**CS673 Software Engineering** 

**Team 5 - Simple Health**

**Project Proposal and Planning**

| Team Member | Role(s) | Signature | Date |
| --- | --- | --- | --- |
| 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**

| **Version** | **Author** | **Date** | **Change** |
| --- | --- | --- | --- |
| 1.0 | Jake Stephens,Seema Palora | 9/9/2021 | First additions, Updated Optional Features |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[Overview](#_87t9hln2vjz0)

[Related Work](#_mps353x5ezyl)

[Detailed Description](#_fg3z0hpd4q9v)

[Management Plan](#_ds8oyr75pnh1)

[Process Model](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.27177f40uci)

[Risk Management](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.a4oqwntk3mw)

[Monitoring and Controlling Mechanism](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.ywdoc2clc9yt)

[Schedule and deadline](#_tadq5mb0pici)

[Quality Assurance Plan](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.72e1f4uawy2r)

[Metrics](#_b2haznn3yyz2)

[Standard](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.vc72k6dweldv)

[Inspection/Review Process](#_f1c69ifi68h7)

[Testing](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.r5d5mhtlf0kq)

[Defect Management](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.54a4wuncjg1c)

[Process improvement process](#_jhct37ebxxpn)

[Configuration Management Plan](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.hw41vg4ykxen)

[Configuration items and tools](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.bwlb4d4vdox2)

[code commit guidelines](#_yyauft6zr9hw)

[References](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.8mva2050iy7t)

[Glossary](#_ty3i2nqffhtc)

# 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.

# 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.

# Proposed High level Requirements

## Functional requirements:

### Essential Features:

* **Title**

#### Medications page

* **Description**
  + As a user, I want to access the medications page, so that I can upload my medicine record and timetable for my medicine.
* **Acceptance tests:**
  + This page should be accessible by clicking a button on the main screen after the user login in.
  + This page should have at least one button for users uploading their medicine information like name and time to take.
  + Whenever users open this page, the data recorded on the machine/server should show up properly.
  + 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:10-20
* **Title**

#### Conditions page

* **Description**
  + As a user, I want to access the Conditions page, so that I can select my current condition. A push notification will pop up randomly and ask me my condition based on this page.
* **Acceptance tests:**
  + This page should be accessible by clicking a button on the main screen after the user login in.
  + This page should have at least a drop down menu for users to select conditions.
  + Push notifications should be able to pop up randomly.
  + Users should be able to access the conditions page directly when clicking on the pop up push notification.
* **Time Estimation:**
  + Hours:10-15
* **Title**

#### Exercise page

* **Description**
  + As a user, I want to access the Exercise page, so that I can set an exercise plan and let the app manage it.
  + As a user, I want to get a push notification when it is the time I plan to do some exercise.
* **Acceptance tests:**
  + This page should be accessible by clicking a button on the main screen after the user login in.
  + This page should be able to add and delete their exercise plan.
  + A time table should be shown according to the user's exercise plan.
  + Push notifications should be able to pop up properly.
* **Time Estimation:**
  + Hours:5-10
* **Title**

#### Mind Games page

* **Description**
  + As a user, I want to access the Mind Game page, so that I can play some mini-brain games when I’m waking up.
* **Acceptance tests:**
  + This page should be accessible by clicking a button on the main screen after the user login in.
  + If the user sets the wake up time, push notification should be able to pop up at that time.
  + This page should have at least one mini-test for users to do.
* **Time Estimation:**
  + Hours:10-15
* **Title**

#### Login page

* **Description**
  + As a user, I want to be able to log in to my account that stored my information.
* **Acceptance tests:**
  + This page should contain 2 text entry boxes and a login button so users can enter username and password then click the button to finish login in.
* **Time Estimation:**
  + Hours: TBD
* **Title**

#### Signup page

* **Description**
  + As a user, I want to be able to create an account by entering username and password.
* **Acceptance tests:**
  + This page should contain 2 text entry boxes and a sign up button so users can enter username and password then click the button to finish sign up.
* **Time Estimation:**
  + Hours: TBD

### Desirable Features:

* **Title**

#### Video tutorial

* **Description**
  + As a user, I want to follow a video tutorial when I start an exercise.
* **Acceptance tests:**
  + The application should be able to create a window that plays YouTube videos.
  + The Youtube video should match the exercising user is currently doing.
  + The YouTube URL has to return status code 200.
* **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:**
  + Read data from the fitness device.

## Nonfunctional Requirements:

### Security requirements

# Management Plan

## 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 .

## Objectives and Priorities

Objective is to deliver high quality and defect free app. 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.

## 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:](https://docs.google.com/spreadsheets/u/1/d/1zdGeYZStMGEk0qYSqInmmaJHKfvkgcYr02fNFc5sWyA/edit) [RiskManagement](https://docs.google.com/spreadsheets/d/1zdGeYZStMGEk0qYSqInmmaJHKfvkgcYr02fNFc5sWyA/edit?usp=sharing)

## Monitoring and Controlling Tools and Mechanisms

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

* + 1. Pivotaltracker Link: <https://www.pivotaltracker.com/n/projects/2531657>
    2. Slack Link: <https://bumetcs673olf21.slack.com/archives/C02DMV6JPDZ>
    3. Github Link: <https://github.com/BUMETCS673/BUMETCS673OLF21P5>
    4. Zoom meeting Link: <https://bostonu.zoom.us/j/91547420830?pwd=VHR1dTNPWmtzNTlqUzNTTXhoTTVhUT09>
    5. Weekly meeting time: 9:15-10:15(tentative)

## 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 | Estimated/real person hours | Presentation Recording Link (5-10 minutes) |
| --- | --- | --- | --- | --- |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

# 

# Quality Assurance Plan

## Metrics

* + 1. Number of defects or issues created in Git will be how the quality of our released code is measured.
    2. As for our code quality overall, the ratio of succeeded to failed pipelines will be tracked.
    3. Results will be documented and posted for the group to track and work on improvements.
  1. Standard
     1. 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 develop it must pass linting as well as unit tests.
     2. Once a branch is merged into develop, before it is deployed to our environment it will be run through our end-to-end test suite to ensure that no bugs have been introduced into the system. The end to end tests will be run automatically through the github CI/CD action step. If bugs have been introduced, then the pipeline will stop here and not build the next version of the APK.
     3. No direct pushes to the main or develop 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.

## 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.

## Testing:

**Refer to the Testing document:** [**Testing**](https://docs.google.com/document/d/1xz9nrPrB4mf8WqU6mYGy7w-WA32dsy5MEdrRv_aNKXc/edit?usp=sharing)

## Defect Management

## Github issues will be opened upon bug discovery and a defect will be created in PivotTracker where then a team member will be assigned to the defect to resolve it. Defects will be 1 point stories and should be resolved within 1 day.

# Configuration Management Plan

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

## Configuration items and tools

1. **Archiving/Version Control**

Github

1. **IDE**

Android Studio

1. **Requirement Control**

Pivotal Tracker

## Change management and branch management

1. **Master branch**
2. **Development/release branch**

Stores fresh code. Merged into master branch at the end of each iteration.

## 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. Fresh code should be reviewed by at least 1 team member before being committed.
3. Make sure codes are complete and bug free before committing.
   1. Integration and deployment plan : CI/CD auto deploy through github actions.

# References

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

# Glossary