**Project Title Name**

Team Name:

Member Names:

CSC480A Computer Science Capstone Project I

Project Proposal

National University, San Diego, CA

mm-yyyy Term

Instructor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Contents

[1.0 Introduction 3](#_Toc14785610)

[2.0 Background and the Need for the Project 3](#_Toc14785611)

[3.0 Project Objectives and Scope 3](#_Toc14785612)

[4.0 Customers and Stakeholders 3](#_Toc14785613)

[5.0 Project Requirements 3](#_Toc14785614)

[5.1 Functional Requirements 4](#_Toc14785615)

[5.2 Non-Functional Requirements 4](#_Toc14785616)

[5.3 High Level Use Case Scenarios 4](#_Toc14785617)

[6.0 Project Assumptions and Constraints 4](#_Toc14785618)

[7.0 Project Delivery Methods 4](#_Toc14785619)

[8.0 Project Issues and Risks 4](#_Toc14785620)

[9.0 Evaluation, Selection of Technology and Tools for Project Implementation. 5](#_Toc14785621)

[10. Prioritized Feature List and estimates 5](#_Toc14785622)

[11.0 Initial Architecture Design 5](#_Toc14785623)

[12.0 Agile Team Formation, Responsibilities and Implementation Iterations (schedule). 5](#_Toc14785624)

[12.1. Team Responsibilities: 5](#_Toc14785625)

[12.2 Iteration Plan 5](#_Toc14785626)

[13.0 Conclusion 6](#_Toc14785627)

[References 6](#_Toc14785628)

[Appendix 6](#_Toc14785629)

[Attachments, if any. 6](#_Toc14785630)

[List of Abbreviations 6](#_Toc14785631)

[Definitions 6](#_Toc14785632)

NOTES: – ALL TEXT HIGHLIGHTED ARE INSRUCTOR’S NOTES TO HELP GUIDE THE STUDENTS.

DURING EACH WEEK’S SUBMISSION, STUDENTS SHOULD REMOVE ALL THE HIGHLIGHTED TEXT INCLUDING THIS TEXT.

# Introduction

Introduce the project along with its importance in today’s business environment.

# Background and the Need for the Project

Briefly describe some background and justify the need for the project.

# Project Objectives and Scope

With a few introductory sentences, list the project objectives to be achieved – as seen from the customer’s perspective.

For scope, clearly indicate the boundaries or framework and indicate what is not included that the customer or the implementors would like to know.

# Customers and Stakeholders

Clearly identify the customer(s) and all other stakeholders involved in the project including marketing, sales, agile team, etc.

(Instructor’s Note – All the above four sections are due at the end of the first week – will be a first pass. Can be updated during subsequent submissions).

# **Project Requirements**

For each of the major objective above, based on the customer need, list the performance expected from the customer and these will drive the architecture, platform and other design criteria. There may be other performance requirements not related to the objective. Break the requirements as functional (features from customer/owner) and non-functional. Suggest creating a table with list of the objectives from 3.0 above and the expected performance for each one. There could other indirect performance expectations too – e.g. downtime and recover timings; reports, metrics, etc.

Performance expectations also become criteria for testing.

High level use case scenarios

Break down the requirements in to three subsections as shown below.

## Functional Requirements

## Non-Functional Requirements

## 5.3 High Level Use Case Scenarios

# 6.0 **Project Assumptions and Constraints**

List all the project assumptions as you start the project

Constraints could be budget, time, staff, technology, etc.

List any project priorities as a part of this section

# Project Delivery Methods

Clearly state how the customer is expecting the system to be delivered – website, mobile or both

What are requirements from the customer’s perspective required on their system to run, if any. This would apply to enterprise systems.

# Project Issues and Risks

List all the items that are obstacles now (issues) and in the future (risks) that would prevent the team from accomplishing their project objectives. Categorize the risks as low, medium and high on their impact to the project to complete.

Also, propose mitigations for the risks.

Simple and effective approach for this is to create a table (see sample below). Add additional columns as needed (e.g. due date) for tracking during implementation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Issue or Risk Brief Detail | Risk Probability:  High, Medium, low | Mitigation Plan | Responsibility | Status: Open, WIP, closed |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

(Instructor’s Note – all above 8 sections are due at the end of second week).

# 9.0 Evaluation, Selection of Technology and Tools for Project Implementation.

The teams would need to do some research for the appropriate technology and tools for the project. Implementation/development requirements: Software, hardware, network, database, platform, etc. Although the goal is not to do any implementation in this first course, but identification of technology and tools are needed at this stage.

# 10. Product Backlog: Prioritized Feature List and Effort Estimates (hrs.)

Breakdown the project main requirements in to prioritized feature list for implementation by the Agile team with estimates for each. This would be the first pass at this stage.

# 11.0 Initial Architecture Design

This will be the initial design aspect when the team designs the high-level architecture using Object Oriented design principles: students should identify the top-level subsystem and components and how they are interconnected. UML activity and class diagrams, objects, etc.

Include a brief description of why the team chose this approach.

(Instructor’s Note – all the above 11 sections are due at the end of 3rd week).

# 12.0 Agile Team Formation, Responsibilities and Implementation Iterations (schedule).

12.1. Team Responsibilities: Individual team member assignment will be listed per the Agile process requirements. Project Manager/Scrum Master, Architect, coder, tester, etc. In a small team, one might wear more than one hat. But the responsibilities must be clear. Also, the specific times for meeting each day (as required by the Agile team – need to be agreed up at this stage which guarantees everyone’s participation and contribution. The project manage/scrum master should be given authority to call upon other as required to help meet the iteration goals.

12.2 Iteration Plan: The team should decide the implementation of the prioritized feature list within the 4 iterations for the rest of the eight weeks with each iteration being 2 weeks.

The team will list here the feature(s) that will be completed and working in each iterations and demonstrable. Again, simply create a table for each iterations.

Part of the last iteration will include a final project presentation during the last session of the class (CSC480C) to the Chair and other School Faculty and submitting a final project documentation as required by the Instructor.

# 13.0 Conclusion

Brief summary of this project proposal and the plan to implement it using Agile methodology in iterations/sprints.

# References

# Appendix 1 - Proposal Plan Responsibility List

|  |  |  |  |
| --- | --- | --- | --- |
|  | **CSC480A Project I - Proposal Plan and Responsibility List** |  |  |
|  |  |  |  |
| **Project Section** | **Section Title** | **Author(s)** | **Reviewer(s)** |
| **1** | Introduction |  |  |
| **2** | Background and Need for the Project |  |  |
| **3** | Project Objectives and Scope |  |  |
| **4** | Customers and Stakeholders |  |  |
| **5** | Project Requirements |  |  |
| **6** | Project Assumptions and Constraints |  |  |
| **7** | Project Delivery Methods |  |  |
| **8** | Project Issues and Risks |  |  |
| **9** | Evaluation, Selection of Technology and Tools for Project Implementation |  |  |
| **10** | Prioritized Feature List and Estimates |  |  |
| **11** | Initial Architecture Design |  |  |
| **12** | Agile Team Formation, Responsibilities and Implementation Iterations (Schedule) |  |  |
| **13** | Conclusion |  |  |
|  |  |  |  |

# Appendix X

# Attachments, if any.

# List of Abbreviations

# Definitions

(Instructor’s Note – the full document is due at the end of 4 weeks).