Software Requirements Specification

for

Open-Jio

Version 3.0

Prepared by

Tio Guo Yong (Leader) (U2123181B)

Chen Zihang (Asst. Leader) (U2121486H)

Zhang Jing Wen (U2121853G)

Jayden Yeo He (U2120348J)

Nema Aarushi (U2120814C)

Team Fivver

3rd November 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Description, Functional & Non-functional requirements	25/8/2022	First version created	1.0
Data Dictionary	29/8/2022	Data Dictionary created	1.1
First check	31/8/2022	First revision	1.2
Second Revision	30/10/2022	Modified and added new requirements	2.0
Third Revision	3/11/2022	Added external interface requirements and modified functional and non-functional Requirements	3.0

1. Introduction

1.1 Purpose

This documents the software requirements and specifications for OpenJio. OpenJio is a mobile application that allows users to self-initiate activities and join open invitations within Singapore. This app can be installed from the Android or IOS App store. This document will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2 Document Conventions

Text formats
Font: Times New Roman
Font size: 12 for Body, 14 for Sub-Headings and 18 for Headings
Line height: 1.15

1.3 Intended Audience and Reading Suggestions

The document is intended for all the stakeholder customers and developers — designers, coders, testers and maintainers. This document need not be read sequentially; users are encouraged to jump to any section they find relevant. Below is a brief overview for each part of the document.

Part 1 (Introduction)
This section offers a summary of the Open-Jio project, including goals, objectives, project scope, general system details.

Part 2 (Architectural and Component-Level Design)
This section describes the Open-Jio system class by class, including interface details, class hierarchies, performance/design constraints and process details.

Part 3 (User Interface Design)
This section covers all of the details related to the structure of the graphical user interface (GUI), including some preliminary mockups of the Open-Jio mobile application. Readers can view this section for a tentative glimpse of what the final product will look like.

Part 5 (Test Cases)
This section offers a list of test cases, testing and expected output, and other pertinent information.

Part 6 (Appendices)
This section includes any additional information which may be helpful to the readers.

1.4 Product Scope

Open-Jio is an utility application in daily life which facilitates physical social activities. Its objectives include

1) Easing open-invitations and to create a convenient and easy-to-use application for users, trying to find an activity to do on the spot.
2) Promoting the use of technology to bond with the community.

1.5 References

Bruegge, & Dutoit, A. H. (2010). Object-oriented software engineering : using UML, patterns, and Java (3rd ed.). Prentice Hall.

Sommerville, I. (2016). Software Engineering. 10th Edition, Pearson Education Limited, Boston.

2. Overall Description

2.1 Product Perspective

The concept of an “open-jio” or open-invitation is popularised in Singapore school hostels, where students plan impromptu sports activities that other students in the hostel are free to join. These invitations are entirely facilitated through Telegram group chats, and our app Open-Jio will be the first dedicated platform of its kind.

2.2 Product Functions

2.2.1 Organise Activities

The app allows users to organise activities, which are categorised into sports, hangout, etc.

2.2.2 Explore Activities

Users can search for activities to join by category, location and time. Upcoming activities in their vicinity will also be automatically recommended to them.

2.2.3 Friends

Users can add other users as friends, allowing them to chat and view what each other are participating in.

2.3 User Classes and Characteristics

a. Youths: It includes students, and young adults who are more familiar with online social interactions.
b. Adults: It includes daily workers or any other individuals who like to join social events.

2.4 Operating Environment

Our mobile app will operate on the IOS and Android platforms, and will require location services.

a. For Android platforms, the OS version must be at least Android 6 (Marshmallow).
b. For IOS platforms, the OS version must be at least IOS 9.
c. For location services, the device must have built-in support of Global Positioning System.

2.5 Design and Implementation Constraints

a. User-interface/front-end: ReactJS version 18.2.0
b. Database/back-end: Firebase version 9.12.1
c. Map: OneMapAPI
d. Navigation: React Router version 6.4.1

e. Android OS: at least Android 6 (Marshmallow).
f. IOS: at least IOS 9.
g. Each page must have finished loading with complete details within 5 seconds.
h. The maximum memory used by the app must not exceed 1GB.

2.6 User Documentation

a. 'How to begin' section containing basic information on how to use the app in the app store.
b. A promotional Youtube video on how to use the app.

2.7 Assumptions and Dependencies

a. OneMap API requests do not exceed 250 calls every minute.
b. The device has built-in support for the Global Positioning System.
c. API calls to Firebase can successfully return complete data to the application.

3. External Interface Requirements

3.1 User Interfaces

a. The buttons on the app must have a noticeable change in colour to indicate the user is hovering through it.
b. Upon signing in, the app must allow the user to navigate to the home page, creation of activity page, saved activities page, and profile page from any pages.

3.2 Hardware Interfaces

a. The device used to open this application must support Global Positioning System for the tracking of current user's location.
b. The device must have at least 300MB free memory for the stability of the application.

3.3 Software Interfaces

a. For Android devices, the OS version must be at least Android 6 (Marshmallow).
b. For IOS devices, the OS version must be at least IOS 9.

3.4 Communications Interfaces (TBD)

a. The app requires internet access.
b. The app must be able to send an email containing a verification link to change the user's password if that user forgets his or her own password.
c. The app must be able to send notification to a user if:
    i. Another user sends a message to that user.
    ii. Another user accepts a friend request from that user.
    iii. The organiser (user) of an activity approves that user's join request.
    iv. The organiser (user) of an activity removes that user as a participant.
    v. The organiser (user) cancels an activity.
    vi. The participant (user) leaves an activity.

4. Functional Requirements

4.1 User Account & Authentication

a. Registration
Description: The app must allow the current user to sign up for an account if there is not one associated with their email address.
Input: Email, username, name, password, and birthdate.
Output: Account successfully created and stored inside database.

b. Location Permission
Description: The app must request for the current user's permission to access their location information.
Input: User's approval of location request.
Output: Current location of user.

c. Login
Description: The app must allow the current user to log into the app using the account they created.
Input: Email, password.
Output: After login information validated, navigate to the home page.

d. Logout
Description: The app must allow the current user to log out of their account.
Input: Click on 'Logout' button.
Output: The sign-in page.

e. Reset password
Description: The app must allow the current user to reset their password by clicking on the verification they received through their registered email.
Input: Registered email.
Output: Email sent to user containing verification link to input new password.

4.2 User Profile

a. Update Profile
Description: The app must allow the current user to edit the information displayed in their own profile page.
Input: Profile photo, username, name and/ or description.
Output: Profile information updated, navigates to profile page.

b. My Friends

Description: The app must display a list of all other Users that current User is friends with, and the list of friend requests.
Input: Click on the “My Friends” icon.
Output: List of friends and friend requests.

c. My Activities
Description: The app must display a list of all activities that User has saved, joined and organised.
Input: Click on the “My Activities” icon.
Output: List of activities.

d. My Chats
Description: The app must display a list of all the chats User has with other users
Input: Click on the “My Chats” icon.
Output: List of all chats .

e. Chats Read
Description: Inside My Chats, the app must indicate whether the chat has been read by current User.
Input: -
Output: A blue banner under all unread chats.

f. Other User Profile
Description: The app must allow the current user to view the profile page of another user.
Input: Click on another user’s profile photo/name.
Output: The profile page of that user.

4.3 Explore Activities

a. Search Activity
Description: The app must allow the current user to search for upcoming activities initiated by other users.
Input: Name of activities.
Output: List of relevant activities filtered by above.

b. Nearby Activities
Description: The app must display to the current user all the upcoming activities in their vicinity (within 10km).
Input: Location permission.
Output: List of relevant activities will be displayed. Location markers of activities will be shown on the map.

c. Sort by Time (Default)

Description: For all the upcoming activities in the user's vicinity (within 10km), the app will display them in order of earliest to latest.
Input: Default, or click on Location Symbol to switch to Time Symbol, to change it from Sort by Location to Sort by Time.
Output: Filtered activities will be displayed in the order of earliest to latest.

d. Sort by Location
Description: For all the upcoming activities in the user's vicinity (within 10km), the app will display them in order of nearest to farthest.
Input: Click on Time Symbol to switch to Location Symbol, to change it from Sort by Time to Sort by Distance.
Output: Filtered activities will be displayed in the order of nearest to farthest.

4.4 Activity Organiser

a. Create Activity
Description: The app must allow the current user to create activities that other users can join.
Input: Activity name, description, location, time, maximum number of participants and/or activity image.
Output: New activity created and searchable by other users.

a. Manage Participants
Description: The app must allow an organiser to approve or reject participants.
Input: Organiser's selection of users.
Output: Participants are rejected or approved.

b. Update Details
Description: The app must allow an organiser to change details of the activity.
Input: Activity name, description, location, time, maximum number of participants and/or activity image.
Output: Activity's details successfully updated.

c. Cancel Activity
Description: The app must allow an organiser to cancel the activity that is at least 30 minutes before commencing.
Input: Click on the 'Cancel' button.
Output: The activity is cancelled for all participants.

4.5 Activity Participant

a. Join Activity
Description: The app must allow the current user to join upcoming activities initiated by other users.
Input: Click on the 'Join' button.

Output: Organiser receives a request from that user to join the activity.

b. Withdraw Join Request
Description: The app must allow the current user to withdraw his or her join request for upcoming activities initiated by other users.
Input: Click on the 'Pending' button.
Output: User's request will be withdrawn and removed from pending requests in organiser's activity

c. Save Activity
Description: The app must allow the current user to save an activity that the user does not intend to join.
Input: Click on the 'Save' button.
Output: The saved activity will appear in the user's list of activities.

d. Leave Activity
Description: The app must allow the current user to leave an activity that he/ she has indicated their participation for.
Input: Click on the 'Leave' button.
Output: User leaves the activity successfully and the organiser is notified.

4.6 Chat Functions

a. Message Users.
Description: The app must allow the current user to send or receive messages to and from other users.
Input: Select a user to chat with.
Output: Users successfully send messages to other users or receive messages from other users.

b. Delete Message
Description: The app must allow the current user to delete messages they've sent.
Input: Click on the message User will like to delete.
Output: Message is removed from database.

4.7 Friend Management

a. Add Friend
Description: The app must allow the current user to add another user as a friend.
Input: Send friend request to another user.
Output: The recipient receives a friend request notification, that they can then approve/ reject.

b. Remove Friend
Description: The app must allow the current user to remove a friend.
Input: Click on the 'Remove friend' button.

Output: Friend relationship removed.

c. Friend's Activities
Description: The app must allow the current user to view the upcoming activities participated by a friend.
Input: Click on the 'View Activity' button on a friend's profile page.
Output: List of upcoming activities participated by a friend.

4.8 Notifications

a. Notifications
Description: User must receive notification for:
   i. Incomplete fields when registering, editing profile, creating activity and update activity
   ii. Incorrect username/password
   iii. Location permission
   iv. New Message
   v. Friends:
      1. New friend request
      2. Accepted friend request
   vi. Activities:
      1. New activity join request
      2. Accepted as participant
      3. Rejected as participant
      4. Removed from event
      5. Activity cancellation

Input: None
Output: Notification alert

5. Other Nonfunctional Requirements

5.1 Performance Requirements

a. The system should be able to support 1000 simultaneous users.
b. The database should be able to store up to 10 thousand active activities, each with an image.
c. The map should be able to render all locations within Singapore.
d. The mean time to view a page over a 56Kbps modem connection will not exceed 3 seconds.
e. The application will run on all web browsers.

5.2 Safety Requirements

a. Users under the age of 16 will not be able to register for an account.

5.3 Security Requirements

a. Users are required to login with their email and password.
b. Users are to keep their password safe and not share it with any other people, applications, or websites under any circumstances.
c. Users are reminded not to leak their personal information such as addresses over the app.
d. User accounts must be verified in order to post or join activities on the application.

5.4 Software Quality Attributes

a. Adaptability
    i. The system will be able to operate on all web browsers.
b. Maintainability
    i. The system will be opened for extension but closed for modification.
    ii. The system code will be easy to read and understand.
c. Reliability
    i. There will be backup servers in place for redundancy.

5.5 Business Rules

a. Developer
    i. Developers can remove users that do not abide by the community guidelines.
    ii. Developers can remove activities that breach the community guidelines.
b. User

i. Initiators can make adjustments to the details of the activity, such as date, description and location and time.
ii. Initiators can manage the participants of their activity.
    1. Accept or reject participants.
    2. Remove participants once they have joined the activity.
iii. Initiators can cancel the activity due to unforeseen circumstances.
iv. Initiators can also pass on the role of host to another user.

Appendix A: Data Dictionary

a. User Account
    i. A user account is a location on a network server used to store a computer username, password, and other information. A user account allows or does not allow a user to connect to a network, another computer, or other share. Any network that has multiple users requires user accounts.
b. Mobile Operating System
    i. A mobile operating system, also called a mobile OS, is an operating system that is specifically designed to run on mobile devices such as mobile phones, smartphones, PDAs, tablet computers and other handheld devices.
c. Database
    i. A database is a collection of information that is organised so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content.
d. Friend
    i. A friend of a user is another user that has been connected to him/her. Adding a friend means the users can see one another's upcoming activities.
e. Activity
    i. An activity is an event initiated and administered by the organiser. Other users can request to join the activity but only the organiser can accept their requests. The activity will include details such as location, date, type of activity as well as participants limit.
f. Organiser
    i. An organiser is an user that initiates the activity. He will provide the details of the activity and be able to accept or reject other users' requests to participate in the activity. He will also be able to edit the details of the activity.
g. Participant
    i. A participant is a user whose request has been approved by the organiser and is taking part in the activity. The participant can withdraw from the activity if he/she wants to do so.
h. Profile page
    i. A page displaying the details of a user that he/she wants to share with other users. These details include profile photo, username, name and/ or description.
i. UI (User Interface)
    i. The means by which the user interacts with the software program. This can include different pages of the application, including (but not limited to) Home page UI, Profile page UI.