Hotel Management System Requirements Specification

Version 1.0

April 19, 2021

Table of Contents

1. Executive Summary

1.1 Project Overview

The tourism industry in Albania has experienced rapid growth due to an ever increasing number of foreign tourists seeking to explore the cultural and archaeological heritage, as well as indulge in the numerous breathtaking landscapes. Our goal is to provide the albanian hotel industry with a flexible solution for their needs, hence we are keen on constructing a robust Hotel Management System capable of rising to the challenges of the market. Our project is principally focused on providing a management system for hotel managers and employees with the goal of facilitating the management of their daily operations. Additionally, possible clients can book rooms, and see reviews of previous clients. The problems that this management system will solve are:

- The management of the room reservations by the receptionist.
- The management of the restaurant and beach by the service staff manager.
- The possibility of managing all the systems of the hotel by the managers.
- A user-friendly webpage displaying hotel services and the bookings for the clients (in this case, the clients are the tourists).

1.2 Purpose and Scope of this Specification

The purpose of our web application is to facilitate the way a hotel works in Albania. Although we are in the Information Age, many records are being stored manually in papers, which makes the data unsafe, and very difficult to process. There is not an actual functional digital management system to help them manage the staff updates, bills and reservations. Since the number of tourists in the summer in Albania is significantly high, many foreign tourists face the problem of not being able to book a hotel room in advance, and they have to go hotel by hotel till they find an empty room.

2. Product/Service Description

Nowadays many hotels in our country face many difficulties in managing the booking, rooms and staff. This is where HMS comes to help.

2.1 Product Context

This project will come to help for managing the important tasks of a hotel in Dhermi. It is not dependent on any other platform or system.

2.2 User Characteristics

There are four types of users - the manager, the receptionist, services staff, and clients - each with their own access level and privileges:

1. **Hotel Manager**

- Add, remove hotel employees
- Check, verify, and update employee information
- Access to hotel expenditures, profit, and financial balance
- Create and manage hotel inventory
- Public relation

2. **Restaurant Manager**

- Add, remove restaurant employees, such as: waiter
- Check, verify, and update restaurant employee information
- Access to hotel expenditures, profit, and financial balance
- Create and manage restaurant inventory

3. **Receptionist**

- Keep guest records
- Make clients' reservations
- Enter guests payment
- Check room availability status

4. **Client**

- Does not have access to management system
- Can book reservations
- Can check for availability of rooms or tables
- Can find information about the facilities of the hotel, location nearby tourist attractions
- In case of cancelling 20% penalty
- Can add a review

5. **Waiters**

- Enter bill records in the management system
- Keep track of his own tables

2.3 Assumptions

- It is assumed that each user has Internet access in order to have access to the web application
- It is assumed that each hotel employee is able to use the Internet and Hotel Management System effectively and efficiently

2.4 Constraints

- Every hotel employee that is a user of the system must have a phone or computer that can be connected to the Internet.

2.5 Dependencies

List dependencies that affect the requirements. Examples:

- This new product will require a daily download of data from X,
- Module X needs to be completed before this module can be built.

3. Requirements

- Describe all system requirements in enough detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements.
- Organize these requirements in a way that works best for your project. See Appendix DAppendix D, Organizing the Requirements for different ways to organize these requirements.
- Describe every input into the system, every output from the system, and every function performed by the system in response to an input or in support of an output. (Specify what functions are to be performed on what data to produce what results at what location for whom.)
- Each requirement should be numbered (or uniquely identifiable) and prioritized.
  See the sample requirements in Functional Requirements, and System Interface/Integration, as well as these example priority definitions:

  Priority Definitions

  The following definitions are intended as a guideline to prioritize requirements.

  - Priority 1 – The requirement is a “must have” as outlined by policy/law
  - Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits
  - Priority 3 – The requirement is a “nice to have” which may include new functionality

  It may be helpful to phrase the requirement in terms of its priority, e.g., "The value of the employee status sent to DIS **must be** either A or I" or "It **would be nice** if the application warned the user that the expiration date was 3 business days away". Another approach would be to group requirements by priority category.
- A good requirement is:
  - Correct
  - Unambiguous (all statements have exactly one interpretation)
  - Complete (where TBDs are absolutely necessary, document why the information is unknown, who is responsible for resolution, and the deadline)
  - Consistent
  - Ranked for importance and/or stability
  - Verifiable (avoid soft descriptions like “works well”, “is user friendly”; use concrete terms and specify measurable quantities)
  - Modifiable (evolve the Requirements Specification only via a formal change process, preserving a complete audit trail of changes)
  - Does not specify any particular design
  - Traceable (cross-reference with source documents and spawned documents).

3.1 Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_01	The system should have a web application in which hotel staff can interact		1	19/04/2021	
BR_02	The system should have a client-side interface	This will provide tourists with the information about the hotel and allow them to make reservations. They can also make reviews about the hotel, they can find information about the best attraction in the area of the hotel etc.	1	19/04/2021	
BR_03	The hotel manager and restaurant manager should have a dashboard where they can manage employees.	This means that they will have the possibility to manage the employees, by determining their role, adding or removing, reviewing their statistics during their work etc.	2	19/04/2021	
BR_04	The system should allow tourists to cancel the reservation.	A 20% penalty fee will be applied in this case	2	19/04/2021	
BR_05	The managers should be able to see the revenue generated in any given time.	This means that they will have the possibility to check their balance.	3	19/04/2021	
BR_06	The hotel manager/receptionist needs to manage the room in a proper way	This will provide the hotel manager/receptionist to manage the rooms in the hotel, which means they can do in the system the reservation of the room, declaring the payment of the room etc.	2	19/04/2021	
BR_07	The waiters should store the bills in the system.	This means that the waiters should declare their bills in the system, they can check their personal stats of the work, they can check of the available products that the restaurant or bar have etc.	2	19/04/2021	
BR_08	The management of the inventory by the manager.	This will help the managers to manage the inventory, by supplying new products, determining the remaining quantity of each product etc.	1	19/04/2021	

3.2 Non-Functional Requirements

3.2.1 User Interface Requirements

The web application can be opened in any browser. It will have an interface where tourists can make reservations. The first page that will display, will be the login page asking for the user's username and password. In case of correct credentials, the user can successfully enter the system. Otherwise they can not enter and an error message will be displayed.

3.2.2 Learnability

- The application is simple to use and understand.
- The web application will come together with a PDF manual, providing a step by step information on how to effectively use the system.
- Specific error messages will be displayed, by also identifying the specific action that caused the error.
- The application is specified for certain users, thus the system will know, when a certain action is not allowed

3.2.3 Performance

3.2.3.1 Capacity

The size of the application is dependent on the number of tables in the database, which will be moderate in complexity and will have a maximum size of 500 megabytes.

3.2.3.2 Availability

- The application is going to run in 24/7
- It is going to cover all geographical areas in the world which have access to the Internet.

3.2.3.3 Latency

Latency of this web application depends on:

1. Internet bandwidth
2. The database size
3. The number of users accessing the system in the same time

3.2.4 Manageability/Maintainability

3.2.4.1 Monitoring

Each user has its own information after he logs in, and therefore, does not have access to other

3.2.4.2 Maintenance

The database will be designed on MySql and Apache for the server. If the system crashes, the application is going to restart and redirect to the user dashboard, where he left. Then it is going to ask the user if he wants to save the unconfirmed changes.

3.2.4.3 Operations

- Some of the operations that users are required to do are:

- Manage employees
- Make or cancel reservations
- Check reservations
- Write a review

3.2.5 System Interface/Integration

Specify the use of other required products (e.g., a database or operating system), and interfaces with other systems (e.g., UWHires package interfaces with PubCookie and ODS, HEPPS system interfaces with Budget system). For each interface, define the interface in terms of message format and content. For well-documented interfaces, simply provide a reference to the documentation.

Outline each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

3.2.5.1 Network and Hardware Interfaces

Specify the logical characteristics of each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

3.2.5.2 Systems Interfaces

Example systems interface requirements:

A. System1-to-System2 Interface

The <external party> will create and send a fixed length text file as an email attachment to System2mail@u.washington.edu to be imported into the System2 system for payroll calculation. This file must be received on EDIT day by 4:00 PM in order to be processed in the EDIT night run. The requirements below document the file specifications, data transfer process, and specific schedule. This file is referred to as "FileName" in this document.

File Structure and Format

A1. The FileName file is a fixed length text file.

A2. The FileName file is an unformatted ASCII file (text-only).

A3. The FileName file contains a batch totals record and several detail records.

File Description: Batch Totals Record

A4. The batch totals record can be placed at the beginning, in the middle, or at the end of the file.

A5. The batch totals record contains the following:

- Record Type (value: XA)
- Process Type (value: A)
- Batch Number (3 digit number assigned by Payroll Dept)
- Origin Code (AIG)
- Total number of detail records
- Total deduction amount

File Description: Detail Records

A6. The FileName file contains a row for each record meeting xxx criteria.

A7. Each row in the FileName file contains the following fields, comma-delimited and encased in double-quotes where the data includes commas or spaces:

- Employee Id
- Record Type
- Process Date (MMDDYY)
- XYG Number

- Element Code
- Amount
- Amount Sign
- Year Flag
- Total Amount
- Total Amt Sign

3.2.6 Security

3.2.6.1 Protection

- In the backend , a function will be implemented to check the credentials of every user who logs in. Data integrity among users should be preserved.

3.2.6.2 Authorization and Authentication

3.2.6 Data Management

Possibles entities that the database of this web app will contain are:

- User
- Inventory
- Bills
- Room reservations
- Invoice

3.2.7 Standards Compliance

The application will respect the rules and regulations determined by the Ministry of Finance.

3.2.8 Portability

- The web application can be accessed in any browser of a mobile device or computer which is connected to the Internet.

3.2.9 Other Non-Functional Requirements

Please provide all necessary non-functional requirements, similar to the requirements explained in the lesson slides or in the textbook.

3.3 Domain Requirements

Everything related to the domain that might be needed in the project shall be mentioned here. Sometimes the domain Requirements might be thought as part of either functional or non-functional requirements.

4. User Scenarios/Use Cases

Provide a summary of the major functions that the product will perform. Organize the functions to be understandable to the customer or a first time reader. Include use cases and business scenarios, or provide a link to a separate document (or documents). A business scenario:

- Describes a significant business need
- Identifies, documents, and ranks the problem that is driving the scenario
- Describes the business and technical environment that will resolve the problem
- States the desired objectives
- Shows the “Actors” and where they fit in the business model
- Is specific, and measurable, and uses clear metrics for success

Scenario 1: Login successful

Flow of events:

1. The user is asked to enter his credentials: username and password
2. If the credentials are correct, he is logged in.
3. He will be redirected to the interface of his dashboard.

Scenario 2: Login unsuccessful

Flow of events:

1. The user is asked to enter his credentials: username and password
2. Credentials are checked in the users’ database
3. If the credentials are incorrect, he can not login and he is prompted to re enter the credentials.

Scenario 3: Manager registers an employee

Flow of events:

1. Manager logs in the system
2. He completes the form with the information about an employee
3. He clicks on add employee button
4. Employee data is stored in the database with its title, roles etc.
5. Employee is added to the system and a confirmation message appears on the screen
6. Employee can now login on the system

Scenario 4:Receptionist (User) Logs in

Flow of events:

1. Customer books a room to the Receptionist

2. Receptionist logins in the system as described in scenario 1 and 2.
3. He checks available rooms from the Dashboard
4. He marks Room as booked or not, taking into account the customer's preferences
5. Database entries are updated accordingly, a timer and a fee is placed based on the services selected
6. Then the room is marked as available when the timer ends

Scenario 5:Restaurant Manager

Flow of events:

1. Log in phase.
2. Manager checks inventory
3. He can select / add and remove items from a wishlist
4. The items will be added/ removed as database entities

Scenario 6 : Manager checks the revenue made in a day

Flow of events:

1. Receptionists and service staff register bills in the management system.
2. Hotel and Restaurant Manager checks the revenue generated in any interval of time they want

Scenario 7 : A tourist can make a reservation

Flow of events:

1. Tourist inserts personal information and the date when he wants to book
2. If the date is available, the user can make the reservation, otherwise a message will be displayed
3. Then, the user is asked to make the payment
4. The user finally submits the payment

APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

- Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
- Supporting or background information that can help the readers of the Requirements Specification;
- A description of the problems to be solved by the system;
- Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

Appendix A. Definitions, Acronyms, and Abbreviations

Define all terms, acronyms, and abbreviations used in this document.

Appendix B. References

List all the documents and other materials referenced in this document.

Appendix C. Requirements Traceability Matrix

The following trace matrix examples show one possible use of naming standards for deliverables (FunctionalArea-DocType-NN). The number has no other meaning than to keep the documents unique. For example, the Bargaining Unit Assignment Process Flow would be BUA-PF-01.

For example (1):

Business Requirement	Area	Deliverables	Status
BR_LR_01 The system should validate the relationship between Bargaining Unit/Location and Job Class.---Comments: Business Process = "Assigning a Bargaining Unit to an Appointment" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_09 The system should provide the capability for the Labor Relations Office to maintain the job class/union relationship.---Comments: Business Process = "Maintenance" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-02 BU Assignment Rules Maint Process Flow Diagram	ReadyForReview

For example (2):

BizReqID	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
BR_LR_01	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_01	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_01	1	BUA	BUA-PF-01	Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_01	1	BUA	BUA-UCD-01	BU Assign LR UseCase Diagram	ReadyForReview

BR_LR_01	1	BUA	BUA-UCT-001	BU Assignment by PC UseCase - Add Appointment and Derive UBU	Reviewed
BR_LR_01	1	BUA	BUA-UCT-002	BU Assignment by PC UseCase - Add Appointment (UBU Not Found)	Reviewed
BR_LR_01	1	BUA	BUA-UCT-006	BU Assignment by PC UseCase - Modify Appointment (Removed UBU)	Reviewed
BR_LR_09	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_09	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_09	1	BUA	BUA-PF-02	BU Assignment Rules Maint Process Flow Diagram	Accepted
BR_LR_09	1	BUA	BUA-UCD-03	BU Assign Rules Maint UseCase Diagram	Reviewed
BR_LR_09	1	BUA	BUA-UCT-045	BU Assignment Rules Maint: Successfully Add New Assignment Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-051	BU Assignment Rules MaintUseCase: Modify Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-053	BU Assignment Rules MaintUseCase - Review Assignment Rules	Reviewed
BR_LR_09	1	BUA	BUA-UCT-057	BU Assignment Rules MaintUseCase: Inactivate Last Rule for a BU	Reviewed
BR_LR_09	1	BUA	BUA-UI-02	BU AssignRules Maint UI Mockups	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-021	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-027	BU Assignment Rules Maint TestCase: Modify Rule - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-035	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Error Condition	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-049	BU Assignment Rules Maint TestCase: Modify Rule - Error Condition	ReadyForReview

For example (3):

BizReqID	CD01	CD02	CD03	CD04	UI01	UI02	UCT01	UCT02	UCT03	TC01	TC02	TC03	TC04
BR_LR_01			X		X		X			X		X	
BR_LR_09	X			X		X			X		X		X
BR_LR_10	X			X					X		X		
BR_LR_11		X											

Appendix D. Organizing the Requirements

This section is for information only as an aid in preparing the requirements document.

Detailed requirements tend to be extensive. Give careful consideration to your organization scheme. Some examples of organization schemes are described below:

By System Mode
Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency.

By User Class
Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fire fighters.

By Objects
Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

By Feature
A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs, and may include validity checks on inputs, exact sequencing of operations, responses to abnormal situations, including error handling and recovery, effects of parameters, relationships of inputs to outputs, including input/output sequences and formulas for input to output.

By Stimulus
Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

By Response
Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc.

By Functional Hierarchy
When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

Additional Comments
Whenever a new Requirements Specification is contemplated, more than one of the organizational techniques given above may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.