

NOTE: This template is shareware downloaded from www.processimpact.com.

(NOTE: Reference: <http://newtours.demout.com>
Login: mercury
Password: mercury

Software Requirements Specification

for

Mercury Tours

Version 1.0 approved

Prepared by **Chaite Kaaji**
(Business Analyst)

Imagine Software, Inc
<The company who is making software for Mercury Tours>

February 10, 2010

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.....	2
1.4 Project Scope	2
1.5 References.....	2
2. Overall Description	3
2.1 Product Perspective	3
2.2 Product Features	3
2.3 User Classes and Characteristics	4
2.4 Operating Environment.....	4
2.5 Design and Implementation Constraints.....	4
2.6 User Documentation	5
2.7 Assumptions and Dependencies	5
3. System Features	5
3.1 System Feature 1.....	Error! Bookmark not defined.
3.2 System Feature 2 (and so on).....	Error! Bookmark not defined.
4. External Interface Requirements	5
4.1 User Interfaces	5
4.2 Hardware Interfaces.....	6
4.3 Software Interfaces	6
4.4 Communications Interfaces	6
5. Other Nonfunctional Requirements	6
5.1 Performance Requirements.....	6
5.2 Safety Requirements	6
5.3 Security Requirements	7
5.4 Software Quality Attributes.....	7
6. Other Requirements	7
Appendix A: Glossary	7
Appendix B: Analysis Models	7
Appendix C: Issues List	7

Revision History

Name	Date	Reason For Changes	Version
Adam Cohn	02/10/2010	Initial Draft	1.0
Chaite Kaaji	02/11/2010	Changed Login Box features	1.10
David Borogh	02/12/2010	Added “Links” on the home page	1.11

1. Introduction

1.1 Purpose

The purpose of this requirement document is to clearly specify the needs of the client for a travel and tour industry so that it will be easier to communicate among the developers, business analysts, test engineers and project managers. This document will be the major document that will be referred by all the team members involved in this project.

There is a need to align several groups and many people about what you're trying to accomplish, for whom, and in what envisioned manner. The requirements process provides the discussions and artifacts to enable the communication to the folks who need to provide pieces of the solution. It also introduces a common set of terminology and templates to facilitate the conversations and to make sure a necessary level of due diligence is happening before committing to a potentially major undertaking.

1.2 Document Conventions

The originator of the Requirements is usually the "business owner" or "customer" while the originator of the Specs is the technical team. There are different levels of Specs with varying degrees of abstraction and implementation details. The Specs can also contain Project information, such as which resources are required, major milestones and development schedule and cost. In other environments, the Product Spec is the sole document driven from the "business owner" to a deep level of detail and the development team signs off on it.

This requirement document is developed by the business analyst with the help of product manager but with possible help from others such as a product designer, or technical staff. Its purpose is to describe the product-level view of what a user could accomplish with it. It expands the features into more detail to define the entire solution. It identifies the users of the product, the activities they would want to perform, external systems connections, how well the product needs to perform, and what constraints are placed on it. It could also contain UI mockups, personas, use cases, process flows, data flows and any level of technical detail to describe the desired results. The key perspective of this document is that it describes the product from the user's point of view.

Depending upon the need, other documents may be developed by a member(s) of the technical team - program manager, technical lead, business analyst, architect or others. Its purpose is to describe the specific functionality the product will provide in response to user and external system interactions. It tells the developers (and testers) what capabilities they need to build and deliver from the system's perspective, and is in effect a translation of the user description into a technical description. It can contain further iteration on use cases, a list of Functional Requirements (responses to user/system actions) and Non-Functional Requirements (qualities and constraints) of the system. As a checkpoint, it also provides feedback on how the requested functionality was understood and defines a specific solution to be able to estimate the effort required to build it.

A separate Use Case will be developed to further specify the needs of the functionalities in detail.

1.3 Intended Audience and Reading Suggestions

The intended audience for this requirement document are the developers, project managers, business team from Mercury Tours, users, Test Engineers, document writers, technical designers and any other related teams what may come across.

All the contents in this requirement documents are intended for the ease of use and understanding of the application functionalities. We recommend you to read the functionalities core in detail and the changes made during the revision process.

This document is dynamic (will keep changing) since we do not expect the client's are the final.

1.4 Project Scope

Mercury Tours web application is our ability to deliver complex ticketing and booking solutions. With an already established product in Mercury Tours, we knew we had the platform that could be used to develop the web-based ticketing engine.

Working to a very tight timescale, an early requirement was to get secure and reliable online ticketing working from the Mercury Tours web site as quickly as possible. Working with the core Mercury Tours Ticketing engine, we got to grips with the complexities of coach timetables and mapped Mercury Tours's schedules into a business logic that would drive the online engine. With hundreds of journeys departing from locations across Italy and the South of England, our challenge was to create both a user friendly booking interface and an intuitive back office system.

After understanding the complexities of route scheduling and ensuring that the system could support an unlimited number of new locations, journey types, and routes to be created within the back office we also had to design the pricing and customer model. This involved developing a customized shopping cart solution to handle multiple pricing types, discounting, promotions and concessions.

The Mercury Tours project entailed at least three complete solutions integrated together: the customer view where journeys could be planned and tickets secured online, the administrator view where all aspects of the system could be controlled along with a detailed and flexible reporting system for business intelligence, and a Point of Sale solution where operators could upload ticket sales data or issue tickets to walk-in customers. Additional modules were also required to handle affiliate sales, to create dedicated micro-sites for airline partners.

1.5 References

We will provide a list of documents as we finalize this requirement documents.

2. Overall Description

2.1 Product Perspective

The concept of traveling for pleasure emerged in the latter half of the twentieth century, when society as a whole became more affluent, achievements in aviation made inexpensive air travel available to the masses, and the basic desire to spend vacations away from home combined to make travel, and particularly long-distance travel, a popular and widespread activity. Whether this transformation of the human mindset was merely the product of a people suddenly and inexplicably desirous for travel, or the result of effective marketing by both national governments and commercial interests is not clear, but whatever the root of the new desire to travel, it did not surface until the dawn of the jet age.

For Americans, travel by plane first became available in the 1930s, when airline service was first established, but many could not afford to fly, and perhaps even more felt no desire to fly, preferring instead to remain close to home. It was not until after World War II that Americans began traveling by plane to any appreciable extent, an activity facilitated by a dramatic increase in their discretionary income and the affordability of travel. But even then, not many traveled, at least not by plane, for it would be another 20 years until more than one in two Americans had ever flown in an airplane. Nevertheless, once Americans began to travel in the 1950s, tour packages designed and arranged by tour organizers appeared immediately, signaling the genesis of the modern tour operator industry at roughly the same time the over-all travel industry began to mature into a formidable economic force.

Today, travel and tours has become a need and further, organizing these through the web based application has taken a significant market in the world. Yet, there are a lot of obstacles, uneasy ways of navigating to applications and getting lost on the way to making a booking which confuses the users to their decisions.

Mercury Tours has realized these difficulties and has understood the needs of the users. Therefore, Mercury Tours decided to build an application that meets the needs of the users of this 21st century.

We expect this application to fulfill the need of the users.

2.2 Product Features

There will be following pages that would be on the web-based Mercury Application:

1. Home Page: This will be the landing page for any user. When the user types in the URL in the address bar, the users will land to this page. This page will contain the following links and functionalities
 - a. Header of the page: Will contain the hottest promotion as a banner
 - b. Left header corner: Will contain the logo of Mercury Tours
 - c. Left navigation menu: Will contain the following Menu:
 - i. Home: Clicking on this will remain on the home page.
 - ii. Flights: Clicking on this link will take to Flight details and reservation page.
 - iii. Hotels: Clicking on this will take to the Hotel Reservation page.
 - iv. Car Rentals: Clicking on this will take to the Car Rental Page.

- v. Cruises: Clicking on this will take to the Car Rental page.
 - vi. Destination: Clicking on this will take to the Destination page.
 - vii. Vacation: Clicking on this will take to the Vacation page.
2. Center of the Home Page: The center of the Home Page will contain the following:
 - a. Features Destination (the hottest destination in promotion)
 - b. Specials: Will give details of the running specials
 - c. Tour Tips: Will contain information on security and precautions
3. Right panel of the page: This will contain:
 - a. Find a Flight: This will have the user name and password. The first time user must create a user name and password to make reservations.
 - b. Destinations: The user will be able to find the destination of need.
 - c. Register: The user will be able to create his/her login information in order to make reservations.
 - d. Links: Will contain several links related to travel and tours.
4. Horizontal Navigation: The horizontal navigation will contain the following menu:
 - a. Sign-On: Clicking on this will the user to the Sign-on page for the registered user.
 - b. Register: This will take the user to create login name and password for this first time user.
 - c. Support: This link take the user to the technical support page for Mercury Tours web application support.
 - d. Contact: This link will take the user to the contact information for Mercury Tours.
5. Login: Once the users login, they will have a lot of functionalities, for example fill out the forms, select your hotels, flights and so on. The details will be discussed in the Use Case documents.

All the above functionalities will be described in details for separate Use Cases (or Design Documents).

2.3 User Classes and Characteristics

The product users will be divided in two categories:

1. Unregistered users: This type of user will not be able to make any reservations. They will just visit the sites for information only.
2. Registered Users: They will be able to make reservations, make request for changes, select their destinations and so on.

2.4 Operating Environment

This web-Mercury Tours application will run in all the available platforms including window 2000 and up including Windows 7 and MAC. This will smoothly run on different browsers including Internet Explorer, version 5 and above, safari, Firefox. However, it will not be using chrome since a lot of issues are still unsolved in chrome.

2.5 Design and Implementation Constraints

The following constraints are strictly followed during the development process:

- ❖ The developers will have no access from home. They must come to the work location. This is for security reasons. Therefore, there will be no remote access.

- ❖ All the requirements, Use Cases are not permitted to take home or email.

2.6 User Documentation

Tutorial: A tutorial will be developed available in PDF format for the users need.

Online Help: A chat application will be developed for online users help.

User Manual: A user manual will be developed for the partners of Mercury Tours.

2.7 Assumptions and Dependencies

Hardware and software installation: It is assumed that all the needed software and hardware be installed and are ready to do the jobs needed. The information are to be posted on the site are authentic, the data supplied to the project are clean and good.

Another assumption is, there is little change in the requirements.

<Note: Here, you will 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).>

3. System Features

All the system features described in 2.2 Product Features will be in detail in each individual Use Case. Each feature will have a separate Use Case with the details of the functionalities how they work.

4. External Interface Requirements

4.1 User Interfaces

There will be user friendly Graphical User Interface (GUI) to interact with the users. These will be discussed more in Uses Cases.

There will be screen shots in the Use Cases.

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

4.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

4.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

4.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

5. Other Nonfunctional Requirements

5.1 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.>

5.2 Safety Requirements

Web security test will be carried out using a third party tool.

<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.>

5.3 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.>

5.4 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.>

6. 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: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>