ISD TERM PROJECT

<Requirements Specification>

<Project Title>



Project Advisor

**<Advisor Name>**

Presented by:

**Group ID: xxxxxxx**

Student Reg# Student Name

**Faculty of Information Technology**

**University of Central Punjab**

Software Requirements Specification

Version <Version #>

<Project Name>

Advisor: <Advisor Name>

Group <Group ID>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Member Name | Primary Responsibility | % Contribution | Roll No | Serial # | Sign |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table of Contents

Table of Contents iii

1. Introduction and Background 1

1.1 Product (Problem Statement) 1

1.2 Background 1

1.3 Scope 1

1.4 Objective(s)/Aim(s)/Target(s) 1

1.5 Challenges 1

1.6 Learning Outcomes 1

1.7 Nature of End Product 1

1.8 Completeness Criteria 1

1.9 Business Goals 1

1.10 Related Work/ Literature Survey/ Literature Review 1

1.11 Document Conventions 1

2. Overall Description 2

2.1 Product Features 2

2.2 User Classes and Characteristics 2

2.3 Operating Environment 2

2.4 Design and Implementation Constraints 2

2.5 Assumptions and Dependencies 2

3. Project Management. 3

3.1 Work Breakdown Structure (WBS). 3

3.2 Develop an Activity Graph. 3

3.3 Find the Critical Path. 3

3.4 Create a Gantt chart. 3

~~~~~~~~~~ Phase II ~~~~~~~~~~ 3

4. Functional Requirements 3

4.1 Name of Use-Case 1 3

4.2 Name of Use-Case 2 (and so on) 4

Requirements Analysis and Modeling 4

5. Nonfunctional Requirements\* 4

5.1 Performance Requirements 4

5.2 Safety Requirements 4

5.3 Security Requirements 4

5.4 Additional Software Quality Attributes 5

6. Other Requirements 5

7. Designing 5

7.1 Complete class diagram 5

7.2 Complete Data Flow Diagram (DFD) 5

7.3 Complete ER Diagram 5

7.4 Physical design of your database 5

7.5 Information on use of design patterns while designing the modules 5

7.6 Make a Sequence and a Collaboration diagram of following. 5

7.6.1 Scenario 1 5

7.6.2 Scenario 2 5

8. Estimation 6

8.1 Cost Benefit Estimation 6

8.1.1 Return of Investment (ROI) {1st year} 6

8.1.2 %gain on ROI {1st year} 6

8.1.3 Payback Period in years 6

8.2 FP based Estimation 6

8.3 COCOMO Estimation 6

9. References 6

Appendix A: Glossary 7

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction and Background

You are required to write a brief introduction of your project and provide background. In this section you should provide the context and initial knowledge of the domain. You should also highlight the significance of problem and provide motivation behind the work being done.

## Product (Problem Statement)

<Provide a short description of the software being specified.>

## Background

<Describe the domain.>

## Scope

< Describe the scope of the product that is covered by this SRS.>

## Objective(s)/Aim(s)/Target(s)

<List all the objectives to be achieved after the completion of this project.>

## Challenges

## Learning Outcomes

## Nature of End Product

## Completeness Criteria

*<State the criteria that if fulfilled will let evaluators consider the project as complete >*

## Business Goals

<List all the business/corporate goals addressed by this software.>

## Related Work/ Literature Survey/ Literature Review

## 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 italicized nouns represent external systems.>

# Overall Description

## Product Features

<Summarize the major features the product contains or the significant functions that it performs or lets the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS.>

## 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 favored user classes from those who are less important to satisfy.>

## Operating Environment

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

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# 

# Project Management.

## Work Breakdown Structure (WBS).

<Create a WBS (approximate) for your project.>

## Develop an Activity Graph.

< Using this WBS develop an Activity Graph. Convert the graph such that it is helpful in estimating project completion.>

## Find the Critical Path.

< Determine the shortest Path(s) required to complete the project and give the detailed path for each of them.>

## Create a Gantt chart.

< Create a **Gantt chart** showing the progress of your project an indicating the current status of your project.>

# ~~~~~~~~~~ Phase II ~~~~~~~~~~

# Functional Requirements

<All functional requirements are expressed as use-cases. Fill out the following template for each use-case. Don’t really say “Use-Case 1.” State the use-case name in just a few words e.g. “Withdraw Cash from ATM”. A use-case may have multiple alternate courses of action.>

<Provide a Use Case Diagram before describing the use cases.>

## Name of Use-Case 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-1 | |
| **Purpose** | | … | |
| **Priority** | | <Choose one from {High, Medium, Low}> | |
| **Pre-conditions** | | … | |
| **Post-conditions** | | … | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** |  | |  |
| **2** |  | |  |
| **3** |  | |  |
| **…** |  | |  |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** |  | |  |
| **2** |  | |  |
| **3** |  | |  |
| **…** |  | |  |

Table 1: UC-1

## Name of Use-Case 2 (and so on)

\*if you are able to discover 15+ use cases describe at least 3 of them minimum

## Requirements Analysis and Modeling

<Include the following analysis models: **use-case diagram,** **state diagram**, **decision table**, event table etc.>

# Nonfunctional Requirements\*

**\*make sure all your requirements functional or non functional are testable**

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## 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.>

## Additional 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.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. These might include database requirements, external (hardware, software, or communication) interface requirements, internationalization requirements, legal requirements, and reuse objectives for the project.>

# Designing

## Complete class diagram

*<including class attributes, methods and multiplicity is required>*

## Complete Data Flow Diagram (DFD)

*< Context level DFD and level 0 DFD are required >*

## Complete ER Diagram

*< Complete ERD is required >*

## Physical design of your database

<i.e. table names with attributes and their types>

## Information on use of design patterns while designing the modules

<e.g. while making class diagram. Highlight/Mention the relevant classes>

## Make a Sequence and a Collaboration diagram of following.

### Scenario 1

### Scenario 2

# Estimation

## Cost Benefit Estimation

### Return of Investment (ROI) {1st year}

*<Approximate estimations should be done after approximate estimation of values.*

*Formula of ROI = Benefits / Cost >*

### %gain on ROI {1st year}

*< Values should be based on estimates of 8.1.1.*

*Formula of %gain on ROI = { (Benefits - Cost) / Cost } x 100>*

### Payback Period in years

*<Values should be based on estimates of 8.1.1.*

*Formula of payback period = Cost / Benefit >*

## FP based Estimation

<Approximate estimation should be done using FP >

## COCOMO Estimation

<Approximate estimation should be done using FP based COCOMO >

# References

<List all books, conference papers, journal articles, websites, etc. used in preparing the content of this SRS. Provide enough information so that the reader could access a copy of each reference, including title, author, volume/edition number, page number(s), and publication year. Mention complete URLs for websites.>

< Must use Zotero reference Manager to maintain references in every phase in this section.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>