Deliverable #1 Template : Software Requirement Specification (SRS)

SE 3A04: Software Design II – Large System Design

Tutorial Number: T03
Group Number: G8
Group Members: Hashim Bukhtiar, Jaden Moore, James Ariache, Olivia Reich, Omar Abdelhamid

- Group Member Name (as listed in Avenue)
- You do not need to use student #s or macid (keep those private).

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 <u>underline</u>
- 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.

1 Introduction

- Provide an overview of the document/SRS.

1.1 Purpose

- Specify the purpose of the SRS.
- Specify the intended audience for the SRS.

1.2 Scope

- Identify the software product(s) to be produced, and name each (e.g., Host DBMS, Report Generator, etc.)
- Explain what the software product(s) will do (and, if necessary, also state what they will not do).
- Describe the application of the software being specified, including relevant benefits, objectives, and goals.

1.3 Definitions, Acronyms, and Abbreviations

- Provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS.
- This should be in alphabetical order.

1.4 References

- Provide a complete list of all documents referenced elsewhere in the SRS.
- Identify each document by title, report number (if applicable), date, and publishing organization.
- Specify the sources from which the references can be obtained.
- Order this list in some sensible manner (alphabetical by author, or something else that makes more sense).

1.5 Overview

- Describe what the remainder of the document/SRS contains.
  (e.g. ”Section 2 discusses...Section 3...”)

2 Overall Product Description

- This section 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

- Put the product into perspective with other related products, i.e., context
- If the product is independent and totally self-contained, it should be stated here
- If the SRS defines a product that is a component of a larger system, then this subsection should relate the requirements of that larger system to the functionality of the software being developed. Identify interfaces between that larger system and the software to be developed.
- A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful

2.2 Product Functions

- Provide a *summary* of the major functions that the software will perform.
  - **Example**: An SRS for an accounting program may use this part to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those functions requires.
- Functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time
- Present the functions in a list format - each item should be one function, with a brief description of it
- Textual or graphical methods can be used to show the different functions and their relationships
  - Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables

2.3 User Characteristics

- Describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise
- Since there will be many users, you may wish to divide into different user types or personas

2.4 Constraints

- Provide a general description of any constraints that will limit the developer's options

2.5 Assumptions and Dependencies

- List any assumptions you made in interpreting what the software being developed is aiming to achieve
- List any other assumptions you made that, if it fails to hold, could require you to change the requirements
  - **Example**: An assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

2.6 Apportioning of Requirements

- Identify requirements that may be delayed until future versions of the system

3 Use Case Diagram

- Provide the use case diagram for the system being developed.
- You do not need to provide the textual description of any of the use cases here (these will be specified under ”Highlights of Functional Requirements”).

4 Highlights of Functional Requirements

- Specify all use cases (or other scenarios triggered by other events), organized by Business Event.
- For each Business Event, show the scenario from every Viewpoint. You should have the same set of Viewpoints across all Business Events. If a Viewpoint doesn't participate, write N/A so we know you considered it still. You can choose how to present this - keep in mind it should be easy to follow.
- At the end, combine them all into a Global Scenario.
- Your focus should be on what the system needs to do, not how to do it. Specify it in enough detail that it clearly specifies what needs to be accomplished, but not so detailed that you start programming or making design decisions.
- Keep the length of each use case (Global Scenario) manageable. If it's getting too long, split into sub-cases.
- You are *not* specifying a complete and consistent set of functional requirements here. (i.e. you are providing them in the form of use cases/global scenarios, not a refined list). For the purpose of this project, you do not need to reduce them to a list; the global scenarios format is all you need.
- Red text below is just to highlight where you need to insert a scenario - don't actually write it all in red.

Main Business Events: List out all the main business events you are presenting. If you sub-divided into smaller ones, you don't need to include the smaller ones in this list.

Viewpoints: List out all the viewpoints you will be considering.

Interpretation: Specify any liberties you took in interpreting business events, if necessary.

BE1. Create an Account #1
Preconditions: The user is not logged in, thus the log-in screen is displayed

VP1. User #1
Main Success Scenario

1. User opens the RideRecon app
2. The system responds by showing the user the log-in screen interface.
3. User selects create account
4. The system displays a screen to request information from the user relating to their account.
5. The user provides information to make the account and selects the option to confirm the account creation.
6. The system ensures the user's information is valid and grants them access to the main functionalities of the system.

Secondary Scenario

3i. User selects ”sign in as guest”
3i.1 User selects ”sign in as guest”

- **3i.2** System grants user access to RideRecon app in guest mode.

- **5i.** User cancels the operation
  - **5i.1** User cancels the operation
  - **5i.2** System returns them to the log-in menu
- **6i.** The system detects the user's information is invalid
  - **6i.1** The system detects the user's information is invalid
  - **6i.2** The system returns an message to the user stating the information is invalid.
  - **6i.3** The user acknowledges the message.
  - **6i.4** The system returns the user to the information request screen in step **5.**
- **6ii.** An error occurs and the system is unable to validate the user's information.
  - **6ii.1** An error occurs and the system is unable to validate the user's information.
  - **6ii.2** The system returns an message to the user stating an error has occurred.
  - **6ii.3** The user acknowledges the error.
  - **6ii.4** The system returns the user to the information request screen in step **5.**

VP2. Customer Support #2

- **5i.** User selects get support option to speak with customer support regarding signup
  - **5i.1** User selects get support option to speak with customer support regarding signup
  - **5i.2** System redirects the user to a location outside of the app where the user can get in touch with our contractor's customer support line.
- **6i.** An error occurs and the user's information cannot be validated.
  - **6i.1** An error occurs and the user's information cannot be validated.
  - **6i.2** The system sends a message to the user asking if they want to submit a crash report for analysis.
  - **6i.3** The user selects yes or no.
  - **6i.4** If yes is selected, the system sends a crash report of the error to the system maintainers. If no is selected, nothing happens.
  - **6i.5** The system returns the user to the information request screen in **step 5.**

VP3. Marketing #3
N/A

VP4. Accounting #4
N/A

VP5. Dealership #5
N/A

Global Scenario:
Main Success Scenario

1. User opens the RideRecon app

2. The system responds by showing the user the log-in screen interface.

3. User selects create account

4. The system displays a screen to request information from the user relating to their account.

5. The user provides information to make the account and selects the option to confirm the account creation.

6. The system ensures the user's information is valid and grants them access to the main functionalities of the system.

Secondary Scenario

3i. User selects "sign in as guest"

3i.1 User selects ”sign in as guest”

3i.2 User gains access to the system in guest mode.

5i. User cancels the operation

5i.1 User cancels the operation

5i.2 System returns them to the log-in menu

5ii. User selects get support option to speak with customer support regarding signup

5ii.1 User selects get support option to speak with customer support regarding signup

5ii.2 System redirects the user to a location outside of the app where the user can get in touch with our contractor's customer support line.

6i. The system detects the user's information is invalid

6i.1 The system detects the user's information is invalid

6i.2 The system returns an message to the user stating the information is invalid.

6i.3 The user acknowledges the message.

6i.4 The system returns the user to the information request screen in step **5.**

6ii. An error occurs and the system is unable to validate the user's information.

6ii.1 An error occurs and the system is unable to validate the user's information.

6ii.2 The system returns an message to the user stating an error has occurred. Alongside the message, the system prompts the user to submit a crash report to the system maintainers for analysis.

6ii.4 The user selects either yes or no to submitting a crash report.

6ii.5 If the user selected yes, a crash report of the error is submitted to the system maintainers. If no is selected, nothing happens.

6ii.6 The system returns the user to the information request screen in step **5.**

BE2. Business Event Name #2

VP1. Viewpoint Name #1
Insert Scenario Here

VP2. Viewpoint Name #2
Insert Scenario Here

Global Scenario:
Insert Scenario Here

BE3. Business Event Name #3

VP1. Viewpoint Name #1

VP2. Viewpoint Name #2

Global Scenario:
Insert Scenario Here

BE4. Compare Expert Answers #4

VP1. User #1
Main Success Scenario

1. User provides the information of a car for the system to process.

2. System feeds input to all of its experts

3. Experts provide the system with their answer on information about the car.

4. The finalizer recognizes the experts agree on what kind of car the user inputs.

5. The finalizer processes the data from the experts for output.

Secondary Scenario

4i. The finalizer recognizes that not all of the experts agree on what kind of car has been input into the system.

- **4i.1** The finalizer recognizes that not all of the experts agree on what kind of car has been input into the system.
- **4i.2** Using conflict resolution methods, the finalizer will determine which experts to accept information from.
- **4i.3** If a certain threshold of certainty is reached by the finalizer, the finalizer will use information from the selected experts in output processing. If not, the system will attempt to provide information to the experts that will refine their outputs and the system returns to step **3.** in the process.
- **4i.4** After a pre-defined maximum amount of attempts to refine the solution is reached, the system will notify the user that it was unable to find the car that was input.
- **4i.5** The user acknowledges the message.
- **4i.6** The system returns itself to the state where the user can provide input for identifying a car.

VP2. Customer Support #2

4i. Should the experts and finalizer fail to determine what car was input by the user, the system will prompt the user with the option to send feedback.

- **4i.1** Should the experts and finalizer fail to determine what car was input by the user, the system will prompt the user with the option to send feedback.
- **4i.2** The user can select yes or no to this prompt.
- **4i.3** If the user selects no, nothing happens and the system is returned to a state where the user can provide input for identifying a car again. If the user selects yes, the system will prompt them for their feedback and any other important information.
- **4i.4** The user inputs their feedback.
- **4i.5** The system thanks the user for their feedback and returns them to a state where they can provide input for identifying a car again.

VP3. Marketing
N/A

VP4. Accounting
N/A

VP5. Dealership
N/A

Global Scenario:
Main Success Scenario

1. User provides the information of a car for the system to process.
2. System feeds input to all of its experts
3. Experts provide the system with their answer on information about the car.
4. The finalizer recognizes the experts agree on what kind of car the user inputs.
5. The finalizer processes the data from the experts for output.

Secondary Scenario

4i. The finalizer recognizes that not all of the experts agree on what kind of car has been input into the system.

- **4i.1** The finalizer recognizes that not all of the experts agree on what kind of car has been input into the system.

4i.2 Using conflict resolution methods, the finalizer will determine which experts to accept information from.

4i.3 If a certain threshold of certainty is reached by the finalizer, the finalizer will use information from the selected experts in output processing. If not, the system will attempt to provide information to the experts that will refine their outputs and the system returns to step **3.** in the process.

4i.4 After a pre-defined maximum amount of attempts to refine the solution is reached, the system will notify the user that it was unable to find the car that was input. Alongside this message will be an option to send feedback to the system maintainers about the experience.

4i.5 The user acknowledges the message and chooses not to give feedback.

4i.5.i.1 The user decides to give feedback.

4i.5.i.2 The system will prompt the user for their feedback and any other important information.

4i.5.i.3 The user inputs their feedback.

4i.5.i.4 The system thanks them for their response.

4i.6 The system returns itself to the state where the user can provide input for identifying a car.

5 Non-Functional Requirements

- For each non-functional requirement, provide a justification/rationale for it.
  Example:
  SC1. *The device should not explode in a customer's pocket.*
  Rationale: Other companies have had issues with the batteries they used in their phones randomly exploding [insert citation]. This causes a safety issue, as the phone is often carried in a person's hand or pocket.

- If you need to make a guess because you couldn't really talk to stakeholders, you can say ”We imagined stakeholders would want...because...”

- Each requirement should have a unique label/number for it.

- In the list below, if a particular section doesn't apply, just write N/A so we know you considered it.

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF-A1. The application must use company colours that align with the ones used in their branding.
Rationale: The user should feel as if the application is closely tied to the company that contracted us and their mission statements.

LF-A2. The application should not use harsh and overly vibrant colours
Rationale: If the application is hard on the eyes to use, the user will have a worse experience overall and be adverse to using the application.

LF-A3. The application should be consistent in how it associates colours with certain buttons or actions.
Rationale: When a user goes to do actions across the application, the colour of buttons relating to certain prompts like ”yes” or ”no” should remain consistent across all of those prompts. For example, if a ”yes” button uses the colour green, then any other time ”yes” comes up it should also be green.

LF-A4. Users should be able to quickly distinguish what is and what is not a button.
Rationale: The user should be able to quickly understand the options available to them without needing a lot of effort or research.

5.1.2 Style Requirements

LF-S1. The system must scale its UI so that it can fit to the size of the screen device.
Rationale: Device screens come in all shapes and sizes, thus the application should be able accommodate all kinds of screen sizes.

LF-S2. Additional options or menus should be hidden under one to a small handful of dropdown menus.
Rationale: The user's screen should not be cluttered with an abundance of buttons that makes the application hard to navigate.

LF-S3. The system should adhere to general practices for making applications for android devices
Rationale: Since the system is made for android devices, we need to follow general practices for making android applications to

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH-EOU1. The system should only require a maximum of two inputs from the user to produce an output.
Rationale: The user should not need to go through an extensive process to identify a car. They should be able to take a picture, type in a model, etc. and receive the results they are looking for.

UH-EOU2. The system should be usable with only one hand.
Rationale: There are many times in a user's daily life where only one hand is available to them. Thus, making the app usable with only one hand means users would be able to use the app even in those moments.

5.2.2 Personalization and Internationalization Requirements

UH-PI1. The system should have a variety of language options to choose from.
Rationale: With a variety of users comes a variety of languages that are spoken. Thus, to improve the user's experience, we should give them a wide range of languages to choose from do they can use the app with the language they are most comfortable with.

5.2.3 Learning Requirements

UH-L1. it is clear how to access the camera through the a

5.2.4 Understandability and Politeness Requirements

UH-UP1. Any icons used in the system without explanation should be universally understood by users or easy to learn.
Rationale: An icon that can be quickly understood or universally known will make it easier for users to understand what our system does.

UH-UP2. The system should be usable on a lower device brightness.
Rationale: Users will not always have their device at full brightness, so if the device becomes harder to use at these low brightness settings the user will find the system more tedious to work with.

5.2.5 Accessibility Requirements

UH-A1. The system should be able to accommodate users with colour blindness.
Rationale: All manners of users will be using this system, including those with colour blindness. Thus, it would be unethical if they were unable to use the app based on our design decisions.

UH-A2. The system should be easy to use by both right and left handed people.
Rationale: A user should have an easy time using the system with either their left or right hand since the user may have a hand that they prefer to use. This requirement also works in conjunction with requirement UH-EOU2, the system should be usable with only one hand since the one hand being used should account for either the left hand or the right hand.

UH-A3. The system should be compatible with accessibility screen readers.
Rationale: Users with vision or comprehension disabilities might use screen readers to better navigate the devices and applications that they are engaging with. Thus, to make the system accessible to those users it would be necessary to make the system friendly to common screen readers.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR-SL1.

5.3.2 Safety-Critical Requirements

PR-SC1.

5.3.3 Precision or Accuracy Requirements

PR-PA1.

5.3.4 Reliability and Availability Requirements

PR-RA1.

5.3.5 Robustness or Fault-Tolerance Requirements

PR-RFT1.

5.3.6 Capacity Requirements

PR-C1.

5.3.7 Scalability or Extensibility Requirements

PR-SE1.

5.3.8 Longevity Requirements

PR-L1.

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

OE-EPE1.

5.4.2 Requirements for Interfacing with Adjacent Systems

OE-IA1.

5.4.3 Productization Requirements

OE-P1.

5.4.4 Release Requirements

OE-R1.

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

MS-M1.

5.5.2 Supportability Requirements

MS-S1.

5.5.3 Adaptability Requirements

MS-A1.

5.6 Security Requirements

5.6.1 Access Requirements

SR-AC1.

5.6.2 Integrity Requirements

SR-INT1.

5.6.3 Privacy Requirements

SR-P1.

5.6.4 Audit Requirements

SR-AU1.

5.6.5 Immunity Requirements

SR-IM1.

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

CP-C1.

5.7.2 Political Requirements

CP-P1.

5.8 Legal Requirements

5.8.1 Compliance Requirements

LR-COMP1.

5.8.2 Standards Requirements

LR-STD1.

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.

Hashim Bukhtiar	Jaden Moore	James Ariache	Olivia Reich	Omar Abdelhamid
1.1, 1.2, 1.3, 1.4, 1.5 Section 3 BE1 in Section 4	5.3, 5.4, 5.5, 5.8 BE2 in Section 4	5.1, 5.2, 5.6, 5.7 BE3 in Section 4	2.1, 2.2, 2.3 BE4 in Section 4	2.4, 2.5, 2.6 BE5 in Section 4
		James Ariache		SIGNATURE

Table 1: Division of Labour