

Iraqi Kurdistan Region
University of Salahaddin-Erbil
College of Engineering
Department of Software and Informatics



Project Title Here

A Project Submitted to the Department of Software and Informatics Engineering
University of Salahaddin-Erbil
In the Partial Fulfillment of the Requirement for the Degree of Bachelor of Science
in Software and Informatics Engineering

Prepared By:

Student Name 1

Student Name 2

...

Project Supervisor:

Supervisor Name

2018 - 2019

ABSTRACT

(Consists of one page)

Abstracts must show the project in summary statement including Purpose, Method, Findings, result and conclusions.

Purpose example: This project is aimed to measure the ease use of a network diagram software particularly to help users to create a network diagram for multiple reasons. This is to find out whether the software can save them more time and money by reducing the effort to manually graph out a network diagram. The aim of this project is also to find out if this software can help them learn more about the general aspects of computer networking. For example, the devices, the physical or logical aspects of a computer network, as well as network topologies.

Method example: A set of questionnaire will be made and then distributed to students of University Industry Selangor. Questions covered will be how fast they can easily learn the use of this software, can they understand most of its functions and whether from their experience, they can gain more knowledge of basic networking. Number of students will be from 10 to 30 and will be comprising of any students in Unisel. Results of the questionnaire will be used as data to form a statistical analysis of how fast the participants can learn how to use the software.

Findings example: Results can show whether this software is useful to help university students learn and gain more knowledge of computer networking. Thus whether the software should be used or implemented in the university.

Research Implication example: From this project, we may have a better understanding on how software introduced will easy for them to use and how

technology can help students learn faster. Maybe from this research, future education may include more of automated technology to make learning easier.

Conclusion example: By using this software, users can create a better looking diagram easily and even help them plan a good layout before implementing a real physical network.

DEDICATION

Our parents who have learned us the way of live, brothers, sisters. And our supervisor who tell us science with all other teachers, and our dear friends that help us in preparing this project, and those who want to learn.

I certify that the engineering project titled " " was done under my supervision at the department of Software and Informatics Engineering, College of Engineering, Salahaddin University –Erbil. In the partial fulfillment of ‘The requirement for the degree of Bachelor of Science in Software and Informatics Engineering’.

Supervisor

Signature:

Name:

Date: / /

LIST OF CONTENTS

1	Introduction	1
1.1	Background.....	1
1.2	Problem Statement.....	1
1.3	Aims and Objectives.....	2
1.4	Chapter Summery	2
2	Methodology	3
2.1	Software Model	3
2.1.1	Requirement gathering and Analysis	3
2.1.2	Software Design	4
2.1.3	Implementation	4
2.1.4	Testing.....	5
3	Analysis and Design.....	6
4	Results and descussions	7
5	Conclusion and Recommendations	8

CHAPTER ONE

1 INTRODUCTION

An introduction should be written to the project and should, at least, involve the following points:

1. A background, history or general information about the project, previous .
2. The problem statement of the project showing the issue that is going to be solved in the project (if applicable).
3. Aims and objectives of the project.
4. An outline of the chapters included in the project (not required for midterm exam).

1.1 BACKGROUND

You should provide enough background to the reader for them to understand what the project is all about. For example:

What the reader needs to know in order to understand the rest of the report. Examiners like to know that you have done some background research and that you know what else has been done in the field (where relevant). Try to include some references.

- Related work (if you know of any)
- How does this relate to other work in this area?
- What work does it build on?

1.2 PROBLEM STATEMENT

This section describe the presented problem and why it is important to implement such project software to solve it. Thus, the main emphasis should be on:

- What problem are you solving?
- Why it is important about it?
- Why are you solving it?

1.3 AIMS AND OBJECTIVES

What are the project aims and objectives which will try to achieve.

1.4 CHAPTER SUMMERY

List brief description about the content of this project upcoming chapters.

CHAPTER TWO

2 METHODOLOGY

This chapter will cover aspects about the project adapted strategy and work phases which have been carried out during implementation. For software engineering the following example for the methodology is used as clarification. Please, don't draw or describe any diagram related to you project implementation. You don't need to list any algorithm you are using or any ER database diagrams of your work. This is only describe your actions or tasks you have or will done/do during this project.

2.1 SOFTWARE MODEL

One of the following models are used:

1. Waterfall model
2. V model
3. Incremental model
4. RAD model
5. Agile model
6. Iterative model
7. Spiral model

Most common phases are requirement gathering and analysis, Design, Implementation, coding and testing.

2.1.1 REQUIREMENT GATHERING AND ANALYSIS

Requirement gathering and analysis is the first stage and major stage of any SDLC model. This phase is basically the brainstorming phase because it has the many sub stages for like Feasibility Analysis Stages to check how much idea can put into action for development.

In this Stages, communication taking place between end users and the project team, so all the person which are related to the project. In which:

- They gather information for software development.
- Identify and capture project requirements using customer interviews and surveys.
- **Build multiple use cases** to describe each action that a user will take on the new system.

In that a brand new software development takes place more requirement gathering process for development and in other already build software not need too much information and data gathering process.

2.1.2 SOFTWARE DESIGN

This section is about the design tools that will be used for highlighting different parts of the system. These includes database design tools, program description design tools (e.g., class diagrams) and others. This should highlight only what kind of tools is used for design not the software design (which will be specified in the next chapter).

The student should emphasis important aspects such as user friendly design, clarity and ease of use while designing the system. These includes the use of detailed process design on human computer interaction methodologies. Refer to the following links for clarification:

<http://www.ece.rutgers.edu/~marsic/Teaching/SE1/report2.html>

2.1.3 IMPLEMENTATION

This section describe the methods and tools used for implementation. The methods may include:

- Are you using procedural, object or functional programming?

- What are the components that should be addressed (database connectivity, object connectivity, security, etc).
- Explore the best practices for implementation.
- Discuss the compatibility issues during this phase. Or what OS or devices this software will be implemented for?

The implementation tools include a **brief** introduction about:

- Software tools used such as programming languages (java, c++, php) or packages (Matlab, autocad, network design, simulators) or servers (apache, WAMP, XAMPP).
- Database tools used such as mysql, sql server, oracle.
- Hardware tools such as sensors, mobiles, interfaces, chips, etc.

It should mention why such tools were selected but not other options (advantage and limitations).

2.1.4 TESTING

This sections introduces the testing methodology used during and after finalizing the implementation to test the units and systems. These should detail the importance of such tools in those phases.

Note:

It should be noted that, these sections can also be used for simulators such as (opnet, matlab, process simulator, questionnaires, and quantitative studies) to describe each phase of the design.

Finally, note that no details about the system design and implementation phase details should be mentioned here. There should be a good understanding between methodology and real implementation.

CHAPTER THREE

3 ANALYSIS AND DESIGN

This chapter should list detailed designs and descriptions of different system components. There should be figures describing different aspect of the design components with their description (Diagrams, relationships, steps and others). All the algorithms and flowcharts should be listed here. You should list the following if applicable:

- Data base design and relationship (e.g., logical and structural design) such as ER diagram and queries.
- Class diagram and program unit interactions.
- Network diagrams.
- Algorithms, pseudo codes and flowcharts.
- Special data structures used if any.
- Equations and formulas.

You should describe each component in details to justify your idea.

CHAPTER FOUR

4 RESULTS AND DISCUSSIONS

The results obtained from the project should be organized in tables or shown by figures and curves, briefly. After that, the results should be clearly and critically discussed. In this chapter, the student should be able to show his/her understanding of the subject and its results. It should also list the results of testing the system with testing sets.

CHAPTER FIVE

5 CONCLUSION AND RECOMMENDATIONS

The conclusions of the project should be briefly presented in this chapter. Note that the conclusion is not an abstract, it should NOT summarize the whole project. It should conclude the main findings and link them to the objectives of the study.

REFERENCES

The references used in the project have to be listed according to an international referencing system. For example:

Bauer, F. L., Programming as an Evolutionary Process, Proc. 2nd. Intern. Conf. Software Engineering, IEEE Computer Society, 223-234, January, 1976.

Beck, K. Extreme Programming Explained, Addison-Wesley, Palo Alto, CA, 1999.

The 1st reference is an example for definite of Journal or conference and the 2nd reference is for definite of Book.

Citing these references using Harvard style (name of author, year) for the above as (Bauer, 1976).