BSCS FINAL PROJECT

<Software Design 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 Design Specification

SDP Phase II

<Project Name>

Advisor: <Advisor Name>

Team <Team #>

|  |  |
| --- | --- |
| Member Name | Primary Responsibility |
|  |  |
|  |  |
|  |  |
|  |  |

Table of Contents

Table of Contents i

Revision History ii

Abstract iii

1. Introduction 1

1.1 Product 1

1.2 Background 1

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

1.4 Scope 1

1.5 Business Goals 1

1.6 Document Conventions 1

1.7 Miscellaneous 1

2. Overall Description 2

2.1 Product Features 2

2.2 Functional Description 2

2.3 User Classes and Characteristics 2

2.4 Design and Implementation Constraints 2

2.5 Assumptions and Dependencies 2

3. Technical Architecture 3

3.1 Application and Data Architecture 3

3.2 Component Interactions and Collaborations 3

3.3 Design Reuse and Design Patterns 4

3.4 Technology Architecture 4

4. Screenshots/Prototype 5

4.1 Workflow 5

4.2 Screens 5

4.3 Additional Information 5

5. Other Design Details 5

6. Revised Project Plan 5

7. References 5

Appendix A: Glossary 6

Appendix B: IV & V Report 7

Revision History

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

# Abstract

Abstract gives the summary of your project. You should focus on the problem description, significance of the problem, knowledge areas to be used and the results to be acquired. Make sure it does not turn out to be an introduction to the introduction / background but summarizes the whole project.

# Introduction

## Product

<Provide a short description of the software being specified. State the problem being solved. Also state whether the end product will be a program that applies certain algorithms to some application, a tool that is end product of a research, a dataset, a simulator developed as a result of research, an animated movie developed using graphics techniques, a system that achieves better performance than its competitors, or a software package that is useful in certain application etc.>

## Background

<Describe the previous work done in particular domain. You should make sure to relate the previous work to your project. Compare your application/work with others and differentiate your work from others’. Also, include the related projects from the list of previous projects provided by the Project Office.>

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

< Objectives are the final results to be achieved after the completion of your project. The objectives must be explicitly stated. The objectives should be achievable in the stipulated time period. Do not be too ambitious and, at the same time, must conform to your program level. These shall not be too many. Research objectives tell what will be the contribution of your project in the area you are working on?>

## Scope

< Describe the scope of the product that is covered.>

## Business Goals

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

## 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.>

## Miscellaneous

< Provide any other information, which you feel, will be helpful in organizing and conduct of the project work.>

# Overall Description

## Product Features

<Summarize the major features the product contains or the significant functions that it performs or lets the user perform. Details have been provided in phase 1 of SDP, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the document.>

## Functional Description

< Provide a brief description of the functionality of the current system. If available, provide a high level context diagram for the current system. >

## User Classes and Characteristics

<Describe the various user classes that you have identified. 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.>

## 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).>

# Technical Architecture

< Identify and describe the technical architecture of the current system. Include a high-level architecture diagram that highlights major subsystems and components.>

<This section should answer the questions similar to the following:

* Is the system custom-built? COTS?
* What type of processing is the current system responsible for?
* Batch and/or online
* Transaction processing and/or analytical reporting
* What are the major application components?
* What data does the current system collect and manage?
* What is the basic application architecture (layered, client/server, etc.)?
* What programming language is the current system built in?
* What is the hardware platform that supports the current system?
* What database platform supports the current system?
* Does the system have an end-user interface? If so, what type of user interface? (e.g., browser based, thick client)?
* What is the basic network architecture (e.g., available on LAN, WAN, Internet)?
* Where is the system hosted (e.g., Enterprise Data Center, Other CMS Data Center, External Data Center)?

>

<You do not need to answer these questions one by one. Answers of these questions are expected in this document. You should provide different design models/diagrams/tabular structures in this document to answer these or similar questions.

In the coming subsections, this section should:

* Identify application components, data components, and interfacing systems
* Illustrate the collaboration and interaction between the major components
* Identify any relevant design patterns or reuse relevant to the design
* Identify tools and technologies used

>

## Application and Data Architecture

*<Describe the application components and/or processing units in the architecture diagram above. Complete logical or physical model is expected. Diagrams/Tools that may be provided in this section include Component Diagram, ER Diagram, Class Diagram (with complete inheritance, composition, and association details), Activity Diagram, Decision Table etc. All these diagrams should have more details than the details provided in Phase 1 of the SDP. Research based projects may provide complete design of the proposed system. Describe each diagram briefly.>*

## Component Interactions and Collaborations

*<Provide interactions and collaborations between your system components/processing units. Diagrams/Tools that may be provided in this section include Design Level Sequence Diagram, Collaboration Diagram, Event Traces, Detailed DFD, Activity Diagram etc. All these diagrams should have more details than the details provided in Phase 1 of the SDP. Describe each diagram briefly.>*

## Design Reuse and Design Patterns

*<Identify and state any reuse during development of the system>*

## Technology Architecture

<Describe the anticipated infrastructure that will be required to support the application and information architecture. At this point in the project, focus on describing the technology architecture at a high level only. In this section provide details regarding platform, system hosting, connectivity requirements, modes of operations etc.>

# Screenshots/Prototype

## Workflow

<Describe complete workflow of your system. Swim-lane diagram may be used.>

## Screens

< Include all screenshots of your software application’s graphical user interface.>

## Additional Information

<Any additional information>

# Other Design Details

<Describe any design details not covered in previous sections. Add subsections as required. There can be details regarding Research Oriented, Game Oriented, or Hardware based projects that have not been covered in this document before, those details can be provided in this section. For example research based projects may use this section to present their results and analysis; hardware based projects may use this section to describe interface dependencies and issues etc.>

# Revised Project Plan

<Show your progress and provide current status of the project in accordance with the plan provided in project proposal. Gantt chart should be used in this regard. Use Microsoft Office to develop the Gantt chart. Also provide an updated project plan.>

# 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.>

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.>

Appendix B: IV & V Report

**(Independent verification & validation)**

**IV & V Resource**

Name Signature

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S#** | **Defect Description** | **Origin Stage** | **Status** | **Fix Time** | |
| **Hours** | **Minutes** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| … |  |  |  |  |  |

**Table 1: List of non-trivial defects**

This document has been adapted from the following:

1. Previous project templates at UCP
2. High-level Technical Design, Centers for Medicare & Medicaid Services. (www.cms.gov)