DESK WORKOUT

Software Requirements Specification

Version 1.0
Status: 1.0

Prepared by Penka Yochkova / Developer, Project Manager

Dida Negasa / UX/UI designer, QA

Nikoleta Stoeva / Analyst, Software Architect

27.02.2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version	Date of Approval
Penka Yochkova, Dida Negasa, Nikoleta Stoeva	27.02.2022	Create main part of specification	1.0	27.02.2022

1. Introduction

1.1 Purpose

To develop a mobile application which will help users to improve their lifestyle and health in time of COVID-19. The application offers different types of exercises and possibility to create custom workouts. The main functionality of it is the possibility to create notifications to remind the user for desk workouts.

1.2 Document Conventions

Each change in document is necessary to be bold and be marked in yellow color. These are some of abbreviations which are used:

OPM	Online Project Management
Admin	Administrator
PM	Project Manager.
QA	Quality Assurance
SA	Software Architect
DB	DB Database Management System
Msg	Message
Doc	Document
BA	Business Analyst
AWS	Amazon Web Services
OS	Operating system

1.3 Intended Audience and Reading Suggestions

ID	Stakeholder	Description
S-1	Customer	Checking correspondence of business goals and functionality requirements to the expectations from implementing the product.
S-2	Development team	Forming the accurate vision of the project, detailed functional and nonfunctional requirements.
S-3	QA team	Making test-plans and test-cases.
S-4	PM/BA	Estimating the quote of the project, planning resources and the timeline of work.
S-5	User	Based on this document the Terms of Service and the Privacy Policy are created.

1.4 Product Scope

- Login / Register in application for free or with paid accounts.
- Create different users with different scopes depending on users' plan (paid or free).
- Users can create custom workouts from exercises in application.
- Create reminders using application notifications.
- Provide a variety of articles with benefits of desk workouts.
- Provide statistics for user workouts number in time (day, month and year).
- Update members about completed sets of workouts - send notification for month workout number.
- Paid plan provides more statistics, exercises and articles.

1.5 References

- IEEE SRS Format.
- TGMC-2008 Sample Synopsis Format.
- Problem Definition Provided By TGMC-2008

2. Overall Description

2.1 Product Perspective

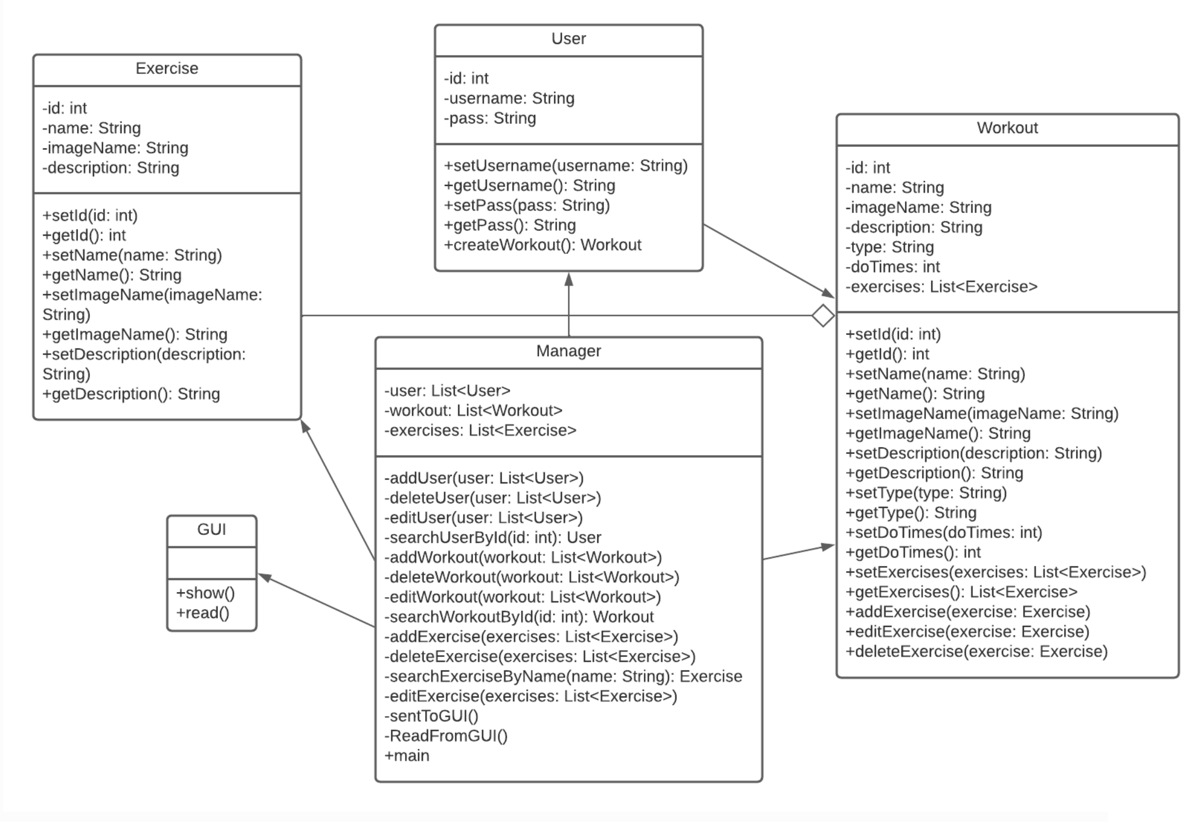
- Supports different versions of Android OS: Offers operating support of older versions of Android OS and the last version of it.
- User account: The application allows the user to create their accounts in the system and provide features of updating and viewing profiles.
- Number of users being supported: The application is able to support a large number of online users at a time.
- Using an external API for getting a variety of health and lifestyle articles.
- Using Chart Android API for creation of statistics.
- Using Amazon Simple Notification Service - send push notifications to users for month statistics.

2.2 Product Functions

The major functions which the product performs are:

- Creation of reminder notifications by user.
- Inform the user for month statistics data using push notifications.
- Custom Workout description: It includes the set of exercises chosen by the users.

Object Class Diagram:


User
-id: int
-username: String
-pass: String
+setUsername(username: String)
+getUsername(): String
+setPass(pass: String)
+getPass(): String
+createWorkout(): Workout
Exercise
-id: int
-name: String
-imageName: String
-description: String
+setId(id: int)
+getId(): int
+setName(name: String)
+getName(): String
+setImageName(imageName: String)
+getImageName(): String
+setDescription(description: String)
+getDescription(): String
Workout
-id: int
-name: String
-imageName: String
-description: String
-type: String
-doTimes: int
-exercises: List<Exercise>
+setId(id: int)
+getId(): int
+setName(name: String)
+getName(): String
+setImageName(imageName: String)
+getImageName(): String
+setDescription(description: String)
+getDescription(): String
+setType(type: String)
+getType(): String
+setDoTimes(doTimes: int)
+getDoTimes(): int
+setExercises(exercises: List<Exercise>)
+getExercises(): List<Exercise>
+addExercise(exercise: Exercise)
+editExercise(exercise: Exercise)
+deleteExercise(exercise: Exercise)
Manager
-user: List<User>
-workout: List<Workout>
-exercises: List<Exercise>
-addUser(user: List<User>)
-deleteUser(user: List<User>)
-editUser(user: List<User>)
-searchUserById(id: int): User
-addWorkout(workout: List<Workout>)
-deleteWorkout(workout: List<Workout>)
-editWorkout(workout: List<Workout>)
-searchWorkoutById(id: int): Workout
-addExercise(exercises: List<Exercise>)
-deleteExercise(exercises: List<Exercise>)
-searchExerciseByName(name: String): Exercise
-editExercise(exercises: List<Exercise>)
-sentToGUI()
-ReadFromGUI()
+main
GUI
+show()
+read()

2.3 User Classes and Characteristics

ID	User classes	Description
U-1	Signed up user	A user who has completed the sign up on the app and the account activation.
U-2	Not signed up user	A user who has completed neither the sign up in the application, nor the account activation. He owns limited rights inside the app.
U-3	Paid user	Paid plan provides more statistics, exercises and articles.

2.4 Operating Environment

The mobile application is developed for Android OS. Android is a Linux based mobile operating system. The hardware which is necessary for running the application is a mobile phone with Android OS.

2.5 Design and Implementation Constraints

Items or issues that will limit the options available to the developers:

- REST services will be used for user authentication (login or registration).
- AWS push notifications.
- Mobile phones should be able to receive push notifications and they should be allowed for the application.
- Android studio and new versions of Android should be supported and used by developers.
- Standard data exchange format is JSON. JSON format will be used for receiving data from the backend.
- Kotlin is the language for front-end development and Java is the language for backend development. Database development language is PL/SQL.
- HTTP communication protocol is used.
- Design conventions and programming standards are defined by our company.
- The application should follow rules in Google Play store otherwise it cannot be uploaded there.
- Security considerations: users' passwords will be hashed in the database, they won't be saved as simple text.

2.6 User Documentation

The user documentation that will be delivered is a video tutorial and images of all functionalities. The video tutorial will be available in Google Play store also.

2.7 Assumptions and Dependencies

Technical requirement "Push notifications" will work on the latest operating system and the smallest one is " Android 5.0 Lollipop".

Our application has a paid plan and when the user wants it, he/she should pay for it. It's necessary to protect the selected payment method and user's personal data.

We can develop the product in the proposed time frame.

3. External Interface Requirements

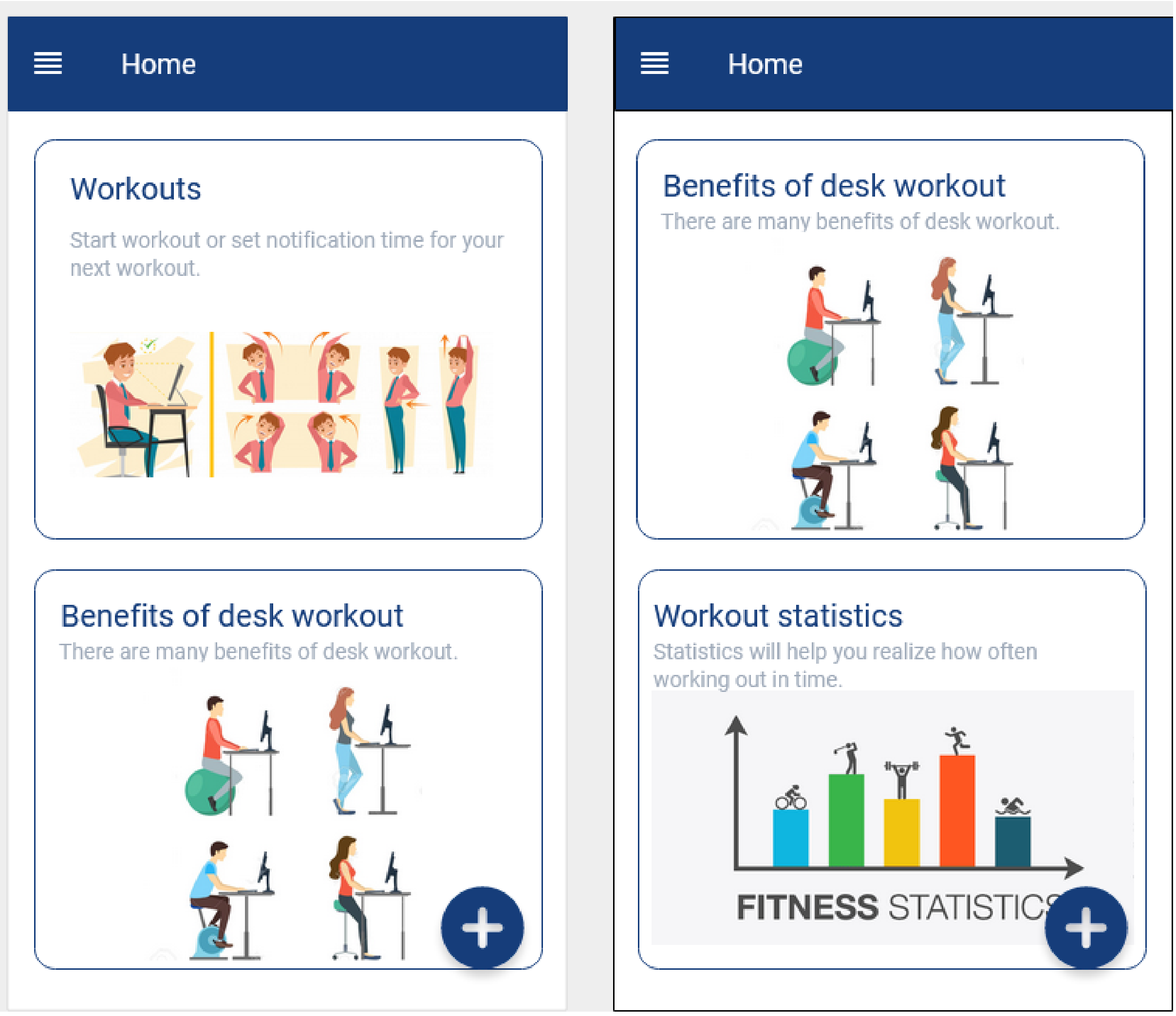
3.1 User Interfaces

When the user opens the app he/she accesses the Login screen but if he/she is not registered, it's necessary to make it. After the user is logged in, he/she will see the Home page where there are default workouts. The user can create a custom workout from the fab button in the right corner(big plus button). The notification can be set from the Home screen also. ***Workout benefits*** and ***Workout charts*** data are the other two functionalities on this screen.

Side menu screen is another important screen because the user can sign out from there. Other functionalities are Feedback and Settings. From this screen the user can open the Home screen again. Side screen is the navigation screen in the application.

Our user interface is made in https://app.moqups.com. We have a prototype realized with a local database in Kotlin. These are some of screens but all of them can be found here: *https://app.moqups.com/83zRcpi4JtEezHeE25zjCVQQU6YQvSXv/view/page/a820c1a3b*

DeskWorkout Home Page



Home

Workouts

Start workout or set notification time for your next workout.

Benefits of desk workout

There are many benefits of desk workout.

Home

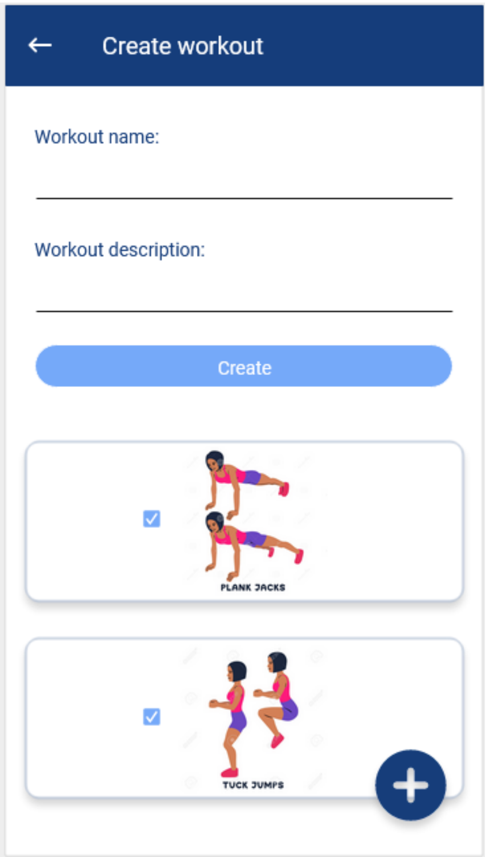
Benefits of desk workout

There are many benefits of desk workout.

Workout statistics

Statistics will help you realize how often working out in time.

DeskWorkout Create Workout Page:

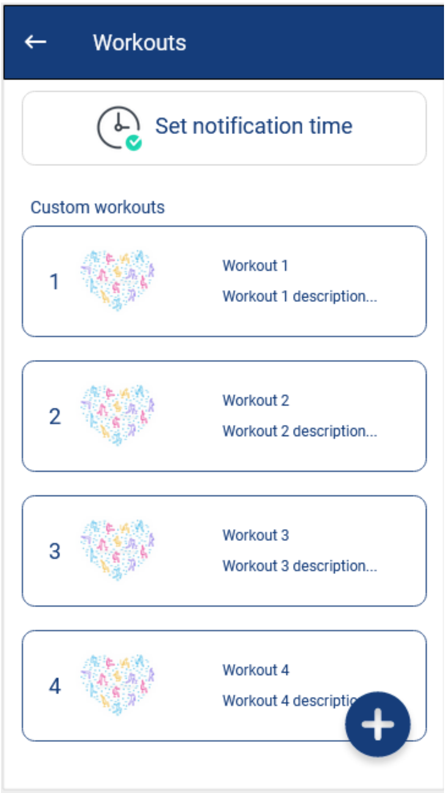


Create workout

Workout name:

Workout description:

DeskWorkout Workouts Page:



Workouts

Custom workouts

1 Workout 1
Workout 1 description...

2 Workout 2
Workout 2 description...

3 Workout 3
Workout 3 description...

4 Workout 4
Workout 4 descriptio

3.2 Hardware Interfaces

There are no restrictions about device models. It’s important for the device to support Android OS and the minimum Android is “Android 5.0 Lollipop”. The device is necessary to have Google services or Huawei services because then the user can download the app from Google play store or AppGallery.

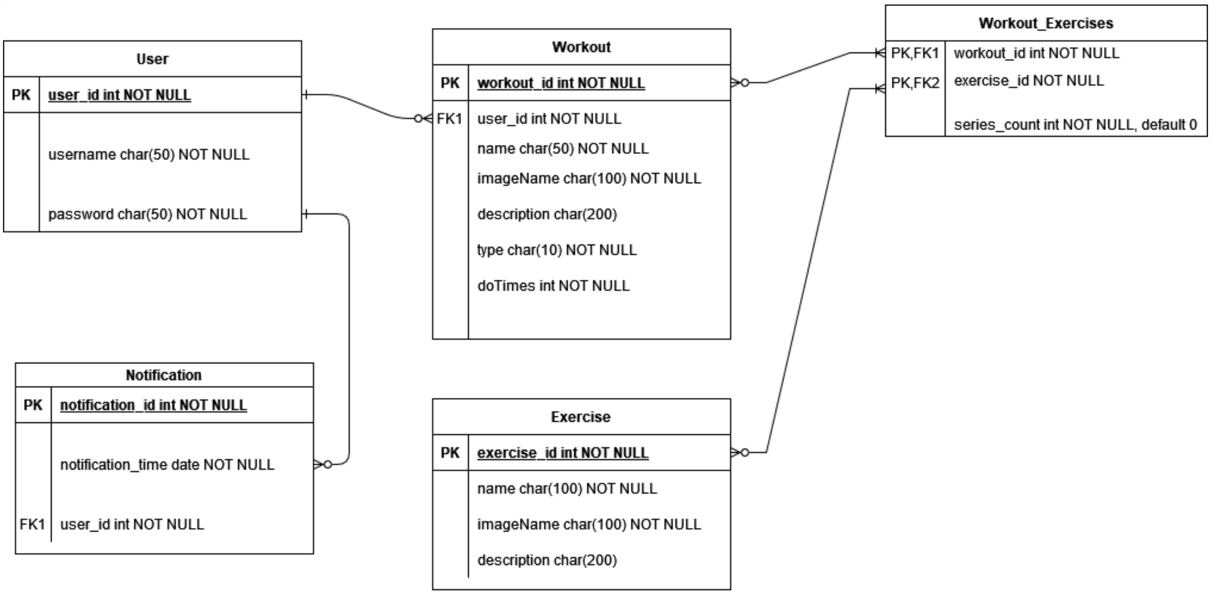
3.3 Software Interfaces

The application is uploaded in Google play store and AppGallery. PL/SQL developer is our database tool and Eclipse is the IDE which we use for our Java backend project. Android studio is the IDE for our front-end project.

3.4 Communications Interfaces

We use communication protocols HTTP, SSL, etc. It’s necessary for the device to have an wireless internet connection.

4. Domain Model



5. System Features (Use Cases)

5.1 Use Case: Create Custom Workout

Brief Description:	The user will create a new Custom Workout containing a single exercise or a set of exercises by his/her choosing.
Business Trigger:	The business event is the user's desire to work out and create a custom workout.
Preconditions:	The user must be logged in the application.

Basic Flow:
The user successfully logs in the application. He/She navigates through the menu and reaches the button for creating a new Custom Workout. A new screen loads up and prompts the user to select at least one exercise from the list of available exercises and choose how much time that specific exercise will last. If the user wishes to add another exercise, there will be an option for that. When the user is satisfied with his/her set of exercises, he/she will save the Custom Workout using the save button. The Custom Workout will have a generated name such as Custom Workout 1, Custom Workout 2 ect., and the user can rename it if he/she wishes to.

Assumptions: 001

Line	System Actor Action	System Response
1	The user wants to log in the application.	The system asks the user for identification.
2	The user identifies themselves to the system.	The system returns the user to the home screen.
3	The user navigates through the menu and selects the button for creating a new Custom Workout.	The system loads up a new screen and prompts the user to select at least one exercise from a list of available exercises and choose how much time that specific exercise will last.
4	The user saves the Custom Workout using the save button.	The system returns the user to the page for creating a new Custom Workout.

Post Condition:	The system saves the Custom Workout in the database.

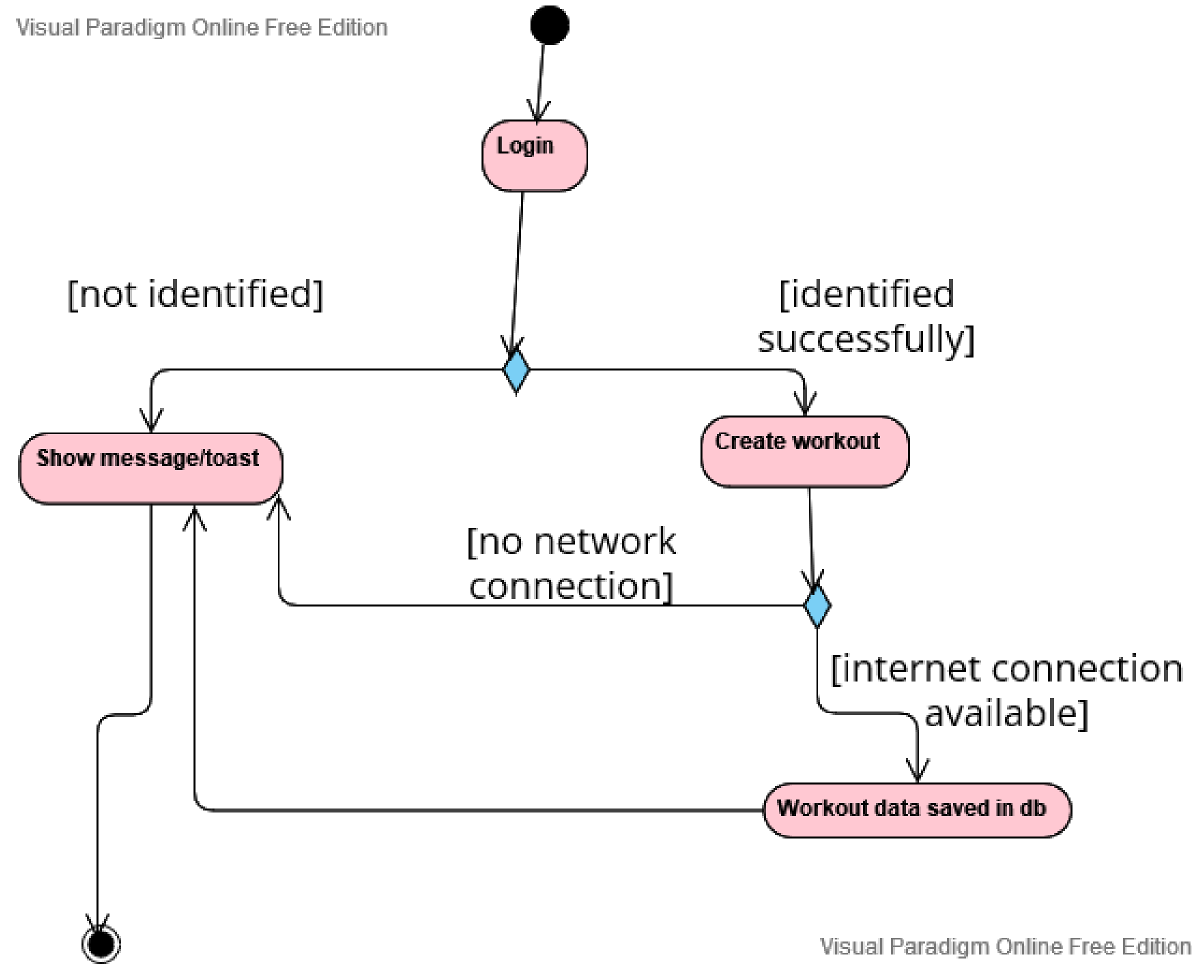
Alternate Flow (AFx): 0001 Failed login in the application

If at line 2 in Create Custom Workout the user fails to correctly identify themselves, the system will prompt the user to either reset their password or give them an option to create a new account.

Line	System Actor Action	System Response
1	The user fails to correctly identify themselves.	The system will prompt the user to either reset their password or give

		them an option to create a new account.
2		
The use case terminates / The use case restarts at line 1 in Create Custom Workout		
Post Condition:	The system returns the user to the login page.	

Activity Diagram:


Visual Paradigm Online Free Edition
Login
[not identified]
[identified successfully]
Show message/toast
Create workout
[no network connection]
[internet connection available]
Workout data saved in db
Visual Paradigm Online Free Edition

Screen Entry Exception Table:		
Field	**Constraint**	**Response**
1 Workout name	50 chars max	Message: “Exceeds 50 chars max – please re-enter”
2 Workout description	200 chars max	Message: “Exceeds 200 chars max – please re-enter”

5.2 **Use Case:** Create Workout Notification

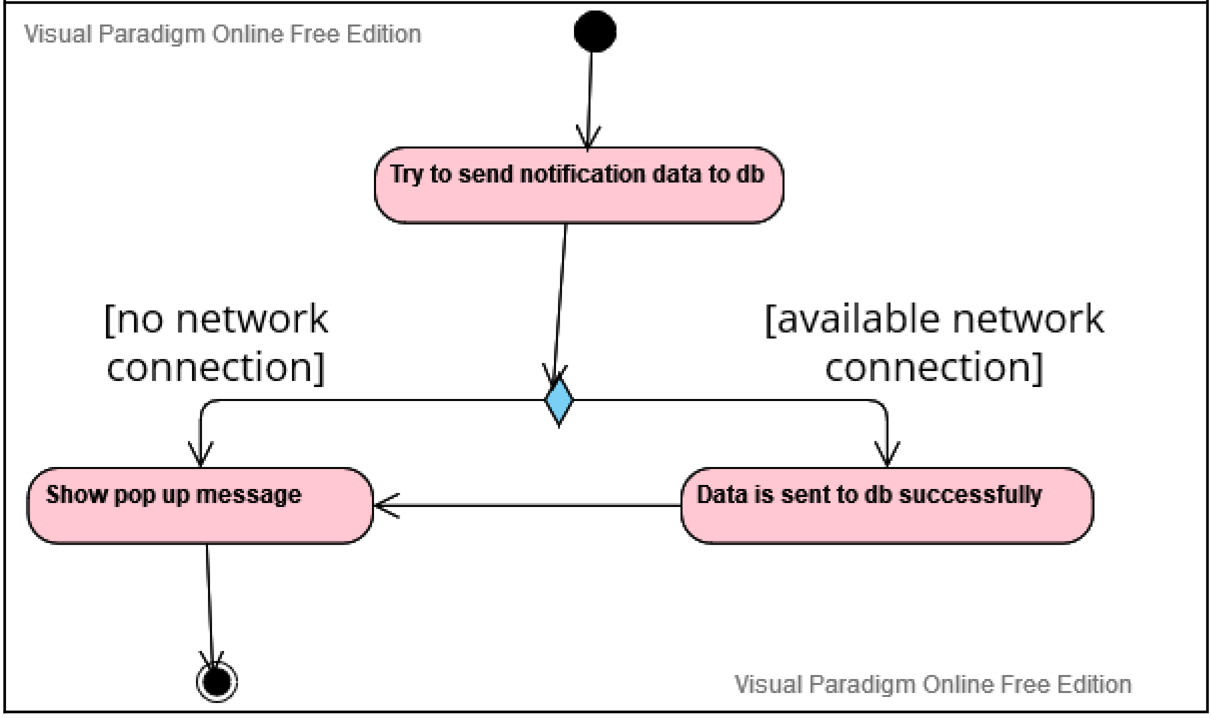
Brief Description:	The user will create a workout notification when he/she wants to start his/her workout.
Business Trigger:	The business event is that the user has the possibility to create notifications like reminders and complete the workouts in app.
Preconditions:	The user must be logged in the application.

Basic Flow: The user successfully logs in the application. He/She navigates through the menu and reaches the button for creating a new workout notification. The user clicks the button "Set notification time". A time dialog will appear where the user can select the notification time. After the notification time is set, a toast should appear on the bottom of the screen. This is the sign that the notification is set successfully. After the notification is set, we should check if the notification will be received in time. If the notification is received when it is set, this part of functionality works correctly.		
Assumptions: 002		
Line	**System Actor Action**	**System Response**
1	The user wants to log in the application.	The system asks the user for identification.
2	The user identifies themselves to the system.	The system returns the user to the home screen.
3	The user navigates through the menu and selects the button "Set notification time" for creating a workout notification.	A time dialog will appear where the user can select the notification time.
4	The user saves the workout notification using the "Done" button.	The system returns the user to the page with all workouts and shows toast on the bottom of the screen with the message "Notification is set successfully".
Post Condition:	The system saves the workout time in the database and the user should receive the notification in that time.	

Alternate Flow (AFx): 0002 Failed to save notification time in the database		
If the user's network connection stops when he/she tries to set notification time then the notification data cannot be saved in the database and he/she shouldn't receive notification. It's necessary the application to show dialog with message "No internet connection" and also to indicate that the notification is not set with toast message.		
Line	**System Actor Action**	**System Response**
1	the user's network connection stops	the application to show dialog with message "Please, check your internet connection"

2	the notification data cannot be saved in the database	the app indicates that the notification is not set with toast message: “Notification is not set”
The use case terminates / The use case restarts at line 1 in basic flow.		
Post Condition:	0002 The user will receive a message to check his/her internet connection and the app returns to the workouts page.	

Activity Diagram:


Visual Paradigm Online Free Edition
Try to send notification data to db
[no network connection]
[available network connection]
Show pop up message
Data is sent to db successfully
Visual Paradigm Online Free Edition

6. Other Nonfunctional Requirements

6.1 Performance Requirements

The load of the initial screen shouldn’t take more than 3 seconds. The user data(caches, etc.) increases and the application should be capable of handling them without delay.

6.2 Security and safety Requirements

Our application has a paid plan and the user can pay for it. It's necessary to protect the selected payment method and user's personal data. The secure requirements which should be applied (which helps us to achieve a secure authentication and authorization):

1. We should use two-factor authentication with OAuth2 and JSON can be used for channel encryption.
2. We should eliminate leakage by encryption. Implement file-level encryption and avoid local storage of sensitive data - if that is essential, it should also be encrypted.

3. We should use encrypted connections for additional protection, such as VPN, SSL or TLS.

6.3 Software Quality Attributes

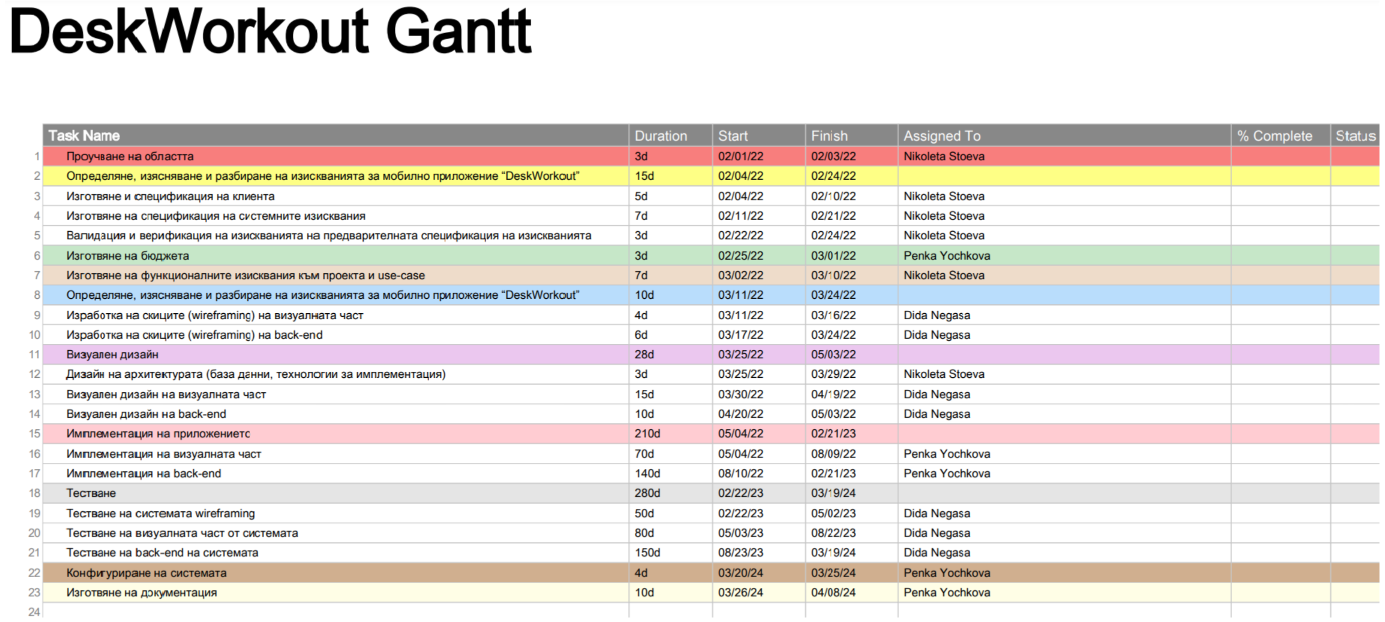
We should ensure that our company code is portable, updatable and open to improvements and repairs. Also, it’s important to have attributes: reusability, reliability, testability, and usability.

7. Other Requirements

For now we don’t have other requirements.

Appendix A: Additional diagrams

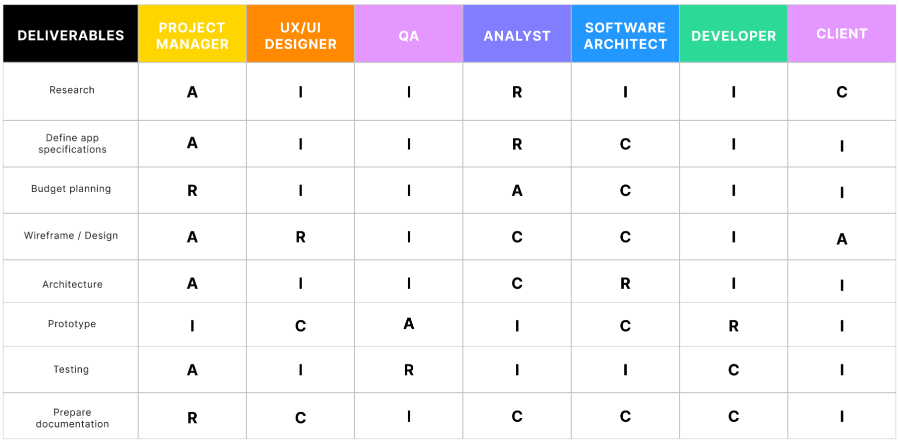
Figure 1: Gantt chart: https://app.smartsheet.com/sheets/cXH468F466C83qqFgqgGV6jxrc8vrRRR8jrHGGr1



DeskWorkout Gantt

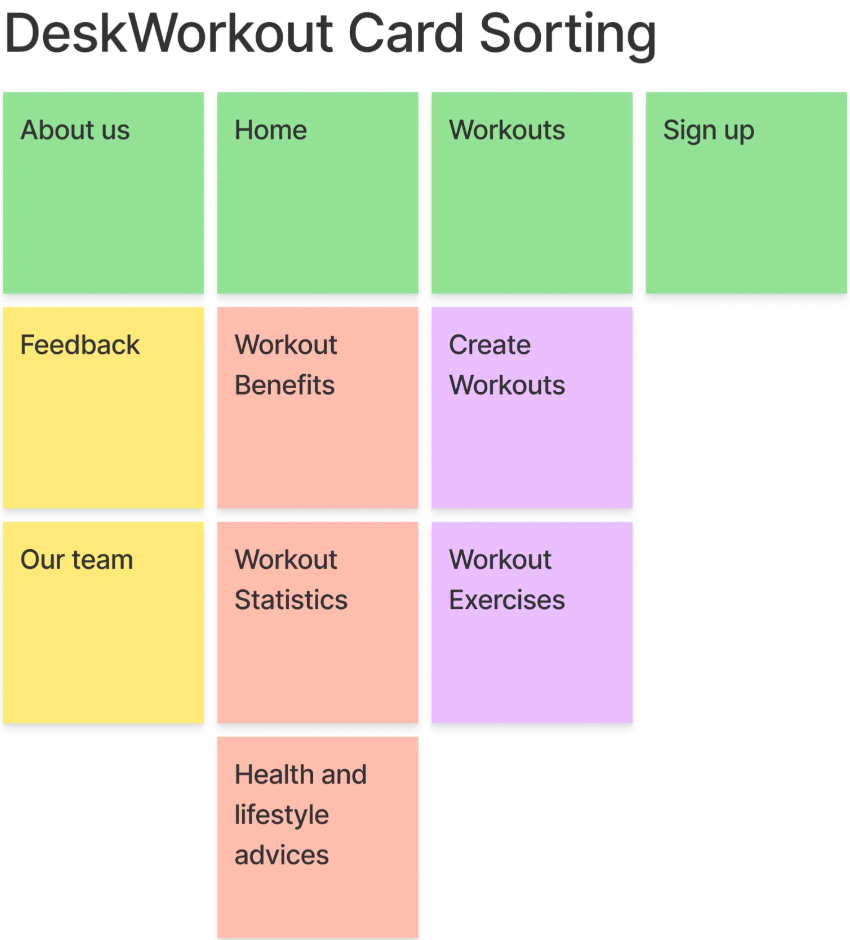
	Task Name	Duration	Start	Finish	Assigned To	% Complete	Status
1	Проучване на областта	3d	02/01/22	02/03/22	Nikoleta Stoeva		
2	Определяне, изясняване и разбиране на изискванията за мобилно приложение “DeskWorkout”	15d	02/04/22	02/24/22			
3	Изготвяне и спецификация на клиента	5d	02/04/22	02/10/22	Nikoleta Stoeva		
4	Изготвяне на спецификация на системните изисквания	7d	02/11/22	02/21/22	Nikoleta Stoeva		
5	Валидация и верификация на изискванията на предварителната спецификация на изискванията	3d	02/22/22	02/24/22	Nikoleta Stoeva		
6	Изготвяне на бюджета	3d	02/25/22	03/01/22	Penka Yochkova		
7	Изготвяне на функционалните изисквания към проекта и use-case	7d	03/02/22	03/10/22	Nikoleta Stoeva		
8	Определяне, изясняване и разбиране на изискванията за мобилно приложение “DeskWorkout”	10d	03/11/22	03/24/22			
9	Изработка на скиците (wireframing) на визуалната част	4d	03/11/22	03/16/22	Dida Negasa		
10	Изработка на скиците (wireframing) на back-end	6d	03/17/22	03/24/22	Dida Negasa		
11	Визуален дизайн	28d	03/25/22	05/03/22			
12	Дизайн на архитектурата (база данни, технологии за имплементация)	3d	03/25/22	03/29/22	Nikoleta Stoeva		
13	Визуален дизайн на визуалната част	15d	03/30/22	04/19/22	Dida Negasa		
14	Визуален дизайн на back-end	10d	04/20/22	05/03/22	Dida Negasa		
15	Имплементация на приложението	210d	05/04/22	02/21/23			
16	Имплементация на визуалната част	70d	05/04/22	08/09/22	Penka Yochkova		
17	Имплементация на back-end	140d	08/10/22	02/21/23	Penka Yochkova		
18	Тестване	280d	02/22/23	03/19/24			
19	Тестване на системата wireframing	50d	02/22/23	05/02/23	Dida Negasa		
20	Тестване на визуалната част от системата	80d	05/03/23	08/22/23	Dida Negasa		
21	Тестване на back-end на системата	150d	08/23/23	03/19/24	Dida Negasa		
22	Конфигуриране на системата	4d	03/20/24	03/25/24	Penka Yochkova		
23	Изготвяне на документация	10d	03/26/24	04/08/24	Penka Yochkova		
24							

Figure 2: DeskWorkout Raci matrix:



DELIVERABLES	PROJECT MANAGER	UX/UI DESIGNER	QA	ANALYST	SOFTWARE ARCHITECT	DEVELOPER	CLIENT
Research	A	I	I	R	I	I	C
Define app specifications	A	I	I	R	C	I	I
Budget planning	R	I	I	A	C	I	I
Wireframe / Design	A	R	I	C	C	I	A
Architecture	A	I	I	C	R	I	I
Prototype	I	C	A	I	C	R	I
Testing	A	I	R	I	I	C	I
Prepare documentation	R	C	I	C	C	C	I

<u>Figure 3: DeskWorkout Card sorting:</u>



DeskWorkout Card Sorting

About us

Home

Workouts

Sign up

Feedback

Workout Benefits

Create Workouts

Our team

Workout Statistics

Workout Exercises

Health and lifestyle advices