**Software Requirements**

**Specification**

**for**

**Resort Reservation System**

**Version 1.0 approved**

Prepared by:  
Reyes, Hannah Mae E.  
Lee, Kyle Vincent V.  
Belchez, Maica L.

BSIT-MI141

Asia Pacific College

October 2016

**Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.**

**Software Requirements Specification for <Project> Page ii**

# Table of Contents

[Table of Contents ii](#_Toc6260)

[Revision History ii](#_Toc6261)

[1. Introduction 1](#_Toc6262)

[1.1 Purpose 1](#_Toc6263)

[1.2 Document Conventions 1](#_Toc6264)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc6265)

[1.4 Product Scope 1](#_Toc6266)

[1.5 References 1](#_Toc6267)

[2. Overall Description 2](#_Toc6268)

[2.1 Product Perspective 2](#_Toc6269)

[2.2 Product Functions 2](#_Toc6270)

[2.3 User Classes and Characteristics 2](#_Toc6271)

[2.4 Operating Environment 2](#_Toc6272)

[2.5 Design and Implementation Constraints 2](#_Toc6273)

[2.6 User Documentation 2](#_Toc6274)

[2.7 Assumptions and Dependencies 3](#_Toc6275)

[3. External Interface Requirements 3](#_Toc6276)

[3.1 User Interfaces 3](#_Toc6277)

[3.2 Hardware Interfaces 3](#_Toc6278)

[3.3 Software Interfaces 3](#_Toc6279)

[3.4 Communications Interfaces 3](#_Toc6280)

[4. System Features 4](#_Toc6281)

[4.1 System Feature 1 4](#_Toc6282)

[4.2 System Feature 2 (and so on) 4](#_Toc6283)

[5. Other Nonfunctional Requirements 4](#_Toc6284)

[5.1 Performance Requirements 4](#_Toc6285)

[5.2 Safety Requirements 5](#_Toc6286)

[5.3 Security Requirements 5](#_Toc6287)

[5.4 Software Quality Attributes 5](#_Toc6288)

[5.5 Business Rules 5](#_Toc6289)

[6. Other Requirements 5](#_Toc6290)

[Appendix A: Glossary 5](#_Toc6291)

[Appendix B: Analysis Models 5](#_Toc6292)

[Appendix C: To Be Determined List 6](#_Toc6293)

# Revision History

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

# Introduction

## Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

## Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

## Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

## References

World Wide Web:

[1] “SRS Document for Hotel Management System”, [Online]. Available:  
<http://www.academia.edu/10313728/srs_document_for_hotel_management_system>  
[Accessed: October 22, 2016]  
[2] “E-Hotel Project”, [Online]. Available: <http://www.ehotailproject.eu.pn/doc/SRS%20-%20Full%20Document> [Accessed: October 22, 2016]  
[3] “Hotel Management System”, [Online]. Available: <http://www.oocities.org/swe626/HotelManagementSystemCorrectFinalSRS>  
[Accessed: October 22, 2016]  
[4] “Hotel Management System”, [Online]. Available: <http://www.docslide.us/education/srs-document-for-hotel-management-system.html> [Accessed: October 29, 2016]

# Overall Description

## Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

## Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

## User Classes and Characteristics

**2.3.1 User Classes**

There are three user levels in Resort Reservation System of Tarangban Falls & Calimbo Farm Resort.

1. Owner (Client)
2. Manager
3. Receptionist

**2.3.2 Characteristics of User Classes**

Owner:

The resort owner has the right of monitoring and authorization of all the tasks handle by the system. He can access every function performed by the system. The owner of the resort, as well as the system, can access to the administration panel that is considered the center of the system. As the head authorized person of the resort, the owner gets the capability to manage the other users including their user levels and privileges. Taking backups of the system and re-establishing system can also be done by the owner. Then, he will be capable of taking all the sorts of reports accessible in the system. As the owner the of the resort and the system, he has the power to set room rates as well. The owner has the sole right of deleting a staff member for the system database.

Manager:

The manager is in charge of managing assets accessible in resort reservation system. The manager also has a large portion of the privileges specified above with the exception of the things in regards to the payment handling. The explanation behind utilizing a manager is to reduce the work load done by the owner that can’t be allotted to the receptionist, as those tasks appear to be much dependable. The manager has the right to take every report accessible in the system yet here also with the exception of the reports related to financial stuff, and hotel income. Manager has various capacities that receptionist, user level has such as adding new employees, guests, inventories, and room types to the system. Also, modifying or removing them in the system.

Receptionist:

As a resort receptionist, his or her part will be to achieve the objectives of bookings and to guarantee that all customers are treated with a high standard of customer service. Progressively, receptionist role has minimal accessibility to the system capacities. Receptionist plays the boundary role of the system. He or She can perform limited functions such as registering new guest to the system, make reservations, sending e-mail notifications to customers for booking confirmation. Management of the resort will prefer to contract receptionist who have a decent standard of general education and perhaps in subjects such as English, Math, and IT.

## Operating Environment

Hardware and Software requirements

Hardware:

1. Operating System: Supports all known operating systems, such as Windows, Linux
2. Computer: 512MB+ RAM, monitor with minimum resolution of 1024x768, keyboard, and mouse
3. Hard Drive: should be in NTFS file-system formatted with minimum 10 GB of free space
4. Printer: need to be used to print reports and notes

Software:

1. Software is designed to run on any platform above Microsoft Windows 7
2. Microsoft .NET Framework 4.0 or above
3. Microsoft SQL Server Management Studio Express 2010

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

User manual, provide to the client, will give a clear idea in communicating with the system. It will be composed in a straightforward justifiable language covering the internal complexity of the system. A printed copy of the user manual will be delivered to the client with the delivery of the system.

## Assumptions and Dependencies

<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

<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.>

## Hardware Interfaces

Section 2.4 includes the requirements of the desktop computer where the system going to be installed. A specific computer must match with the above mentioned requirements in order to gain the maximum benefits from the system in an efficient manner.

Reservation alerts will be sent to the one of the member of hotel employee as an e-mail notification. So there is a need of broadband internet connection. Client should be able to keep a stable internet connection.

Also, a printer will be needed when printing bills and several reports.

## 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.>

## 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.>

# 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.> 4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<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

Performance requirements define sustainable response times for system functionality. Despite the fact that the system is created suiting for the least system performances, the performance of the system will highly depend on the performance of the hardware and software components of the installing computer. At the point when consider about the timing relationships of the system, the load time for user interface screens might take no longer than two seconds. It makes fast access to system functions. The log in details shall be verified within five seconds causes’ efficiency of the system. Returning query results within five seconds makes search function more accurate.

## Safety Requirements

There are several user levels in resort reservation system, access to the different subsystems will be ensured by a user log in screen that requires a username and password. This gives different perspectives and accessible functions of user levels through the system. Maintaining backups ensure the system database security. System can be restoring in any case of emergency.

## 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.>