

School of Computer Science and Engineering

CZ3002 Advanced Software Engineering

**System Requirement Specifications (SRS)**

Project Name: HangOut

****

Group Name: Mac & Cheese

Lab Group: TDDP1

|  |
| --- |
| **Group Members** |
| Jethro Phuah An Ping (Team Leader) |
| Lam Zhi Fah (Deputy Team Leader) |
| Alicia Chua Jieying |
| Ernest Ang Cheng Han |
| Ong Jing Heng Shaun |
| Ong Sim Hao |
| Shannon Tan Xinyi |

# 1 Table of Contents

[Table of Contents](#_heading=h.gjdgxs) 1

[Problem Statement](#_heading=h.30j0zll) 3

[Overview](#_heading=h.1fob9te) 3

[Background](#_heading=h.3znysh7) 3

[Overall Description](#_heading=h.2et92p0) 3

[Investigation & Analysis Methodology](#_heading=h.hnorbsrlquyz)5

[System Investigation](#_heading=h.3dy6vkm) 5

[Analysis Methodology](#_heading=h.p6zdwbssb5pd) 5

[Feasibility study and requirements elicitation](#_heading=h.9z8310e6f3f8) 5

[System analysis and requirements specification](#_heading=h.ptoszv7p9ioz) 5

[Object-oriented design using UML](#_heading=h.i80x26uji5ia) 6

[4.2.4 Prototyping](#_heading=h.84t2jp8r5f0r) 6

[**Constraints**](#_heading=h.iupei4mxh15r)7

[Maintenance](#_heading=h.lnxbz9) 7

[Language](#_heading=h.35nkun2) 7

[Proprietary hardware and software](#_heading=h.1ksv4uv) 7

[Project Schedule](#_heading=h.2jxsxqh) 7

[Operational Requirements](#_heading=h.z337ya) 8

[Help Desk Support](#_heading=h.3j2qqm3) 8

[Application Services and Technical support](#_heading=h.1y810tw) 8

[Administration Features](#_heading=h.4i7ojhp) 8

[System Interface of HangOut](#_heading=h.2xcytpi) 8

[Functional Requirements](#_heading=h.2bn6wsx) 9

[Community discovery and hangout](#_heading=h.qsh70q) 9

[Access and Modify User’s Profile](#_heading=h.5cyeqx2ldiox) 9

[Find Food Options](#_heading=h.gp7w9ovygww5) 9

[Find Places of Interest](#_heading=h.1rcs76ayjvz8) 9

[Events happening around them](#_heading=h.ja8wbdrvj40b) 9

[Create Meetups](#_heading=h.9f4ksww4f0x8) 9

[Join Meetups](#_heading=h.eb2f0zvs52lg) 9

[Admin’s Profile](#_heading=h.penko51wtpay) 10

[Add Places of Interest](#_heading=h.dmkbd4bh916h) 10

[Add Food Options](#_heading=h.g00e1fft3tjq) 10

[Add Event](#_heading=h.ay1t4qfmrioi) 10

[Input Requirements](#_heading=h.147n2zr) 10

[User account](#_heading=h.3o7alnk) 10

[Food Option Input](#_heading=h.23ckvvd) 11

[Places of Interest Input](#_heading=h.lldhgyc2h6uc) 11

[Events Input](#_heading=h.juncqvds3ppb) 11

[Create Meetup Input](#_heading=h.2inyj2dj2dh1) 11

[Join Meetup Input](#_heading=h.dqgxm2j8co96) 11

[Process Requirements](#_heading=h.32hioqz) 12

[Firebase transaction](#_heading=h.1hmsyys) 12

[Data integrity](#_heading=h.41mghml) 12

[Data validation](#_heading=h.2grqrue) 12

[Performance](#_heading=h.vx1227) 12

[Data repository](#_heading=h.3fwokq0) 12

[Output Requirements](#_heading=h.1v1yuxt) 13

[List of Favourites](#_heading=h.4f1mdlm) 13

[Meet-up summary and confirmation](#_heading=h.50f7q8l0vvxn) 13

[Hardware Requirements](#_heading=h.3tbugp1) 14

[Network](#_heading=h.28h4qwu) 14

[Client Computers](#_heading=h.nmf14n) 14

[Mainframe](#_heading=h.37m2jsg) 14

[Production support systems](#_heading=h.1mrcu09) 14

[Software Requirements](#_heading=h.46r0co2) 14

[Client Operating Systems](#_heading=h.2lwamvv) 15

[Client Application](#_heading=h.adw69qk1s0w) 15

# Problem Statement

A survey conducted in 2020 reported that 8 in 10 Singaporean employees experience burnout from work (Singapore Business Review, 2020). This comes as no surprise, given that Singapore was placed 32nd out of 40 countries when it came to work-life balance in a study recorded in 2019 (Joanna Saw, 2019). Singapore was also ranked 2nd in terms of hours worked per week, clocking in 45.6 hours, second only to Hong Kong (50.1 hours) (Channel News Asia, 2018). To make matters worse, there is a phenomenon of an “always on” culture permeating in Singapore; The rapid growth of technology has made it a norm for employees to be able to perform work tasks through their technological devices at any time of the day. Given these statistics, it is clear that a large majority of the Singapore working population are constantly preoccupied with their work and lack time for social interactions with family and friends.

The COVID-19 pandemic has also done nothing in enabling the social relationships of Singaporeans. According to reports by The Straits Times in 2021, the extended circuit breaker imposed by the government in 2020 resulted in social isolation amongst many Singaporeans (A-Star, 2021). Alarmingly, there is a significant and unique association of different social isolation indicators with depressive symptoms in community-dwelling adults aged 21 and above (Ge L, Yap CW, Ong R, Heng BH, 2017). What this means is that the social isolation that arises from strict government regulations can potentially lead to depression amongst Singaporeans in the long run.

Therefore, the workaholic culture in Singapore, compounded by the strict government regulations against COVID-19, mean that the average Singaporean has little to no time at all to engage in social activities. The negative implications that social isolation has on one’s mental health means that there is an urgent need to promote social interaction amongst Singaporeans.

# Overview

## Background

As we are more than a year into the COVID-19 pandemic in Singapore, Singaporeans are starting to experience cabin fever and COVID-19 fatigue. Many have aired their frustrations over being unable to travel and lament about how Singapore is lacking in activities and sights to see. While Singapore may be small, there are actually many undiscovered hidden gems and activities to do. However, there are currently few applications or websites which compile such places and events.

Currently, there are various websites and social media pages which provide either places of interests, events or food locations, but never all 3. At the same time, such pages are often not filtered based on a specific neighbourhood, but contain locations island wide. This makes it difficult for citizens to have a bird’s eye view of what is happening in their neighbourhoods. Hence, there is an urgent need to develop such a one-stop application in order to improve the quality of life and improve Singaporeans’ mental states. The application will not only compile the list of places of interests in the neighbourhoods, it will also encourage cohesion and social interaction.

## Overall Description

In essence, our app, Hangout, will contain 4 main features - Event finder, Places-of-interest finder, Food finder, Meet-up Initiator. It provides the interface to the main realtime database system which contains places of interest, upcoming and current events, as well as food locations. It also provides an interface to allow authorised administrators and personnel to update the database in real time.

Singaporeans will be able to learn that our neighbourhoods and communities are vibrant and full of excitement, where they are able to unwind after a hard day at work.

# Investigation & Analysis Methodology

## System Investigation

The application is based on the idea of providing ease for finding hotspots around the area. As such, HangOut will take the desired destination of the user and display a wide range of food locations, popular places of interests and upcoming events nearby. Users are also able to create or join meet-ups with other users at a desired destination. As each account is tagged to a unique email address and handphone number, this ensures that individual users will be notified easily through email or handphone messages and this can be done via verification of user information with the database. Each user’s list of favourite locations will also be stored in the database.

## Analysis Methodology

Before any development takes place, it is important that we carry out feasibility studies and requirements elicitation as part of the requirements specification. Requirements elicitation will consist of strict reviews and validations in order to ensure correctness and consistency. Following which we will also utilise an object-oriented approach to analyse the problem statement and identify the main scope of the project, as well as list down any potential limitations and challenges that we might face. The requirements will then be formalised through Use Case documents containing corresponding information and supported by Use Case diagrams.

### Feasibility study and requirements elicitation

Focus groups will be conducted to foster an open and honest dialogue with our customers. By communicating our project directly with our target customer group, we can develop an in-depth understanding of customers’ reactions and responses to the products and solutions we aim to provide. In the case that customers are unable to find time to join these sessions, we can do simple customer surveys to receive feedback, which serves as an alternative to focus groups.

As many people often seek product recommendations online through social media platforms, we can carry out social media listening to understand our customer needs better and extract relevant information to compute the feasibility of our solutions. Furthermore, we also plan to gain insights from other similar products on social media and application stores by reading their customer reviews and feedback, which will help to keep our product expectations in line with our customers’ expectations.

We can also carry out online research to help give us a better idea as to why a customer might feel a certain way about our product and how to modify the product to complement the customer needs that are identified by the earlier methods. With in depth analysis and background information of the industry, we can understand the root of the issue and tackle it.

### System analysis and requirements specification

Analysis methodologies involve business analysis, requirement analysis, data analysis, process analysis, and application architecture:

* Business analysis – State the business rules, business system interfaces, business function, business ownership, sponsorship and associated project budget requirement
* Requirement analysis – System I/O description, user requirement definition, functional and security requirement
* Data analysis – Involve data collection process, data validation, data storage, manipulation and retrieval
* Process analysis – Data/process flow analysis, process decomposition and system interfaces
* Application architecture – Analyze application information structure, usability, user interface design, interaction and application implementation.

4.2.2.1 Perform an analysis of the problem using object-oriented techniques

An external view of the enterprise model of the system will be developed using Unified Modeling Language (UML). This System Requirement Specifications document will form part of the documentation for the project. Some desired features of the new system include:

* The ability to add places of interest, food locations and events as an admin
* The ability to search/view places of interest, food locations and events as a public user
* Allow users to create or join meetups with other users
* Enable users to add desired destinations to favourites in profile

### Object-oriented design using UML

A detailed object-oriented design for the registration system will be developed. UML will be used again for the graphical representation and documentation of the design. Our application will concern itself with the 4 main features of finding food places, finding places-of-interest, exploring events and creating meet-ups. Users will also be able to create an account through registration for ease of retaining information and also customising their preferences. Users’ information will be secured with their email and password once they have registered for an account (see Use Case Diagram in section 14.1).

### 4.2.4 Prototyping

The Object Oriented Rapid Prototyping (OORP) method will be used to implement a limited and functional prototype for the Hangout application. The prototype will be a working example of part of the system for demonstration and proof of concept purposes only. The prototype will be presented to the implementation team for analysis and once the actual requirements are understood, the prototype is to be discarded. The actual system is developed with much clearer understanding of the user requirements.

# Constraints

## Maintenance

Admins from the GRC play key roles in ensuring users receive the latest information on the neighbourhood as well as making improvements to the user interface of the application. By providing an admin interface, updating of locations of interest can be done easily. Moreover, our source code supports modularity, high cohesion and low coupling, which means the development team can easily make changes and improvements to the code, without worrying about crashing the system. Both the admins and development ensure we are able to maintain the usability and accuracy of the application.

## Language

The HangOut application currently provides one language interface. In further development, Hangout can consider including other languages.

## Proprietary hardware and software

HangOut will require a dedicated Internet Service Provider that is configured to handle up to 200,000 visits per day in order to be operational. The database utilised must also be able to handle and store big data containing user information without any losses or crashes. For clients, they are required to have an internet connection at all times on their mobile devices in order to access the features, and also have a personal contact number or email address for notification purposes.

## Project Schedule

There is a one-year timeframe to implement a production system of Hangout application from project commencement in time for its launch date in Fall 2021.

# Operational Requirements

## Help Desk Support

System users have a 24x7 access to telephone assistance for questions that are technical in nature, such as, slow or sluggish system response time, incompatible browser features, application errors, system downtime inquiries, account lock-out assistance, etc.

## Application Services and Technical support

All of our developers will have full access to HangOut source code and database system to ensure that our system can be maintained and improved whenever needed.

## Administration Features

System security and access levels are provided in the app. There are varying levels of system access and functional authority. Each public user’s access is limited to searching for food, events, places of interest and meetups. Only authorized system administrator(s) has access to edit the information stored in the database.

## System Interface of HangOut

The admin interface runs on the latest version of Chrome, Internet Explorer or Safari on Windows, Linux and Mac. The interface for public users can only login to the HangOut app with their registered account from one device at any point of time. Both interfaces will require the system to be online.

# Functional Requirements

The HangOut application allows the public to discover the vicinity around them and to initiate meet-ups with strangers. Registered GRC members would also have admin access to add more information into the database for the application.

## Community discovery and hangout

The public user can create an account to view and favourite all the available activities in their chosen vicinity. All system (browser) interfaces are based on ISO accepted industry standards for the [WWW.](http://www/) Among others the Hangout App will have the following functionalities:

### Access and Modify User’s Profile

* User’s home address
* User’s phone number
* User’s email address
* User’s favourite places, food and events to do
* User’s activity level with hangout

### Find Food Options

* View food options around user’s chosen vicinity
* Search for food options around user’s chosen vicinity
* Filter food options base on price, location, cuisines and dining type
* View selected dining’ address, menu, price and its opening days and hours

### Find Places of Interest

* View places of interest around user’s chosen vicinity
* Filter places of interest based on interest type
* View selected places of interest address, opening days and hours

### Events happening around them

* View list of upcoming events around user’s chosen vicinity
* View selected event address, time and signup fee (if needed)

### Create Meetups

* Create meetup around the chosen vicinity which shall include address, time of meetup, a short description of the meetup, a title for the meetup and images regarding the meetup (optional)
* Users who create the meetup can accept or reject another user’s request to join the meetup

### Join Meetups

* View a list of upcoming meetups around user’s chosen vicinity
* Search for meetups based on keywords
* View selected meetup’s address, time, location, description and a list of those who have signed up
* Request to join the meetup
* Receives a notification of entry if request is accepted by meetup initiator
* User will be added into the list of people in the meetup

Admins can create an account to add food options, places of interest and events onto the database.

### Admin’s Profile

* Admin’s Grass Root Committee (GRC)
* Admin’s phone number
* Admin’s email address

### Add Places of Interest

* Admin shall add the name, description, geolocation coordinates and images of the places of interest (optional)

### Add Food Options

* Admin shall add the name, description, geolocation coordinates and images of the food option (optional)

### Add Event

* Admin shall add the name, description, geolocation coordinates and images of the event (optional)

# Input Requirements

## User account

Each user is required to key in their desired email and password during registration. When logging into HangOut, the user must use the same email and password to login their account. Should the user forget their password, they may choose to reset their account password and login to their email to reset it.

## Food Option Input

The user will be able to select the “Food” option. The user will be given the option to select the cuisine of his/her choice. A filter option will be given to allow the user to filter according to price, dining type and cuisines (given a dropdown list for each). In the event that the user wants to search for a specific stall, he/she can manually search by keyword. Once the user finds his/her desired stall, he/she will be able to tap on it to view its address, available menu and opening days and hours.

## Places of Interest Input

The user will be able to select the “Places of Interest” option. The user will be given the option to select the category of his/her choice. A filter option will be given to allow the user to filter according to price and category (given a dropdown list for each). In the event that the user wants to search for a specific place, he/she can manually search by keyword. Once the user finds his/her desired place, he/she will be able to tap on it to view its address and opening days and hours.

## Events Input

The user will be able to select the “Events” option. The user can select from a list of upcoming events that are happening around the chosen vicinity. In the event that the user wants to search for a specific event, he/she can manually search by keyword. Once the user finds his/her desired place, he/she will be able to tap on it to view its address, time, registration link and sign up fee (if necessary).

## Create Meetup Input

The user will have to set a time for the meetup using a timer UI. The user will have to manually set the address, description, name for the meetup. The user can upload pictures relating to the meetup.

## Join Meetup Input

The user will be able to select the “Meetup” option. The user can select from a list of upcoming meetups that are happening around the chosen vicinity. In the event that the user wants to search for a specific meetup, he/she can manually search by keyword. Once the user finds his/her desired place, he/she will be able to tap on it to view its address, time and list of people who joined for each of the meetups. The user can also tap on the request button to request to join the meetup.

# Process Requirements

The following are among the inherent requirements that the HangOut app must be able to handle.

## Firebase transaction

The system must be able to send, receive and trigger transactions to the Firebase database system.

## Data integrity

Commit transactions that are completed and/or rollback unfinished or time-out transactions.

## Data validation

Data error from the user’s end and from the back-end database-processing end must be gracefully handled. There will be data validation and error-handling routines as part of the community discover app.

## Performance

Must resolve locking issues and handle concurrent use of the system on a 24x7 basis. Send, receive and display user messages to assist the overall user experience.

## Data repository

The Hangout app will maintain the existing Firebase database as the main repository of data.

# Output Requirements

## List of Favourites

Each registered user will be able to view a list of food places, places-of-interest and events that they favourited to view as a quick access.

## Meet-up summary and confirmation

Each registered user must have a view of a summary of meet-ups that they have requested to join. The summary will also be updated with confirmation of meet-up upon the request is approved. Initiators of the meet-ups will be able to view a list of users that have been confirmed to attend the meetup.

# Hardware Requirements

## Network

Since our interfaces requires the use of a hosted database service, Firebase, we will minimally require internet connection and the necessary network protocols set up in order for the proper functioning of our interfaces (e.g. DNS, HTTP, HTTPS, FTP, TCP)

## Client Computers

Any computer running a Desktop Operating System (Windows, MacOSX, Linux)

## Mainframe

Our database and mobile application does not require any environment. As for our web application, we have multiple options available to deploy our flutter based web application:

1. Platform as a Service (PaaS) - Netlify, AWS Elastic Beanstalk
2. Infrastructure as a Service (IaaS) - AWS Elastic Compute Cloud (EC2), Digital Ocean Droplets

As a result, we will not require any mainframe computer environments to host our applications

## Production support systems

For our mobile application, we will be building our source code into the respective Android and iOS file packages and hence, no further production support systems will be required.

For our web application, a load balancer may be needed to support high traffic load but this is mostly unlikely needed given that only few individuals will be tasked to administrators for their own areas. Additionally, it is extremely unlikely that administrators will simultaneously try to access the web application at the same time. Thus, a production support system for our web application is not required.

We are using a serverless Database and Backend as a Service - Firebase, which has different tiers to support different project types. Given the nature of our project, we will be using the free-tier and hence, our database may not be production grade, but it is definitely possible to modify and improve the performance of querying our database by purchasing higher and more premium tiers within the pricing options given by Firebase. Thus, no further production support systems will be required.

Continuous Integration and Continuous Deployment (CI/CD) pipelines maybe created to automate testing and integration of new features in the code base, as well as to automate the delivery and deployment of such features after testing into production

# Software Requirements

## Client Operating Systems

Our project will encompass 2 interfaces:

1. A web application interface for community administrators from their respective areas to perform Create, Read, Update, Delete (CRUD) operations on the area’s places of interests, food options and events for the users to browse through and discover
2. A mobile application interface for the end users to register and login to an account, where they will be able to view places of interests, food options and events in the vicinity they desire to browse through and discover. Additionally, they will be able to organise and/or join ‘meetup’s, a function which aims to encourage wholesome communal bonding especially within the restrictive nature and times of COVID-19

The web application should be accessible on any platform - Desktop, Mobile application. Hence, responsive web design should be taken into account and implemented during the development stage. This makes our web application compatible with most desktop (MacOSX, Windows, Linux) and mobile operating systems (iOS, Android).

Our mobile application should be compatible, downloadable and accessible by most mobile operating systems (iOS, Android)

## Client Application

We will be using Flutter to design both interfaces. For our mobile application, Flutter is a cross-platform mobile framework, allowing us to build our source code into Android Package Files (APK) or iOS App Store Package Files (IPA). Flutter also gives us the flexibility of building web applications and by using a common stack for both interfaces, it reduces the complexity of development.

For our backend and database, we will be using firebase which is a serverless database service provided by Google. By leveraging on the benefits of serverless technologies, it reduces the burden of us needing to design a backend service since this responsibility in itself is handled by Firebase (e.g. Authentication, Oauth, OpenID Connect).

* 1. Network system

Network software and protocols in order for systems to communicate with each other:

1. HTTP, TCP on port 80
2. HTTPS, TCP on port 443

We will be primarily using HTTP and HTTPS to connect to our interfaces as well as for communication between our interfaces and firebase.

12.4 Mainframe system

1. Google Firebase System
2. Flutter Framework System

12.5 Licenses

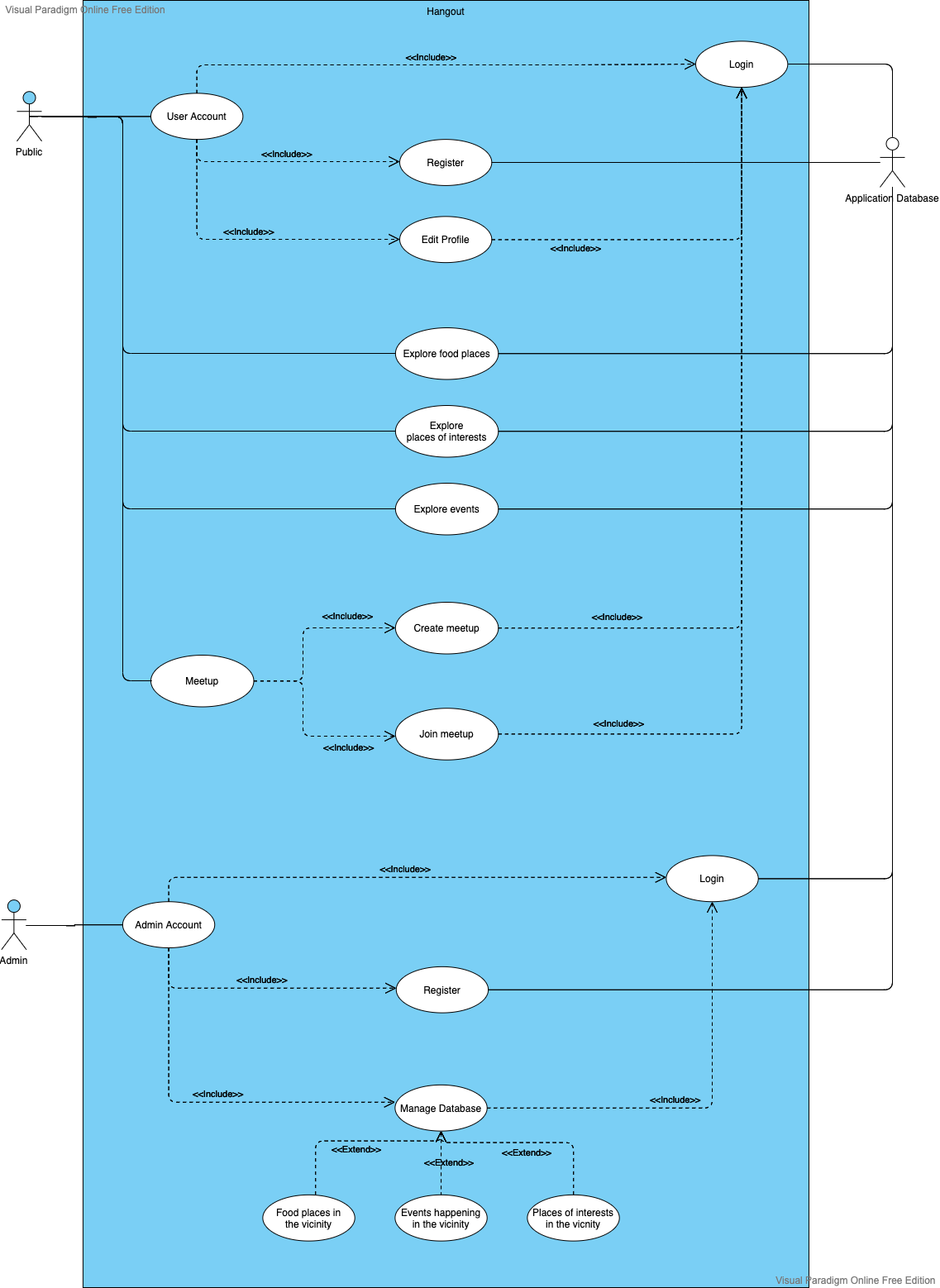
1. Google Firebase License
2. Flutter Framework License

Apart from the licenses involved in using the above technologies and services, we will be mainly developing our interfaces using Dart which is an open-sourced programming language developed by Google. Given its open-sourced nature, no further licenses is required

1. Deployment Requirements

See ‘14.2 Deployment Diagrams’ for more information

1. Diagrams
   1. Use Case Model



* 1. Deployment Diagram

