

<Project Name>

Functional Specifications and Methodology

**Project Advisor:**

<Adviser name>

**Group Members:**

<Name of Member 1> <Roll No>

<Name of Member 2> <Roll No>

<Name of Member 3> <Roll No>

<Name of Member 4> <Roll No>

National University of Computer and Emerging Sciences

Department of Computer Science

Lahore, Pakistan

Table of Contents

[Table of Contents i](#_Toc23336587)

[List of Tables ii](#_Toc23336588)

[List of Figures iii](#_Toc23336589)

[1. Introduction 3](#_Toc23336590)

[1.1 Purpose of this Document 3](#_Toc23336591)

[1.2 Intended Audience 3](#_Toc23336592)

[1.3 Definitions, Acronyms, and Abbreviations 3](#_Toc23336593)

[2. General Description 3](#_Toc23336594)

[2.1 User Characteristics 3](#_Toc23336595)

[2.2 Domain Overview 3](#_Toc23336596)

[3. Functionality 3](#_Toc23336597)

[3.1 Functional Requirements 3](#_Toc23336598)

[3.2 Non-Functional Requirements 3](#_Toc23336599)

[3.3 Assumptions 3](#_Toc23336600)

[4. System Architecture 3](#_Toc23336601)

[5. Use Cases 3](#_Toc23336602)

[5.1 Sample Use Case Name Here 3](#_Toc23336603)

[6. Graphical User Interfaces 3](#_Toc23336604)

[7. Database Design (*if required*) 3](#_Toc23336605)

[7.1 ER Diagram 3](#_Toc23336606)

[7.2 Data Dictionary 3](#_Toc23336607)

[8. Risk Analysis 3](#_Toc23336608)

[9. System Requirements 3](#_Toc23336609)

[9.1 Hardware Requirements 3](#_Toc23336610)

[9.2 Software Requirements 3](#_Toc23336611)

[10. References 3](#_Toc23336612)

[11. Appendix 3](#_Toc23336613)

[12. Formatting Guidelines (This chapter is not part of FS) 3](#_Toc23336614)

[12.1 Tables and Figure 3](#_Toc23336615)

[12.2 Equations 3](#_Toc23336616)

[12.3 Header/Footer 3](#_Toc23336617)

[12.4 References 3](#_Toc23336618)

List of Tables

[Table 2: This is Sample table caption 3](#_Toc19700198)

List of Figures

Figure 1: List of Styles 3

# Introduction

The introduction of the **Software Functional Specification** provides an overview of the entire **Project**. It includes the purpose, scope, definitions, acronyms and abbreviations used in this document.

## Purpose of this Document

Specify the purpose of this Project. The FS fully describes the external behavior of the system. It also describes nonfunctional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

The purpose of this document is to identify the users of the system. It mentions both the functional and nonfunctional requirements for the FYP management system. It also describes the use cases and the system architecture. Furthermore the document explains about other factors and requirements necessary to provide a complete and comprehensive description of the requirements for the *FYP management system*.

## Intended Audience

The intended audience of this document is,

* Developers
* Evaluation panel
* Teachers and Supervisors
* Quality Assurance

## Definitions, Acronyms, and Abbreviations

Please note that the following two sections may or may not be the same as in the RS. The purpose of having the same sections in the FS is that there may have been some changes in the project scope and/or functionality, since the RS was made. Students must note that their implementation etc. should be reflected from the FS document, not the RS document.

# 

# General Description

## User Characteristics

Identify and briefly describe who the users of this system will be, and their roles.

The FYP management system will support three kinds of users; which are the FYP program organizer (PO) or the administrators, project supervisors and the students.

### FYP programme organizer or the administrator

The system is therefore useful to help FYP Programme organizer to arrange project selection and allocation procedure. During the academic year, the system would provide different functionalities for the PO to collect student’s assessments and markers’ scores via the submission and grading module.

### Project Supervisors

Project supervisors will also be able to employ the system for keep tracking the progress of the projects with the use of the project management tools, and online chat function.

### University Students

Search for instructors, search for group members,

Project group members themselves are provided with similarly effective communication tools to allow easy of discussion on project issues among members. Moreover, they can share resources about the project including source code and data files using our online repository.

## Domain Overview

Describe in full detail what your system does. This section can be further divided into sub-sections, but basically a paragraph description of the system is required.

The aim is to develop an automated system which is self-sufficient for organizing, managing, controlling all the activities of the Final year project and to better support three kinds of users; which are the FYP programme organizer (PO) or the administrators, project supervisors and the students. The system will facilitate the whole FYP process to support the users mentioned above. The coordinator will get a platform to separately carry out the registration and scheduling procedure. The system will also allow the coordinator to inform students about deadlines of the deliverables, provide templates and assign evaluators to projects. Rather than having to rush to offices, students will be offered a list of available faculty members, information about their availability, their preferred field of interest, their current and previous FYP’s details via an online portal so that it is easy and less stressful for students to select their FYP supervisor. Once registered, both faculty and students will interact via the portal, with both parties receiving appropriate notifications on pending work and incoming deadlines. The supervisor will be able to view the evaluations as well as progress of his students through a timeline. The students will get their FYP related queries, whether they are related to deadlines or grading, all at one place. Meetings can also be scheduled through the portal.

# 

# Functionality

## Functional Requirements

Identify and list each functionality and give a brief description, along with the user of each functionality.

The functionality may or may not have changed from the RS.

### Functional Requirements for a User (Coordinator, Supervisor, Student)

1. System shall allow user to login when user shall enter his /her email and password correctly.
2. Sign up
3. change password or edit profile

### Functional Requirements for an Administrator

1. PO operations kai liye interface
2. PO - registration process kai liye interface
3. PO - project allocation wala process hello
4. PO - deadlines inform karnay kai liye interface
5. PO - share available instructors, share their details
6. PO marks and grades show karay ga

### Functional Requirements for an Instructor

1. Instructor - share available slots
2. accept and reject a groups
3. instructor can view progress of his students
4. instructor can arrange meetings

### Functional Requirements for a Student

1. Students can search and contact new students to make a group
2. Students can accept or reject the offer
3. students can view instructors availability and info to select the instructor they want
4. contact instructors
5. see progress of their FYP
6. view grades of their fyp
7. share resources with memebers

## Non-Functional Requirements

This section should describe all the non-functional requirements including: reusability, performance (how many maximum users can access it at a time), extensibility etc.

## Assumptions

List down all the assumptions, made for the specification

# 

# System Architecture

This section should describe both the internal architecture of the modules, as well as the external architecture of the system with other systems, if any. Diagrammatic architecture is compulsory.

# 

# Use Cases

This section lists use cases or scenarios from the use-case model if they represent some significant, central functionality of the final system, or if they have a large architectural coverage—they exercise many architectural elements or if they stress or illustrate a specific, delicate point of the architecture.

## Sample Use Case Name Here

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Sample Use Case Name Here | | |
| Actors | | Admin, Business Owner, Store Manager | | |
| Summary | | The user shall provide their email and password on the login form and after successful verification, redirect the user to the home page. | | |
| Pre-Conditions | | The user must be in the database records either added by any of the authorized users or added manually by a developer.  The user must not already be logged in. | | |
| Post-Conditions | | The user’s session is successfully established and shall be redirected to the home page. | | |
| Special Requirements | | None | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | The user opens the login page. | | 2 | The login page is displayed asking for email and password. |
| 3 | The user enters valid email and password. | | 4 | The system verifies the email and password, establishes a session for the user and redirects the user to the home page. |
| **Alternative Flow** | | | | |
| 3 | The user enters invalid email or password. | | 4-A | The system responds with an error message: *Incorrect email or password entered.* |

**Note: A sample use case is given in the above table. You have to follow this tabular format for use cases. Following is the description of content required in this section:**

This section lists use cases or scenarios from the use-case model if they represent some significant, central functionality of the final system, or if they have a large architectural coverage—they exercise many architectural elements or if they stress or illustrate a specific, delicate point of the architecture.

**Summary:**The description briefly conveys the role and purpose of the use case. A single paragraph will suffice for this description.

**Basic Flow:**This use case starts when the actor does something. An actor always initiates use cases. The use case describes what the actor does and what the system does in response. It is phrased in the form of a dialog between the actor and the system.

The use case describes what happens inside the system, but not how or why. If information is exchanged, be specific about what is passed back and forth. For example, it is not very illuminating to say that the actor enters customer information. It is better to say the actor enters the customer’s name and address. A Glossary of Terms is often useful to keep the complexity of the use case manageable⎯you may want to define things like customer information there to keep the use case from drowning in details.

Simple alternatives may be presented within the text of the use case. If it only takes a few sentences to describe what happens when there is an alternative, do it directly within the **Flow of Events** section. If the alternative flow is more complex, use a separate section to describe it. For example, an **Alternative Flow** subsection explains how to describe more complex alternatives.

A picture is sometimes worth a thousand words, though there is no substitute for clean, clear prose. If it improves clarity, feel free to paste graphical depictions of user interfaces, process flows or other figures into the use case. If a flow chart is useful to present a complex decision process, by all means use it! Similarly for state-dependent behavior, a state-transition diagram often clarifies the behavior of a system better than pages upon pages of text. Use the right presentation medium for your problem, but be wary of using terminology, notations or figures that your audience may not understand. Remember that your purpose is to clarify, not obscure.

**First Alternative Flow:**More complex alternatives are described in a separate section, referred to in the **Basic Flow** subsection of **Flow of Events** section. Think of the **Alternative Flow** subsections like alternative behavior⎯ each alternative flow represents alternative behavior usually due to exceptions that occur in the main flow. They may be as long as necessary to describe the events associated with the alternative behavior. When an alternative flow ends, the events of the main flow of events are resumed unless otherwise stated.

**Second Alternative Flow:**There may be, and most likely will be, a number of alternative flows in a use case. Keep each alternative flow separate to improve clarity. Using alternative flows improves the readability of the use case, as well as preventing use cases from being decomposed into hierarchies of use cases. Keep in mind that use cases are just textual descriptions, and their main purpose is to document the behavior of a system in a clear, concise, and understandable way.

**Special Requirements:**A special requirement is typically a nonfunctional requirement that is specific to a use case, but is not easily or naturally specified in the text of the use case’s event flow. Examples of special requirements include legal and regulatory requirements, application standards, and quality attributes of the system to be built including usability, reliability, performance or supportability requirements. Additionally, other requirements⎯such as operating systems and environments, compatibility requirements, and design constraints⎯should be captured in this section.

**Pre-Conditions:**A pre-condition of a use case is the state of the system that must be present prior to a use case being performed.

**Post-Conditions:**A post-condition of a use case is a list of possible states the system can be in immediately after a use case has finished.

# Graphical User Interfaces

This section should give the GUI dumps of each screen, with reference to the user. The navigation flow of each user is also required, and each GUI should mark the functionality/use case that it covers.

# 

# Database Design (*if required*)

## ER Diagram

## Data Dictionary

# 

# Risk Analysis

List and explain the risks that maybe encountered during the project. For e.g.: technical risks, business risks etc.

# 

# System Requirements

List the hardware and software requirements that will be required to develop and deploy the project.

## Hardware Requirements

## Software Requirements

# References

List all important sources of information which have been consulted for this project

# Appendix

This section should include all supporting information from the project that was not included in the body of the report.  You should include surveys, complex statistical calculations, certain detailed tables and other such information in an appendix.  The information presented in this section is important to support the work presented in the body of the report but would make it more difficult to read and understand if presented within the body of the report.

Cite the appendix items in the report narrative (write "see Appendix A") and organize appendices (e.g., Appendix A, Appendix B,

Any tables, figures, forms, or other materials that are not totally central to the analysis but that need to be included are placed in the Appendix.

# Formatting Guidelines (This chapter is not part of FS)

This document also serves as style guide for final year project reports. In order to give a similar high quality appearance to all final year software project reports this template uses a collection of predefined Microsoft Word formatting styles. **These styles should be used without modification or replacement.** Following styles have been used

* **Title** – the main title style
* **Title2** – the subtitle style
* **Heading 1, Heading 2, Heading 3** – styles for different levels of section headings
* **Body Text** – style for paragraphs
* **Caption** – the style for a figure or table caption
* **Table Description** – the style for description of table
* **Figure Description** - the style for description of figure
* **Code** – the style for program source code
* **Table Header Row** – Style for the header row of table
* **Table Grid** – the style for the data rows in the tables
* **Reference** – The style for reference
* **Bullets** – The style for this list
* **Numbered** – Style for numbered list

Press Ctrl+Shift+S to see list of styles mentioned above. Figure 1 shows the Apply Style window displaying the list of styles. Select any text then press Ctrl+Shift+S, the Apply Style window will show you the current style applied on that text and if required, you can change the style by selecting any other style from the “Style Name” dropdown.

This is brief description of above figure.

Figure 1: List of Styles

## Tables and Figure

Tables and figures should be centered horizontally. The caption button should be used to insert caption for both the figures and tables. All figures and tables must be numbered properly. Always refer to tables and figures according to their numbers. A table or figure can be cited as follows: ‘see Table1’ or ‘as shown in Table1’. The caption of table should be centered above the table and figure caption should be centered below the figure. Place the tables/figures close to their reference. Use “Table Header Row” and ‘Table Grid’ style for table’sheader and data rows respectively. It is compulsory to provide brief description of table/figure after its caption. Styles for table and figure descriptions are “Table Description” and “Figure Description” respectively.

Table 1: This is Sample table caption

This is brief description of following Table.

|  |  |  |  |
| --- | --- | --- | --- |
| Header row | Header row | Header row | Header row |
| Row1 col1 | Row1 col2 | Row1 col3 | Row1 col4 |
| Row2 col1 | Row2 col2 | Row2 col3 | Row2 col4 |

## Equations

Use equation editor to write equations in this report. Use last button of the custom tool bar to invoke equation editor. Similar to tables and figures, equations should also be aligned centered horizontally. Number all important equations and insert them in parenthesis. Below is a sample equation and its reference number. An equation can be referenced like this: ‘it is clear from (1)’.

 (1)

## Header/Footer

Notice the headers in this document, before Introduction (i.e. the main content of this document) page numbers are in roman numerals. The page numbers of the actual content start with Arabic numerals i.e. 1, 2, 3 and so on. All the odd numbered pages contain title of your project while the even numbered pages contain the section heading (i.e. chapter’s name) in the headers.

## References

Always refer to the source of information by inserting the reference number in square brackets like this [5]. The reference numbers can either be added at the end of the sentence or within the sentence without changing the punctuation of sentence. A reference can also be cited as follows: ‘as Ruskey [2] mentioned’. List each source only once on your reference page.

**Following is a list of sample reference for various typed of sources in IEEE format.**

**Book**

1. P.M. Morse and H. Feshback, *Methods* of *Theoretical Physics*. New York: McGraw Hill, 1953.

**Journal Article**

1. S.K. Kenue and J.F. Greenleaf, “Limited angle multifrequency diffiaction tomography,” *IEEE Trans. Sonics Ultrason*., vol. SU-29, no. 6, pp. 213-2 17, July 1982.

**Dissertation or thesis**

1. B. Tsikos, “Segmentation of 3-D scenes using multi-modal interaction between machine vision and programmable mechanical scene manipulation,” Ph.D. dissertation, Univ. of Pennsylvania, BCE Dept., Philadelphia, 1987. [Add if applicable: University Microfilms, Inc., University of Michigan, Ann Arbor, Michigan.]

**Proceedings paper**

1. R. Finkel, R. Taylor, R. Bolles, R. Paul, and J. Feldman, “An overview of AL, programming system for automation,” in *Proc. Fourth Int. Joint Conf Artif. Intell*., pp. 758-765, Sept. 3-7, 1975.

**Newspaper article**

1. “Technology threatens to shatter the world of college textbooks, *The Wall Street Journal*, vol 91, pp. Al, A8, June 1, 1993.

**Technical Report**

1. R. Cox and J. S. Turner, “Project Zeus: design of a broadband network and its application on a university campus,” Washington Univ., Dept. of Comp. Sci., Technical Report WUCS-91-45, July 30, 1991.

**Software**

1. M. Janzen, *Instant Access Accounting*. Computer software. Nexus Software, Inc IBM-PC, 1993.

**World Wide Web** (give author and title if named)

1. Fuminao Okumura and Hajime Takagi, “Maglev Guideway On the Yamanashi Test Line,” *http://www.rtri.or.jp/rd/maglev2/okumura.html*, October 24, 1998.
2. “AT&T Supplies First CDMA Cellular System in Indonesia,” *http://www.att.com/press/1095/951011.nsa.html*, Feb 5, 1996.