**King Saud University**

**Collage of Computer and Information Sciences**

**Computer Science Department**

**LOGO**

**“Project Title”**

*CSC 49X – Final Report*

**Prepared by:**

Student 1 Name Student 1 ID

Student 2 Name Student 2 ID

Student 3 Name Student 3 ID

Student 4 Name Student 4 ID

**Supervised by:**

Dr. Name

Software project for the degree of Bachelor in Computer Science

First/Second Semester 14##/14##

Autumn/Spring 20##

# Acknowledgements

# English Abstract \*

The abstract should complement your proposal abstract and might include additions that do not contradict the previously proposed abstract.

\* These sections should be modified based on the phase of the graduation project.

\*\* These sections are to be included in the final report of CSC497

# Arabic Abstract \*

Your abstract translated to Arabic.

Table of Contents

I. Acknowledgements 2

II. English Abstract \* 2

III. Arabic Abstract \* 2

Chapter 1: Introduction 4

1.1 Problem Statement 4

1.2 Goals and Objectives 4

1.3 Solution 4

1.4 Research Scope 4

Chapter 2: Background 5

Chapter 3: Literature Review 5

Chapter 4: System Analysis \* 5

4.1 Functional Requirements 5

4.2 Non-Functional Requirements 5

Chapter 5: System Design \* 5

5.1 System use-cases: 5

5.2 Interaction Diagrams: 5

5.3 Class Diagram: 6

5.4 System Architecture: 6

5.5 Database Design: 6

5.6 User Interface Prototype: 6

5.7 Algorithms: 6

Chapter 6: System Implementation\*\* 6

Chapter 7: System Testing \*\* 6

7.1 Unit testing 7

7.2 Integration and regression testing 7

7.3 Performance and stress testing 7

7.4 User acceptance testing 7

7.5 Test cases 7

Chapter 8: Conclusion \* 7

References 7

List of Tables

Table of Figures

# Chapter 1: Introduction

The introduction of the report should aim to catch the reader’s interest and should be written in a style that can be understood easily by any reader with a general computer science background. It gives an overview of the research project you propose and explains the background of the project, focusing briefly on the major issues of its knowledge domain. It then proceeds with the presentation of the project focus, which can take the form of a hypothesis, a research question, a project statement, or a goal statement.

## Problem Statement

This section states the problem that you are exploring and emphasizes the importance of your research. It should also include a clear description of the context of your research.

## Goals and Objectives

This section should include a precise description of the goal you are planning to achieve and how you will achieve it. It should explicitly emphasize the contribution you are planning to make with the intended study.

Goals describe what you want to achieve. Objectives describe how you are going to achieve those goals (what do you want to know, prove, demonstrate, analyze, test, investigate or examine?)

Objectives should be S.M.A.R.T.:

* **Specific** – be precise about what you are going to do.
* **Measureable** – specify an indicator for success, so that you will know when you have reached your goal
* **Achievable** –a less ambitious but completed objective is better than an over-ambitious one that you cannot possible achieve.
* **Realistic** – do you have the necessary resources to achieve the objective?
* **Time constrained** – determine when each stage needs to be completed.

## Solution

A brief description of how your software can contribute in solving the above stated problem.

## Research Scope

Define the boundaries of your project; describe precisely what is included in your project and what is not.

# Chapter 2: Background

This chapter should give an overview of the knowledge domain where your research takes place. You should define the notations you will be using later in your report, and introduce the user to any concepts, terms, algorithms, or tools you will be using in your project.

# Chapter 3: Literature Review

In this chapter, you should show familiarity with the literature. By exposing the reader to the relevant published work. The literature review should be comprehensive for the problem you are writing about. The purpose of this chapter is to put your work in context with others, and benefit from their experience. You may mention any changes that you have made to your original solution after you have conducted the literature review. Your conclusions should be included in a discussion section by the end of this chapter.

# Chapter 4: System Analysis \*

## Functional Requirements

List all user and system requirements for your software.

## Non-Functional Requirements

List all non-functional requirements for your software. (All requirements should be precise and measurable).

# Chapter 5: System Design \*

Give a clear description of your software design using sufficient diagrams/subsections from those mentioned below.

## System use-cases:

This section lists use cases or scenarios to represent some significant, central functionality of the final system.

## Interaction Diagrams:

In this section you show the interaction in terms of sequence or communication diagram between the objects/classes for different use case scenarios. You may ignore the trivial use cases. You can also show the interaction in case of exceptional flows in a use case. Exceptional flows include the following:

* Error handling. What should the system do if an error is encountered?
* Time-out handling. If the user does not reply within a certain period, the use case should take some special measures.
* Handling of erroneous input to the objects that participate in the use case (for example, incorrect user input).

## Class Diagram:

In this section all classes, their attributes and methods should be defined. It should also indicate the interaction and relationships among the defined classes.

## System Architecture:

Provide the software architecture diagrams and descriptions.

## Database Design:

This section should include a precise entity relationship diagram, and a corresponding schema describing the databases and file systems used in this project.

## User Interface Prototype:

Provide screen shots of the developed user interfaces, or mockups to illustrate the looks and feel of the system for critical scenarios.

## Algorithms:

Give brief overview of any special algorithms you will use (e.g. image processing, game strategy, scheduling etc.) And provide pseudo codes for the major functionalities in the system.

# Chapter 6: System Implementation\*\*

This chapter should provide an exhaustive explanation of the implementation stage of your project, review and explain all used technologies, describe the adopted integration process, and mention any limitations in the system (if any).

You should also give a clear detailed description of the software’s main interfaces and core logic. You can also include a walkthrough of the system by showing the sample interfaces to demonstrate other functionalities. Provide code snippet to show the main logic but do not give the whole code.

# Chapter 7: System Testing \*\*

This chapter should describe the test strategies and methodologies used to plan, organize, execute and manage the testing of your software project. Mention and explain any tools used for testing the software.

## Unit testing

Show the results of testing each component separately.

## Integration and regression testing

Describe how components where tested during the integration process, and report any issues or unexpected behavior and how it has been resolved.

## Performance and stress testing

Measure and report the performance of your software, and explain the behavior of your system under extreme cases.

## User acceptance testing

The purpose of acceptance testing is to confirm that the system is ready for operational use. During acceptance test, end-users (customers) of the system compare the system to its initial requirements. You should describe this process and may list people involved in testing, and the feedback you obtained from them.

## Test cases

The goal of any given test case or set of test cases is to detect defects in the system being tested. This section should provide descriptions of various test cases to test each component in your application with all the possible actions and input. Each test case should include a brief description of the sequence of events being tested, the test data, testing environment, expected results, actual results, and whether the software passed or failed that test case.

# Chapter 8: Conclusion \*

The final chapter of this report should provide a clear, insightful summary of your project, briefly mention the major findings/output, and emphasize the local and global impact of the project. Future directions may also be added.

# References

A bibliography of all cited works and sources you have used throughout this report. All references should follow IEEE format and should be ordered by their occurrence in the report.