**Final Year Project/Thesis**

**Handbook**

**(BSCS / MCS)**



**Department of Computer Science**

**Lahore Garrison University**

**Lahore**

**Foreword**

This is the Final Year Project handbook Department of Computer Science. This handbook contains guidelines for the conception, preparation, implementation, completion and finally the assessment of Final Year Projects.

The intention of this handbook is to develop guidelines and a uniform structure and outline for undergraduate students. It serves as an instructional manual for the expected contents, deliverables, quality and the required quantity of the final projects for students and also provides evaluation rubrics for supervisors and evaluators.

**FYP Coordinator:**

Taimoor Hassan

Basit Ali Gilani

Eisha-Tir-Razia

**Advisor:**

Dr. Khalid Masood

**Head of Department:**

Dr. Tahir Alyas

**Dean:**

Dr. Shazia Saqib

**Table of Contents**

[1.1 Introduction 5](#_Toc12806572)

[1.2 Degree Program Learning Outcomes 5](#_Toc12806573)

[1.3 Overview of Final Year Project 6](#_Toc12806574)

[1.4 FYP Milestones & Evaluation Stages 6](#_Toc12806575)

[1.4.1 Proposal Defense 7](#_Toc12806576)

[1.4.2 Mid Defense 7](#_Toc12806577)

[1.4.3 Final Defense 7](#_Toc12806578)

[1.5 Guidelines for Project Supervision 7](#_Toc12806579)

[1.5.1 Tasks Expected from Supervisors 7](#_Toc12806580)

[1.5.2 Project Development Life Cycle 8](#_Toc12806581)

[1.6 Team Leadership 9](#_Toc12806582)

[1.7 Students Responsibility 9](#_Toc12806583)

[1.8 Late submissions 9](#_Toc12806584)

[1.9 Plagiarism 10](#_Toc12806585)

[1.10 Project Documentation Guidelines 10](#_Toc12806586)

[Annex-A 12](#_Toc12806587)

[Annex-B 16](#_Toc12806589)

[1. Introduction 18](#_Toc12806590)

[1.1 Purpose 18](#_Toc12806591)

[1.2 Document Conventions 18](#_Toc12806592)

[1.3 Intended Audience and Reading Suggestions 18](#_Toc12806593)

[1.4 Product Scope 18](#_Toc12806594)

[2. Overall Description 18](#_Toc12806595)

[2.1 Product Perspective 18](#_Toc12806596)

[2.2 Product Functions 18](#_Toc12806597)

[2.3 User Classes and Characteristics 18](#_Toc12806598)

[2.4 Operating Environment 18](#_Toc12806599)

[2.5 Design and Implementation Constraints 18](#_Toc12806600)

[2.6 User Documentation 18](#_Toc12806601)

[2.7 Assumptions and Dependencies 19](#_Toc12806602)

[3. External Interface Requirements 19](#_Toc12806603)

[3.1 User Interfaces 19](#_Toc12806604)

[3.2 Hardware Interfaces 19](#_Toc12806605)

[3.3 Software Interfaces 19](#_Toc12806606)

[3.4 Communications Interfaces 19](#_Toc12806607)

[4. System Features 19](#_Toc12806608)

[4.1 System Feature 1 19](#_Toc12806609)

[4.1.1 Description and Priority 19](#_Toc12806610)

[4.1.2 Stimulus/Response Sequences 19](#_Toc12806611)

[4.1.3 Functional Requirements 19](#_Toc12806612)

[4.2 System Feature 2 (and so on) 20](#_Toc12806613)

[5. Other Nonfunctional Requirements 20](#_Toc12806614)

[5.1 Performance Requirements 20](#_Toc12806615)

[5.2 Safety Requirements 20](#_Toc12806616)

[5.3 Security Requirements 20](#_Toc12806617)

[5.4 Software Quality Attributes 20](#_Toc12806618)

[5.5 Business Rules 20](#_Toc12806619)

[References 20](#_Toc12806620)

[Annex-C 21](#_Toc12806621)

Annex-D ………………………………………………………………………………………………………………………………………………..37

# 1.1 Introduction

The Final Year Project (FYP) is the culmination of students’ degree program. The main purpose of this project is to encourage students to apply the knowledge acquired during their studies. It allows them to work on a substantial problem for an extended period of time, show how proficient they are in solving real world problems. It brings them a sound opportunity to demonstrate their competence as professionals and to apply what they have learnt in the other components of the degree. Besides, they get a chance to improve their technical skills, communication skills by integrating writing, presentation and learn how to work in teams. With a real-world problem at hand, they get to learn professional practice and a variety of non-technical issues such as management, finance, safety, reliability, environment and social impacts. Moreover, it provides an integrated assessment of the progress of the students toward the training they went through during their academic tenure at the university.

FYP course is different other courses because it demands independent objective formulation, planning, management and self-motivation. It is therefore essential to design fair and comprehensive guidelines for the students, supervisors and the evaluators. A structured manual and lifecycle process are therefore essential in order to help students conform to the required quality standards, outline general expectations from the supervisors and sketch assessment criteria for the evaluators. Hence, contribute as a fundamental underpinning to achieve high quality learning outcomes of the projects.

# 1.2 Degree Program Learning Outcomes

Program Learning Outcomes are the narrower statements that describe what students are expected to know and be able to do by the time of graduation. These relate to the knowledge, skills and attitude that the students acquire while progressing through the program. The program must demonstrate that by the time of graduation the students have attained a certain set of knowledge, skills and behavioral traits, at least to some acceptable minimum level. Specifically, it is to be demonstrated that the students have acquired the following graduate attributes:

1. **Engineering Knowledge:** An ability to apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis:** An ability to identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
3. **Design/Development of Solutions:** An ability to design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
4. **Investigation:** An ability to investigate complex engineering problems in a methodical way including literature survey, design and conduct of experiments, analysis and interpretation of experimental data, and synthesis of information to derive valid conclusions.
5. **Modern Tool Usage:** An ability to create, select and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modeling, to complex engineering activities, with an understanding of the limitations.
6. **The Engineer and Society:** An ability to apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice and solution to complex engineering problems.
7. **Environment and Sustainability:** An ability to understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.
9. **Individual and Team Work:** An ability to work effectively, as an individual or in a team, on multifaceted and /or multidisciplinary settings.
10. **Communication:** An ability to communicate effectively, orally as well as in writing, on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project Management:** An ability to demonstrate management skills and apply engineering principles to one’s own work, as a member and/or leader in a team, to manage projects in a multidisciplinary environment.
12. **Lifelong Learning:** An ability to recognize importance of, and pursue lifelong learning in the broader context of innovation and technological developments.

# 1.3 Overview of Final Year Project

A Final Year Project is a two-semester course in which students usually of 2-3 members (3 members are allowed, if project size is big) select a project and are supervised by a faculty member. In this course, students choose a project subject and define the objectives of the project under the supervision of a faculty member, and prepare the project proposal including: defining the statement of the problem, defining system requirements, defining different candidate solutions for the problem of study, making feasibility study for different candidate solutions, defining the best candidate solution, defining time table schedule. Students present the final project report at the end of the semester to an evaluation a committee.

# 1.4 FYP Milestones & Evaluation Stages

The FYP milestones, deliverables, evaluation stages along with their timelines is given the Table 1 and Table 2.

Table 1: Milestones & Marks Distribution

|  |  |
| --- | --- |
| **Milestones** | **Marks Distribution** |
| Proposal Defense | 20 |
| Mid Defense | 30 |
| Documentation | 50 |
| Internal Supervisor | 50 |
| Final Defense (Presentation/Viva) | 50 |
| **Total** | **200** |

Table 2: Detailed timeline of FYP phases

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Week** | **Duration** | **Semester** |
| Project & Supervisor Selection | Week 02 | 2 Weeks | 7th |
| Proposal Defense | Week 06 | 1 Week | 7th |
| Mid Defense | Week 14 | 1 Week | 7th |
| Final Report Submission | Week 14 | 1 Week | 8th |
| Final Defense | Week 14/15 | 1 Week | 8th |

## 1.4.1 Proposal Defense

The students must choose a project subject and define the objectives of the project under the supervision of a faculty member and prepare the project proposal. The template for FYP proposal is added in Appendix A.

## 1.4.2 Mid Defense

After proposal defense, next FYP activity is Mid-Defense. Students have to present their work and progress. Students also have to submit the Software Requirement Specifications (SRS) document at the time of mid defense. Template for SRS is attached in Annex B.

## 1.4.3 Final Defense

Final Defense is the final activity for FYP students, which is evaluated by Advisor, Co-advisor and evaluation committee. The template for FYP final report submission is attached in Annex C.

# 1.5 Guidelines for Project Supervision

Following rules should be taken under considerations during project supervision.

1. Each group (cross class section students are allowed) will work under the supervision of an assigned supervisor throughout the final year (term 7 & 8).
2. Students are recommended to meet with their supervisor at least once a week. The students are expected to discuss their progress with their supervisors in these weekly meetings. Depending on students’ requirements and the supervisor’s availability, supervisors may also arrange additional meetings (physical/online) as requested.
3. Supervisors might also arrange communication with student groups via email or other means for the purpose of advising project groups.
4. It is the responsibility of the supervisor to inform his students with this handbook and all the included instructions and regulations.

## 1.5.1 Tasks Expected from Supervisors

During these meetings supervisors are expected to:

1. **Orientation:**

* To provide FYP outlines/Objectives
* Discuss project expectations and the plan with the group
* To share pervious practice experience, research, skills and expertise

1. **Provide Knowledge:**

* Assign/Recommend related literature
* Give training sessions on the respective research area and tell them what they need to know
* To clarify students queries effectively as needed
* To make students aware of professional ethics and standards
* To advise students on how to deal effectively as a team while working under pressure, remaining optimistic and persistent, and how to meet milestone deadlines

1. **Assess:**

* To monitor the project progress on a weekly/fortnightly basis
* To ensure students are completing outlined project deliverables
* To grade students, work (at individual/group level) at the end of each semester

## 1.5.2 Project Development Life Cycle

The supervisors will guide the group through different steps in the software engineering life cycle and describe, discuss, assign, receive and review the corresponding outcomes/artifacts at the end of each step as described in Figure 1.

Figure 1: Software Engineering Lifecycle & Respective artifacts

During the Project Proposal, students undertake the initial phases of project planning, selection, analysis and designing phases. In the Project Implementation, students proceed with the implementation phase of their proposed project. As part of SDLC, the supervisors should guide the students to follow, but not limited to, the following best practices:

* 1. Having a life cycle or system development methodology
  2. Ensure proper research and background knowledge is acquainted
  3. Feasibility study is conducted on the proposed project
  4. Scope of the project is precise and crystal clear
  5. Generating and comparing alternative designs to determine best match for the requirements
  6. Roles & responsibilities of individual student working within the group is clear and accepted
  7. Able to apply project resources as per the approved project plan
  8. Track and report any issues and risks in completing assigned tasks
  9. Both logical and physical design aspects are analyzed
  10. Proper programming standards are maintained during the development of the project
  11. Auto or Manual Test Cases are implemented and executed
  12. Source control with versioning tools are used for developing as a team
  13. Documenting required deliverables using industry standards
  14. Participating in Seminars, Events, Publications and Workshops relevant to the project

# 1.6 Team Leadership

Every graduation project group is assigned with a team leader who is essentially a cross functional key player working within the project group. It is extremely important to get the right student within each group fit for the role of team leader which is crucial for the success of any project. Team leader should work very closely with the supervisor with the following, but not limited to, the primary tasks:

* 1. Provide input on the performance of team members
  2. Resolve any conflicts within group members and maintain healthy group dynamics
  3. Inform supervisor of any task delays and meeting hours change requests from students
  4. Ensure rest of the group understands their roles and responsibilities on the assigned tasks;
  5. Coordinate with internal or external project stakeholders on behalf of the team
  6. Provide weekly status report – completed and signed by each team member as per the schedule

**Note:** During the project, if the supervisor finds team leader is not performing as per the above expectations, the supervisors can request for the replacement of team leader with an alternative group member fit for this role.

# 1.7 Students Responsibility

During the Final Year Project, students are responsible for the following:

1. Agree with their assigned supervisors on the topic
2. Perform weekly tasks, assigned by the supervisor (or distributed by the team leader)
3. Discuss problems and seek advice from the supervisor in order to accomplish the assign tasks.
4. Provide supervisor weekly status reports and get his/her feedback
5. Apply recommendations to refine the previous task
6. Finalize the project proposal and implementation, incorporating all the feedbacks and
7. comments provided by the supervisor and evaluators.
8. Conduct presentation at the end of each semester and defend project to the evaluation panel

# 1.8 Late submissions

It is the responsibility of each group to ensure they complete the milestones of each semester and submit deliverables by the cutoff submission date. No project will be accepted after the cutoff date and necessary actions will be taken as per the supervisor and the evaluation committee decision policy.

# 1.9 Plagiarism

Each project (thesis base) must be the original work of student groups. At the end of each semester, students will be required to present their project proposal and implementation outcomes as per the provided deliverables guidelines and the original work undertaken throughout each semester.

In the project report, for instance, if students have taken ideas or referencing other work as part of the proposed project, then, it must be cited and reference should be clearly specified. Same is the case while implementing the proposed solution. For instance, if students are developing project using 3rd party tools and libraries, it must be referenced, and relevant comments and notes must be highlighted and will not be regarded as part of the original work of student groups. Hence, it is extremely important to note that it is the responsibility of students to ensure they are not plagiarizing knowingly or unknowingly.

# 1.10 Project Documentation Guidelines

Kindly find below the list of deliverables of your project.

1. **Project report (Minimum number of pages are 50)**
2. **Project one-page brochure/standee (On open house time)**

* One-page brochure depicting the true picture of your software application. One-page brochure must be designed to convince the Small Business Enterprises (SMEs) while if your application is ready to be purchased.

1. **Soft copies & hard copies**

* All files (project source code/presentation/documentation/project .exe file/) should be submitted on a CD. Use your One-Page brochure to make the cover of CD-Casing.
* Initial 1 document copy (Tape Binding) are required for evaluation time.
* After Approved by panel, then submit 3 hard copies of document. (Black color cover for BS program and Green color cover for MCS program)

1. **Project flex (size:4x3)**

A flex should be including

* Project title
* Student names with program/roll#/section
* Supervisor name
* 1-Snapshot of your Project Screen to encourage viewer to buy your project
* Write introduction/methodology/results/diagram

1. **Formatting guidelines**
2. Use TIMES NEW ROMAN font style for whole paragraph text.
3. For chapter name use Heading1 on right alignment (Bold with 20 font size), For Topic Heading use Heading2 on left alignment (Arial/Bold/Italic/14 font size), For Sub Heading use Heading3 on left alignment (Arial/Bold/13 font size).
4. Topics Heading on Left alignment.
5. For paragraph text, use FONT SIZE must be 12.
6. All text must be in justify alignment.
7. Paragraph spacing must be 6pt after and 12pt before.
8. Line spacing must be 1.5 lines.
9. Use Caption (For Table: caption on top side/For Figure: caption on bottom side)
10. Before Chapter 1 start, use roman numbering (i.e. I-V) as page number at bottom right and after that use (i.e. 1-10) as page number at bottom right.
11. **References**

All the references should be alphabetically ordered

* Journals/Conferences

Author Name/s (Surname, initial), year of publication in-parenthesis, title of the article, name of the journal/conference, volume number, issue number (in parenthesis) followed by a colon and page numbers

* Books

Author Name/s (Surname, initial), year of publication in-parenthesis, title of the book, publisher’s name, place of publication, page numbers

* Reference from Internet

Name of the Author/s (If Known), Title of the topic followed by complete web address

# Annex-A

**Project Title**

Final Year Project Proposal

(BSCS/MCS) write only one program

By

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S# | **Name** | **Registration #/Roll #/Section** | **Mobile #** | **Project ID** |
| 1. |  |  |  |  |
| 2. |  |  |  |

**Supervised by:**

Teacher Name **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (**Signature**)**

****

Department of Computer Science

Lahore Garrison University

Lahore

**Abstract**

Abstract of the proposal

**Introduction**

Discuss the opening perspective of the problem area, the challenge in that area and refine the challenge into a concise

**Problem statement**

What is problem the FYP is aiming to solve. How significant is the problem? Quantify as much as possible. In case of a research problem, show the significance of the unsolved problem.

List the type of customers who want a solution to the problem. For each type of customers indicate the potential market size. In case of a research problem, identify where this research will be used?

**Literature review**

What has been done by others to solve the problem? What solutions are already present in the market? what are their disadvantages? In case of a research problem, literature review of the state-of-the-art should be included.

**Project scope**

Discuss the overview/goal of the project and highlight the proposed solution. Give your value proposition. How is your solution going to be different and better than others? Students must describe to maximum detail the final project output, its expected packaging, and hardware and software components. In case of a research problem, how the proposed research solution is expected to be better than the state-of-the-art?

**Project development methodology**

Students may also mention tools, technologies and suitability of the method(s) to be employed with justification. In case of a research problem, show the few approaches that will be investigated in the project?

**Project milestones and deliverables**

Clear milestones should be defined at the start of the project which includes a Gantt chart in the project management document

**References**

A comprehensive list of references is cited using a standard format.

**Thesis Title**

Final Year Project Thesis Proposal

(BSCS/MCS) write only one program

By

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S# | **Name** | **Registration #/Roll #/Section** | **Mobile #** | **Project ID** |
| 1. |  |  |  |  |
| 2. |  |  |  |

**Supervised by:**

Teacher Name **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (**Signature**)**

****

Department of Computer Science

Lahore Garrison University

Lahore

**Abstract**

Abstract of the proposal

**Introduction**

Discuss the opening perspective of the problem area, the challenge in that area and refine the challenge into a concise

**Problem statement**

What is problem the FYP is aiming to solve. How significant is the problem? Quantify as much as possible. In case of a research problem, show the significance of the unsolved problem.

List the type of customers who want a solution to the problem. For each type of customers indicate the potential market size. In case of a research problem, identify where this research will be used?

**Literature review/Related work**

What has been done by others to solve the problem? What solutions are already present in the market? what are their disadvantages? In case of a research problem, literature review of the state-of-the-art should be included.

**Project scope**

Discuss the overview/goal of the project and highlight the proposed solution. Give your value proposition. How is your solution going to be different and better than others? Students must describe to maximum detail the final project output, its expected packaging, and hardware and software components. In case of a research problem, how the proposed research solution is expected to be better than the state-of-the-art?

**Project Objective**

Students may also mention tools, technologies and suitability of the method(s) to be employed with justification. In case of a research problem, show the few approaches that will be investigated in the project?

**Research Methodology**

## In this section, define the research methodology of your problem solution.

**Project milestones and deliverables**

Clear milestones should be defined at the start of the project which includes a Gantt chart in the project management document

**References**

A comprehensive list of references is cited using a standard format.

[1] I. Thompson, “Women and feminism in technical communication,” Journal of Business and Technical Communication, vol. 13, no. 2, pp.154–178, 1999. (**Journal Article**)

[2] M. S. MacNealy, Strategies for Empirical Research in Writing. Boston, MA: Allyn and Bacon, 1999. (**Book**)

[3] K. St.Amant, “Virtual office communication protocols: A system for managing international virtual teams,” in Proceedings of IEEE International Professional Communication Conference, 2005, pp. 703–717. (**Conference Paper**)

# Annex-B

**FYP SRS Document Template**

Final Year Project

Software Requirement Specification

For

Project Title

(BSCS/MCS) write only one program

By

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S# | **Name** | **Registration #/Roll #/Section** | **Mobile #** | **Project ID** |
| 1. |  |  |  |  |
| 2. |  |  |  |

**Supervised by:**

Teacher Name **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (**Signature**)**

****

Department of Computer Science

Lahore Garrison University

Lahore

Table of Contents

1. Introduction........................................................................................................... Page#

1.1 Purpose ........................................................................................................................

1.2 Document Conventions ................................................................................................

1.3 Intended Audience and Reading Suggestions ..............................................................

1.4 Product Scope ..............................................................................................................

2. Overall Description ...............................................................................................

2.1 Product Perspective......................................................................................................

2.2 Product Functions ......................................................................................................

2.3 User Classes and Characteristics ...............................................................................

2.4 Operating Environment ..............................................................................................

2.5 Design and Implementation Constraints ....................................................................

2.6 User Documentation ..................................................................................................

2.7 Assumptions and Dependencies ................................................................................

3. External Interface Requirements .......................................................................

3.1 User Interfaces ...........................................................................................................

3.2 Hardware Interfaces ...................................................................................................

3.3 Software Interfaces ....................................................................................................

3.4 Communications Interfaces .......................................................................................

4. System Features ..................................................................................................

4.1 System Feature 1 ........................................................................................................

4.2 System Feature 2 (and so on) .....................................................................................

5. Other Nonfunctional Requirements ...................................................................

5.1 Performance Requirements ........................................................................................

5.2 Safety Requirements ..................................................................................................

5.3 Security Requirements ...............................................................................................

5.4 Software Quality Attributes .......................................................................................

5.5 Business Rules ...........................................................................................................

References ....................................................................................................................

# 1. Introduction

## 1.1 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.>

## 1.2 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.>

## 1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## 1.4 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.>

# 2. Overall Description

## 2.1 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.>

## 2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high-level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top-level data flow diagram or object class diagram, is often effective.>

## 2.3 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.>

## 2.4 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.>

## 2.5 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).>

## 2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## 2.7 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).>

# 3. External Interface Requirements

## 3.1 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.>

## 3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## 3.3 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.>

## 3.4 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. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# 4. 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.>

## 4.1 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:**

## 4.2 System Feature 2 (and so on)

# 5. Other Nonfunctional Requirements

## 5.1 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.>

## 5.2 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.>

## 5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## 5.4 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.>

## 5.5 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.>

# 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.>

# Annex-C

**Project Report**

**Project Title**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Submitted by**

**Student Name** (Program-Roll#-Section)

**Student Name** (Program-Roll#-Section)

Session Fall 2017-2021

**Supervised by**

**Teacher Name**



**Department of Computer Science**

**Lahore Garrison University**

**Lahore**

**Project Title**

A project submitted to the

Department of Computer Science

In

Partial Fulfillment of the Requirements for the

Bachelor’s/ Master Degree in Computer Science

By

**Student Names**

**Supervisor**

**Teacher Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Designation   
Department of Computer Science

**External Examiner** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Chairperson**

**Dr. Tahir Alyas \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Head of Department   
Department of Computer Science

**COPYRIGHTS**

This is to certify that the project titled “**Project Title”** is the genuine work carried out by **Student Names,** student of BSCS of Computer Science Department, Lahore Garrison University, Lahore during the academic year 2015-19, in partial fulfilment of the requirements for the award of the degree of Bachelor of Computer Science and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Student Name \_\_\_\_\_\_\_\_\_\_\_\_

Student Name \_\_\_\_\_\_\_\_\_\_\_

**DECLARATION**

This is to declare that the project entitled “**Project Title**” is an original work done by undersigned, in partial fulfilment of the requirements for the degree “Bachelor of Science in Computer Science” at Computer Science Department, Lahore Garrison University, Lahore.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Student Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGEMENTS**

**DEDICATION**

**Table of Contents**

**List of Tables**

**List of Figures**

**List of Abbreviation**

**Abstract**

Include a brief summary of the problem statement, challenges, proposed solution, approaches, scope, and comparison with existing systems/evaluation methods, conclusion and future directions. Recommend length is 1 page maximum. Abstract must be self-explanatory and should not include any references or shorthand notations in the abstract.

**Chapter 1**

**Introduction**

Introduction is mostly written for non-specialists so that they can get an overview of the project without technical details. It should provide a brief overview of the project aims and structure of the solution. It should also specify what unmet need or problem the FYP caters for and who needs it. At the end of chapter, provide a summary of the report organization, chapter outlining what has been covered in this chapter and explain what comes in the following chapters.

**Chapter 2**

**Problem Definition**

Describe the problem definition according to the project.

**Chapter 3**

**Software Requirement Specification**

Describe the SRS of project.

**Chapter 4**

**Methodology**

The methods, approaches, tools, techniques, algorithms, or other aspects of the solution are sufficiently discussed with sufficient details and supporting figures.

**Chapter 5**

**Detailed Design and Architecture**

**5.1 SYSTEM ARCHIECTURE**

This section should provide a high-level overview of how the functionality and responsibilities of the system were partitioned and then assigned to subsystems or components. Don't go into too much detail about the individual components themselves (there is a subsequent section for detailed component descriptions). The main purpose here is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together to provide the desired functionality.

At the top-most level, describe the major responsibilities that the software must undertake and the various roles that the system (or portions of the system) must play. Describe how the system was broken down into its components/subsystems (identifying each top-level component/subsystem and the roles/responsibilities assigned to it). Describe how the higher-level components collaborate with each other in order to achieve the required results. Don't forget to provide some sort of rationale for choosing this particular decomposition of the system (perhaps discussing other proposed decompositions and why they were rejected). Feel free to make use of design patterns, either in describing parts of the architecture (in pattern format), or for referring to elements of the architecture that employ them.

If there are any diagrams, models, flowcharts, documented scenarios or use-cases of the system behavior and/or structure, they may be included here (unless you feel they are complex enough to merit being placed in the Detailed System Design section). Diagrams that describe a particular component or subsystem should be included within the particular subsection that describes that component or subsystem.

**5.1.1 Architecture Design Approach**

Describe the architectural design approach.

**5.1.2 Architecture Design**

Provide and describe a figure that depicts the overall system architecture. Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high-level overview of how responsibilities of the system were partitioned and then assigned to subsystems. Identify each high-level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality. Don’t go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagram if required.

**5.1.3 Subsystem Architecture**

Provide a decomposition of the subsystems in the architectural design. Supplement with text as needed. You may choose to give a functional description or an object-oriented description. For a functional description, put top level data flow diagram (DFD) and structural decomposition diagrams. For an OO description, put subsystem model, object diagrams, generalization hierarchy diagram(s) (if any), aggregation hierarchy diagram(s) (if any), interface specifications, and sequence diagrams here.

**5.2 DETAILED SYSTEM DESING**

Most components described in the System Architecture section will require a more detailed discussion. Other lower-level components and subcomponents may need to be described as well. Each subsection of this section will refer to or contain a detailed description of a system software component. The discussion provided should cover the following software component attributes:

**5.2.1 Classification**

The kind of component, such as a subsystem, module, class, package, function, file, etc.

**5.2.2 Definition**

The specific purpose and semantic meaning of the component. This may need to refer back to the requirements specification.

**5.2.3 Responsibilities**

The primary responsibilities and/or behavior of this component. What does this component accomplish? What roles does it play? What kinds of services does it provide to its clients? For some components, this may need to refer back to the requirements specification.

**5.2.4 Constraints**

Any relevant assumptions, limitations, or constraints for this component. This should include constraints on timing, storage, or component state, and might include rules for interacting with this component (encompassing preconditions, post conditions, invariants, other constraints on input or output values and local or global values, data formats and data access, synchronization, exceptions, etc.)

**5.2.5 Composition**

A description of the use and meaning of the subcomponents which are a part of this component.

**5.2.6 Uses/Interactions**

Description of component collaboration with other components. What other components is this entity used by? What other components does this entity use (this would include any side-effects this entity might have on other parts of the system)? This concerns the method of interaction as well as the interaction itself. Object-oriented designs should include a description of any known or anticipated subclasses, super classes, and meta classes.

**5.2.7 Resources**

A description of any and all resources that are managed, affected, or needed by this entity. Resources are entities external to the design such as memory, processors, printers, databases, or a software library. This should include a discussion of any possible race conditions and/or deadlock situations, and how they might be resolved.

**5.2.8 Processing**

A description of precisely how this component goes about performing the duties necessary to fulfill its responsibilities. This should encompass a description of any algorithms used; changes of state; relevant time or space complexity; concurrency; methods of creation, initialization, and cleanup; and handling of exceptional conditions.

**5.2.9 Interface/Exports**

The set of services (resources, data, types, constants, subroutines, and exceptions) that are provided by this component. The precise definition or declaration of each such element should be present, along with comments or annotations describing the meanings of values, parameters, etc. For each service element described, include (or provide a reference) in its discussion a description of its important software component attributes (Classification, Definition, Responsibilities, Constraints, Composition, Uses, Resources, Processing, and Interface).

Much of the information that appears in this section is not necessarily expected to be kept separate from the source code. In fact, much of the information can be gleaned from the source itself (especially if it is adequately commented). This section should not copy or reproduce information that can be easily obtained from reading the source code (this would be an unwanted and unnecessary duplication of effort and would be very difficult to keep up to date). It is recommended that most of this information be contained in the source (with appropriate comments for each component, subsystem, module, and subroutine). Hence, it is expected that this section will largely consist of references to or excerpts of annotated diagrams and source code. Any referenced diagrams or source code excerpts should be provided at any design reviews.

**5.2.10 Detailed Subsystem Design**

Provide a detailed description of this software component (or a reference to such a description). Complex diagrams showing the details of component structure, behavior, or information/control flow may be included in the subsection devoted to that particular component (although, unless they are very large or complex, some of these diagrams might be more appropriately included in the System Architecture section. The description should cover any applicable software component attributes (some of which may be adequately described solely by a source code declaration or excerpt).

**Draw following diagrams according to your project**

1. Use case Diagram
2. ER Diagram
3. Architectural Diagram
4. Activity Diagram
5. Sequence Diagram
6. Component Diagram
7. State Machine Diagram
8. Class Diagram
9. Data Flow Diagram
10. Database Diagram

**Chapter 6**

**Implementation and Testing**

Explain the methods, tools and techniques used to develop the software. What kind of software and testing methodologies implemented. Explain core functionalities in narrative format. Controlled libraries, templates, code walkthroughs, explain how the proposed software has been evaluated and compared at runtime with the original specifications. The Accuracy, Performance and Scalability of the proposed software must be critically analyzed and should solve identified problem statement.

**Chapter 7**

**Results and Discussion**

A comprehensive evaluation of the solution is presented with supporting figures and graphics. System testing is performed through a strong testing strategy and the test cases cover all the use cases.

**Chapter 8**

**Conclusion and Future Work**

Include a brief summary of how the proposed solution is going to/has addressed the problem statement specified in the introduction section. Provide an overview of what kind of evaluations were undertaken in order to prove that the solution really solves the problem with evidence on results findings. Provide an overview of the recommendations and Include a future direction which is required as part of the future work.

**References**

A comprehensive list of references is cited using a standard format.

[1] I. Thompson, “Women and feminism in technical communication,” Journal of Business and Technical Communication, vol. 13, no. 2, pp.154–178, 1999. (**Journal Article**)

[2] M. S. MacNealy, Strategies for Empirical Research in Writing. Boston, MA: Allyn and Bacon, 1999. (**Book**)

[3] K. St.Amant, “Virtual office communication protocols: A system for managing international virtual teams,” in Proceedings of IEEE International Professional Communication Conference, 2005, pp. 703–717. (**Conference Paper**)

**Thesis Report**

**Thesis Title**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Submitted by**

**Student Name** (Program-Roll#-Section)

**Student Name** (Program-Roll#-Section)

Session Fall 2017-2021

**Supervised by**

**Teacher Name**



**Department of Computer Science**

**Lahore Garrison University**

**Lahore**

**Thesis Title**

A thesis submitted to the

Department of Computer Science

In

Partial Fulfillment of the Requirements for the

Bachelor’s/Master Degree in Computer Science

By

**Student Names**

**Supervisor**

**Teacher Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Designation   
Department of Computer Science

**External Examiner** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Chairperson**

**Dr. Tahir Alyas \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Head of Department   
Department of Computer Science

**COPYRIGHTS**

This is to certify that the project titled “**Thesis Title”** is the genuine work carried out by **Student Names,** student of BSCS of Computer Science Department, Lahore Garrison University, Lahore during the academic year 2015-19, in partial fulfilment of the requirements for the award of the degree of Bachelor of Computer Science and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Student Name \_\_\_\_\_\_\_\_\_\_\_\_

Student Name \_\_\_\_\_\_\_\_\_\_\_

**DECLARATION**

This is to declare that the project entitled “**Thesis Title**” is an original work done by undersigned, in partial fulfilment of the requirements for the degree “Bachelor of Science in Computer Science” at Computer Science Department, Lahore Garrison University, Lahore.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Student Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGEMENTS**

**DEDICATION**

**Table of Contents**

**List of Tables**

**List of Figures**

**List of Abbreviation**

**Abstract**

Include a brief summary of the problem statement, challenges, proposed solution, approaches, scope, and comparison with existing systems/evaluation methods, conclusion and future directions. Recommend length is 1 page maximum. Abstract must be self-explanatory and should not include any references or shorthand notations in the abstract.

**Chapter 1**

**Introduction**

Introduction is mostly written for non-specialists so that they can get an overview of the project without technical details. It should provide a brief overview of the project aims and structure of the solution. It should also specify what unmet need or problem the FYP caters for and who needs it. At the end of chapter, provide a summary of the report organization, chapter outlining what has been covered in this chapter and explain what comes in the following chapters.

**In this also explain the project related concepts, problem description, research objectives, Gantt Chart and thesis organization.**

**Chapter 2**

**Literature Review** (for research base projects)

Provide an overview to the projects background knowledge without too much in detail (stick to the scope of the project). The background can refer to previous work referenced from journals, articles, newspapers, or any academic literature providing evidence that the proposed problem is significant and real problem worth solving. If available, provide closely related work done within the project scope and the challenges or defects identified which can be considered as part of the new solution. Describe why you worked on this project in light of the literature review?

**Chapter 3**

**Research Methodology**

The methods, approaches, tools, techniques, algorithms, or other aspects of the solution are sufficiently discussed with sufficient details and supporting figures.

**Chapter 4**

**Core Construction/Framework Model**

This document discusses the exact change/idea what you are add in this thesis. The indented to be guiding to begin the project development plan. The plan should be dynamic, changing with the project changes, but keeping the overall development plan documented. Also explain the system detailed design (if required in thesis)

**Draw following diagrams according to your thesis requirements**

1. Use case Diagram
2. ER Diagram
3. Architectural Diagram
4. Activity Diagram
5. Sequence Diagram
6. Component Diagram
7. State Machine Diagram
8. Class Diagram
9. Data Flow Diagram
10. Database Diagram

**Chapter 5**

**Experiments and Results**

**Chapter 6**

**Conclusion and Future Work**

**References**

A comprehensive list of references is cited using a standard format.

[1] I. Thompson, “Women and feminism in technical communication,” Journal of Business and Technical Communication, vol. 13, no. 2, pp.154–178, 1999. (**Journal Article**)

[2] M. S. MacNealy, Strategies for Empirical Research in Writing. Boston, MA: Allyn and Bacon, 1999. (**Book**)

[3] K. St.Amant, “Virtual office communication protocols: A system for managing international virtual teams,” in Proceedings of IEEE International Professional Communication Conference, 2005, pp. 703–717. (**Conference Paper**)