Software Requirements Specification

For the Software Engineering course

Version 1.2 approved

Prepared by Group “C”
Reviewed by D.Mustafa Assaf

An-Najah National University

27th November 2021

Team member

Taha Hmmoouz, 11925452
Mawada Sweis, 11924962
Abood Ghalayini, 11926807
Abed Kharoubi, 11925569
Abood Jallad, 11926600

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Group C	27.11.2021	SRS version	1.2
Group C	21.11.2021	Primary actors and use cases, use case diagram.	1.1
Group C	22.10.2021	Vision and Scope	1.0

1. Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1. Purpose

The purpose of this document is to give a detailed description of the requirements for the system. It will illustrate the purpose and complete declaration for the development of the system.

This document is intended for anyone included in the stakeholder, development or administration of this application. Which includes software developers, team managers, clients and users.

1.2. Scope

The application facilitates the provision of transportation in times of severe crisis, hears the opinion of the citizen, tracks his request and guarantees his right.
The application is to facilitate the process of requesting a trip within cities, between cities, and to and from the bridge.
Through the application, you can request a trip by specifying the property of the trip, displaying the driver's biography, providing feedback after each request, and submitting an application to join the company's list of drivers.
One of the features of the application is that it tells you how long it takes to provide the car to you, and the most important thing is to display the path of the car until it reaches you, and it deducts the price of the trip directly from the application.

1.3. Definitions, acronyms, and abbreviations

Table 1 - Definitions

Term	Definition
SRS	Software Requirements Specification.
User	Anyone who uses the application.
Admin	The system administrator whose given specific permission for managing and controlling the system.
HR	Human Resources.
Stakeholder	Any person who has interaction with the system who not a developer.

1.4. References

1.4.1. IEEE. IEEE Std 830–1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
1.4.2. Vision and scope document.
1.4.3. Use cases document.

1.5. Overview

The remainder of this document includes two chapters. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system assumptions about the product.
The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

2. Overall Description

This section will give an overview of the system, It will mention its basic functionality. It will also describe what type of stakeholders will use the system and what functionality is available for each type. At last, the assumptions and dependencies for the system will be presented.

2.1. Product Perspective

This subsection of the SRS should put the product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the SRS defines a product that is a component of a larger system, as frequently occurs, then this subsection should relate the requirements of that larger system to the functionality of the software and should identify interfaces between that system and the software.
A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.

This subsection should also describe how the software operates inside various constraints. For example,
these constraints could include
a) System interfaces;
b) User interfaces;
c) Hardware interfaces;
d) Software interfaces;
e) Communications interfaces;
f) Memory;
g) Operations;
h) Site adaptation requirements.

2.2. Product Functions

We divided our function due to the access.

This subsection of the SRS should provide a summary of the major functions that the software will perform.
For 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.

Sometimes the function summary that is necessary for this part can be taken directly from the section of the higher-level specification (if one exists) that allocates particular functions to the software product. Note that for the sake of clarity
a) The 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.
b) 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 Classes and Characteristics

1. Admin
   He is an employee of the company who has certain powers over the application, he should have previous knowledge about the usage of the application and the internet.
2. HR
   Should have previous knowledge about the usage of the application and the internet. He has good communication and interviewing skills.
3. Passenger
   Passengers should be able to use the system, have a connection with the internet. and are at least 15 years old.
4. Driver
   Passengers should be able to use the system, have a connection with the internet. Should have driver licenses and already submitted a request to be a driver (candidate).
5. Candidates
   A user that submitted a request to be a driver. Should have drivers licenses.

Primary Actor	Use Cases
Passenger	1. Make trip requests by filling out the form. 2. Confirm the request and pay the cost for the request. 3. Tracking the trip request. 4. Cancel the trip request. 5. View the last trip request.

	6. Show the driver profile. 7. Feedback on the last trip.
Driver	1. Accept specific trip requests. 2. View the list of trip requests. 3. Set the status of the driver. 4. Cancel trip request. 5. Send notification if he arrived. 6. Feedback on the last trip.
Candidate	1. Send a Job Application by filling out the form. 2. View interview appointment. 3. Cancel the Job Application.
HR	1. Set candidate as Driver. 2. Remove or Freeze specific users. 3. Add new users.
Admin	1. Set the fixed cost. 2. View drivers account statement. 3. Remove or Freeze specific users. 4. Add new users.

2.4. Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not designed constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For 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.

3. Specific requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

3.1. Functional requirements

This section includes the requirements that specify all the fundamental actions of the software system.

Split the functional requirement into user class, each user class includes its functional requirements

3.2. Use case

3.2.1. Use case diagram

3.2.2. Use case documentation

3.3. Other Nonfunctional Requirements

There are a number of attributes of software that can serve as requirements. It is important that required attributes be specified so that their achievement can be objectively verified. Subclauses 5.3.6.1 through 5.3.6.5 provide a partial list of examples.

3.4. System models