**[Your Project Title]**

A Technical Report

Project -3-

Programming & Database Diploma

**Prepared by:**

[Student1 Name] [ Student1 ID]

[Student2 Name] [ Student2 ID]

**Supervisors:**

Principal Supervisor: [Title and name of your principal supervisor]



Department of Natural Science and Engineering

College of Applied Studies and Community Services

King Saud University

Riyadh, Saudi Arabia

First/Second Semester 14##/14##

Autumn/Spring 20##

Declaration

[I/we] declare that this report is [my/our] own work and that the work of others is

acknowledged and indicated by explicit references.

[Month Year]

[Student Names]

# Arabic Abstract

Your abstract translated to Arabic.

# English Abstract

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

Table of Contents

[Arabic Abstract ii](#_Toc140217)

[English Abstract ii](#_Toc140218)

[List of Figures iv](#_Toc140219)

[List of Tables v](#_Toc140220)

[Abbreviation vi](#_Toc140221)

[Chapter 1 Introduction 1](#_Toc140222)

[1.1 Problem Statement 1](#_Toc140223)

[1.2 Goals and Objectives 1](#_Toc140224)

[1.3 Solution 1](#_Toc140225)

[1.4 Project Scope 1](#_Toc140226)

[1.5 Report Organization 1](#_Toc140227)

[Chapter 2 System Requirements 1](#_Toc140228)

[2.1 System user characteristics 1](#_Toc140229)

[2.2 General system requirements 2](#_Toc140230)

[2.2.1 Functional requirements 2](#_Toc140231)

[2.2.2 Non-functional requirements 2](#_Toc140232)

[2.3 Policy and regulation requirements 2](#_Toc140233)

[2.4 Security requirements 2](#_Toc140234)

[2.5 Training requirements 2](#_Toc140235)

[2.6 Initial capacity requirements 2](#_Toc140236)

[Chapter 3 System Analysis 2](#_Toc140237)

[3.1 System Acceptance Criteria 2](#_Toc140238)

[3.2 System use-cases 2](#_Toc140239)

[3.3 Interaction Diagrams 3](#_Toc140240)

[3.4 Class Diagram 3](#_Toc140241)

[Chapter 4 System Design 3](#_Toc140242)

[4.1 System Architecture 3](#_Toc140243)

[4.2 Database Design 3](#_Toc140244)

[4.3 User Interface Prototype 3](#_Toc140245)

[4.4 Algorithms 3](#_Toc140246)

[Chapter 5 System Implementation 3](#_Toc140247)

[Chapter 6 System Testing 4](#_Toc140248)

[6.1 Unit testing 4](#_Toc140249)

[6.2 Integration and regression testing 4](#_Toc140250)

[6.3 Performance and stress testing 4](#_Toc140251)

[6.4 User acceptance testing 4](#_Toc140252)

[6.5 Test cases 4](#_Toc140253)

[Chapter 7 Conclusion 4](#_Toc140254)

[References 5](#_Toc140255)

[Appendices 6](#_Toc140256)

# List of Figures

# List of Tables

# Abbreviation

*Define of all terms and acronyms required to properly interpret the requirements contained within this document.*

# 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 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 project. It should also include a clear description of the context of your project.

## 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, analyse, 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.

## Project Scope

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

## Report Organization

# System Requirements

## System user characteristics

*Identify each type of user of the system by function, location, and type of device. Specify the number of users in each group and the nature of their use of the system.*

## General system requirements

### 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).

## Policy and regulation requirements

*Specify relevant applicable laws, regulations, policies, and standards that will affect the operation and performance of the system, as well as any relevant external regulatory requirements, or constraints imposed by normal business practices.*

## Security requirements

*Specify security requirement for users of the system.*

## Training requirements

*Specify Training requirements for the system.*

## Initial capacity requirements

*Specify the initial capacity requirements for the system. An initial estimation can be established using current data amounts, planned number of users, and estimated number of transactions.*

1. *Identifies the highest and lowest estimated number of transactions and processing frequency expected usage (including any seasonal peaks) for capacity planning for storage and memory requirements for the application or project. Identifies the highest and lowest estimated number of transactions and processing frequency expected usage (including any seasonal peaks) for capacity planning for storage and memory requirements for the application or project.*

# System Analysis

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

## System Acceptance Criteria

In this section specify the general system acceptance criteria specified and agreed upon by the project sponsor and key stakeholders that will be used to accept the final end product. Acceptance criteria could establish a boundary that helps team members to understand what is included and what is excluded from the scope of the user story.

## System use-cases

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

## 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 Design

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

## 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 mock-ups 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.

# 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.

# 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.

# 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 and main challenges 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.

# Appendices

Anything that is not essential for reading the report, but the reader might find useful. I can think of a few possibilities for suitable contents in project reports: software code listings; introductions to subjects that some readers will need, but others will not, if they have been drawn in a suitable computer-aided design (CAD) package, could usefully be included here; or perhaps a lot of raw test-result in list or graphical form which are discussed in more detail in the text. Again, the rule is that they should not be essential to the text: the reader should be able to read the entire document without feeling they are missing anything.

Appendices do not share the same numbering scheme as chapters. The first appendix is usually appendix A, then appendix B, etc; whereas chapters are more conventionally numbered.