Software Requirements Specification

for

JustJio

Version 1.0.1 approved

Prepared by

TEY KAI SEONG,
CHEUNG ZHI HENG,
GOH KUAN WEI, ELDRICK,
AMABEL LIM HUI XIN,
JEYASEELAN HARISH VASANTH,
LOH SIJING ALOYSIUS

Nanyang Technological University, Team OneStart

28/8/2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Tey Kai Seong	28/8/2022	Initial documentation after specifying requirements	1.0.0
Tey Kai Seong	5/9/2022	Update non-functional requirements, use case models and user interface	1.0.1

1. Introduction

1.1. Purpose

When planning a party, a lot of hassle goes into the preparation of food, drinks, entertainment and of course the splitting of the bill. With JustJio, everything can be made easily available to you. JustJio reduces the amount of time people spend looking for good food to bring to the party. For example, if the user wants a hotpot party, ingredients can be easily found nearby through JustJio's food recommender; and if none are available, a delivery option is also open to them. This allows users to spend a substantially lower amount of time searching for good food and drinks, and it may even save them the hassle of going out to buy. JustJio also has a game recommender that recommends party games to facilitate entertainment.

JustJio also provides online rooms for the attendees of the party to join. This allows easier communication between the attendees and it also acts as a way to spilt the bill at the end of the party. This causes less confusion when it comes to paying the bill as JustJio's algorithm will take into account all expenses from different people and tell each person individually how much to pay and who to pay to. A transaction history will also be provided, making it easier to track transfers of money.

Overall, JustJio is here to enhance the party experience by making every step of the way less taxing on the individual.

1.2. Document Conventions

Software Requirement Specification Format: This document follows IEEE standard. Priorities of higher level requirements are inherited by detailed level requirements.
Font: Times New Roman
Main Header: Size 18, Bold
Subsection Header: Size 14, Bold
Content: Size 12
Further conventions on the terms used could be found at **Appendix A – Data Dictionary Section**

1.3. Intended Audience and Reading Suggestions

This document is intended for all stakeholders involved. This includes the users of the app, the software developers of the app, project managers, marketing staff, testers and all business partners involved such as the delivery company used etc.

The document contains details of the different functions of the app, its intended purpose and the reasoning for having such a feature. A detailed description of each feature is provided and examples of the different app pages will be shown to allow better understanding of the app layout.

The document is best read in sequence for users, testers, marketing staff and business partners. For the software developers and project managers, the document is best read by looking at the overall description first, followed by the individual system features, and then finally the requirements needed for the app to function well, which includes the external interface requirements and the non-functional requirements.

1.4. Product Scope

Our product will be presented in the form of an Android mobile application. The application will include a UI interface that is easily maneuverable. The application will be able to provide a smoother experience for users to plan and host a party by providing ease of access to food and drinks, as well as an easy way to coordinate the details and the bill of the party with the other party members.

1.5. References

I. IEEE 830-1998 Template
II. HTTP Status Code - https://developer.mozilla.org/en-US/docs/Web/HTTP/Status
III. RESTful API - https://developer.mozilla.org/en-US/docs/Glossary/REST
IV. React Native Framework - https://flutter.dev/
V. Golang Fiber Framework - https://gofiber.io/
VI. PostgreSQL Database - https://www.postgresql.org/

2. Overall Description

2.1. Product Perspective

JustJio is a new refreshing app for both experienced and inexperienced party hosters in Singapore. It allows users to easily pick up items for their party such as food and drinks from nearby locations using Google Maps API, and if the user is unhappy with all nearby options, JustJio provides a delivery service using a delivery API. On top of that, JustJio allows users to open rooms. These rooms can be used to invite people to the party via an invite link. The rooms allow people to chat with each other but more importantly it also serves as a way to split the bill after the party. Each person can input what they are bringing to the party and how much was spent on them, which will all be calculated at the end using an algorithm. Each person will be given an amount that they have to pay to a certain person or up to multiple different people. When someone transfers money to another, the transaction will be recorded down and the transaction history can be viewed from the main menu. This allows efficient and smooth transfer of funds between people. Lastly, JustJio includes a list of the most played party games for people to refer to if they are unable to think of any.

2.2. Product Functions

JustJio has the following main functions:

1. The app allows User to find food and drinks from places near any location they input.
2. The app provides delivery options for User to choose from.
3. The app can create rooms for User to invite people that are going to the party.
4. The app allows User to chat with his/her fellow party-mates in the rooms.
5. The app allows User and all Users in the room to effectively split the bill after the party.
6. The app provides recommendations for party games.
7. The app provides a complete transaction history of all.

2.3. User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4. Operating Environment

This section lists the operating environments for both product and development environments of JustJio.

Product Environment of JustJio:

The version of the Android operating system must be Android 4.4 and above.

Development Environment of JustJio:

Development Environment	Description
Front-end: React Native for Mobile Application	React Native is a User Interface (UI) Software Development Library that is created by Facebook. It allows the development of native apps. Hence, the same codebase could be used to run in both Android and iOS phones.
Back-end: Fiber	1. Authentication is handled by Fiber. 2. All data is stored in PostgreSQL.

2.5. Design and Implementation Constraints

1. This application requires a backend server to function. The backend server uses PostgreSQL as its database and is hosted on Heroku. Due to the row limitation of Heroku's database, the data storage available is greatly reduced. This can be resolved by upgrading the service or switching to another cloud platform.
2. This application shall use PostgreSQL as the database.

2.6. User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7. Assumptions and Dependencies

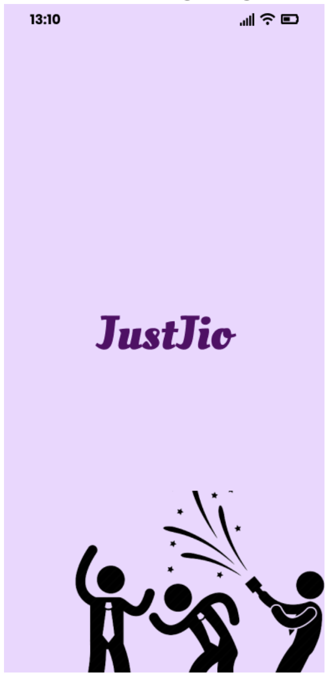
1. The application depends on external APIs to obtain related information.
   a. LINK_LOCATION_API =
   "https://maps.googleapis.com/maps/api/place/findplacefromtext/json"
2. We assume the data obtained from this API are accurate.
3. This application assumes that users have constant strong Internet connection, as most data are delivered by the server.

3. External Interface Requirements

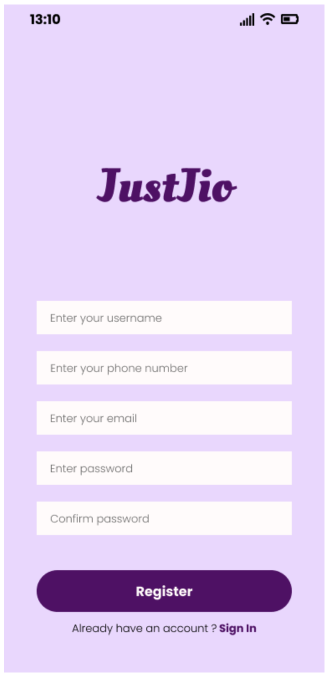
3.1. User Interfaces

The user interface of JustJio is designed to be user friendly. Additionally, to reduce the load for short-term memory, the layouts are designed with consistency in mind.

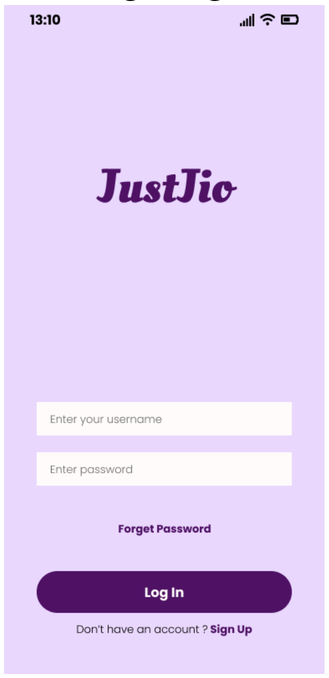
3.1.1 - *Landing Page*



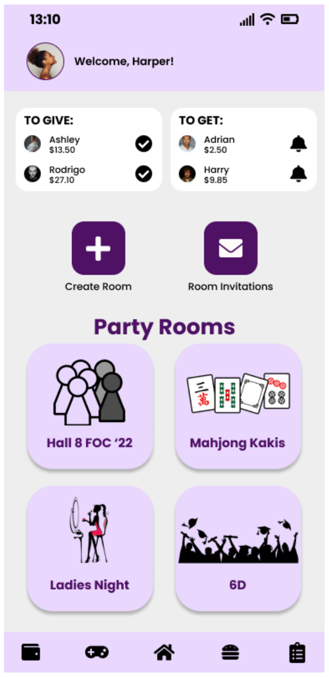
3.1.2 - *Registration Page*



3.1.3 - *Login Page*



3.1.4 - *Home Page*

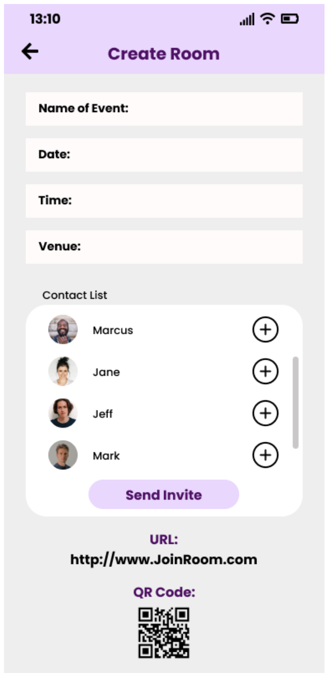

13:10

Welcome, Harper!

TO GIVE:
Ashley $13.50
Rodrigo $27.10

TO GET:
Adrian $2.50
Harry $9.85

3.1.5 - *Create Room Page*


13:10

Create Room

Name of Event:

Date:

Time:

Venue:

Contact List

Marcus

Jane

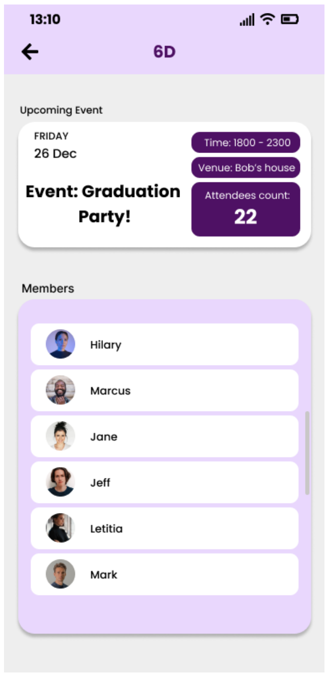
Jeff

Mark

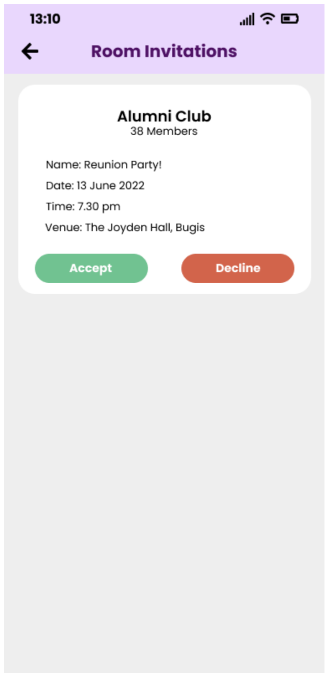
URL:
http://www.JoinRoom.com

QR Code:

3.1.6 - *Room Page*

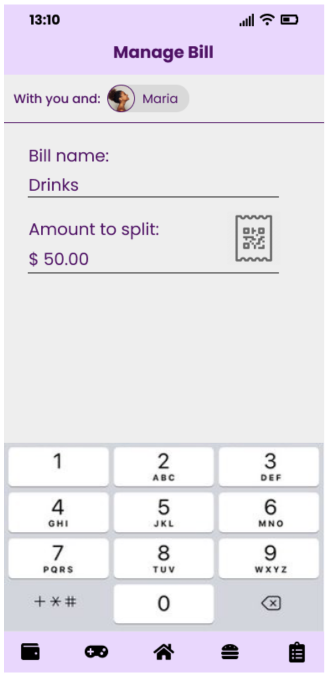


3.1.7 - *Join Room Page*



Name: Reunion Party!
Date: 13 June 2022
Time: 7.30 pm
Venue: The Joyden Hall, Bugis

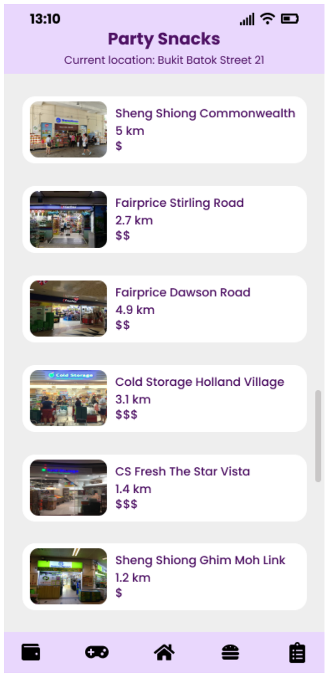
3.1.8 - *Split Bill Page*



Bill name: Drinks

Amount to split: $ 50.00

3.1.9 - *Party Snacks Page*



Sheng Shiong Commonwealth
5 km
$

Fairprice Stirling Road
2.7 km
$$

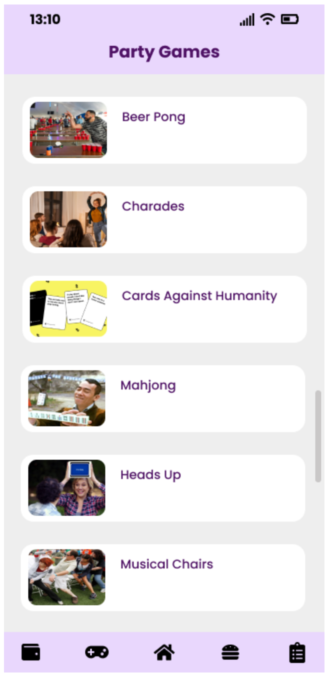
Fairprice Dawson Road
4.9 km
$$

Cold Storage Holland Village
3.1 km
$$$

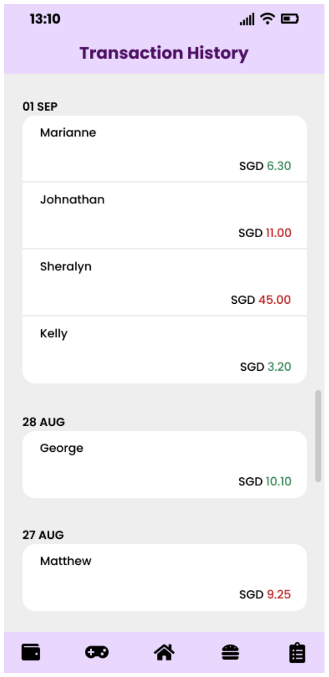
CS Fresh The Star Vista
1.4 km
$$$

Sheng Shiong Ghim Moh Link
1.2 km
$

3.1.10 - *Party Games Page*



3.1.11 - *View Transactions Page*



01 SEP

Marianne	SGD 6.30
Johnathan	SGD 11.00
Sheralyn	SGD 45.00
Kelly	SGD 3.20

28 AUG

George	SGD 10.10

27 AUG

Matthew	SGD 9.25

3.2. Hardware Interfaces

- The OneStart app requires a mobile device that supports the Android platform.
- The device must have Internet connection capability in order to transmit and receive data from the server.

- The device must contain a GPS unit in order to query for food recommendations around the current location of the user.

3.3. Software Interfaces

OneStart is supported by mobile devices running on the Android operating system. The version of the Android operating system must be Android 4.4 and above.

The architecture of this application has the following details:
- Front-end
    - React Native - framework to develop native mobile applications
- Back-end
    - PostgreSQL – relational database
    - Golang Fiber Framework – framework for developing and exposing REST API endpoints

3.4. Communications Interfaces

The communication between front-end and back-end will be following HTTPS protocol and TCP/IP network protocol.

4. System Features

4.1. Registration

4.1.1 Description

First time users can register for an account in OneStart. By registering, a record of users' rooms, transaction history and games history will be saved in the system.

4.1.2 Response Sequences / Use Case

Use Case ID:	REG1		
Use Case Name:	Register		
Created By:	Cheung Zhi Heng	Last Updated By:	
Date Created:	28/08/2022	Date Last Updated:	

Actor:	User
Description:	First time users can register for an account by clicking on the button 'Register'. Being a registered user, they will be able to manage rooms and monitor their transaction history and games history.
Preconditions:	The user should not have registered an account previously.
Postconditions:	The user has successfully created the account and logged in.
Priority:	High
Frequency of Use:	Low

Flow of Events:	1. User clicks on 'Sign Up' in the 'Log In' page. 2. System prompts the user to enter username, password, phone number and a valid email address. 3. User inputs the required information and clicks 'Register'. 4. System checks whether the information submitted is sufficient and valid. 5. System stores these details into the database. 6. User is logged in.
Alternative Flows:	REG1-AF-S3 If username is taken by another user: 1. System displays "Username already exists. Please use another username." 2. System returns to Step 2. REG1-AF-S3 If password do not meets the requirements: 1. System displays "Passwords do not match requirements.". 2. System returns to Step 2. REG1-AF-S3 If email address is invalid: 1. System displays "Please enter a valid email address.". 2. System returns to Step 2. REG1-AF-S3 If phone number is non-verifiable: 1. System displays "Please enter a valid phone number.". 2. System returns to Step 2.
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	1. User should connect to the Internet when registering.
Notes and Issues:	Nil

4.1.3 Functional Requirements

1. The user must be able to register for an account on the system.
   - 1.1. The system must display text fields for the user to enter his information.
     - 1.1.1. The text field must consist of username.
     - 1.1.2. The text fields must consist of email.
     - 1.1.3. The text fields must consist of password.
     - 1.1.4. The text fields must consist of phone number.
   - 1.2. The user must fill in all of the text fields before clicking 'Register' button.
     - 1.2.1. The system must verify the fields when the user clicks the 'Submit' button.
   - 1.3. The system must verify the fields filled in by the user before creating an account.
     - 1.3.1. The username must be unique for all users of this system.
     - 1.3.2. The email given has never registered in the system.
     - 1.3.3. The email given is of a correct email format.
     - 1.3.4. The password given must contain at least 1 uppercase character and 1 special character.
     - 1.3.5. The password given must contain at least 8 characters.
     - 1.3.6. The phone number given must be a verified phone number.
       - 1.3.6.1. The system must send a OTP to the user's phone number.

1.3.7. The system must provide error messages describing the reason the user's information is rejected.

1.4. The system must create an account for the user upon verification.

1.5. The system must log the user into the main page of the system.

4.2. Login

4.2.1 Description

After successful registration, the user must log into the system in order to access member features.

4.2.2 Response Sequences / Use Case

Use Case ID:	L1		
Use Case Name:	Login		
Created By:	Loh Sijing Aloysius	Last Updated By:	
Date Created:	28/08/2022	Date Last Updated:	

Actor:	User
Description:	User must log in to access features of the application
Preconditions:	User has registered for an account
Postconditions:	User has logged in
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User inputs their username and password into the respective text boxes. 2. User clicks on the login button 3. System verifies the user's credentials from the database 4. System displays the home page
Alternative Flows:	L1-AF-S2 If username not found: 1. Display "Incorrect username or password" 2. System returns to Step 1 L1-AF-S2 If incorrect password is entered: 1. Display "Incorrect username or password" 2. System returns to Step 1
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	1. User is connected to the Internet
Notes and Issues:	Nil

4.2.3 Functional Requirements

1. The user must be able to log into the system.

1.1. The system must display text fields for the user to enter his information.

1.1.1. The text fields must consist of email.

        1.1.2. The text fields must consist of password.
    1.2. The user must fill in all of the text fields before clicking 'Log In' button.
        1.2.1. The system must verify the fields filled in by the user before logging the user into the system.
    1.3. The system must log the user if the information obtained from the text fields are verified.
        1.3.1. The email must be found in the database of the system.
        1.3.2. The password entered must match the password of the user.
    1.4. The system must log the user into the main page of the system.

4.3. Create Room

4.3.1 Description

The host can create a room to invite other members to the event.

4.3.2 Response Sequences / Use Case

Use Case ID:	CR1		
Use Case Name:	Create Room		
Created By:	Loh Sijing Aloysius	Last Updated By:	
Date Created:	28/08/2022	Date Last Updated:	

Actor:	User
Description:	User creates a room for the event being hosted
Preconditions:	User must have logged in
Postconditions:	System displays the room that the user has created
Priority:	High
Frequency of Use:	Medium
Flow of Events:	1. User clicks on the "Create Room" button 2. User enters the name, date, time and venue of the event 3. System generates a unique URL and QR code for the room 4. User has the option to send invitations to other users by clicking on the "Send Invite" button
Alternative Flows:	Nil
Exceptions:	Nil
Includes:	Send Notification
Special Requirements:	Nil
Assumptions:	1. User is connected to the Internet
Notes and Issues:	Nil

4.3.3 Functional Requirements

1. The host must be able to create a room.
    1.1. The system must display text fields for the user to enter event information.
        1.1.1. The text field must consist of event name.

1.1.2. The text field must consist of event date.
1.1.3. The text field must consist of event time.
1.1.4. The text field must consist of event venue.

1.2. The system must be able to generate a unique URL for the specific room.
1.3. The system must be able to generate a unique QR Code for the specific room.
1.4. The host must be able to invite at least 1 other member(s) to the room.
1.5. The system must be able to send a notification to the invitee.
1.6. The system must be able to track attendance of invitee once invitee has indicated their attendance.
1.7. The system must be able to store attendees' list of items to be brought to the event.
1.8. TBD - add end event function

4.4. Join Room

4.4.1 Description

Upon receiving an invitation, members can choose to accept or decline to join a room created by a host.

4.4.2 Response Sequences / Use Case

Use Case ID:	JR1		
Use Case Name:	Join Room		
Created By:	Loh Sijing Aloysius	Last Updated By:	
Date Created:	28/08/2022	Date Last Updated:	

Actor:	Users
Description:	User join a room which have been created by a host.
Preconditions:	User must have logged in and received an invitation to the room for the event
Postconditions:	User will have access to the room
Priority:	High
Frequency of Use:	Medium
Flow of Events:	1. User either accepts or declines the invitation sent by the creator of the room 2. System adds the user into the room
Alternative Flows:	JR1-AF-S1 If user declines the invitation to the room: 1. System requests for user to enter reason for declining invitation 2. Invitation is dismissed
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	1. User is connected to the Internet
Notes and Issues:	Nil

4.4.3 Functional Requirements

1. The invitee must be able to choose to join a created room.
   - 1.1. The system must display event information for the created room.
     - 1.1.1. The event information must consist of event name.
     - 1.1.2. The event information must consist of event date.
     - 1.1.3. The event information must consist of event time.
     - 1.1.4. The event information must consist of event venue.
   - 1.2. The system must display option buttons for the invitee to indicate attendance.
     - 1.2.1. The option button must consist of 'Accept'.
     - 1.2.2. The option button must consist of 'Decline'.
   - 1.3. The system must add the invitee to the room once he clicks 'Accept' button.
     - 1.3.1. The attendee must be able to indicate list of items to be brought to the event if he clicks 'Add Food' button
     - 1.3.2. The system must be able to send a reminder notification to the attendant 1 day prior to the event.
   - 1.4. The system must display a text field if invitee clicks 'Decline' button.
     - 1.4.1. The text field must contain reasons for declining invitation to event.

4.5. Split Bill

4.5.1 Description

After the event has passed, all the attendees will be notified the amount they have to pay the other attendees and host.

4.5.2 Response Sequences / Use Case

Use Case ID:	SB1		
Use Case Name:	Split Bill		
Created By:	Cheung Zhi Heng	Last Updated By:	
Date Created:	31/08/2022	Date Last Updated:	

Actor:	User
Description:	When the event has finished, every user will be notified of the amount they need to pay to other users.
Preconditions:	User is logged in to the application and attended an event.
Postconditions:	User is able to see the amount that the user owes to the host or other individuals.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User will click on the "Split Bill" button. 2, System will display text fields (item brought, cost of item) for attendees to enter total cost of items purchased for the event.

	3. User will key in the fields and click “Submit”. 3. System will generate a list of how much each user owes the other users. 4. User will be able to view personally how much the user owes the other users.
Alternative Flows:	SB1-AF-S3 If item brought is empty: 1. System displays “Please fill in an item.” 2. Return to Step 2 SB1-AF-S3 If cost of item is empty: 1. System displays “Please fill in cost of item.”. 2. Return to Step 2
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	1. User is connected to the Internet.
Notes and Issues:	NIL

4.5.3 Functional Requirements

1. The system has to notify the attendees of their costs to let them pay once the event has passed.
   1.1. The system must display text fields for the attendees to enter cost information of the items they are purchasing for the event.
      1.1.1. The text field must consist of item brought
      1.1.2. The text field must consist of cost of item
   1.2. The system must list the amount each attendee owes the host and other attendees.
      1.2.1. The system must be able to calculate the individual cost each attendee owes the others.
      1.2.2. The system must display the amount owed each attendee.

4.6. Pay Bill

4.6.1 Description

Attendees can pay the amount they owe to other attendees after the event has ended.

4.6.2 Response Sequences / Use Case

Use Case ID:	PB1		
Use Case Name:	Pay Bill		
Created By:	Cheung Zhi Heng	Last Updated By:	
Date Created:	31/08/2022	Date Last Updated:	

Actor:	User

Description:	User is able to pay the amount that the user owes to other users after the event has ended.
Preconditions:	User is logged in to the application and attended an event.
Postconditions:	System shows a “Paid” button after they have successfully paid to the specific user.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. System will display the list of users which the user owes money to. Each user will have the names of the people and the amount of money the user owes them. 3. User will select the specific user to pay money to. 4. System will connect to the payment API to transfer the money to the user. 4. User clicks on the tick button. 4. System displays a 'Paid’ button after payment is made successfully. 5. System updates the list of users the user owes money to.
Alternative Flows:	NIL
Exceptions:	E1. System goes down/Unsuccessful Payment - In the case where the application has system error or there is a bug : 1. System displays an error message. 2. User will try again and select a specific user to pay money.
Includes:	Send Notifications
Special Requirements:	NIL
Assumptions:	1. User is connected to the Internet.
Notes and Issues:	NIL

4.6.3 Functional Requirements

1. The attendee must be able to choose who to pay from the list of attendees they owe money to.
   - 1.1. The system must display the list of other attendees whom the current attendee owes money to.
     - 1.1.1. The list of other attendees must consist of attendee name.
     - 1.1.2. The list of other attendees must consist of amount owed to that specific attendee.
   - 1.2. The system must display a ‘Paid’ button for attendees to pay once they have transferred the amount to the other attendee.
     - 1.2.1. The system must remove the other attendee from the list of attendees.
     - 1.2.2. The system must add the transaction details to both attendees’ transaction history.
     - 1.2.3. The system must notify the other attendee that the bill has been paid.

4.7. View Transaction History

4.7.1 Description

Users of the app must be able to see their transaction histories.

4.7.2 Response Sequences / Use Case

Use Case ID:	TH1		
Use Case Name:	View Transaction		
Created By:	Cheung Zhi Heng	Last Updated By:	
Date Created:	31/08/2022	Date Last Updated:	

Actor:	User
Description:	User is able to view the transaction histories that they have made previously in the application.
Preconditions:	User must have logged in to the application and made a transaction previously.
Postconditions:	System displays who and how much they paid for all previous events that the user has participated.
Priority:	Low
Frequency of Use:	High
Flow of Events:	1. System displays the transactions that has been made previously in the application. 2. User can click and view details of each transaction which contains the date and event name.
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	1. User is connected to the Internet.
Notes and Issues:	NIL

4.7.3 Functional Requirements

1. The attendees must be able to see who they have paid and who paid them in all events participated.
   - 1.1. The system must have a page for previous transactions done under this user account.
     - 1.1.1. The system must display the name of the person the user has paid to or received payment from.
     - 1.1.2. The system must display the amount that the user has paid to or received payment from.
   - 1.2. The system must include a dropdown for each transaction in the page to show more information.
     - 1.2.1. The system must display the date of each transaction.
     - 1.2.2. The system must display the event name.

4.8. Recommend Party Games

4.8.1 Description

Attendees browse through selected games and their descriptions to decide on what to play at their event.

4.8.2 Response Sequences / Use Case

Use Case ID:	PG1		
Use Case Name:	Game Recommendation		
Created By:	Tey Kai Seong	Last Updated By:	
Date Created:	31/08/2022	Date Last Updated:	

Actor:	User
Description:	User is able to access the list of recommended party games.
Preconditions:	User is logged in to the application.
Postconditions:	User is shown and chooses party games that the user want to play.
Priority:	Low
Frequency of Use:	Medium
Flow of Events:	1. User will click on the “Party Games” button. 2, System will display a catalog of up to 30 party games that the user can choose from.
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	1. User is connected to the Internet.
Notes and Issues:	NIL

4.8.3 Functional Requirements

1. The attendees must be able to read through descriptions of selected games.
   1.1. The system must display a browsing catalog of party games and their information.
      1.1.1. The system must display up to 30 game recommendations
      1.1.2. The system must display a description of up to 200 words of the game.

4.9. Recommend Nearby Party Snacks

4.9.1 Description

Members can search for nearby party snacks around a specified search radius by using his current location or entering a custom location address into the search field. The system will then retrieve a list of nearby options for the member.

4.9.2 Response Sequences / Use Case

Use Case ID:	PS1		
Use Case Name:	Recommend Party Snacks		
Created By:	Tey Kai Seong	Last Updated By:	
Date Created:	31/08/2022	Date Last Updated:	

Actor:	User
Description:	User is able to access the various nearby party snack options.
Preconditions:	User is logged in to the application.
Postconditions:	User can enter an address and system will then generate a list of options for which users can pick.
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	1. User will enter address on search field. 2, System will fetch and generate a list of nearby options within a 10km radius from Google Maps API. 3. System will display the price range, postal code and estimated travel time of each option.
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	1. User is connected to the Internet. 2. User's device has location service turned on. 3. System is connected to Google Maps API
Notes and Issues:	NIL

4.9.3 Functional Requirements

1. The system must display a text field for members to enter an address.
   - 1.1. The system must include a dropdown for members to select a radius.
   - 1.2. The system must display each nearby option on the map as a list.
   - 1.3. Members must be able to click on the list to view more information.
     - 1.3.1. The system must display the price range of the food option.
     - 1.3.2. The system must display the address and postal code of the food option.
     - 1.3.3. The system must display the estimated travel time to the food option.

4.10. Order Delivery

4.10.1 Description

Members have the option to order delivery for party food via the system.

4.10.2 Response Sequences / Use Case

Use Case ID:	OD1		
Use Case Name:	Order Delivery		
Created By:	Cheung Zhi Heng	Last Updated By:	
Date Created:	31/08/2022	Date Last Updated:	

Actor:	User
Description:	User will be able to order delivery if they are unable to locate any nearby locations for party snack.
Preconditions:	User is logged in to the application.
Postconditions:	User successfully orders delivery from the application and shows the delivery progress and details in the application.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	1. User will click on the "Recommend Nearby Party Snacks" button. 2, User may choose to click on the "Order Delivery" button. 3. User picks and chooses what to order for delivery as well as timing for delivery. 4. User then checks the information and clicks "Submit" button. 5. System then shows that it is searching for a delivery person to deliver to the user. 6. System shows successful delivery order and shows the delivery details.
Alternative Flows:	OD1-AF-S5: If no delivery person is willing to pick up: 1. System displays "Please try again.". 2. System returns to Step 4.
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	1. User is connected to the Internet.
Notes and Issues:	NIL

4.10.3 Functional Requirements

1. The system must display the "Order Delivery" button.

The system must retrieve a list of delivery options.

The system must display the list of delivery options.

The system must display the information regarding the delivery option.

The system must display the "Submit" button..

2. The system must make a request to Delivery API when members submit the order..

   The system must display the "Successful Delivery" message.

   The system must display the "Please try again." message if order is unsuccessful.

   2.1.1. The system must redirect members back to delivery options page.

4.11. Send Notification

4.11.1 Description

System will send notifications to members when they have an event coming up or still owe another member a bill

4.11.2 Response Sequences / Use Case

Use Case ID:	SN1		
Use Case Name:	Send notification		
Created By:	Tey Kai Seong	Last Updated By:	
Date Created:	5/09/2022	Date Last Updated:	

Actor:	User
Description:	User is able to receive and send notifications.
Preconditions:	User is logged in to the application.
Postconditions:	Users will receive a reminder notification when an event is coming up or a bill is still unsettled.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User will receive a notification within a time period specified by the host reminding the user about the event. 2. Users can click the send reminder button to remind other users regarding the unsettled bill. 3. System sends a notification to the specified other user.
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	
Special Requirements:	NIL
Assumptions:	1. User is connected to the Internet. 2. User has indicated attendance for an event.
Notes and Issues:	NIL

4.11.3 Functional Requirements

1. The system must send a notification to users within a host specified period reminding them about an upcoming event.

1.1 The system must include a dropdown for host to select a reminder period.

2. The system must send a notification to users reminding of outstanding bills.

    2.1 The system must display a 'Remind' button for users to click.

5. Other Nonfunctional Requirements

5.1. Usability Requirements

5.1.1. 85% of the users must be able to find the details of the nearby snacks within two minutes. The system must sort the list of snacks by distance.

5.2. Performance Requirements

5.2.1.The system must display the user's search results of snacks within 5 seconds.

5.3. Security Requirements

5.3.1. The system must make use of hashing algorithm to encrypt the user's password.
5.3.2. The system must only process requests for personal data with an authentication token.
5.3.2.1. The system must issue an authentication token to the user when the user successfully logs in.

5.4. Reliability Requirements

5.4.1. The app must load within 10s after launching the app.

Appendix A: Data Dictionary

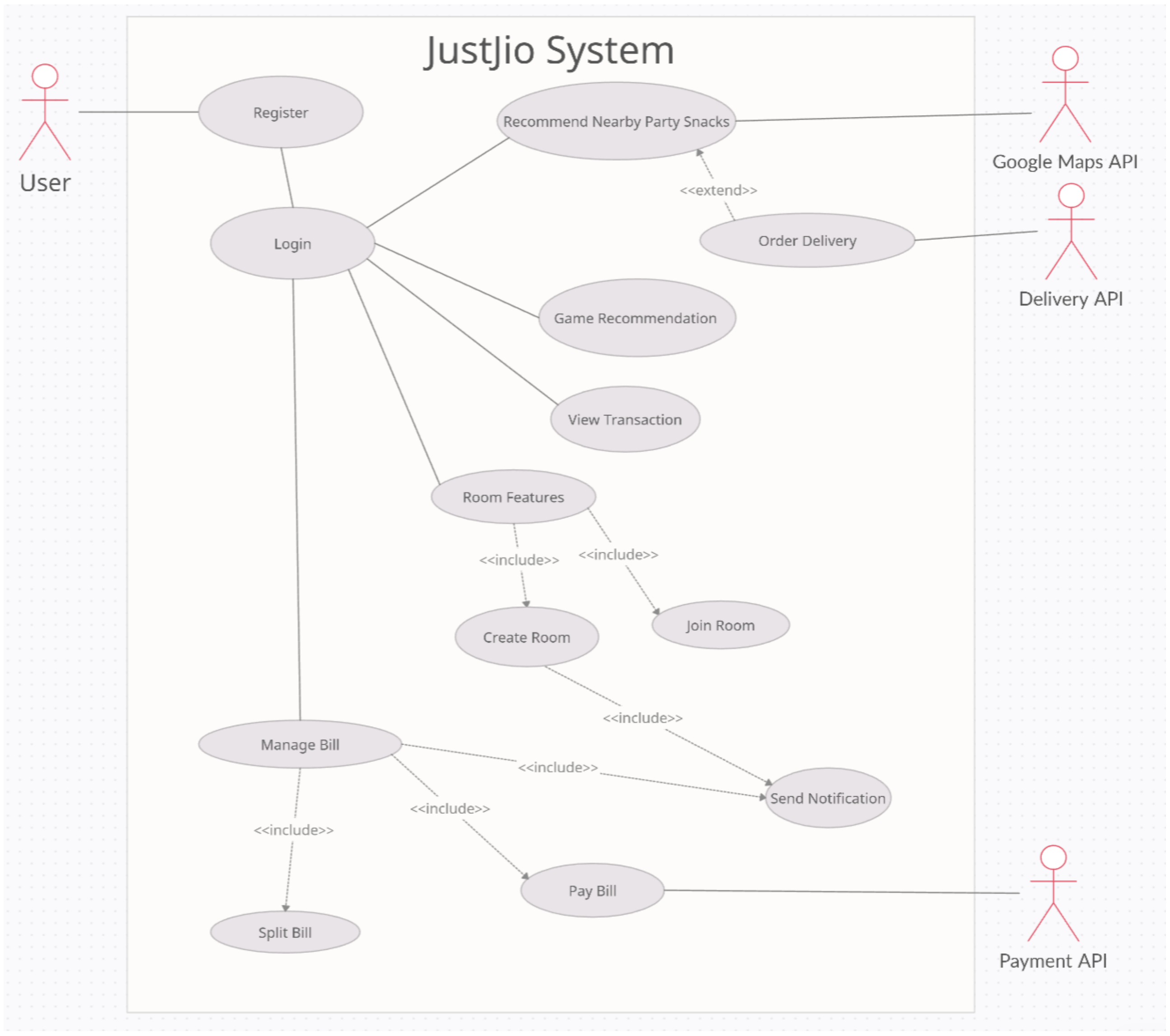
Term	**Description**
Front-end	This refers to the presentation layer of our software. In this context it is the U/I and U/X.
Back-end	The part of system that is not directly accessed by the user, responsible for storing and manipulating data
API	API serves as the intermediary between the front-end and back-end for transfer of data.

Login details	This consists of an unique 'username' created by the users consisting of a minimum 8 and maximum of 32 alphanumeric characters. A password is associated with each username to authenticate user identity.
User	The user pertains to anyone who registers and uses the system. External users that accept a user's invitation are referred to as attendees.
OTP	One time generated password, this is used as a secondary key to verify the identity of the user. The password is sent to the mobile of the user.
Room	A virtual space created for a specific event that contains the attendees and the host.
Member	A user who has registered for an account
QR Code	A generated pattern that can redirect users to a specific URL when scanned
URL	The webpage address used to reference a room or the profile of an user that is provided to an potential attendee
Invitee	Invitee pertains to users of the system that have been invited to join the room created by the host.
Attendee	Attendee pertains to invitees that have been invited to join the room created by the host and accepted the invitation. These are the members that will be going to the event hosted.
Event	This can be any group of three (3) or more persons who have assembled or gathered together for a social occasion or other activity. Events include but are not limited to parties, gatherings , meetings and social meetups
Host	The user who has created the room pertaining to a certain event. This user also has control over inviting other users to the room and modifying the event information
Transaction history	Transaction history records all previous transactions done by the user. It can also cover in-depth details including but not limited to : name of receiver, amount sent, and date of transaction.
GPS	Global Positioning System. It is used to show the location of different places on the map.

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

- Use Case Diagram



Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>