System Requirements Specification

for

Cafe Bunny

SRS_REQ_001_V1.2

Version 1.2 approved

Prepared by Team CodeNation

Nanyang Technological University,

School of Computer Science & Engineering

26/03/2021

Submitted to:

Dr. Shen Zhiqi, Lab Supervisor

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Ong Yi Shen	19/02/2021	Chew Zhi Kang	20/02/2021	SRS(All Sections) 1st Draft
1.1	Chew Zhi Kang	20/02/2021	Ong Yi Shen	21/02/2021	SRS(All Sections) 2nd Draft
1.2	Ong Yi Shen	26/03/2021	Chew Zhi Kang	27/03/2021	Standardization

TABLE OF CONTENTS

1 Problem Statement

Cafe hopping is a popular activity among the younger demographic, it involves travelling from one cafe to another to try out their food, take pictures for social media and give reviews. By posting to social media, cafe hoppers are also showing off how many cafes they have visited. There is an opportunity to create an application that can help bridge the gap between cafe hopping and achievement hunting which can benefit both users and cafe owners alike.

2 Overview

2.1 Background

In in era where cafe food are becoming increasingly popular as people are willing to venture out of their neighbourhood to try out new cafes or even to neighbouring countries like Malaysia, there is a need for a mobile application that is able to connect and relate to cafe hoppers by allowing them to keep track of the cafes visited before and allow for quick access to the reviews of the cafes. Currently, there is no mobile application that taps on the interests of cafe hoppers and our application will enrich their cafe hopping experience and further encourage more people to join in this trend.

2.2 Overall Description

Cafe bunny is an application that can assist users with cafe discovery, get information on nearby cafes such as their menus, photos and reviews and show off their achievements. Incorporating simple gameplay elements such as unlocking achievements when users visit cafes and adding a level system, users have an additional avenue to show off how many cafes they have visited to other users. Cafe owners can also provide discount codes to customers and the application can give cafes additional publicity. Hence, our application will not only benefit those who love cafe hopping, but also benefit the cafe owners who may not get enough publicity due to their secluded locations or marketing budget constraints.

3 Investigation & Analysis Methodology

3.1 System Investigation

Cafe bunny will use google's existing map api to display the map and get the locations of nearby cafes. Location of the user will be obtained from their mobile phone's GPS. Firebase Realtime database will be used to store user's information as well as the information of the cafes such as reviews and pictures. Flutter will be used to create the application. The application will be available on iOS and Android.

3.2 Analysis Methodology

3.2.1 Feasibility study and requirements elicitation

Organize a team to perform interviews at different cafes throughout Singapore to get feedback from cafe owners and cafe goers. This team can also contact well known cafe hoppers via their social media to get their feedback as well. After getting the results of the interviews, a Feasibility and Risk Assessment study will be conducted, taking into consideration the feedback from the interviewees. After conducting the field study, adjustments to the requirements of the application will be made after identifying redundant or less popular features and adding in the more popular features recommended by cafe hoppers and cafe owners.

3.2.2 Prototyping

A prototype containing the core features of the app will be developed within 8 weeks. This prototype will be used as a proof of concept only. Further feedback from the prototype will be taken into consideration and integrated into the fully complete app. Feedbacks can include ways to improve on the User Interface experience such as the positioning of the buttons and the intuitiveness of the flow from one page to the other.

4 Constraints

4.1 Project Schedule

There is a 8 week timeframe to implement a working prototype for a presentation to stakeholders. Thereafter, refinements and improvements to the prototype will be made taking into account the suggestions from the stakeholders.

4.2 Potential abuse of allowing users to upload pictures

There is the potential of inappropriate content to be uploaded via the application. A team of moderators will not be sufficient when the number of users of the application increases. An automated system that automatically flags offending images will have to be implemented once the number of users grows. This is to ensure that our application is child-friendly whereby young children are also welcomed to use the application without having to worry about being exposed to inappropriate content on our application.

5 Operational Requirements

5.1 Application Services and Technical Support

Programmers and developers will require access to the source code and database to provide support for bugs and technical issues that might occur. The team will document any reported technical issues in a problem report.

5.2 Routine backup of database

The backend Firebase Realtime Database will have its data routinely backed up to a Google Cloud Storage bucket every 1 month to allow for the restoration of data in the event of any data loss. This is to increase the reliability of our application as we know that the reviews and images uploaded are important for cafe hoppers, a loss of such vital information on our application could damage the trust of our users and possibly reduce the user base of our application.

5.3 Help desk support

There will be a help desk available with an operational time of 10am - 6pm from Monday to Saturday to provide assistance to users and cafe owners and answer any inquiries. Any technical issues received by the help desk will be forwarded to the technical team. This will ensure that we are able to receive feedback from our users as soon as possible to reduce the application downtime in correcting any major bugs. It will also provide users with added trust of our application as help can be provided readily.

5.4 Scalability

The system must be able to rapidly scale up to meet the demands of the users. Additional servers must be able to be quickly deployed in the event of increased user demand. This is to reduce the possibility of losing new users when we launch the application as it is important to portray a good image of our application to new users that our application is reliable and matches their interest.

5.5 Moderation Team

A moderation team will be required to assess reviews or pictures that have been flagged to be unsuitable for the app. They can then decide whether to keep or remove the offending review or picture. By keeping watch on the content posted to the application, we can ensure that our application will be child friendly and prevent our application being flagged as inappropriate on Google Play Store or Apple Appstore.

6 Functional Requirements

6.1 Account Sign Up

The system must allow users to sign up for an account.

1. The user must input their username and password.
2. The user must select the roles of either Cafe Hopper or Cafe Owner.

6.2 Log In

The system must allow users with an account to log in.

1. The user must log in as either a Cafe Hopper or Cafe Owner.
2. The user must enter his or her username and password in the login.

6.3 Display Map

Using google's map API, a map must be displayed based on the user's location. All nearby cafes must be marked by waypoints that are color coded.

6.3.1 Cafe Waypoints

Cafe waypoints must be coloured differently depending on whether the user has visited the cafe before. Visited cafe's must be coloured blue while unvisited cafe's must be coloured red.

6.3.2 Player Marker

A marker must be placed in the center of the map that shows where the user currently is.

6.4 Display Achievements and levels

As users visit cafes, their level must increase and they can unlock achievements. Achievements can also unlock discount codes which they can use at cafes.

6.4.1 Achievement Page

A page that displays all the achievements that the player has earned and achievements that have yet to be earned.

6.5 Cafe Page

Clicking on a cafe waypoint on the map must display the cafe page. The cafe page contains information on the cafe such as contact information, address, pictures that other users have posted and their reviews.

6.5.1 Posting Cafe reviews and pictures

Users can post their reviews of cafes they have visited. Users that are of a higher level will have their reviews shown first. Users can also post pictures of the cafe renovation and the cafe food.

7 Input Requirements

7.1 Login

Users must login with their username and password before they have access to the application's features.

7.2 Cafe Reviews

Users can leave a written review of cafes that they have visited on the cafe page.

7.3 Cafe Pictures

Users can post pictures of the cafe and cafe food onto the cafe page.

7.4 Cafe Hopper and Cafe Owner Account

Each type of user creates a unique username and password upon sign up for their account type.

8 Process Requirements

8.1 Google Maps API

The system must be able to communicate with Google Maps API to retrieve GPS and cafe information.

8.2 Data Storage

The system must be able to communicate with the Firebase Database to store and retrieve user information as well as cafe reviews and pictures.

8.3 Data Validation

The system must be able to properly validate the user's information during the login process and reject the requests if it is an invalid account.

8.4 Performance

The system must allow concurrent use on a 24-7 basis. The system must notify the backend of issues reported by users within 5 minutes.

9 Hardware Requirements

9.1 Network

Users must have access to Wi-Fi/Data connection with location turned on their device to use the application.

9.2 Memory

User devices should have at least 2GB of ram when using the application to ensure a smooth user experience. Recommended 1GB of free storage in order to download and run the application.

10 Software Requirements

10.1 Device (Client) Operating System

For Android Users, android version 9.0 and above is required.
For IOS Users, IOS 10.0 and above is required.

11 Deployment Requirements