Software Requirements Specification

For the Software Engineering course

Version 1.1 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	Initial SRS version	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

This subsection should
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

Table 1 - Definitions

Term	**Definition**
SWE	Software Engineer.
SRS	Software Requirements Specification.
User	Anyone who uses the website.
Admin/ Administrator	The system administrator whose given specific permission for managing and controlling the system.
Stakeholder	Any person who has interaction with the system who not a developer.
Database	Collection of all the information monitored by this system.

1.4 References

1.4.1 IEEE. IEEE Std 830–1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

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

This subsection of the SRS should describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise. It should not be used to state-specific requirements, but rather should provide the reasons why certain specific requirements are later specified in Section 3 of the SRS.

2.4 Assumptions and Dependencies

External hardware

2.4.1 · Employee workstations:

these are the personal computers used by the workers to carry their daily tasks.

2.4.2· User client:

refers to all the devices used in communication, such as PCs and smartphones.

2.4.3· Local Area Network:

this is the internal local area network that is used for communication among WeShift employees.

2.4.4· Internet:

the global network applied in communication among users of the app, employees, drivers, and the WeShift platform.

2.4.5· Other assumptions :

i. It is assumed that all the riders are requesting for the ride while at the cities where WeShift operates.

ii. Another assumption is that the riders will pay the fare through option they chose before placing a ride.

iii. It is also assumed that drivers are all-over the city so no customer will miss their ride.

iv. All the users of the application must have a user account in order to access these services.

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

1. The Application must have a logo at the start of application.

2. Every online booking needs to be associated with an account

3. One account cannot be associated with multiple user.

4. Search results should enable users to find the most recent and relevant booking rides options.

5. System should enable users to book / pay for their rides only in cash or credit card.

6. The application shall keep track of all processes and changes happening to the data between login and logout times of the users.

7. The application should enable the users to logout after using the application when the user clicks on the logout button.

3.2 Use case

3.2.1 Use case diagram

3.2.2 Use case documentation

3.3 Other Nonfunctional Requirements

3.3.1-Usability:
users should be able to book a ride in less than a minute, access personal accounts, get their payments processed in less than a minute. The app should be transparent about the way it connects clients and drivers, record past activities, and provide detailed reports on all financial transactions.

3.3.2-Legal requirements:
WeShift should comply with local transportation and employment laws. Additionally, personalization and data collection.

3.3.3-Reliability:
WeShift should provide fast and error-free performance in low-bandwidth zones. All payments, driver requests, and updates should go through smoothly.

3.3.4-Security:
how is the data on user location processed and stored? How is the financial information handled by the application? How is personal account information saved and encrypted?

3.4 System models