Software Requirements Specification

For

Hostel Management System

(Version 1.0)

*Prepared by:*

Laiba binta tahir

Arfah Ali

<Organization>

<Date created>

Contents

[1. Introduction 1](#_Toc152346147)

[1.1 Purpose 1](#_Toc152346148)

[1.2 Document Conventions 1](#_Toc152346149)

[1.3 Product Scope 2](#_Toc152346150)

[1.4 References 2](#_Toc152346151)

[2. Overall Description 2](#_Toc152346152)

[2.1 Product Perspective 2](#_Toc152346153)

[2.2 User Classes and Characteristics 3](#_Toc152346154)

[2.3 Operating Environment 3](#_Toc152346155)

[2.4 Design and Implementation Constraints 3](#_Toc152346156)

[2.5 Assumptions and Dependencies 3](#_Toc152346157)

[3. External Interface Requirements 4](#_Toc152346158)

[3.1 User Interfaces 4](#_Toc152346159)

[3.2 Hardware Interfaces 4](#_Toc152346160)

[3.3 Software Interfaces 4](#_Toc152346161)

[3.4 Communications Interfaces 5](#_Toc152346162)

[4. Data requirements 5](#_Toc152346163)

[4.1 Logical data model 5](#_Toc152346164)

[4.2 Data dictionary 5](#_Toc152346165)

[4.3 Reports 5](#_Toc152346166)

[4.4 Data acquisition, integrity, retention, and disposal 5](#_Toc152346167)

[5.System Features 5](#_Toc152346168)

[5.1 System Feature 1 5](#_Toc152346169)

[5.2 System Feature 2 (and so on) 6](#_Toc152346170)

[6 Other Nonfunctional Requirements 6](#_Toc152346171)

[6.2 Performance Requirements 6](#_Toc152346172)

[6.3 Safety Requirements 7](#_Toc152346173)

[6.4 Software Quality Attributes 7](#_Toc152346174)

[6.5 Business Rules 7](#_Toc152346175)

[7 Other Requirements 7](#_Toc152346176)

[7. Internationalization and localization requirements 8](#_Toc152346177)

[8. Other requirements 8](#_Toc152346178)

[Appendix A: Glossary 8](#_Toc152346179)

[Appendix B: Analysis models 8](#_Toc152346180)

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.>

This Software Requirements Specification (SRS) adheres to the following document conventions to ensure clarity, consistency, and effective communication:

1. Typography:

Requirements and sections titles are presented in bold.

Emphasis or key points may be italicized for emphasis.

2. Priority Notation:

Each requirement statement is associated with a priority level to indicate its relative importance and urgency. The priority levels are denoted as follows:

- High (H): Critical to the core functionality and success of the system.

- Medium (M): Important but not critical to the core functionality.

- Low (L): Desirable but can be deferred without compromising core functionality.

3. Inheritance of Priorities:

Unless explicitly stated otherwise, higher-level priorities are assumed to be inherited by detailed requirements. However, each requirement statement is capable of having its own priority assignment for fine-grained control.

4. Formatting:

Use of a consistent format for requirement statements, including actor-action-expected outcome structure where applicable. Diagrams and charts may be used to illustrate complex concepts, workflows, or relationships.

5. Color Coding:

In cases where color is used, it will be for enhancing readability or emphasis rather than carrying specific meaning.

## 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

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

# 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.>

## User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## 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).>

## 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

## 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.>Top of Form

## 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.>

# 4. Data requirements

## 4.1 Logical data model

## 4.2 Data dictionary

## 4.3 Reports

## Data acquisition, integrity, retention, and disposal

# 5.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.>

## 5.1 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)

This section outlines the major system features of the Hostel Management System, describing each feature's purpose, priority, stimulus/response sequences, and associated functional requirements.

# 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.>

## 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.>

# 7. Internationalization and localization requirements

# 8. Other requirements

## Appendix A: Glossary

## Appendix B: Analysis models