**Software Requirement Specification**

**For**

**Travelie**

**Group Members:**

**Muhammad Usman Abbasi (2014262)  
Syed Saadat Hussain Rizvi (2014370)**

**Raheel Jameel (2014294)**

**Revision History:**

|  |  |  |
| --- | --- | --- |
| ***Revision History*** | ***Date*** | ***Comments*** |
| 1.00 | 11-Oct-2016 | - |
| - | - | - |
| - | - | - |

**Document Approval:**

The following document has been accepted and approved by the following:

|  |  |  |
| --- | --- | --- |
| ***Signature*** | ***Date*** | ***Name*** |
|  |  |  |
|  |  |  |
|  |  |  |

List of Contents

[1 INTRODUCTION 5](#_Toc402274572)

[1.1. PURPOSE 5](#_Toc402274573)

[1.2. PRODUCT SCOPE 5](#_Toc402274574)

[2 OVERVIEW 5](#_Toc402274575)

[2.1 THE OVERALL DESCRIPTION 5](#_Toc402274576)

[2.2 PRODUCT PERSPECTIVE 5](#_Toc402274577)

[2.2. PRODUCT FUNCTIONS 5](#_Toc402274578)

[2.3. USER CHARACTERISTICS 5](#_Toc402274579)

[2.3. CONSTRAINTS 5](#_Toc402274580)

[2.4. ASSUMPTIONS AND DEPENDENCIES 5](#_Toc402274581)

[3 USER REQUIREMENTS 6](#_Toc402274582)

[3.1 External Interface Requirements 6](#_Toc402274583)

[3.1.1 User Interfaces 6](#_Toc402274584)

[3.1.2 Hardware Interfaces 6](#_Toc402274585)

[3.1.3 Software Interfaces 6](#_Toc402274586)

[3.1.4 Communication Interfaces 6](#_Toc402274587)

[4 Functional Requirements 6](#_Toc402274588)

[4.1 Functional Requirements with Traceability information 6](#_Toc402274589)

[5 Nonfunctional Requirements & Software System Attributes 7](#_Toc402274590)

[5.1 Performance Requirements 7](#_Toc402274591)

[6 USE CASES: (Diagrams) 7](#_Toc402274592)

List of Figures

[Figure 1 Overall Description](#_Toc381361180) 5

[Figure 2 Guest Use Case](#_Toc381361181) 9

[Figure 3 Staff Use case](#_Toc381361182) 9

Figure 4 Overall Use Case........................................................................................................................10

Figure 5 UML Diagram.............................................................................................................................11

1. Introduction

***1.1 Purpose***

This document is the Software Requirement Specification for **Travelie*.*** The ultimate goal of this is to develop an easy and interactive web-app to provide travel solutions for the Students and Staff of GIKI. This documentation will cover the whole proposed system.

***1.2 Project Scope***

Our web app, Travelie, will allow consumers to submit a trip request for immediate or later-scheduled local van pickups from GIK Institute to almost anywhere in Pakistan. Our main consumers/users are students and staff of GIK Institute.

When a user will submit a request along with desired time and place, the request will be displayed on our web app till the departure time of van. Meanwhile, if some other student also wishes to travel at that time can also reserve a seat in that van through our web app. As more and more people will reserve a seat in one specified van, the travel cost per head will be automatically updated. Other such features will also be added incrementally.

Estimated travel time, fare per head, driver details, vehicle type and license number will also be shown along any request.

***1.3 Intended Audience and Document Overview***

This document is intended for Team Travelie as well as Faculty, students, and teaching assistants associated with CS325. This document will have information that may be deemed pertinent for each member of the intended audience, but it is likely that certain sections will be of more interest to each type of audience member. Those interested in the functionality of the software should read section 2 (Overall Description) in order to learn more about the capabilities of the software. Those interested in the internal workings of the software should read section 3 (Specific Requirements). Finally, those interested in the development process should read through the entirety of this document.

***1.4 Definitions, Acronyms and Abbreviations***

* API – Application Programming Interface
* Web-app – Web application
* CS – Computer Science
* JRE – Java Runtime Environment
* JDK –Java Development Kit
* DPM – Distributed Power Management
* DRS – Distributed Resource Scheduler
* GUI – Graphical User Interface
* RAM – Random Access Memory
* SRS – Software Requirements Specifications

***1.5 References and Acknowledgments***

Design, Implementation, and Lessons Learned from the book: SOFTWARE ENGINEERING (Ninth Edition) by Ian Sommerville.

2. Overall Description

***2.1 Product Perspective***



***2.2. Product Functions***

***Main features of the system are listed below.***

* Online Ticket Reservation.
* Travelling-class.
* Countdown timer till next departure
* Dynamic ticket price
* Customer Registration
* Guest Booking
* Phone verification

***2.3 User Classes and Characteristics***

***Guest Users***

Standard users are guest users. The user who is to reserve a ticket would find the appropriate bus (required time, travel-class and location) at the reservation section and reserve a seat. If the user doesn’t find the required bus, he can post a new bus request and then wait till the requirement of minimum travelers is fulfilled. The user request is then phone verified and the booking process completes.

***Administrator Users***

Administrator users are the Travelie management and staff. They will have all the privileges and can change anything on the website. They are to update the information. For example there is change in ticket prices or a bus cancellation or time delay; it would be the staff that is to update the information.

***2.4 Development Environment***

### We will be using Github for our repository. The code will be developed on our personal computers and deployed on the remote server hosted at [HostGator](http://www.hostgator.com/).

.

***2.5. Design and Implementation Constraints***

* The design is only a small scale prototype of the large scale systems that can be built using it
* Some features of the design later direct to manual functions such as getting to the bus location and asking the driver to wait if the customer is late.

***2.6 User Documentation***

A user manual explaining what the functionality and usage of the visualization software will be required. Additionally, the team developing the software would potentially need to be available in case of questions or problems with the software once it starts being used by other users/administrators. This will make it easier for other people to understand and continue development on our product if needed.

***2.7 Assumptions and Dependencies***

We're assuming there are only two types of travel-class. Additionally, Customers and bus drivers will be on-time.

4. Non-Functional Requirements

**Hardware Requirement:**

* Disk space: 100 MB
* Memory: 512 MB

**Software Requirements:**

* Apache Tomcat, but this website would work fine under any web server that supports Java web applications, such as NGINX. The general advice is to use the newest version possible of your chosen web server.
* Latest version of JRE installed
* A working databases server: MYSQL

**Operating System:**

* It can work on any Operating System.

5. Nonfunctional Requirements

***5.1 Performance Requirements***

Performance is key requirement in our system. The website will be updated dynamically as soon as the user completes the booking process. Concurrent booking should not interrupt any process.

***5.2 Security Requirements***

Viewing the other guests’ information is an issue of confidentiality. This should be allowed only to the authorized staff. And also the new updates must be sent from an authorized staff in charge. Therefore the filtering part of our system should be made in order to affect those constraints. By considering these issues, we will categorize the accounts.

6. Use Case Diagrams

### *Guest Use Case*

**Diagram:**



**Brief Description:**

Guests access the Hotel Website to reserve rooms or to get information regarding Hotel Services.

**Initial Step-By-Step Description:**

It is assumed that the Guest has a computer and an internet connection.

1. Guest opens his web browser and goes to the Hotel Website.
2. Guest goes through the tabs to find the one he wants (say Room Reservation).
3. Guest open the Room Reservation Tab and Room Reservation page opens.
4. Guest enters his details and his desired room, date of arrival etc.

Room will be reserved for him.

### *Staff Use Case*

**Diagram:**

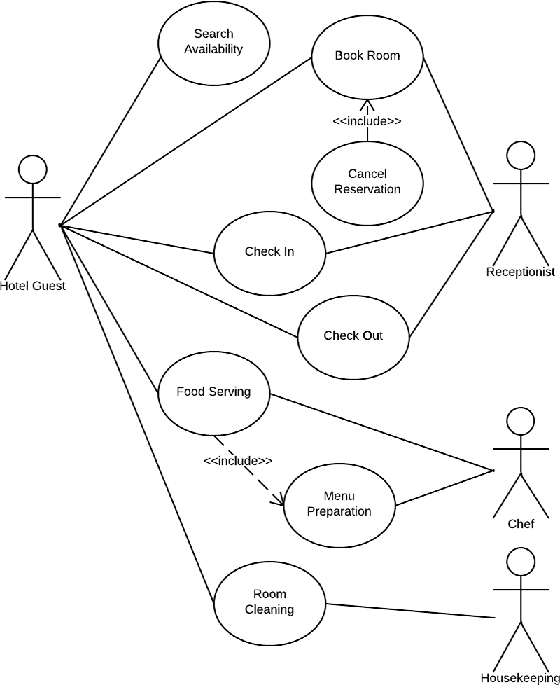


**Brief Description:**

Staff accesses the Hotel Website to for reviewing or updating the information.

**Initial Step-By-Step Description:**

1. Staff can check credibility of the guests’ record.
2. Staff makes sure guest cannot access unprivileged information.
3. Staff can launch discussion forums and news.
4. Staff can change or update any kind of information.
5. Staff can keep track of changes made in website.



7. UML Diagram

