of the program. From here you can go directly to profile, progress tracker, calendar, and the matcher. Also, in this page is possible to see some open events scheduled which are recommended based on the user interests.

You will be taken to the main page if you press the back arrow at the upper left side of the screen on the corresponding pages.

After workshop: After matchmaking page

This page will appear after the matchmaking algorithm find a good match. The user would be able to accept or reject the recommendation. Also, it is possible to check the buddy profile by clicking the button, see profile. If a user rejects the match, they can start the matchmaking again automatically.

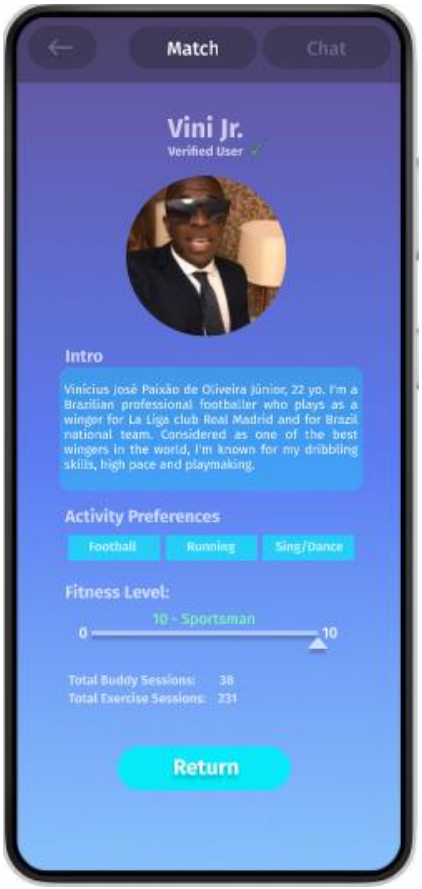
Based on the previous setups that the user filled before starting the matchmaking, the activity and the period will appear. For example, football next Thursday 20:00.



After workshop: Before matchmaking page

This page will appear when the user clicks buddy match either on the main page or in the profile page. The content of this page will vary depending on the user's profile, where the activities shown will be based on profile's preferences. The option *Surprise me!* will appear only if the user previously marked the checkbox about being open minded to try different activities in exercise preferences page. This option will find a different activity that is not on their preferences.

The period option is the previous agenda that user input by default. However, if the user would like to specify an exception in the agenda it is possible inserting the period in clicking the Enter manually button. After setting up the information the user can start the matchmaking by clicking in Start Buddy Match.



After workshop: Buddy Profile Page

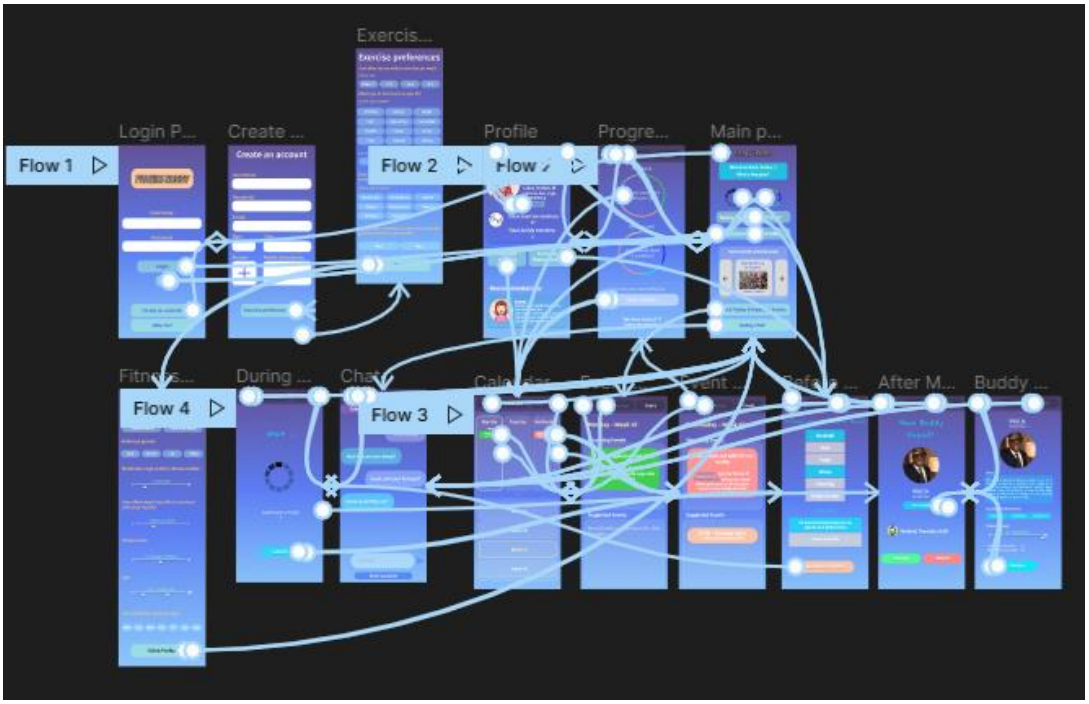
This page will appear when the user clicks the see profile in the after-matchmaking page. Its purpose is to show the core information about the buddy which the current user matched. This information should help their decision about rejecting or accepting the match. All the data on this page should be approved by the profile's owner first. Also, in the future it will be customizable.

14.3) Mock-up workshop

14.3.1) Designing the workshop

The approach to design the workshop started with a discussion about how the flow through the application should be. After discussing about the screens, we created some flows which passes through all functionalities of our application. Starting with the login page, passing through the profile creation, exercises and fitness buddy preferences until the main page which can calls all the other functionalities. We focused in prepare a real use case of the application using Figma. Therefore, it would be possible after the project/application initial presentation let the participants navigates by themselves through it. The picture below shows the sketch of how our application was created in Figma and how are the interactions between the screens, buttons and other components.

Figure 6 Preparing the mock-up for the workshop



14.3.2) Conducting the workshop

The Workshop was conducted on ITU campus, Skybox No 2A03, on Thursday the 24th of November between 14.00hrs and 15.15hrs in collaboration with Group 16. Th meeting begun by presenting ourselves, a quick briefing of our project and the full agenda of the workshop. After, the introduction, we kicked off the workshop by letting the participants experiencing their first contact with the app. In order to conduct the workshop according to Scenario 1 (as described above), we have connected one of our laptops to the big screen in the SkyBox, so everyone being in the room could observe the participant navigating through the first pages. The users were able to simulate a

real time simulation of the Interface pages of the application by utilizing Figma's prototype creation feature.

The first participant had to navigate through the Log In page and start the process of creating a profile as a gym-member. As stated above, everyone could see the navigation process of the user. The participant went through the first pages, created a profile and once the profile was created, explored the extra features of the app like the Calendar and the chat function.

The second participant has been asked to follow the same process of creating a profile as a non-gym member. It is important to note that the second participant was not in the meeting room the time that the first one was conducting the first part of the workshop. Thus, the process of creating a profile was new to them.

Finally, the last part of the workshop was conducted as an open dialogue between all the participants in the room that have observed the process. In this part, the participants had the chance to share second their experience and provide ideas of improvement as real time user stories. Detailed time schedule of the workshop as following:

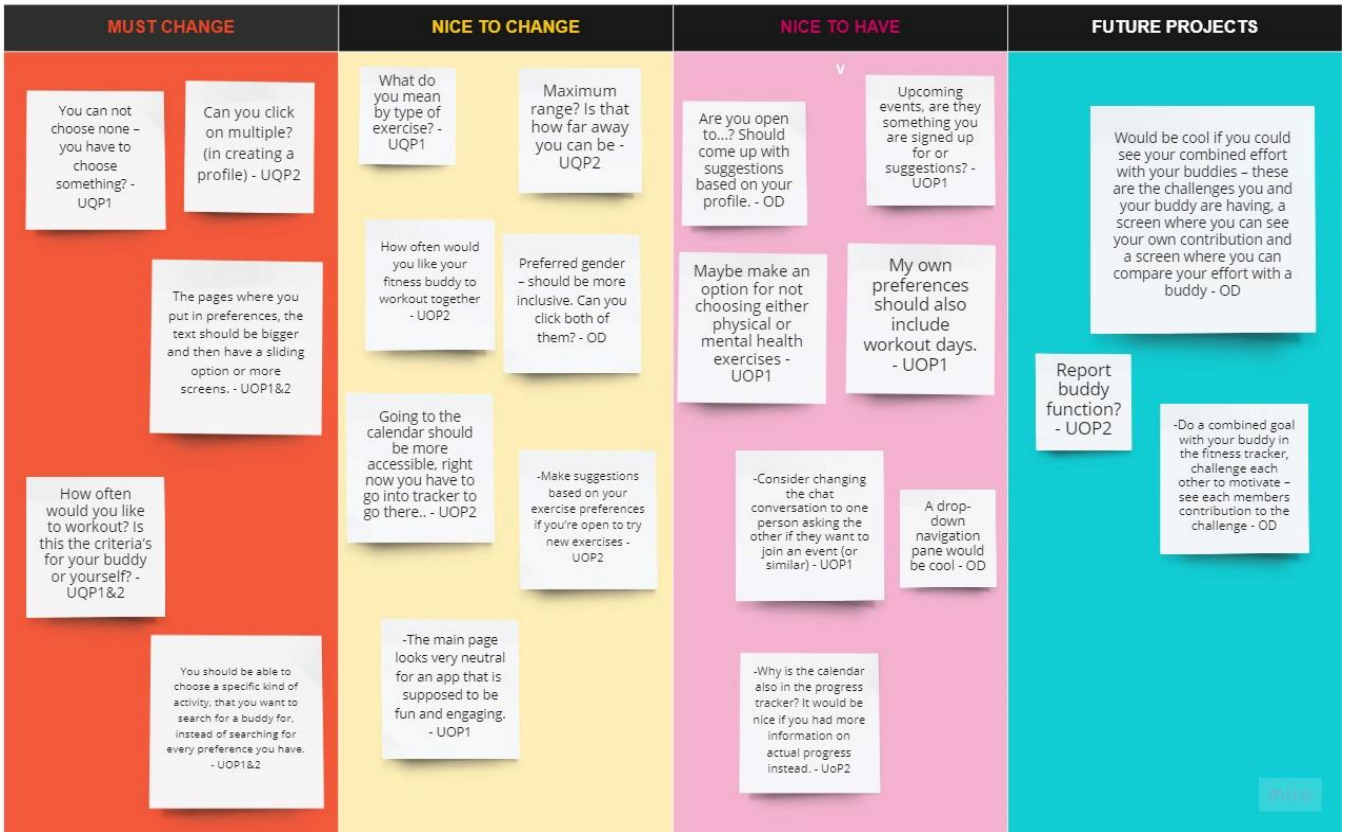
Table 20: Workshop structure

Place&Date	**ITU, Skybox 2A03, 23/11/2022**
14:00-14:15	Presentation of our team – short Introduction of our project – Presentation of the Scenarios
14:15-14:35	Participant 1 performing as per Scenario 1 – Gym member
14:35-15:00	Participant 2 performing as per Scenario 1 – Non-gym member
15:00-15:15	Open discussion – feedback from participants

Notes from the workshop were kept by two different members of our Group, so to analyze them and re-act accordingly as per the "Results" section below. The notes were kept as the participants were conducting their tasks in a free language expression basis. By the end of the workshop, we have gathered our notes and decided to divide them by their importance in terms of usability, ease of access, restrictions that the user faced and general ideas. Thus, we have used three levels of importance as "Must change"," Nice to change", "Nice to have", "Future Projects" as it can be seen in the below table.

Explanation of abbreviations:
UQ: User Question
UO: User Observation
P1: Participant 1
P2: Participant 2
OD: Open Dialogue

Figure 7: Workshop Notes



How the workshop was conducted, how was the setup, who participated, time, delegation of time, notes taken, thoughts for delegation of feedback

14.3.3) Results

The process of the workshop made it clear that the fitness buddy finder app has some flaws that must be reviewed for the app to work intuitively for the user. Everything marked as must change and some posts in nice to change can be categorized as changes to be made, for basic usability of the program.

The rest of the changes are less important and could be considered nice to have.

After the workshop we started to work on the feedback based on its importance. We have made the following changes to the pages as described in figure 7

Figure 7: Changes based on feedback:

Page	Based on this feedback:	Changes:
Profile page	The main page looks very neutral for an app that is supposed to be fun and engaging.	Added a background color to the title to make a stronger brand identity
Exercise preferences page	Can you click on multiple (in creating a profile) UQP2	Added text "Choose one or more"
Exercise preferences page	You cannot choose none, you have to choose something? UQP1	Changed text to "Choose as many as you prefer"
Fitness buddy criteria	How often would you like to work out? Is this the criteria for your buddy or yourself? UQP1&2	Changed the wording to "How often would you like to work out with your fitness buddy?"
Exercise preferences page	What do you mean by type of exercise? UQP1	Changed text to make it more clear that it refers to the listed activities.
Fitness buddy Criteria	Maximum range? Is that how far away you can be? UQP2	Changed wording on the text for the maximum range slider.
Progress Tracker	Why is the calendar also in the progress tracker? It would be nice if you had more information on actual progress instead.	Moved calendar overview out of progress tracker and added a second progress tracker circle.
Matcher	You should be able to choose a specific kind of activity, that you want to search for a buddy for, instead of searching for every preference you have.	Added a field to specify which type of exercise you wish to find a buddy for.
Main page	Going to the calendar should be more accessible. UOP2	Added a main page, where you can easily access every key page in the program.
Fitness buddy Criteria	Preferred gender should be more inclusive. 0D	Added "other" option to the gender selection
Exercise preferences	The text should be bigger and then have a sliding option or more screens	Added a sliding option for the exercise preferences.

We are planning the following features for a future version of the app based on the feedback we got in the workshop. As seen on figure 8

Figure 8: Planned features based on feedback:

Page	Based on this feedback:	Changes to be made:
Profile page and chat page should have a link to a report page.	Report buddy function? UOP2	A report page, where the user can report other users for inappropriate communication or inappropriate behavior at exercise sessions.
Progress tracker	Would be cool if you could see your combined effort with your buddies... OD	An extension or subpage to the progress tracker page, where the user can compare exercise lessons, current progress and take part in challenges together with buddies.
Every page except the login and profile creation page.	A drop-down navigation pane would be cool. OD	Make a drop down navigation pane, that can be used on every page to quickly navigate between pages without having to go back to the main page.

I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 15 of 16
{Quality Assurance Plan}

Author	Date	Version	Status	Description
NK	01/12	15.1.1	DR	Communications Tools test case
NSV	5/12	15.1.2	DR	Calendar test case
MHG	6/12	15.1.3	DR	Dynamic test template – Progress Tracker test case
AE	07/12	15.1.4	DR	Recommendations test case
ET	08/12	15.1.5	DR	Access user information test case
NK	08/12	15.1.6	DR	QA - Goals and Strategy
ET	11/12	15.1.7	DR	Adding Test Table to User Information
MHG	12/12	15.1.8	DR	First version of Review and Inspection
AE	12/12	15.1.9	DR	First version of Reporting
NSV	14/12	15.1.10	DR	Revised calendar test case
ATM	14/12	15.1.9	FV	Reviewed

Copenhagen 2022

15) Quality Assurance Plan

15.1) Goals and strategy

By implementing the Quality Assurance Plan our major target is to make sure that the Product is delivered according to the Software Qualities. Since there was no coding written during the development of these projects, our quality control will focus on the procedures that have been followed, the iteration of these and the documentation process through the Configuration Management Plan. In order to achieve that, we will have to ensure that the product is delivered according to the course requirements. In addition to meeting the mandatory requirements, we need to ensure that the document is coherent and keeps its consistency throughout the project development, besides the iterations and the revisions that we performed. The latter along with not meeting the project's requirements are the higher risks of the quality control.

Our approach towards Software Quality Management (SQM) is in parallel with the agile structure of our Scrum team, so we aim for frequent quality controls over the end of each Sprint, minimize our documentation volume but maximize the handling, accessing, and updating processes of those and by emphasizing that quality control is a team's responsibility and not extra individual workload.

The quality control will be performed in the form of Review and Inspection of our procedures and their documents and Testing of our functions. Thus, we will collect the following data:

1. Dynamic metrics, which are collected by measurements made of a program in execution. These metrics can be collected during system testing or after the system has gone into use. An example might be the number of bug reports or the time taken to complete a computation.
2. Static metrics, which are collected by measurements made of representations of the system, such as the design, program, or documentation. (D. Galin, Addison-Wesley, 2004)

Since the Development and testing go together in Agile Scrum, the Development Team as allocated according to Roles Rotation Plan in Document 4, is performing as Testing Team.

The PO explains the user story at the same time to the Developer and the Tester. While the Developer creates the code, the Tester creates the Test Cases and Test Data. Once the code has been finished, the Tester tests the code. If no bugs are found and all the criteria of the DoD are satisfied, then the User Story will be closed.

15.2) Review and inspection

In the review we will go through the project with a critical set of eyes like a review team would do (Software Engineering, Sommerville). We delegated the documents amongst the scrum team members, and held a review team meeting, where we went through all the documents systematically and discussed our questions and disagreements with a positive and constructive tone.

The project will be carried out using agile development following the scrum-method, therefore it is expected that the scrum team members will spend as little time as possible in the planning phase and then start the code sprints. In document 7.2, there is a picture of a Gantt Chart that estimates the time schedule for the entire project. It seems quite unusual for a scrum team to spend that much time in the planning phase and the Gantt Chart might need to be revised and updated. That time could have been spent on more collaboration with the project's stakeholders, ensuring the product lives up to its expectations.

As for the testing part of the documentation, in document 15.3. If the scrum team has the time for it the different features should probably also be tested together since they are very dependent on each other to provide the expected experience. For example, the progress tracker and the calendar function obviously rely on each other to provide the user with the data that shows their improvement in a given time.

In the Configuration Management document, the configuration management artefacts aren't identified. Given the scrum team won't be producing any actual software in this project identifying configuration artefacts might be a bit difficult but identifying a few artefacts could be beneficial to add to the configuration management part of the project.

According to the management of project risks in document 5.1 and 5.2, having staff turnover is quite likely, because developers are often in high demand, having one of the key developers leaving the scrum team at a critical moment would have a higher effect on the project than Effect 3. Since a key developer leaving at a dire moment could cause the entire time schedule to collapse because the scrum-method used in this project expects to have a shippable product ready for the customer every 2-3 weeks. Because the code sprints in agile development are so short, the team needs to be functioning at a high level to be effective enough to live up to the scrum method, and also the guarantees given to the customer at the start of the project.

Table 21: Document Control checklist

Document No:	Checklist:	Criteria met: Yes – No
Document 1	Application and problem domain of the case explored	
	Core software qualities highlighted	
	Project idea based on case exploration completed	
Document 2	Decide on group activity	
	Complete group activity	
	Document group activity	
Document 3	Each member's experience in working in groups discussed	
	Collaboration protocol agreed	
	Projects and team's expectations agreed	
	Procedures on how and when to revise the protocol agreed	
Document 4	Roles identified	
	Responsibilities identified	
	Roles and responsibilities documented	
	Role rotation set	
Document 5	Risks identified	
	Risks categorized and prioritized	
	Risks analysed	
	Risk mitigation completed	
Document 6	Most appropriate agile software process model chosen	
	The software process models pros identified	
	The software process models cons identified	
Document 7	List of tasks specified	
	Planning method chosen	
	Project schedule created	
	Project management tool used	
	Initial task time initiated	
Document 8	Configuration items identified	
	Configuration management plan set	
	Document Control agreed	
Document 9	Structure of the requirements document decided	
	Functional software requirements identified	
	Non-functional software requirements identified	
Document 10	Brainstorm research for use context	
	Research implementation	
	Results documented	
Document 11	Problem and application domain analysis prepared	
	Rich pictures created and developed	
	Explanatory text for each rich picture added	
Document 12	Class-event table developed	
	Class diagram developed	
	Class diagram relationships represented	
	Class diagram evaluated	

	Classes described	
Document 13	Software activities defined	
	Scenarios developed	
	Actors defined	
	Use case diagram completed	
Document 14	Developing Scenarios agreed	
	Interface designs developed	
	MockUp workshop prepared	
	MockUp workshop conducted	
	Results of MochUp workshop documented	
	Feedback implemented and previous documents revised	
Document 15	Skeleton of quality assurance plan created	
	Quality assurance goals and strategies agreed	
	Dynamic test cases developed	
	Review and inspection-plan conducted	
Document 16	Wrap-up Project activities	
	Reflexions on the project written	

Inspection Plan:

Before the inspection process begins, an inspection-checklist is put together and agreed upon by the developers and the scrum master. The inspection plan consists of scrum team members going through the code in pairs they don't usually work in, to see if new sets of eyes can spot unseen bugs or other shortcomings of the source code. Such as if the structure of the source code will be difficult to expand upon etc. Each pair will highlight their findings and when all the pairs are ready the team will have an inspection meeting to discuss all the different findings. To then discuss, if necessary, how to tackle the shortcomings of the code and prioritize which parts of the source code that needs to be refactored first (Sommerville, 2016).

15.3) Dynamic testing

15.3.1) Communication Tools Test Case

Module Name	Communication Tool
Test Case ID	ct_01
Tester Name/Abbreviation	Neilos/NK
Test Case Description	Check that the users can send and receive text messages
Prerequisites	(e.g., stable connection to the internet)
Environmental Information	1: OS: IOS 2: System: iPhone
Test Scenario	Checking if a text message sent by one user can safely delivered and read by another user.
Non-functional requirement test:	Functional requirement test: X

Test Case ID	Test Steps	Test Input	Expected Results	Actual Results	Status	Comments
pg_01	1. Write text message	1. Enter input content by using iPhone keyboard	We expect that a text message sent from User A can be delivered and read by User B. A notification message of delivery should pop out to user A, while a notification message of receive should pop out on user B. The user should be able to flawlessly send and receive text messages.			Security: The test should ensure that the communication is only between the users and no third parties are involved.
	2. Send test message from user A to User B	2. Enter recipient				User Experience: The test should ensure that the communication tool serves the purpose of connecting the users to expand their exercising network.
	3. Receive delivery notification on user A	3. Write different messages with different characters				Usability: The test should ensure that the communication function is simple and clear.
	4. Receive test message and notification on User B	4. Enter multiple recipients				
	5. Read same text message on User B					
	6. Repeat	5. Measure time between send/receive process and notifications				

15.3.2) Calendar test case

Module Name	Calendar
Test Case ID	C_01
Tester Name/Abbreviation	Nikoline/NSV
Test Case Description	Check if the user is able to use the calendar in the app to manage and book activities.
Prerequisites	Internet connection and access to app
Environmental Information	1: OS: IOS 2: System: iPhone
Test Scenario	Checking if an activity shows up in the calendar when it's been created, and if the user can edit activities through the calendar.
Non-functional requirement test:	Functional requirement test: X

Test Case ID	Test Steps	Test Input	Expected Results	Actual Results	Status	Comments
C_01	1. Create activity	1. Enter input content by using iPhone keyboard	We expect that when an activity has been created, it automatically shows up in the calendar for the user to view. Furthermore, the user should be able to edit the activity when clicking on it in the calendar, whereafter the updates will be saved and updated in the calendar.			Security: The test should ensure that the activity only shows up in the calendar of users connected to the activity
	2. View calendar	2. Click on calendar button in app				User Experience: The test should ensure that the activities are synchronized to the calendar to create an easy overview for the user
	3. Activity can be viewed in the calendar	3. Click on the day/week of the activity to see if it appears				Usability: The test should ensure that the calendar is functioning correct and has a simple overview of activities connected to a user
	4. Edit activity	4. Click on activity and make changes				
	5. Save activity	5. Click on the save button				

15.1.3) Progress Tracker Test Case

Module Name	Progress Tracker
Test Case ID	pg_01
Tester Name/Abbreviation	Marcus/MHG
Test Case Description	Check if the Progress Tracker can keep track of input data over a long period of time.
Prerequisites	Working test set-up.
Environmental Information	1: OS: Mac 2: System: Laptop
Test Scenario	Checking if the progress tracker can represent the input in a nice, clear way, after a long period of random inputs. So, we fast forward a year of using the progress tracker.
Non-functional requirement test:	Functional requirement test: X

Test Case ID	Test Steps	Test Input	Expected Results	Actual Results	Status	Comments
pg_01	1. Enter data on physical performance	1. Enter physical personal bests on the first workout.	We expect the progress tracking on the user's physical fitness shape to run smoothly, with progress being measured easily by numbers. The progress trackers performance on the user's mental health is more unsure because is harder to measure.			Usability: The testing should ensure that the progress tracker is intuitive and easy to use. Understandability: The testing should make it clear whether the progress trackers functions is easy to understand or not. Adaptability: The progress tracker should be easy to adapt to new activities within sports, physical fitness, or mental health exercises, if these are updated with new activities to try out.
	2. Enter data on mental health exercises	2. Enter state of mental health on day one.				
	3. Repeat random inputs in both categories.	3. Randomly inserting data on improvements in both categories				
	4. Ask for progress information.	4. Tracking the progress				

15.3.4) Access the user information Test Case

Module Name	User Information
Test Case ID	ui_01
Tester Name/Abbreviation	Eduardo/EDTR
Test Case Description	Check if the user information is accessible and secure about data leak.
Prerequisites	(e.g., stable connection to the internet) (application's compatible tool to mockup functionalities)
Environmental Information	1: OS: Windows 2: System: Laptop
Test Scenario	Checking if the user information is accessible by itself or by a buddy matched. The user should see only what the profile's owner wants to show. The test scenario includes mocking a profile match as well as a profile editing process. Monitors the whole of matching and create/edit profile identifying weak and vulnerability points where data could be stolen.
Non-functional requirement test:	Functional requirement test: X

Test Case ID	Test Steps	Test Input/ Output	Expected Results	Actual Results	Status	Comments
ui_01	1. Profile creation/edited procedure	1. Enter well known data as a normal user would do.	We expect the user can check their profile information as well as the information from the buddy matched without compromising the security. All the information requests should be done using security layers. Also, we expect that the application be trustful against the well-known			
	2. Monitors how the data goes from the front application to the protected databases and vice versa.	2. Try to get data without authorization. Verify data leak. Simulate cyberattack (e.g., SQL injection).				Security: Block personal information leak. Be a trustful platform where the user can put their data without worries. Assurance that the user data will not be used without authorization.

	3. Simulate a Fitness Buddy Match	3. Input data about preferences and start the match-making.	types of cyber-attacks.			
	4. Access information of the match results.	Check the user's visible information about the buddy matched. Check also all the information that come from the backend as response of the API request. Repeat step 2.				Usability: The user and the buddy data information should be easy to be checked. User Experience: The application is giving to the user the chance to accept or reject a buddy match focusing on let it be more comfortable with who it would like to do an activity together. Simplifying the process to be faster and continue the matchmaking if the buddy information is not affordable based on the user judge.
	5. Repeat the previous processes several times with different data.	5. Insert different data and repeat the match / monitoring step (1,2,3,4). Also, check if all the requests are being doing with good security levels (e.g., HTTPs).				

15.3.5) Recommendations Test Case

Module Name	Recommendations page.
Test Case ID	rp_01
Tester Name/Abbreviation	Andreas/AE
Test Case Description	Check if the recommendations are going through the process of verification before being put up on the profile page. E.g., Check if the internal profanity filter is working to block recommendations that are not acceptable.
Prerequisites	(e.g., stable connection to the internet) (Verification process has been defined)
Environmental Information	1: OS: Mac 2: System: Laptop
Test Scenario	Checking if the recommendations process is working, such that entries are being verified before they are put up on the corresponding users profile page for other users to see.

Test Case ID	Test Steps	Test Input	Expected Results	Actual Results	Status	Comments
rp_01	1. Enter data to the recommendations page.	1. Enter one recommendation on a user's page. one that should pass the verification process.	We expect the recommendations page to block any recommendation that contains profanity and accept recommendations that does not. The test should also show that the recommendation is going to the correct profile page and that other users are able to see the recommendations.			
	2. Enter additional data to the recommendations page.	2. Enter one recommendation on the same user's page. One that contains profanity, such that it should be blocked.				
	3. Repeat inputting different data to the system.	3. Enter additional recommendations on pages to ensure that the profanity filter works as intended. Such that profanity is blocked and other recommendation are passed on to the profile page.				Security: The test ensures that no personal data or profanity goes onto the recommendations page.
	4. Ensure that recommendation goes to the correct page	4. Manually check that when feedback is entered in the system, that it goes to the correct profile page after being verified.				User Experience: The test must show that the recommendations are easily accessible to another user's and that they can access it before

						accepting a match with another user.
	5. Ensure that other users can read the recommendation on a user's profile page.	5. Check that one user can access and read recommendation on the profile page of user's they are matched with.				Usability: The test must show that the recommendations page works together with the profile page to show the most relevant recommendations on a user's page.

I.T. University of Copenhagen, MSc in Software Design

Course: Software Engineering
Group 17

Fitness Buddy Finder Case

Students: Nikoline Stensbæk Vejby, Andreas Elsberg, Neilos Kotsiopoulos, Eduardo Trindade, Marcus Holst Gamer

Project Supervisor: Zachery Russel Kuntz

Document 16 of 16
{Reflections on the project}

Author	Date	Version	Status	Description
ATM	14.12	16.1.1	FV	All group members sat together and reflected on the project

Copenhagen 2022

16) Reflections on the Project

During our time working on the project, we learned new tools and methods in the process of project management within the Software Development. But we have also learned a lot of things about each other and how to work efficiently as a group when implementing fundamentals of agile development. We have learned about each other's strong sides and preferred working patterns, and it took some time getting to know each other, but we became more effective as a group as time went on. During the semester we experienced that it was often harder to achieve a good outcome with the project activities, when we just delegated the task amongst us and went our separate ways. Instead, we started spending more time together working as a real scrum-team. We did experience problems with communication and coordination, especially in the beginning of the project, as we had to find a way to get used to each other's schedules.

Project activities that went well from the get-go was for example when we had to choose our software process model. We disagreed at first on what model we wanted to use, which resulted in both getting to learn a lot about XP-programming and Scrum, because all team members had to come up with arguments for picking their preferred software process model. This resulted in a good in-depth discussion on the different process models, before we decided to go with scrum, with all group members agreeing on Scrum being the best fit for this project.

Project activities that proved to be more challenging was the use-case diagram, the class diagram and the mock-ups for our workshop. The class diagram proved to be a demanding task, partly because it was a new concept for most of us and because it had to be aligned with other parts of the project such as class events table. When creating both the class event table and the class diagram, we were initially very challenged in understanding the approach to it correctly. We had problems fully understanding which classes were a part of the application domain and which were included in the problem domain, and therefore had to revise the diagrams several times. Likewise we had to do several revisions on the use case diagram, because we had trouble understanding how our actors should be set up properly. Initially we made the diagram and the actors too complex, showing all of our different types of users/customers under a generalizing actor. After the feedback we got and following discussions, we realized that this was not necessary and ended up taking a step back and simplifying our diagram for it to make more sense.

The mock-ups were a fun project activity because a lot of us enjoy being creative, and designing the mock-ups was a good way to see what our app could actually have looked like. However, learning to use Figma, the tool we chose for the mock-ups,

proved to be time consuming, and so designing the mock-ups took significantly longer than expected.

As for our chosen software qualities for the Fitness Buddy Finder app, we are looking back at the software qualities we originally picked and are still happy with our choices. The software qualities we picked in the early stages of our product remained a good match for the duration of the project. The app's ability to be easy to use, easy to expand upon with new functions and activities, and safe for the users to keep personal information seems vital for any social app. Therefore, as our product became more defined as time went on, our software qualities always seemed to be a good choice.

The different project activities required us to learn new tools along the way to complete our tasks. For project planning we used Trello, which proved to be easy to learn and very useful for us. To keep track of all the steps that needed to be done to finish each activity. When we created our Class Events Table all five group members worked on it simultaneously and for that we used Miro, which proved to be a fun experience. For our use case diagram and class diagram we used the tool Lucid Chart and its UML-functions proved to be easy to learn and very useful. The most demanding tool we got to use during the project was Figma, which we used for designing our mock-ups by making interfaces, creating buttons for different functions in the app, and making our front and profile page etc.

References

Bendix, L., 2022. [online] *A short introduction to Software Configuration Management* Fileadmin.cs.lth.se. Available at: <https://fileadmin.cs.lth.se/cs/Personal/Lars_Bendix/Teaching/1-ECTS-SCM/90min/ITU-22/LNSCM.pdf> [Accessed 22 September 2022].

D. Galin, Addison-Wesley, 2004.Software Quality Assurance: From Theory to Implementation.

Sommerville, 2016 - Software Engineering 10th GLOBAL Edition

Jenn, Fit Bottomed Girls. 2015. *5 Apps to Help You Find Your Perfect Workout Buddy*. [online] Available at: <https://fitbottomedgirls.com/2015/09/5-apps-to-help-you-find-your-perfect-workout-buddy/> [Accessed 8 September 2022].

Studysites.sagepub.com. 2022. [online] Available at: <https://study-sites.sagepub.com/northouseintro2e/study/resources/questionnaires/89527_03q.pdf> [Accessed 20 September 2022].

Wiki.c2.com. 2022. [online] Available at: <https://wiki.c2.com/?ExtremeRoles> [Accessed 18 September 2022].

Coursea, 2022. [online] Coursera. Available at: <https://www.coursera.org/articles/scrum-roles-and-responsibilities> [Accessed 20 September 2022].

Gualtieri, M., 2011. *The Seven Qualities Of Wildly Desirable Software*. [online] Forrester. Available at: <https://www.forrester.com/blogs/11-02-11-the_seven_qualities_of_wildly_desirable_software/#:~:text=Forrester%20defines%20the%20seven%20qualities,adaptability%2C%20security%2C%20and%20economy> [Accessed 12 September 2022].

Hoffmann, J., 2022. *A Day in the Life of a Scrum Master*. [online] Resources.scrumalliance.org. Available at: <https://resources.scrumalliance.org/Article/day-life-scrum-master> [Accessed 20 September 2022].

Software testing help. 2022. *Software testing help*. [online] Available at: <https://www.softwaretestinghelp.com/scrum-roles-responsibilities/> [Accessed 20 September 2022].

Sommerville, I., 2011. *Software engineering*. Boston: Pearson, p.78.

Steinhilber, B., 2017. *Why it's easier to get fit in a group*. [online] NBC News. Available at: <https://www.nbcnews.com/better/health/why-you-should-work-out-crowd-ncna798936> [Accessed 12 September 2022].

Wells, D., 2022. *When should I use Extreme Programming*. [online] Extremeprogramming.org. Available at: <http://www.extremeprogramming.org/when.html> [Accessed 18 September 2022].

2022. *Why Does Scrum Work?*. [online] Available at: <https://www.agilest.org/scrum/why-does-scrum-work/> [Accessed 19 September 2022].

Plante, T., 2001. Does Exercising with Another Enhance the Stress-Reducing Benefits of Exercise? [online] Available at: https://www.psychologytoday.com/files/attachments/34033/exercise-another.pdf

Patkar, M., 2022. 5 social workout apps to exercise with Friends and meet fitness goals together. [online] Available at: https://www.makeuseof.com/social-workout-apps-to-exercise-with-friends/

Appendix A – Subtasks

Subtask		Person-day	Fibonacci nº	Task
1	Abstract / Case description	15	8	8.2
2	Software qualities	10	3	8.2
3	Sharing of personal things	10	3	8.3
4	Documentation of fun activity together	10	3	8.3
5	Debriefing	10	3	8.4
6	Agree on team's protocol	10	3	8.4
7	Use case stories	15	5	8.2
8	XP roles and descriptions	5	2	8.10
9	Scrum roles and descriptions	5	2	8.10
10	Scrum research	5	2	8.10
11	Xp research	5	2	8.10
12	Add and review the text	4	1	8.10
13	Why did we choose this model	25	13	8.11
14	Pros and cons	12	3	8.11
15	Description + pros and cons of other models	16	5	8.11
16	Brainstorm on risks	10	3	8.8
17	Categorize and prioritize risks	15	5	8.8
18	Risk mitigation	5	2	8.8
19	Risk register	3	1	8.8
20	Identify roles	10	3	8.6
21	Role descriptions and responsibilities	10	3	8.6
22	Implementation of roles	5	2	8.6

23	Documentation of roles	6	2	8.6
24	Schedule - Gantt Chart – Trello	16	5	8.7
25	Initial planning	9	3	8.7
26	Task estimations - Planning Poker	15	5	8.7
27	Final meeting and agreement	5	2	8.7
28	Work breakdown structure	8	3	8.7
29	Create project schedule	6	2	8.7
30	Assign estimations/due dates	2	1	8.7
31	Content - Configuration Items	10	5	8.9
32	Plan Draft	10	5	8.9
33	Review of the Configuration Plan and Group Discussion	5	2	8.9
34	Possible Plan Changes	2	1	8.9
35	Research and brainstorm	10	5	8.9
36	Mental Health	3	1	8.14
37	Manager of a fitness center	3	1	8.14
38	Someone who doesn’t exercise	3	1	8.14
39	User who’s not a member of fitness	3	1	8.14
40	User who's a member of fitness	3	1	8.14
41	Reassign Roles based on SPM	10	5	8.13
42	Redefine roles based on SPM	10	5	8.13
43	Use the Template's document with the defined metadata	2	1	8.12
44	Documents combination	4	2	8.12
45	Design a Logo	8	3	8.20
46	Chose the Color scheme	8	3	8.20

47	Discuss about the results / Group agreement	5	2	8.20
48	Understand and Define the Template's Metadata	1	1	8.9
49	Create the document	3	1	8.9
50	Apply the Template in the previous documentation	1	1	8.9
51	Structure and Notation Definition	4	2	8.9
52	Define Visual components	8	3	8.20
53	Define usability and main functionalities applying UI/UX design	8	3	8.20
54	Think about secondary screens called by the main one	12	5	8.20
55	Review the final main screen	8	3	8.20
56	Build a visual sketch with a properly tool	60	21	8.20
57	List Functional Requirements	15	8	8.15
58	Describe Functional Requirements	20	8	8.15
59	List Non-Functional Requirements	15	8	8.15
60	Non-Functional Requirements Descr.	20	8	8.15
61	Observation Case Analysis	12	5	8.16
62	Technical Research	12	5	8.16
63	Questionnaire	12	5	8.16
64	Rich Picture Observation Case	8	3	8.17
65	Rich Picture Technical Research	8	3	8.17
66	Rich Picture Questionnaire	8	3	8.17
67	Class Event Table	25	13	8.18
68	Class Diagram	25	13	8.18
69	Classes description	10	5	8.18
70	Scenarios Development	14	8	8.19

71	Actors' description	12	5	8.19
72	Use case diagram	15	8	8.19
73	Prepare Workshop	15	8	8.21
74	Execute the Workshop	10	5	8.21
75	Skeleton Test Document	5	3	8.22
76	Test cases	30	13	8.22
77	Quality Plan Skeleton	20	13	8.23

Appendix b – Initial User case stories

User without a fitness membership

Anna is a single woman in her thirties who likes to exercise, particularly running outside and exercising with her own body weight, but she is also curious to experiment and try out new ways to exercise. However, sometimes she finds it hard to motivate herself to exercise and push herself to try out new things, because she's doing it by herself.

When Anna signs up on the fitness buddy app, she is first asked to answer a series of questions to identify her needs and wants in exercising and a potential fitness buddy. This e.g. includes her favorite way to exercise, how often she exercises, what level she is at and what she would prefer in a fitness buddy.

After the initial setup, Anna is ready to use the app where all the people that specifically match her criteria will pop up in her search/home page. Here she's able to click on any of these profiles to read more about their exercise habits (maybe exclude personal detail until you become "approved/friends"), and if she likes the profile, she can send a "fitness buddy request". If the other person accepts, she will then gain access to additional and more personal information about this person, and they will be able to contact each other to plan their exercises.
When they've done their initial introductions and planning, they can plot in their exercise schedule in the app, where the exercises will automatically be synced with their private calendars, they will receive reminders/notifications on upcoming or changed exercises etc.

Likewise, a simple overview (not containing private information) will be available for other people in the fitness buddy search, where they e.g. will be able to see that "two women are running in x area on Mondays and Thursdays at 17:00, intermediate runners ~7 km at a 5.10 pace", and then request to join as a fitness buddy.

Gym manager user story

As a gym manager you can’t help but notice that about 75% of the gym goers work by themselves. The problem with this is that, since there is no social part of going to the gym, more people will lose interest and stop going to the gym. And after a while of not using the gym, they will cancel their membership, because it’s easy, and they won’t really lose anything because of it. They can always join the gym again or find

another gym that is more suitable for their needs. Another problem is that most people go to the gym at the same time because most people have the same work schedule throughout the week.

Being mashed up against an ocean of strangers while standing in line for the dumbbells you want to use, or a specific workout machine that is part of your workout, is also considered uncomfortable by most people and could also be the reason why some members would quit the gym and find other ways to exercise. If we had a tool or a method that could bring people together in pairs or in small groups for their workouts, the social aspect would make it more meaningful to go the gym, because members would lose a valuable social part of their life, if they quit the gym. It wouldn't be as frustrating standing in line for equipment, because you could chat with your friends while waiting, and other social benefits like doing other stuff with the people you work out with etc.

Most gyms offer group training sessions, but it is difficult to cover every member's interest with these sessions, and it typically also cost more to participate in these group workouts, which most members would like to avoid. So, a tool that could bring people together for a workout without the involvement of the gym, would make it easier for us to hold on to our members, and happy gym goers would also increase the chances of enticing other people to join a gym. Therefore, such a tool, which could be an app or a website, would be very helpful for present and future gym members, but also for the gym itself, maximizing its number of members.

User focusing in social aspects

As a non-member of any fitness club, I want to find people that enjoy long walks as a form of a non-intense physical activity but rather just maintaining a healthy life cycle while socializing a bit.

User without prior exercising experience

As a person who does not have the habit of doing any organized exercise on a weekly basis, I would like to find a community where I can start doing exercise together with other people. I know that I lose motivation if I try to get started exercising alone. Therefore, I would like a calendar function in the app where I can enter my schedule and preferred type of exercise, so that I can match with other people who have the same preferences as I do. I want to create a network of people, with whom I can commit to getting my weekly exercise done in a fun and casual way.

Steps

1. Implement a profile page, where users can specify their preferred type of exercise and preferred time of week/day for exercising. (High priority, core feature)
2. Implement a calendar function, where users can see other people who have the same preferences for exercise as themselves. (High priority, core feature)
3. Implement a feature that suggests types of exercise based on users' preferences (Low priority, optional feature)

If the user is already a member of the fitness center.

Imagine a person that goes to the gym almost every day and tries to be focused only on the exercises. Many people that sometimes only do that start to lose their motivation if they don't create personal boundaries inside the gym or go with friends to practice the activities. The fitness buddy is the perfect way to cover that problem, combining both focused exercises with social engagement. Besides meeting new people, you can share your progress and knowledge about the exercises as well as learn more and different things with your buddy! Also, if you want even to train harder you can contact professionals such as personal trainers to increase even harder you work.

Even meeting new people, sharing knowledge about the exercises, and having fun at the same time, the person that already is a member of a fitness center wants to try a different sport or activity. Yes! Fitness Buddy also helps with that. Based on your profile, the application can give you the best options to you discover a new activity and link you with people that want the same thing and with a high probability to match the person's taste for sports or mental health practices.

Therefore, that person that was losing the motivation to go alone to the gym can increase it by meeting new people, improving their abilities, and discovering new hobbies! Furthermore, there are several articles proving that training or doing exercises or activities together increases willpower and the results. Thus, this is the key point for people to already go to the gym: increase their performance, meet new people and discover new things. Now imagine that same person initially, with the app he or she will be probably happier to go to an activity because it is now also more of a social thing that helps you in different ways including frequency and performance.