CLOUD-BASED MOBILE TRAVEL JOURNAL APPLICATION WITH MAP TRACING

BY

SHALON TEOH SIN YONG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

INFORMATION SYSTEM ENGINEERING

Faculty of Information and Communication Technology
(Perak Campus)

MAY 2017

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Cloud-Based Mobile Travel Journal Application with Map Tracing

Academic Session: 2017/05

I SHALON TEOH SIN YONG

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____ _____

(Author's signature) (Supervisor's signature)

Address:
Blk26D-6-4 Taman Kristal
Lengkok Erskine Mount Erskine
10470 Penang, Malaysia

Date: 21 August 2017 **Date**: _____

CLOUD-BASED MOBILE TRAVEL JOURNAL APPLICATION WITH MAP TRACING

BY

SHALON TEOH SIN YONG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology
(Perak Campus)

MAY 2017

DECLARATION OF ORIGINALITY

I declare that this report entitled "**CLOUD-BASED MOBILE TRAVEL JOURNAL APPLICATION WITH MAP TRACING**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :

Name : SHALON TEOH SIN YONG

Date : August 21, 2017

ACKNOWLEDGEMENTS

I would like to state my sincere thanks to all those who contribute to complete my final year project. I appreciate the advices and guidance given by my project's supervisor, Mr. Sun Teik Heng, who helped me to coordinate my project, report writing and evaluate project deliveries over and over.

I would also like to express my appreciation to my project's moderator, Ms. Wong Pei Voon who fairly commented and suggested crucial points to improve my project.

Last of all, my special gratitude goes to my family and friends who supported me with means of finance, unbeatable spirit, attention and time.

ABSTRACTS

Cloud-based mobile travel journal application with map tracing is a vision to renovate travel journal and brings great convenience to the travelers. The project was motivated by the fact that most of the travelers do not keep a detailed travel journal or do not document travel journal at all. The excuses are the concern with writing from scratch and claimed that current travel journal applications are not useful for their future reference. The proposed project intends to renovate the perception of people towards travel journaling.

This project is to be developed and managed with popular system development methodology – Scrum and Kanban methodology. Both methodologies support incremental and iterative development in a consistent and disciplined way. Where the system is to divide to smaller task according to user stories. Sprint backlog and Kanban board will be used to monitor development progress.

The proposed project aimed to be developed with current popular open source technologies. These include React Native framework, React framework and Mongoose MongoDB. With these frameworks, the proposed system is expected to deliver great user experiences and high-speed performance to complete task.

The main system feature, within 4 steps to create journal entry aimed to save time for travelers from finding a theme to write. Then travel rewarding feature allows travelers to keep track travel statistics and earn travel achievements at the same time. With the proposed system's entry filtering capability, look back to memory lane just made easier and faster.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SDK	Software development kit
HTML	Hypertext markup language
CSS	Cascading Style Sheets
APP	Application
JS	JavaScript
.NET	.NET Framework
API	Application program interface
REST	Representational State Transfer
ACID	Atomicity, Consistency, Isolation, Durability
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
XML	Extensible Markup Language
AJAX	Asynchronous JavaScript and XML
UI	User Interface
NOSQL	Non relational
SQL	Structured query language
Iaas	Infrastructure as a service
Paas	Platform as a service

Chapter 1: Introduction

1.1 Problem Statement

The practice of keeping a travel journal isn't common although traveling is trendy nowadays. The effort, time, and money spent on every single trip are wasted without preserving the memories. With the present of both physical notebook and digital travel journal applications, why people like travel enthusiasts are not moved to document such precious story of their life?

a. **Not everyone is a born writer.**

"I tried to maintain, but never seems to keep it up."

Travelers/Adventurers often get caught up in the day to day when get to a place and often lose sight of time for journaling or only able to journal a little for the trip. Most of the time people don't know exactly what to write down for their trip and having trouble identify themes for an entry. This is part of the reason travel blog and travel journal applications don't work. Not everyone is able to freely write from scratch and doesn't feel awkward when re-reads it. In this case, people would consider sharing a limited, but special moment on the popular social media platform like Instagram is better off than keeping a travel journal.

b. **Lacking the proof of achievement.**

"Even I document my trip; I can't simply show other my travel millstones because it is not great enough to acknowledge my travel achievements."

Users' input of travel data just remained as raw as inserted, not much information extracted from these data that can aid user to further understand their travel statistics. Furthermore, sense of achievement needs to be supported by having people know the marks a travel enthusiast left in the places of this world. However, these marks fade away as time goes by because the details documented are not

meaningful and detail enough to trigger the feels and experiences they've been through. Eventually, trip is just a trip; it does not constitute a long-term impact on the time and money spent.

c. Missing of handy filtering.

"I do document travel journal, but I find it difficult to read them."

Keeping travel journal using notebook, blog, existing apps or even social media is sometimes difficult to do a speedy trace back to a list of main destinations of the trip and where to find them. Common approach for journal entry trace back is based on published date time or country indexing. However, most often re-reading a journal is to do a speedy trace back on the name of places went, or to find out some crazy encounter at some place. Search result that is based on date time and country may not be enough to the user's intention. Hence, reader-friendly travel journal is the backbone of a great traveler.

1.2 Background and Motivation

These days, the chances to travel domestically and internationally increased immensely for civilized individual. In the 21st century, the world just became more affordable and convenient to visit with the assistance of technology for trip planning, booking, navigating, exploring, communicating, and documenting. (Jones) While travel writing has been a way to document travel and a source of inspiration since the 16th century. (Krook, 2013) People love telling others about their adventures because there are travellers who seek to mark achievements, and there are readers who simply enjoy experiencing the world through the lens of another human being. (Go2, 2017)

This is the age of the pocket computing, one of the greatest trend must be smartphone. (Lomas, 2016) The mobility of travel writing improved since the introduction of mobile application and networking. Having travel journal in the pocket is more convenient as ever. The advantage of creating multimedia travel journal with photos, videos, audio recording, Geo-tagging, and texts could make the convey of experiences more solid and powerful.

The development of this project is motivated due to the fading practice of keeping a travel journal by travel enthusiast and the lacking in existing solution to counter problems as stated in Chapter 1.1 that fellow travelers faced while practicing travel writing. This project is motivated to make contribution to the traveller community by providing them a convenient way to create journal entry, to fast trace entries, and to feel more achievements from previous travel and encourage more people to travel writing.

1.3 Project Objectives

a. **To develop guided 4 steps interface for detail travel journal entry creation.**
   This project allows users to publish a travel journal entry of their trip into the application within 4 steps procedure. The travel data entry includes date, start/end location with transportation used, emotion felt, weather encountered, medias, and experience in words. Simple and guided interfaces aim to encourage user to document more detail information from their trip especially for users who do not know what to write and users who prefer quick and easy way to write a meaningful detail entry.

b. **To motivate traveler/adventurer to use this mobile application for travel journaling by involving level reward to measure their travel achievement.**
   This project measures users past travel statistics and reward user with different level of traveler title based on points accumulated from entry posted, country traveled and user rate. Traveler title level is rewarded from bottom to top hierarchy: "Starter", "Holidaymaker", "Traveler", "Globetrotter", and "Voyager".

c. **To develop a travel journal application with map tracing feature.**
   This project provides interactive map tracing capability by displaying a number of entries available for locations and a list of journal entries based on location plotted on the map view. The objective is aimed to expose user to more interesting/potential travel destination and showcase the popularity of a destination through the number of entries created by other fellow travelers.

d. **To provide journal entry search filter for quick trace back.**
   This project allows user to perform travel journal entry traced back based on countries, emotion felt, weather encountered, and dates. This feature is first to the travel journal market, and thus aims to provide easier way for user to trace back to their memory lane.

1.4 Proposed Approach

The proposed approach in this project is to design and develop an improved cloud-based mobile travel journal application, which able to overcome current limitations of documenting travel trips. The target user of this project will be traveler, adventurer, and travel enthusiast with smartphone. The following describes system modules with titles and descriptions.

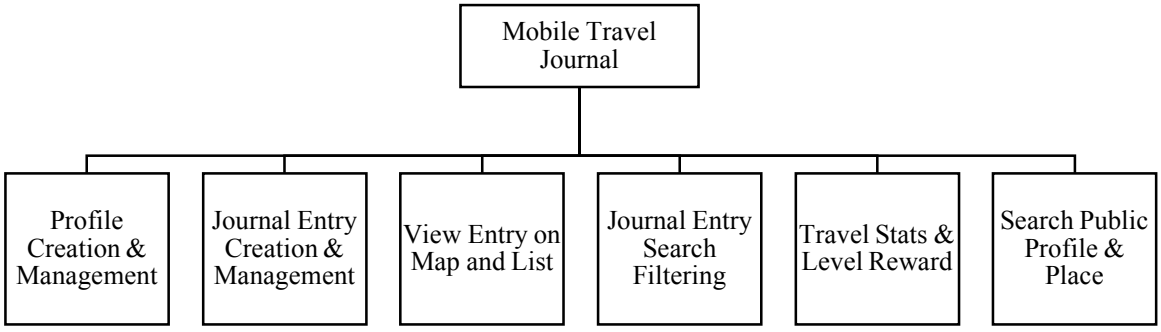


Figure 1-4-1 System Modules for Mobile Travel Journal Application

a. Profile Creation and Management Module

This module provides form fill-in interface for user's sign up, sign in, and profile edit interface also provided to let user make changes on their profile details. User is allowed to making changes on the profile picture, username, password, residence country, and bio-description.

b. Journal Entry Creation and Management

This module allows users to add, edit, view, and delete their own journal entry.

4 steps interface provided for new journal entry creation. User can choose to set privacy of each entry to either public or private during entry creation or update privacy through entry edit interface. Edit of entry also allowed for details including date time, location, emotion, weather, and text. The public user is only allowed to

view journal entry that is open to public. This protects owners right for their journal security and privacy.

c. View Entry on Map and List Module

This module provides user two ways to view entries. View entry on map will display a list of entries based on locations. The user is allowed to interact with the map to view journal entry on different location by gesture manipulating the map view interface. While view entry on list will display all entries sorted from most recent to oldest. User is allowed to vertically scroll through the list to view more journal entries.

d. Journal Entry Search Filtering Module

This module will allow user to narrow down the display of entry list with search filters. User is provided with an interface to start the search by choosing entry location, emotion, weather, and dates. This module does not provide a way to search entry using keywords.

e. Travel Stats and Level Reward Module

This module is responsible for keeping track of user's travel statistics and rewarding users with different level of title when they achieve more travel achievements. The public users can only view user's travel statistics if the profile is set to public.

f. Search Public Profile & Place Module

This module allowed the user to search and view related profile & place using keywords. All related public and private profile will show in the search result. However, public users have no access to view travel statistics and all journal entries of a private profile. If return result is a place, user is allowed to view all public journal entry that related that place through map view interface. User interaction with the map view is only restricted to the specific place.

1.5 Project Achievement

This project is about motivating traveller, adventurer, and people who wish to travel to keep a travel journal in a convenient and modern way. This project makes contribution by providing a mobile platform for them to document trip details, trace trip moments, and share trip experiences either privately or publicly.

This project also achieved by providing traveller community a convenient way to create journal entry, to fast trace entries using maps and handy indexes, and to feel more achievements by viewing travel summaries from previous travel and personal accomplishments after continuous traveling. Therefore, people can be motivated to keep travel journal that isn't time-consuming to write and useful for future trace back. This application not only preserve one's memories of the experience, but also do justice to the effort, time, and money spent on every single trip.

Furthermore, this project also contributes greatly to the travel enthusiasts by serving as an adventurous repository for them to discover and read interesting experiences from other travel enthusiast. Hence, this project achieved as a travellers' hub for travel enthusiast to find tips and information easily and immediately.

1.6 Report Organization

The headings of this report include table of contents, list of tables, figures, and abbreviations. The following is the arrangement of chapters and what is included as part of each chapter.

Chapter 1 identifies the problem statement, describes the project background and motivation, defines project objectives, states proposed approach, and project achievements.

Chapter 2 contains literature reviews on existing mobile travel journal applications, best mobile application development platform, best cross platforms development framework, backend technologies and project development methodologies. Comparison table is presented for each review to select the best approach.

Chapter 3 clarifies system design with system flowchart, use case diagram, activity diagram, class diagram, context diagram, level-1 data flow diagram, sequence diagram and system architecture diagram.

Chapter 4 discusses the methodologies, work procedures implemented in the project, and technologies involved to develop the system.

Chapter 5 states functional/non-functional requirements, and verification plan for system testing.

Chapter 6 concludes project achievement, objectives, challenges and future improvements.

References include a list of Harvard style citation that cite contents referred from other source.

Poster and Turnitin Plagiarism check result is included at the end of the report.

Chapter 2 Literature Review: Application Design and Development

2.1 Review on existing market travel journal application

This section highlights 5 existing travel journal applications from the market that are useful based on features, quality based on ratings, and popular based on reviews. This section also reviews on the applications' competitive advantages and drawbacks based on features and user experiences.

a. Bonjournal (Bonjournal LLC, 2015)

Bonjournal emphasizes a minimalist travel journal. The journals arranged in simple and clean view that can be shared through email, social media platform or PDF.

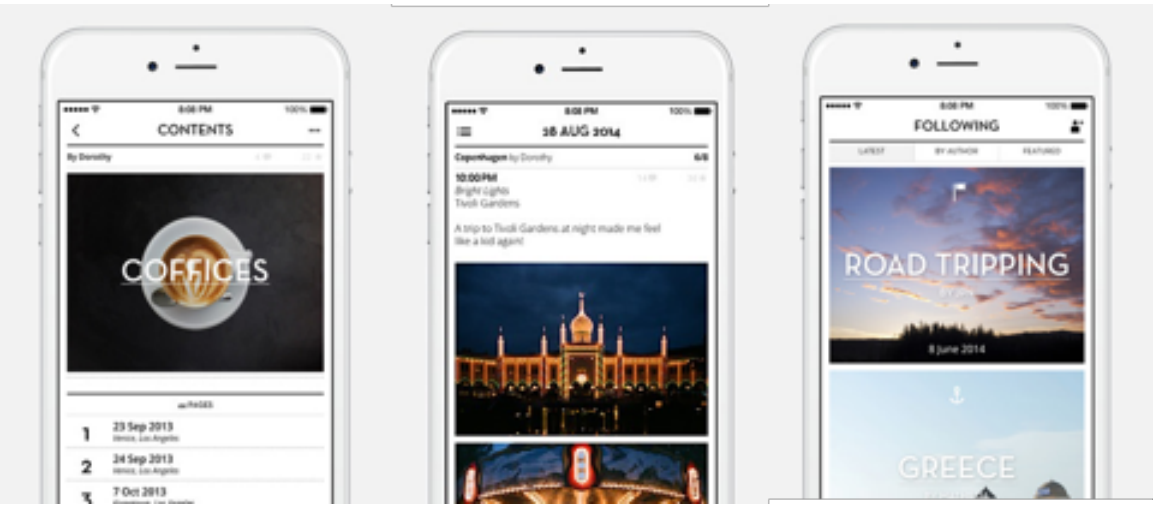


Figure 2-1-1 Bonjournal Screenshots

	Advantage	**Drawback**
Feature	- User can create posts offline and upload them later. - Privacy lock to keep post private.	- Lack of journal prompts during entry creation. - Moment trace back is difficult without search tool.
User Experience	- Useful and elegant entries organized approach to avoid long lists of item pile together.	- Only useful for users who express experience in words. - App colour palette is dull and tedious, only black & white.

Table 2-1-0-1 Advantages and Drawbacks of Bonjournal

b. Tripcast (Cluster Labs, Inc., 2016)

Tripcast is a travel journal application that emphasises on trip photo sharing. Tripcast let user share their travel photos with selected people who really want to see everything.

Figure 2-1-2 Tripcast Screenshots

	Advantage	**Drawback**
Feature	- Travel pals can build albums together so everyone's memories are in one spot. - New post notification to alert selected family and friends.	- Photos pin on map get over-crowded easily. - Moment trace back is difficult without search tool. Only allow sorting.
User Experience	- No more over posting anxiety.	- Not optimized for over-posting. Borderline spam. - Follower gets irritated with notification spamming.

Table 2-1-2 Advantages and Drawbacks of Tripcast

c. Mark O'Travel (Sergei Shpygar, 2017)

Travellers can create their own travel map by select country they've been or allowed the application to automatically fetch location info from photo metadata.

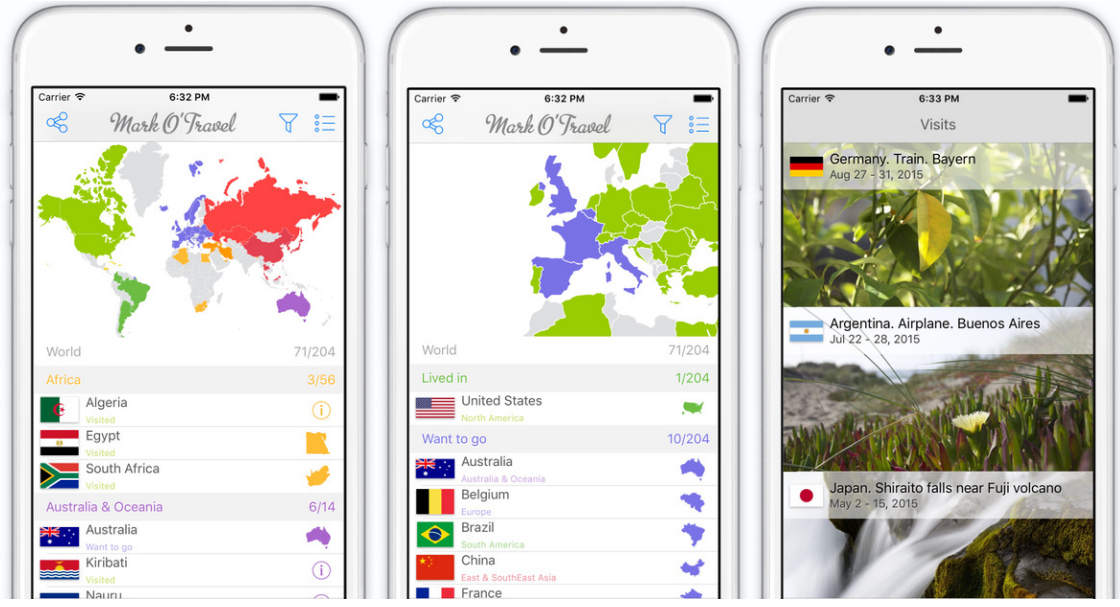


Figure 2-1-3 Mark O'Travel Screenshots

	Advantage	**Drawback**
Feature	- Auto-fetch location info from photo stored in photo gallery/ camera roll. - Keep travel stats on iCloud server and Dropbox.	- No pinning visit based on place. Only mark country. - No record of visit date.
User Experience	- Bird's-eye view on visited country - Ease of differentiating continents by coloured.	- Journal entry input is too simple and pointless. - User interaction with the map is restricted to country zoom.

Table 2-1-3 Advantages and Drawbacks of Mark O'Travel

d. **Polarsteps** (Polarsteps, 2017)

Polarsteps is an application that emphasises on automated travel blogging. Polarsteps automatically tracks user's travel route and places they've visited while travelling by just carrying the phone in the pocket.

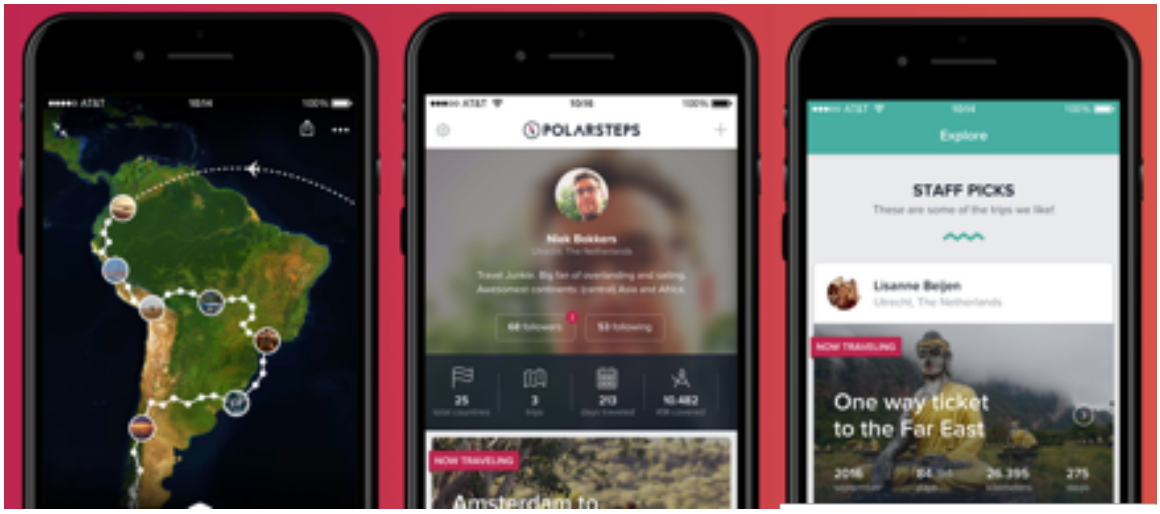


Figure 2-1-4 Polarsteps Screenshots

	Advantage	**Drawback**
Feature	- GPS works independently without data roaming and cell coverage. - Auto pin routes on Maps.	- GPS tracking can't pause, only able to start and stop. - View route summary only for 1 trip. - Journal entry input only for photos and texts.
User Experience	- Track and share journeys in an easy and visually appealing way. Kudos!	- Everything looks very big in the app. Too much dead space.

Table 2-1-4 Advantages and Drawbacks of Polarsteps

e. **Kid's Travel Journal HD** (Appcrawlr, 2015)

Kid's travel journal HD is an application that emphasises on fun and simple travel journal logging for kids to write their own travel journals.

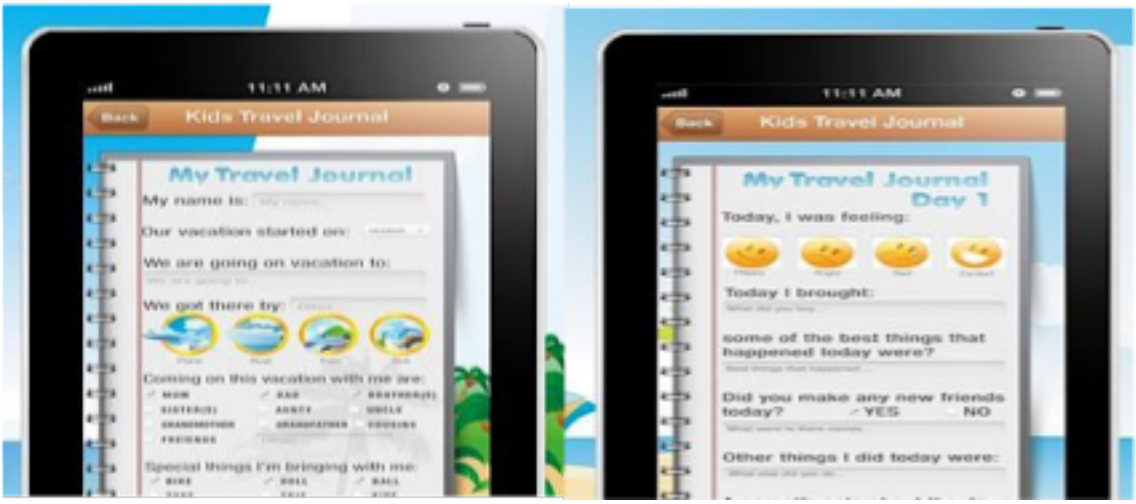


Figure 2-1-5 Kid's Travel Journal HD Screenshots

	Advantage	**Drawback**
Feature	- Journal entry creation with variety of prompts and hints. - Simultaneous multi-user journal.	- No Geo-location tagging. - No video input. - No cloud storage feature. - No sharing features.
User Experience	- Instruction for new entry creation is short and simple. - Easy navigation interface.	- Lack of active instruction to get user attention. Eg Animations.

Table 2-1-5 Advantages and Drawbacks of Kid's Travel Journal HD

2.2 Comparison between travel journal applications and Proposed Solution

	Bonjournal	Tripcast	Mark O'Travel	Polarsteps	Kids' TJ. HD	Proposed Solution
Price	FREE	FREE	$3.99	FREE	$1.99	FREE
Platform	iOS, Web app	Android, iOS, Web app	Android, iOS	Android, iOS	iOS	Android, iOS
Objective						
Detail travel journal	Text, location, photos, category, rating	Photos, videos, text,	Location, text, photos	Location, photo, text	Text, feeling, weather, photo	Location, feeling, weather, photo, video, transport, expense, text
Easy journaling						
Sense of achievements						
Ease to trace previous trip						

User-friendly interface						
Audience						
Explorer						
Adventurer						
Normal Traveler						
Tourist						
Feature						
Geo-location auto-tagging	No	No	Yes	Yes	No	Yes
Weather auto-tagging	No	No	No	No	No	Yes
Public journal discovery	Only featured journal	No	No	Only featured journal	No	Yes
Search people, place	Yes	No	No	Yes	No	Yes

Entries search filter	No	No	No	No	No	Yes
View entry by list	Yes	Yes	Yes	Yes	Yes	Yes
View entry on map	No	Yes	No	Limited for 1 trip	No	Yes
Quick view entry details on map	No	No	No	Yes	No	Yes
Keep travel stats	No	No	Yes	Yes	No	Yes
Badge rewarding	No	No	No	No	No	Yes
Offline mode	Yes	No	Yes	No	Yes	Save as draft.

Table 2-2-1 Comparison between Similar Applications

2.3 Critical Review on Similar Applications

a. Issues with starting to write a travel journal for every trip

The main issue about starting to write is the time consumed to form the words. All 5 reviewed travel journal applications use the combination of text and image as the main entry input. From a purely neurological point of view, the fact is that looking at an image can stimulate our brains to concept a story without taking much time. (Shah, 2014) However, pictures cannot replace words that brings the ability to deliver clear information. (Gaudiano, 2014) People who cannot write are still having a hard time to remind them to write down the essential details observed and experience undergo along with the emotion attached.

Unlike the applications reviewed, the proposed application can offer a structure for user to describe their experiences in detail, that allows user to look back their previous adventure, and still able to relive their timeless, irreplaceable memories. The benefit of the structure is user will not have to worry about what to write and time to write.

b. Issues with lack of long-term achievement from travel

"Satisfaction and goal achievement are important for voluntary behavioural change." (Taniguchi, Graas, & Friman, 2014) If a person loves travelling, then they might sometimes wonder "How much of the world I actually seen? How many countries, cultures & cities are out there, that I haven't explored yet?" To keep track of travel stats, one nice old fashion way is to pin marking every place on a world map and hanging it in the living room. (Stotz & Tack, 2014)

But what about the digital world? Only Mark O'Travel and Polarsteps offered travel statistics feature to allow user to track their travel data. For instance, how many countries they travelled? How many trips they've been to? How many days did they travel? Keeping travel stats is good however, the existing application does not acknowledge users' achievements. Acknowledging their achievements, even in a small way could actually

increase positive emotions such as appreciation, happiness, confidence, and of course possibly more urge to travel and write.

The proposed application keeps track of user's travel stats and also offers a level reward system to appreciate user' effort of travelling and writing by travel stats. The proposed application aims to encourage users to travel and create a journal entry for every place they went and hope to celebrate the users' achievement with them.

c. Issues with difficult to trace back previous travel trips

None of the reviewed application provides a feature to quick search to previous entries. Some applications do offer general search of all public and private entries, but that does not serve the purpose of quick trace back to user's memory lane. The benefit of quick tracing entries is for user to recall special moment or place or experience they documented. The proposed application offers entry search filter feature by allowing user to retrieve a list of entries according to country, emotion, weather, transportation, and dates. The proposed application provides convenient access for previous journal entry viewing. However, at this time there is no way to search using keywords.

Conclusion

To wrap up, the existing application still require some enhancement for journal entry input design and add on side-features like reward system and journal entry search filtering to attract more traveler to use travel journal application as their choice for trip documenting.

2.4 Review on Mobile Application Development Platforms

2.4.1 Comparison between three platforms

	Native	Cross-Platforms	Mobile Web
Programming skills to code for iOS & Android	Objective-C, Swift, iOS SDK, Java, Android SDK	HTML, CSS, JavaScript, Hybrid Framework	HTML, CSS, JavaScript, Web app framework
Common Framework	Swift, Java, Objective-C	Ionic, PhoneGaps, Xamarin, React Native, etc.	Angular, React, Ember, Backbone, etc.
Platform Distribution	App Store/ Google Play	App Store/ Google Play	Web
Development efficiency	Slow	Fast / Moderate	Fast
Development cost	High	Moderate	Low
Maintenance cost	High	Low / Moderate	Low
Graphical Performance	High	High / Moderate	Moderate
Native API			
Camera	Available	Available	(via web API)
Microphone	Available	Available	(via web API)
Contacts access	Available	Available	Not Available

Gyroscope	Available	Available	(via web API)
Accelerometer	Available	Available	(via web API)
File upload	Available	Available	(via web API)
Offline access	Available	Available	(via web API)
Geolocation	Available	Available	(via web API)
Gesture Navigation	Available	Available	Available
Push notification	Available	Available	(via web API)

Table 0-2-4-1 Comparison between 3 mobile application development platforms

2.4.2 Review on Best Used Platform for Proposed Project

a. Concerns for development speed & maintenance speed

The proposed project is looking for a platform that offers fast application development for both iOS and Android application. The goal is to write once, run everywhere. Note: Certain cross-platforms development framework is capable of offering fast development speed and maintenance speed.

Best used: Mobile Web / Cross-platforms

b. Concerns about user experience and graphical performance

The proposed project should provide good graphical performance and user experience to the user. However, this project does not require necessary high-speed performance, high graphical performance, and excellent user experience unlike games.

Best used: Native / Cross-platforms

c. Concern for native feature access

The proposed project requires access to some device features like camera, offline access, Geolocation, file upload, swipe navigation, and microphone. Best to assume the access of native feature will definitely workable for both platforms.

Best used: Native / Cross-platforms

Conclusion

Based on the outcome from 3 major concerns reviewed, the cross-platforms mobile development approach is the only approach that met 3 requirements. Thus, the proposed project will choose to be developed under cross-platforms frameworks rather than native nor mobile web framework.

2.5 Review on Best Hybrid Mobile Application Development Frameworks

a. Ionic

Ionic built on top of Angular JS and Apache Cordova.

Advantages	Disadvantages
App development using HTML, CSS, and JS.	Learning curve for Angular JS is stiff, Ionic is not a better option for developers not acquainted with Angular JS.
The use of powerful Angular JS framework that is embedded in by default.	Use WebView, so performance can be slow compared to native applications.
Good and versatile availability of plugins.	Not suitable to create graphic games and high-end apps.

Table 2-5-1 Advantages and Disadvantages of Ionic

b. PhoneGap

PhoneGap is an open source platform and is often known as Apache Cordova.

Advantages	Disadvantages
Reduced learning curve as it utilizes JavaScript, HTML5, and CSS3 languages.	Uses WebView which results in performance challenges.
Follows plug-in architecture increases the chances of extending native device APIS in modular ways.	Still does not support many of native APIs.
It is free and open source.	Lack of standard UI library.

It is easier to seek assistance from libraries for further enhancing the functionality and flexibility of App.	Does not have cache/memory management.

Table 2-5-2 Advantages and Disadvantages of PhoneGap

c. Xamarin

Xamarin is a mono framework used for cross-platform app development, build applications using C# running on .NET Common Language Infrastructure (CLI).

Advantages	Disadvantages
Xamarin re-uses business logic codes and data to develop effective offline applications.	Good for .NET developer. But it is still necessary to have native application development knowledge.
Wrapping native libraries with .NET layer allow customizations and third party libraries.	The issue of unreasonable app size to be downloaded from app stores due to framework libraries.
Improved overall performance of the app compares to other hybrid frameworks.	Only achieve 75% of the code is shareable, the developer will occasionally need to write platform specific code.

Table 2-5-0-3 Advantages and Disadvantages of Xamarin

d. Titanium

Titanium is a widely used cross-platform mobile application development framework by Appcelerator.

Advantages	Disadvantages

Comprising HTML5, CSS5, JQuery, JavaScript and Ajax is easier for mobile developers to learn.	Reported that increase application complexity will lead to some weird glitches in app. E.g. App crash randomly.
Easy for quick prototyping and get instant user feedback.	SDK related issues leading to loss of time and efforts to build better deliveries.
200,000+ developers and 35,000+ apps developed, Titanium is a large community.	Introduction of Titanium Studio has increased the learning curve.

Table 2-5-0-4 Advantages and Disadvantages of Titanium

e. React Native

React Native were introduced by Facebook and presents a framework for building cross-platform mobile applications with React technology.

Advantages	Disadvantages
Offers highly responsive UI due to asynchronous JavaScript interactions (JSX) with the native environment.	Requires extensive knowledge of React and may need Objective-C or Java knowledge.
Can use native libraries and write Objective-C, Swift, or Java if needed to further optimize performance.	Longevity of the React Native project is a concern. Hence, not a good choice for long run application especially enterprise app.
Hot reloading allows changes to be reloaded to app instantly without waste time recompiling.	

Table 2-5-0-5 Advantages and Disadvantages of React Native

2.5.1 Comparison between five cross-platform development framework

	Ionic	PhoneGap	Xamarin	Titanium	React Native
Development Language	Angular JS, HTML5, CSS3	JavaScript, HTML5, CSS3	C# .NET	JQuery, JavaScript , HTML5, CSS3	React JS, JavaScript
Type	Interpreted on top of natively compiled.	Interpreted on top of natively compiled.	Natively Compiled	Interpreted on top of natively compiled.	Natively Compiled
Common uses	Enterprise Mobile, Consumer mobile	Enterprise Mobile, Consumer mobile	Enterprise Mobile, Consumer mobile	Enterprise Mobile, Consumer mobile	Consumer mobile
Showcase	ChefSteps, Pacifica, Sworkit, etc.	Logitech Squeezebox Controller, etc.	Slack, Pinterest, eBay Classifieds, etc.	eBay, PayPal, Pwc, Cisco, iTranslate, etc.	Facebook iOS, Instagram, Airbnb, Baidu, etc.
Selection criteria					
Code reuse across platforms	>98%	>95%	75%-96%	60%-90%	80%-85%
Development speed	Fast (Overall)	Slow (Debug)	Slow (Compile, test)	Fast (Prototyping)	Fast (Overall)
Learning curve	Angular JS	Easy	C#	Easy	React JS
Access to native APIs	Easy plugin	Plugins	Direct access	Direct access	Also allow native code
Performance	Slow but improving	Slow	Comparable with native	Not responsive, laggy	Just like Native

Native UI look feel	Poor	Poor	Limited UI customizatio n	Mostly	Yes & customizabl e

Table 2-5-6 Comparison between 3 mobile application development frameworks

Conclusion

Main framework selection criteria for proposed project is:

a. Fast development speed
b. High user experience and graphical performance
c. Easy native API access

Based on the comparison chart and the 3 main selection criteria for the proposed project, the preferable framework would be React Native. On the other hand, Xamarin is a good alternative.

2.6 Review on Backend Technologies

2.6.1 Use of Restful API for application web service

The advantages of REST are: (BBVAOPEN4U, 2016)

a. **Separation between the client and the server**
   Allows the different components of the developments to be managed independently.
b. **Transparency, reliability, and scalability**
   Allow easy migrate to other servers or make all kinds of changes in the database. The separation makes it easier to have the front and the back on different servers, and this makes the apps more flexible to work with.
c. **Always independent from the type of platform or languages**
   REST API always adapts to the type of syntax or platforms being used, which gives considerable freedom when changing or testing new environments within the development.

2.6.2 Use of NoSQL database – MongoDB over SQL database – MySQL

MongoDB is an open-source NoSQL database that stores data in JSON-like structure.

Comparison between SQL and NoSQL (Microsoft Azure, 2016)

	SQL	NoSQL
Model	Relational (Table)	Non-relational (JSON doc)
Data	Fix properties, normalized model to join tables.	Flexible properties, semi-structured, or nested data.
Schema	Strict schema	Dynamic schemas
Transactions	Supports ACID	Support ACID for varies solution.

Consistency & Availability	Prioritized consistency over availability & performance.	Consistency, availability, and performance can be traded to meet app needs.
Performance	Entity info spread across many tables, require joins for update or query.	Entity info normally in a single record, update or query in one operation.
Scale	Scale vertically with more server resources.	Scale horizontally with data partitioned to span servers.

Table 2-6-1 Comparison between SQL and NoSQL

Feature comparison between MySQL and MongoDB: (Mongo, 2017)

	MySQL	MongoDB
Rich Data for Business	No	Allow
Dynamic Database Schema	No	Allow
Different Data Types	Allow	Allow
Data locality with UUIDs	No	Allow
Field updates	Allow	Allow
Quick Programming	No	Allow
Complex transactions	Allow	No
Database Auditing	Support	Support
Auto-sharding	Support	Support
Better fit for:	Applications that require complex, multi-row transactions.	Applications that engage with users. serving up content.

Table 2-6-2 Feature comparison between MySQL and MongoDB

2.6.3 Use of Place API

The Place web service is a service that provides information about point of interest, geometry locations using HTTP requests.

Comparison between four Place API (Susilo, 2014)

	Google	Foursquare	Factual	Facebook
Release	Nov 2010	Nov 2009	2008	Aug 2010
Data size (2016)	100 mils data, 200 countries	User generated data.	100 mils data, 50 countries	Factual provides data.
Format	JSON, XML	JSON, JSONP	JSON	JSON
Create place	Yes	Yes	No	No
Geocoding	Yes	No		No
Reverse Geo	Yes		Yes	
Nearby	Yes	Yes		Yes
Search	Yes	Yes		Yes
Autocomplete	Yes	Yes	No	Yes
Rate Limits per day	Free <100k	Free <120k	Free	N/A

Table 2-6-3 Comparison between 4 Place API

Solely based on the comparison table above, the proposed project will integrate with Google Places API to retrieve global places information. The integration of Google Places API for proposed project include Google place search API into search module and integration of place autocomplete API for place suggestion. (Google Inc., 2017)

2.6.4 Use of Mapping tool

Following is a capability comparison between top four mapping tools and maps APIs providers.

Comparison between Maps APIs (Wagner, 2015)

	Google Maps	**Ms. Bing Maps**	**Apple Maps**	**Mapbox**
API	Web, SDKs	Web	Web, iOS SDK	Web, SDKs
React Native compatible	Yes. Support by Airbnb.	Web	Yes. Support by Airbnb.	Yes. Experimental
Free views /day	25k	Free	Free	50k
Pricing	$0.50 /1k	$0.50 /1k	Free	$0.50 /1k
Ease of use	API picker, SDKs, libraries, etc.	AJAX interactive SDK	Code samples, SDKs	Live demos, code samples
Map styles	Custom map, Imagery, Satellite	Streetside cities, imagery, 3-D cities	Tile-based, 3-D building	Custom map, Streets, Satellite
Offline	No	No	No	Yes

Table 2-6-4 Comparison between maps APIs

Based on table above, it is possible that the proposed project integrates mapping tools and features from both Google Maps and Apple Maps according to the device nature. E.g. Google Maps for Android device and Apple Maps for iOS device.

2.7 Review on System Development Methodologies

The following are descriptions, strengths and weakness of three methodologies: Waterfall, Spiral, and Agile. (CMS, 2008)

2.7.1 Waterfall methodology

Waterfall methodology strictly follows sequential projects development processes. Waterfall is best used for simple, fixed requirement projects. (Smartsheet, 2016)

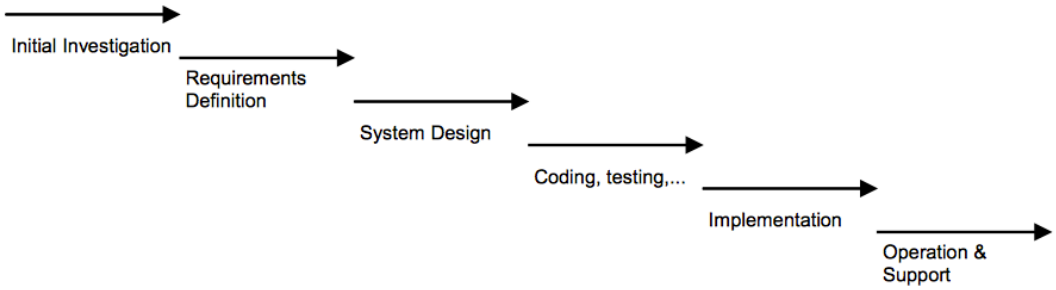


Figure 2-7-1 Waterfall model

Advantages	Disadvantages
Easy to use and manage • Each phase has specific deliverables.	Changes cannot be easily accommodated • Phases that completed, can't go back.
Discipline is enforced • Focus on requirements and design first.	Software isn't delivered until late • Coding begins at 4-5 phase.
Requires a well-documented approach • Better understanding of code logic/test.	Challenge to gather accurate requirements • Most user requirement is uncertain initially.

Table 2-7-1 Review of Waterfall model

2.7.2 Iterative framework - Spiral methodology

Iterative development divides system development of an application into small chunks. Each chunk is designed, developed and tested in repeated cycles. (CMS, 2008)

Spiral model is a mixture of iterative and sequential development model. Spiral methodology is best used to evaluate cost and risk for long-term project due to expectations and external changes. (CMS, 2008)

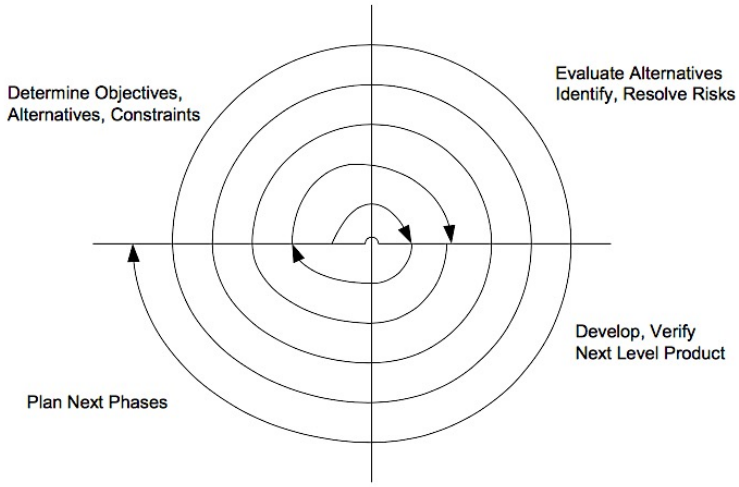


Figure 2-7-2 Spiral model

Advantages	Disadvantages
Changing requirements can be accommodated.	Management is more complex.
Development divided into sub part and high priority part can be developed earlier.	Many sub-stages involved therefore requires excessive documentation.
User can see system early.	Timeline of project may not be known early.

Table 2-7-2 Review of Spiral model

2.7.3 Agile framework – Scrum methodology with Kanban

Agile software development does not focus on in-depth planning at the early phase and open to constant feedback and changing requirements over time. Backlog contains workloads that are prioritized according to business or customer value. (Smartsheet, 2016)

Scrum is one of the most used methodologies under Agile. It follows consistent set of roles, responsibilities, and meetings for all phases. Each Sprint usually takes one or two weeks allows delivering of the system on a regular basis. (Smartsheet, 2016)

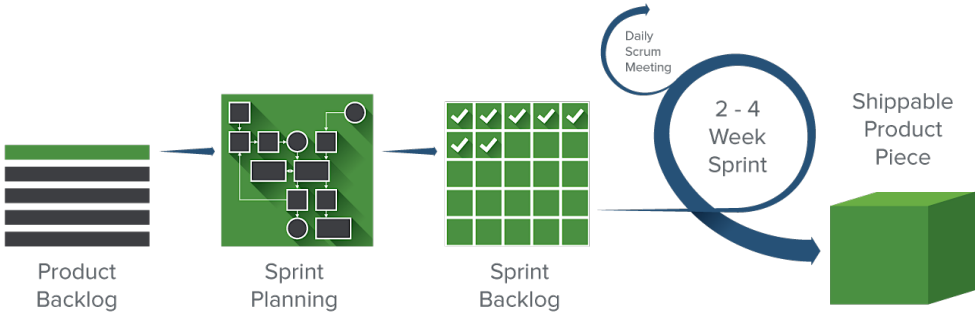


Figure 2-7-3 Scrum model

Advantages	Disadvantages
More transparency and project visibility.	Risk of scope creep, accommodate request.
Increased team accountability.	Team requires experience & commitment.
Easy to accommodate changes.	Poorly defined tasks can lead to inaccuracies.
Increased cost savings.	Wrong Scrum master can ruin the project.

Table 2-7-3 Review of Scrum model

Kanban shows continuous development progress of current system/Sprint. Kanban helps visualize the to dos, limit number of task in progress, manage and enhance the workflow, make plans clear, and encourage work improvement. Visual nature of Kanban can be easily applied with other process models. (Smartsheet, 2016)

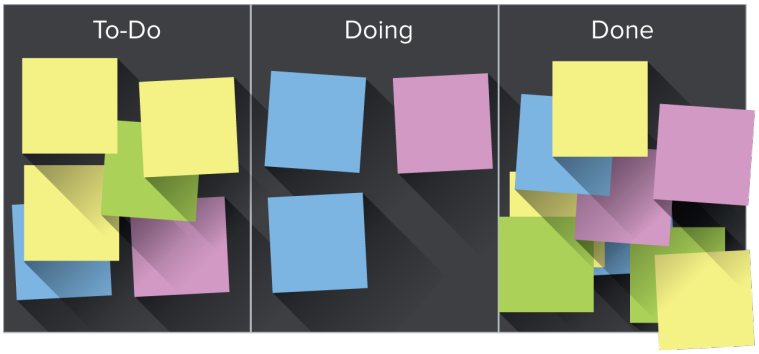


Figure 2-7-4 Kanban model

Advantages	Disadvantages
Reduce risk of scope creep • Priority of Sprint items is reevaluated as new info comes in.	Overcomplicate Kanban board • Apply too much "new tricks" buries the important info.
Visualize status of work at a glance. • Kanban board have 3 columns: to do, in progress, and done.	Outdated board can lead to issues • Working off inaccurate info is a waste of commitment.
Minimizes Sprint time • Focus on ensuring work move quickly and successfully.	

Table 2-7-4 Review of applying Kanban on top of Scrum

Conclusion

Based on the evaluation of each methodology's strength and weaknesses, the proposed project will implement the combination of Kanban model on top Scrum methodology. Whereby both methodologies are based on Agile framework, that emphasize on incremental and iterative development to reduce the risk of project failure and ensure deliverables qualities.

2.8 Review on Popular Cloud Platform

Below is a table compare five cloud platforms for hosting the server-side application of the proposed project. The comparison is based on five criteria including service type, ease of use for developer, performance and value, scalability, and best use case.

	AWS	Heroku	Digital Ocean	Azure	Google
Owner	Amazon	Salesforce	DigitalOcean	Microsoft	Google
Service Type	Iaas, Paas	Paas	Iaas	Iaas, Paas	Iaas, Paas
Languages support	.NET, Ruby, Node, Go, Docker, PHP, Python	Node, Java, Ruby, PHP, Python, Go, Scala,Clojure	Go, Ruby, Python, Node, Java, PHP, Scala, Clojure, Haskell, .NET, Objective-C, Perl	.NET, PHP, Node, Java, Python	Go, PHP, Java, Python, Node, .NET, Ruby
Data Storage options	Amazon RDS, Amazon DynamoDB	Amazon RDS, MySQL, MongoDB, Neo4j	Self-configure block storage or MongoDB	SQL, MySQL, Table Storage	GQL, Google Cloud SQL
Ease of Use	Steep learning curve. Console interface available but with AWS specific lingo & concept	Simple. Can deploy from Github. Documentation is fair.	Manually set up instance and VM. Best if org has a system admin.	Admin portal available, but non-.NET app setup is not intuitive.	Fast, ready-to-go development stacks for WordPress, LAMP, MongoDB, Redis, Casandra, etc.
Performance and value	Very good performance with multiple service available if	Free tier has enough performance to run college project. Infrastructure runs on top of	$5/month can host multiple apps in 1 instance. SSD cloud runs instances and storage.	Iaas offering and price is top notch and cheap,	Fast, cheap, scalable, but Iaas and Paas services offering is basic.

	willing to pay.	AWS. 150+ add-ons available include Mongoose database in free tier.		but Paas offering is poor.	
Best Use Case	Work the best in any scenario with the present of system admin.	Simple, no configuration, smooth scaling, easy continuous deployment, and decent value.	Good for single instance apps. Especially portfolio app that only access occasionally.	Very useful for .NET or Windows needs.	Simple web app, and the need to utilize Google big data APIs.

Table 2-8-1 Cloud Platforms Comparison

Conclusion

To draw out the conclusion of the best suitable cloud platform for proposed project based on the comparison table. Heroku would be perfect for hosting the current state of the proposed project, considering the complete documentation support of Node.js, simple and direct deployment of application from Github, with zero cost (Not even payment method registration). In additional Heroku offers add-on for the integration of Mongoose MongoDB which is more convenient for developer to focus on the client-side development than worrying the complex configuration of cloud platform.

Chapter 3: System Design

3.1 System Flowchart

3.1.1 First time user System Flowchart

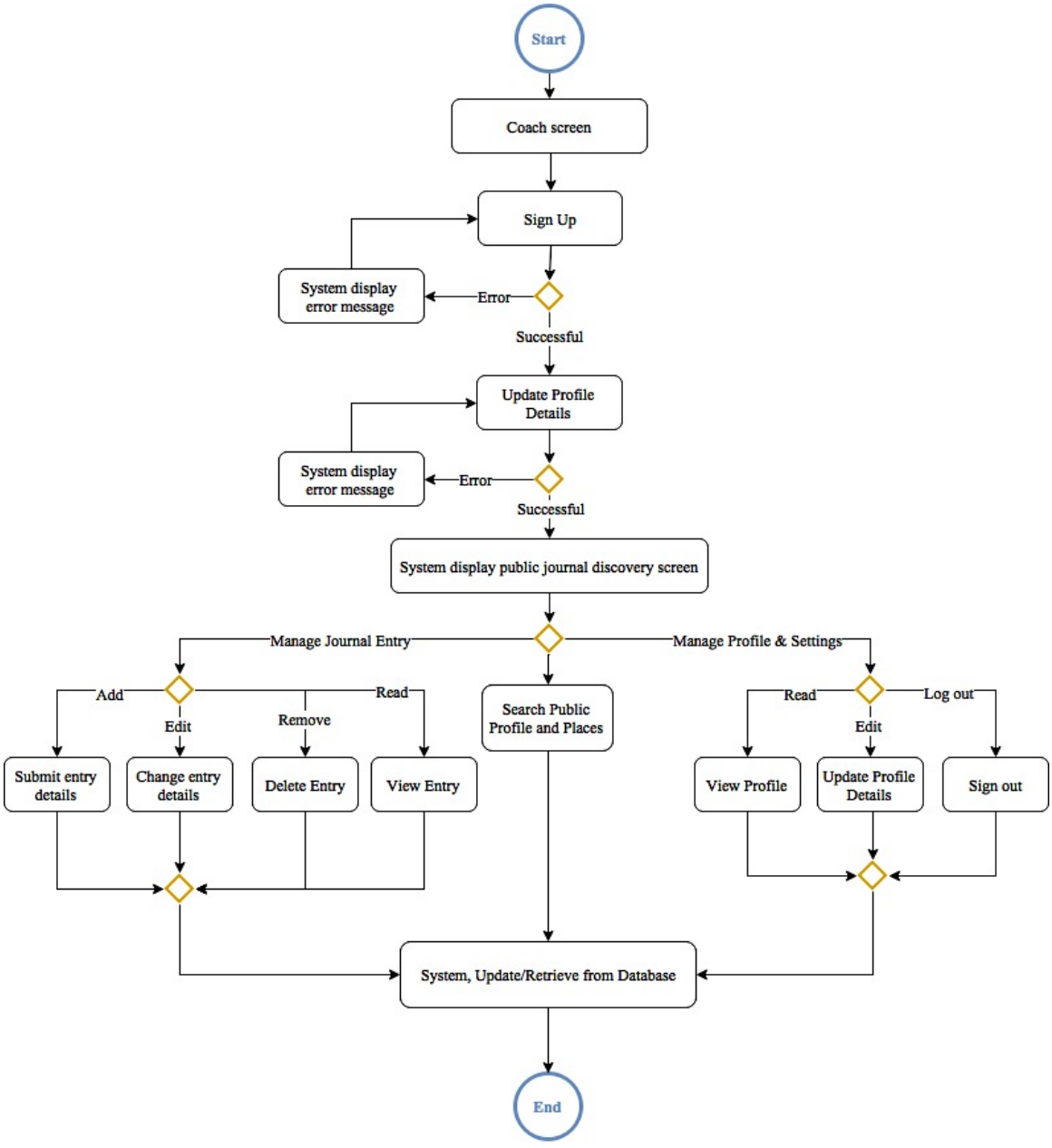


Figure 3-1-1 First time user system flowchart

3.1.2 Existing user System Flowchart

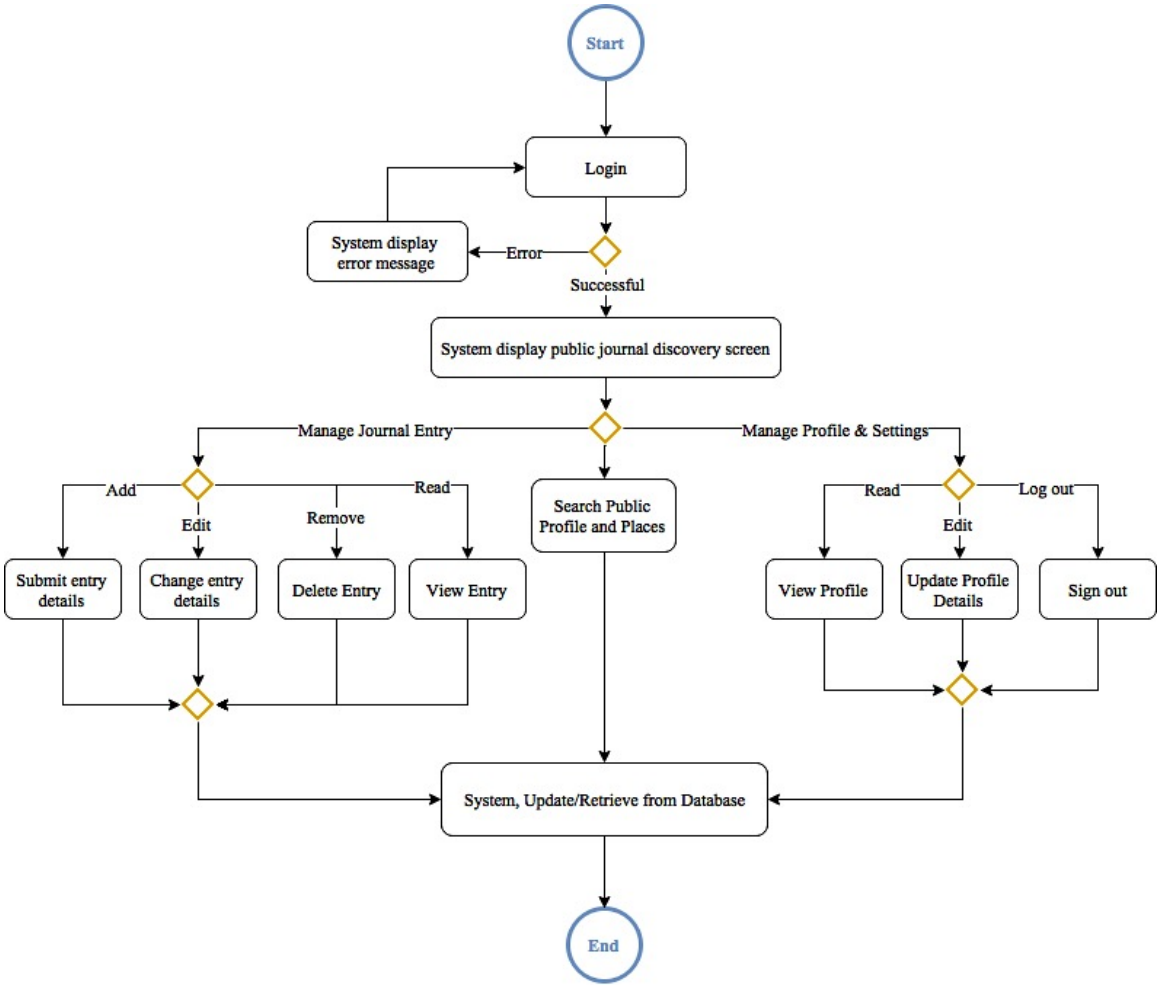


Figure 3-1-2 Existing user system flowchart

3.2 Use Case Diagram

This project uses Use Case Diagram to model user interaction with system components. This methodology helps to identify and clarify user requirements.

a. Authentication

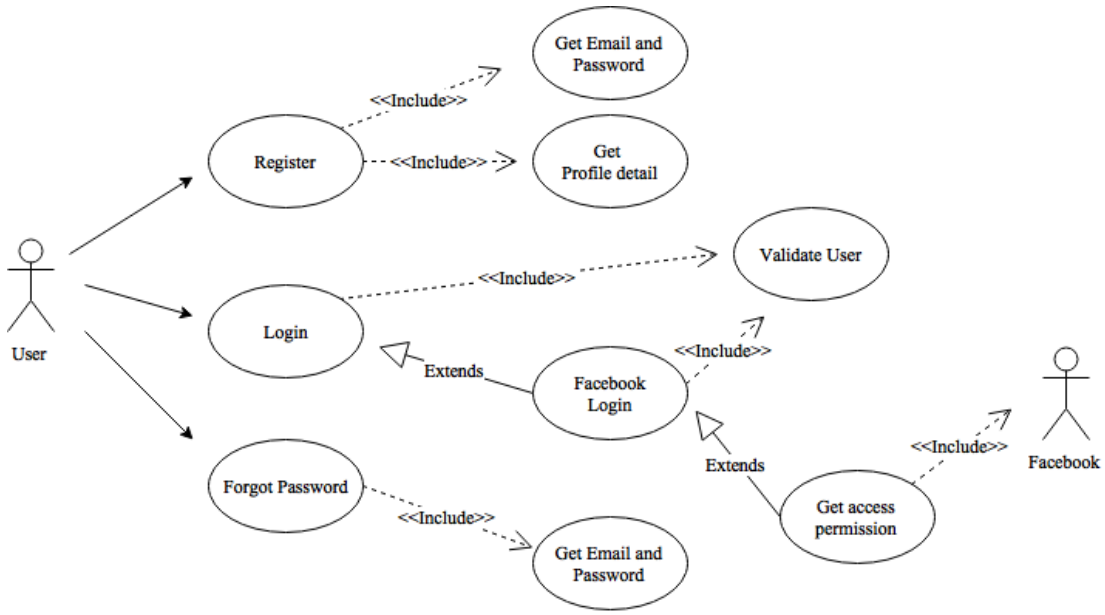


Figure 3-2-1 User Authentication Use Case

b. Home

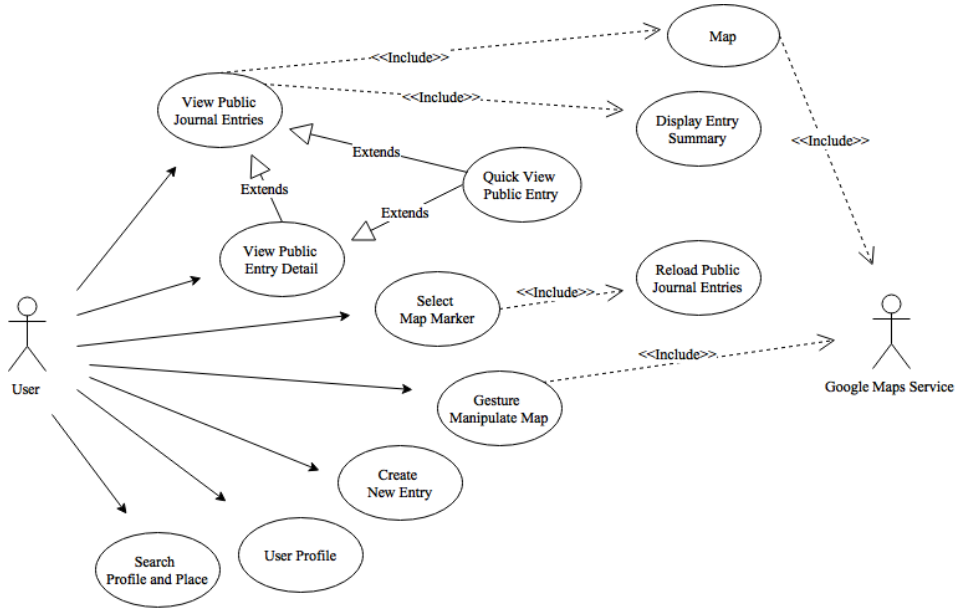


Figure 3-2-2 Home Page Use Case

c. Create New Entry

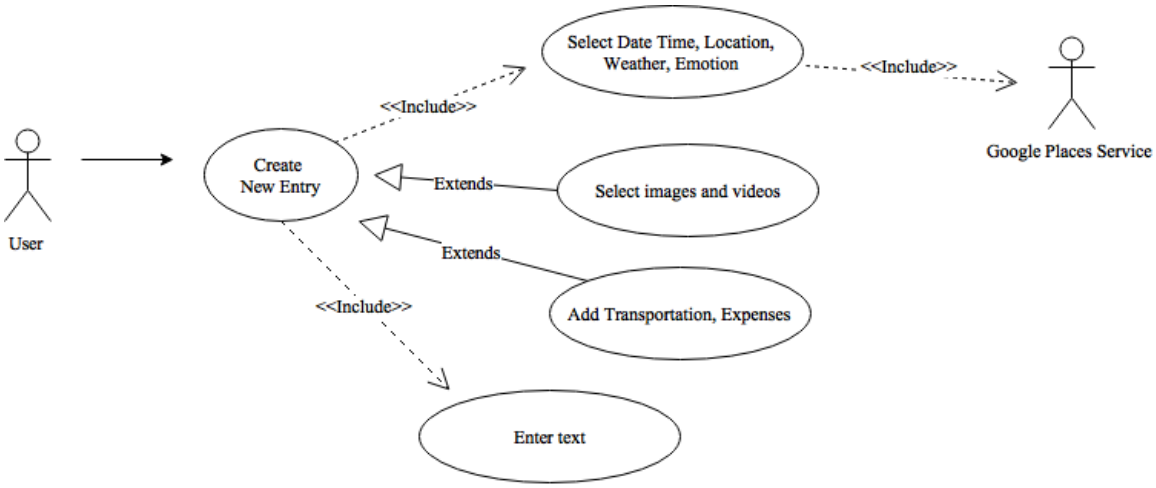


Figure 3-2-3 Create New Entry Use Case

d. User Profile

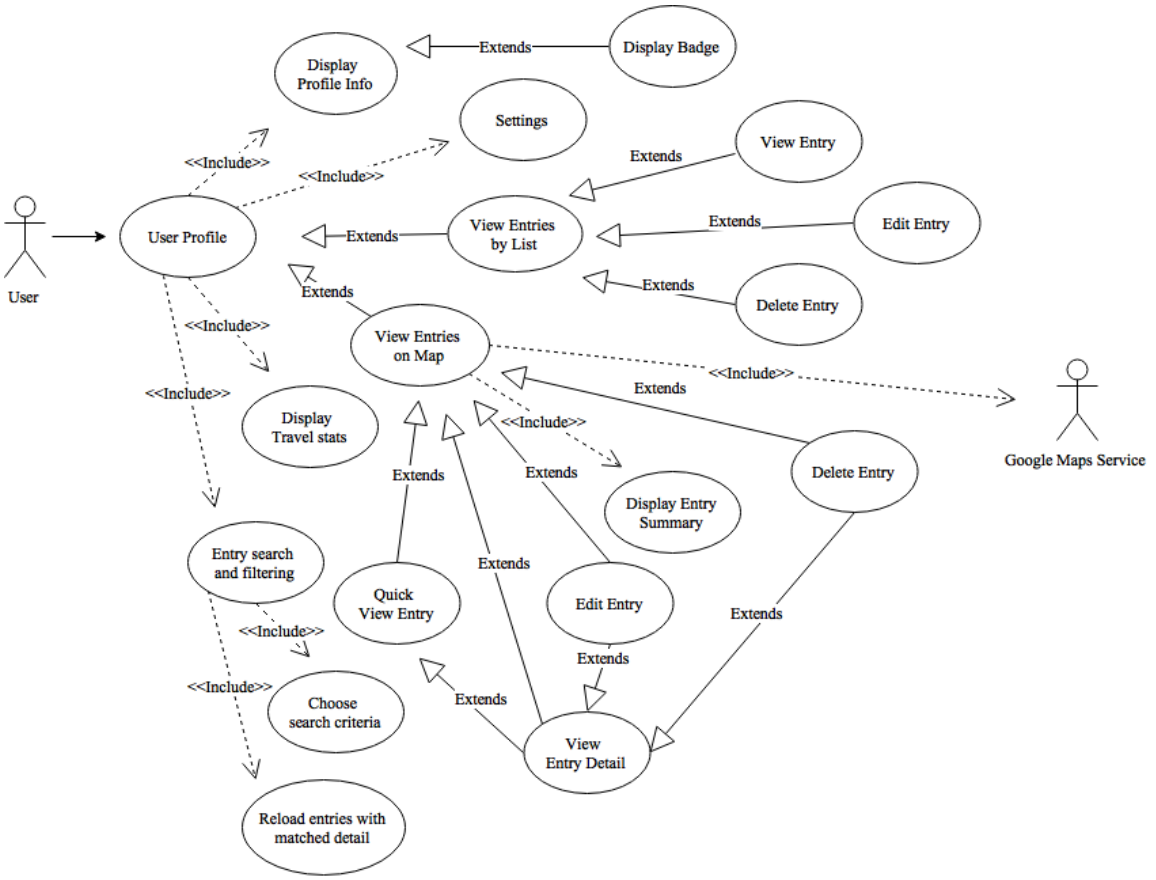


Figure 3-2-4 User Profile Use Case

e. Settings

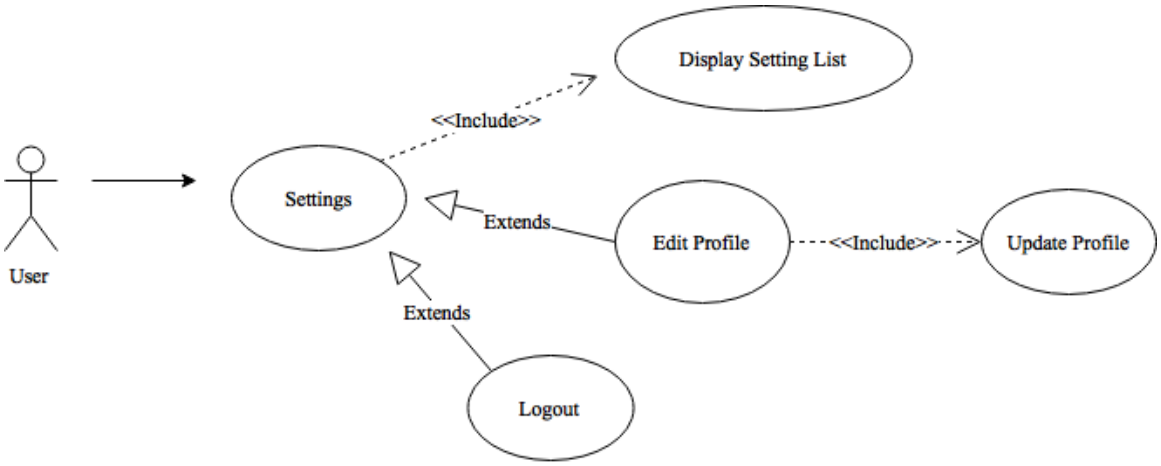


Figure 3-2-5 Settings Use Case

f. Search Profile and Place

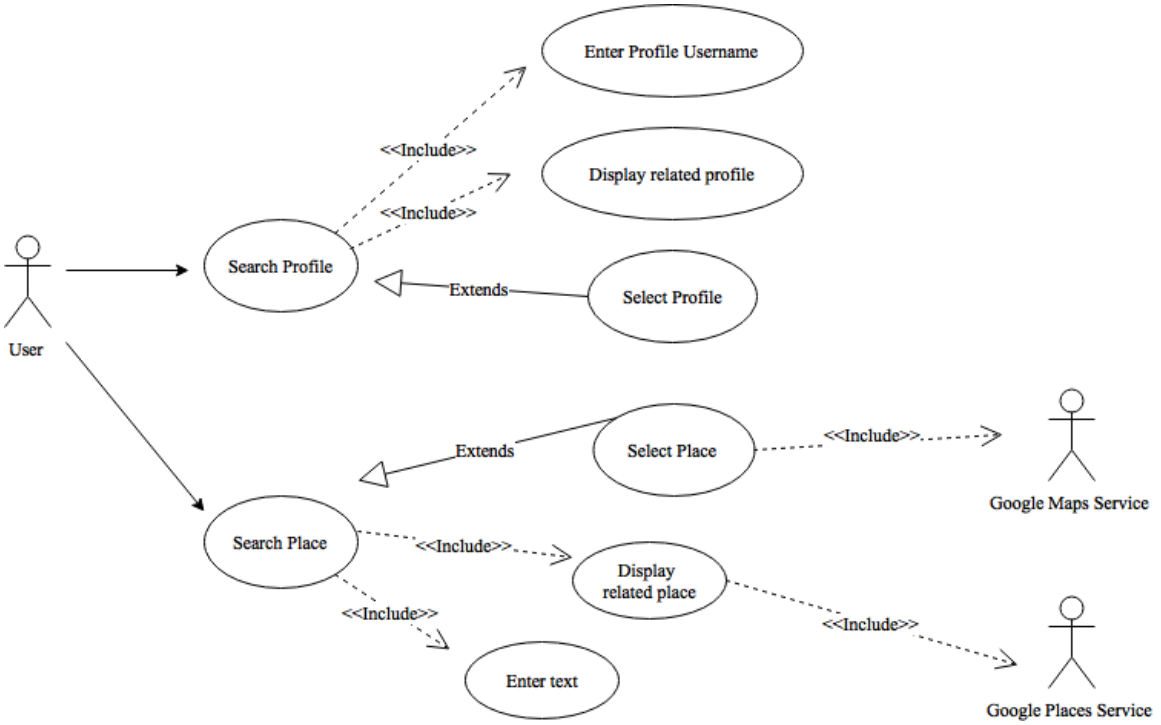


Figure 3-2-6 Search Profile and Place Use Case

3.3 Activity Diagram

This project uses Activity Diagram to model activities process flow between user-system.

a) Sign Up

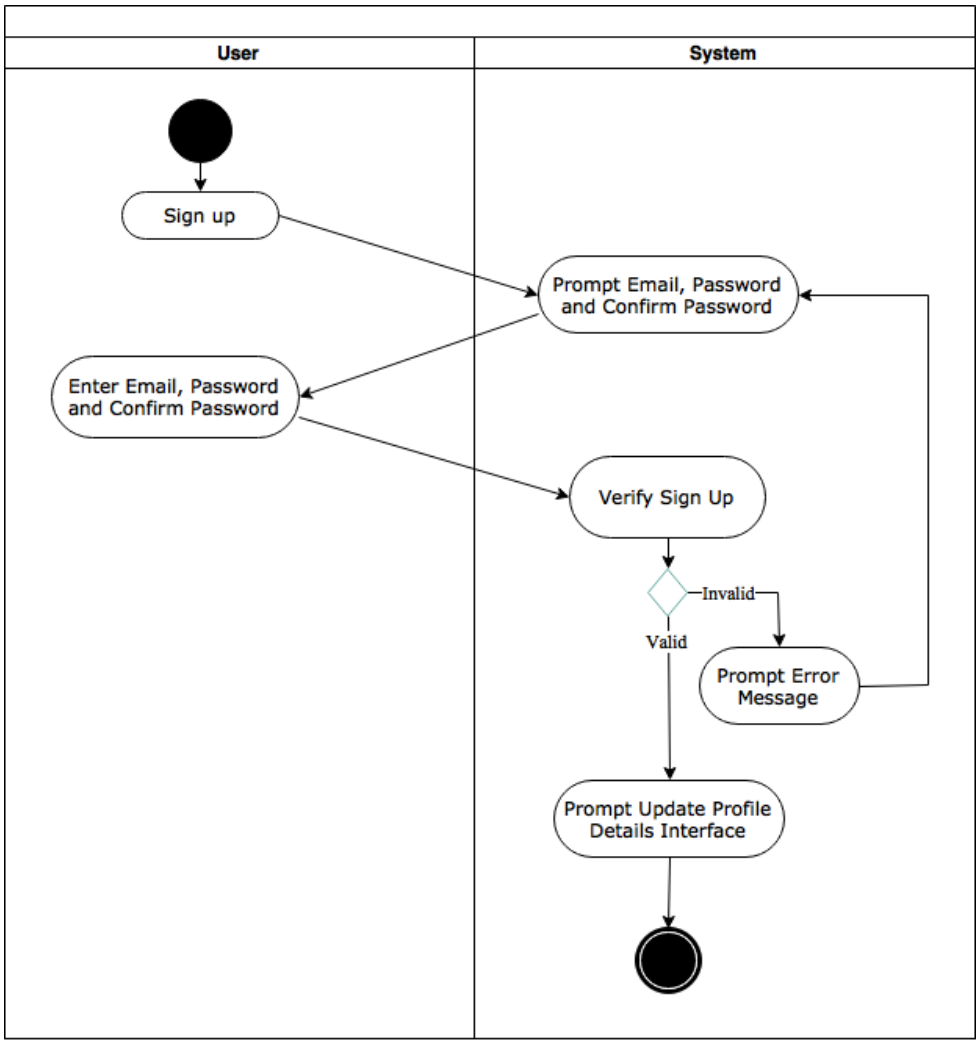


Figure 3-3-1 Sign Up activity diagram

User request to register a new profile in the system. System prompt user to enter email address, password, and confirm password again. Then user's submission of authentication credentials, system validate these credentials. If valid then new user profile is created, user will redirect to update user profile details screen. Else invalid, then system prompt user to re-enter authentication credentials.

b) Sign In

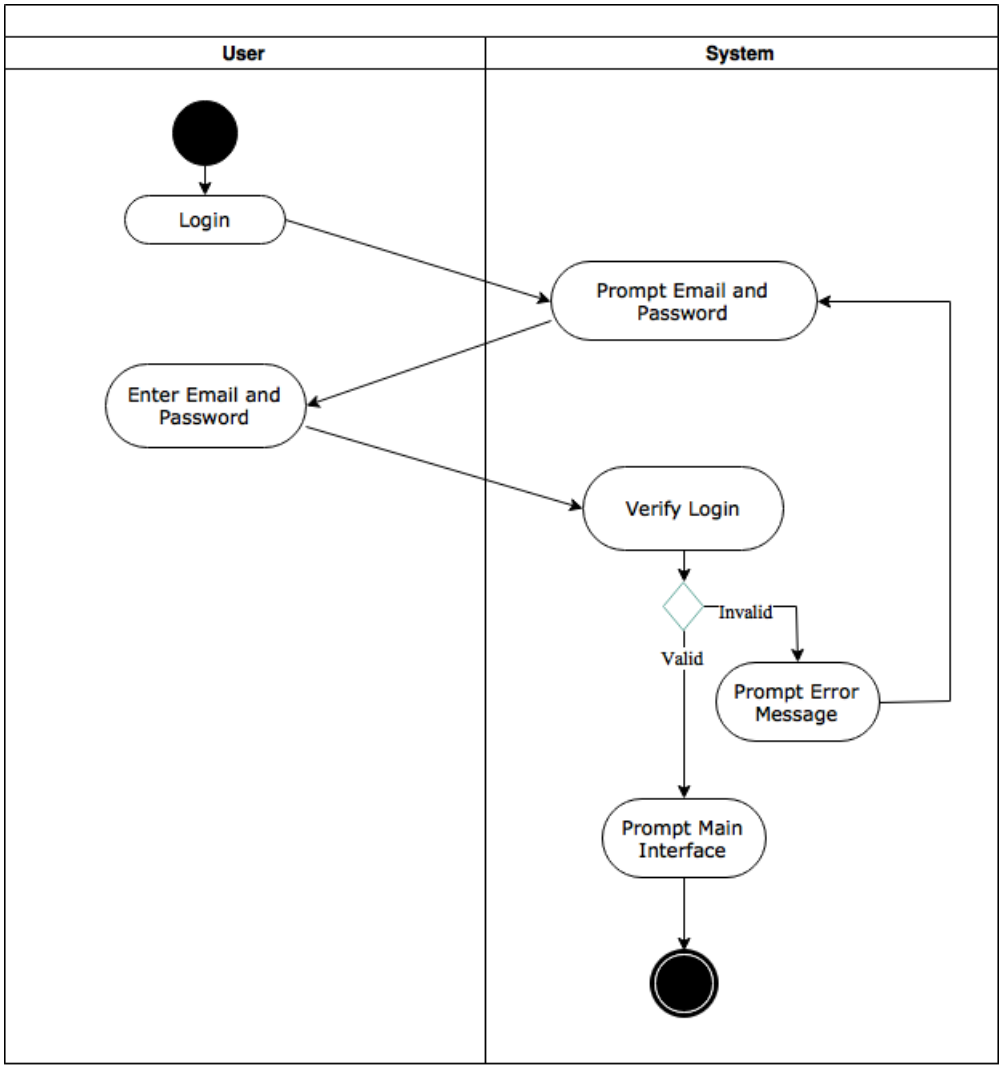


Figure 3-3-2 Sign in activity diagram

User request to log into system. System prompt user to enter required fields then validate user login credentials. If login credential is invalid then system shall prompt user to re-enter user credentials, else system redirect user to main screen.

c) First time user update Profile Details

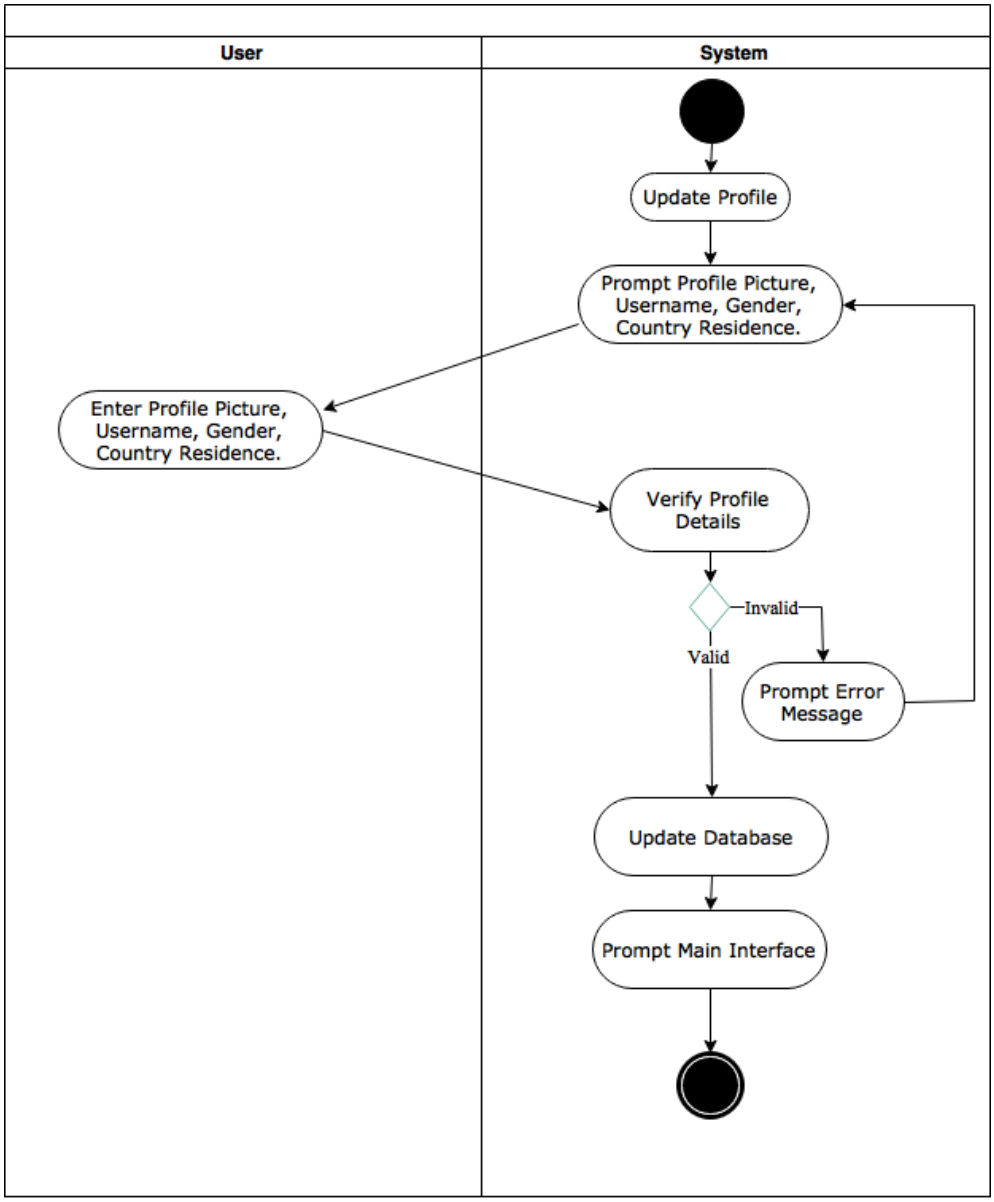


Figure 3-3-3 Update Profile Details activity diagram

System redirect user from sign up screen to update user profile screen. System prompts user to select profile picture (launch camera/ choose from library) enter username, gender, and country currently residence. Upon user submit profile details, system validate profile details. If details are valid then system update database, else system prompt user to re-enter profile details. Assume successfully update database, system redirect user to application main interface.

d) Main screen

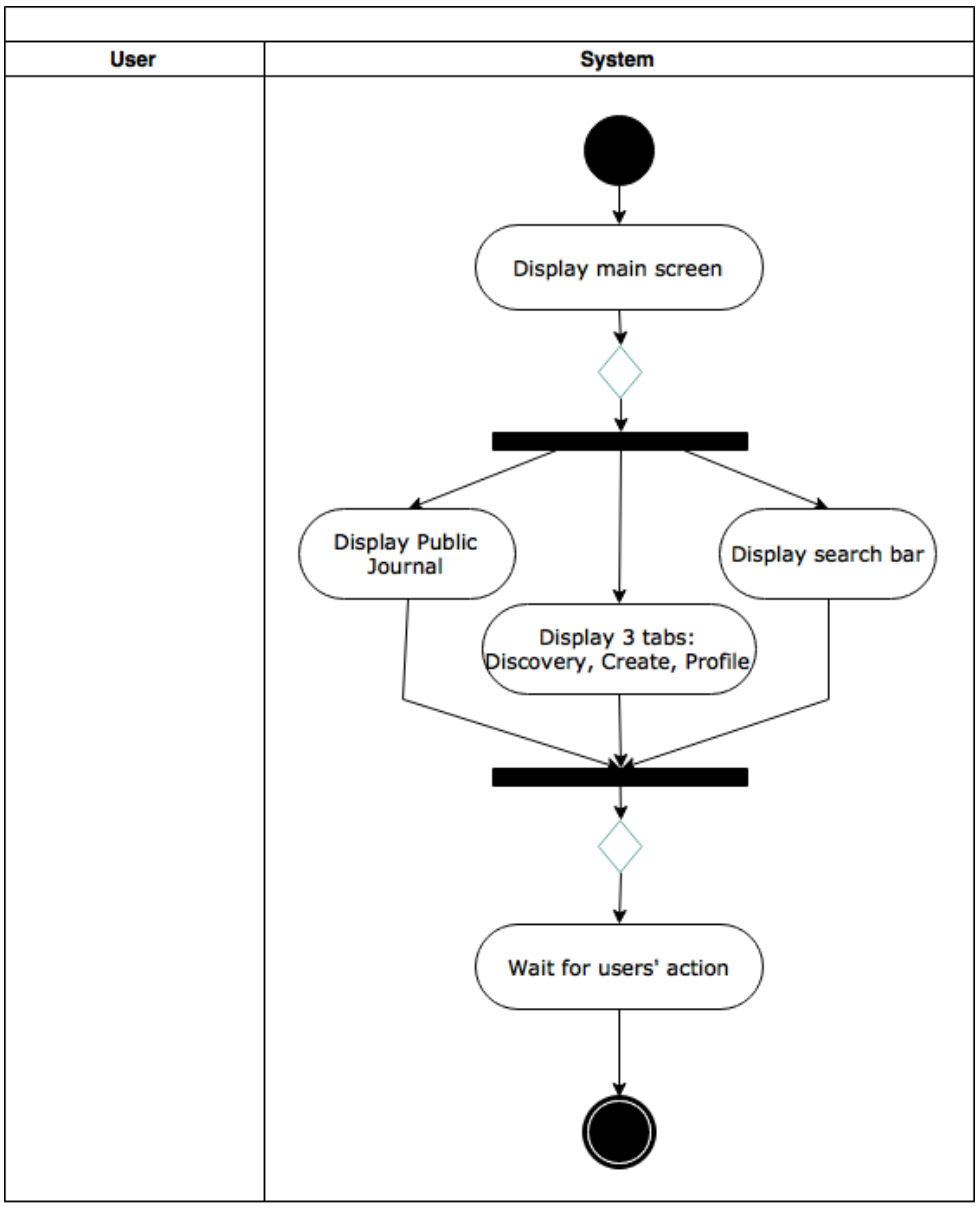


Figure 3-3-4 Main activity diagram

System redirect user to main screen after every successful login. System display public journal, main function tabs, and search bar for profile and place searching. After the completion of interface rendering, system wait for user's further command/action.

e) View Public Journal Entry

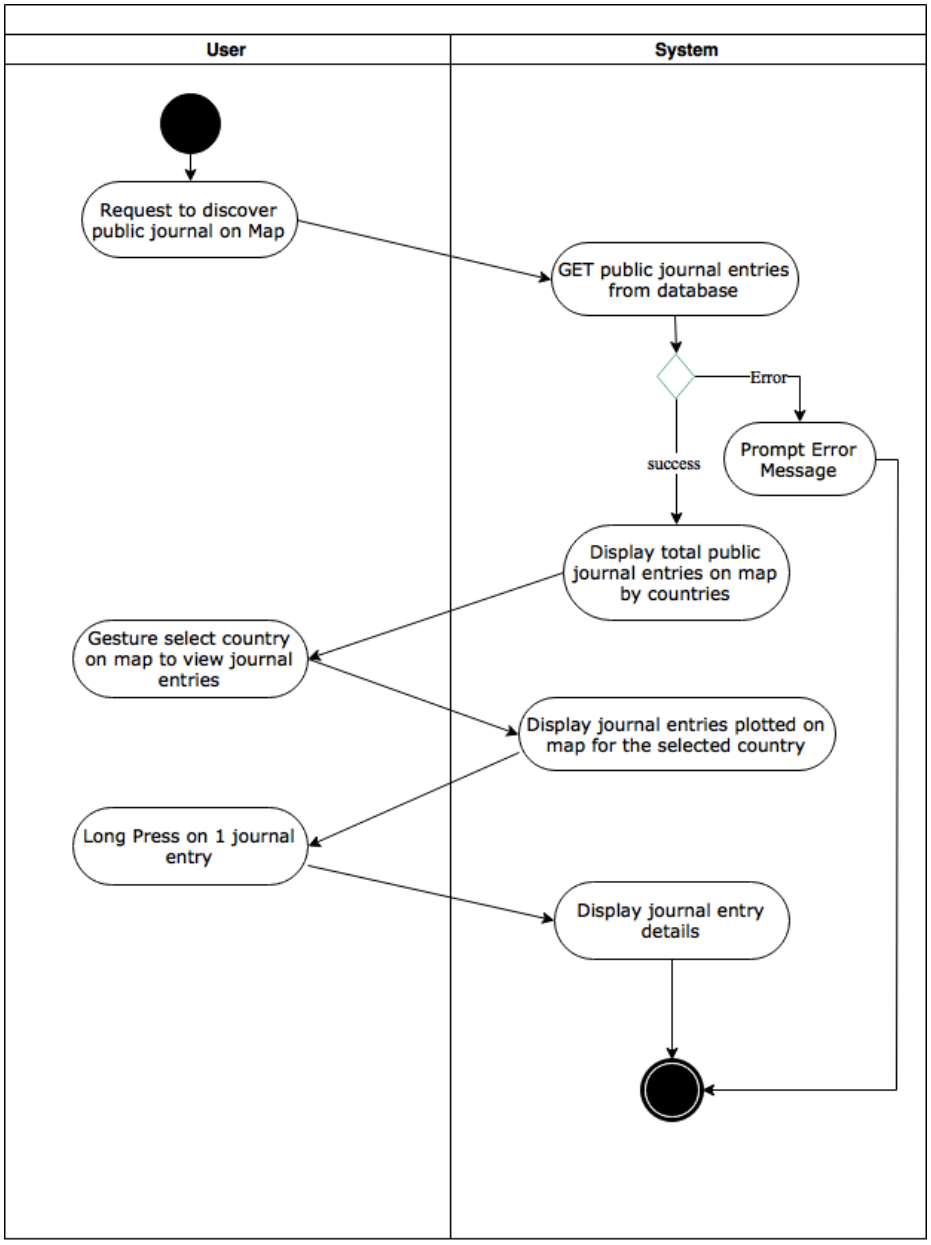


Figure 3-3-5 View Public Journal activity diagram

System perform HTTPS request and display total number of public journal entries available by country on map interface. If user select a country to view all journal entries, system perform HTTPS request then display list of journal entries available and plot number of entries available in a particular location. User can perform gesture on the map, this includes pinching, tap on map and tap on markers on map.

f) View Journal Entry

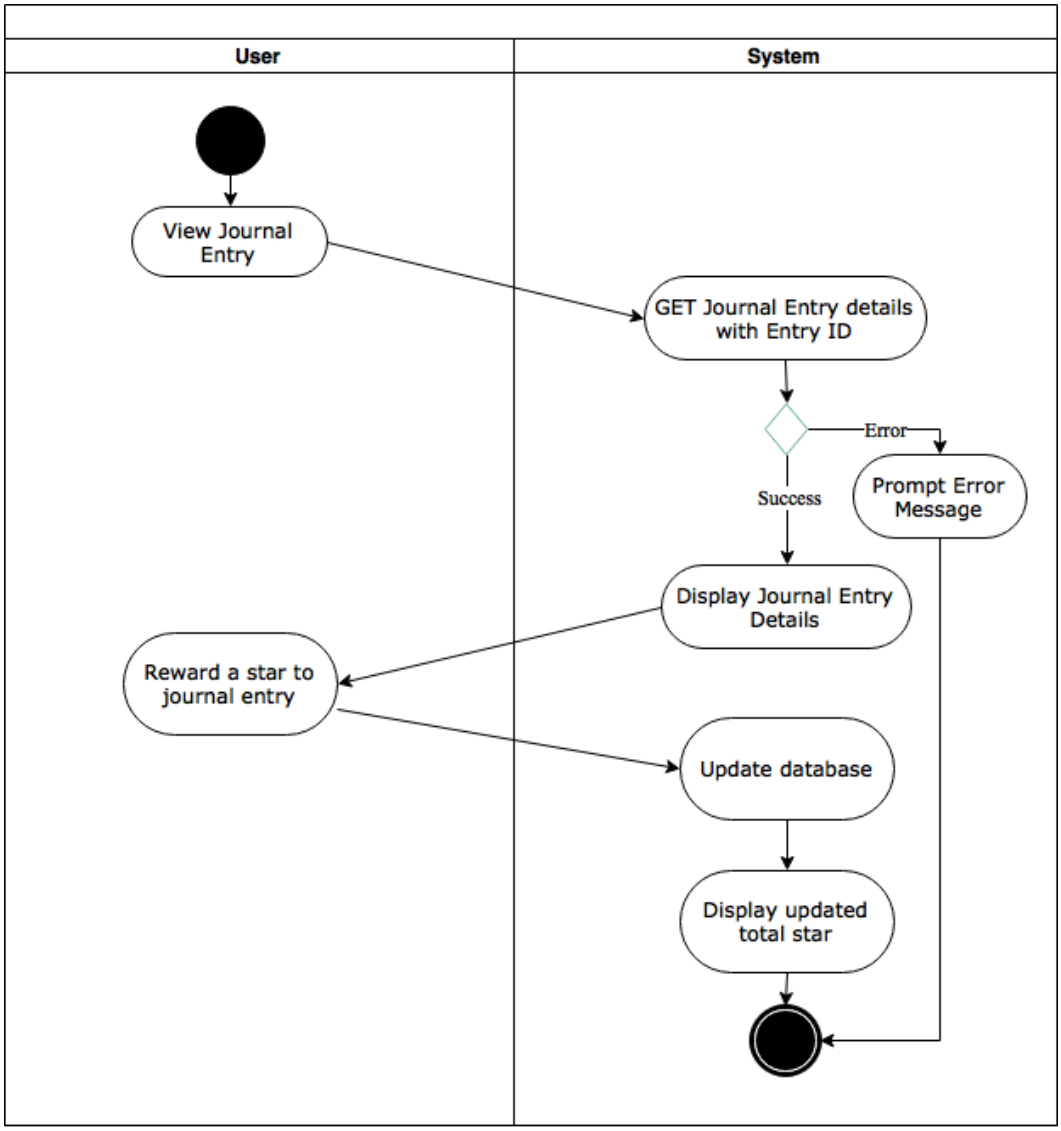


Figure 3-3-6 View journal entry activity diagram

User request to view journal entry details. System perform HTTPS request to get journal entry details using Entry ID. If success, system display journal entry details on entry details interface. If user can reward a star to the entry as a gesture of appreciation. If star reward is made, system update database and display updated total number of stars if the transaction is successful.

g) Create Journal Entry

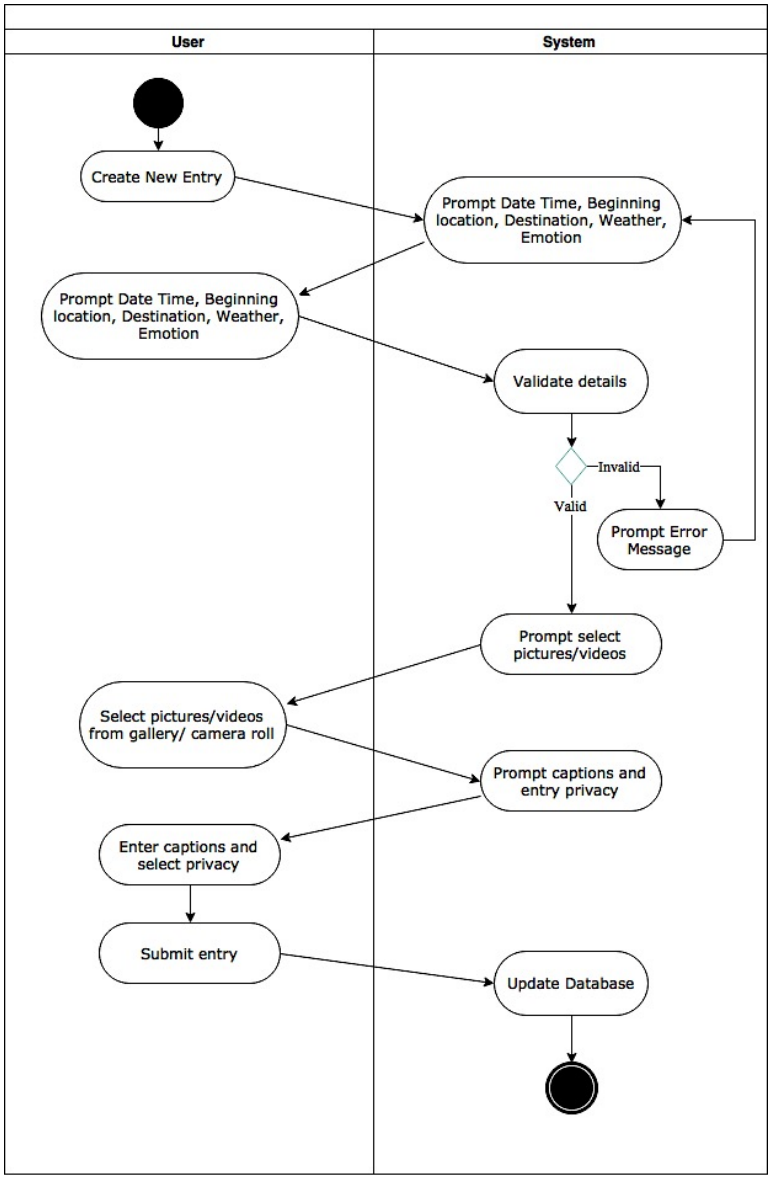


Figure 3-3-7 Create Journal Entry activity diagram

User request to create new journal entry. System display create new entry interface, then prompt user to enter date time (system auto select current date time), Beginning & Destination location, Weather (system auto request API to get weather & temperature based on location), emotion, photos/videos, captions, and select privacy for entry (private/public). After user submit new entry details, system update database and return to previous screen.

h) Edit Journal Entry

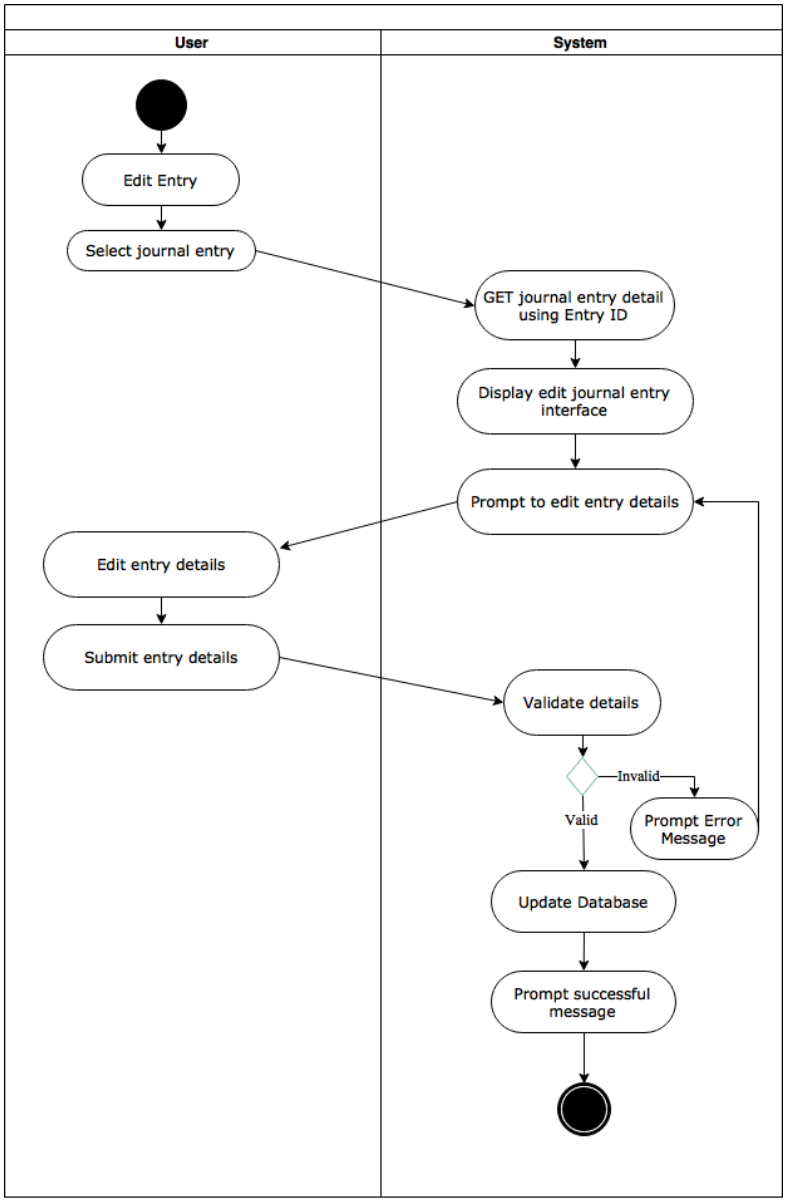


Figure 3-3-8 Edit journal entry activity diagram

User request to edit entry by select the entry to edit. System perform HTTPS request to retrieve entry details using entry ID. Upon successful retrieve, system display edit journal interface and prompt user to edit entry details. Once user done editing entry and submit entry to update. System validate details, if valid system update database and prompt successful message if successful transaction.

i) Remove Journal Entry

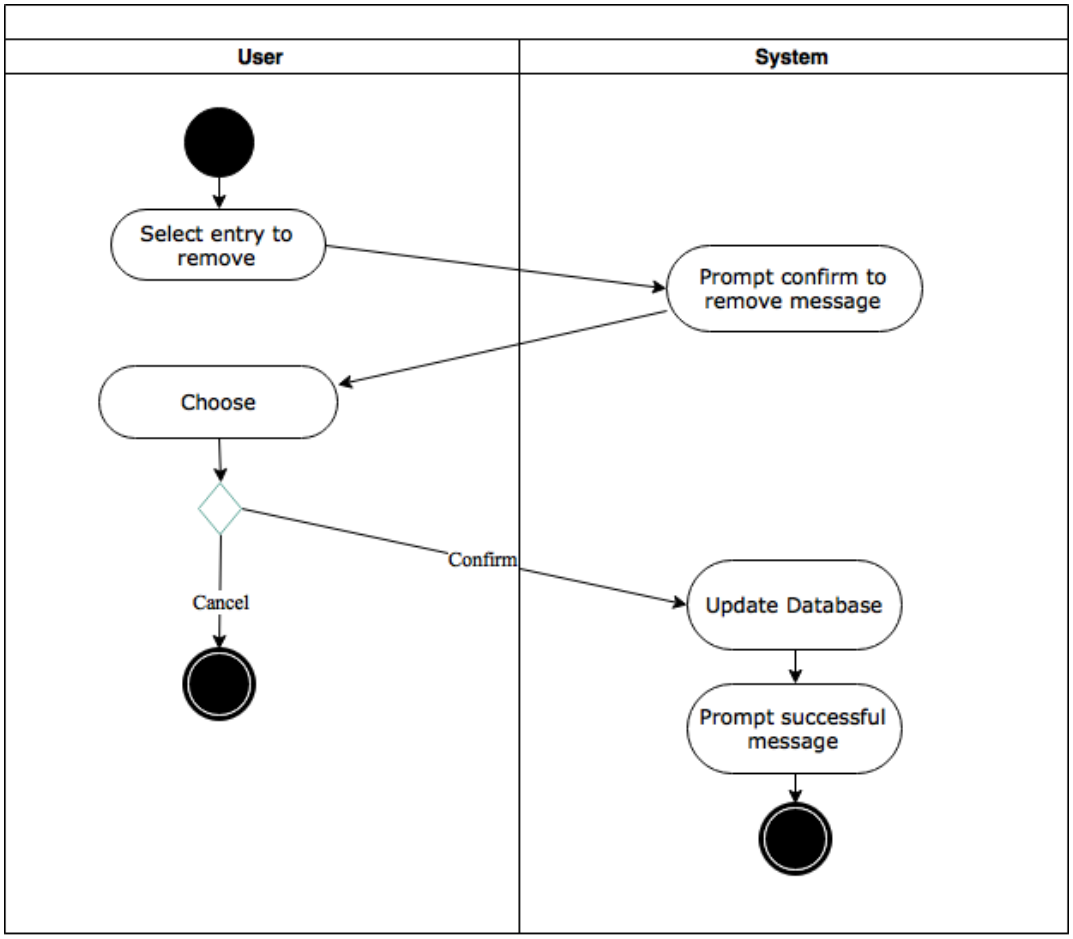


Figure 3-3-9 Remove Journal Entry activity diagram

User request to remove a journal entry by selecting an entry. System prompt to get user confirmation to remove the entry from database. If user select "Confirm" , system update database and prompt successful message for successful transaction else operation is canceled.

j) View Profile

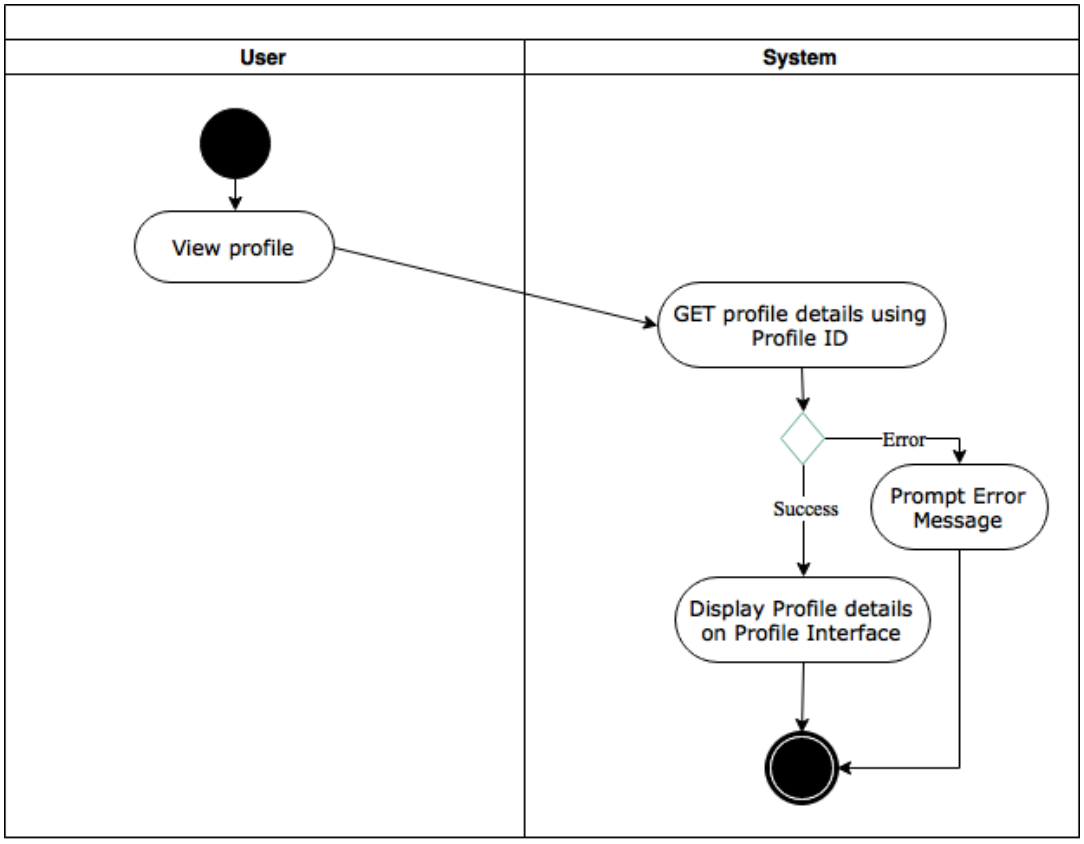


Figure 3-3-10 View profile activity diagram

User request to view profile details. System perform HTTPS request to retrieve profile using profile ID. If successfully retrieve the correct profile, system display profile details on profile interface. Else system prompt error message.

k) Edit Profile

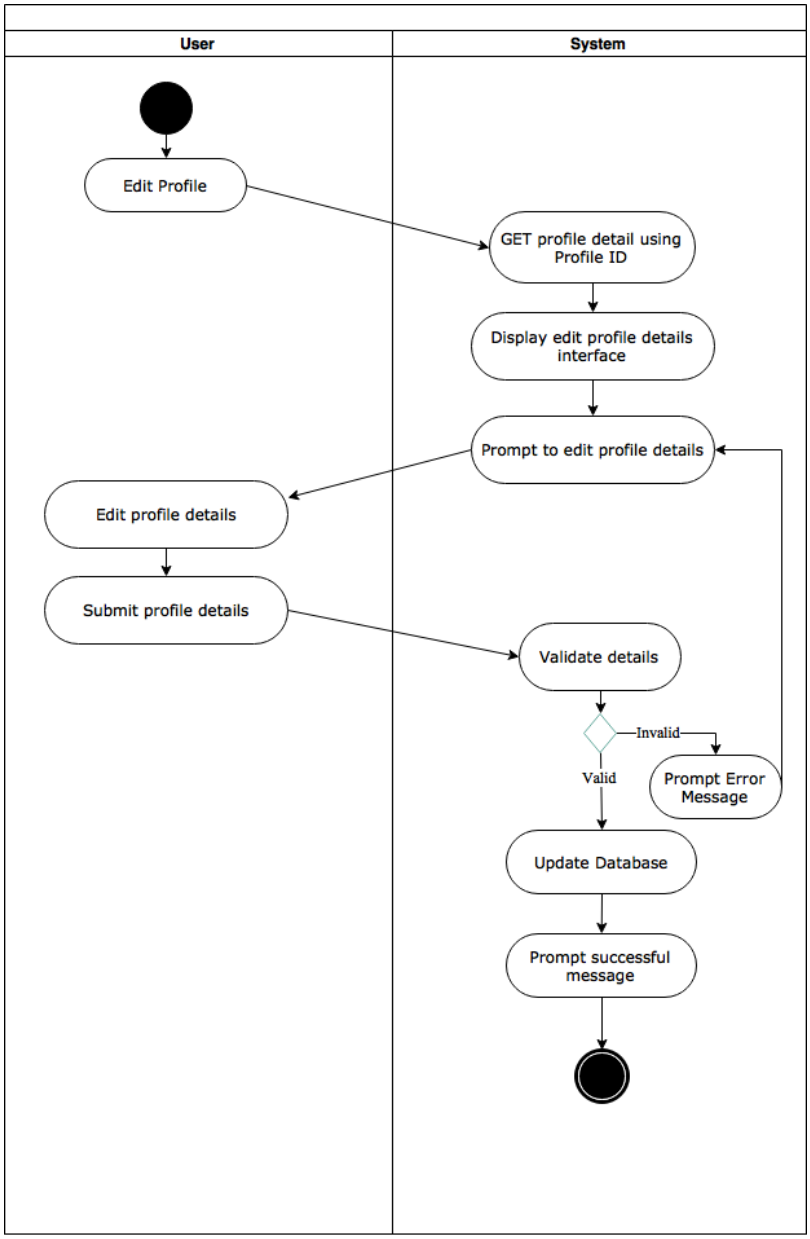


Figure 3-3-11 Edit Profile activity diagram

User request to edit profile details. System perform HTTPS request to retrieve editable profile details then display existing profile details on profile details edit interface. System prompt user to edit profile details. After user done editing profile details, user submit profile details for system to validate. If the profile details entered is valid, system update database and prompt successful message of transaction succeed.

l) Search Profile

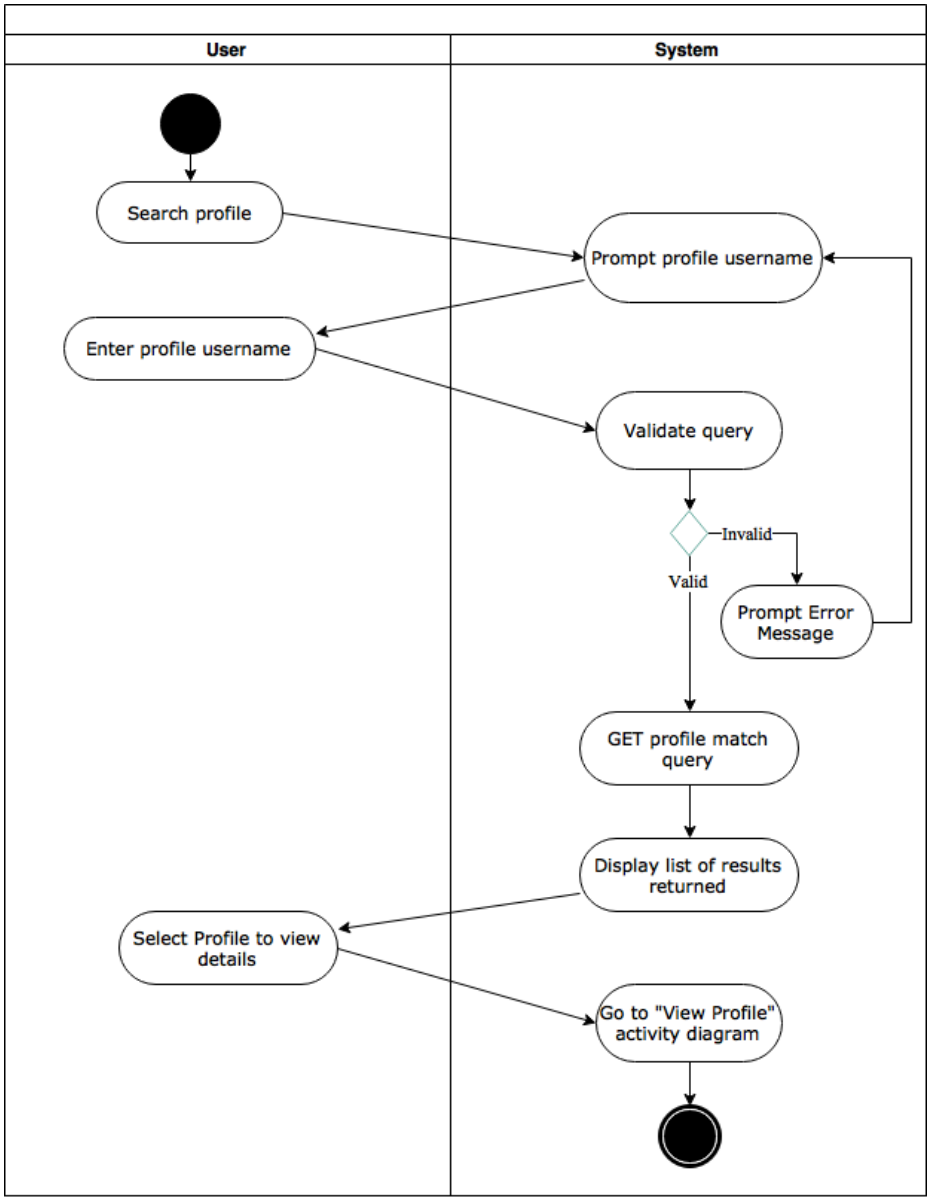


Figure 3-3-12 Search Profile activity diagram

User request to search profile. System prompt user to enter profile username. User enter profile username or keywords then submit query for system to validate query. If the query is valid, system perform HTTPS request to retrieve a list of profile that username matches query. Upon successful retrieval of profiles, system display results in a list. User select a profile to view profile details.

m) Search Place

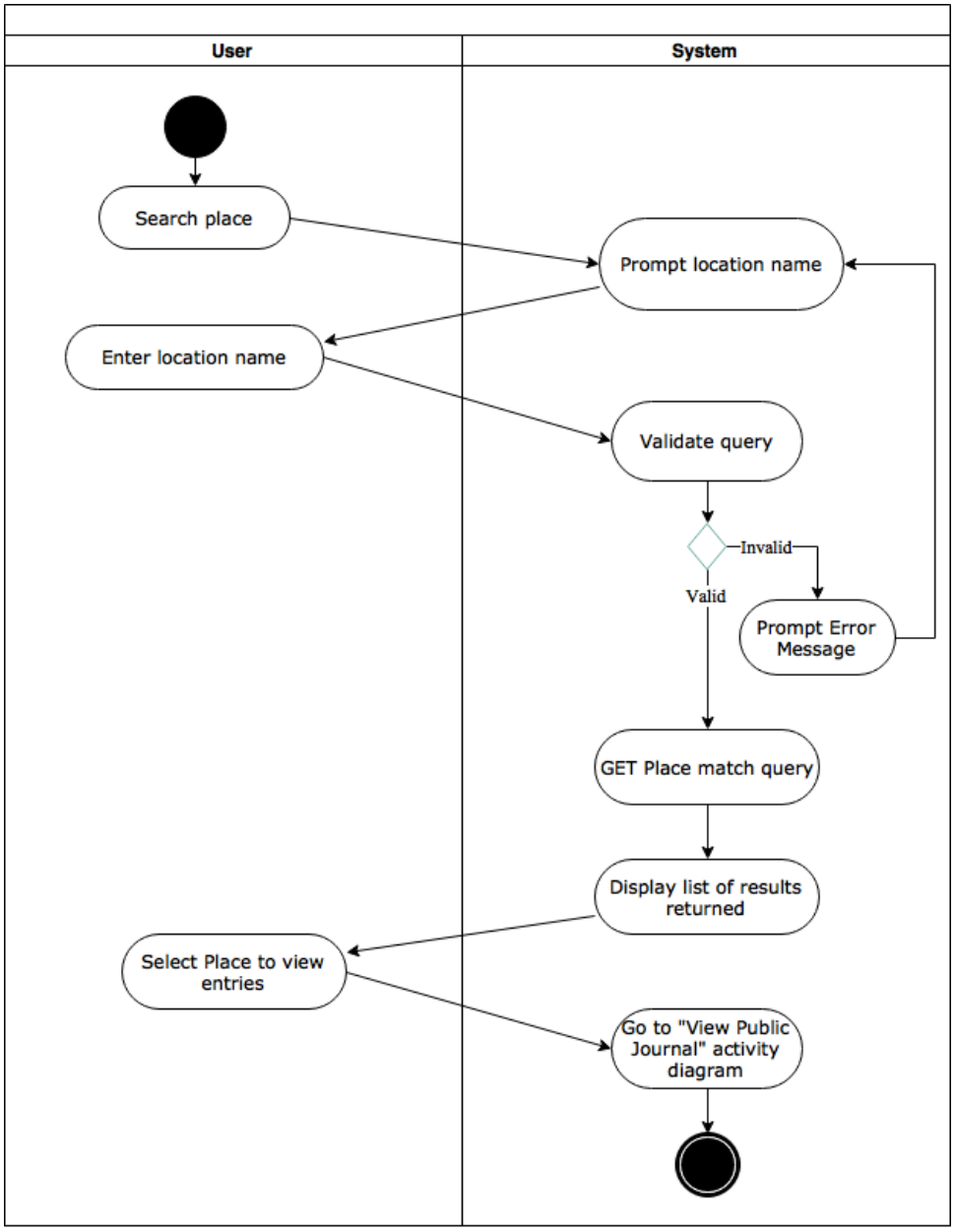


Figure 3-3-13 Search Place activity diagram

User request to search place. System prompt user to enter location name. User enter location name or keywords then submit query for system to validate query. If the query is valid, system perform HTTPS request to retrieve a list of places that matches query via Google Place API. Upon successful retrieval of places, system display results in a list. User select a place to view public journal entries and location on map.

n) Sign Out

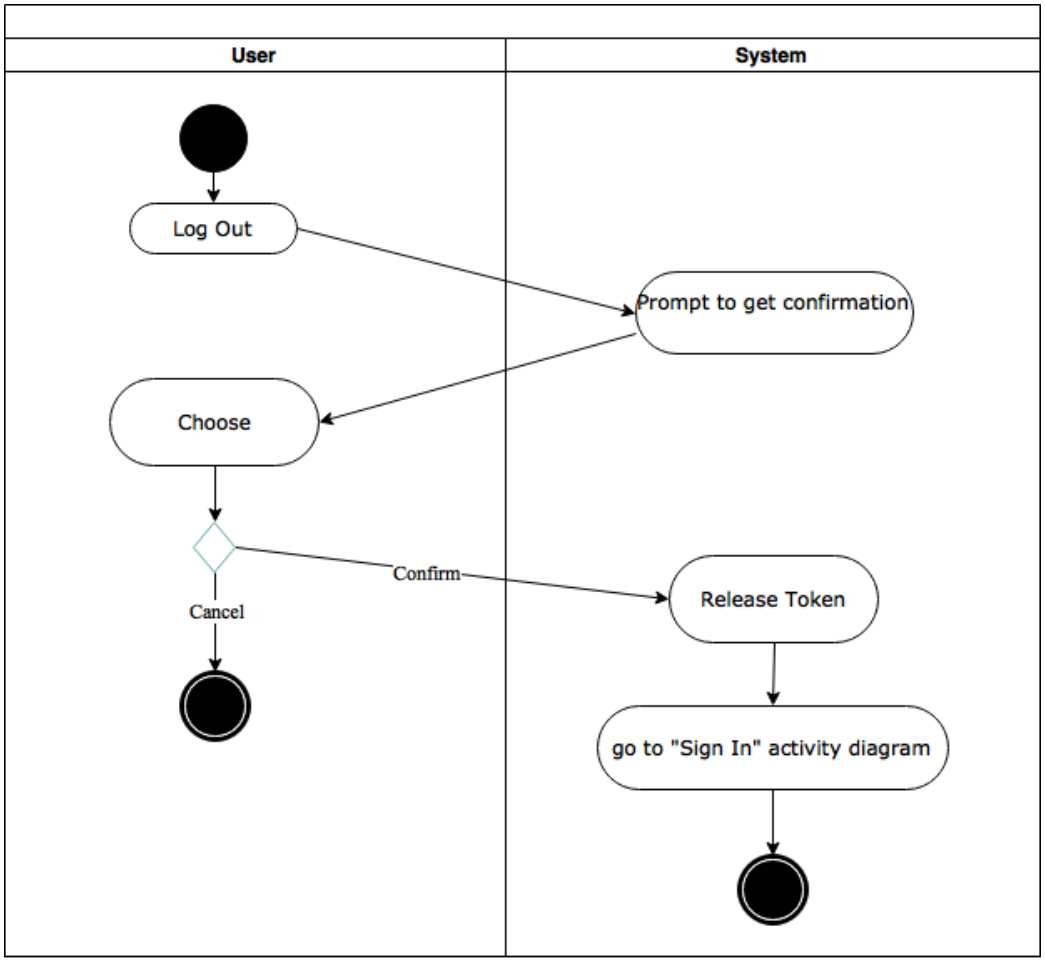


Figure 3-3-14 Sign Out activity diagram

User request to sign out the system. System prompt to get confirmation for sign out. If user select “Confirm”, system release server assigned token and redirect user to sign in interface. Else if user choose to cancel operation, nothing happens.

3.4 Sequence Diagram

Sequence Diagram shows interaction between user and objects in sequence.

a) Manage Journal Entry

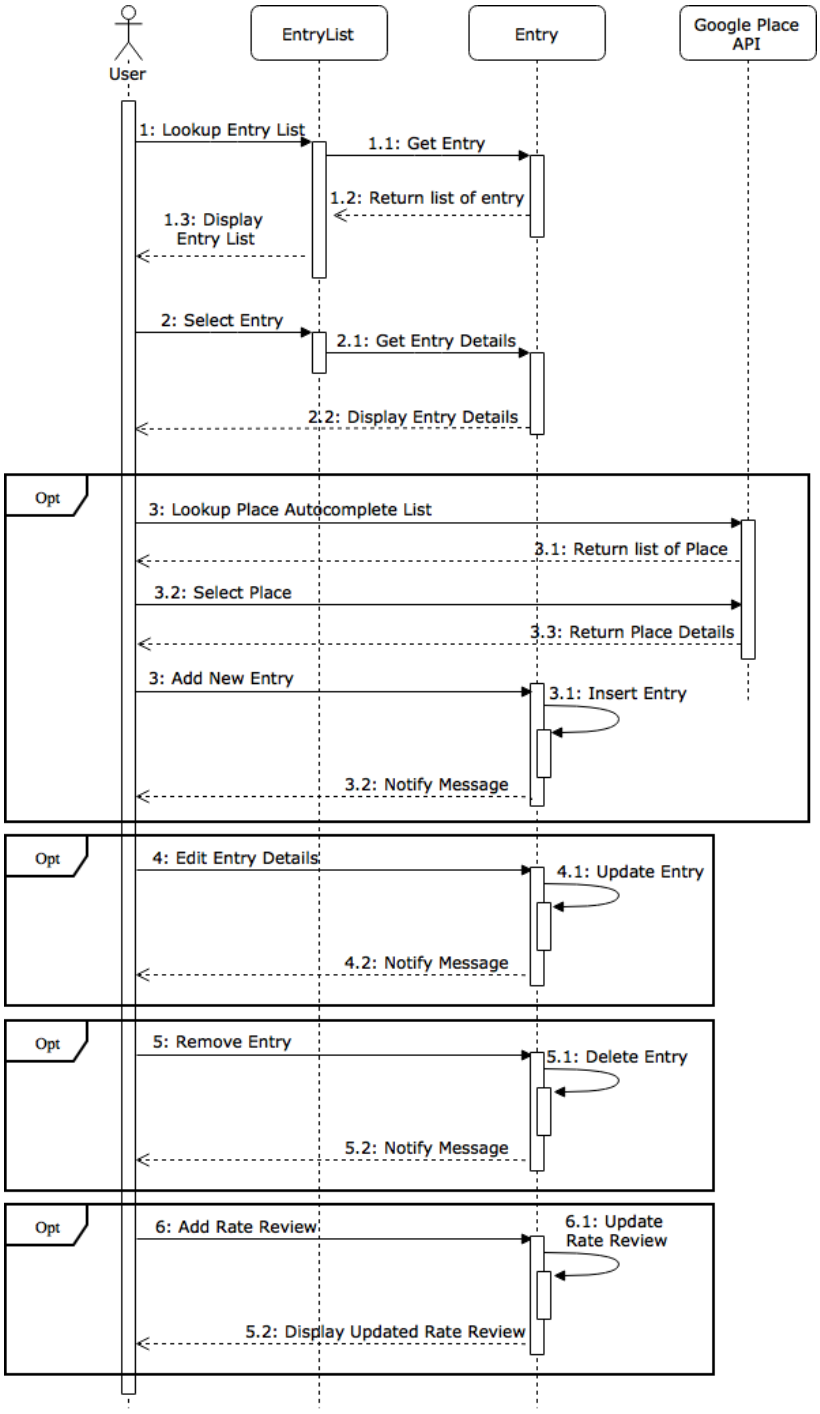


Figure 3-4-1 Manage Journal Entry sequence diagram

b) Manage Profile

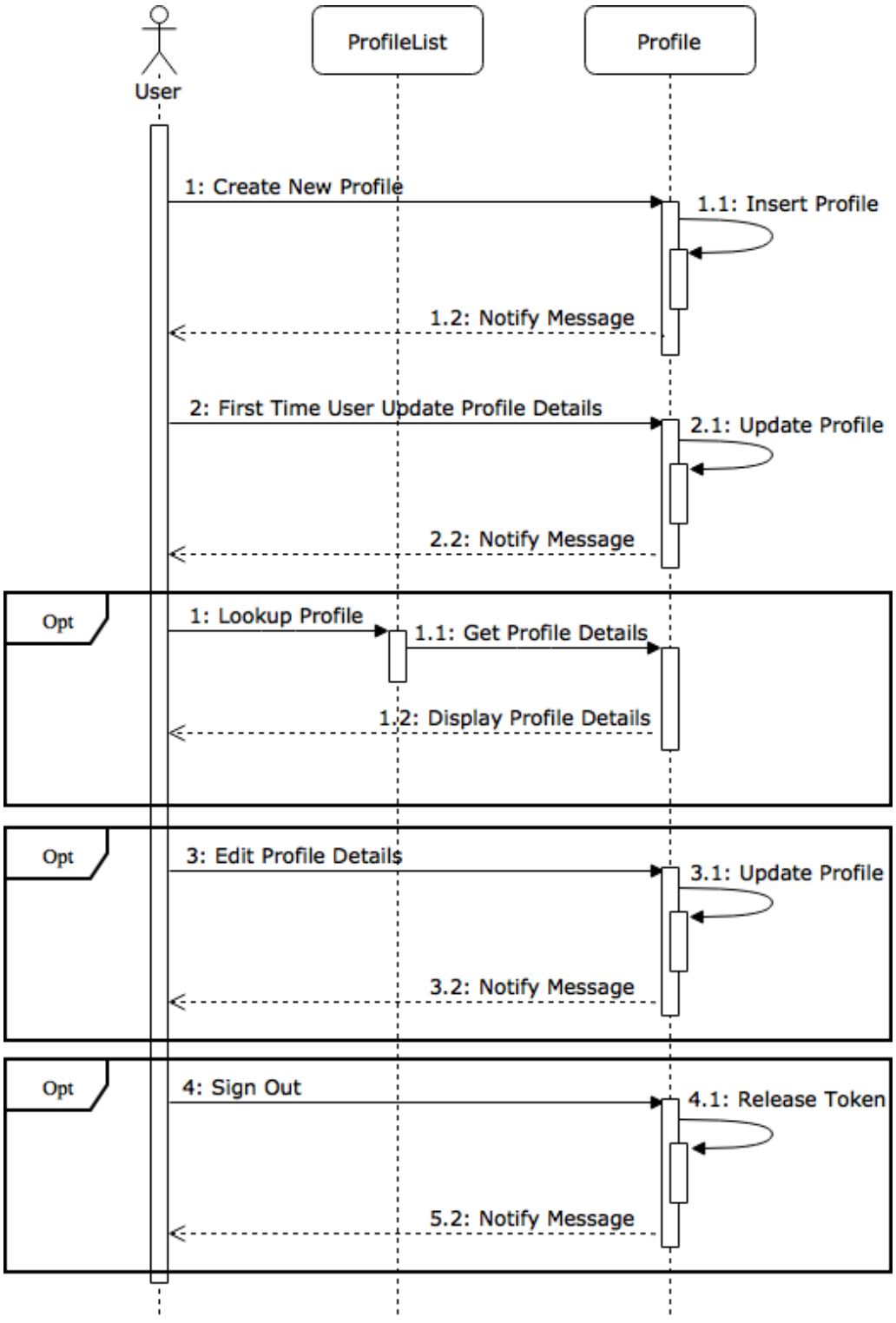


Figure 3-4-2 Manage Profile sequence diagram

c) Search Profile / Place

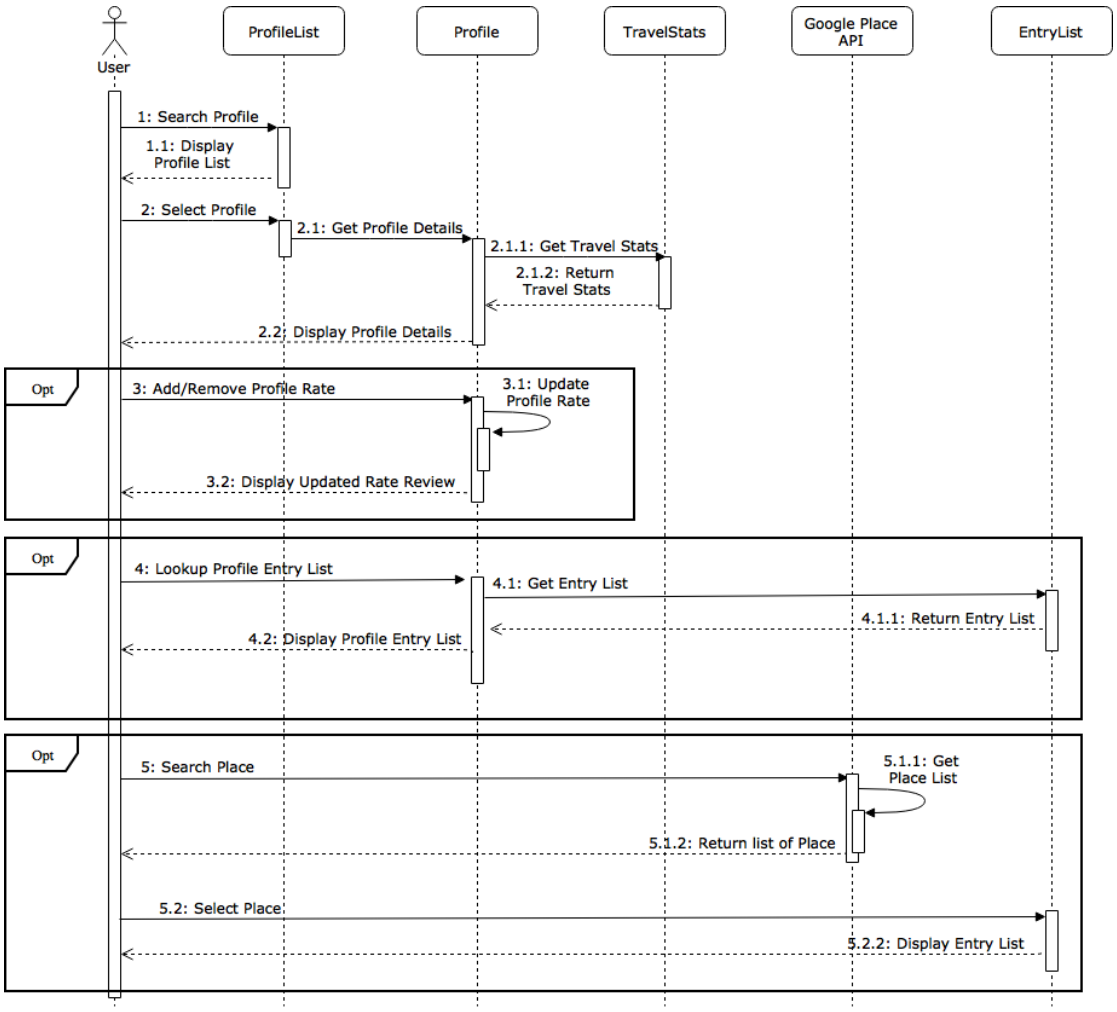


Figure 3-4-3 Search Profile/Place sequence diagram

3.5 Class Diagram for Data Modeling

UML class diagram is used to model the schema for MongoDB entity collections. This diagram illustrates relationship representation approach for document database, which is references documents.

a. Collection identified as Entity

b. Array identified as One-to-Many Relationship

c. Reference-document identified as Relationship

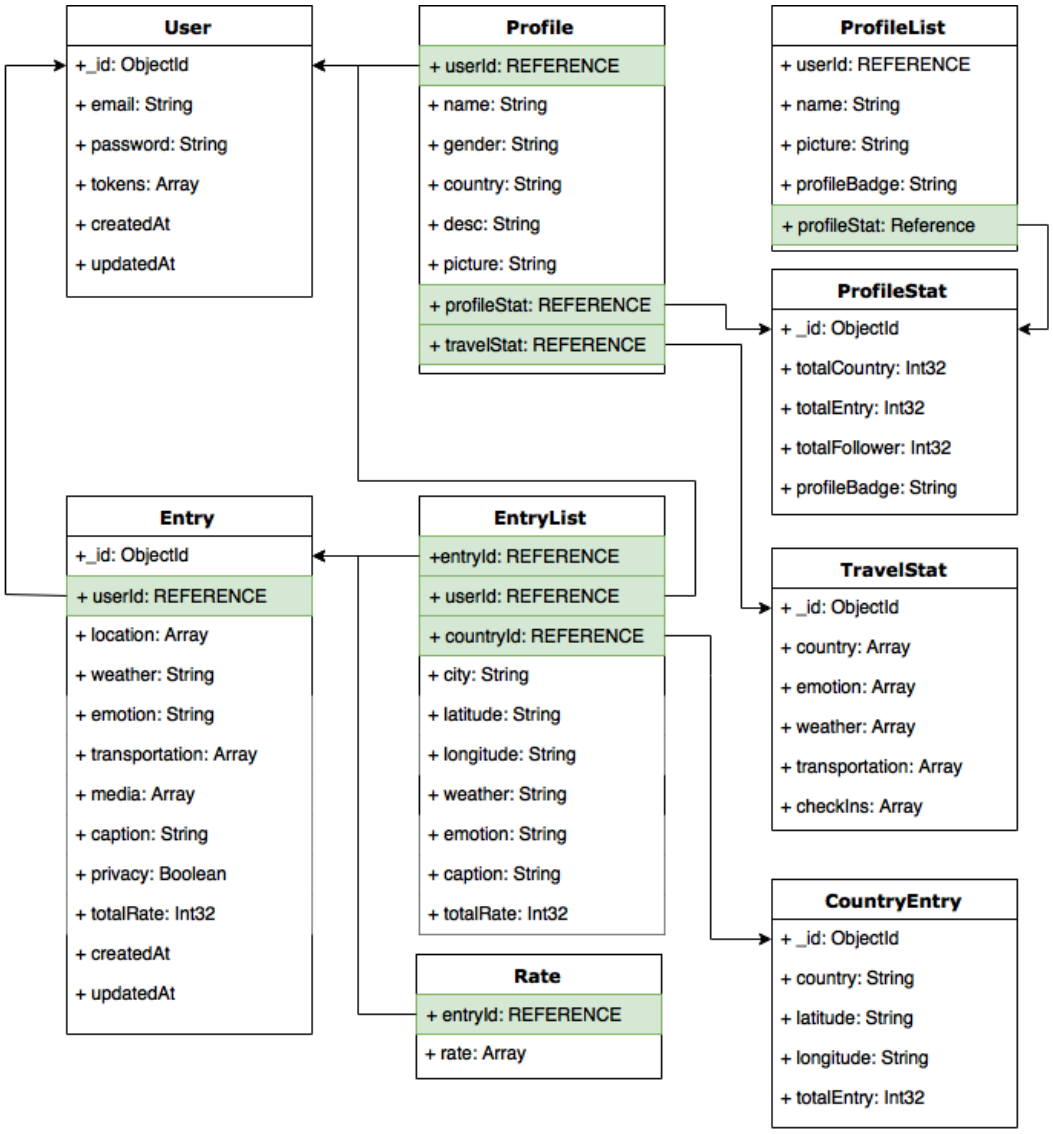


Figure 3-5-1 Data Modeling class diagram

3.6 Data Dictionary

a) User Collection

Name	Type	Description	Remarks
userId	ObjectId	Unique identity of User.	Unique
email	String	User registered email address	Unique, cannot modify.
password	String	Password for authentication	Can modify.
tokens	Array	Passport to access server.	Issue after login.
createdAt	Date	User account creation date.	
updatedAt	Date	User account detail modification date.	

Table 3-6-0-1 User collection data dictionary

b) Profile Collection

Name	Type	Description	Remarks
userId	Reference	Reference to User collection.	Unique
name	String	Username	Not unique, can modify.
gender	String	User's gender.	
country	String	User's residence country.	
desc	String	User's profile description.	
picture	String	User's avatar filename.	
profileStat	Reference	Reference to ProfileStat collection	
travelStat	Reference	Reference to TravelStat collection	

Table 3-6-2 Profile collection data dictionary

c) ProfileList Collection

Name	Type	Description	Remarks
userId	Reference	Reference to User collection	Unique
name	String	Username	
picture	String	User's avatar filename.	

profileStat	Reference	Reference to ProfileStat collection.	
profileBadge	String	Profile level based on travelStats	

Table 3-6-3 ProfileList collection data dictionary

d) ProfileStat Collection

Name	Type	Description	Remarks
profileStatId	ObjectId	Unique ID of profileStat.	Unique
totalCountry	Int32	Total country user check-in.	
totalEntry	Int32	Total entry user created.	
totalFollower	Int32	Total follower user has	
profileBadge	String	Profile level based on travelStats	

Table 3-6-4 ProfileStat collection data dictionary

e) TravelStat Collection

Name	Type	Description	Remarks
travelStatId	ObjectId	Unique ID of travelStat.	Unique
country	Array	List of country user traveled.	
emotion	Object	Consist of total emotion for each type of emotion.	
weather	Object	Consist of total weather for each type of weather.	
transport	Object	Consist of total transport for each type of transport.	
checkin	Object	Consist a list of place user check-in.	

Table 3-6-5 TravelStat collection data dictionary

f) Entry Collection

Name	Type	Description	Remarks
entryId	ObjectId	Unique ID of entry.	Unique
userId	Reference	Reference to User collection	
location	String	Location name	
weather	String	Weather name	

emotion	String	Emotion name	
transport	Array	Consist a list of transport user used.	
media	Array	Consist a list of photo & video user uploaded.	Array item not more than 10.
caption	String	Caption user entered	
privacy	Boolean	Private or Public entry	true == Private
totalRate	Int32	Average rating the entry received.	Each user can only rate once.
createdAt	Date	Date time of the entry created.	

Table 3-6-6 Entry collection data dictionary

g) EntryList Collection

Name	Type	Description	Remarks
entryId	Reference	Reference to Entry collection	
userId	Reference	Reference to User collection	
countryId	Reference	Reference to CountryEntry collection.	
location	String	Location name	
latitude	String	Location latitude	
longitude	String	Location longitude	
weather	String	Weather name	
emotion	String	Emotion name	
caption	String	Caption user entered	no more than 30 words
totalRate	Int32	Average rating the entry received.	Each user can only rate once.
createdAt	Date	Date time of the entry created.	

Table 3-6-7 EntryList collection data dictionary

h) Rate Collection

Name	Type	Description	Remarks
entryId	Reference	Reference to Entry collection.	
rate	Array	List of user and rate user gave.	

Table 3-6-8 Rate collection data dictionary

i) CountryEntry Collection

Name	Type	Description	Remarks
CountryId	ObjectId	Unique ID of Country.	Unique
country	String	Country name	
latitude	String	Country latitude	
longitude	String	Country longitude	
totalEntry	Int32	Total Entry refer the country.	

Table 3-6-9 CountryEntry collection data dictionary

3.7 REST API Index

This section consists a list of base URLs and example of requests & responses.

Title	Base URL	Request	Response
Sign Up	POST /signup	email: roger@g.com password: roger222	user_id: 1 token: sdsdsfWMM
Sign In	POST /login	email: roger@g.com password: roger222	user_id: 1 token: sdsdsfWMM
Edit Profile	POST /account/profile	name: rogerClinton gender: male country: Malaysia desc: … picture: 123.jpg	name: rogerClinton gender: male country: Malaysia desc: … picture: 123.jpg
Upload Avatar	POST /upload/avatar	file: base64 filename: 123.jpg	-
Country Entry	GET /countryentry	-	country_id: 1 country: Malaysia latitude: 1.9234 longitude: 39.4678 total_entry: 2000
Country Entry List	GET /entrylist?countryid=1	-	entry_id: 1 user_id: 1 name: rogerClinton picture: 123.jpg location: KLCC

			caption: … weather: sunny emotion: happy rate: 5 created_at: … latitude: 1.9093 longitude: 40.0976
View Entry	GET /entry?id=1	-	user_id: 1 location: KLCC weather: sunny emotion: happy transport: bus, walk media: … caption: … privacy: false total_rate: 5 created_at: …
Create Entry	POST /entry	location: KLCC country: Malaysia latitude: 1.9078 longitude: 39.0945 weather: sunny emotion: happy transport: bus, walk media: … caption: … privacy: false created_at: …	

Upload Entry media files	POST /upload/entrymedia	file: base64/video filename: …	-
Edit Entry	POST /entry?id=1	location: KLCC country: Malaysia latitude: 1.9078 longitude: 39.0945 weather: sunny emotion: happy transport: bus, walk media: … caption: … privacy: false created_at: …	
Delete Entry	DELETE /entry?id=1		
View Profile	GET /profile?userid=1	-	name: rogerClinton gender: male country: Malaysia desc: … picture: 123.jpg profilestat: 1 travelstat: 1
View Profile Stat	GET /profilestat?id=1	-	total_country: 12 total_entry: 50 total_follower: 300 profile_badge: 2
View Travel Stat	GET /travelstat?id=1	-	country: [] emotion: [] weather: [] transport: []

			checkins: []
Entry List	GET /entrylist?userid=1	-	entry_id: 1 user_id: 1 name: rogerClinton picture: 123.jpg location: KLCC caption: … weather: sunny emotion: happy rate: 5 created_at: … latitude: 1.9093 longitude: 40.0976
Search Profile	GET /profilelist?name=roger	-	user_id: 1 name: rogerClinton picture: 123.jpg profile_badge: 2 profileStat: 1
Search Weather	http://api.openweathermap .org/data/2.5/weather?q= ${city} &APPID= 9b85550b5edf68bc55 19b7208dc55084	-	city: Beijing weather: sunny temperature: 24°c

Table 3-7-1 API index

3.8 Context Diagram

Context diagram illustrates the generalized relationship between system and external entities. Context diagram also is known as level 0 data flow diagram.

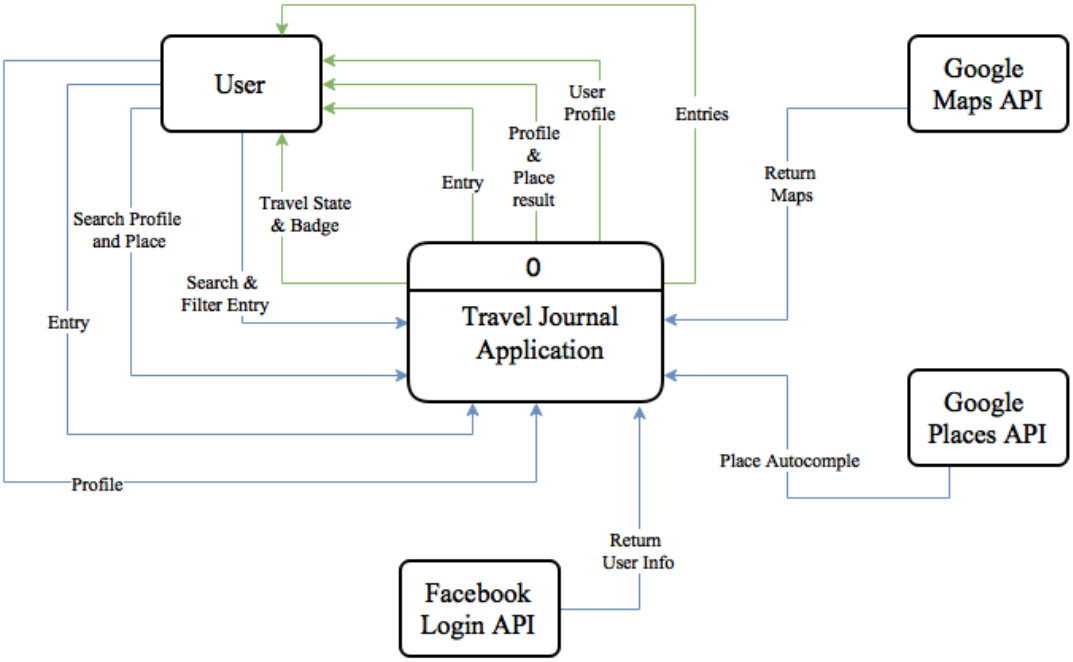


Figure 3-8-1 Context Diagram

3.9 Level-1 Data Flow Diagram (DFD)

This diagram illustrates data flow between system and subsystems. The level-1 DFD shall illustrate all functionality of the whole system.

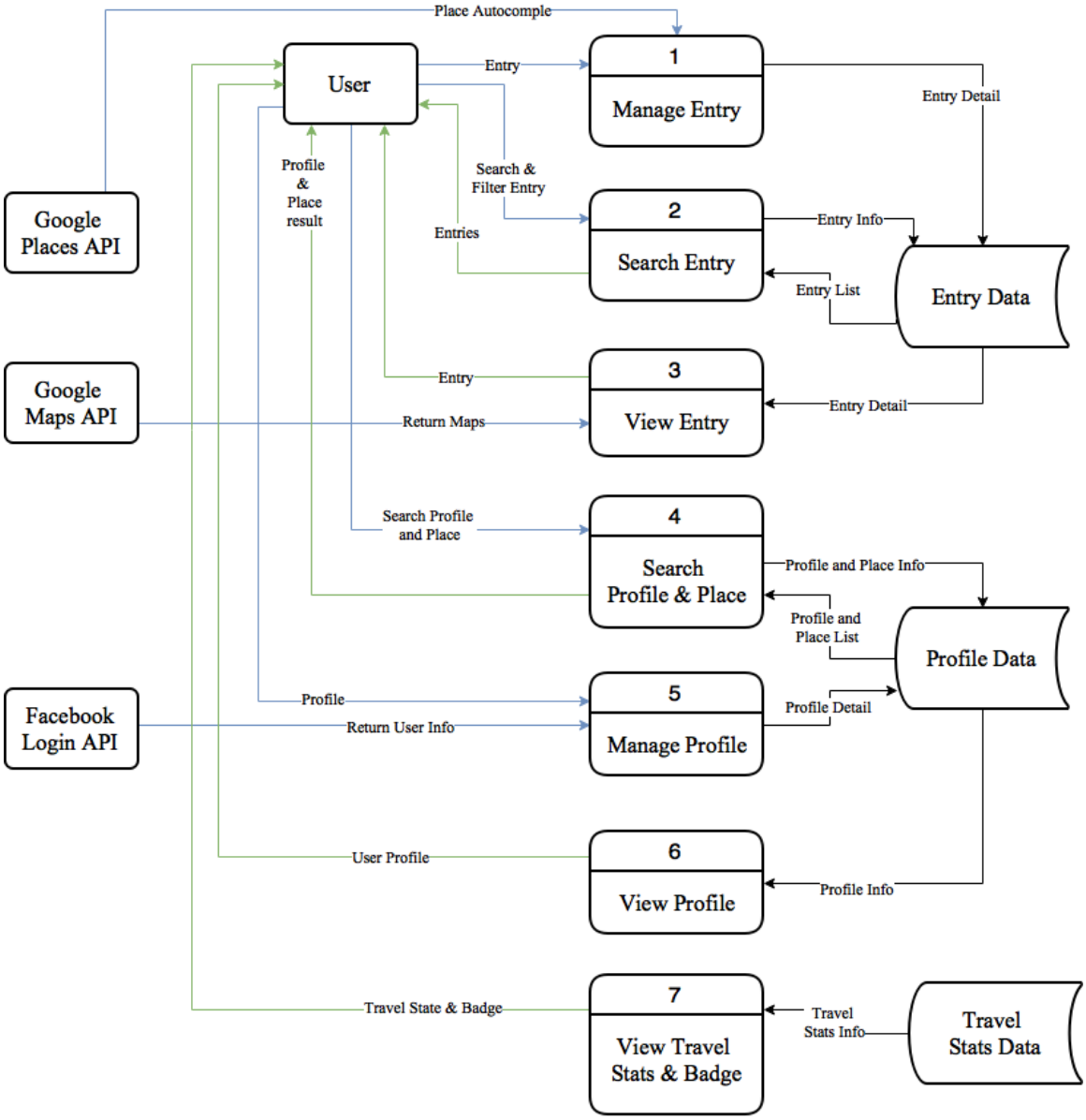


Figure 3-9-1 Level-1 DFD

3.10 System Architecture Diagram

This diagram illustrates a conceptual model of the system structure. This diagram describes the system as a 3-tier architecture that uses mobile application for presentation and semi-logic layer. While the business logic layer is taking care under Node.js and Express.js. Finally, data layer consists of MongoDB is use for application data storage read and write through a network connection.

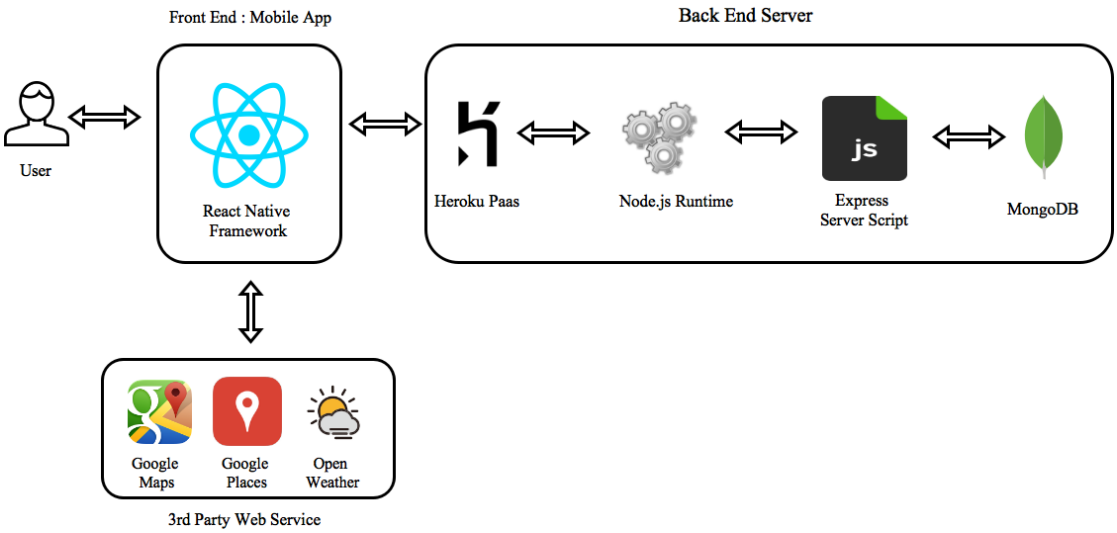


Figure 3-10-1 System Architecture Diagram

User access the travel journal application through smartphones. Data exchange between application database (MongoDB) and client is handled by server-side script (Express.js). The connection between database layer (MongoDB) and business logic layer (Express.js) is established by Node.js Runtime which run on server under cloud platform Heroku.
3rd party web services are involved in the system architecture. They are Google Maps APIs, Google Places APIs, and Open Weather .

Chapter 4: Methodology and Tool

4.1 Software Development Methodology and General Work Procedures

Scrum methodology will be implemented with addition use of Kanban model on each Sprint cycle for the development of proposed project. There are total 14 processes address the specific activities and flow of a Scrum project. These processes are grouped into the following five phases:

Phase	Processes
Initiate	1. Create prioritized product backlog (High to low) a. Backend API module b. Profile management module c. Journal entry management module d. Entry viewing module e. Entry filtering module f. Profile viewing module g. Search profile and place module 2. Conduct release planning for FYP 2 a. Sprint 1: (week 1 – 5) – Staging i. Draft project report ii. Plan estimate user stories iii. Backend API module iv. Profile management module b. Sprint 2: (week 6-8) – Staging i. Draft project report submission ii. Journal entry management module iii. Entry viewing module c. Sprint 3: (week 9 – 10) – Staging i. Entry filtering module

	ii. Profile viewing module iii. Search profile & place module d. Sprint 4: (week 11 – 12) – Testing i. User acceptance test plan ii. User acceptance test e. Sprint 5: (week 13) – Production i. Project report submission. ii. Application pilot run.
Plan and Estimate	3. Create user stories a. Break down each module into small tasks. 4. Approve, estimate, and commit user stories a. Present and get feedback from supervisor. b. Estimate effort to develop functionalities in each story. 5. Create tasks a. Visualize tasks on Kanban board to do list. 6. Estimate tasks a. Estimate effort to accomplish task in task list. 7. Create Sprint backlog a. Add Sprint and Sprint tasks into Trello log
Implement	8. Create deliverables a. Move tasks from to do list to on-going list. 9. Conduct daily progress review a. Move completed task to completed column. b. Review progress and identify challenges. 10. Adjust prioritized product backlog a. If challenges cannot be tackled.
Review and Retrospect	11. Demonstrate and validate Sprint a. Demonstrate Sprint deliverables to supervisor or relevant stakeholders.

	b. Conduct User Acceptance Test (UAT) 12. Retrospect Sprint a. Update Trello log and add recommendations.
Release	13. Present final deliverables a. Oral, poster, and app demo to supervisor and moderator. 14. Retrospect project a. Wrap FYP 2.

Table 4-1-1 Scrum phases and processe

4.2 Travel Level Rewarding Methodology

Travel statistics and achievements are accumulate and distribute according to following diagrams.

a. Travel level color representation and total points earn to complete a level.

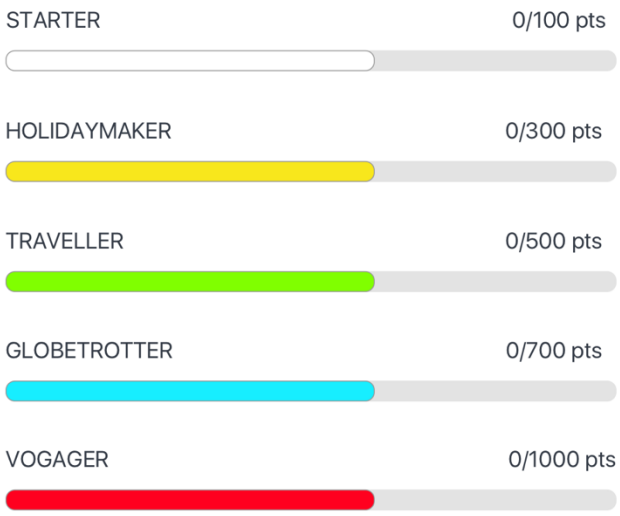


Figure 4-2-1 Travel level and points

b. Points distribution method

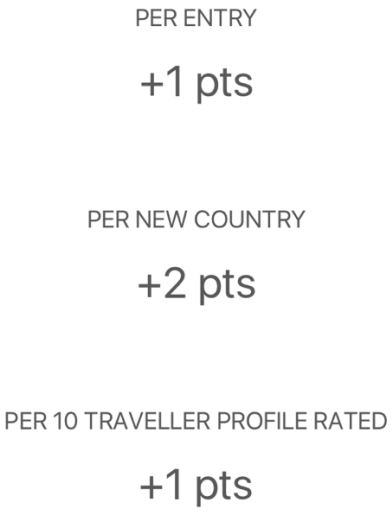


Figure 4-2-2 Travel points distribution

4.3 Technologies Involvement

a. Sketch

Sketch is used to create high-level application screen flow design. Sketch is a vector design tool for user interface design. It allows 3rd party plugins installation and export designs to JSX, HTML, CSS, PNG, GIF, PDF, etc. Sketch also allow preview screen design on a phone while working on the design.

b. React Native framework

React Native is an open source cross-development platform that helps developers to reuse code to build native applications. React Native aim to bring the power of ReactJS, web component-based frontend applications development to mobile application development.

React Native application logic code is written in JavaScript ES6, the sixth major release of standard scripting language. JavaScript is an object-oriented programming language allow two ways communication within web browsers.

React Native application UI components are written in JavaScript XML (JSX). JSX is a syntax extension for ReactJS that adds support for writing HTML tags in JavaScript.

c. Express.js framework and MongoDB and Node.js runtime

In this project, Express.js is used to create a server-side application and REST API endpoints. MongoDB an open source database that uses a document-oriented NoSQL database is used as application remote database. While Node.js is used to connect Express.js to MongoDB. Node.js is a JavaScript runtime environment useful in building server-side applications.

d. Visual Studio Code 2

The React Native application uses VS Code 2 as text and source code editor. This tool able to provide plug-ins support for application development such as code debugging, Express.js syntax autocomplete, JavaScript syntax autocomplete, and React Native syntax autocomplete.

e. Apple iOS simulator and Google Android SDK emulator

Simulator for both platforms is used to test build React Native application during the development process.

f. Trello

The proposed project uses Trello as Scrum and Kanban productivity tool to organizes and manage product backlog and Sprint backlog.

g. Github

This project also uses version control system and repository to store version of applications and release new versions during project staging and production.

h. Heroku

Heroku is a cloud platform to provide Platform-as-a-service (Paas) for proposed project web application and database hosting. The proposed project utilize free tier package offered by Heroku. This package includes 1000 monthly pool of dyno hours (single thread/process instance), 512 MB RAM, direct application deployment via Github repository, and unlimited 3rd party technologies add-on.

Chapter 5: User requirements

5.1 Functional requirements

a. Register and manage user account

   This application should allow user to register a user account and manage account information from settings.

b. Create and manage journal entries

   This application should allow user to create new journal entry and manage journal entry.

c. View travel statistics

   This application should allow user to track their travel statistics accumulated from previous entries created into the application.

d. View travel level and badge

   This application should allow user to find out their account travel level and badge rewarded to the level.

e. Filter journal entries

   This application should allow user to filter journal entries based on location, emotion, weather, transportation, and dates from previously created entries.

f. View journal entries on map

   This application should allow user to look for previously created entries on map based on country location.

g. View journal entries on list view

   This application should allow user to look for previously created entries on list view based on most recent create entry.

h. View journal entry details

This application should allow user to look for one journal entry details by selecting the journal entry from a list of journal entries.

i. View public journal entries on map

This application should allow user to look for public journal entries created by other fellow travelers on map based user current location.

j. Search for profiles and places

This application should allow user to search for profiles and places by entering keywords.

k. Login application

This application should allow user to perform login action by submitting an email address and password or login via social media connect.

l. Logout application

This application should allow user to perform logout action by clicking logout button in settings.

5.2 Non-functional Requirements

a. Usability

This application should provide useful function and features that user to can learn and perform their intended task easily.

b. User friendliness

This application should have a minimalist user interface and provides interactive user experiences for user.

c. Performance

The application should retrieve journal entries and display the search result in a rationally time.

d. Accuracy

This application should provide correct results for profile and place searching, and journal entries search filtering user queried.

e. Extensibility

This application should be able to extend with additional functionality or adjustment of existing functionality without compromising the existing application.

f. Effectiveness

This application should provide functionality that allows user to document travel journal in more effective way.

5.3 Verification Plan

a. Register

No	Test Steps	Attribute and Value	Expected Result
1	Register using **Correct** details and submit.	Email: shalonteoh@1utar.my Username: shalonteoh Password: P@ss0rd Confirm password: P@ssw0rd	User account is created successfully. Go to **Update Profile**.
2	Register using **Used** email and submit.	Email: shalonteoh@1utar.my Username: shalonteoh Password: P@ss0rd Confirm password: P@ss0rd	User is rejected to register followed by an error message. E.g. "Invalid Email"
3	Register using **Incorrect** details and submit	Email: shalonteoh@gmail.com Username: shalonteoh Password: shalon Confirm password: shalon	User is rejected to register followed by an error message. Value: "Invalid email, password."
4	Register with **Empty** details and submit.		User is rejected to update details followed by an error message. Value: "All details are required."

Table 5-3-1 Register verification plan

b. Update Profile

No	Test Steps	Attribute and Value	Expected Result
1	Redirected with from **Register** or **Settings**		Display **Correct** account detail if available in editable fields.
2	Update using **Correct** details and submit.	Profile Picture: [image] Username: Shalon Teoh Gender: Female Country: Malaysia Description: Royale with cheese.	Account detail is updated successfully. If 1st time user, go to **Landing page**.
3	Update with **Empty** details and submit.	All details are required except description field.	User is rejected to update, followed by an error message. Value: "Please enter require fields."

Table 5-3-2 Update Profile verification plan

c. Login

No	Test Steps	Attribute and Value	Expected Result
1	Login using **Correct** details and submit.	Email: shalonteoh@1utar.my Password: P@ssw0rd	User is logged in successfully. Go to **Landing page.**

2	Login using **Incorrect** details and submit.	Email: shalonteoh@gmail.com Password: Password	User is rejected to login followed by an error message. Value: "Invalid login"
3	Login with **Empty** details and submit.		User is rejected to login followed by an error message. Value: "All field is required."

Table 5-3-3 Login verification plan

d. Landing page

No	Test Steps	Attribute and Value	Expected Result
1	Click "**Discover"** icon or redirected from **Login**.		Display map pinned with number of journal entries by country. Display Top 10 popular country with most entries at bottom.
2	Click a country from list of Top 10 popular country with most entries.	country name: Malaysia	Display map pinned with coordinate of all public entries available in Malaysia. Display public entry list based in Malaysia at bottom.
3	Click a country pin on map.	country name: Malaysia	Display map pinned with coordinate of all public entries available in Malaysia.

			Display public entry list based in Malaysia at bottom.
4	Click an entry from entry list at bottom.	entry id: 1	Move map focus center to coordinate of the entry created.
5	Click an entry pin on map	entry location: Kuala Lumpur	Filter and Display bottom entry list created in Kuala Lumpur.
5.1	Click the same entry pin again	-	Display map pinned with coordinate of all public entries available in Malaysia. Display public entry list based in Malaysia at bottom.
6	Long press a journal entry from entry list at bottom.	entry id: 1	Go to **Quick view journal entry**.

Table 5-3-4 Main page verification plan

e. Search profile and place

No	Test Steps	Attribute and Value	Expected Result
1	Click "**Search**" bar		Display Search screen
2	Search profile with username and submit.	Search: shalonteoh	Display **Correct** list of related account.
3	Select an account from account list.		Go to **View profile**.

4	Search place with place name and submit.	Search: Kampar	Display **Correct** list of related place.
5	Select a place from place list.		Go to **View journal entries on map**.

Table 5-3-5 Search profile and place verification plan

f. View Journal Entry Detail

No	Test Steps	Attribute and Value	Expected Result
1	Select a journal entry from entries.		Display **Correct** journal entry detail.
2	Select "**more**" icon from entry.		Display more menu.
2.1	Select "**Edit"** from menu.		Go to **Edit journal entry detail**.
2.2	Select "**Delete**" from menu.		Journal Entry is deleted successfully. Go to **"Back"**.

Table 5-3-6 View journal entry detail verification plan

g. Edit Journal Entry Detail

No	Test Steps	Attribute and Value	Expected Result
1	Select "**Edit"** from more menu.		Display **Correct** journal entry detail in editable form.
2	Edit using **Valid** detail and submit.	Date Time: 07/12/2017 12:00AM Emotion: Happy Weather: Sunny Location: Kampar, Perak Thought: This is a beautiful day. Images Videos: [images videos] Transportations: Bus No.12 Learns: Always walk on the left side , Use touch & go card.	Journal Entry is updated successfully. Go to **View journal entry detail**.
3	Edit using **Invalid** detail and submit.		User is rejected to update journal entry, followed by an error message. Value: "Invalid detail."

Table 5-3-7 Edit journal entry detail verification plan

h. View Profile

No	Test Steps	Attribute and Value	Expected Result
1	Click "**Profile**" icon at bottom tab or redirect from **Search**		Display **Correct** profile, and content.

2	Select **view travel stats** from view tab.		Display **Correct** travel stats.
3	Select **view journal entries on list** from view tab.		Go to **View journal entry on list**.
4	Select **view journal entries on map** from view tab.		Go to **View journal entry on map**.
5	Click "**Settings"** icon.		Go to **Settings**.

Table 5-3-8 View profile verification plan

i. View Journal Entries on Map

No	Test Steps	Attribute and Value	Expected Result
1	Click **"View journal entries on map"** or redirect from **Search**		Display map pinned with number of return result and list of journal entries.
2	Select a journal entry from entries.		Go to **View journal entry detail.**
3	Long press a journal entry from entries.		Go to **Quick view journal entry**.
4	Click "**filter**" icon.		Go to **Journal entries search filtering**.

Table 5-3-9 View journal entries on map verification plan

j. View Journal Entries on List

No	Test Steps	Attribute and Value	Expected Result
1	Click "**View journal entries on list"**.		Display **Correct** list of journal entries.
2	Select a journal entry from entries.		Go to **View journal entry detail.**
3	Click "**filter**" icon.		Go to **Journal entries search filtering**.

Table 5-3-1 View journal entries on list

k. Quick View Journal Entry

No	Test Steps	Attribute and Value	Expected Result
1	Long press a journal entry from entries.		Display pop up window with **Correct** preview journal entry.
2	Click "**View Detail**" from pop up window.		Go to **View journal entry detail**.

Table 5-3-2 Quick view journal entry verification plan

l. Create Journal Entry

No	Test Steps	Attribute and Value	Expected Result
1	Create using **Valid** details and submit**.**	Date Time: 07/12/2017 12:00AM Emotion: Happy Weather: Sunny Start Location: Kampar, Perak	Journal entry is created successfully.

		Destination: Kuala Lumpur, MY Text: This is a beautiful day. Images Videos: [images videos] Transportations: Bus No.12	Go to **Landing page**.
2	Create with **Invalid** details and submit.		User is rejected to create journal entry followed by an error message. Value: "Invalid detail."

Table 5-3-3 Create journal entry verification plan

m. Journal entry search filtering

No	Test Steps	Attribute and Value	Expected Result
1	Click "**filter**" icon.		Display popup window with filter selection.
2	Filter using location, emotion, weather, and dates. Click submit.	Filter: Location: Kampar Emotion: happy Weather: sunny Dates: 07/12/2017 – 10/12/2017	Dismiss popup window. Display **Correct** list of journal entries.

Table 5-3-4 Journal entry search filtering verification plan

n. Settings

No	Test Steps	Attribute and Value	Expected Result
1	Click "**Settings"** from profile.		Display list of settings.

2	Select "**Edit Profile**" from setting list.		Go to **Update profile.**
3	Select "**Change Password**" from setting list.		Go to **Change password.**
4	Select "**Logout**" from setting list.		User is logged out successfully. Go to **Login.**

Table 5-3-5 Settings verification plan

o. Rate Entry

No	Test Steps	Attribute and Value	Expected Result
1	Select an entry, select "Star" icon		Total number of star is updated by adding one. Highlight "Star" icon.
2	Select an entry, deselect "Star" icon		Total number of star is updated by decrease one. Unhighlight "Star" icon.
3	Select or Deselect "Star" icon without Internet connection		User is rejected to update total number of star.

Table 5-3-6 Rate Entry verification plan

Chapter 6: Project Challenges and Future Improvements

6.1 Implementation Issues and Challenges

Following are the issues for the implementation phase of the project:

a. **React Native application high device processing power consumption.**

   Runing iOS application could take up to 80% of device processing power consumption. This is due to display of animated map on landing page with continuous map interface gesture manipulation.

b. **Require an API integration for user travel level calculation**

   Calculation of user travel level requires the accurate total number of countries and provinces in the world. The issue is finding the source to retrieve latest and accurate number.

Following are the challenges for implementation phase of project:

a. **Possible of deprecated React Native API due to frequent update by Facebook**

   ListView API is not supported by official React Native api. An improved implementation of list called “Flatlist” was introduced to replace ListView. Cases like this could happen very frequent as Facebook Inc. release a new update for React Native every 2 weeks. Therefore developers are advised to implement new APIs to avoid bugs and improve efficiency of application.

6.2 Conclusions and Future Improvements

Cloud-based mobile travel journal application with map tracing is a vision to renovate travel journal and brings great convenience to the travelers. The project was motivated from the practice that most of the travelers do not keep a detail travel journal or do not document travel journal at all. Excuses like concern with writing from scratch, and claimed that current travel journal applications is not useful for their future reference. The proposed project intends to renovate the perception of people towards travel journaling. With the proposed system, travelers can save time from finding a theme to write. With the proposed system's travel rewarding feature, travelers can make use of their effort for journaling and keep track travel milestones at the same time. With the proposed system's entry filtering capability, look back to memory lane just made so easy and faster.

This project has many ways to improve to motivate more travelers to start travel journaling. The best case is to integrate with current popular travel reward program to help travelers fund their trip by writing an exclusive review to convey their experiences. While another idea is to add export and sharing feature to the application, allowing travelers to export their experiences into hardcopy or share their experiences to more audience that not using the application.

REFERENCES

Jones, W. (n.d.). *A brief visual history of travel*. Retrieved February 16, 2017, from gapyear.com: https://www.gapyear.com/history-of-travel/

Morris, J. (2009, September 1). *The allure of travel writing*. Retrieved February 14, 2017, from Smithsonian.com: http://www.smithsonianmag.com/travel/the-allure-of-travel-writing-42681966/?all

Penzu Inc. (2017). *Travel journal: Ideas, Tips, ans how to write a travel journal*. Retrieved February 16, 2017, from Penzu: https://penzu.com/travel-journal-guide

Krook, N. (2013, February 4). *A short history of travel writing*. Retrieved February 14, 2017, from the travel tester: http://www.thetraveltester.com/a-short-history-of-travel-writing/

Go2. (2017). *Culture and History of Travel Blog*. Retrieved February 16, 2017, from go4travelblog: http://www.go4travelblog.com/general-topics/culture-and-history-travel-blog/

Lomas, N. (2016, April 11). *Computing's future? It's in your pocket* . Retrieved February 19, 2017, from techcrunch: https://techcrunch.com/2016/04/11/computings-future-its-in-your-pocket/

Statista. (2016). *Smartphones industry: Statistics & Facts* . Retrieved February 19, 2017, from Statista: https://www.statista.com/topics/840/smartphones/

Statista. (2016). *App Developers - Statistics & Facts*. Retrieved February 19, 2017, from Statista: https://www.statista.com/topics/1694/app-developers/

Bonjournal LLC. (2015, Jan 14). *Home: Bonjournal*. Retrieved February 21, 2017, from Bonjournal: https://bonjourn.al

Cluster Labs, Inc. (2016, July 20). *Home: Tripcast*. Retrieved February 21, 2017, from Tripcast: https://tripcast.co

Sergei Shpygar. (2017, Jan 20). *Mark O'Travel - your travel map. Where you've been*. Retrieved February 21, 2017, from iTunes: https://itunes.apple.com/us/app/mark-otravel-your-travel-map./id866778149?mt=8

Polarsteps. (2017, February 06). *Polarsteps*. Retrieved February 21, 2017, from Home: Polarsteps: https://www.polarsteps.com

Appcrawlr. (2015). *Kid's Travel Journal HD*. Retrieved February 21, 2017, from Appcrawlr: http://appcrawlr.com/ios/kid-travel-journal-hd#authors-description

Shah, A. (2014, July 2). *Why We Prefer Pictures: It's the Way That We're Wired*. Retrieved February 25, 2017, from shutterstock: https://www.shutterstock.com/blog/why-we-prefer-pictures-its-the-way-that-youre-wired

Gaudiano, P. (2014, March). *As it turns out, a picture is not worth a thousand words*. Retrieved February 25, 2017, from wired: https://www.wired.com/insights/2014/03/turns-picture-worth-thousand-words/

Taniguchi, A., Graas, C., & Friman, M. (2014). Satisfaction with travel, goal achievement, and voluntary behavioral change. *Transportation Research Part F: Traffic Psychology and Behaviour , 26*, 10-17.

Stotz, P., & Tack, A. (2014, March). *Travel Score*. Retrieved February 25, 2017, from Mappable: http://mappable.info/projects/travel-score/

Stangarone, J. (2016, June 14). *The Mobile App Comparison Chart: Hybrid vs. Native vs. Mobile Web*. Retrieved March 1, 2017, from mrc-productivity: http://www.mrc-productivity.com/blog/2016/06/the-mobile-app-comparison-chart-hybrid-vs-native-vs-mobile-web/

Kukic, A. (2017, January 10). *Alternatives to Native Mobile App Development*. Retrieved March 2, 2017, from auth0: https://auth0.com/blog/alternatives-to-native-mobile-app-development/

BBVAOPEN4U. (2016, March 26). *REST API: What is it, and what are its advantages in project development?* Retrieved March 3, 2017, from BBVAOPEN4U:

https://bbvaopen4u.com/en/actualidad/rest-api-what-it-and-what-are-its-advantages-project-development

Microsoft Azure. (2016, November 22). *NoSQL vs SQL*. Retrieved March 4, 2017, from Microsoft Azure: https://docs.microsoft.com/en-us/azure/documentdb/documentdb-nosql-vs-sql

Mongo. (2017). *MongoDB and MySQL Compared*. Retrieved March 3, 2017, from Mongodb: https://www.mongodb.com/compare/mongodb-mysql

Susilo, A. R. (2014, April 17). *What are the pros and cons of each "Places" API?* Retrieved March 4, 2017`, from Quora: https://www.quora.com/What-are-the-pros-and-cons-of-each-Places-API

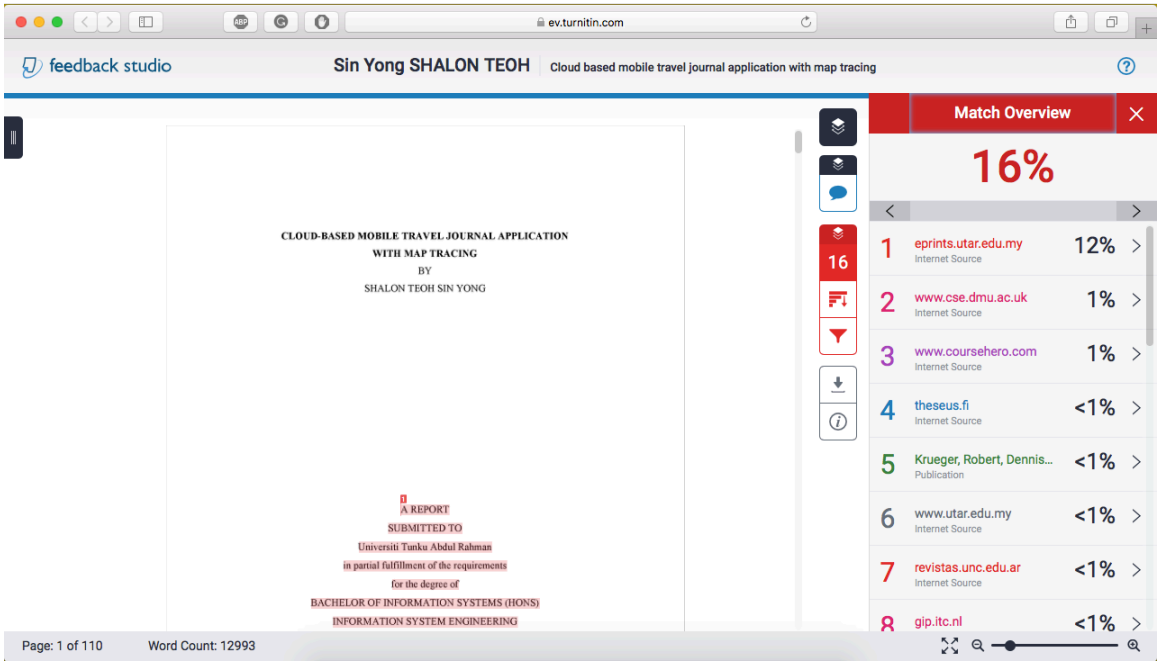
Google Inc. (2017, February 21). *The Google Places API Web Service*. Retrieved March 3, 2017, from Google Places APIs: https://developers.google.com/places/web-service/intro

Wagner, J. (2015, February 2015). *Top 10 mapping APIs*. Retrieved March 4, 2017, from programmableweb: https://www.programmableweb.com/news/top-10-mapping-apis-google-maps-microsoft-bing-maps-and-mapquest/analysis/2015/02/23

CMS. (2008, March 27). *Selecting a developemnt approach.* Retrieved March 5, 2017, from cms.gov: https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf

Smartsheet. (2016, May). *What's the difference? Agile vs Scrum vs Waterfall vs Kanban*. Retrieved March 5, 2017, from smartsheet: https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban

SUMMARY OF PLAGARISM CHECKING RESULT



CLOUD-BASED MOBILE TRAVEL JOURNAL APPLICATION WITH MAP TRACING

INTRODUCTION

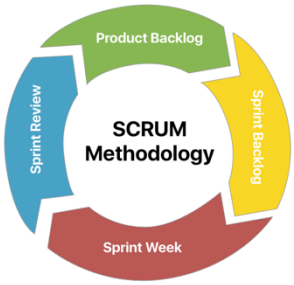
Mobile Travel Journal Application assist travellers to start travel writing with following simple steps.

1. Go to a trip
2. Keep the moment with exact emotions and story
3. Recall moments with family, friends, strangers, yourself
4. Track previous travel and cherish accomplishment

METHODS

The mobile application development uses React Native framework. While SCRUM and Kanban methodology for project management.

React Native



DISCUSSION

This project has many ways to improve & motivate more travellers to start travel journaling. Future improvement can consider providing public entry discovery insight to suggest public entry that meet user's interest.

CONCLUSIONS

This project delivers a React Native mobile app that aim to renovate travel journaling & brings great convenience to the traveller community.