

YCS673 Software Engineering
Team 5 - Simple Health
Project Proposal and Planning



<u>Team Member</u>	<u>Role(s)</u>	<u>Signature</u>	<u>Date</u>
Seema Palora	Team Leader	Seema Palora	09/08/2021
Yang Ye	Requirement Leader, Security Leader	Yang Ye	09/08/2021
Jake Stephens	Backup Team Leader, QA Leader	Jake Stephens	09/08/2021
Sherperd Liu	Configuration Leader	Sherperd Liu	09/08/2021
Sandeep Agrawal	Design and Implementation Leader	Sandeep Agrawal	09/08/2021

Revision history

<u>Version</u>	<u>Author</u>	<u>Date</u>	<u>Change</u>
1.0	Jake Stephens, Seema Palora	9/9/2021	First additions, Updated Optional Features
2.0	Seema Palora, Yang Ye	09/23/21	Updated Timeline Update most of the user stories to match the <i>Pivotaltracker</i> . Still some of the user stories are in old status.

[Overview](#)

[Related Work](#)

[Proposed High level Requirements](#)

[Functional requirements:](#)

[Essential Features:](#)

- [Sign up](#)
- [Forget Password](#)
- [Splash Screen](#)
- [Medications](#)
- [Conditions](#)
- [Exercise](#)
- [Mind Games](#)

[Desirable Features:](#)

- [Video tutorial](#)
- [Goal setting](#)
- [Goal tracking](#)

[Optional Features:](#)

- [Integrate with real data](#)
- [Credential hashing](#)
- [Connecting with a fitness device](#)

[Nonfunctional Requirements:](#)

[Security requirements](#)

[Management Plan](#)

[Process Model](#)

[Objectives and Priorities](#)

[Risk Management \(need to be updated constantly\)](#)

[Monitoring and Controlling Tools and Mechanisms](#)

[Timeline \(need to be updated at the end of each iteration\) : Will update this section by start of Iter1](#)

[Quality Assurance Plan](#)

[Metrics](#)

[Inspection/Review Process](#)

[Every iteration, two days prior to the submission, the team will have a Review meeting to go through completed docs, code . Along with that for every code commit , a new pull request will be created and once reviewed only it will be merged to the main branch.](#)

[Testing:](#)

[Defect Management](#)

[Pivotal tracker defects will be opened upon bug discovery and then a team member will be assigned to the defect to resolve it. Defects with High priority should be resolved within 1 day. Defects with Medium or Low priority can be resolved within 3 days.](#)

[Configuration Management Plan](#)

[Configuration items and tools](#)

[Change management and branch management](#)

[Code commit guidelines](#)

[References](#)

[Glossary](#)

1. Overview

This app will be a healthcare related app, and will be advertised for its simplicity and how it will enhance the users life. It happens often where people forget to take their prescribed medications on time, this app will help with that problem by alerting users when it is time to take a medication. This app will also increase the happiness and wellbeing of the user by providing exercises, as well as mitigations to deal with symptoms they may be dealing with from a condition they have. The users of this app will range from young adults to seniors as nearly everyone can benefit from using this app.

2. Related Work

Wellbeats is a virtual fitness training application that helps its users stay fit. This app is similar to the exercise feature of our app but has a lot more features and customization as the apps entire purpose is exercise.

3. Proposed High level Requirements

Functional requirements:

Essential Features:

- **Title**
 - Login
- **Description**
 - As a user, I want to be able to login so that I can access the simple health application features.
- **Acceptance tests:**
 - Given a user is not logged in, when the user launches the app from the home screen, then the user will land on the signup screen and user will click on sign in tab to launch the sign in screen
 - Given the sign in tab is selected, when the user fills the username and password and click on sign in button then the user will be queried in the

database and if found and the credentials are valid, they will see a success message otherwise they will see an error message "Please check your credentials"

- Given the sign in tab is selected, when the user does not fill all the mandatory field and click on sign in button then the user will see an error message "Please enter all required fields"

- **Time Estimation:**

- Hours: 10-15h

- **Title**

- Sign up

- **Description**

- As a new user, I want to create an account so that I can access the simple health application features.

- **Acceptance tests:**

- Given a user is not registered, when the user launches the app from the home screen, then the user will land on a new screen with the Sign up tab selected by default.
- Given the signup tab is selected, when the user fills in all the mandatory fields such as name, username, email, dob, password, confirm password and click on signup button then the new user will be added to the Database, display the success/error if account creation is successful or failure.
- Given the signup tab is selected, when the user does not fill all of the required fields and click on signup button then the user will be displayed with an error message "Please complete all required fields"
- Given the signup tab is selected, when the user fills in all the mandatory fields but the password and confirm password field does not match then the user will be displayed with an error message "Password and confirm password does not match"
- Given the signup tab is selected, when the user fills in all the mandatory fields and clicks on the signup button and signup is successful, the user will be redirected to the home screen.

- **Time Estimation:**

- Hours: 10-15h

- **Title**

- Forget Password

- **Description**

- As a registered user, I want to reset the password when the current password is locked/forgotten, so the updated password can be used for the next login.
 - **Acceptance tests:**
 - Given a forgotten password link is available, when the user clicks on the link, a page will be displayed having the title "Forgot Password" to the user to enter the username, new password and confirm password.
 - Given the forgot password screen is available, when the user fills all mandatory fields(username, new password, confirm password) and click on submit button, backend system is updated with new password and display the success message
 - Given the forgotten password screen is available, when the user fills all mandatory fields(username, new password, confirm password) and clicks on submit button, the proper error message will be displayed to the user , if the new password is not updated to the backend system.
 - **Time Estimation:**
 - Hours: 5-10h
-

- **Title**
 - Splash Screen
 - **Description**
 - As a new/registered user, I want to see a splash screen when I start up the application.
 - **Acceptance tests:**
 - Given an app is installed in the device, when the user taps on the app icon, the app opens displaying a splash screen , which stays for 1 or 2 sec and then navigates to the signin/signup screen.
 - **Time Estimation:**
 - Hours: 1h
-

- **Title**
 - Medications
- **Description**
 - As a user, I want to be able to add my medication to my list of medications in the Simple Health app so that I can be notified when it is time to take my medication.
- **Acceptance tests:**
 - Given that the medication screen is available in the app, the user will be able to enter the name, dosage, and frequency of how often they take their

medication and then the user will be sent a push notification at the appropriate time to take their medication.

- Given that the medication screen is available in the app, the user will be able to login to the app and then be able to view and edit their medications on the medications screen.
- Push notification should be able to pop up when the user needs to take medicine and the time should match the user's timetable.

- **Time Estimation:**

- Hours:10-20h

- **Title**

- Conditions

- **Description**

- As a user, I want to save my current condition and be able to receive an alert, so that I can eat medicine or do some medical checks on time.

- **Acceptance tests:**

- Given a "Add new" button is available, when the user clicks on the button, a text entry will pop up and let the user fill in the blanks. After the user fills in the information and clicks the 'Done' button on the pop-up text entry, the application will connect to the database and write the new condition to it. While processing the database, the textbox will also update the list, so the change will happen immediately.
- Given a checkbox to on/off the Alerts, when the user checks the checkbox, the alert is on and will send a notification at setting time. Vice versa.
- Users should be able to access the conditions page directly when clicking on the pop up push notification.

- **Time Estimation:**

- Hours:10-15h

- **Title**

- Exercise

- **Description**

- As a user, I want to be able to select any exercises that I want to be reminded to do throughout my day.

- **Acceptance tests:**

- Given a user has selected exercises to be notified to do, then when that notification comes, the user will be able to click a "Tutorial" button on the notification which will bring up a video on how to do that certain exercise.

- Given an exercise page is available, when the user selects what exercises they want to be notified about, then they will receive push notifications at random times instructing them to do a certain exercise.
- **Time Estimation:**
 - Hours:5-10h

- **Title**
 - Mind Games
- **Description**
 - As a user, I want to access the Mind Game page so I can play some mind games to help me wake up
- **Acceptance tests:**
 - Given a user is on main screen and taps on 'mind game' button, the user will be navigated to the mind game screen
 - Given a user is on mind game screen, the user will be able to set wake up time and change push notification settings
 - Given at least one mini-test for users to choose, so users can avoid playing the game users don't like.
- **Time Estimation:**
 - Hours:10-15h

Desirable Features:

- **Title**
 - Video tutorial
- **Description**
 - As a logged in user, I want to access the video tutorial for the selected exercise category, so I can add to my fitness routine to achieve a specific goal.
- **Acceptance tests:**
 - Given Exercise screen is available, When the user clicks on the "Video tutorial" button, the page will be displayed to the user to display the Respective exercise video tutorial.
 - Given the Video tutorial screen is available, When the user clicks on the "save" button, Then this tutorial will be saved to the DB system, so this can be displayed in user exercise routine plan.

- **Title**

- Goal setting
 - **Description**
 - As a user, I want to set an exercise goal on the app.
 - **Acceptance tests:**
 - The function should be under the exercise page, and data should bind with the user.
 - Users can set plans as per day/week/month.
 - The user should be able to start, remove, or edit any plan at any time.
 - The database needs to update instantly after the user modifies a plan.
-

- **Title**
 - Goal tracking
 - **Description**
 - As a user, I want to track my exercise progress by looking at the time table or view the exercise history.
 - **Acceptance tests:**
 - A graph is required to present the goal tracking, and the graph needs to update anytime the user wants to view it.
-

Optional Features:

- **Title**
 - Integrate with real data
 - **Description**
 - As a developer, I want to be able to use data through a database or server.
 - **Acceptance tests:**
 - Verify the service call request/response or database entries to check all mock data is replaced with database /service response.
-

- **Title**
 - Credential hashing
- **Description**
 - As a developer, I want to hash all the user-related information before saving them into the database.
- **Acceptance tests:**
 - Implement a hash function at the back end to deal with any user-related information.

-
- **Title**
 - Connecting with a fitness device
 - **Description**
 - As a developer, I want to connect a fitness device with our application.
 - **Acceptance tests:**
 - Given a API, so developer can read data from the fitness device.
-

Nonfunctional Requirements:

- **Title**
 - Quick respond
 - **Description**
 - As a user, I want the application to respond within max of 1-2 seconds for every click/page load actions , so it will be a user-friendly experience.
 - **Acceptance tests:**
-

Security requirements

- **Title**
 - Forget Password
 - **Description**
 - As a user, I want my Password to be stored safely using a proper algorithm in a database, so that my account will remain secure .
 - **Acceptance tests:**
 - Given a button, when user click on it, user will be direct to another page and be able to
-

- **Title**
 - Log out
- **Description**
 - As a user, I should be able to logout from my application, so post login features should not be accessible.

- **Acceptance tests:**

4. Management Plan

a. Process Model

We are following AGILE methodology , where each week there are new iterations.Using “PivotalTracker” to create epic and stories. Start of iteration , there will be a story grooming held to go over the stories for the current iteration and assigned to Developer.Once Developer completes the dev work, story will be assigned to the Test lead for the testing process. Any defect will be addressed and fixed before closing the story. Thus in each iteration, we will be delivering a set of features which will be thoroughly tested and any defects will be identified and fixed .

Team communication will be through SLACK . Any open issue , update on task, questions to be answered will be posted by team members in slack. Two days before the submission, the team will have a meeting to review the work .

b. Objectives and Priorities

Objective is to deliver a high quality app without any known defects. To achieve this, developers will be following standard coding practices and proper documentation and architecture design for the app. Testing team will cover a wide range of test cases. As a team , we will target to complete the essential, desirable feature, if time permits we deliver optional features. Configuration lead will be handling CI/CD deploy for the Git repo.

c. Risk Management (need to be updated constantly)

1. **Communication** : Weekly meeting at start of iteration to go over task assignment, any team member is unclear with requirement should be connecting with Requirement lead , if needed meeting will be set up with the entire team.
2. **Requirement** : Design and implementation lead is creating wireframe for the entire flow and if any requirement change in upcoming iteration should be updated in wireframe and communicated to Requirement lead for the story creation, developer , tester will be notified .
3. **Management** : Start of iteration, Requirement leader will be having grooming sessions to go over stories for the iteration and assigning stories to proper owner, so tasks will be assigned and completed in a timely manner.
4. **Design and implementation** : Design and implementation Lead will be coming up with proper architecture for the app and will be reviewed by the team before starting with development to avoid design error and to achieve clean coding.

5. **Testing** : Thorough testing before each iteration closure . Test lead will be working with the developer to get the test case reviewed which will make sure that proper flow of testing is done.
6. **Technology competence** :some members are only familiar with Android development, so members who are proficient in coding will be doing critical parts of app development, meanwhile other members will be contributing to the app development and proper code review will be done. As a team, helping each other to understand tool usage , unblocking the task if any.
7. **Team member absence**: If any team member is sick and not able to complete a task, should be communicating with Project lead/backuplead inorder to work on alternate options to get the task completed.

[Risk Management Sheet Link: RiskManagement](#)

d. **Monitoring and Controlling Tools and Mechanisms**

We will use the following tools to facilitate group communication and monitor the project progress.

- i. Pivotaltracker Link: <https://www.pivotaltracker.com/n/projects/2531657>
- ii. Slack Link: <https://bumetcs673olf21.slack.com/archives/C02DMV6JPDZ>
- iii. Github Link: <https://github.com/BUMETCS673/BUMETCS673OLF21P5>
- iv. Zoom meeting Link: <https://bostonu.zoom.us/j/91547420830?pwd=VHR1dTNPWmtzNTlqUzNTTXhoTTVhUT09>
- v. Weekly meeting time: 9:15-10:15(tentative)

e. **Timeline** (need to be updated at the end of each iteration) : Will update this section by start of Iter1

Iteration	Functional Requirements(E/D/O)	Tasks	Estimate d/real persons on hours	Presentation Recording Link (5-10 minutes)
0	Initial Planning Phase (Listing E/D/O features)	<ul style="list-style-type: none"> Project Proposal, Presentation , Documents Preparation(SPPP, Risk management...etc) Basic Project template CI/CD setup 	/27.5	team5 presentation .mp4

Commented [1]: You should still fill in your plan for future iterations,. You will also need to be updated at the end of each iteration.

Commented [2]: Sure, We will make sure to update this for every iteration.

1	Login,Signup,Splash ,Forgot Password	1.Pivotal Tracker: Stories created 2.Development : <ul style="list-style-type: none"> • UI development • Activities and Fragments. • DB setup and Repository,Entity and DAO Created for features • Test Case Preparation • Dev and Integration testing • Defect Fixing 3. Configuration: <ul style="list-style-type: none"> • Build creation • Branch merging from dev to main 		https://drive.google.com/file/d/134PpnkX9YrTiMdDBYhgtWQglXTnP1f3j/view?usp=sharing
2				
3				

5. Quality Assurance Plan

a. Metrics

- Per iteration, the number of bugs or defects opened will be compared to the number of issues resolved by the team at the end of each iteration.
- Results will be documented in the table below and updated at the end of every iteration.

	Iteration 0	Iteration 1	Iteration 2	Iteration 3	Iteration 4
QA Tests Conducted	0	6			
Defects Opened	0	2			
Defects Resolved	0	2			

b. Standard

- i. For coding standards, all code that is checked into Git must pass linting using the Checkstyle linter. As part of the CI/CD pipeline inside Jenkins, in order for a branch to be able to be merged into dev it must pass linting as well as unit tests.
- ii. Once a branch is merged into dev, before it is deployed to our environment it will be run through our QA tests to ensure that no known or new bugs have been introduced into the system. These tests will be conducted by the QA lead and documented in the Testing document.
- iii. No direct pushes to the main or dev branch is allowed. A pull request must be created for a feature branch to be merged into develop and it must be reviewed and approved by at least one team member.

c. Inspection/Review Process

- Every iteration, two days prior to the submission, the team will have a Review meeting to go through completed docs, code . Along with that for every code commit , a new pull request will be created and once reviewed only it will be merged to the main branch.
- Test lead will set up a meeting for the test case review with Dev lead to make sure all possible flows are covered for positive and negative use cases.

d. Testing:

- Our plan for testing is as follows, in order for a feature branch to be merged into dev or main, all unit tests must pass. If that feature branch added new functionality or methods, then unit tests must be added to confirm these new changes are working as expected. Once the changes are merged the unit tests will run once more as part of the github action. After the changes are merged into dev and the pipeline has finished, the QA lead will run the quality assurance tests to confirm no new bugs have been introduced into the existing system. Then new QA tests will be created from the help of the development lead to test any and all new features that have been added to the application. Finally the results of the QA tests will be recorded in the testing document, any bugs will have a defect opened in pivotal tracker and the team will be notified of the results of the test.
- **Refer to the Testing document for our testing report: [Testing](#)**

e. Defect Management

Pivotal tracker defects will be opened upon bug discovery and then a team member will be assigned to the defect to resolve it. Defects with High priority should be resolved within 1 day. Defects with Medium or Low priority can be resolved within 3 days.

6. Configuration Management Plan

(For more details, please refer to SCMP document for encounter example)

- a. Configuration items and tools
 1. **Archiving/Version Control**
Github
 2. **IDE**
Android Studio
 3. **Requirement Control**
Pivotal Tracker
 4. **Database management**
Room
- b. Change management and branch management
 1. **Master branch : origin/main**
 2. **Dev branch : origin/dev**
Stores fresh code. Merged into master branch at the end of each iteration.
 3. **User branches :**
Each team member who wishes to commit code into repo should create their own branch. User branches are merged into Dev upon completion of a feature.
- c. Code commit guidelines
 1. Commits should include a log that briefly describes the commit. (i.e. what's new, what's changed, what's removed)
 2. Pull requests should be opened for team members to review. A minimum of 1 review is needed before merging.
 3. Make sure codes are complete and without any known bugs before committing.
- d. Integration and deployment plan : CI/CD auto deploy through github actions.

Commented [3]: how? through pull request?

Commented [4]: Yes , pull request will be created

7. References

(For more details, please refer to the encounter example in the book or the software version of the documents posted on blackboard.)

8. Glossary