Twiss, a car sharing application

Supplementary Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 18.03.2018 | <1.0> | First version of the document | Andrei Gog |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Non-functional Requirements 4

2.1 Availability 4

2.2 Performance 4

2.3 Security 4

2.4 Testability 4

2.5 Usability 4

3. Design Constraints 4

Supplementary Specification

# Introduction

The introduction of the **Supplementary Specification** provides an overview of the entire document.

The **Supplementary Specification** captures the system requirements that are not readily captured in the use cases of the use-case model. Such requirements include:

Legal and regulatory requirements, including application standards.

Quality attributes of the system to be built, including usability, reliability, performance, and supportability requirements.

Other requirements such as operating systems and environments, compatibility requirements, and design constraints.

# Non-functional Requirements

## Availability

Twiss should be available 24 hours a day, including night rides, because even if by night the roads are almost empty, Twiss also provides a way of saving gas money and this could be very useful for most of the users.

## Performance

The application is designed for users to easily interact with other users and should respond and behave well even in busy environment with many users requesting data at the same time. Also there is a data base provided for storing only useful information.

## Security

Since Twiss is using personal information and payment methods are involved, each user will have a personal password and also he will be noticed if unusual activity is going on his account.

## Testability

The property of testability involves two components: the effort and the effectiveness of software tests that depends on numerous factors:

* Properties of the software requirements
* Properties of the software itself (such as size, complexity and testability)
* Properties of the test methods used
* Properties of the development- and testing processes
* Qualification and motivation of the persons involved in the test process

Besides the user interface, Twiss will go through automated testing methods.

## Usability

The application should perform with no problems on most operating systems, but it has been implemented on Windows 10. As a future development, Twiss will also be integrated as a web application or a mobile application on Android or iOS..and as a final stage, on all the platforms available out there.

# Design Constraints

Twiss only available language will be English. It will be compatible with the last Java runtime environment. There will be no hardware constraints because the functionality of the application doesn’t require computing power or too much storage space.