Deliverable #1 Template

SE 3A04: Software Design II – Large System Design

1 Introduction

This section of the SRS should provide an overview of the entire SRS.

1.1 Purpose

a) Delineate the purpose of the SRS

b) Specify the intended audience for the SRS

1.2 Scope

a) Identify the software product(s) to be produced by name (e.g., Host DBMS, Report Generator, etc.)

b) Explain what the software product(s) will, and, if necessary, will not do

c) Describe the application of the software being specified, including relevant benefits, objectives, and goals

d) Be consistent with similar statements in higher-level specifications (e.g., the system requirements specification), if they exist

1.3 Definitions, Acronyms, and Abbreviations

a) Provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS

1.4 References

a) Provide a complete list of all documents referenced elsewhere in the SRS

b) Identify each document by title, report number (if applicable), date, and publishing organization

c) Specify the sources from which the references can be obtained

1.5 Overview

a) Describe what the rest of the SRS contains

b) Explain how the SRS is organized

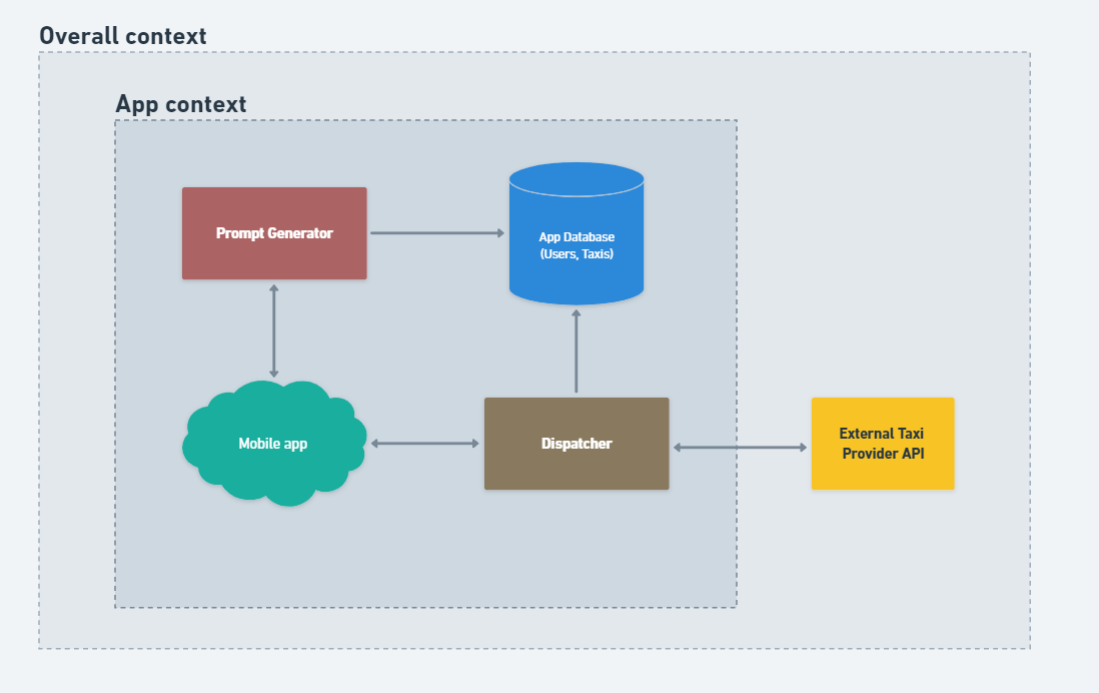
2 Overall Description

This section of the SRS should describe the general factors that affect the product and its requirements. It does not state specific requirements; it provides a background for those requirements and makes them easier to understand.

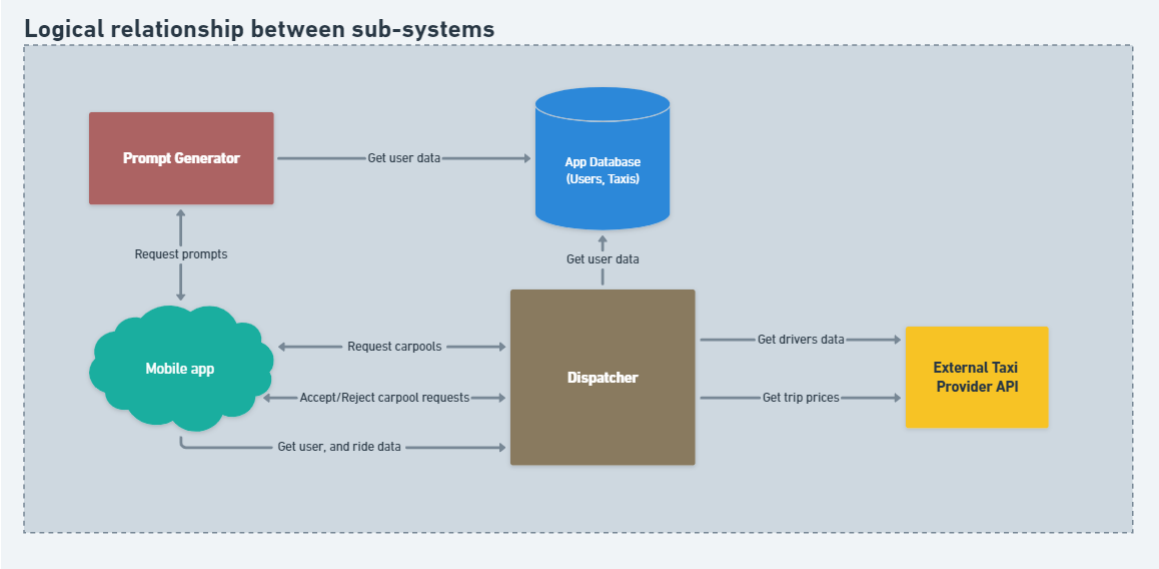
2.1 Product Perspective

a) The product being developed is an Android application which empowers the ability to book carpool via a user-friendly interface using a taxi company. The application will securely store customer personal information such as carpool request histories, and personal data inputted by the user. The product is not self-contained since its functionality depends heavily on the taxi service provider. There are several out-of-scope concerns such as: driver's information input (profile, shift, locations, etc.) and payment processing.

b) The general interaction can be visualized as follows:



2.2 Product Functions



a) The app will be able to book taxi.

- Able to communicate with taxi's service API.
- Able to handle and display taxi data correctly.

b) The app be able to handle carpool scheduling and coordination.

- Able to plan the most optimal route and let the driver follow the calculated route.

c) The app be able to display the estimate time-of-arrival (ETA) of taxis.

- Able to calculate ETA in real-time

d) The app be able to input the taxi's unique identifier via the mobile app.

2.3 User Characteristics

a) Riders

- Riders are users who use the app to request car pool.
- Riders can be anyone from any background of tech-expertise
- Riders are expected to:
  - Input personal information only a few times.
  - Have stable and highly available mobile internet connection.
- Riders might be interested one or more following characteristics of the app:
  - User interface and experience of navigating the mobile app.
  - Realtime taxi information display (ETA, location, route, ...)
  - Ease of taxi booking, and check-in.

b) Taxi Drivers

- Drivers are users who drive and operate taxis, fulfill carpool requests, and ensure the safety of the riders.
- Drivers must be registered with the taxi service and from any background of tech-expertise.
- Drivers are expected to have stable and highly available mobile internet connection.
- Drivers might be interested one or more following characteristics of the app:
  - Accuracy display of real-time pick-up and drop-off information.
  - Accuracy of aggregating and storing of rides information (since it ties directly to their performance).
  - Accuracy display of user's profile summarization.
  - User interface and experience of navigating the mobile app.
  - Ease of accepting/ rejecting rides.

2.4 Constraints

a) **Geographical**: The app can only operate on the land which the taxi service operate.

b) **Technological**: App must be integrated with softwares that the taxi service provider uses.

c) **Fare Determination**: Since the app does't support payment processing directly, the fare has to be determined and processed through the taxi service provider.

d) **Data Privacy Regulation**: Data which transmits and stored within the app needs to be processed in a way which complies with the data protection act.

2.5 Assumptions and Dependencies

a) Assumptions

- Drivers information are provided by the taxi provider (driver profile and shift).
- The taxi provider has dedicated API which provide real-time data on driver's availability.
- The taxi provider determines trip prices.
- Payment processing is external and not under the scope of the application.

b) Many problems may occur when the following assumptions failed to hold:

- If trip prices are not determined by the taxi service provider, additional calculations must be performed to find fare fees after each trip.
- If drivers data are stored and handled within the context of the app, additional mechanism must be provided for drivers to update their information. In addition, driver's data storage must comply with the Canadian Data Protection Act.

2.6 Apportioning of Requirements

a) **Technological**: Some specific technilogical requirements (Database type, Integration with taxi provider, etc...) have to be delayed untill the implementation phase.

b) Non-function Requiremens such as UI,UX, or performance can be finetuned when the app is being implemented or after release.

3 Use Case Diagram

This section should provide a use case diagram for your application.

a) Each use case appearing in the diagram should be accompanied by a text description.

4 Functional Requirements

This section of the SRS should contain all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems. These requirements should include at a minimum a description of every input (stimulus) into the system, every output (response) from the system, and all functions performed by the system in response to an input or in support of an output.

You normally have two options for organizing your functional requirements:

1. Organize first by *business events*, then by *viewpoints*
2. Organize first by *viewpoints*, then by *business events*

Choose the one which makes the most sense.

For example, if you wish to organization by business events:

BE1. Business Event

VP1.1 Viewpoint

i. Requirement
ii. Requirement
iii. ...

VP1.2 Viewpoint

i. Requirement
ii. Requirement
iii. ...

VP1.3 ...

BE2. Business Event

VP2.1 Viewpoint

i. Requirement
ii. Requirement
iii. ...

VP2.2 Viewpoint

i. Requirement
ii. Requirement
iii. ...

VP2.3 ...

OR, if you wish to organization by viewpoints:

VP1. Viewpoint

BE1.1 Business Event

i. Requirement
ii. Requirement
iii. ...

BE1.2 Business Event

i. Requirement
ii. Requirement
iii. ...

BE1.3 ...

VP2. Viewpoint

BE2.1 Business Event

i. Requirement
ii. Requirement
iii. ...

BE2.2 Business Event

i. Requirement
ii. Requirement
iii. ...

BE2.3 ...

5 Non-Functional Requirements

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF1.

5.1.2 Style Requirements

LF1.

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH1.

5.2.2 Personalization and Internationalization Requirements

UH1.

5.2.3 Learning Requirements

UH1.

5.2.4 Understandability and Politeness Requirements

UH1.

5.2.5 Accessibility Requirements

UH1.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR1.

5.3.2 Safety-Critical Requirements

PR1.

5.3.3 Precision or Accuracy Requirements

PR1.

5.3.4 Reliability and Availability Requirements

PR1.

5.3.5 Robustness or Fault-Tolerance Requirements

PR1.

5.3.6 Capacity Requirements

PR1.

5.3.7 Scalability or Extensibility Requirements

PR1.

5.3.8 Longevity Requirements

PR1.

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

OE1.

5.4.2 Requirements for Interfacing with Adjacent Systems

OE1.

5.4.3 Productization Requirements

OE1.

5.4.4 Release Requirements

OE1.

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

MS1.

5.5.2 Supportability Requirements

MS1.

5.5.3 Adaptability Requirements

MS1.

5.6 Security Requirements

5.6.1 Access Requirements

SR1.

5.6.2 Integrity Requirements

SR1.

5.6.3 Privacy Requirements

SR1.

5.6.4 Audit Requirements

SR1.

5.6.5 Immunity Requirements

SR1.

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

CP1.

5.7.2 Political Requirements

CP1.

5.8 Legal Requirements

5.8.1 Compliance Requirements

LR1.

5.8.2 Standards Requirements

LR1.

A Division of Labour

Include a Division of Labour sheet which indicates the contributions of each team member. This sheet must be signed by all team members.

IMPORTANT NOTES

- Be sure to include all sections of the template in your document regardless whether you have something to write for each or not
  - If you do not have anything to write in a section, indicate this by the *N/A*, *void*, *none*, etc.
- Uniquely number each of your requirements for easy identification and cross-referencing
- Highlight terms that are defined in Section 1.3 (**Definitions, Acronyms, and Abbreviations**) with **bold**, *italic* or underline
- For Deliverable 1, please highlight, in some fashion, all (you may have more than one) creative and innovative features. Your creative and innovative features will generally be described in Section 2.2 (**Product Functions**), but it will depend on the type of creative or innovative features you are including.