COMP 3059 – Capstone Project I

Software Requirements Analysis and Design Assignment

1.0 Introduction

1.1 Purpose

The purpose of this document is to give a thorough explanation of the Travel Adviser. It will describe the functions and characteristics of the system, its interface, what the system will perform, the conditions that must be met for operation, and how the system will respond to outside stimuli.

1.2 Scope

The main goal of our project is to make stakeholders aware of a unique and exceptional web service called Travel Advisor. By offering reviews, images, and ratings, Travel Advisor aims to assist travellers in finding the places they want to visit. It will provide complete website discovery, a comprehensive user experience, cutting-edge design, and an intuitive interface. Service companies can also advertise their products and services. System administration will be made easier for administrators by the new dashboard. The customer will receive an error-free, user-friendly, and feature-rich web application. Customers may access the web application using a high-responsive, multi-size screen on a phone, tablet, or PC.

The main goal of our project is to make stakeholders aware of a unique and exceptional web service called Travel Advisor. By offering reviews, images, and ratings, Travel Advisor aims to assist travellers in finding the places they want to visit. It will provide complete website discovery, a comprehensive user experience, cutting-edge design, and an intuitive interface. Service companies can also advertise their products and services. System administration will be made easier for administrators by the new dashboard. The customer will receive an error-free, user-friendly, and feature-rich web application. Customers may access the web application using a high-responsive, multi-size screen on a phone, tablet, or PC.

2.0 System Overview

The System Overview section introduces the system context and design.

2.1 Project Perspective

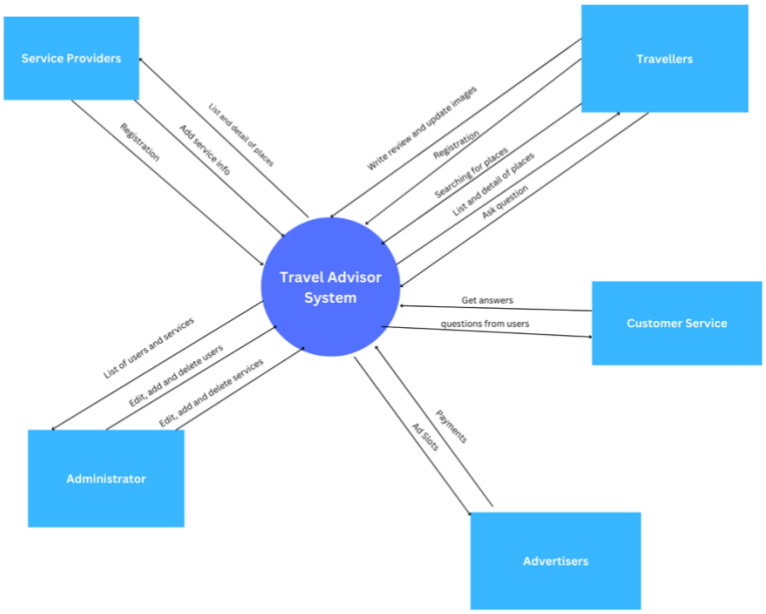
- We are building a new self-contained system which is smaller, isolated units of application focusing on specific functionality. SCS will have an user interface, business logic and persistent storage.
- The characteristics of an SCS are:
    1. Each SCS is an autonomous web application.
    2. One team owns each SCS to develop, test, maintain and operate.
    3. Communication between SCSs or 3rd party systems occurs asynchronous whenever possible using RESTful HTTP or lightweight messaging.
    4. An optional service API can be included.

5. Features of an SCS are made useful though its own user interface using ROCA principles which may be linked to another SCS UI but is not shared.
6. No business logic is shared between SCSs
7. Technical decision within an SCS is made independent from other systems, such as programming language, frameworks, tooling and workflow.

- Advantages of SCS:
  1. Multiple SCSs can be combined to provide greater functionality without adversely impacting the individual SCS.
  2. Each SCS can be independently scalable and replaceable.

2.2 System Context

The System Context describes the resulting software within the business case, including strategic issues in which the system is involved or which it specifically addresses.



2.3 General Constraints

- Our team don't have much experience in building completed product, so that there is some bugs and unprofessional in part of project that the users might not happy.
- We don't have strategy marketing, so that not many users can know about the website.
- We don't have enough time to work close to each other because we still are studying and attending many classes in college.
- Scope in this project still small, it can be scaled but in the future with some additional features such as booking or live navigation system.

- Need more time to organize and optimize the system.

2.4 Assumptions and Dependencies

Assumptions:

- Technology will remain consistent through the project.
- The team members will work and commute closely together until the project completed.
- Project team member will have the resources they need to complete their individual tasks on time, from specialized equipment and software to electricity during working hours.
- All equipment will be in working good condition through the project cycle.

Dependencies:

- Finish-to-start: a wireframe needs to be completed before start coding user interface.
- Finish-to-finish: website features can't be completed until figuring out project requirements.
- Start-to-start: building wireframe, database, website server and user interface … can't be start until project plan has been done.
- Start-to-finish: a new software installation must start before the old installation can be stopped

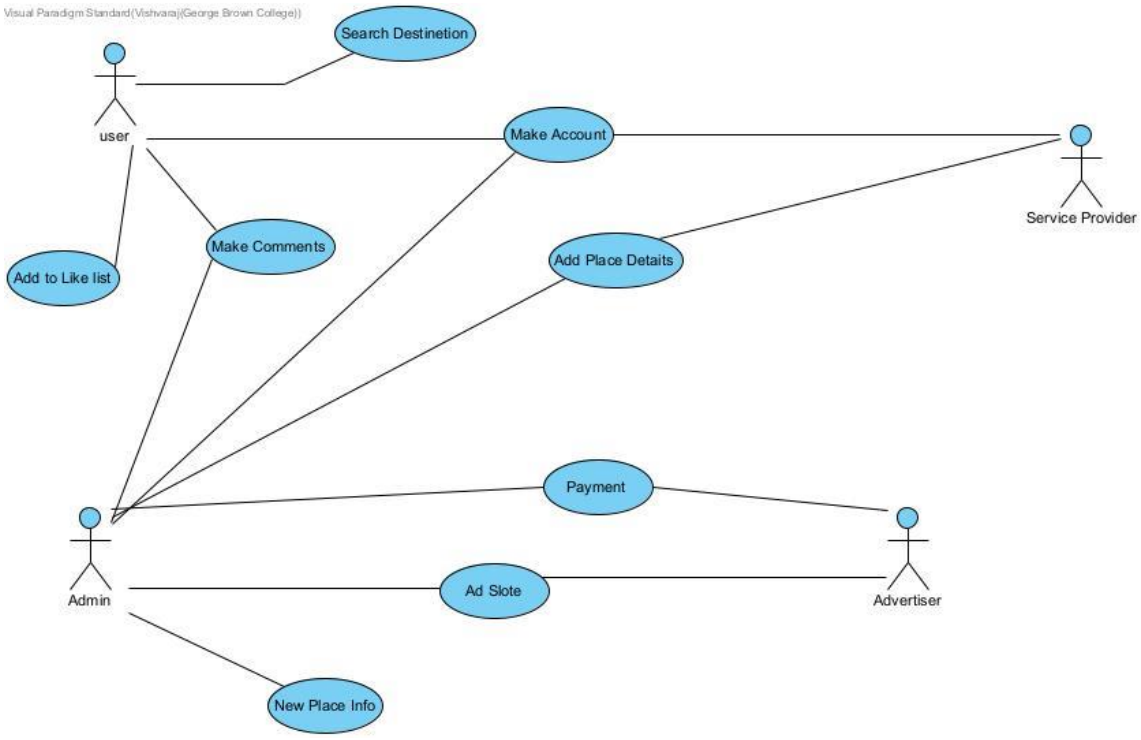
3.0 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.1 Use Cases

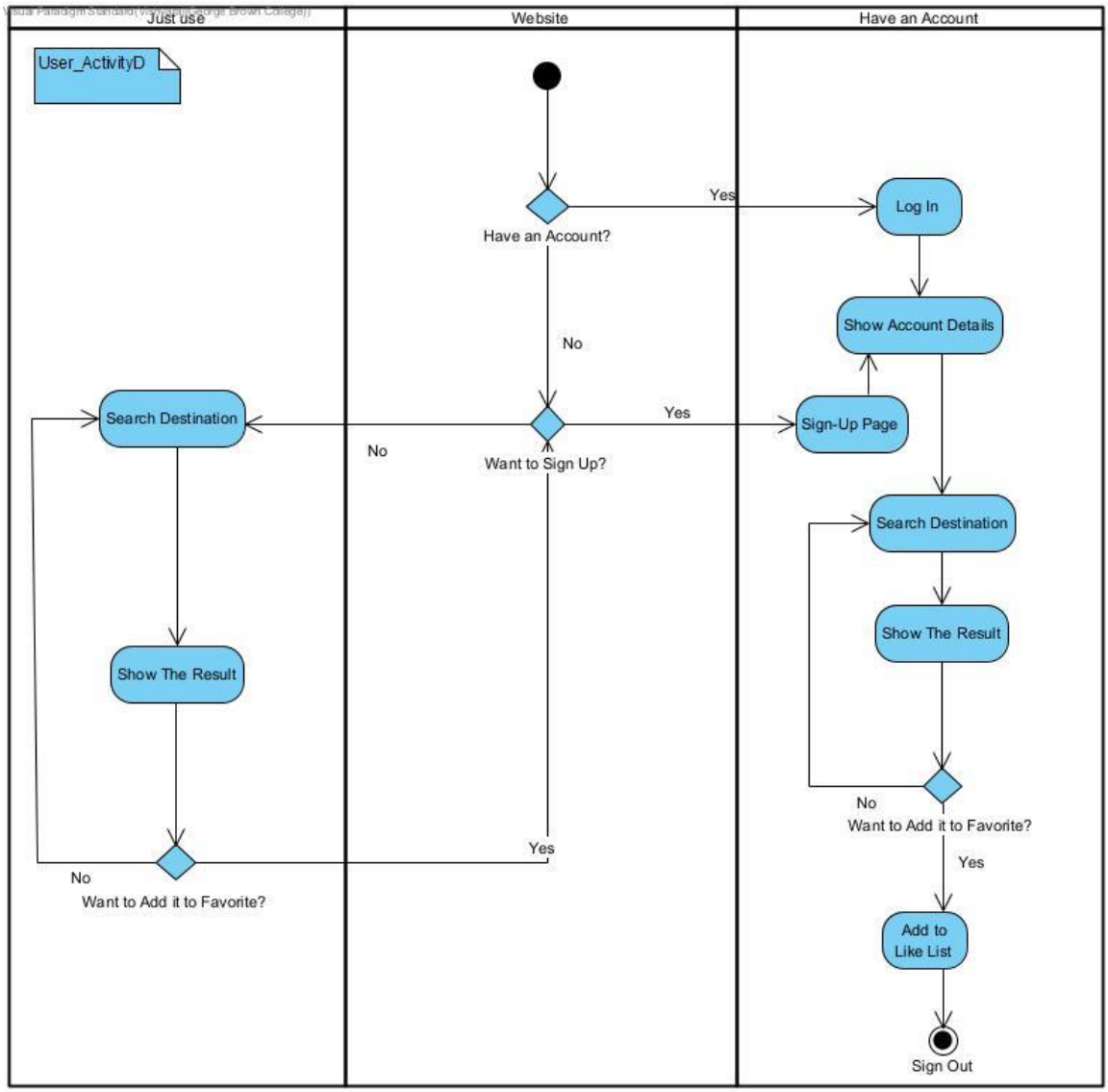
3.1

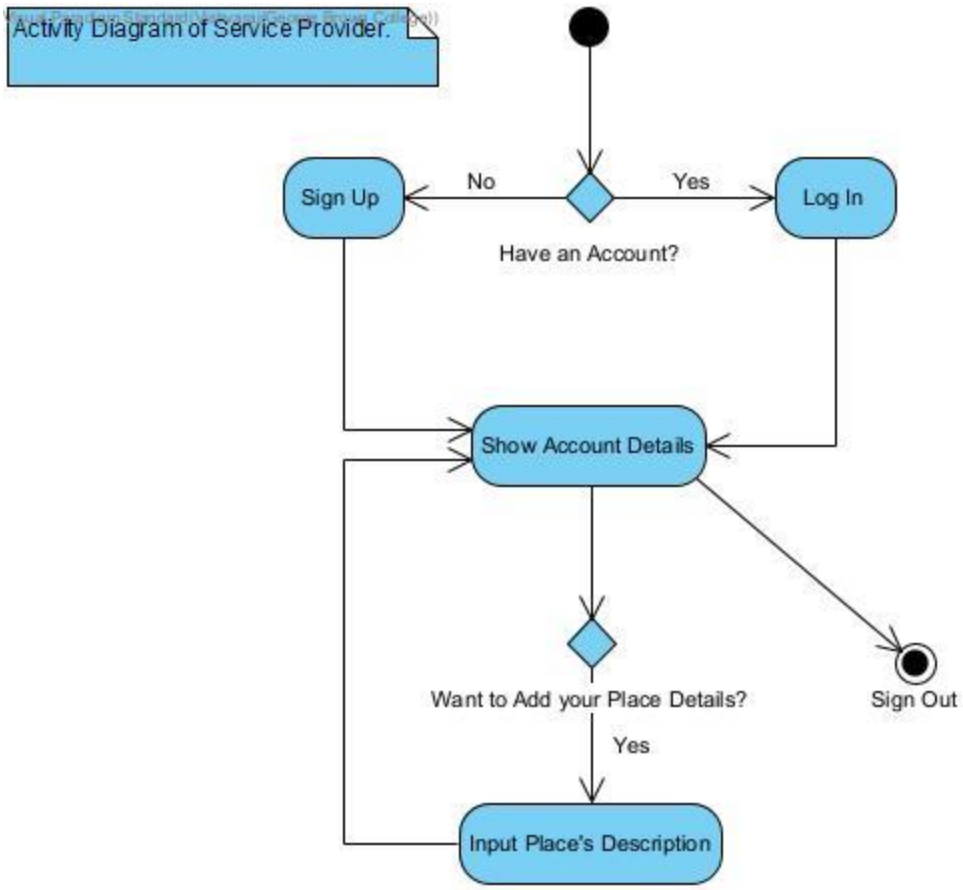
.1 Use Case #1



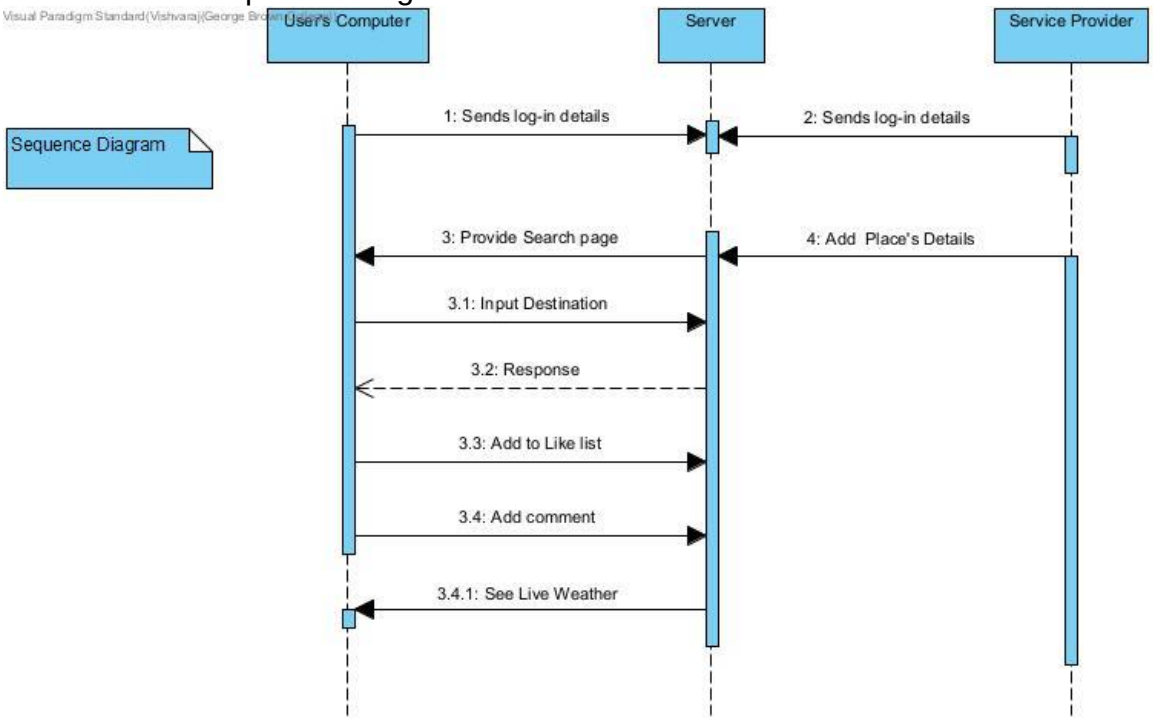
3.2 Data Modelling and Analysis

- Activity Diagrams

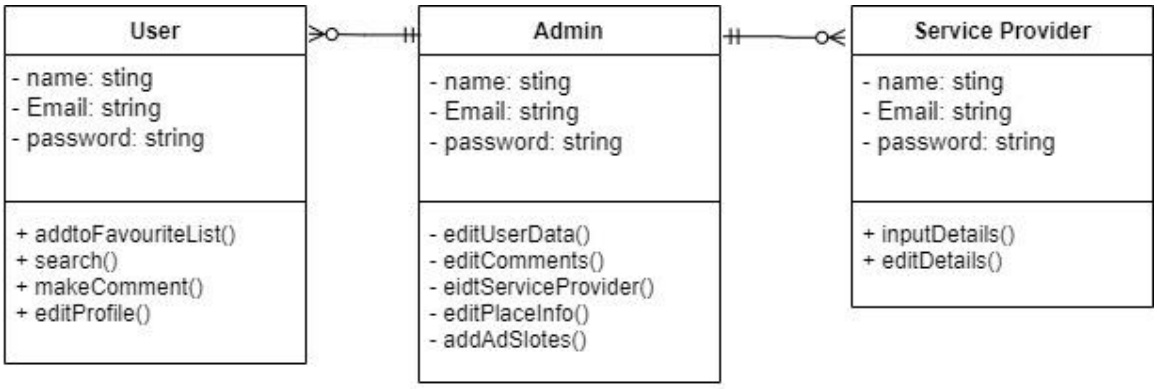

Just use
Website
Have an Account
User_ActivityD
Have an Account?
Yes
Log In
Show Account Details
No
Search Destination
Yes
Sign-Up Page
No
Want to Sign Up?
Search Destination
Show The Result
Show The Result
No
Want to Add it to Favorite?
Yes
Yes
No
Want to Add it to Favorite?
Add to Like List
Sign Out



- Sequence Diagrams

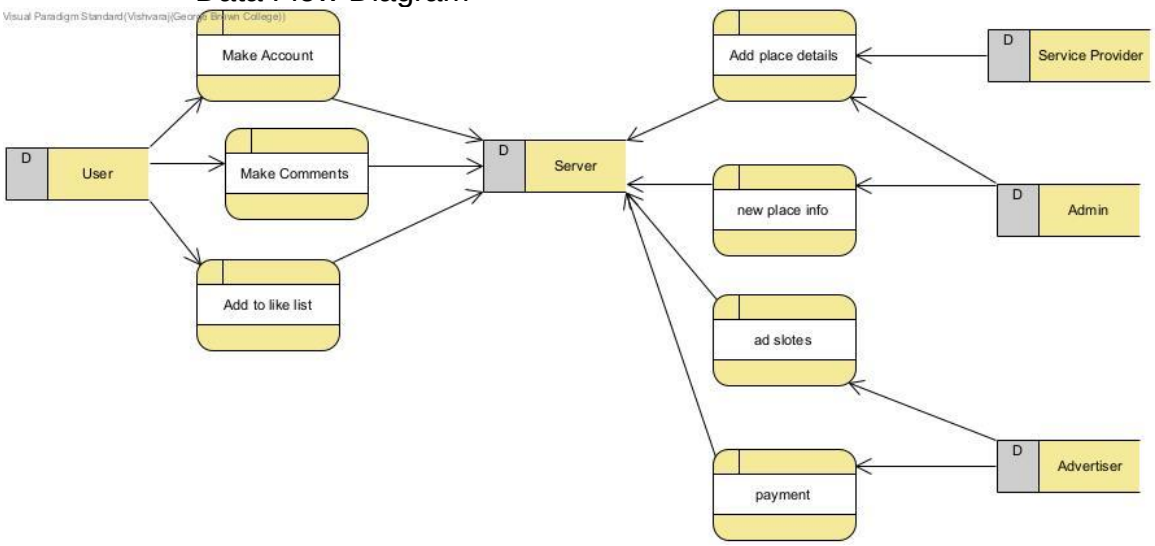


- UML Class Diagram



3.3 Process Modelling

- Data Flow Diagram



4.0 Non-Functional Requirements

- System shall find the easiest and fastest route to the different places
- Website will efficiently work on any device such as PC, Tablet, and smartphone

The non-functional requirements for a system are typically constraints on the functional requirements – that is, not what the system does, but how it does it (e.g., how quickly, how efficiently, how easily from the user's perspective, etc.).

Non-functional requirements may exist for any of the following attributes – Performance, Reliability, Availability, Security, Maintainability, Portability.

Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, etc).

5.0 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc?

6.0 Other Requirements

Additional requirements, if any.

7.0 Approval

The signatures below indicate their approval of the contents of this document.

Project Role	Name	Date
Stakeholder	Vishvarajsinh Rana	08/11/2022
Developer	Vien Nguyen	08/11/2022
Developer	Arnur Azangaliyev	08/11/2022
Developer	Taric Folkes	08/11/2022
Developer	Maitri Patel	08/11/2022