Final Year Design Project System Requirements Specification

AI-Powered Course Recommendation System

Final Year Design Project SRS

by

Names Roll Number

Names Roll Number

Names Roll Number

Names Roll Number

Names Roll Number

Project Advisor:

Faculty of Computing & Information Technology

University of the Punjab Lahore, Pakistan

(20xx)

***Project Title***

# **Executive Summary**

*[A brief summary of software requirements specification ]*

Table of Contents

[Executive Summary 2](#_Toc179539322)

[Requirements Analysis 5](#_Toc179539323)

[User classes and characteristics 5](#_Toc179539324)

[Requirement Identifying Technique 5](#_Toc179539325)

[Functional Requirements 5](#_Toc179539326)

[Functional Requirement X 5](#_Toc179539327)

[Non-Functional Requirements 6](#_Toc179539328)

[Reliability 6](#_Toc179539329)

[Usability 7](#_Toc179539330)

[Performance 7](#_Toc179539331)

[Security 7](#_Toc179539332)

[External Interface Requirements 7](#_Toc179539333)

[User Interfaces Requirements 7](#_Toc179539334)

[Software interfaces 8](#_Toc179539335)

[Hardware interfaces 8](#_Toc179539336)

[Communications interfaces 8](#_Toc179539337)

[Use case Analysis 9](#_Toc179539338)

[Use Case #1 (use case name and unique identifier – e.g. U1) 9](#_Toc179539339)

[Use Case #2…. 9](#_Toc179539340)

[Use Case Diagram 9](#_Toc179539341)

[Storyboards 10](#_Toc179539342)

[Summary 10](#_Toc179539343)

[References 11](#_Toc179539344)

**List of Tables**

[Table 1 Function Requriements Specification for Requirement Search Courses 6](#_Toc179539074)

[Table 2 Use case Description for the Use case Recommend Course 9](#_Toc179539075)

# Requirements Analysis

Provide an introduction to work done in SRS in few lines.

Write details in each of the sections provided ahead.

## User classes and characteristics

Identify the various user classes that you anticipate will use this product and describe their pertinent characteristics using 2 column table.

|  |  |
| --- | --- |
| User Class | User Characteristics |
|  |  |

## Requirement Identifying Technique

This section describes the requirements identifying technique(s) which further help to derive functional requirements specification. The selection of the technique(s) will depend on the type of project. For instance,

* **Use case includes** **detailed use case descriptions & use case diagram)** is an effective technique for interactive end-user applications. Use case name and associated requirements must be presented here. However, use case descriptions must be given in fully dressed format.
* **Storyboarding** for graphically intensive applications. Visual representation of sequence of events, designs to represent the flow.

## Functional Requirements

This section describes the functional requirements of the system expressed in the natural language style. This section is typically organized by feature as a system feature name and specific functional requirements associated with this feature.

### Functional Requirement X

Itemize the specific functional requirements associated with each feature. These are the software capabilities that must be implemented for the user to carry out the feature’s services or to perform a use case. Describe how the product should respond to anticipated error conditions and to invalid inputs and actions. Uniquely label each functional requirement, as described earlier. You can create multiple attributes for each functional requirement, such as rationale, source, dependencies, etc. The following template is required to write functional requirements.

Table 1 Function Requriements Specification for Requirement Search Courses

|  |  |
| --- | --- |
| **Identifier** | FR-1 |
| **Title** | Title of requirement |
| **Requirement** | Description of requirement which may be written either from the user or system perspective e.g.  If written in a user perspective  The [user class or actor name] shall be able to [do something] [to some object] [qualifying conditions, response time, or quality statement].  If written in a system perspective  [optional precondition] [optional trigger event] the system shall [expected system response] |
| **Source** | Where this requirement comes from (who originate it) |
| **Rationale** | The motivation behind the requirement |
| **Business Rule (if required)** | Any restriction, policy, the rule that the particular requirement must be fulfilled through its functional behavior |
| **Dependencies** | Requirements ID that is dependent on this requirement |
| **Priority** | High/Medium/Low |

## Non-Functional Requirements

This section specifies nonfunctional requirements other than constraints, supporting requirements recorded in Functional Requirements section, and external interface requirements. These quality requirements should be specific, quantitative, and verifiable. The following are some examples of documenting guidelines.

### Reliability

Requirements about how often the software fails. The measurement is often expressed in MTBF (mean time between failures)., and Mean time to Recover Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error detection, and a strategy for correction.

Example:

Incremental Backup will be automatically saved on every Saturday at 9pm

Data Integrity will ensure that 95%

### Usability

Usability requirements deal with ease of learning, ease of use, error avoidance and recovery, the efficiency of interactions, and accessibility. The usability requirements specified here will help the user interface designer create the optimum user experience. It may be quantified using the concepts of effort required to perform a task in terms of number of interactions required, considerations and solutions to make an accessible solution.

e.g. Learner would be able to get course recommendations with atmost 4 clicks .

### Performance

State specific performance requirements for various system operations. It may include response time or expected resources utilization requirements. If different functional requirements or features have different performance requirements, it’s appropriate to specify those performance goals right with the corresponding functional requirements, rather than collecting them in this section.

e.g. Learner would be able to get course recommendations within 10 seconds.

### Security

One or more requirements about protection of your system and its data. The measurement can be expressed in a variety of ways (effort, skill level, and time) to break into the system.  Do not discuss solutions (e.g. passwords) in a requirements document.

## External Interface Requirements

This section provides information to ensure that the system will communicate properly with users and with external hardware or software elements. A complex system with multiple subcomponents should create a separate interface specification or system architecture specification. The interface documentation could incorporate material from other documents by reference. For instance, it could point to a hardware device manual that lists the error codes that the device could send to the software.

### 

### User Interfaces Requirements

Describe the logical characteristics of each user interface that the system needs. Some possible items to include are

* References to GUI standards or product family style guides that are to be followed.
* Standards for fonts, icons, button labels, images, color schemes, field tabbing sequences, commonly used controls, and the like.
* Screen layout or resolution constraints.
* Standard buttons, functions, or navigation links that will appear on every screen, such as a help button.
* Shortcut keys.
* Message display conventions.
* Layout standards to facilitate software localization.
* Accommodations for visually impaired users.

Document the user interface design details, such as specific dialog box layouts, in a separate user interface specification, not in the SRS. Including screen mock-ups in the SRS to communicate another view of the requirements is helpful but make it clear that the mock-ups and storyboards are not the committed screen designs. If the SRS is specifying an enhancement to an existing system, it sometimes makes sense to include screen displays exactly as they are to be implemented. The developers are already constrained by the current reality of the existing system, so it's possible to know up front just what the modified, and perhaps the new, screens should look like.

### Software interfaces

Describe the connections between this product and other software components (identified by name and version), including other applications, databases, operating systems, tools, libraries, websites, and integrated commercial components (If any).

### Hardware interfaces

Describe the characteristics of each interface between the software components and hardware components, if any, of the system. This description might include the supported device types, the data and control interactions between the software and the hardware, and the communication protocols to be used.

### Communications interfaces

State the requirements for any communication functions the product will use, including email, web browser, network protocols, and electronic forms.

# Use case Analysis

## Use Case #1 (use case name and unique identifier – e.g. U1)

TO DO: Provide a specification for each use case diagram, add the table caption for each use case description using “References > Insert Caption” option

Table 2 Use case Description for the Use case Recommend Course

|  |  |
| --- | --- |
| UC Identifier | UC1 |
| **Requirements Traceability** | Mention all requirements traced to this use case by referencing with their requirements number mentioned in functional requirements |
| **Purpose** | What is the basic objective of the use-case. What is it trying to achieve? |
| **Priority** | What is the priority. Low, Medium, High. Importance of this use case being completed and functioning properly when system is deployed. |
| **Preconditions** | Any condition that must be satisfied before the use case begins |
| **Post conditions** | The conditions that will be satisfied after the use case successfully completes |
| **Actors** | Actors (human, system, devices, etc.) that trigger the use case to execute or provide input to the use case |
| **Extends** | If this is an extension use case, identify which use case(s) it extends |
| Main Success Scenario | flow of events normally executed in the use-case (in terms of actor action and system response). It should start describing the events assuming that precondition is already true. Description should end leading to the post conditions specified. |
| Alternate Flows | a secondary flow of events due to infrequent conditions |
| Exceptions | Exceptions that may happen during the execution of the use case |
| Includes | Use case ID of the included function |

## Use Case #2….

## Use Case Diagram

Show the use cases contained in, and actors outside the system boundary. Show the actor interactions with respective use cases.

Note: System is not the actor. Use Computer Aided Software Engineering Tool (CASE tool) to design use case diagram.

# Storyboards

Storyboards are sequence of drawings representing how user might use the product. They show user steps and corresponding responses by the system.

Design storyboards for atleast 3 (OTHER THAN Login, or register user)

## **Summary**

Summary of the processes involved in requirements identification and usescase analysis. It may also reiterate requirements engineering process’s importance in addressing the projects objectives.

# References

List any documents or other resources to which this SRS refers, if any. These might include user interface style guides, standards, system requirements specifications, interface specifications, or the SRS for a related product. The following are a few examples of different resources.

**Book**

Author(s). Book title. Location: Publishing company, year, pp.

Example:

W.K. Chen. Linear Networks and Systems. Belmont, CA: Wadsworth, 1993, pp. 123-35.

**Article in a Journal**

Author(s). “Article title”. Journal title, vol., pp, date.

Example:

G. Pevere. “Infrared Nation.” The International Journal of Infrared Design, vol. 33, pp. 56-99, Jan. 1979.

**Articles from Conference Proceedings (published)**

Author(s). “Article title.” Conference proceedings, year, pp.

Example:

D.B. Payne and H.G. Gunhold. “Digital sundials and broadband technology,” in Proc. IOOC-ECOC, 1986, pp. 557-998.

**World Wide Web**

Author(s)\*. “Title.” Internet: complete URL, date updated\* [date accessed].

M. Duncan. “Engineering Concepts on Ice. Internet: www.iceengg.edu/staff.html, Oct. 25, 2000 [Nov. 29, 2003].