



Guidelines for the Final Year Project Report

Bachelor of Science in Software Engineering (Bsc. SWE)

Prepared by:

Department of Software Engineering

Abaarso Tech University

Last Updated:

May 2018

Important Notice to All Students

PLAGIARISM IS A SERIOUS OFFENCE.

Downloading materials from websites for the report contents is prohibited.

You must sign the report certificate of originality form to confirm that your Final Year Project report has been done by your own efforts without any plagiarism

1.0 INTRODUCTION

One of the assessment methods of a final year project is through a report prepared by the students. Students need to submit the report to the supervisor on a specified date given by the Department of Software Engineering. This document contains a guideline preparing the project report including the content guideline as well as the format guideline.

2.0 FORMAT GUIDELINES OF THE REPORT

Format refers to the overall appearance of the report. Its structure must contain the following features to maintain the **INTEGRITY** and facilitate the reading of the text:

2.1 Margins

Set margins for the **text** according to the following specified distance from the **edge** of the paper:

Top margin: 30 mm

Bottom Margin: 25 mm

Right margin: 25 mm

Left margin: 38 mm (to allow for binding)

NOTE THAT **ALL TITLES** except title of the report, references and appendices MUST BE TYPED **50 MM** FROM THE TOP EDGE OF THE PAPER.

See SAMPLE PAGE 1 for a sample on page set-up

See SAMPLE PAGE 2 for a sample on chapter heading

See SAMPLE PAGE 3 for a sample of the text following the chapter

2.2 Page Numbering

Page numbers should be placed at the **bottom right corner** of the page. Number the pages consecutively with numerals (1,2,3,...) in the text. Use lower case Roman numerals (i, ii, iii,... refer *APPENDIX A* for guideline) for the preliminary pages (beginning with the Title Page and ending at Nomenclature/Abbreviations and Terminology). Page number should **NOT BE SHOWN** on the **first page** of the preliminary (Title Page) and the first page of each chapter (e.g. Chapter I).

2.3 Spacing and Typing

The report must be typed **DOUBLE SPACING** on quality A4 80gm papers. Single spacing is permitted for Declaration, Dedication, Acknowledgments, Abstract, Table of Contents, List of Tables and Illustrations, Footnotes and References. Text must be typed on **double sided paper** only using a font size of twelve (12) points of Times-Roman typeface.

2.4 Paragraphing

No indentation is necessary when starting a new paragraph. Separation of paragraphs is effected through 1 line (double) spacing.

3.0 ORDER OF SECTIONS OF THE REPORT

The contents of the report must be ordered according to the following sections:

COVER PAGE

TITLE PAGE

CERTIFICATE OF ORIGINALITY

DEDICATION (*OPTIONAL*)

ACKNOWLEDGEMENT

ABSTRACT

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS/TERMINOLOGIES

BODY OF THE REPORT

REFERENCES

APPENDICES

3.1 Front Page – Cover page

This page must contain College name, project title, student names and year submitted.

See SAMPLE PAGE 4

3.2 Page i – Title page

This page contains project title, student name, supervisor name, and the phrase “A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING, DEPARTMENT OF SOFTWARE ENGINEERING, ABAARSO TECH UNIVERSITY” and year submitted.

See SAMPLE PAGE 5

3.3 Page ii – Declaration

This page contains the signed declaration from the student on the authenticity of the report. It must be typed at the center of the body of a new page immediately after the title page. The wordings of the declaration should conform to the required standard format.

See *SAMPLE PAGE 6*

3.4 Page iii – Dedication (*optional*)

The dedication honors those who inspired or encouraged the writing of the report. Names of persons whom the author hopes to inspire through his/her writing may also be mentioned. It should not exceed one (1) page and must be typed at the center of the page.

3.6 Page iv– Acknowledgment (*optional*)

Any form of special professional assistance or financial support that is extended or granted by an individual or organization in the preparation of the report is recorded on this page. It should not exceed one (1) page.

3.7 Page v – Abstract

This page contains the synopsis of the report. Its length is normally between 150 and 250 words (**should not be more than one page**). It must contain the following information:

- (i) The aims and objectives of the research project
 - (ii) Research methodology such as methods, techniques and their application
- somebody unfamiliar with your project should have a good idea of what it's about having read the abstract alone and will know whether it will be of interest to them.

See *SAMPLE PAGE 8*

3.8 Page vi– Table of Contents

This page serves as a guide to the contents of the text. Every heading and subheading within the text should be listed verbatim in the Table of Contents.

See *SAMPLE PAGE 9*

3.9 Page viii – List of Tables

See *SAMPLE PAGE 10*

3.10 Page ix – List of Figures

This list includes diagrams, graphs, maps, photographs, and others.

See *SAMPLE PAGE 11*

3.9 Page vii: Abbreviations / Terminologies

This page contains the abbreviations and terminologies of the words used throughout the thesis (if any).

3.10 References

All references used in carrying out the project and in producing the report must be numbered in square bracket.

For example,

[1] Hager, W.W. 1988, *Applied numerical linear algebra*. London: Prentice-Hall International.

[2] Wortman, L.A., Sidebottom, T.O. 1984. *The C programming tutor*. London: Prentice Hall International.

3.11 Appendices

The appendix normally contains the program codes, questionnaires or surveys given to the users, supporting diagrams or sketches, which are not the main component of the research. If there is more than one appendix in a report, then it has to be properly and systematically numbered as *Appendix A*, *Appendix B*, *Appendix C* and so on.

3.12 Body of the Report

The technical contents start after the contents described above end. The contents of a report will **depend on the type of the project carried out**.

Department of Software Engineering

CHAPTER 1: INTRODUCTION

This is one of the most important components of the report. It should begin with a clear statement of what the project is about so that a lay reader can understand the nature and scope of the project.

This chapter may contain the following subsections:

1.0 Introduction

1.1 Problem Background

1.2 Problem Statement

- Describe the nature of the problem in detail.

1.2 Objectives of the Project

- Clear statement of project objectives is necessary. It is recommended to describe in an itemized manner.

1.3 Scope of the Project

- Outline the scope of your project. How did the problem present itself to you in the first place?

1.4 Procedure/Methodology

1.5 Expected Outcome

CHAPTER 2: LITERATURE REVIEW

This chapter involves the systematic identification, locating and analysis of documents containing information related to the research problem being investigated. There are usually many ways of solving a given problem, and you should not just pick one at random. Describe and evaluate as many alternative approaches as possible. The published work may be in the form of research papers, articles, textbooks, technical manuals, or even existing software or hardware of which you have had hands-on experience. Do not be afraid to acknowledge the sources of your inspiration; you are expected to have seen and thought about other people's ideas; your contribution will be putting them into practice in some other context. However, avoid plagiarism: if you take another person's work as your own and do not cite your sources of information you are being dishonest; in other words you are cheating. When referring to other pieces of work, cite the sources where they are referred to or used, rather than just listing them at the end. Describe how

the project work will be implemented. What methodology will be used? What tools and technique(s) will be employed?

CHAPTER 3: ANALYSIS

This chapter documents the analysis of the system to be implemented. An analysis of the requirements should be provided. For example, the requirements of the system should be listed. Functional requirements covering system functionality expected by the users and non-functional requirements covering reliability, portability, and response and processing times should be addressed with detailed justification.

This chapter may contain the following subsections:

3.1 Existing System Description

- In this section, information on the existing system should be provided.
-

3.2 Requirements specifications of the proposed system

3.2.1 Functional Requirements

A functional requirement defines the functionality (modules) of a system or one of its subsystems. Functional requirements deal with what the system should do or provide for users. They include description of the required functions, outlines of associated reports or online queries, and details of data to be held in the system.

Example: Store Management System

- Log on: the system validates the store staff to use the system.
- Receive Item: the system allows the store keeper to enter a new item which comes from deliverer or donor at the acquisition time.
- Approve Request: the system allows the store administrator to search the availability of the items in the store before approving the requested item availability and relevance.

3.2.2 Non-Functional Requirements

Non-Function requirements mean the characteristics that are not related to system's physical functions but are the characteristics that favorite to the physical functions.

Example: Store Management System

- Maintenance: The store Management System is being developed in PHP. Php is an object oriented programming language and shall be easy to maintain
- Portability:-The store Management System shall run in any Microsoft Windows environment that contains PHP Runtime.
- Reliability: - The store Management System service should not access without authenticate user.
- Standards Compliance: - The graphical user interface of the system shall have easily understood to the user (have consistent look and feel graphical user interface).
- Performance: -Acceptable response times for system functionality.
- Security: - Access to the various subsystems will be protected by a user log in screen that requires a user name and password.

3.2.3 Hardware requirements

Hardware	Minimum System requirement
Processor	2.4 GHZ processor speed
Memory	128 MB RAM (256 MB Recommended)
Disk space	80 GB (including 20 GB for database Management system)
Display	800 x 600 colors (1024 x 768 High color-16 bit Recommended)

3.2.4 Software requirements

Software	Minimum System requirement
Operating System	Windows 7 or later
Database	MYSQL
Programming languages	JAVA, JavaScript, PHP
Compiler	Visual Studio SDK
Web browser	Firefox, Chrome etc

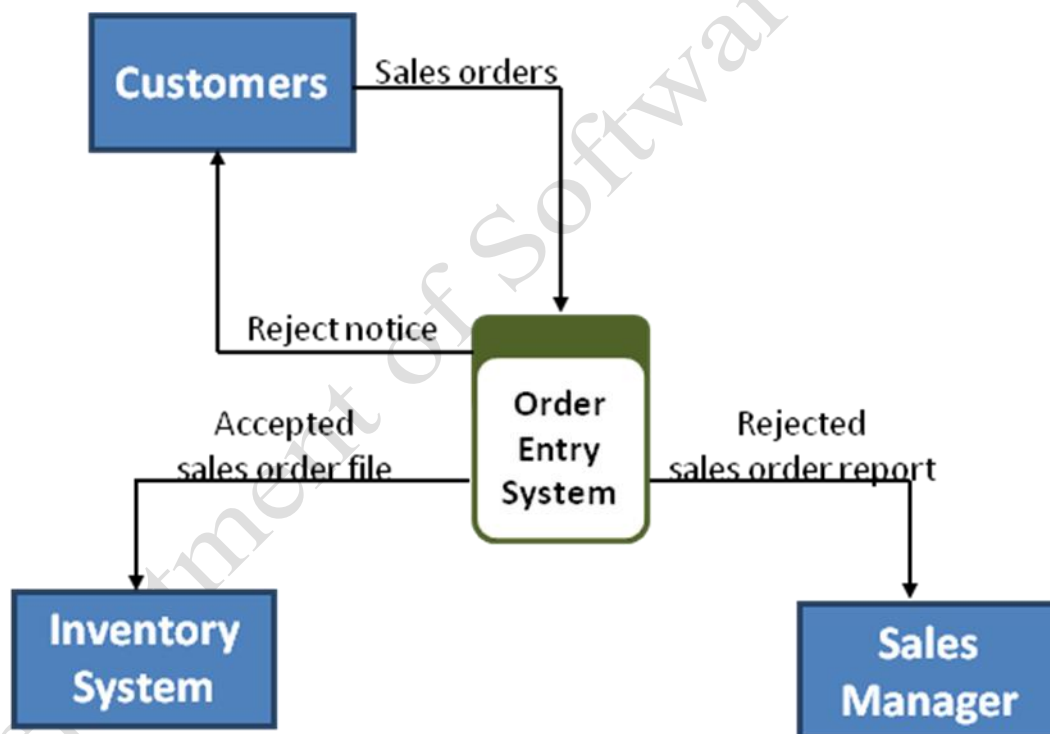
3.3 Requirement (Functional) modeling of the proposed system

Functional models are used to document and understand *requirements*. Describe business processes and interaction with environment. Used for both as-is and proposed systems. The specific models created depend on the technique being used for systems analysis. The modern structured analysis technique uses data flow diagrams (DFDs) and entity-relationship diagrams (ERDs). Object-oriented techniques produce class diagrams and use case diagrams.

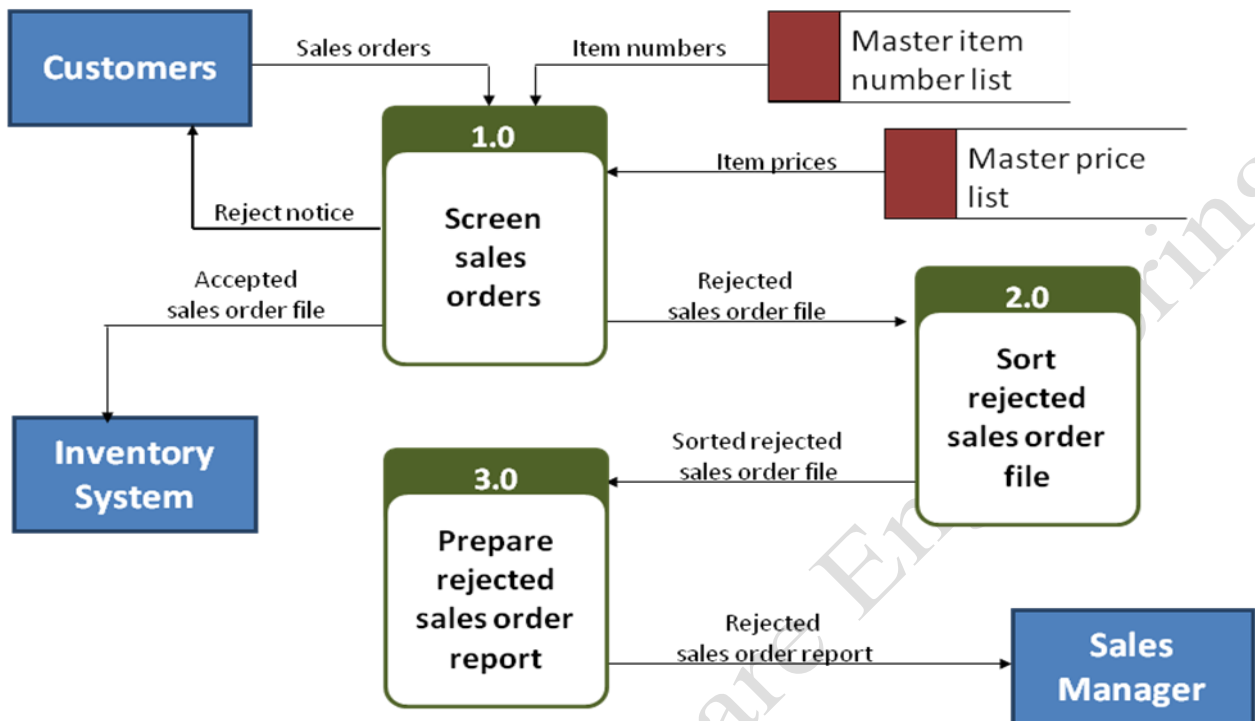
3.3.1 Data Flow Diagrams (DFD)

The traditional approach to information system development describes activities as processes carried out by people or computers. A graphical model that has proven to be quite valuable for modeling processes is the data flow diagram.

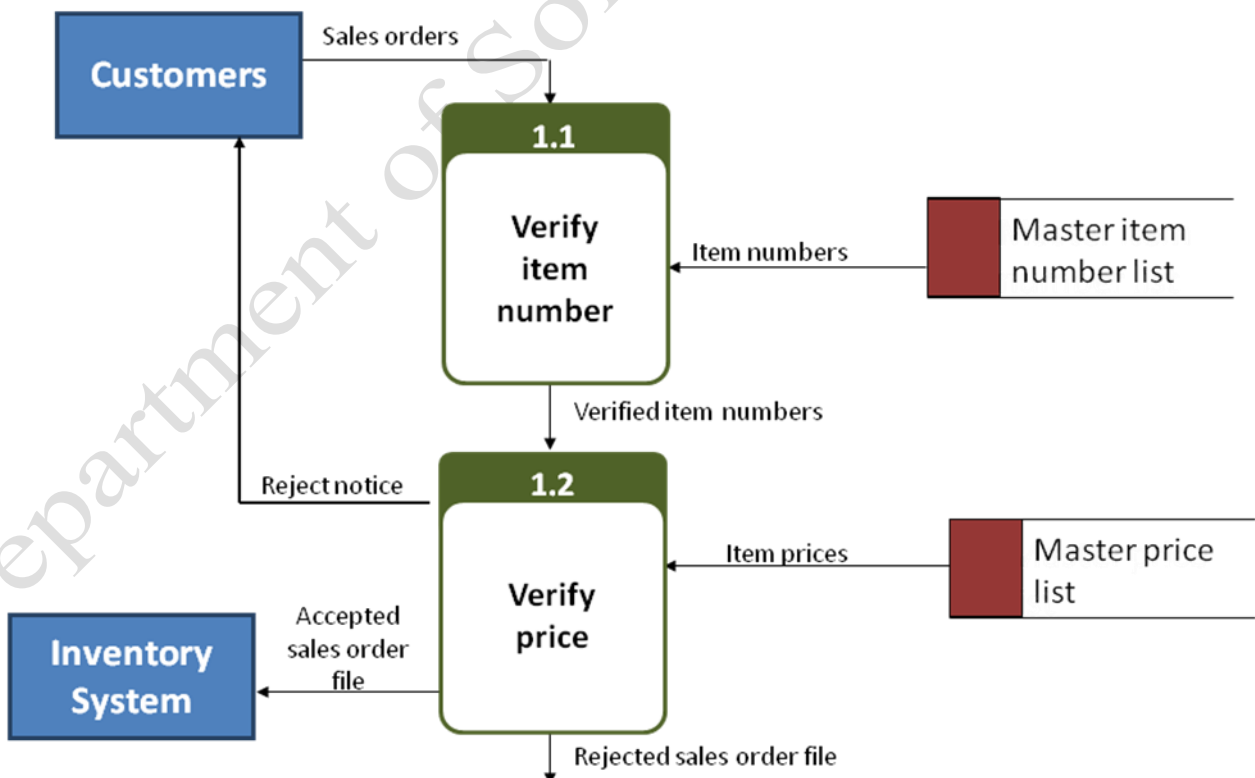
3.3.1.1 Context Diagram (sample)



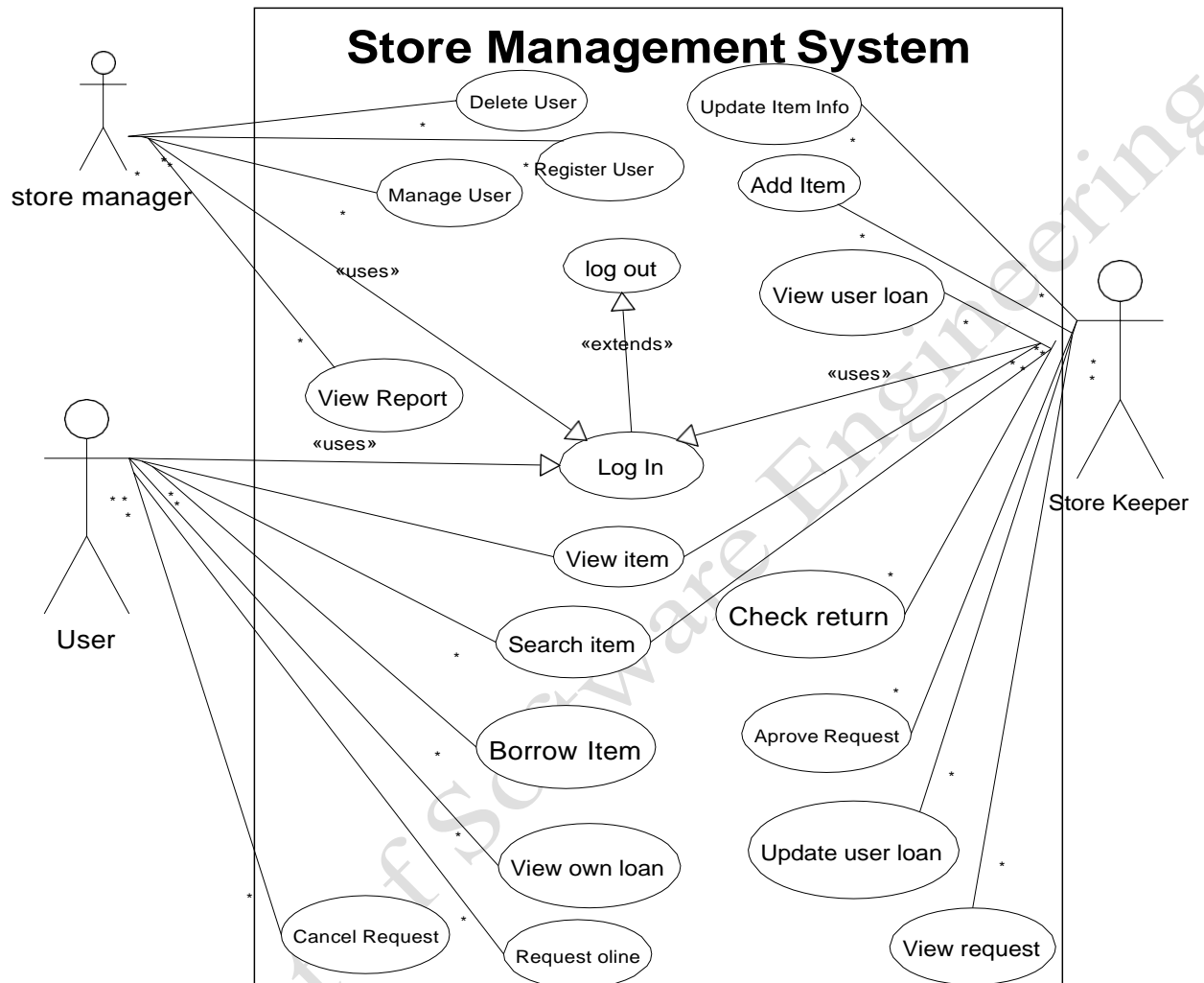
3.3.1.2 Level-0 Diagram (sample)



3.3.1.3 Level 1 Diagram (for Process 1.0)



3.3.1 Use Case Diagrams (Sample)



3.3.1.1 Use Case Descriptions

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. Use case is a list of steps, typically defining interactions between a role (known in UML as an actor) and a system, to achieve a goal. The actor can be a human or an external system.

Sample: Store Management System (Register user Use Case)

Use case name	Register User
Participating actor	Store manager
Entry condition	The actor activates the form.
Basic course of action	<ol style="list-style-type: none">1. The system display the form2. The actor fills the attributes of the beneficiary3. The actor clicks register button4. The beneficiary is registered to the Database
Exit condition	When the actor exit from the form.
Pre condition	The actor should log in to the system.
Post condition	The beneficiary is registered to the Database
Alternative course of action	If the entered attribute(s) is(are) incorrect go to step 2 [alternative course of action 1]

3.3.2 Activity Diagrams (Example: Register user)

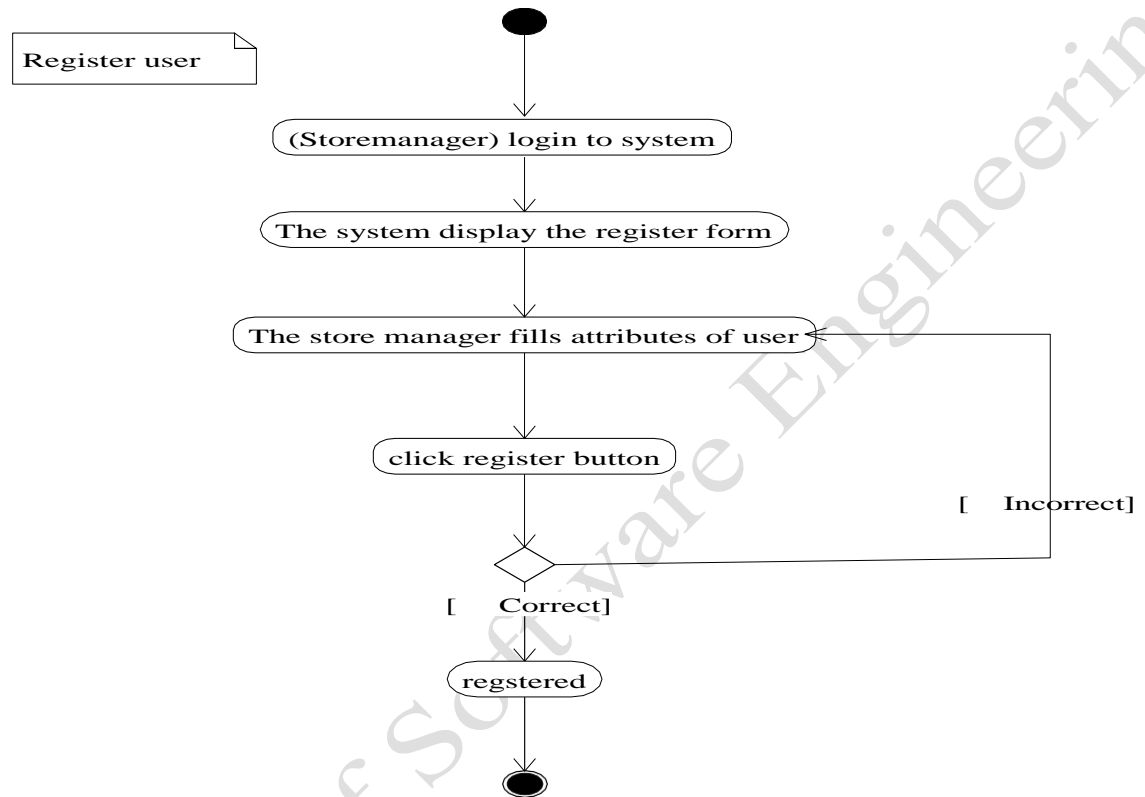


Figure 2.13: Register user

3.3.3 Sequence Diagrams (Example: Register user)

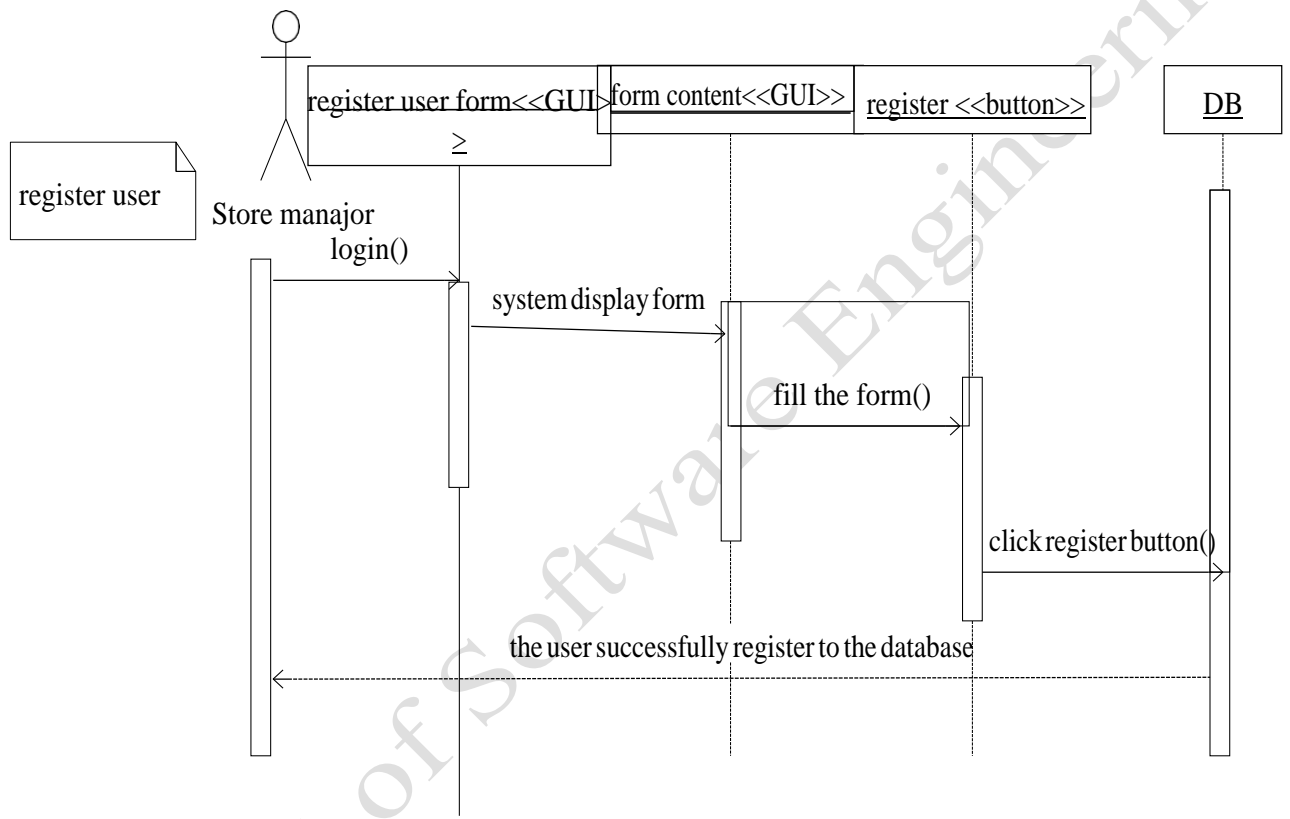


Figure 2.30: Register user

3.3 Object modeling of the proposed system (Class Diagram)

3.3 Data modeling of the proposed system (Conceptual Data Modeling using ERD)

A logical data model describes the data and the relationships in detail at a very high level. This does not include how data is represented physically in the database , but describes at a very abstract level. It basically includes the entities and the relationships among them along with attributes of each entity.

CASE Tools that could be used:

- Microsoft Visio or Any UML modeling Software
- Data Modeling Software such as Toad or ERWin

CHAPTER 4: SYTEM DESIGN

The Design section documents the design decisions that have been taken. The structure of the system and its components has to be established.

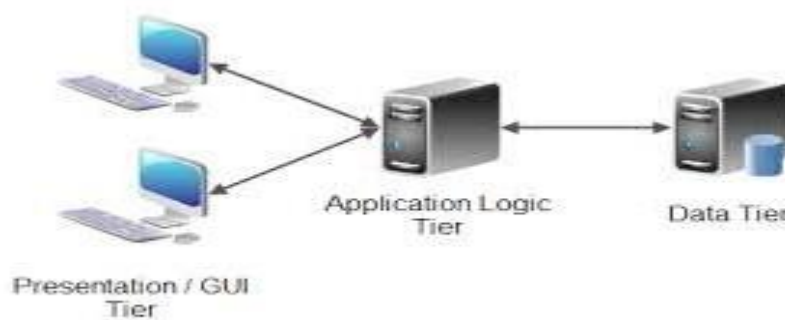
The design should consist of the following:

- Architectural Design - Interaction between components and/or modules
- Interface Design
- Physical Database Design

4.1 Architectural Design

The objective of architecture design is to determine how the software components of the information system will be assigned to the hardware devices of the system. The architecture design flows primarily from the nonfunctional requirements, such as operational, performance, security, cultural, and political requirements.

FIGURE 8-3



n-Tiered Client-Server Architecture

4.2 User Interface Design

A user interface is the part of the system with which the users interact. It includes the screen displays that provide navigation through the system, the screens and forms that capture data, and the reports that the system produces.

Output Design: Screen & Print layouts

Input (form) Design: For entering data

4.3 Physical Database Design- is the process of transforming a *logical* data model into an actual *physical database*. Physical database modeling deals with designing the actual database based on the requirements gathered during logical database modeling. All the information gathered is converted into relational models and business models. Objects are defined at a level called a schema level. A schema is considered a group of objects which are related to each other in a database

- ☐ A physical database definition (or schema) based on the entity model (ER) developed during the analysis phase
- ☐ Data Dictionary

CHAPTER 5: IMPLEMENTATION AND TESTING

The implementation chapter describes how the different components in the project have been implemented. It should also consist of:

- ☐ Developments tools and environment used
- ☐ Implementation of different modules (including detail steps about how they were developed)
- ☐ Sample codes of the main pieces of logic (including standards and conventions)
- ☐ Difficulties faced and how they were addressed.

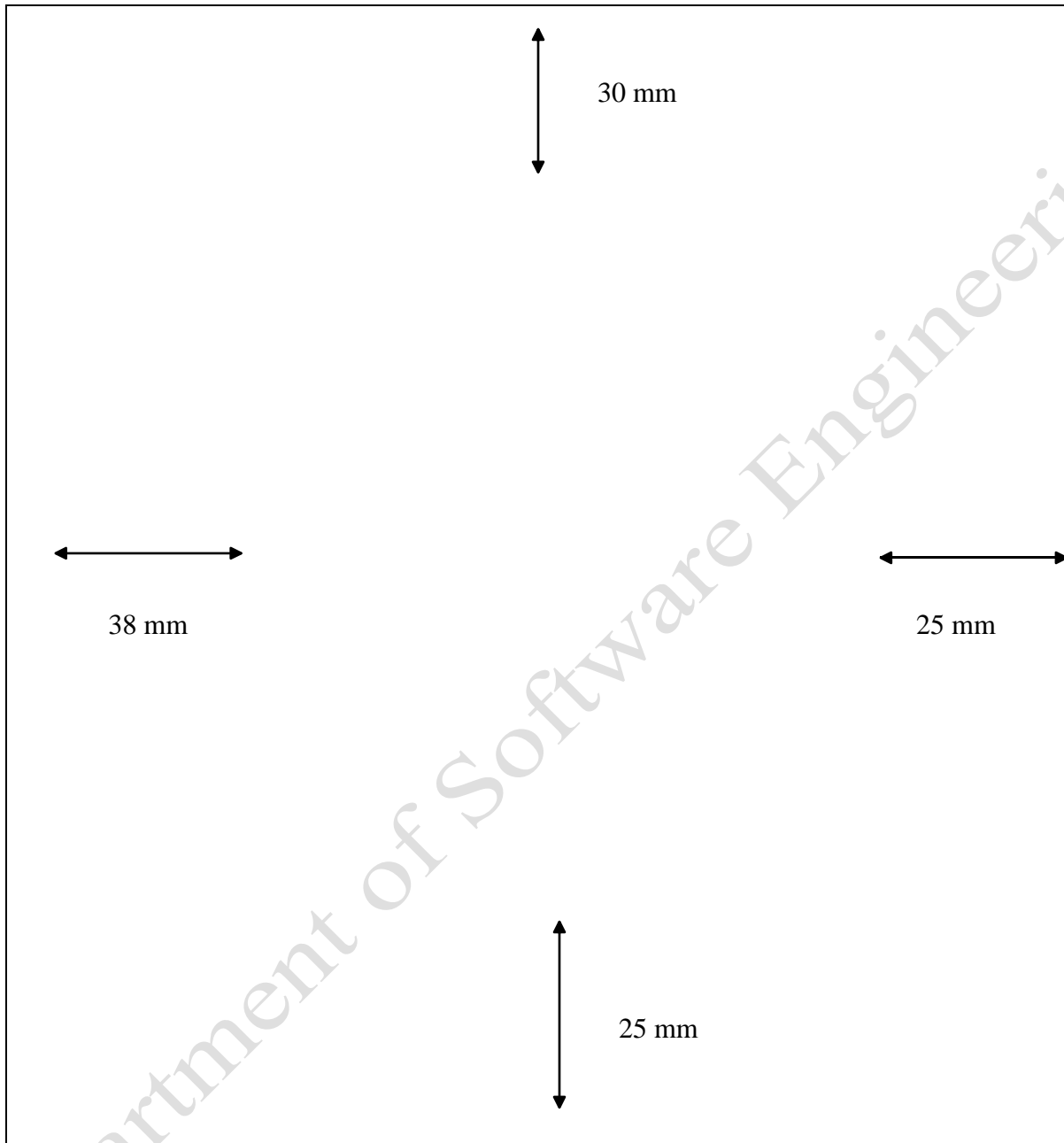
This chapter should also prove that proper testing of system was done. For example, a comprehensive test plan that was used to verify and validate the system should be provided. Evidence should be provided of using a wide range of test data.

CHAPTER 6: SUMMARY AND CONCLUSION

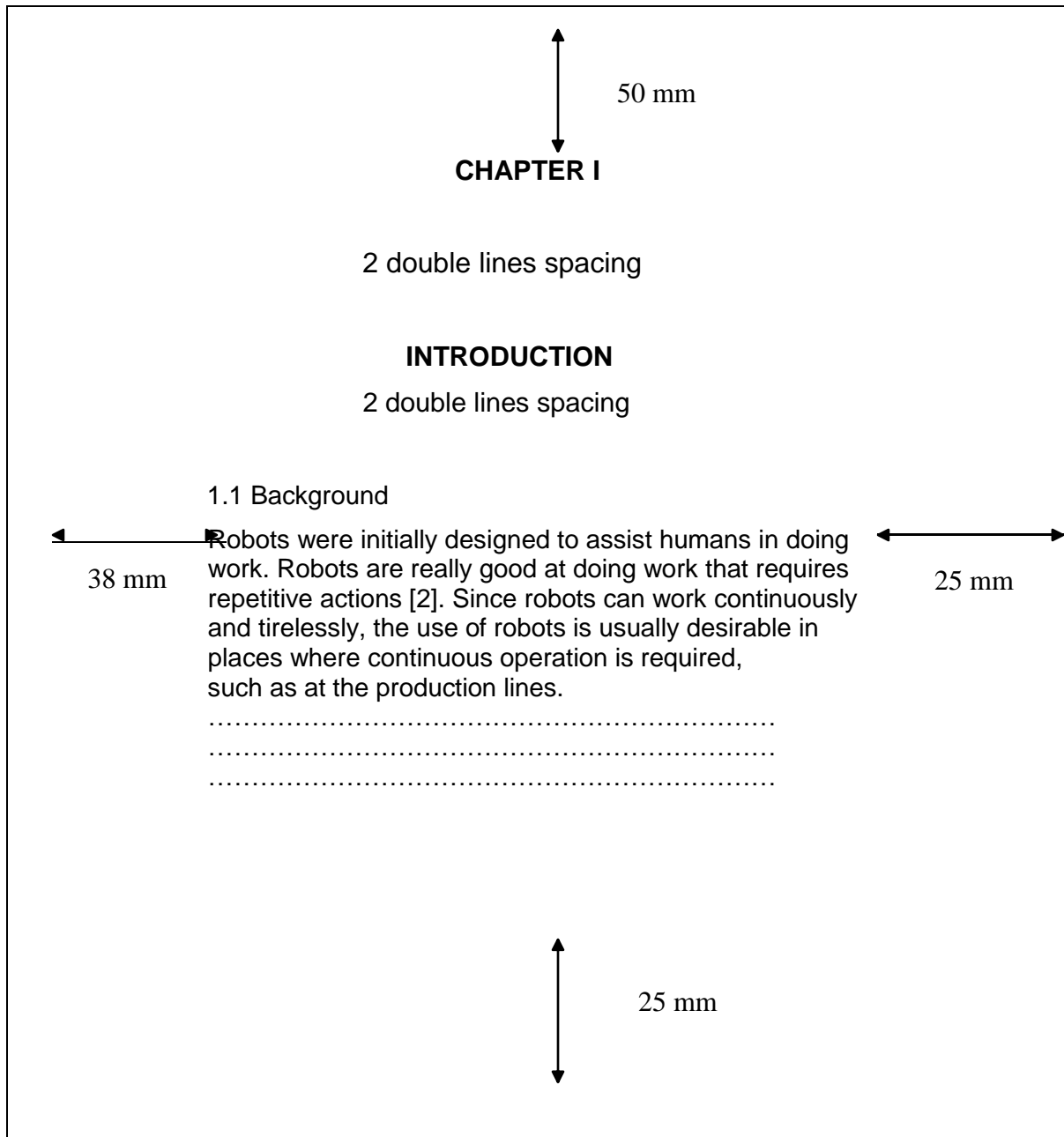
The overall result of the project is summarized as clearly as possible.

What has been accomplished for each project objective (Students should clearly state the conclusion for each project objective) what are the major things that you learned from this project? What work still needs to be done on the system and how can it be improved and/or enhanced?

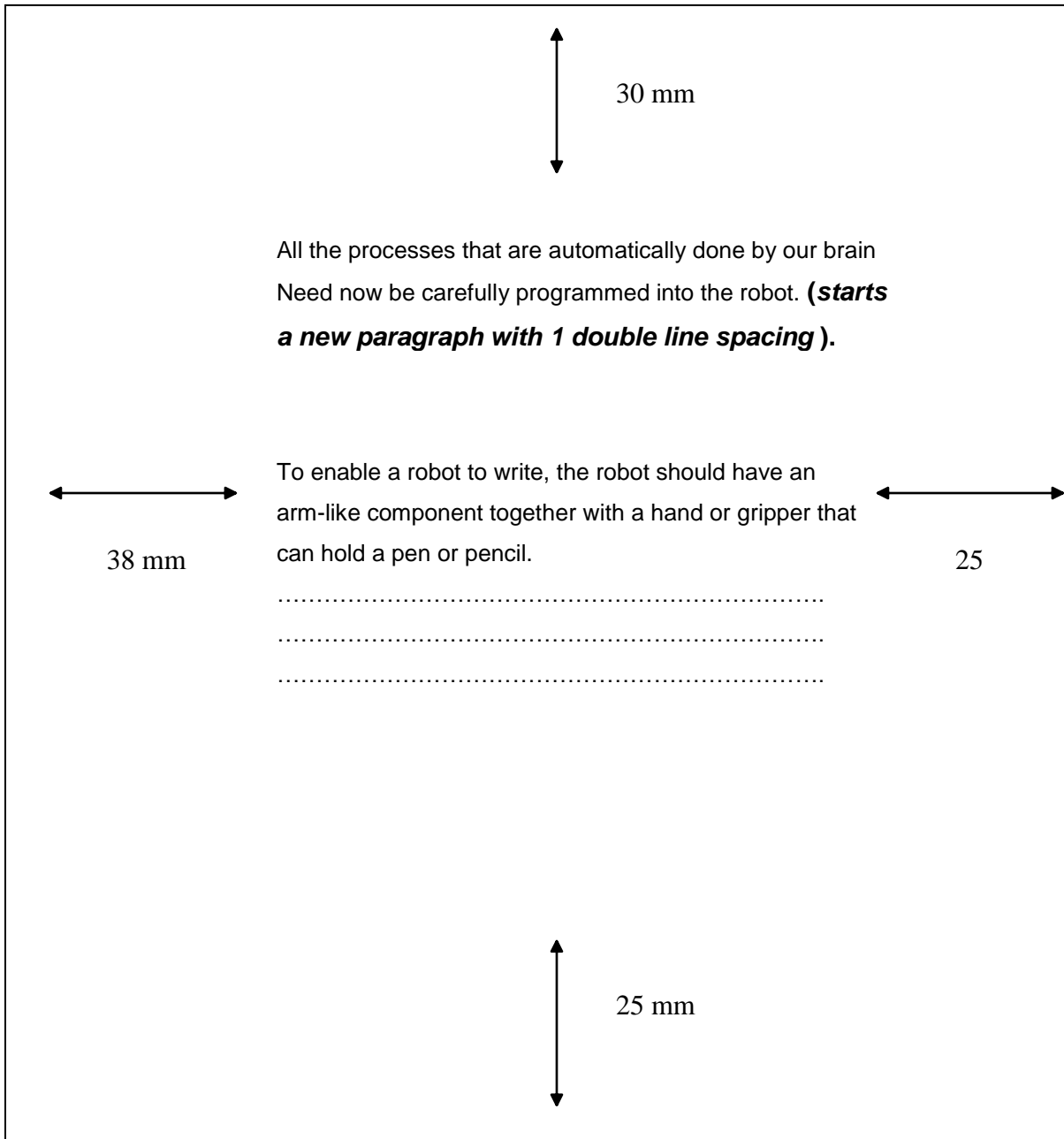
SAMPLE PAGE 1: MARGINS FOR TEXT



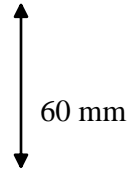
SAMPLE PAGE 2: CHAPTER HEADING



SAMPLE PAGE 3: PAGE FOLLOWING CHAPTER HEADING



SAMPLE PAGE 4: COVER PAGE OF REPORT

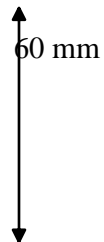


DEPARTMENT OF SOFTWARE ENGINEERING
ABAARSO TECH UNIVERSITY

[PROJECT TITLE]

1. STUDENT NAME
2. STUDENT NAME

date



SAMPLE PAGE 5: TITLE PAGE OF REPORT

(Left margin 38 mm, right margin 25 mm)

↕ 60 mm

[PROJECT TITLE]

By

1. STUDENT NAME

2. STUDENT NAME

Project Supervisor: [Name]

**A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE BACHELOR OF SCIENCE IN SOFTWARE
ENGINEERING**



**DEPARTMENT OF SOFTWARE ENGINEERING
ABAARSO TECH UNIVERSITY**

↕ 2018

SAMPLE PAGE 6: REPORT DECLARATION

(Left margin 38 mm, right margin 25 mm)

CERTIFICATE OF ORIGINALITY

(2 double lines spacing)

I hereby declare that this report, submitted to the Department of Software Engineering of the Abaarso Tech University as a partial fulfillment of the requirements for the Bachelor of Science in Software Engineering has not been submitted as an exercise for a degree at any other university. I also certify that the work described here is entirely my own except for excerpts and summaries whose sources are appropriately cited in the references.

This report may be made available within the university library for the purposes of consultation.

DATE

STUDENT NAME (ID NO.)

STUDENT NAME (ID NO.)

Signature _____

SAMPLE PAGE 7: APPROVAL SHEET

(Left margin 38 mm, right margin 25 mm)

Approval Sheet

This is to certify that this project report entitled "Project Title" submitted by:

STUDENT NAMES (ID NO.)

In partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering of the Department of Software Engineering, Abaarso Tech University during the academic year 2017-18 has been accepted.

Supervisor:

Signature:

Date:

SAMPLE PAGE 8: ABSTRACT OF REPORT

(Left margin 38 mm, right margin 25 mm)



30 mm
↓
ABSTRACT

(2 double lines spacing)

Currently, there are many ongoing researches that are targeted at making robots more human-like. One of the tasks that can be done by humans easily but is difficult to be done by robots is writing. In this research, a flexible algorithm that can allow a robot to write is presented.

SAMPLE PAGE 9: TABLE OF CONTENTS

(Left margin 38 mm, right margin 25 mm)

	
TABLE OF CONTENTS	
<i>(2 double lines spacing)</i>	
	Page
TITLE PAGE	i
CERTIFICATE OF ORIGINALITY	ii
DEDICATION	iii
ACKNOWLEDGMENTS	iv
ABSTRACT	v
TABLE OF CONTENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	x
ABBREVIATIONS 7 DEFINITION	xi
CHAPTER I INTRODUCTION	
1.1 Background	1
1.2 Problem Statement	2
1.3 Objectives	
1.3.1 Project Objectives	3
1.4 Scope	
1.4.1 User Scope	9
1.4.2 System Scope	10
	

CHAPTER 2 LITERATURE REVIEW

2.1 Robotic 12

2.2 Development methodologies 14

CHAPTER 3 SYSTEM ANALYSIS 20

.....

CHAPTER 4 SYSTEM DESIGN 45

.....

CHAPTER 5 IMPLEMENTATION AND TESTING 70

....

CHAPTER 6 SUMMARY AND CONCLUSION 80

.....

REFERENCES 85

APPENDICES

APPENDIX A: Requirement Survey 93

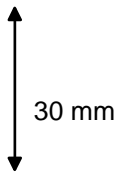
APPENDIX B: Source Code 94



25mm

SAMPLE PAGE 10: LIST OF TABLES

(Left margin 38 mm, right margin 25 mm)



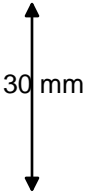
LIST OF TABLES
(2 double lines spacing)

Table No.	PAGE
3.1 Example of a Segment Table	98
3.2 Range for each zone block	105
.....	
.....	
5.1	

25 mm

SAMPLE PAGE 11: LIST OF FIGURES

(Left margin 38 mm, right margin 25 mm)



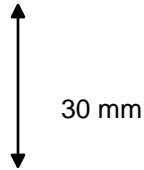
LIST OF FIGURES
(2 double lines spacings)

Figure No.	Page
2.1 Segmentation on letter „a“, „e“ and „n“	19
2.2 Block diagram of the robot writing system	21
2.3	25
2.4

25 mm

SAMPLE PAGE 12: REFERENCES

(Left margin 38 mm, right margin 25 mm)



REFERENCES

(2 double lines spacing)

- [1] Blatner, D., "Spacing out with your type; leading and kerning", MacWorld, vol. 14, pp. 158 (2), 1997.
- [2] P. Michelman, P. Allen, "Compliant manipulation with a dextrous robot hand", in *Proc. IEEE International Conference on Robotics and Automation*, vol. 3, pp. 711-716, 1993.
- [3] Darken R. P. and Peterson B. 2001. Handwriting and Representation.
URL: <http://citeseer.nj.nec.com/561810.html> Last Date of extraction: 8/8/2006
- [4]

25 mm

Final Year Project

What should be in your Final Report?

- Abstract
- Chapter 1: Introduction
 - Problem statement;
 - Provide background information about the problem domain, which highlights the need for a solution.
 - Proposed Solution;
 - Objectives;
 - Scope;
 - Constraint;
 - Methodology;
 - Schedule.
- Chapter 2: Literature Review
 - Background information;
 - Current systems/domain analysis; Comparison of similar systems.
 - Describe/criticise the current available systems/technology.
 - Compare and contrast the existing systems/technology with the proposed solution.
 - Describe how the proposed solution fits into the overall business or strategic objectives of the organisation/community (i.e. how it will work with other systems or the environment).
- Chapter 3: Requirements Specification
 - System Requirements (Functional & Non-functional)
 - Specify the functional system requirements for each functional user requirement.
 - Specify the non-functional requirements and domain requirements (if any).
 - Hardware & Software Requirements (Development & Usage)

- Specify the hardware and software requirements during development (for developers to build the product) and usage (for users to use the product).
- Chapter 4: Design Specification
 - High-level System Modelling (e.g. Context Diagram; Use Case; Activity Diagram)
 - Illustrate the system context and boundary by using Context Diagram.
 - Illustrate the system's functional requirements by using Use Case Diagrams and Use Case Specification (fully dressed).
 - Illustrate the business process by using Activity Diagram/Document Flowchart.
 - Architecture Design
 - Identify and describe the suitable design pattern(s) to be used (e.g. Layered, Repository, Client-Server, Pipe-Filter etc.).
 - Illustrate modules/sub-systems and their relationships using the appropriate diagram(s) (e.g. Module Hierarchy Chart/Structural Chart etc.).
 - Architecture/Structural (i.e. Class Diagram; Sequence Diagram/DFD; Application Architectures; Module Hierarchy Chart);
 - Database Design
 - Design the data objects and structures (i.e. Entity-Relationship Diagram/Class Diagram).
 - Specify the data flows between processes/objects (i.e. Data Flow Diagram/Sequence Diagram).
 - Describe the attributes of each data object/class (i.e. Data Dictionary).
 - Interface Design
 - Sketch the user interfaces using low-fidelity (i.e. card-based prototypes) or high-fidelity (i.e. Macromedia Director/Visual Basic/MS PowerPoint etc.) prototypes.

- Design the system interfaces between modules/sub-systems.
- Algorithm Design
 - Construct the algorithms used to provide system functionalities (e.g. pseudocode, decision table/tree, system/program flowcharts).
 - Illustrate system behaviours, if any (e.g. Statechart Diagram).
- Chapter 5: Development and Testing
 - How the system is implemented; Explanation of critical code segments.
 - Testing methods and strategies (i.e. white box, black box testing); Process (unit /component testing; system/integration testing).
- Chapter 6: Conclusion
 - Summary; Strengths and limitations; Future Improvement.
- References
 - List the sources of information that support the proposal (i.e. books; newspapers cuttings; conference and journal articles; web portals etc.).
 - Use the appropriate referencing formats (i.e. Gaya UKM).
- Appendices (User manual; Outputs)
 - Briefly explain the requirements elicitation/gathering process (who, when, what, why, how). Include some evidence (e.g. interview questions/scripts; survey questionnaires; shots of observation scenes etc.)