Final Year Design Project System Requirements Specification

AI-Powered Course Recommendation System

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Software Design Specification Document

by

|  |  |
| --- | --- |
| **Student Name** | **BXXF21Mxxx** |
| **Student Name** | **BXXF21Mxxx** |
| **Student Name** | **BXXF21Mxxx** |
| **Student Name** | **BXXF21Mxxx** |

Project Advisor:

**Prof. Dr. Engr. Shahzad Sarwar**

Faculty of Computing & Information Technology,

University of the Punjab, Lahore, Pakistan.

(20xx)

# AI-Powered Course Recommendation System

## Executive Summary

*[A brief summary of software design specification document, not more than two paragraph/1 page ]*

Table of Contents

[AI-Powered Course Recommendation System 2](#_Toc181298187)

[Executive Summary 2](#_Toc181298188)

[System Design 5](#_Toc181298189)

[1. Design Considerations 5](#_Toc181298190)

[2. Requirements Traceability Matrix 5](#_Toc181298191)

[3. Design Models 6](#_Toc181298192)

[3.1 Architectural Design 6](#_Toc181298193)

[3.2 Data Design 6](#_Toc181298194)

[3.2.1 Data Dictionary 6](#_Toc181298195)

[3.3 User Interface Design 7](#_Toc181298196)

[3.3.1 Screen Images 7](#_Toc181298197)

[3.3.2 Screen Objects and Actions 7](#_Toc181298198)

[3.4 Behavioural Model 7](#_Toc181298199)

[4. Design Decisions 8](#_Toc181298200)

[5. Summary 8](#_Toc181298201)

[References 9](#_Toc181298202)

[[1]](#footnote-1)

**List of Tables**

[Table 1 Requirements Traceability Matrix 5](#_Toc181297964)

**Tables of Figures**

**No table of figures entries found.**[[2]](#footnote-2)

# System Design

Explain the product perspective in which software will operate including dependencies and interaction with other system. List the design constraints (like performance requirements) impacting design.

## Design Considerations

**Assumptions and Dependencies**: Lists any assumptions and dependencies that influence the design decisions.

**Limitations**: Describes any limitations in the software that could affect its functionality, performance, or compatibility.

**Risks**: Identifies potential risks and how they were addressed in the design

## Requirements Traceability Matrix [[3]](#footnote-3)

Shows how each design element traces back to a specific requirement in the Software Requirements Specification (SRS), ensuring that all requirements are accounted for in the design. Use table to document requirements traceability, where requirement ID is a unique identifier assigned to the requirement, requirement description refers to brief description of the requirement discussed. Design specification entails list of the components/classes/algorithm where the requirement is addressed. [[4]](#footnote-4)

Table 1 Requirements Traceability Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | | **Requirement Description** | **Design Specification** |
| R1 | The system must be able to store and retrieve course details, including prerequisites, description, duration, and credit hours. | | Component “Course Data Managemen” |
| R2 | User profiles should capture information such as academic background, course preferences, and previous courses taken | | Component “User Authentication” |
| Rm | The system must provide course recommendations based on user profiles, academic history, and preferences. | | Component “AI-Driven Recommendation Engine” |
| Rn | Track course | | Component “Track”, class course |

## Design Models [[5]](#footnote-5)

Provide the descriptions of the following models used to describe the system design. Also ensure visibility of all diagrams.

The applicable models for the project using object-oriented development approach may include:

1. Class Diagram
2. Interaction Diagram (Either sequence or collaboration)
3. State Transition Diagram (for the projects which include event handling and backend processes)

## Architectural Design [[6]](#footnote-6)

Develop a component structure and explain the relationships between these components to achieve the complete functionality of the system. This is a high-level overview of system’s inter-modules collaboration to achieve the desired functionality.

Provide a diagram showing the major subsystems and their connections.

* UML Class relationship diagram and UML Component diagram
* After finalizing architecture style/pattern diagram (MVC, Client-Server, Layered, Multi-tiered) create a detailed mapping modules/components to each part of the architecture

## Data Design

Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed, and organized. This includes the dataset/ database design and/or other design of other data structures used.

### Data Dictionary

 A**lphabetical List of System Entities or Major Data with Types and Descriptions:**

* Prepare an alphabetical list of all system entities or significant data. For each entry, include the **data type** and provide a brief **description** of its purpose within the system.
* Present this information in a **two-column format**:
  + The **first column** will contain the **terminology** (e.g., object name, attribute, method, or method parameter).
  + The **second column** will provide the **description**, including data types, roles, or any other relevant information

 **Structured Approach: Functions and Function Parameters:**

* For systems following a structured programming paradigm, list **all functions** along with their **function parameters**. For each function, provide a description detailing its functionality, and for each parameter, specify the **data type** and its role within the function.

 **Object-Oriented (OO) Approach: Objects, Attributes, Methods, & Method Parameters:**

* For systems using the Object-Oriented (OO) approach, list the **objects** and their **attributes**, followed by a description. Additionally, list the **methods** associated with each object, along with their **method parameters**.

## User Interface Design

Describe overview of the functionality of the system from the user’s perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user. [[7]](#footnote-7)

### Screen Images

Display few screenshots showing the interface from the user’s perspective. These can be hand-drawn, or you can use Prototyping tool like Canva/ Figma. Just make them as accurate as possible. [[8]](#footnote-8)

### Screen Objects and Actions

A discussion of screen objects and actions associated with those object for two use cases.

## Behavioural Model

**Interaction Diagrams**: Includes diagrams (Sequence or Collaboration diagram) that illustrate interactions among components.[[9]](#footnote-9)

**State Diagrams**: Shows state transitions within the software, especially useful for systems with complex workflows or lifecycle states.[[10]](#footnote-10)

## Design Decisions

Highlight the design choices while designing your system (e.g. choosing the Object oriented design pattern, choosing between the algorithmic approaches, normalization level of the database etc). Clearly specify which approach you have used and why you have made a specific design choice.

## Summary

Summary of the key design ideas discussed, design refinement and decisions taken in the chapter. It may also reiterate chapter’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].

1. Once the document is completed, Right click on table of contents, Choose “Update Field” then choose “Update entire table” to automatically update table of contents [↑](#footnote-ref-1)
2. Once the document is completed, Use References -> Insert Table of Figures option in MS Word to automatically generate this table [↑](#footnote-ref-2)
3. Please don’t forget to add the caption of table (Use References -> Insert Caption option in MS Word) [↑](#footnote-ref-3)
4. Make sure that the list is complete and consistent with requirements mentioend in SRS [↑](#footnote-ref-4)
5. Please don’t forget to add the caption of each figure (Use References -> Insert Caption option in MS Word) [↑](#footnote-ref-5)
6. Please don’t forget to add the caption of each figure (Use References -> Insert Caption option in MS Word) [↑](#footnote-ref-6)
7. [User interface design will be presented in the Deliverable 4 - Prototype] [↑](#footnote-ref-7)
8. Please don’t forget to add the caption of each figure (Use References -> Insert Caption option in MS Word) [↑](#footnote-ref-8)
9. . Make sure that objects & methods mentioned in the interaction diagram are consistent with the class diagram [↑](#footnote-ref-9)
10. Make sure that each diagram has assigned a proper caption, so that table of figures can be created. [↑](#footnote-ref-10)