

# UNIVERSITETET I TROMSØ

Ifi – Institutt for informatikk - UiT

INF2900 Software engineering  
Spring 2023



**UiT** Norges  
arktiske universitet

*Project Document. Skooba*

Magnus Dahl-Hansen, Eivind Østlyngen, Fedor Semaev, Jørgen Løken and  
Sigurd Uhre

# Project Document. Skooba

Magnus Dahl-Hansen, Eivind Østlyngen, Fedor Semaev, Jørgen Løken and Sigurd Uhre

May 19, 2023

# Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Product backlog</b>                          | <b>4</b> |
| 1.1      | Introduction . . . . .                          | 4        |
| 1.2      | Format and structure . . . . .                  | 4        |
| 1.3      | User stories . . . . .                          | 6        |
| <b>2</b> | <b>plans and progress at various stages</b>     | <b>7</b> |
| <b>3</b> | <b>Testing and results</b>                      | <b>7</b> |
| 3.1      | Unit testing . . . . .                          | 7        |
| 3.2      | Integration tests . . . . .                     | 8        |
| 3.3      | System tests . . . . .                          | 8        |
| 3.4      | Results . . . . .                               | 8        |
| 3.5      | Testing automation . . . . .                    | 8        |
| <b>4</b> | <b>Notes on meetings and retrospectives</b>     | <b>9</b> |
| 4.1      | Project Backlog . . . . .                       | 9        |
| 4.1.1    | Sprint 1 Backlog (16.jan - 23.jan) . . . . .    | 9        |
| 4.1.2    | Sprint 2 Backlog (23.jan - 06.feb) . . . . .    | 9        |
| 4.1.3    | Sprint 3 Backlog (06.feb - 13.feb) . . . . .    | 10       |
| 4.1.4    | Sprint 4 Backlog (13.feb - 27.feb) . . . . .    | 10       |
| 4.1.5    | Sprint 5 Backlog (6.mars - 20.mars) . . . . .   | 11       |
| 4.1.6    | Sprint 6 Backlog (20.mars - 27.mars) . . . . .  | 11       |
| 4.1.7    | Sprint 7 Backlog (27.Mars - 10.April) . . . . . | 12       |
| 4.1.8    | Sprint 8 Backlog (17.April - 01.May) . . . . .  | 12       |
| 4.1.9    | Sprint 9 Backlog (01.May - 08.May) . . . . .    | 12       |
| 4.1.10   | Sprint 10 Backlog (08.May - 19.May) . . . . .   | 13       |
| 4.2      | Remaining project backlog . . . . .             | 13       |

# 1 Product backlog

## 1.1 Introduction

The product backlog that follows is represented as list of features and enhancements which are created and implemented during the development of the product. Each feature and enhancement is derived from a roadmap. The purpose of a roadmap is to provide a clear vision of the product that is to be developed and to create a flexible plan of action to achieve the vision. The roadmap is divided into epics or milestones which are going to be achieved under the duration of the project. Due to the development team being new to the concept of agile development and SCRUM at the beginning of this project the roadmap were only partially completed before starting the development.

The following roadmap were created at the beginning of the project:

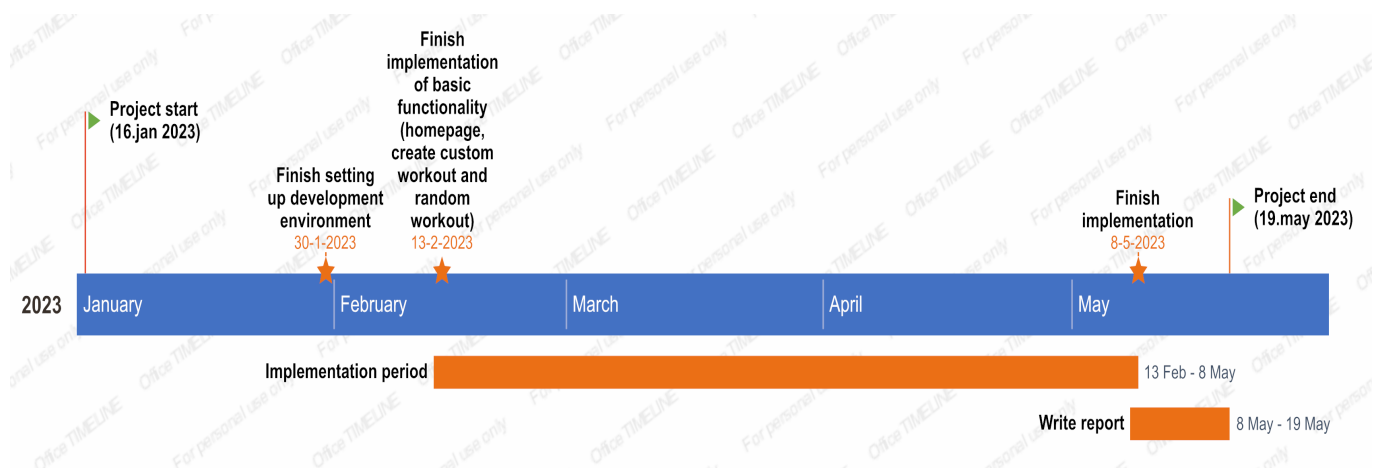


Figure 1: Roadmap

## 1.2 Format and structure

The product backlog is organized as a table giving an explanation of who wants to use each feature, what the feature is and why it should exist, how important each feature is to the overall vision of the product, the number of story points needed to implement each feature and the current status of each feature. The features and enhancements which the development team has rated as most important appears at the top of the table. The criteria for rating a feature as important is how much it contributes to the main vision and functionality of the product.

| Product backlog |  |  |          |             |
|-----------------|--|--|----------|-------------|
| As a...         | I want to be able to...                  | So that...   | Priority | Status      |
| Visitor/user    | Create a custom workout                  | I can easily create and execute a workout of my choice   | High     | Done        |
| Visitor/user    | Create a random workout                  | I can get a good and effective workout without having to plan and think about what exercises to do   | High     | Done        |
| Visitor/user    | Create a partially random workout        | I can decide which body parts I want to train, but get new ideas for exercises to perform  | High     | Done        |
| Visitor/user    | Get assistance with the training session | I do not have to think or forget about how long I should rest between sets and to be reminded of which exercises and number of repetitions I should do | High     | Done        |
| User            | Register as a new user                   | I am able to log my training by saving and deleting workouts and be able to replay workouts  | High     | Done        |
| User            | Log in                                   | I am able to inspect, save, replay and delete training sessions  | High     | Done        |
| User            | Change password                          | I am able to change password in case the current one is too weak or other  | High     | In progress |
| Visitor/user    | See information about Exercises          | I am able to get inspiration of which exercises exist and how they should be performed correctly   | High     | Done        |
| Visitor/user    | Edit a given workout                     | I can remove exercises or adjust the number of repetitions/sets in case I am not satisfied with the workout  | High     | Done        |
| Visitor/user    | Quit the workout midway                  | I am able to safely end and save the workout in case I have to attend to anything more important   | Medium   | Done        |
| Visitor/user    | End the rest time between sets early     | I can start the next sets when I feel like it  | Medium   | Done        |
| User            | Save my workout                          | I can replay the workout or inspect it at a later time   | Medium   | Done        |
| User            | Replay my workout                        | I can get better and achieve my goals  | Medium   | Done        |
| User            | Delete my workout                        | I can remove a workout I know was bad or will not repeat again   | Medium   | Done        |
| User            | See my personal info                     | I am able to see the progression of my training and potentially gain motivation or feedback  | Low      | In progress |
| User            | Rename my workout                        | I can get a more clear view of my workouts   | Low      | Done        |

Figure 2: Product backlog

## 1.3 User stories

User stories is an important part of a product backlog. A user story is created to help the development team understand the requirements, needs and goals from the perspective of a user and to describe how a feature should work.

### **Create a custom workout**

As a fitness enthusiast, I want to be able to create a workout which is tailored to my specific needs and goals. When I open the app I expect to be provided with a button which provides me with a functionality to select workout exercises, set the number of repetitions and sets and choose the rest time between sets.

### **Create a random workout**

As someone who avoids difficult and advanced training applications I am in need of a simple training application which can provide me with a functionality to generate a completely random workout. I want this functionality so that I can challenge myself, try new exercises and train effectively without having to invest a lot of time into planning a workout session and reading about how to train. When I use the application I want to be able to decide the duration of the training session and choose the type of training I want to do, such as cardio or hypertrophy.

### **Create a partially random workout**

As a well experienced athlete I want to create a partially random workout so that I can work out in different ways and challenge myself. I want to be able to decide what muscle groups to train, what type of training the workout should be and its duration, but not which exercises to do.

### **Re-create workout**

As a user of the training application I want to be able to re-create a given random or partially random workout if I am not satisfied with the first one or if there are some exercises I can not do. After I have generated a random/partially random workout I want to be presented with a "re-create workout" button which will generate a new workout based on the same criteria I made during the first try.

### **Pre-view of the workout**

As a user of the workout application I want to be able to preview the exercises of a workout before starting the workout. I want to do this so that I can familiarize myself with the workout and its exercises while also prepare myself mentally and physically. When I either have created the workout myself or been given a random workout, I want the preview to display each exercise, the number of repetitions for each exercise, the number of sets and how much rest I will get between each set.

### **End the rest time between sets early**

As an athlete and daily user of the training application I sometimes feel like the resting time between each set lasts too long. I want an option during the simulation of the workout where I can end the resting time earlier if I feel like it. I want a button to appear when the simulation tells me to rest which allows me to skip the resting time and start the next exercise early.

### **Get assistance with the training session**

As someone who does not train very often I am in need of someone to help me execute a training session to keep me motivated, focused and makes sure I perform all exercises. After getting a workout plan I want to be presented with an option which simulates the workout telling me what exercise I should do, the number of reps, the current set I am on, what exercise comes next and when I am finished with the workout. I also want to simulation to tell me when I should rest and when I should

start the next set (keep track of the rest time in some sort).

### **Register as a new user**

As a daily user of the training application I want to be able to create an account so that I can start to access all the features provided by the application. When I open the application I want to be guided to a register option which allows me to register with a unique username followed by a password so that only I can access my account.

### **Log in**

As a user I want to be able to log in to the application using my username and password so that I can save, replay, delete or rename the workouts I have created, see information about myself and log my training sessions. When I use the applications I want to be provided with a log in button which redirects me to the log in page. When I enter the correct log in credentials I want to be redirected to the main home page of the application and gain access to all the applications functionality.

### **Change password**

As a registered user of the training application I want to be able to change my password to enhance my accounts security. When I am logged in I want to be able to navigate to a "change password" feature via a settings feature because this is where one would usually locate such a functionality.

### **See information about exercises**

As a personal trainer I want to be able to access information about all exercises the training application currently has in its database. I want to do this because I want to see how any exercise should be performed correctly and the benefits of each exercise. I want the application to provide easy access to the exercise bank so that I can easily inspect exercises.

### **Quit workout midway**

As a medical personnel and a user of the application I often find myself having to quit a workout prematurely because of work. I therefore want the app to provide me with a functionality to quit a workout midway and resume it at a later time. I want this feature to be available during the simulation process of the workout so that I can easily quit the workout in a safe way and save the workout if I so please.

## **2 plans and progress at various stages**

## **3 Testing and results**

To test the entirety of this project we have written different kinds of tests at all levels, unit, Integration and system. This is done using multiple different tools.

### **3.1 Unit testing**

To test the base functionality of the models and serializers, without being impacted by other components, unit tests were written. This was especially a focus at the start of the project, including for standard models and serializers. After progressing further in the project, we encountered no issues with the behaviour of models using standard Django functionality, these tests also broke often as

a consequence of expanding our models as the project progressed. The focus of these tests were therefore shifted to focus on other parts of the project.

The current focus of our unit testing shifted on validating our implementations that overwrote the default Django functionality. The testing includes both their behaviour and it's effect on the database. Examples of such behaviour are hashing the password automatically of the User model before saving it to the database.

### **3.2 Integration tests**

As a large part of our project is an API serving data to the website, testing the endpoints using Integration tests were the main focus of our testing. The endpoints uses all models and serializers and are a great way to test the interaction between them. The focus of these test were to test that the endpoints returned the expected result body and HTTP status codes given input, such that it could easily be called and used in in the frontend of our application. These tests were implemented using APITestCase which is a test class integrated in the Django REST framework.

All currently used endpoints in the website are tested to a varying degree depending on their complexity and uses. Examples of implemented tests are using the endpoints with both incorrect and correct input. Additional rules on endpoints are also tested, such as registering a user with an already existing username or generating a random workout with an insufficient duration.

### **3.3 System tests**

System testing, also called e2e tests, involves testing the application as a whole including the django backend, react frontend and the communication between them. We have implemented these tests using cypress(cite?) which can emulate user behaviour by simulating a browser interacting with the site. Most interactions uses the real server to ensure correct behaviour, but API calls which manipulates the database are simulated/stubbed to reduce the complexity of the tests. We consider this ok, as the functionality of the API endpoints are tested in depth at the integration tests. As the tests are implemented with a new tool the implementation of these tests can be very time-consuming to implement, especially if we want to cover all cases. Therefore the tests are more limited to assert that the main functionality of the site is working as intended when making changes, and more complex edge cases while interacting with the site are covered by the integration tests.

### **3.4 Results**

Currently there are 35 tests implemented in total covering all levels, all tests pass.

### **3.5 Testing automation**

-



## 4 Notes on meetings and retrospectives

### 4.1 Project Backlog

- Login
- Homepage
- Register
- Info
- Settings
- My workout
- Random workout
- Custom workout
- Saved workouts
- My progression
- Tests
- Report and document project
- (Mobile App)

#### 4.1.1 Sprint 1 Backlog (16.jan - 23.jan)

Familiarize yourself with Django [All team members]

Familiarize yourself with React [All team members]

Set up project git repository [Eivind]

#### 4.1.2 Sprint 2 Backlog (23.jan - 06.feb)

- Fedor [4 points]
  - Homepage
  - Homepage endpoint infrastructure
  - React-router dome. Route between URLs.
- Jørgen and Magnus [8 points]
  - Login
  - Username and passwords to be sent to the DB
  - Create DB-model that stores username and password. (Unique username)
  - API-endpoint. To validate the existence of user name and corresponding password
  - Validation of login.

- Sigurd og Eivind [6 points]
  - Set up and populate backend exercise DB
  - create DB-modell
  - API-endpoint (Test to to extract data from DB)

#### 4.1.3 Sprint 3 Backlog (06.feb - 13.feb)

- Magnus [4 points]
  - Start implementation of random workout
  - Set up endpoint buttons
  - Create random workout with exercises from database
  - Set up infrastructure for save workout
- Eivind and Fedor [6 points]
  - Create and design all databases
  - Populate databases
- Jørgen [4 points]
  - Add test for login and register
  - Fix hide-password as user type in login and register
- Sigurd [4 points]
  - MyPage implementation
  - Change username and password
  - If no user registered: "Register user" button (redirect to register user page).

#### 4.1.4 Sprint 4 Backlog (13.feb - 27.feb)

- Magnus [4 points]
  - "Random workout" - Select workout type, part of body, duration,
  - Create table with exercises to present to simulation (preview)
  - Save workout
- Eivind [4 points]
  - Create database with information about each exercise
  - Create a web-page that displays the info about each exercise
- Sigurd [3 points]
  - Create an overview of functionality design
  - Front-end adjustment
- Jørgen [8 points]
  - Implement custom workout
  - Create table with chosen exercises depending on chosen muscle group

- Fedor [8 points]
  - Start the implementation of "time" database for all workouts.
  - Front-end adjustment
  - Create table with chosen exercises depending on chosen muscle group

#### **4.1.5 Sprint 5 Backlog (6.mars - 20.mars)**

- Feder [6 points]
  - Complete random-function. Add reps and sets i DB.
  - Join other tasks when finished.
- Jørgen [6 points]
  - Continue with custom workout. Post all chosen exercises to the workout DB.
- Magnus [4 points]
  - Save workout endpoint. (Set name and save).
- Eivind og Sigurd [6 points]
  - Implementing simulation

#### **4.1.6 Sprint 6 Backlog (20.mars - 27.mars)**

- Fedor [6 points]
  - Complete random workout
- Mangnus [4 points]
  - Fix seved workout after the DB is fixed
- Eivind [4 points]
  - Fix the DB to link custom workout to simulation
- Jørgen [4 points]
  - Fix POST-call to DB with custom workout. Pending on the DB to be fixed
- Sigurd [4 points]
  - Continue implementation of the simulation
- All team members
  - Testing
  - Clean code, review code
  - Fix errors

#### **4.1.7 Sprint 7 Backlog (27.Mars - 10.April)**

- Fedor [5 points]  
Fix present/custom/random workout to the new DB
- Magnus [8 points]  
Fix present/custom/random workout to the new DB  
Delete workout if "save" is not selected  
Find a way to create "username already exist"
- Jørgen [8 points]  
Fix present/custom/random workout to the new DB
- Sigurd [8 points]  
Extend the database with enough exercise  
Extend the database with enough exercise info  
Css - login
- Eivind [8 points]  
Permission check: Check if the correct user is in present/workout simulation.

#### **4.1.8 Sprint 8 Backlog (17.April - 01.May)**

- Sigurd [6 points]  
CSS - Application layout - Logo
- Eivind [8 Points]  
Create tests  
Fix edge cases
- Magnus [4 points]  
Select/modify/delete existing workouts
- Fedor [3 Points]  
Fix edge cases with "random workout"
- Jøregn [6 Points]  
Style custom workout with CSS

#### **4.1.9 Sprint 9 Backlog (01.May - 08.May)**

- Sigurd [8 Points]  
Create report layout  
Start writing report

- Eivind [8 Points]
  - Back-end Test
  - Hash passwords - Security
  - Clean code directives - remove unused files
- Jøregn [6 Points]
  - Style app with CSS
- Fedor [6 Points]
  - Improve existing code and make it more readable
- Magnus [6 Points]
  - Style app with CSS

#### **4.1.10 Sprint 10 Backlog (08.May - 19.May)**

- All team members [6 points each]
  - Report and project document

## **4.2 Remaining project backlog**

- My progression
- Settings
- (Mobile App)