THE COPPERBELT UNIVERSITY

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

COMPUTER SCIENCE DEPARTMENT

COMPUTER ENGINEERING DEPARTMENT

INFORMATION TECHNOLOGY AND SYSTEMS DEPARTMENT

PROJECT REPORT GUIDELINES

CS 400, IS 400, CS 301, BIT 300, DIT 500, BIT 590

Table of Contents

INTRODUCTION

All undergraduate students studying the various degree programmes of SICT at CBU should complete independent projects as part of their studies. Except the BIT students, other students (Computer Engineering, Computer Science and Information Systems) will do their projects in Year3 and Year4. In year3 the students will do a Group project and in the Final year the students will do a Major project. Based on the complexity of the final year project students are allowed to do in groups not exceeding more than two students.

Note that CS301 is the course code for the Group Project and CS400 is the course code for Major Project Both are 4 credit courses. Generally, a 4-credit lecture course typically requires you to spend 4 hours in lecture and an average of 8 hours each week completing specific assignments outside of class. Therefore, for a project, you will use all 12 hours of each week working independently on your project. You will meet with your supervisor each week, so if there are 11 hours left, this adds up to 275 hours of work over the course of 25 weeks spread across the three terms.

The BIT students will do their projects in levels 5 and 10. If you are an IT student, the level 5 and level 10 projects are your only academic responsibility for a whole semester, and we expect you to devote yourselves entirely to your projects during that time.

All these projects require a significant amount of time and effort from both students and supervisors and replace other material that could be learned during that time. The students can't accomplish a project in only a week or even a month, while they are also attending to your other courses. They need to set small, frequent goals throughout the project and to make weekly progress attaining those goals.

EXPECTED LEARNING OUTCOMES FOR POJECTS

- Having successfully completed the project the students should be able to demonstrate knowledge and understanding of:
- Project methodology
- Apply mathematical, scientific and engineering knowledge to a technical investigation.
- Analyse published technical work through literature reviews.
- Produce a risk assessment for the proposed project.
- Specify, Plan and initiate implementing a yearlong technical project.
- Carry out initial data acquisition and analysis for the project.
- Define and manage the resources required for the investigation
- Carry out practical and computational work as required.
- Solve problems of implementation and analysis.
- Acquire new knowledge and information independently.
- Use ICT in information gathering, analysis and presentation
- Prepare a technical report and give a technical presentation.
- Communicate effectively both in written and verbal ways.

As the curriculum of the various degree programmes under SICT requires two such projects to be carried out, the student's ability to do projects will grow with each project. To demonstrate that the students are gaining the above-mentioned skills, they will have to produce a number of deliverables throughout the year.

The main deliverable varies with the specialization of the degree programme the students are enrolled with

- Computer Engineering (CE) students will focus on hardware construction, hardware-software integration, analyse equipments to determine the best way to improve it, and craft solutions to problems.
- Computer Science (CS) students will develop software systems focussing on analysing algorithms for performance, generating formula for simulations, network design, modeling data and information processes
- Information Systems (IS) students will design, develop and deliver an information system, focussing on solutions to business problems/opportunities or use a quantitative approach to do so.
- Information Technology (BIT/DIT) students should be involved in the analysis, design, and development of an information technology service or product.

TIMELINE AND MILESTONES

The deliverables are spaced out evenly throughout the year.

Note: The students are expected to submit their proposed project topics at the end Term III of the previous academic year. For example, CS400 topics should be submitted at the end of Term III during the third year of study. CS301 topics have to be submitted at the end of Second year.

Milestones	**Timeline**			
	CS400/IS400 (common to CS, CE, IS students)	**CS301 (common to CS, CE students)**	**BIT590 (level 10)**	**DIT 500/BIT300 (level5)**
Allocation of Supervisors	1st week of Term1	1st week of Term1	1st week of Semester	1st week of Semester
Written Project Proposal submission to the department	4th week of Term1 of the Academic year	4th week of Term1 of the Academic year	4th week of Semester	4th week of Semester
Proposal oral presentation in front of SICT panel of Staff members	5th week of Term1	N/A	N/A	N/A
Student meeting with supervisors	Weekly once as agreed with the Supervisor throughout the year	Weekly once as agreed with the Supervisor throughout the year	Weekly once as agreed with the Supervisor throughout the Semester	Weekly once as agreed with the Supervisor throughout the Semester
Project Progress Report submission to the department	6th week of Term2	N/A	N/A	N/A
Project progress Seminar (oral presentation and	8th week of Term2 (in front of	N/A	N/A	N/A

prototype demonstration) to evaluate student progress	Department Panel of Staff members)			
Final Project Report Submission to the department	4th week of Term3	4th week of Term3	Last week of the Semester	Last week of the Semester
Final Project defense (Oral presentation and final running system demonstration)	6th week of Term3 (in front of SICT panel of staff members)	6th week of Term3 (in front of Department Panel of Staff members)	Last week of final examinations of that particular Semester (in front of Department panel of staff members)	Last week of final examinations of that particular Semester (in front of supervisor and one SICT staff member)
Corrected Project Report submission	7th week of Term3	7th week of Term3	To be submitted a week after final presentation	To be submitted a week after final presentation

Note: The student-supervisor meeting Register has to be signed for every meeting. Find the template in the Appendices III & IV.

The deliverables related to each milestone are explained below:

Project Proposal
A written proposal that gives the topic you have chosen, the objectives for completing the project (in other words, the reasons why completing this project is important), and a project plan including resources you require, specific written sources you will consult to complete the project, the risks you face, and a project timeline listing the basic activities you will complete and how long you expect each to take. *(Refer sample Project proposal template in Appendix 1)*

Proposal oral presentation
A 10-minute oral presentation of your topic to the SICT panel of staff members

Student meeting with supervisors
To meet the major milestones, you need to make continuous progress. You must meet with your supervisor weekly to demonstrate your progress, to receive feedback, and to discuss the next steps. The outcome of these sessions will be recorded in the project log book. As much as possible the supervisor will encourage the student to generate ideas and carry out the work on his/her own, but directing the student where to find information and how to carry out investigations if required.
The aim will be to ensure the students are becoming independent investigators, making use of various resources, including that of the supervisor, in the course of his/her research. The Supervisor will assign 35% (refer below) of the total marks based on his or her assessment of the student's weekly progress. This rewards the students for making steady progress.

For example, your supervisor will expect you to submit your requirements document (which will become a chapter) soon after you submit your proposal. You will submit your design document shortly after that.

Project Progress Report
A written report that gives chapters 1-4 of the final report (Introduction, Literature Review, Requirements Analysis and Specification, System Design), plus a brief description of why you choose that specific aspect of the system to prototype.

Project Progress Seminar
1. A working prototype of some aspect of your system. This ensures that you begin construction of the system early enough that you can follow a contingency plan if you encounter problems.
2. A 10-minute oral presentation of your topic to the department, which includes a 5-minute demonstration of your prototype.
NOTE: This seminar will determine whether you must continue with the project or not. If there is unsatisfactory progress on the project work at this point, the candidate will be disqualified from continuing with CS400.

Final Project Report
A final written report, in which the chapters in the progress report are updated and the rest of the chapters are added. It will be reviewed by the department faculty. *(Refer Project Final Report Template in Appendix II)*

Final Project defence
1. Your complete working system. This will be demonstrated in front of a panel of SICT Staff members where the faculty can evaluate your skills on system development.
2. A brief oral presentation of your topic in front of a panel of SICT Staff members where they ask questions to test the depth of the knowledge you have gained so far in studying the programme and how you have efficiently applied it in doing your project.

Corrected Project Report
The corrections given in the report by the panel members are incorporated and the final corrected report have to be submitted back to the department.

ASSESMENT STRATEGY OF THE PROJECT WORK

The supervisor will assess the level of independence of the student during the course of the project as well as the student's enthusiasm, application and effectiveness in the project. Apart from this the students' progress will be assessed through the Progress Report Seminar in Term2. An updated literature review, including Requirements and Design specifications will be included in the Report together with indication of future work.

At the end of Term 3 the Final project report and presentation will provide evidence of the communication skills as well as the content of the work and the applications of the principles of research methodology. Levels of understanding will also be assessed through the oral examination. The Final Project report will be expected to contain details of the investigations carried out, the results and their analysis, a comprehensive discussion of the outcomes and a formulation of the important conclusions and significance of the work.

Marks are distributed as follows:

CS400

Milestones Achieved	Percentage of Marks allocated	Presentation panel	Who Assesses the Performance
Project proposal & Oral presentation	10%	SICT	SICT panel
Project progress Report and Seminar	20%	Department	Department panel
Supervisor CA marks based on weekly progress	35%	Weekly meetings with Supervisor	Supervisor
Final Project Report & Final defence	35%	SICT	SICT panel

CS301

Milestones Achieved	Percentage of Marks allocated	Presentation panel	Who Assesses the Performance
Project proposal	10%	Supervisor	Supervisor
Supervisor CA marks based on weekly progress	30%	Weekly meetings with Supervisor	Supervisor
Final Project Report	30%	Supervisor	Supervisor
Final defence	30%	Department	Department panel

BIT590 and DIT500/BIT300

Milestones Achieved	Percentage of Marks allocated	Presentation panel	Who Assesses the Performance
Project proposal	20%	Supervisor	Supervisor
Supervisor CA marks based on weekly progress	30%	Weekly meetings with Supervisor	Supervisor
Final Project Report & Final defence	50%	Department	Department

STYLE GUIDELINES FOR ALL PROJECT REPORTS

The main text of the final report, excluding preliminaries and other functional parts like appendices, references, table of contents, acknowledgements, etc. should not exceed 50 pages. Refer to Sample Project Report in Appendix II.

While your report must contain all the required details, you will not gain extra points by “padding” your document with extra words or by rambling.

The font shall be Times New Roman or some other serif font with a size of 12 points. Line spacing will be 1.5 in the body, and single spacing in the Abstract, in indented long quotations, and in footnotes and within bibliographic entries. All margins will be 1 inch on each side. Abbreviations such as "e.g." should not be used in your report. One exception is et al. which is used when citing sources written by more than one author (see citations later in this document).

Binding
A4 paper shall be used and should be of good quality and sufficient quality for normal reading. Final reports will be bound within boards, the binding being of a fixed kind of which leaves are permanently secured in the manner of a hardback book. The boards shall have sufficient rigidity to support the weight of the work when standing on a shelf.
The spine shall be labelled with the following information:

- The surname and initials for the candidate (or group number).
- The full or abbreviated title of the report.
- The name of the program for which the report is being submitted (BSCS, BSCE, DIT, or BIT)
- The year of completion.

For example: Lengwe M. Voice Messenger BSCS 2007

Page Numbering
All pages before the Motivation section should be numbered in Roman numerals. The title page will be counted, but not numbered. The numbering format for the rest of the document will be 1,2,3, … or page 1, page 2, page 3, … and at the bottom-right corner of the page.

CONTENTS OF THE PROJECT PROPOSAL

Your proposal is a short document. It starts by describing the topic you have chosen and the objectives for completing the project (in other words, the reasons why completing this project is important). (If this sounds like chapter 1 of the final report, it should! Most students turn the beginning of their project proposal into chapter 1.) It also contains a project plan including resources you require, specific written sources you will consult to complete the project, the risks you face, and a project timeline listing the basic activities you will complete and how long you expect each to take. Below is the guideline of the project proposal specifics:

HEADING	**PAGE LIMIT**	**REQUIREMENT**
Title	not more than 15 words	Must be clear and understandable, a recent ICT area or field of Computer Science, Engineering and Information Systems.
Background information	Half page	It covers the historical background of the project subjected to research.
Introduction	Up to One page	This relates to the topic at hand and how it ties with background information. An outline of the sections you will cover should be presented here.
Problem statement	Maximum One page	What is the real issue which the scholar is going to investigate? Use previous literature.
Objective & Specific Objectives	Half page	What objectives the proposed research wishes to accomplish.
Hypothesis/hypotheses Or Assumption (where applicable)	One page	What are the unknown factors which the scholar is going to probe?

Scope of the study	Half page	Area which the scholar is covering (ex: Kitwe, Lusaka? etc) or the period which the scholar is covering (ex: years 2016). It is not the period taken to complete the project.
Literature review	2 pages	Wide literature review (at least recent 10 pieces of work) required in the concerned topic. Go through the books, journals, web pages and conference papers.
The Research Methodology	1 page	Explain how you will undertake your study, the materials you will use, the procedures you will undertake, the analysis you will use. And clearly state which system development methodology you will use were applicable e.g. Agile, Waterfall methodology.
Significance of your study	Half page	What is the significance of your study?
Expected contribution and implications of your study	Half page	How does your study contribute to the field of ICT at the global level, regional or Zambian Context?
Ethical Issues in Computer Science, Computer Engineering and Information Systems research	Half page	Explain how you will account for ethical issues raised by your study.
Project Timeline/Gant Chart	Half a Page	Use a chart to show how you will accomplish your study during a specified period.
Financial Implications	Half a Page	Provide a budget based on your project requirements.
References	List of references cited in your research.	Use APA style of referencing

Note: The proposal will allow for 10 maximum number of pages and ensure the document is formatted to "Times new roman and font size 12"

CONTENTS OF THE PROJECT REPORT

The following sections are the suggested parts of a CS400/ IS400/ CS301/ IS301/ BIT590/ DIT500/ BIT300 project report. Each section/chapter may have one or more sub-section(s)/sub-chapter(s). This will normally depend on the nature of your project.

Title Page

This page has the title of the project, the name of the student (or names of the students if it is a team project). CS students will include the statement "*This project report is submitted in partial fulfillment of the requirements for the award of a Bachelor of Science Degree in Computer Science*". DIT students will include, "*This project report is submitted in partial fulfillment of the requirements for the award of a Diploma in Information Technology*" and BIT students will include "*This project report is submitted in partial fulfillment of the requirements for the award of a Bachelor Degree in Information Technology*".

Abstract

An abstract is a one-page summary of the entire report. This will be the most widely published, and most read, part of your report. It may even be published in journals and catalogues. Typically, it comprises four main points: a concise description of the problem(s) addressed, methods of solving it/them, results and conclusions. It should also highlight the benefits, advantages and originality of the presented work. An abstract must be self- contained. It usually does not contain references. If a reference is necessary, its details should be included in the text of the abstract. A typical abstract has only one page.

Declaration

This section is used to show and authenticate the author of the report. It includes the statement "*I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text*".

(signature/name/date)

Dedications

This part is included if the author wishes to dedicate their work to a person or people who is/are special in his/her life. It should not take more than one page.

Acknowledgements

In this part, the author thanks those who helped him or her in scientific matters related to his or her project; also others who provided indirect help: physical, financial, social, emotional, psychological etc.

Table of Contents

This section contains the headings (and one or more sub-heading(s)) of all the sections and chapters in your project report, and their associated page numbers. Chapter 1 of your report should start on page 1. The earlier pages, including the Table of Contents and Lists of Figuresand Tables, should be numbered using Roman numerals. Microsoft Word can generate this table automatically (including inserting and updating the page numbers for you!) if you use the Heading styles that are provided in the electronic sample document. In MS Word 2007 and later, Heading Styles are found in the ribbon tool bar under Home. Just click on each of your headings, and select the appropriate style (Heading 1, Heading 2, etc.). Then, to update the headings andpage numbers in the Table of Contents, just click on the table and choose **Update Table**.

List of Figures

This section contains the headings of all the figures (i.e. diagrams, pictures and graphs) in the main body of your project report, and their associated page numbers. Again, Word can keepthese up to date for you. To insert figure captions that can be referenced in the List of Figures, choose **Insert Caption** whenever you are going to write a caption, and choose the **Figure** label option. Applying the Caption Figure style to each figure caption will give you consistent spacing throughout your document.

List of Tables

This section contains the headings of all the figures tables in the main body of your project report, and their associated page numbers. To insert table captions that can be referenced in theList of Tables, choose **Insert Caption** whenever you are going to write a caption, and choose the **Table** label option. Applying the **Caption Table** style to each table caption will give you consistent spacing throughout your document.

CHAPTER 1: INTRODUCTION

1.1 Introduction

The introduction chapter should set the scene and give a high- level problem statement/ specification, so that after reading the introduction the reader understands roughly what the problem is and what you intend to do about it. Is the idea to write software, or develop an algorithm, or produce hardware, or something else? Finally, you must briefly introduce the structure of report (what you will cover in which chapters and how these relate to each other).

This chapter can be presented in the following sub-sections: (1) introduction (2) background or motivation and (3) Problem Statement (4) Objectives of the Research project (5) Purpose Scope & Applicability (6) Organization of the Project Report (7) conclusion of the chapter

1.2 Background of the Study

You also introduce the reader to the benefits, advantages, and possibly novelty (originality) of your work. In order to achieve the amazing benefits your project has to offer, specific challenges must be overcome, both conceptual and technical. Those challenges can be mentioned here.

This chapter should provide the motivation for the project. What is the topic and why is it important? How does it fit into the broader world of your discipline of Computer Science or Information Technology?

1.3 Statement of the Problem

State the problem as simply as you can. It may be necessary to give a brief background of the problem and its implications if not attended to and the broader context in which those questions or problems are situated. You should then highlight and summarise that how your project will address the problem.

1.4 Objectives

Concise statement of the aims and objectives of the project. Define exactly what is going to be done in the project; the objectives should be about 30 /40 words. Start by using the phrase “The overall objective of this project is……”. Specific Objectives are:

1. “To research and investigate…..”
2. “To design…..”

You can break the overall objective in to three to four specific objectives.

1.5 Purpose, Scope and Applicability

The description of Purpose, Scope, and Applicability are given below:

- Purpose: Description of the topic of the project that answers questions on why this project is being done. How the project could improve the system its significance and theoretical framework.
- Scope: A brief overview of the methodology, assumptions and limitations. The students should answer the question: What are the main issues being covered in the project? What are the main functions of the project?
- Applicability: The student should explain the direct and indirect applications of their work. Briefly discuss how this project will serve the computer world and people.

1.6 Organisation of the Project

Summarizing the remaining chapters of the project report, in effect, giving the reader an overview of what is to come in the project report.

- Chapter One is the introductory part of the project showing why the project is undertaken. It also presents the problems, the purpose of the study, the scope and limitations.
- Chapter Two is the review of literature.
- Chapter three is Research Methodology. This chapter discusses the methodology of the research, the sources of data and the procedure for collecting the data, analysis and fixing the requirements specification.
- Chapter Four is System design and so on.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Every Chapter should start with an Introduction to the chapter and end with the Conclusion of that chapter.

This chapter describes the research the author has done in order to prepare for the project. The review can be presented in the following sub-sections: (1) introduction (2) related work and (3) previous systems (4) lessons learnt from the review (5) a critique of the review (6) conclusion of the chapter

Most important of all, ensure that in your review you have attempted to answer the following questions: Where did the problem come from? What is already known about this problem? What other methods have been tried to solve it?

2.2 Related Work

You should provide enough background to the reader for them to understand what the project is all about, and what is the relevant prior work.

The related work section demonstrates to the reader that you have done your homework (research), reviewed the previous literature, and now are ready to present your contribution based on what has been previously published. Examiners like to know that you have done the appropriate background research and it is important that you review either what has been done previously to tackle related problems, or perhaps what other products exist related to your deliverable. Clear references are important here.

One of the difficult aspects of the related work section is choosing the proper scope. There is some subjectivity in choosing which books or papers to refer to and also importantly, which previous literature not to refer to. This is something a supervisor is able to help with. Each previous publication you choose to refer to should get at least a one-paragraph description.

2.3 Previous Systems (or Similar Applications)

If you are building a piece of software, i.e., an application, then there's a good chance that other closely related applications already exist. This section describes those applications. For each previous system, all or part of the following information should be given:

- The name of the system or application,
- The URL of the system,
- A screen shot of the application, and
- Which platforms the application runs on,
- The duration time of the trial license, e.g., 30 days

A one or two paragraph description of the application including:

- Why the application was written
- Its target users.

Note that each screen shot should be accompanied by a figure number and caption, including a citation to the source of that screenshot, e.g., where the software comes from (including company name, web page author, web page title, URL, and last access date, etc.).

2.4 Citations

Any figure, image, equation, number, or year that is taken from another source must be cited. Content and terminology from other sources must also be cited. For more information about citations and their use, refer the Section REFERENCES and Appendices.

When referring to previous work use names. References should be accurate and complete, including things like page numbers, date accessedif it is an online document. Refer appendix VI for APA style of referencing.

2.4.1 Referencing Your Document's Figures

Each figure and table in your document must include a caption. A table caption goes above the table, while a figure caption goes under the figure. Instructions for creating captions are discussed in the List of Figures and List of Tables sections above.

You must refer to each table and figure in the body of the text. Here are some examples. "Figure 1-15 shows the screen that is displayed when the user chooses the *New Employee* option." "The ER diagram for the Employee class is shown in Figure 3-4." "Each test case corresponds to a use case in the functional requirements (see Table 6-2)."

Again, Word can help keep the figure numbers correct, even when you add figures or move them around. In your text, choose **Cross-reference**; in Word 2007+, this is under the **References** menu. Choose **Figure** for the reference type, and only label and number where it says "insert reference to".

2.4.2 Updating References

It is very easy to update all your tables of contents, item numbering, and the references to items all at once: select the whole document, right-click in it and choose **Update Field**.

2.5 Plagiarism

Plagiarism is presenting somebody else's work as your own work. It includes: copying information directly from the web or books without referencing the material; working with one or more other people on an individual project and submitting the joint project as your own individual effort; copying another student's project; paying someone else to do the work; stealing a project from another student and submitting it as your own work. The person you copy from could be another student, a lecturer or someone outside the university. The school takes plagiarism very seriously and all submitted reports will be checked for plagiarism electronically. The similarity index should not be more than 20%.

Plagiarism includes, but is not limited to:

- using published work without referencing (the most common)
- collaborating with any other person when the work is supposed to be individual
- taking another person's computer file/program
- submitting another person's work as their own
- the use of unacknowledged material published on the web
- copying another student's results
- falsifying results.

2.6 Reuse of programming code

In industry reuse of code is to be encouraged and both Web sites and books will provide numerous examples of code but, students should realise that part of the purpose of doing a programming project is to develop your skills. If most of the code comes from other sources then you will not be awarded a very high mark and you will have learnt very little.

If however you choose to make use of other people's code then in order to avoid an accusation of plagiarism, you must annotate their listing identifying the lines of code which are not your own. You must clearly state their source e.g. name of author, page in the book that they have taken the code from, Web page address. Failing to reference work taken from other sources is a plagiarism offence and will be dealt with as such.

2.7 Conclusion

Every chapter should have a conclusion explaining the activities achieved in that chapter.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

In this chapter the system development methodology is chosen and how you proceed with your project based on the methodology have to be explained. Apart from that you have to explain the methods used for Information gathering and analysis. If there are existing systems in place, system analysis should be carried out too. Then based on the information gathered and analysis the proposed system’s Requirements specification have to be specified. The proposed system can be a pure software project or it can be a Hardware cum software embedded project. This is related to the problem you have to tackle and the exact form of this will vary from project to project.

For example, any software project should include a discussion of the principles which underlie the program that has been written: the significance of its data structures, the way that its procedures and modules interact and the processes involved in discovering and documenting these requirements has to be discussed.

3.2 Methodology

This section should comprehensively describe the chosen system development life cycle (SDLC) methodology. Any sequential or iterative methodologies like Waterfall, Prototyping, Agile, Evolutionary, RUP etc could be chosen. Your approach inside the SDLC methodology can be Structured Analysis and Development (SAD) approach or Object-Oriented Analysis and Development (OOAD) approach. These approaches are further backed by tools, like OOAD is modelled using Unified Modelling Language (UML).

Some of the above methodologies can be used not only for software development but for Hardware project development as well. For example, as shown in Figure 1, Agile can be applied to hardware development. With Agile, both hardware and software features are broken down into smaller chunks – only the methodology is a bit different for each. Once software is working, it can be deployed either on any available hardware "modules", or in a test or simulation environment.

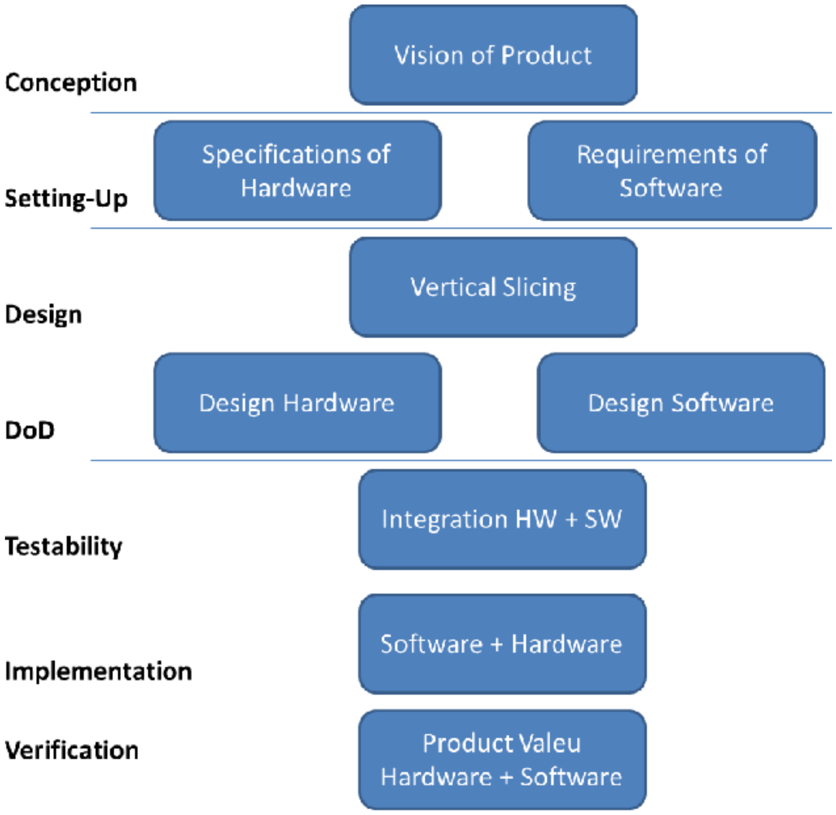


Figure 1: Proposed Phases-Agile and Co-Design

3.3 Information Gathering and Analysis

The section should explain the methods used in your data collection process. For instance, if you perform experimental tests on samples, conduct surveys or interviews or use existing data to form new studies, reading existing documents (literature review), etc.

Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the requirements of the various stakeholders, objectives and other external requirements. Then analyse and develop requirements specifications from the cconcept of ooperations and validate software or system requirements.

3.4 Requirements Specification

In this phase the student should define the requirements of the system, independent of how these requirements will be accomplished.

The requirements for a system are the descriptions of what the system should do—the services that it provides and the constraints on its operation. These requirements reflect the needs of customers for a system that serves a certain purpose such as controlling a device, placing an order, or finding information. The requirements specification can be distinguished as two main categories 'user requirements' to mean the high-level abstract requirements and 'system requirements' to mean the detailed description of what the system should do. This is in fact a formal point wise enumeration of the expectations from the proposed solution. The point wise nature entails that the solution might be evaluated according to those points. Students are therefore expected to elicit both user and system requirements. Refer Appendix VIII for more specifics on Requirements Specification.

Note: While the requirements analysis describes the problem, the requirements specification defines the desired characteristics of the solution.

3.5 System Analysis

It is a systematic approach, which uses graphical tools to analyze and refine the objectives of an existing system and develop a new system specification which can be easily understandable by user. SAD or OOAD approaches and related tools can be used for Analysis. The tools and techniques used for system analysis are

- UML
- ER diagrams
- Data Flow Diagrams (DFD)
- System Flowcharts etc.

For example, if an information system is involved in the project, include an ER diagram to depict the pieces of information that need to be handled and the relationships existing between them.

Note: The above tools can be used for the System Analysis and the Design of the System as well.

3.6 Conclusion

Every chapter should have a conclusion explaining the activities achieved in that chapter.

CHAPTER 4. SYSTEM DESIGN

4.1 Introduction

The examiners are interested in the engineering process you went through in performing your project work as the results you finally produced. So, make sure your report identifies what design choices have to be made, what were the possibilities, and why you made the particular choices and decisions that you did. They are looking for principled rational arguments and for critical assessment. The reasons for a design decision may be various, and may in some cases be out of your control. Explicit understanding of this, and the ability to communicate it, is important. Describe the architecture, the system modules, the interfaces, the database schema, important algorithms, system maintenance recommendations/requirements, etc.

4.2 Analysis of the System

Analysis emphasizes an investigation of the problem and requirements, rather than a solution. Which means, it focusses on *What the system should do?*

While Design emphasizes a conceptual solution (in software and hardware) that fulfills the requirements, rather than its implementation. Which means, it focusses on *How to accomplish the objectives of the system?*

In the design phase the Requirements Specification document is converted into a format that can be implemented and decides how the system will operate.

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization.

4.3 Context Model of the System

The System Context is the part of the environment of a system that influences the system. This was relevant for the collection of the requirements. Therefore, the first abstract model to depict the system is the context model. Draw the context model by following the steps

1. Place your system in the center of your context diagram.
2. Add all external entities around your system.
3. Add and specify data flows between your system and external entities.

4.4 Design Methods

The choice of the design methods varies from project to project. Some of them are;

4.4.1 Architectural Design

It is a high-level design to describes the views, models, behaviour, and structure of the system. The architecture diagram should depict the hardware and software (high-level only) modularity of the system, the client-server identification of modules, and other such high-level details. Keep in mind the system requirements and see that the architecture meets the requirements.

Note: All projects should include Architectural Design.

4.4.2 Detailed Design

It follows Architectural design and focuses on development of each module. A systemic approach is required for a coherent and well-running system. You can also explain in system modules, the interfaces, process diagrams, the database schema, important algorithms, pseudocode and other documentation following any of the approaches. Refer Appendix IX for Object Oriented Analysis and Design Approach.

Structured Analysis & Design (SAD) Approach	Object-Oriented Analysis and Design (OOAD) Approach
Modelling Tools used are Data Flow Diagram (DFD) & E-R diagram model the data.	*Modelling Tools used are Class diagram design, Sequence diagram, State Chart diagram, and Use Case diagram all contribute.*

Note: Students are therefore expected to model the requirements elicited in CHAPTER 3 and produce models using the various design tools.

4.4.3 Physical Design

Physical design relates to the actual input and output processes of the system. It focuses on how data is entered into a system, verified, processed, and displayed as output.

a) How users add information to the system and how the system represents information back to the user.

b) How the data is modelled and stored within the system.

c) How data moves through the system, how data is validated, secured and/or transformed as it flows through and out of the system.

It is concerned with user interface design, process design, and data design.

4.5 Conclusion

Every chapter should have a conclusion explaining the activities achieved in that chapter.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter shows the implementation details of the Project. It will also show the steps required to achieve the complete project. In this section all the implementation details are presented including the software and hardware used.

Describe the system set-up that is required for developing your solution, the components of your deliverables, the system set-up that is required to deploy the solution, installation procedure, user training requirements and status, testing of the system, trouble-shooting guidelines, guidelines for further work.

5.2 System Implementation

This chapter will describe in detail the software components, hardware components, the algorithms used in implementing the system etc.

- Choice of programming language or Algorithm: this should state programming language or algorithm used and justify its use for this particular student project
- Choice of hardware for hardware-based projects: you should justify the use of specific hardware and show the circuit diagrams for the hardware
- System cutover from the development architecture to the implementation architecture: this could be an extension of the technical architecture diagram. You can use arrows to show the linkage. Show what was moved from the development architecture to the implementation architecture in terms of code/hardware. The supporting narrative should explain why and how.
- Data migration from the development architecture and/or existing systems to the implementation architecture (only applicable to systems installed at a client's sites)
- Training; how, why and when particular groups of users will be trained
- Other project Issues if any (Optional)
  - Project Management – this section should include a Gantt Chart and supportive narrative that explains how the project was managed.
  - Risk Management – this section should overview the approach to risk management.
  - Configuration Management – this section should overview the approach to configuration management.

For example,
For a project that involves construction of a software system, this chapter will contain a description of the implementation, showing how it arises from the design, illustrating its output, outlining particularly interesting elements of the program and so on.
For a project concerned with developing new styles of user interface, this chapter will contain a description of the interfaces and of the experiments that were carried out to compare them.

5.3 Coding

Codes included should illustrate algorithmic flow, or highlight an interesting optimisation, demonstrate interactions with a data-structure, or give an example of input for a tool that has been designed. You should be able to explain what message or point a line of code in the fragment is conveying, and if you cannot justify it shouldn't be in the code fragment.
Note: Complete listings of code may be included as appendices of your report.

5.4 Results

Results are shown as screenshots. Screen-shots are sometimes used instead of drawing a picture or in order to capture the results of running a tool. They are also sometimes used as page filler, or as "proof" that a tool was launched and something compiled, which is not necessary. Use a screen-shot only if it is demonstrating a particular point, such as a particular interaction that is difficult/impossible to highlight, but make sure you edit and annotate the figure to show and highlight the important parts of results.

5.5 Conclusion

Every chapter should have a conclusion explaining the activities achieved in that chapter.

CHAPTER 6: EVALUATION AND TESTING

6.1 Introduction

The purpose of this chapter is to assess the functionality (functional and non-functional requirements) of the system and evaluate the approach taken. This section is normally useful for software or hardware deliverables and less relevant in analytical projects.

Testing: It is performed to check whether the deliverable (algorithm, program, or hardware) is a product with easily quantified performance. Does the software meet specifications? Does the hardware exhibit the expected behaviour? What are the different test cases? (See appendix 5) What sources of data will be used to test your system?
Performance Evaluation: How well does your application perform? How fast (or slow) is your application? How memory efficient (or inefficient) is your algorithm or system?

6.2 Testing Results

An accurate summary of the test scripts must be accompanied by the actual results when testing is complete and should be included in the Report itself. It is not necessary to show all test scripts and results in this section, they must be placed in the appendix. However, you can show some screen shots of the results and provide associated narratives about the results.

In this section, the images representing the must have functions of your system or algorithm are presented. Your system is applied to interesting data sets and the insight provided is described in addition to a concise description of the data sets.

Sometimes non-working designs are described in project reports as though they work, when in reality they don't, or only partially work. Therefore, a precise description of what works and *how this has been established* is important. Examiners may try to compile, use, or test deliverables themselves (even after your report is submitted), and your report should accurately reflect the state of the project.

6.3 Evaluation

It must contain your critical evaluation of your work as compared to previous analysis, algorithms, products, and when related to your original objectives.
To write an evaluation you must provide answers to the following:

- To what extent have your original objectives been fulfilled?
- What are the advantages, disadvantages of your approach compared with related work?
- How does the scope of your work differ from related work?
- What was the most difficult and/or clever part of the project?
- Was the development process model used appropriate (if not, why not and what else should have been used)?
- Was the programming language (if used) suitable?
- What difficulties did you face and how did you overcome them?
- What have you learnt and experienced from doing the project?
- How do you recommend the project should be taken forward in the future?

6.4 Conclusion

Every chapter should have a conclusion explaining the activities achieved in that chapter.

CHAPTER 7: CONCLUSION AND RECOMMENDATIONS

This chapter should include an assessment of the success of the finished product. Have you achieved your objectives? If not, why not? It should also contain suggestions for future extensions, or alternative methodologies that, with hindsight, might have led to a better system.

7.1 Conclusion

This section summarizes the project work carried out and lists the resulting advantages. Remind the reader what the original problem to be solved was. Restate the specific objectives of the project and undertake a critical analysis of how well these objectives have been met.

1. Conclusion is written to relate directly to the main objective /specific objectives of the project stated in the introduction chapter.
2. Indicate the extent to which the aims have been achieved.
3. Summarise the key findings, outcomes or information in your report.
4. Acknowledge limitations and make recommendations for future work (where applicable)

7.2 Recommendations

In this subsection, make recommendations for future work on the topic. You should offer suggestions for further work. If you had ideas that you didn't have time to explore or implement, this is your chance to tell the examiners. If you didn't achieve your original objectives, this is your chance to describe how you might do things differently if you were starting over.

For example, give answers for the following questions. Does your work suggest any other interesting avenues? Are there ways in which your work could be improved by future workers? What are the practical implications of your work?.

REFERENCES

A reference list is a complete list of references used in a piece of writing including the author name, date of publication, title and more. This list is provided after the last chapter of the report.

As you write sections of your report, it is a requirement to provide in-text references using APA referencing. In-text references must be included following the use of a quote or paraphrase taken from another piece of work. In-text citations are citations within the main body of the text and refer to a direct quote or paraphrase. They correspond to a reference in the main reference list. These citations include the surname of the author and date of publication only.

You encouraged to cite literature that has been published in the last five years from books and journal articles. If you cite a journal article or book, the reader can go to a library and check the cited document to confirm whether or not it says what you say it did. Therefore, ensure that your references are correct and specific.

Should you reference websites? If so, how should it be done? A website may disappear, or it may have its content updated or changed completely. Nevertheless, websites are very useful sources of information. You should give the URL and the date you accessed it. If there is a date on the site itself (last updated on …), you should include that as well.

Detailed formatting style of APA is shown in the Appendices VI

APPENDICES

If there is material that should be in the report but which would break up the flow or bore the reader unbearably, include it as an appendix. Make references in the body of the report to appended material. Some items that should be included in the Appendices are:

- The original project proposal – you may just copy and paste it here without modification.
- Installation manual – include the steps to guide the operator in installing the program. Screen dumps of confusing steps are helpful.
- User manual – include how to start the system, how to run it (for example, adding records to a database, running queries, and setting up security for the database administrator). Also include screen dumps of important components of the working program, for example, data entry, queries, and reports.
- Sample code – include important code. You must include code that solves the key problem, server code, and input validation. You may include any other code that you

think is important to show. You must include explanations if the code is not well commented. Do not include auto-generated code, for example, C# code auto generated by .net framework.

- Questionnaires – if you used them.
- Data files – if you expect any to be particularly helpful to the reader in understanding your project.

FINAL DEFENSE GUIDELINES

This is the finished, fully-functional product. You will demonstrate it in action to your supervisor and the department faculty at the end of the project (CS301 teams will demonstrate to their supervisor only). It will then be "handed over" to the department. You must also upload the system onto the University system (opus) with your final bound report.

In addition to writing the report, you will be assessed on your ability to present your project.The presentation procedure will apply to all the groups i.e. Computer Science, Computer Engineering and Information Systems.

Your final year project would be presented through three main items, namely, *(i) Hardware/Software demonstrations, (ii) Power point presentation, and (iii) Project Report.* Project report is a key document that shows how you have accomplished your project. It presents your understanding on the project, explains the project plan, provides the results of each stage on your project, describes the outcome, and presents the evaluation and your conclusion on your final year project.

In fact, the project report is one of the main deliverable items based on which your project would be evaluated and marketed. Refer Appendix II for the Template of Project Report. However, a general outline of a typical project report was presented in the previous chapters. It is crucial to pay attention to all the details that you have to present.

Tips on presenting software code in your report

- Include those parts of code that is your work.
- Make sure that you have properly commented the codes.
- Provide a title that represents the main functionality of the code.
- If you have customized an open source, make sure that you have properly referenced its origin.
- Do not include parts of code, which are automatically generated by the tool you have used for the implementation.

Test Documents

- Provide your test cases, test data, and test results as evidence of your testing process.
- Categorize test documents based on the guidelines, which were provided in previous chapters.
- Do not forget to provide this document as a supplementary document to show what has happened during your long journey through your final year project.

Electronic Documents

Obviously, in almost the entire computing projects your documents and products are mainly in electronic format. Your final year project is exceptional in this regard. According to your department procedures or the final year project, you might be asked to deliver your project on a physical medium, and electronic medium, or both. Regardless of the format, paying attention to the preparation of these documents and organizing them properly have a great impact on your project evaluation. A link to uploading files / projects on opus shall be provided so that you upload your electronic medium.

If you are asked to prepare a physical medium on which you should provide your final year project, make sure that you have paid utmost attention in preparing this item. Sometime, because of the procedures, your physical medium could not operate properly it would be considered as incomplete project. This may cause you a severe penalty in your project evaluation.

Tips on using Non-Physical Medium for the project submission

- Prepare a folder based on the template the department shall give you.
- Make a compressed file out of which (e.g. zip, rar, etc.).
- Check your file by uncompressing it in another location.
- If it is successful, then upload your file to the electronic postbox (opus), otherwise fix the problem and repeat this step.
- Do not forget to save any feedbacks or receipts showing that your document has been received, as evidence.

Presentation

Presentation is the way that you are going to sell your products. You might face with different situations. All departments ask students to present the outcome of their projects in front of a panel of academic staff in a way that is expected. Presentation is an activity, which takes place in a short period, usually between 10 to 20 minutes. Therefore, its structure, preparation, contents, and delivery should be very well designed and implemented. Presentation, provides you with a unique opportunity within which you are able to introduce your capabilities on the subject, understanding of the chosen topics, and presenting the results and outcomes of your project. This is an important event. Do not forget that this activity must have been planned and scheduled in your project plan and you should be very careful on the timing of the event.

Proposed Presentation Structure: (Presentation time: 20 minutes)

- Preparation: 5 minutes
- Questions/Answers: 10 Minutes
- Demonstration: 5 minutes

12 slide Presentation Structure:

Slide 1: Project information (15 seconds)
Slide 2: Agenda/Topics (15 seconds)
Slide 3: Problem Statement/Project Definition (30 seconds)
Slide 4: Main Requirements (30 seconds)
Slide 5: Literature Review (1.5 minutes)
Slide 6: Methodology (30 seconds)
Slide 7: Analysis (1 minute)
Slide 8: Design (2 minutes)
Slide 9: Implementation (1 minute)

Slide 10: Findings and Evaluation/ Links to live system or other documents (1.5 minutes
Slide 11: Conclusion and Future Works (1 minute)
Slide 12: Thank you and Q/A

CONCLUSION

We hope that this document will help you to create an excellent project and report. For questions related to the project that are not specified in this document, please consult your advisor. Good luck!

APPENDICES

Appendix 1-Project Proposal Template

Cover Page

1. Introduction
1.1. Background Information
1.2. Problem Statement
1.3. Research Objectives
1.4. Hypothesis/hypotheses Or Assumption
1.5. Scope of the study

2. Literature review
2.1.
2.2.

3. Research Methodology
3.1.
3.2.

4. Significance of the study
4.1. Expected contribution and implications of the study

5 Ethical Issues in the research

6. Project Timeline/Gantt Chart

7. Financial Implications/Budget

REFERENCES
(APA format)

Appendix II- Final Project Report Template

Front Page
Abstract
Student Declaration,
Dedication
Acknowledgement
Table of Contents
List of Figures
List of Tables

CHAPTER 1: Introduction of Topic/ Motivation
- 1.1 Introduction
- 1.2 Problem Statement
- 1.3 Objectives
  - 1.3.1 Specific Objectives
- 1.4 Scope & Limitation
- 1.5 Organization of the Report
- 1.6 Conclusion

CHAPTER 2: Review of Literature/ Literature Review
- 2.1 Introduction
- 2.2 Related work
- 2.3 previous systems
- 2.4 Lessons learnt from the review
- 2.5 A critique of the review
- 2.6 Conclusion

CHAPTER 3: Research Methodology
- 3.1 Introduction
- 3.2
- …
- 3.6 Conclusion

CHAPTER 4: System Design
- 4.1 Introduction
- 4.2
- ….
- 4.8 Conclusion

CHAPTER 5: System Implementation
- 5.1 Introduction
- 5.2
- …
- 5.5 Conclusion

CHAPTER 6: Evaluation and Testing
- 6.1 Introduction

6.2
…
6.5 Conclusion

CHAPTER 6: Conclusion & Recommendation
6.1 Conclusion
6.2 Recommendations

REFERENCES
(APA format)

APPENDICES
Appendix 1: The original project proposal,
Appendix 2: Installation Manual
Appendix 3: User Manual
Appendix 4: Sections of Program Code
Appendix 5: Questionnaires – if you used them
Appendix 6: Data Files if any
Appendix 7: Component Specifications if any

APPENDIX III-Attendance Register Final year

COPPERBELT UNIVERSITY

SCHOOL OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

CS 400, IS 400, DIT 590

FINAL PROJECT ATTENDANCE REGISTER

NAME:

STUDENT NO:

DEPARTMENT:

SUPERVISOR:

	DATE	STUDENT SIGNATURE	SUPERVISOR SIGNATURE	SUPERVISOR COMMENT
1.				
2.				
3.				
4.				
5.				
6.				
7.				
8.				
9.				
10.				
11.				
12.				

APPENDIX IV-Attendance Register Third year

COPPERBELT UNIVERSITY

SCHOOL OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

CS, CE AND IS THIRD YEAR

GROUP PROJECT ATTENDANCE REGISTER

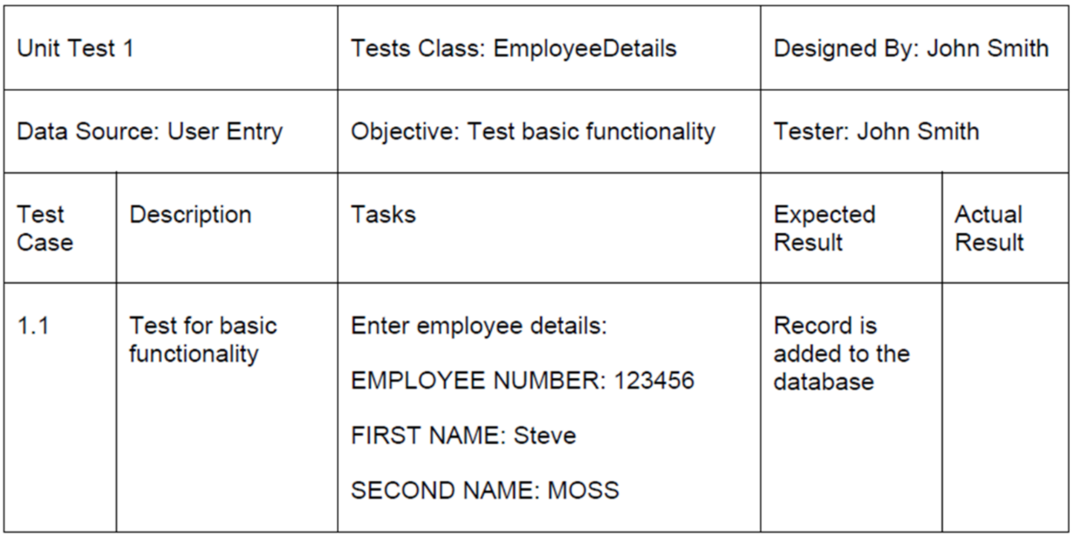
DEPARTMENT:

SUPERVISOR:

NAME OF GROUP:

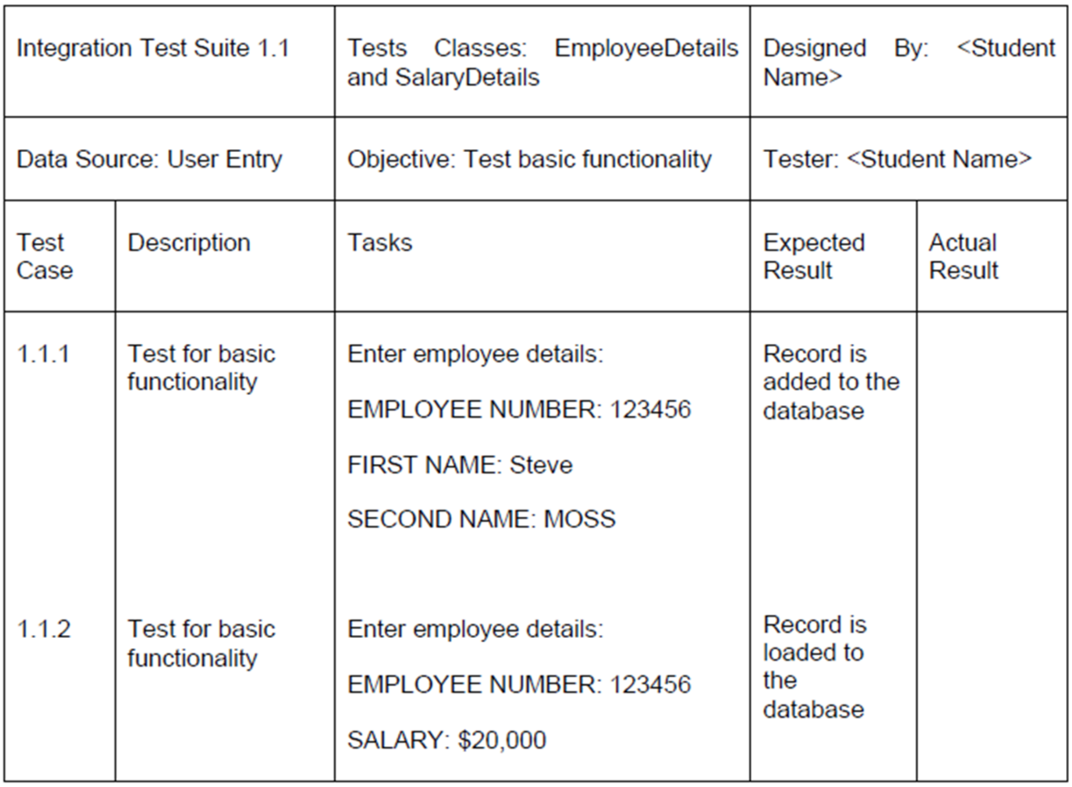
	DATE	STUDENT SIGNATURE	SUPERVISOR SIGNATURE	SUPERVISOR COMMENT
1.				
2.				
3.				
4.				
5.				
6.				
7.				
8.				
9.				
10.				
11.				
12.				

Appendix V- Sample Test Cases



Unit Test 1		Tests Class: EmployeeDetails	Designed By: John Smith	
Data Source: User Entry		Objective: Test basic functionality	Tester: John Smith	
Test Case	Description	Tasks	Expected Result	Actual Result
1.1	Test for basic functionality	Enter employee details: EMPLOYEE NUMBER: 123456 FIRST NAME: Steve SECOND NAME: MOSS	Record is added to the database	

Figure 1 Sample Unit Test Script



Integration Test Suite 1.1		Tests Classes: EmployeeDetails and SalaryDetails	Designed By: <Student Name>	
Data Source: User Entry		Objective: Test basic functionality	Tester: <Student Name>	
Test Case	Description	Tasks	Expected Result	Actual Result
1.1.1	Test for basic functionality	Enter employee details: EMPLOYEE NUMBER: 123456 FIRST NAME: Steve SECOND NAME: MOSS	Record is added to the database	
1.1.2	Test for basic functionality	Enter employee details: EMPLOYEE NUMBER: 123456 SALARY: $20,000	Record is loaded to the database	

Figure 2 Sample Integration Test Script

Appendix VI- APA Referencing Format

In APA Style Using an example author James Mitchell, this takes the form given for a single author.

Single author	Mitchell (2017) states… Or …(Mitchell, 2017).
Two Authors	Mitchell and Smith (2017) state… Or …(Mitchell & Smith, 2017).
Three to five authors	For the first cite, all names should be listed: Mitchell, Smith, and Thomson (2017) state… Or …(Mitchell, Smith, & Thomson, 2017). Further cites can be shorted to the first author's name followed by et al: Mitchell et al (2017) state… Or …(Mitchell et al, 2017).
Six or more authors	Only the first author's surname should be stated followed by et al, Mitchell et al (2017) state… Or …(Mitchell et al, 2017).
No Author	If the author is unknown, the first few words of the reference should be used. This is usually the title of the source. (A guide to citation, 2017).
A Group or Organisation	First cite: (International Citation Association, 2015) Further Cites: (Citation Association, 2015)

An APA reference list must:

- Be on a new page at the end of the document
- Be centred
- Be alphabetically by name of first author (or title if the author isn't known, in this case a, an and the should be ignored)
- If there are multiple works by the same author these are ordered by date, if the works are in the same year they are ordered alphabetically by the title and are allocated a letter (a,b,c etc) after the date
- Contain full references for all in-text references used

Formats for specific types of documents follow:

Books

Jones, A.F & Wang, L. (2011). *Spectacular creatures: The Amazon rainforest* (2nd ed.). San Jose, Costa Rica: My Publisher

Mitchell, J.A., Thomson, M., & Coyne, R.P. (2017). *A guide to citation*. London, England: My Publisher

Articles/Chapters in Book

Author of Part, A. A. (Year). Title of chapter or part. In A. A. Editor & B. B. Editor (Eds.), *Title: Subtitle of book* (edition., inclusive page numbers). Publisher.

Articles in Periodicals (journals, magazines, etc.)

Mitchell, J.A. (2017). Citation: Why is it so important? *Mendeley Journal, 67*(2), 81-95

Mitchell, J.A. (2017). Citation: Why is it so important? *Mendeley Journal, 67*(2), 81-95. Retrieved from https://www.mendeley.com/reference-management/reference-manager

Papers Published in Proceedings

Author, A., & Author, B. (Year, Month date). Title of session [Paper presentation]. In A. Editor, & B. Editor. *Title of Published Proceedings*. Title of Conference: Subtitle of Conference, Location (inclusive page numbers). Publisher.

Article in an Electronic Journal

Article Author, A. A., & Article Author, B. B. (Year). Title of article. *Title of Journal, volume number*(issue number), inclusive page or paragraph numbers. DOI

Website

Mitchell, J.A. (2017, May 21). *How and when to reference*. Retrieved from https://www.howandwhentoreference.com.

Appendix VII- Functional and non-functional requirements

System Requirements Specification may have the following sections and subsections.

Functional Requirements

These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do. Functional system requirements vary from general requirements covering what the system should do to very specific requirements reflecting local ways of working or an organization's existing systems.

- Describe each functionality of the system one at a time;
- A functionality can be an information processing functionality involving some mathematical functions;
- A data input/output/transfer functionality;
- Special processing functionality for system maintenance, etc.
- A data storage requirement can be expressed in terms of appropriate input and output functions.

Non-functional requirements

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements often apply to the system as a whole, rather than individual system features or services.

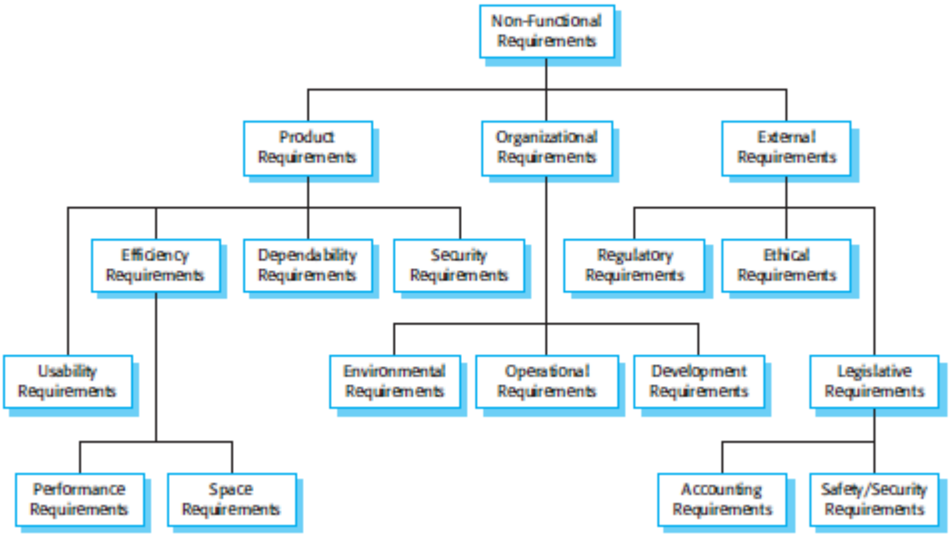


Figure 3.3 ; Non-Functional Requirements Classification

Figure 3.2 is a classification of non-functional requirements.

Performance Requirements

These requirements specify or constrain the behaviour of the software. Examples include performance requirements on how fast the system must execute and how much memory it requires, reliability requirements that set out the acceptable failure rate, security requirements, Software Quality Attributes and usability requirements.

Software and Hardware Requirements

Define the details of all the software and hardware needed for the development and implementation of the project.

- Hardware Requirement: In this section, list the hardware units that have been selected for the system. The components such as various sensors display cards, microprocessors, storage capacity, etc. necessary to run the software must be noted.
- Software Requirements: In this section, describe the software features over which your system shall run which OS, which DBMS, the compiler, testing tools, linker, and the libraries etc. necessary to compile, link and install the code/software must be listed.
- System Development Platform: Describe the platform that have been used to develop the system. This includes, hardware units, programming environment (including choice of compilers), DBMS, software development tools (e.g., front-end tools), etc.

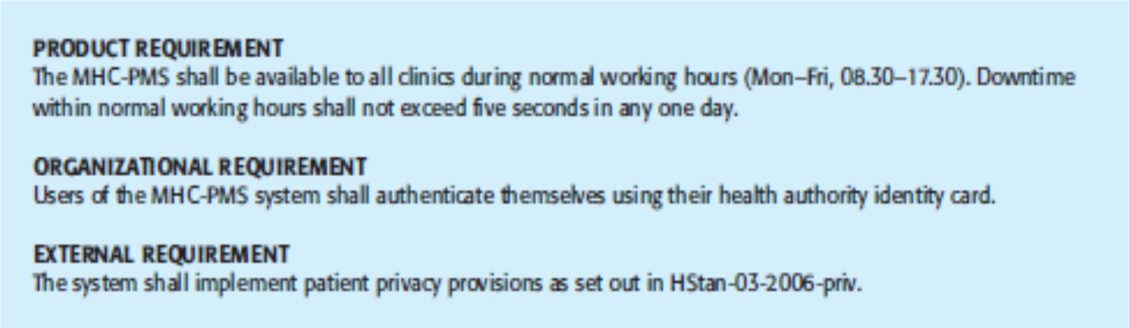
Preliminary Product Description

Identify the requirements and objectives of the new system. Define the functions and operation of the application/system the students are developing as project. Suitable diagrams may be used here.

Organizational requirements: These requirements are broad system requirements derived from policies and procedures in the customer's and developer's organization. Examples include operational process requirements that define how the system will be used, development process requirements that specify the programming language, the development environment or process standards to be used, and environmental requirements that specify the operating environment of the system.

External requirements: This broad heading covers all requirements that are derived from factors external to the system and its development process. These may include regulatory requirements that set out what must be done for the system to be approved for use by a regulator, such as a central bank; legislative requirements that must be followed to ensure that the system operates within the law; and ethical requirements that ensure that the system will be acceptable to its users and the general public.

Figure 3.3 shows examples of product, organizational, and external requirements taken from the MHC-PMS whose user requirements were introduced in Section 3.3.



PRODUCT REQUIREMENT

The MHC-PMS shall be available to all clinics during normal working hours (Mon–Fri, 08.30–17.30). Downtime within normal working hours shall not exceed five seconds in any one day.

ORGANIZATIONAL REQUIREMENT

Users of the MHC-PMS system shall authenticate themselves using their health authority identity card.

EXTERNAL REQUIREMENT

The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

Figure 4: Examples of non-functional requirements for the MHC-PMS

Appendix VIII- Object Oriented Design Approach

Example: This section uses the Radio Frequency Identification (RFID) student attendance system to help comprehend the modelling.

Systems Models Overview

System models are used in comprehending the functionalities of the system which can encompass the behavioural perspective depicting the various behaviour of the system, the structural perspective illustrating the data architecture of the system and the external perspective which shows the system context. The system models covered for the Radio Frequency Identification (RFID) student attendance system include the context model which shows the operational boundaries of the Radio Frequency Identification (RFID) student attendance system, the sequence diagrams which depicts the various interactions between the actors, the system and system components of the Radio Frequency Identification (RFID) student attendance system, the activity diagrams which illustrates the activities involved in the processes of the Radio Frequency Identification (RFID) student attendance system, data flow diagrams show how data flows from one process to another to generate a certain output in the Radio Frequency Identification (RFID) student attendance system and object oriented models which illustrates the object classes and the association of the object classes in the Radio Frequency Identification (RFID) student attendance system.

Context Model

The operational boundaries of the Radio Frequency Identification (RFID) student attendance system for Binary University will encompass the finance management system, the student management system, lecturer management system and class scheduling management system. The finance management system will provide student financial details that will be used to prevent the student who have not settled their school fees from accessing the Radio Frequency Identification (RFID) student attendance system. The student management system will provide the existing student details while the lecturer management system will provide the lecturer information such as staff identification, lecturer names, subjects take by distinct lecturer and any other relevant information patterning to attendance. The class scheduling management system on the other hand will provide the exact location and time a particular class will be taking place. The system context for the Radio Frequency Identification (RFID) student attendance system is as shown in figure below.

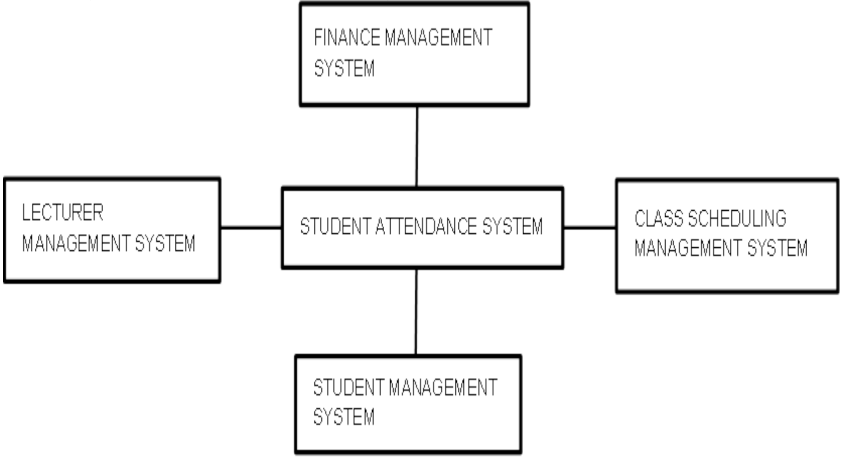

FINANCE MANAGEMENT SYSTEM
LECTURER MANAGEMENT SYSTEM
STUDENT ATTENDANCE SYSTEM
CLASS SCHEDULING MANAGEMENT SYSTEM
STUDENT MANAGEMENT SYSTEM

Figure: Context Model-RFID Student Attendance System

Use Case Model

Figure 4.2 illustrates the use case model for the Radio Frequency Identification (RFID) student attendance system which will be web based depicting the actors as student, lecturer and administrator respectively. The subsequent tables show the descriptions of these use cases in terms of actors, use case data, stimulus and response.

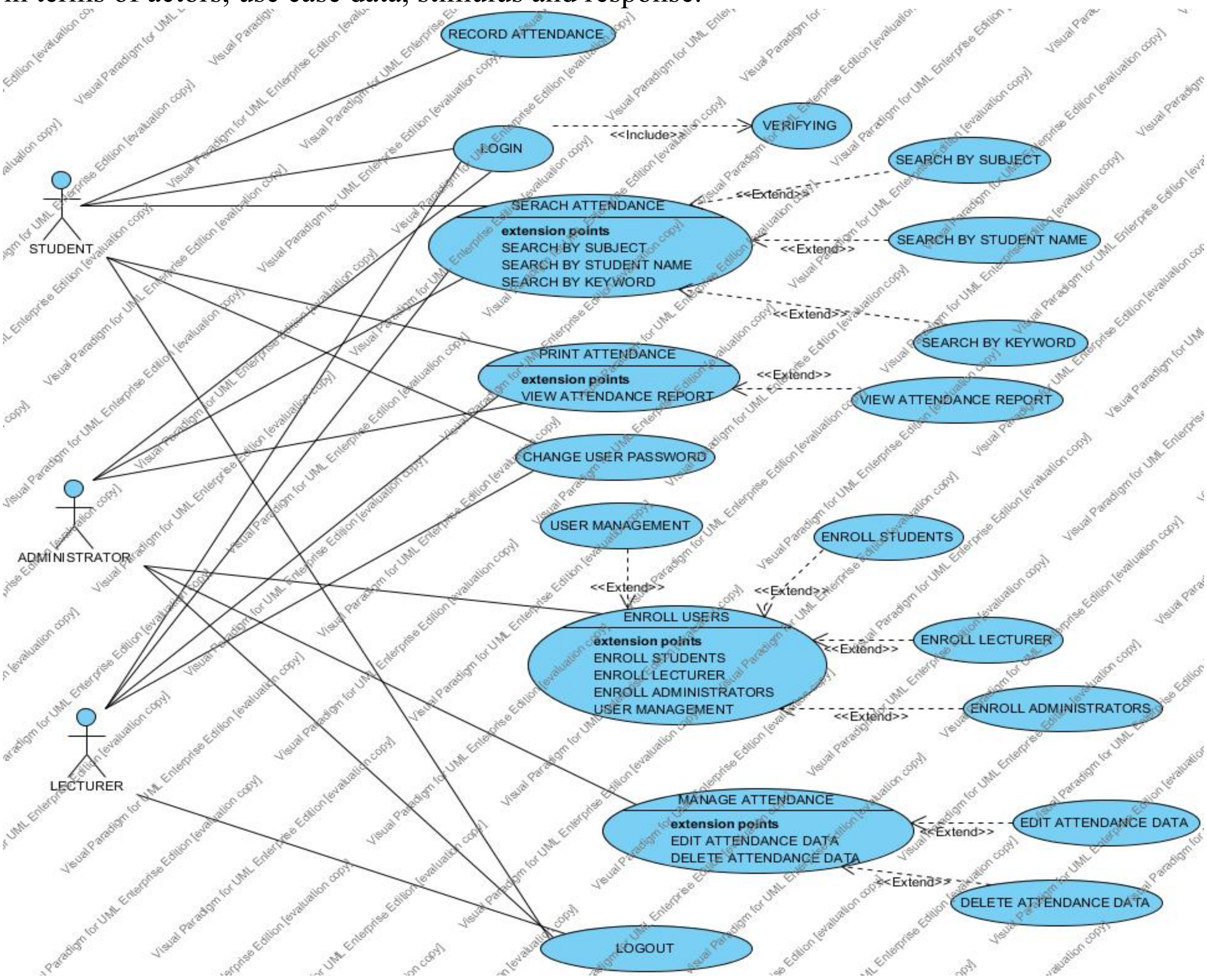


Figure: Use Case Model-RFID Student Attendance System

TABLE 1: RECORD ATTENDANCE USE CASE DESCRIPTION

SYSTEM	Student Attendance System
USE CASE	Record Attendance
ACTORS	Student
DATA	The RFID reader read the presented RFID tag and sends the Tag details for authentication against the details in the attendance database and the authenticated attendance details are then saved in the attendance database.
STIMULUS	The student presents their RFID tag to invoke the attendance recording process.
RESPONSE	The RFID reader reads the presented RFID tag and sends the Tag details for authentication against the details in the attendance database and the attendance is recorded against the

	authenticated details and the System produces a feedback either by LED light or Text via the RFID reader.

TABLE 2: LOGIN USE CASE DESCRIPTION

SYSTEM	Student Attendance System
USE CASE	Login
ACTORS	Student, Lecturer, Administrator
DATA	The user enters their username and corresponding password in the user interface which is then sent for authentication and authorisation with the predefined user rights embedded according to the accessibility levels.
STIMULUS	The user enters their login credential in the user interface.
RESPONSE	The system authenticates and authorises the user according the predefined user rights and provide a login success or login failure feedback.

TABLE 3: SEARCH ATTENDANCE USE CASE DESCRIPTION

SYSTEM	Student Attendance System
USE CASE	Search Attendance
ACTORS	Student, Lecturer, Administrator
DATA	The user enters their search details either by subject, name or keyword which is then validated by the user interface and the validated details queried from the attendance database.
STIMULUS	The user enters their preferred search details either by name, keyword or subject in the user interface.
RESPONSE	The system returns the search details and displays the search results on the user interface.

TABLE 4: PRINT ATTENDANCE USE CASE DESCRIPTION

SYSTEM	Student Attendance System
USE CASE	Print Attendance
ACTORS	Student, Lecturer, Administrator
DATA	The user selects the particular attendance report and the system displays the selected attendance report on the user interface. The user then request for printing by invoking the print option displayed in the user interface.
STIMULUS	The user invokes the print option in the user interface.
RESPONSE	The system provides a print feedback displayed on the user interface.

TABLE 5: CHANGE PASSWORD USE CASE DESCRIPTION

SYSTEM	Student Attendance System
USE CASE	Change Password
ACTORS	Student, Lecturer

DATA	The user enters their preferred new password in the change password form displayed in the user interface which is then validated and save in the database.
STIMULUS	The user enters their desired new password in the provided change password form displayed in the user interface.
RESPONSE	The system validates the entered new password and the validated password saved in the attendance database. The system then provides a password change feedback which is displayed on the user interface.

TABLE 6: ENROL USERS USE CASE DESCRIPTION

SYSTEM	Student Attendance System
USE CASE	Enrol Users
ACTORS	Administrator
DATA	The administrator enter the new user details in the new user form displayed on the user interface depending on the new user accessibility rights whether student, lecturer or administrator which is then saved in the database.
STIMULUS	The administrator enters the new user details in the new user form displayed in the user interface.
RESPONSE	The system validates the entered new user details and the validated details are then saved in the attendance database.

TABLE 7: MANAGE ATTENDANCE USE CASE DESCRIPTION

SYSTEM	Student Attendance System
USE CASE	Manage Attendance
ACTORS	Administrator
DATA	The administrator requests for the manage attendance form which is then displayed on the user interface. The administrator then edits or deletes the attendance data and the managed data changes are then saved on the attendance database.
STIMULUS	The administrator requests the manage attendance form and fills the form.
RESPONSE	The system displays the manage attendance form and validates the entered attendance data and the changes are then saved in the attendance database. The system then displays a manage attendance success feedback on the user interface.

Table 8: LOGOUT USE CASE DESCRIPTION

SYSTEM	Student Attendance System
USE CASE	Logout
ACTORS	Student, Lecturer, Administrator
DATA	The user invokes the logout process by selecting the logout option in the user interface and the session is ended.
STIMULUS	The user invokes the logout process through the logout option in the user interface.

RESPONSE	The system ends the user session and displays the logout success message on the user interface.

Sequence Diagrams

The interactions between the actors of the Radio Frequency Identification (RFID) attendance system, the Radio Frequency Identification (RFID) attendance system and the Radio Frequency Identification (RFID) attendance system components are as illustrated in the subsequent sequence diagrams.

Record Attendance Sequence Diagram

The student invokes the read tag method by presenting the RFID tag. The RFID reader reads the presented RFID tag and sends the Tag details for authentication against the details in the attendance database through the RFID application logic. The database returns the queried RFID tag details and the RFID application logic then authenticates the information and returns the authentication status. The RFID reader then sends a request to record the attendance against the authenticated tag details which is then saved in the attendance database and a corresponding attendance feedback displayed on the RFID reader by means of an LED light or text message on the RFID reader screen.

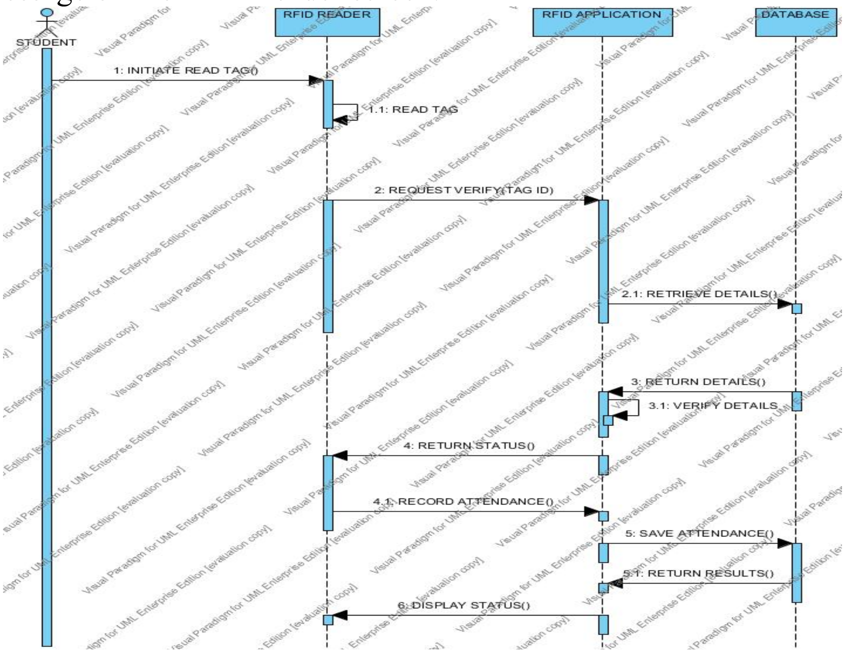


Figure: Record Attendance Sequence Diagram

ACTIVITY DIAGRAMS

RECORD ATTENDANCE ACTIVITY DIAGRAM

The RFID reader reads the presents RFID tag through the backscattering process and sends the Tag details to the RFID application that then invokes the authentication process which then retrieves the corresponding tag details from the attendance database and check the validity of the tag details. If the presented tag details are valid, the RFID application invokes the process to record the attendance against the authenticated tag details and save the attendance in the attendance database. However, if the presented tag details are not valid, the RFID application discards the whole process as shown in figure below.

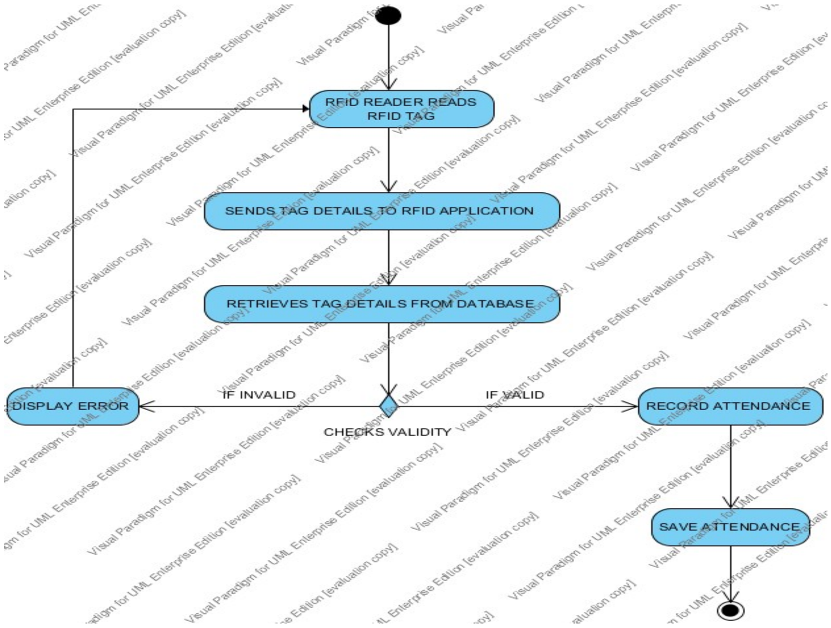

RFID READER READS RFID TAG
SENDS TAG DETAILS TO RFID APPLICATION
RETRIEVES TAG DETAILS FROM DATABASE
DISPLAY ERROR
IF INVALID
IF VALID
CHECKS VALIDITY
RECORD ATTENDANCE
SAVE ATTENDANCE

Figure: Record Attendance Activity Diagram

ATTENDANCE RECORDING PROCESS DATA FLOW DIAGRAM

The RFID reader reads the presented RFID tag and sends the tag details to the RFID application that invokes a process to retrieve the corresponding details for the presented tag ID defined in the attendance database. The RFID application then verifies the presented tag details against those in the database. The method to capture the attendance then records the attendance against the verified tag details and saves the attendance in the database as shown in figure below.

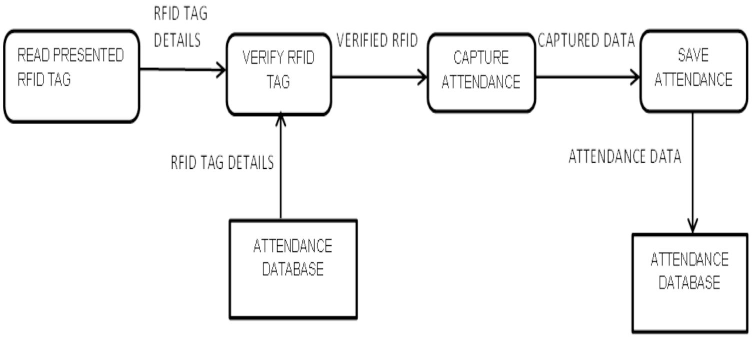


Figure: Attendance Recording Data Flow Diagram

OBJECT ORIENTED MODELLING

The object oriented is classified into two distinct categories i.e. the collaboration model depicting the RFID attendance application process and the inheritance hierarchy for the distinct system users for the web based RFID attendance application as illustrated in figures below. The RFID reader object will have a static IP address with methods to read presented RFID tag, send attendance details, provide attendance recording feedback by means of LED light or text displayed on the RFID reader screen. The Attendance application logic object will have date and time variables and methods to validate the attendance data, compute attendance and record the attendance. The report generator variable will have the date and report type variables and the method to generate the attendance reports defined as daily, weekly, monthly, semester, yearly etc. The attendance data object will have the date variable and methods to collect the attendance data and summaries the attendance data as shown in figure below.

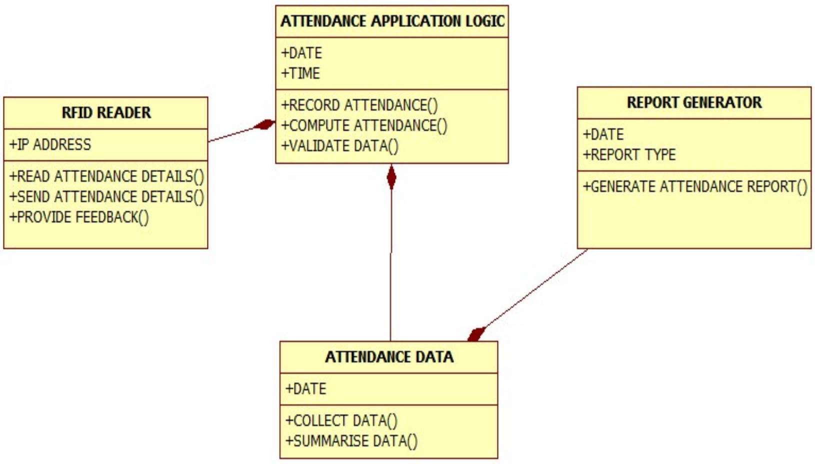


Figure: Attendance Application Process Object Oriented Model

On the other hand, the system users object model shown in above figure has the following objects. The system users object will have name, username and password variables with methods to login, search attendance, view attendance details, print attendance reports and logout which will be inherited by all the users i.e. students, lecturers, administrators. The student object will have unique attributes such as student ID, school, cohort, year, program and a method to change password. The lecturer object will have the unique attributes such as staff ID and a method to change password. The administrator on the other hand will have attributes such as admin ID and methods to edit user data, delete user data, enroll new users, manage attendance reports and manage users.

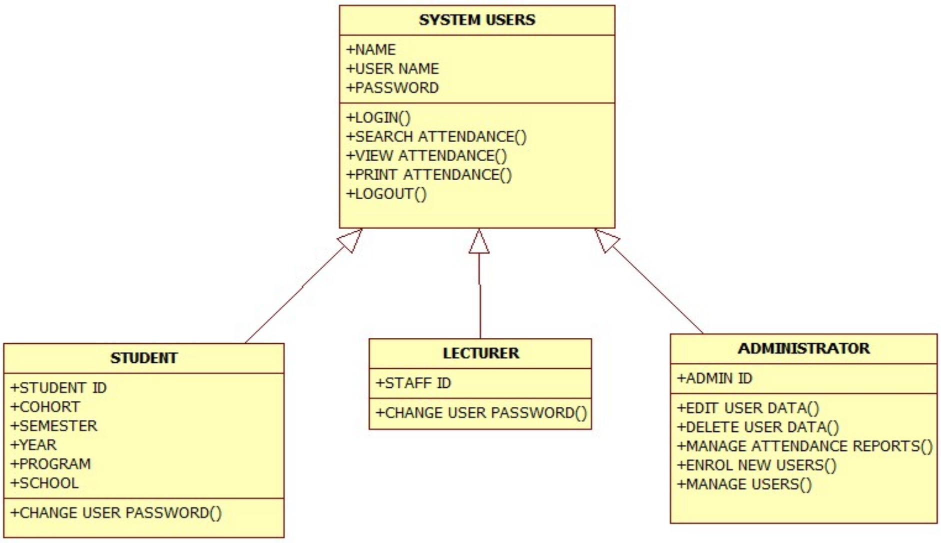


Figure: Object Oriented Model-System Users

PURPOSE OF RFID SYSTEM ARCHITECTURE DESIGN

The purpose of this architecture design is to identify the subsystems. This section also uses the RFID student attendance system to help elucidate the architectural design. The framework for subsystem control and communication are to attain comprehensive software architecture for the web-based RFID student attendance system.

SYSTEM ARCHITECTURE ORGANISATION

The system organization can be according to the client-server architecture which can be realized as 3-tier client-server architecture. The system can also be realized as a layered architecture. The user interface, user interface management, applications and system utilities as well as system support (attendance database) illustrated in a layered architecture. Other appropriate system organization architectures include the repository model and the model view controller architecture as explained below.

CLIENT SERVER ARCHITECTURE

In this client server architecture, the functionality of the RFID student attendance system is organized into services which are delivered by the application server and the attendance data server. The clients (RFID readers) will be installed at every classroom and will be distributed over the TCP/IP network and will access the services of the system provided by the application server and the attendance server which will be implemented in 3-tier client server architecture as illustrated in figures below respectively.

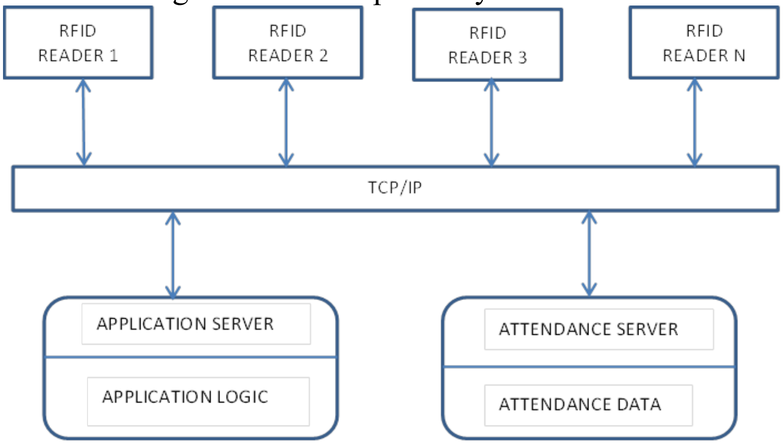


Figure: RFID Client Server Architecture

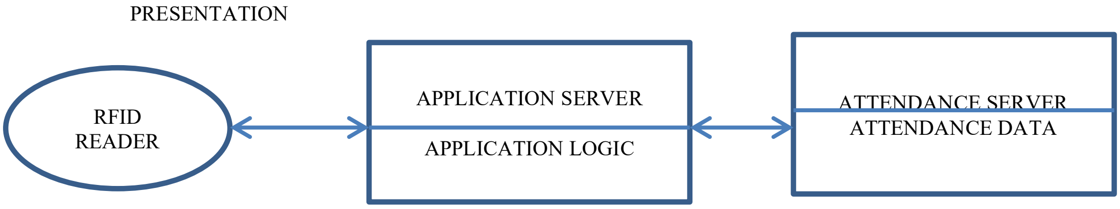


Figure: RFID 3-Tier Client Server Architecture

LAYERED ARCHITECTURE

The functionality of the RFID system is organized into layers with related functionality associated with each layer. The lowest layer will encompass the system support software mainly the attendance database. The next layer will be the application layer encompassing the search functionality, the view functionality, print functionality, the report manager and the security management.

The third layer will include user interface management notably login functionality (authentication and authorization), form management, query management and attendance data validation. The top layer will consist of the user interface (web browser interface) as shown in figure above.

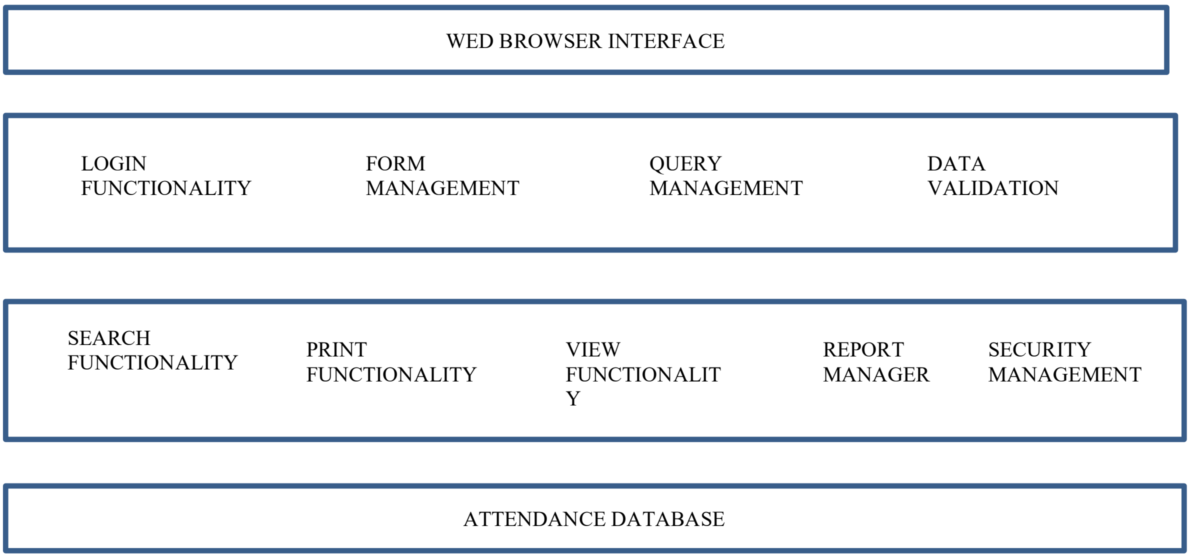


Figure: RFID Layered Architecture

REPOSITORY ARCHITECTURE

The repository architecture will encompass the shared repository (attendance database) that will be accessed by the RFID detection subsystem that will include the RFID reader and processes to read the presented RFID tag and send the read details for authentication against those defined in the database. Other subsystems will include the report generator subsystem to generate the various attendance reports (daily, weekly, monthly, semester, yearly etc.) and the attendance recording subsystem for recording attendance for the authenticated RFID tag and saving the attendance in the attendance database. Other subsystems will also include the attendance administration subsystem and online RFID attendance subsystem that will allow users to login, search for attendance, view attendance, print attendance reports as illustrated in figure below.

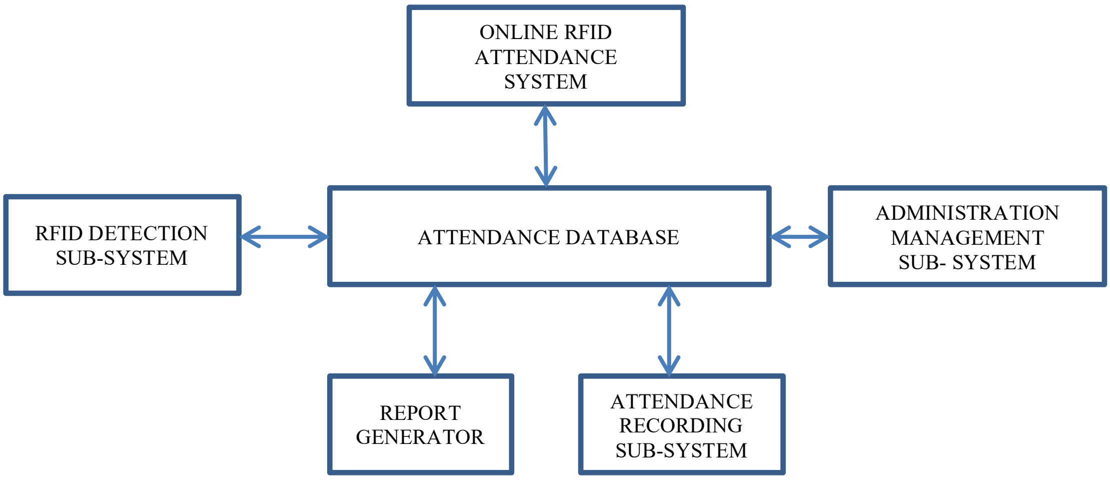

ONLINE RFID ATTENDANCE SYSTEM
RFID DETECTION SUB-SYSTEM
ATTENDANCE DATABASE
ADMINISTRATION MANAGEMENT SUB- SYSTEM
REPORT GENERATOR
ATTENDANCE RECORDING SUB-SYSTEM

Figure: RFID Repository Model

Note: The students are expected to adopt one of the four architectural styles.