|  |  |
| --- | --- |
|  | **Qatar University**  **College of Engineering**  **Department of Computer Science and Engineering** |

Senior Project Report

**<ToDo: Insert Project Title />**

**Project Group Members:**

<ToDo: StudentName1 (StudentId1)

StudentName2 (StudentId2)

StudentName3 (StudentId3) />

**Supervisor**: <ToDo: SupervisorName />

**Co-Supervisor**: <Remove if not applicable: ? />

**20??**

This project report is submitted to the Department of Computer Science and Engineering of Qatar University in partial fulfillment of the requirements of the Senior Project course.

# Declaration

This report has not been submitted for any other degree at this or any other University. It is solely our work except where cited in the text or the Acknowledgements page. It describes work carried out by us for the senior project. We are aware of the university policy on plagiarism and the associated penalties and we declare that this report is the product of our own work.

< ToDo: For each student add the signature and the date of signing the declaration />

Student: Date:

Signature:

Student: Date:

Signature:

Student: Date:

Signature:

# Abstract

< ToDo: The abstract is a brief overview of your project and its objectives. It should present an accurate summary of the problem your project has addressed and a summary of your solution. Highlight key achievements and most important conclusions. The length of your abstract should not exceed 500 words.

/>

<ToDo – **Writing the final report**

To produce the final report, you can follow the following recommended steps:

* Where appropriate, copy material from your interim report into the final report. Go through that material, and update it based on changes that have occurred in your project between last semester and now.
* Revise the Abstract and enhance it by adding the project’s key achievements and most important conclusions. The last paragraph should highlight the novelty of your design (e.g., what makes your design unique and what are the impacts of your engineered solution, etc.).
* Fill in all of the appropriate material required for the final report.
* Update the Table of Content, the List of Figures and the List of Tables.
* Review the whole document to make sure that it is coherent and to ensure that it addresses all the requirements listed in the Project Guide and the Project Grading Rubrics. Also make sure that the tense used is the present tense and the past and not the future (e.g., avoid ‘we will’ or ‘system should’ and report what has been done).
* Seek your supervisor’s feedback and address any issues raised.
* Note that the template is only provided as a guide. In consultation with your supervisor, you can add other sections to align it with the nature of your project.
* If you are using Word ‘Track Changes’ you must accept all the changes before submitting the softcopy of your report.

/>

# Acknowledgment

< ToDo: Acknowledge any assistance you received for your project. />

Table of Contents

[Declaration ii](#_Toc525597397)

[Abstract iii](#_Toc525597398)

[Acknowledgment iv](#_Toc525597399)

[List of Figures vii](#_Toc525597400)

[List of tables vii](#_Toc525597401)

[1. Introduction and Motivation 8](#_Toc525597402)

[1.1. Problem statement 8](#_Toc525597403)

[1.2. Project significance 8](#_Toc525597404)

[1.3. Project objectives 8](#_Toc525597405)

[2. Background and related work 8](#_Toc525597406)

[2.1. Background 8](#_Toc525597407)

[2.2. Related work 8](#_Toc525597408)

[3. Requirements analysis 9](#_Toc525597409)

[3.1. Software development process 9](#_Toc525597410)

[3.2. Functional requirements 9](#_Toc525597411)

[3.3. Non-functional requirements 9](#_Toc525597412)

[3.4. Assumptions 10](#_Toc525597413)

[3.5. Ethics 10](#_Toc525597414)

[4. Solution Design 11](#_Toc525597415)

[4.1. Overview 11](#_Toc525597416)

[4.2. Architecture Diagram 11](#_Toc525597417)

[4.3. Structural model 12](#_Toc525597418)

[4.4. Behavioral model 12](#_Toc525597419)

[4.5. Database design 12](#_Toc525597420)

[4.6. User interface design 12](#_Toc525597421)

[4.7. Design patterns 13](#_Toc525597422)

[4.8. Hardware/software used 13](#_Toc525597423)

[5. Implementation 14](#_Toc525597424)

[6. Testing 15](#_Toc525597425)

[7. Impact of your project solution 16](#_Toc525597426)

[8. Conclusion 17](#_Toc525597427)

[9. Future work 18](#_Toc525597428)

[10. Student reflections 19](#_Toc525597429)

[References 20](#_Toc525597430)

[Appendix A – Project Plan 21](#_Toc525597431)

[A.1. Project milestones 21](#_Toc525597432)

[A.2. Project timeline 21](#_Toc525597433)

[A.3. Anticipated risks 21](#_Toc525597434)

[Other Appendices 22](#_Toc525597435)

# List of Figures

[Figure 1. Use cases diagram 9](#_Toc525597436)

# List of tables

[Table 1. Use cases summary 9](#_Toc525597437)

# Introduction and Motivation

<ToDo:

The introduction is where you introduce the reader to the problem your project is trying to solve, why is it important, why is it challenging, and briefly summarize your solution.

/>

## Problem statement

< ToDo: Present a summary of the problem you intend to solve. You should also describe what you believe the technical and non-technical challenges of the project. />

## Project significance

< ToDo: Discuss the importance of your project. What is the impact of not having a solution to the problem being addressed? How the outcomes of this project will benefit the project stakeholders? What are the anticipated benefits of the project in Qatar and the region? You can cite relevant references that highlight the significance of the problem your project is trying to address.

Also discuss, why do you want to do this project? What got you interested in it? And how this project will help you further your career goals? />

## Project objectives

< ToDo: Define clear project objectives, desired results and specific deliverables. The objectives should be made very specific and expressed in terms of desired results against which the success of the project will be judged. />

# Background and related work

## Background

< ToDo: Discuss briefly the major concepts, issues and key problems related to your project. This should give the reader the necessary background information to better understand your project. />

## Related work

<ToDo: Discuss the related work and how others researchers or developers have dealt with the problem at hand. Cite appropriate references including relevant research papers. A list of 10 to 15 references should be adequate to include in this section. Seek your supervisor assistance to find and select relevant related work.

At the end of this section, you should highlight how your project is different than previous approaches. You may use a table to summarize the key similarities and differences between your project and the related work. />

# Requirements analysis

< ToDo: In this section you are describing and justifying your selection of the most appropriate software development process for your specific project and presenting the project requirements. />

## Software development process

<ToDo: Select an appropriate software development process (life cycle model) for your project. You MUST provide sufficient justifications why the selected software development process is most suitable for your specific project. Note that once a software development process is selected, you must show detailed evidence in this report that you have used this process throughout your project from the requirements analysis phase to testing. />

## Functional requirements

< ToDo: Collect, document and analyze the project requirements. The requirements specification should be done using suitable methods and tools such as use cases. You may use flow-charts to describe the specific logic used in various components of your system.

Once you identify and describe all the use cases, you need to prioritize them and decide your project scope by highlighting the use cases that you will design and implement. />

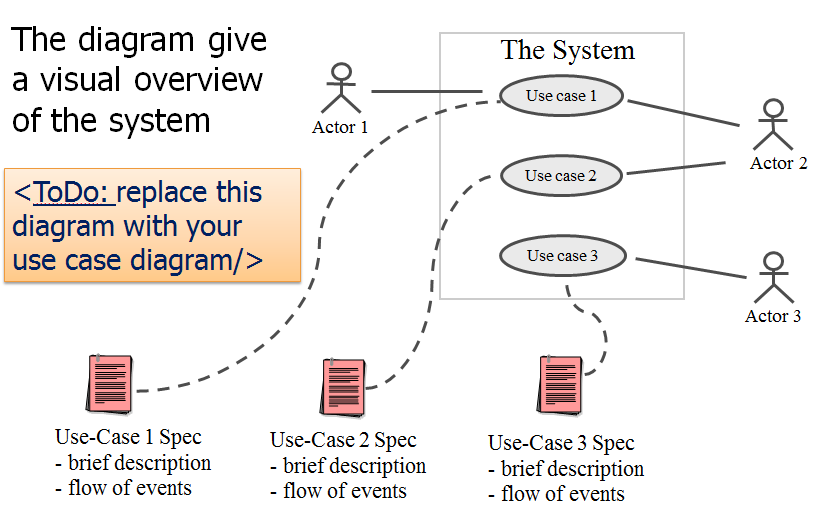


Figure 1. Use cases diagram

Table 1. Use cases summary

|  |  |
| --- | --- |
| Use case | Brief description |
|  |  |
|  |  |

## Non-functional requirements

<ToDo: Identify non-functional requirements (i.e., desired quality attributes) and clearly document them using measurable scenarios such that you can later test whether they are met. E.g., instead of just saying the system should be fast, you should be more specific e.g., ‘Registering a new user should take less than 20 seconds, 95% of the time’. Instead of just stating that the system should be secure, you should specify what attacks must be prevented and what threats it must defend against. You should specify measurable targets for relevant qualities such as availability, performance, portability, reliability, safety, security, usability, etc.

Remember: If your non-functional requirement isn’t specific and measurable, then it isn’t a good non-functional requirement.

/>

## Assumptions

<ToDo: List any specific project assumptions. Assumptions are potential failure points in a project. They need to be monitored and managed. A bulleted list of items is the best format to use for this section. />

## Ethics

<ToDo:

Identify ethical issues/concerns and professional responsibilities **relevant** to your project (e.g. proper handling of private information collected during the project, not to exaggerate the nature of work in the report rather the report should truly reflect the actual work completed, not to use someone’s work without citing proper reference, etc.). You need to discuss how you will address them during the design, implementation, testing and delivery of your project. For further guidance, you may refer to and apply professional code of ethics such as ACM/IEEE Software Engineering Code of Ethics and Professional Practice <https://ethics.acm.org/code-of-ethics/software-engineering-code/> .

/>

# Solution Design

<ToDo:

In this section you need to document the detailed design specifications of the software subsystems used in your design to meet the functional and the non-functional requirements of your project. You should document your detailed design using suitable methods and tools such as class diagrams, sequence diagrams, activity diagrams and state diagrams. Also, you need to provide detailed justification of design choices. Additionally, you should highlight the novel aspects of your design.

Note that for every design aspects you will present in the sections below, **you must include**:

* **Identify and evaluate possible alternative solutions** (i.e., design choices) and analyze their tradeoffs. You must provide sufficient justifications on why you have preferred and selected a particular solution or design choice over others.
* Justify your optimal design choice by **discussing how your design choices** enables you to achieve the desired system quality attributes documented in Section ‎3.2. Non-Functional Requirements.
* Further justify your design choice by **discussing relevant design principles** that influenced your system design. For example, highlight how your design comply with key software design principles such as:
  + Separation of concerns: e.g., separation of components from connectors; separation of the UI from the business logic.
  + Abstraction: hide the component’s complexity behind simple interface
  + Modularity: divide the system into components
  + High cohesion: apply cohesive responsibility principle so that the component’s functions should be functionally related.
  + Low coupling: reduce dependencies between components

You could consider breaking the sub-sections listed below to clearly document and communicate your design.

/>

## Overview

< ToDo:

* Present an overview of the proposed solution. This can be done in the form of a simple one or two paragraphs giving a birds-eye view of the solution. The description should be kept very generic so that even a layman will be able to comprehend the scope of your work. />

## Architecture Diagram

< ToDo:

* Add a high level architecture diagram of the proposed solution. The diagram should show how your solutions is decomposed and organized into components. Describe the role and the interfaces of key components of your high level architecture.
* Discuss and justify the architectural style (e.g., MVC, SPA) used by your solution.
* Discuss the key interactions between the components identified in the high-level architecture diagram.

/>

## Structural model

<ToDo:

You may break this section into subsections to document the structural model of each major module or use case.

The structure of your software components could be documented using:

* Class diagram for the whole system. If the model is too big partition the diagram using some reasonable criteria. For example, you may provide the entity classes and the controller classes as separate diagrams.
  + For every class, specify the declaration of all attributes (names, data types).
  + For every class, specify the signatures for all methods (i.e., parameter names and their types as well as the return type).
  + Show associations and aggregations between classes. You should define the association name and the multiplicities on both ends.
  + Show inheritance relationships between classes
  + A brief explanation should accompany each diagram. Add few paragraphs discussing major design decisions such as how inheritance and polymorphism were used to improve the system design. Also briefly discuss how design principles such as the ‘Information Expert’ and ‘Whole controls the Parts’ principles were applied to your design.
* For the external services (i.e., boundary classes) the system interacts with (e.g., BankingService, MailService) you should design the interface exposed by each of these services.

/>

## Behavioral model

<ToDo:

You may break this section into subsections to document the behavioral model of each major module or use case.

The behavioral aspects of your software components could be documented using:

* Activity diagrams to describe the overall flow of control of your application (step-by-step workflows of components in your system).
* Design Sequence Diagrams (DSD) to document how to realize key use cases. Method calls in the DSD should be numbered and you show the parameters and the return type of each method.
* State diagrams for important parts of your system. You need to identify states and events that trigger state transitions.

/>

## Database design

<ToDo:

* Include a logical data model describing the major entities and the relationships between them. Also include the attributes for each entity. Make sure that the data model is consistent with (or derived from) the entities of the class diagram.

/>

## User interface design

<ToDo:

* Include screen mockups for major use cases.
* Discuss the navigation design of the UI (e.g., movement between screens to achieve a particular use case).
* Discuss any novel aspects of your UI design and how they enhance the usability of your system.

/>

## Design patterns

<ToDo:

Note that this subsection is required.

* Document and evaluate the design patterns applied to your design such as the Model View Controller (MVC) pattern, the Factory pattern, the Proxy pattern, the Adapter etc.
* For each selected architectural pattern:
* Discuss the design rationale justifying the choice of the selected design pattern.
* Draw a UML diagram(s), e.g., class diagram, sequence diagram, to illustrate how the selected design pattern have been applied to your design.
* Evaluate of the used pattern the effect of selected pattern on your system quality attributes. Highlight the benefits introduced by the selected pattern potential problems or limitations introduced by the selected pattern.

/>

## Hardware/software used

<ToDo:

Note that this subsection is required.  
Discuss the hardware/software platforms and tools to be used for the design and implementation and justify your choices. Also list the hardware/software platform(s) on which the system will run. />

# Implementation

<ToDo:

* List the different tools and framework used for the implementation.
* Discuss any novel aspects of your implementation (if applicable). You may link this discussion to the comparison table at the end of literature review and elaborate on the steps taken in achieving these novelties in your design.
* Discuss the challenges encountered during the implementation and how they were addressed.

You may organize any of the above recommended points as subsections.

/>

# Testing

<ToDo:

In this section you should **describe in detail the tests** you have conducted to verify that your design and implementation satisfies the desired functional and non-functional requirements. The testing should verify and provide evidence that your solution solved the stated problem and satisfied the requirement specifications (if not explain what is lacking).

* Describe in detail the tests you have run to verify that your solution satisfies the functional requirements of your project. For example, if you used Use Cases to describe your functional requirements, then for each use case you should write a test case, run it and report the testing results (the key test cases can be added as an Appendix). Functional testing will allow you to find defects then fix them and also identify possible improvements. You need to have a comprehensive set of tests that verifies correct functionality of every component of your system.
* Describe in detail the tests (e.g., performance testing, usability testing, etc.) you have run to verify that your solution satisfies the non-functional requirements of your project. Include a summary table that shows each non-functional requirement and the measurements that prove your system met/did-not-meet the requirement. In case a requirement is not met, then explain the reason for that.

You should **present the test results**, with appropriate level of details in addition to accuracy and completeness**,** using tables, graphs, diagrams, screen shots etc. Additionally, **discuss these results** and explain whether the prototype has achieved the requirements. If not state what is lacking or still need improvement then explain the reason for that. There will probably be multiple subsections under this section to describe each system test and its result.

Note that this section is a substantial portion of the grade for your final report, and will require a significant amount of effort.

/>

# Impact of your project solution

<ToDo:

Identify and discuss the local and global impacts of your project solution on individuals, organizations, and society

/>

# Conclusion

< ToDo:

* Discuss the main conclusions (e.g., match the project objectives with the achievements in your work and state the degree of achievement).
* Highlight the strengths of the solution and list down its shortcomings (what worked? what didn’t work?).
* Highlight the key contributions and the novel aspects of your work.

/>

# Future work

< ToDo:

Suggested improvements and further work: identify areas of improvement in the project and features of interest that can be added later on. How the solution shortcomings could be addressed? What things could be done better? What additional resources are required to implement the extended / not-implemented design or features?

/>

# Student reflections

< ToDo:

Add individual student **reflections** (**add a sub-section for each student**):

* Lessons learned from the project.
* Professional development you have achieved during your project experience (i.e., new skills gained) and explain its value for your future career.

You can discuss (1) new technical skills acquired such as solving problems, designing and realizing solutions (2) interpersonal skills such as team leading and effective communication (3) personal growth such as adapting to change and acting professionally and ethically.

* Key shortcomings that you should avoid in future projects.
* Key lessons and new attitudes to carry forwards to your professional life from the personal experiences and the team work experienced during the project.

/>

# References

< ToDo: Include the list of references relevant to your work using IEEE citation style. Use research papers and good quality references (such as journal and conference papers from ACM, IEEE and Springer). Do not just use web links. Embed citations of the references in the text of your document.

For further details refer to http://libraries.dal.ca/content/dam/dalhousie/pdf/library/Style\_Guides/IEEE\_Citation\_Style\_Guide.pdf

You can get the PDF documents of good quality papers related to your work from IEEE, ACM and Springer via QU Journal Databases @ http://library.qu.edu.qa/index.php/en/#tab-2

A better way is to search via [http://scholar.google.com](http://scholar.google.com/) (if you do it in QU Campus you can easily get the paper PDF for free).

Remember, always seek your supervisor’s advice and feedback.

/>

# Appendix A – Project Plan

## A.1. Project milestones

< ToDo: List and explain the major milestones of the plan used to deliver the project. Milestones are checkpoints which have specific deliverables to be produced.

It is highly encouraged to adopt an incremental and iterative approach to deliver your project. Each iteration/milestone involves choosing a use case (or a component from your high-level design) and quickly designing, implementing and testing its realization. This way, you can get an early prototype that grows incrementally and converges towards the desired system. You can demo your early prototype to your supervisor to get valuable feedback that can be considered for subsequent iterations.

For the most part, this is taken directly from your interim report, but you should update the plan to reflect what actually happened. />

## A.2. Project timeline

< ToDo: Project timeline defines who will do what and when. You need to:

* Breakdown each project milestone into manageable tasks. Using a table, provide a brief description of each task.
* Estimate the time required to complete each task.
* Allocate the tasks to individual team members and define a work schedule stating the estimated beginning and completion dates of each task. You can use Microsoft Project to create your project timeline. Insert a Gantt chart and/or a project timeline table in this section.

Keep in mind the total number of days budgeted for the project. Note that the role and the responsibilities of each team member should be clearly decided.

Be sure to include time to evaluate your work, to reflect on the experience and to document your experience in intermediate reports at each milestone which will accumulate to form your final report. You should include time to write and review the final report and the final presentation. />

## A.3. Anticipated risks

< ToDo: Present a table of risks highlighting the potential events that might result in failure of successfully completing the project. What is your approach to minimize each risk? />

# Other Appendices

< ToDo:

The following are possible, additional appendices you may add to the project report:

* 1. Detailed use-cases
  2. Testing plan including key test cases
  3. Any questionnaires, interview questions, etc. used in your project.
  4. Any other appendices may be included to provide supporting details that could aid in the understanding this report.

/>