**Software Requirements**

**Specification**

**for**

**SM Hotels: Service Request Management System**

**Version 1.0 approved**

**Prepared by Lurenne Tangi, Kenneth Abuel**

**School of Computing and Information Technology**

**October 8, 2017**

**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** |
| Lurenne Tangi | 10/8/2017 | Document Created, Added Purpose | 1.0 |
| Kenneth Abuel | 10/8/2017 | Added Document Conventions, Intended Audience, Product Scope, Operating Environment and Reference | 1.1 |
| Kenneth Abuel | 10/9/2017 | Minor Revisions. | 1.2 |
| Kenneth Abuel | 10/9/2017 | Minor revisions | 1.3 |
| Kenneth Abuel | 10/15/2017 | Minor Revisions | 1.4 |

# Introduction

## Purpose

The purpose of this SRS is to provide information to the people who will involve in the service request system. In the system, it will help people especially the customers or the Hotel Guest to have a better requesting system. With this, the department will also have a better service for the hotel guest. This will be the biggest part of the system.

## Document Conventions

In this document the font has already been set at “Arial” with a font size of 11, the titles and sub titles are in bold text, the team decided to follow the format of this document. Also we did some modifications in the SRS documentation, we edited the spacing into “1.5” in order to have a better format.

## Intended Audience and Reading Suggestions

The document is intended for the following readers:

* Client – In order to have a better understanding in the system.
* Users – In the document we provided a user manual so that the user won’t have any hard time using the system
* Developers – The document can be used as a reference for the future projects of the other developers

## Product Scope

Our product the SM Hotels Service Request Management System, is a system that handles and monitors the Service Request Tickets of the SM Hotels. The system also has a function that automatically generates a ticket when a hotel guest requested a service. The team’s goal is to provide a user-friendly interface system that will help the SM Hotels to provide a better service management.

## References

Know Service. (10/8/2017). Retrieved from https://www.knowcross.com/know-service/

The Most Overlooked Hotel Management Technology. (10/8/2017). Retrieved from https://blog.capterra.com/guest-request-management-the-most-overlooked-hotel-management-technology/

Hospitality Technology Matters. (10/8/2017). Retrieved from https://info.aliceapp.com/blog/guest-request-management-the-most-under-prioritized-hotel-management-technology

# Overall Description

## Product Perspective

Currently the SM Hotels does not have yet a Service Request or Guest Request Management System they currently using Microsoft Excel to record or handle the request tickets. The SM hotels decided that they needed a Service Request system in order to have a more efficient way of providing service to their hotel guests.

<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

The major function of the SM Hotels Service Request Management System:

* Automates the ticket generation
* List of available and non-available service employees
* A detailed view of ticket information and employee information
* Importing and Exporting of Data

## User Classes and Characteristics

* Department Supervisor / Employees – They are responsible in monitoring and handling of the service tickets
* Quality Assurance Supervisor – The supervisor is able to view the reports of the most common requested services.

## Operating Environment

The system will operate on the workstations at the SM Hotels, it will work on different kind of operating system. It will also work on low specification workstations; the software does not require a more advanced specification. The system will use MySQL as the database all data will be stored in it.

## 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

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## 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

<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.>

## 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

Here are the following features that has been developed throughout the project time.

## Automatically Generate Ticket

Description and Priority

When a hotel guest, that operator of the system creates a ticket and the system will automatically get the time when the ticket was created and it will automatically assign an employee who is currently available.

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:

## Employee List and Notification

Description and Priority

This feature provides a list of available and unavailable employees and a system notification that can notify the employees that they have a new service to render.

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:

# 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

There are no really concern about the safety requirements of the system as long as the user doesn’t accidentally delete any kind of data that the client needs.

## Security Requirements

Since we used Yii Framework for the development of the SM Hotels: SRMS, Yii’s security comes as standard, it is equipped with many security measures to help prevent the web application from attacks such as SQL Injection, Cross Site Scripting and many more.

## Software Quality Attributes

In the development of the system we used Yii, it is a high performance modern PHP framework best for developing both web applications and APIs. The system helps to ensure an extremely efficient, extensible and maintainable product. The team ensures that the system will be extremely optimized because the framework that we used was written in PHP5 that promotes clean, DRY design and encourages rapid development.

## 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

The other requirements that we needed was the proper terms that we needed to be labeled in our system that’s why we asked our client for proper naming or terms that the SM Hotels is using.

# 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.>