Software Requirements Specification

for

Sale Opportunity

**SOP Systems**

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this project is to provide the company a organizational ability and efficient way to manage the company transactions, by developing a user-friendly booking management system.

## Document Conventions

In this document topics are listed numerically, every new topic start with big bold font headline and every subtopic start with smaller bold font headline.

## Intended Audience and Reading Suggestions

The intended audience of this document are the client and the development team, the project will be supervised by the project guider.

## Product Scope

The software will provide an adjusted management system to the users of the company,

the Travel agents will have efficient and organized way to manage and access the company data base, And the clients will have an easy and wide functionality user experience.

## References



*SRS template.*

# Overall Description

## Product Perspective

This product is a new system that will generate tracking data base based on the user existing data base of products, as the described below:

**clients**

**Travel agents**

**system**

**packages**

**transictions**

## Product Functions 2.2.1. Add a package

The travel agent will have the ability to add a package to the data base.

### 2.2.2. delete package

The travel agent will have the ability to remove a package from the data base.

### 2.2.3. Track orders

Tracking and sorting orders performed by customers

### 2.2.4 change order status

***2.2.4.1 by agent***

The travel agent will have the ability to change status of an order

***2.2.4.2 by client***

The client will have the ability to cancel an order.

### *2.2.4.* Order management

### The agent will have the ability to change the quantity of packages and watch in real time at the updates.

### *2.2.5.* Contact

### Absorption of all details for a contact request

### *2.2.6. menu*

### *2.2.6.1 agent menu*

### The agent will have a customized menu.

### *2.2.6.2 client menu*

### The client will have a customized menu

## User Classes and Characteristics

There will be two kinds of user's classes agent and client. The agent will have access to all the software features. client will have a more limited number of features. The additional features of the agent are: packages management.

## Operating Environment

The system runs on Microsoft Windows 7/8/10 versions 86x and 64 bits.

The system has to use a local database, the user will have the ability to manage packages.

## Design and Implementation Constraints

The software depends on high internal memory of the computer, be it RAM or HDD and its permutations. The developer must have access to the employee database. There is a limitation to the usage of its features by different users based on roles (agent/client). Software’s security must be assigned by the customer and customers own programming team as required by its own security protocols. The customers organization will be responsible for Integrating/maintaining the delivered software.

## User Documentation

The user will receive a manual detailing the functionality of the software.

## Assumptions and Dependencies

Our assumption that employee Database will be received from the HR department in ‘.txt’ file which the program will assign 'LOG IN' details of all company agents.

The file will be as follows: First name, Last name, Password, Role. (all on different lines) employees separated by '$' sign

The database includes details of access for each user such as Employee & Manager stated by Role.

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

## User Interfaces

The software will have basic menu bar with options, one for agents and one for the clients. - TBD

See the prototype presentation (style may differ from the final product).

## Hardware Interfaces

The application requires a monitor and working computer with OS installed.

## Software Interfaces

The application will have access to data base that was written by a third-party software program (in a .txt format).

The interface is as shown in the prototype.

## Communications Interfaces\*\*

Communication with human resources databases to transfer data into the software.

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

### Description and Priority

The clients will be able to leave a message to reserve a certain package.

The agents will be shown processing orders.

The agents will be able track and recommend packages for potential customers.

The client will be able to see the availability of the package and how many are left.

### Stimulus/Response Sequences

There will be an option to rate each package.

The software will show recommended packages for the client.

The client will be able to search packages by different categories (price, location , ect..).

Clients will be able to see previously searched packages.

### Functional Requirements

Clients who are yet to be registered will be asked to login to access certain actions.

Error messages will show for wrong details entered.

Packages without any quantity left will not be available to order.

Problems with server overload will be dealt with - TBD

Entering overlapping details will not be accepted and will give out an error message.

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>