THE STATE UNIVERSITY OF ZANZIBAR

SCHOOL OF COMPUTING COMMUNICATION AND MEDIA STUDIES

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

FINAL YEAR PROJECT REPORT GUIDELINE

Software Based Project

2023/2024

1. Contents of Project Progressive report

The following sections are the suggested parts of a project progressive 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.

1. Report preliminaries

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".

Declaration (plagiarism).

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)

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.

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 Figures and 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 and page 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 keep these 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 the List 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.

Table of abbreviations (optional).

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 accessed if it is an online document. For example, following IEEE style, “This section can be found verbatim at Carleton University [1]. For example, here is the reference, formatted correctly:

1. N. Artevema, " Sample IEEE Documentation Style for References,” [Online document], 2000, [2012 May 4], Available at http://http- server. carleton.ca/ ~nartemev/ IEEE_style.html

References to sources should be numbered sequentially by order of mention in the text, with the number placed in brackets and printed on line (not as a super- or subscript) like [1]. The list of all references used in the text should appear in numerical order of mention at the end of the document.

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”.

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.

Conclusion

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

Main content

1. CHEPTER 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

1.1. Introduction

Provide a short and clear description and intension of the project you are working on. For example, if you are working on a title called "Online Admission for SUZA", then you write about Online Admission in general. Also write about the organization for which you develop the system. Based on the previous title, this is about SUZA.

1.2. Project background and motivation

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/worth doing? Specify the significance of your project. What are effects of the project on the company or other company's in general. What is its contribution to the project area?

1.3. Problem statement –

Provide brief investigation of the system under consideration and gives a clear picture of what actually the physical system is? 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

Brief analysis or summary of the problems identified relating to the project or issue to be addressed by the project in not more than three paragraphs. Be sure to go and point out the problem explicitly. Also discuss about the effects of the problems on the company's operations

1.4. Problems Solution and the scope

You should then highlight and summaries that how your project will address the problem. Explain the scenarios of how your proposed system will be used and explain how this solution is better than the current practice. Describe the envisioned benefits of your proposed solution.
Write the scope of your project. To what extent your system solves the problem of the organization.
Generally, the services, the deliverability you have to accomplish in the given time

1.5. Objectives –

Concise statement of the aims and objectives of the project. Define exactly what is going to be done in the project. What you want to achieve at the end of the project. This is written as general /main objective. An example of statements in this part could be:

"The main objective of this project is"

There may also be specific objectives to accomplish the aim (main objective) of the project specific objective that help to achieve the general objective.
Identify the specific objectives:

a. The objective(s) should be precise, clear, Measurable, Relevant, achievable, and well defined.
b. The list of objectives should be presented in point form.
c. Consult supervisor for advice.

NB: [Four specific objectives are more than enough!]

1.6. Feasibility study report – operational, economic, legal and technical feasibility

Senior business analysts perform a feasibility study to determine the software's viability. The usual approach is to focus primarily on these five factors:

Budget constraints.
Legal implications.
Operational requirements.
Available in-house skills.
The required project timeframe.

Write about the feasibility of your project in terms of economic benefit, technical knowledge required to implement the system, and the time available for the project.

1.7. Organization 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 source of data and the procedure for collecting the data, analysis of the data and the system and fixing the requirements specification.
- Chapter Four is System design and so on.

2. 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.
To understand what your project is about, you should do a small survey of any related ideas and projects. which have been done so far which are related to your project. In essence, the ones that you have studied and hence lead to the identification of the problem/knowledge gap that you are trying to address. You should comment on existing works related to your problem.

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

What past works by others have been done on the problem? 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.).

3. CHAPTER 3: Project Methodology

3.1. Introduction

In this chapter the methodology selected and how you proceed with your project with the chosen 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 requirements gathered and analysis results 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 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.

This section should also describe and justify all the methodology choices you made. For example, how you collect your data, how you analyze your data and who or where you collected data from (sampling). The methodology includes an explanation of 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, this section of your methodology should detail what you do and how you do it. Therefore, include a clear description of the methodology you used. Then proceed with your project work based on the methodology chosen.

3.2. Software development approach (object oriented or structured)

A systemic approach is required for a coherent and well-running system. Bottom-Up or Top-Down approach is required to take into account all related variables of the system. Top-Down approach is a Structured Approach and Bottom-Up is an Object-Oriented Approach. 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.

Structured Approach	Object-Oriented Approach
Structured Design is more suitable for offshoring.	It is suitable for in-house development.
It shows clear transition from design to implementation.	Not so clear transition from design to implementation.
It is suitable for real time system, embedded system and projects where objects are not the most useful level of abstraction.	It is suitable for most business applications, game development projects, which are expected to customize or extended.

3.3. Software development life cycle model (SDLC)

Describe what software development process style you have used in pursuing different phases of the software development life cycle. (E.g. Agile, Waterfall, Incremental, etc.). and justify why you decide to use that SDLC model.

3.4. Software development tools

Describe any software tools you will use to accomplish different activates. Software tools for requirement analyzing, designing, implementing and testing the system.

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.

3.5. Information Gathering and Analysis

In this section write few lines about the methods you used to understand the problem, i.e., how you gathered information about the problem - interviewing concerned persons, reading existing documents (literature review), using questionnaires, 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 analyze and develop requirements specifications from the concept of operations and validate software or system requirements.

3.6. 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. The tools and techniques used for system analysis are

- Object-Oriented diagrams
- ER diagrams
- Data Flow Diagrams
- 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 Analysis and the Design of the System as well.

After requirement gathering, you have to structure the requirements and develop software models. So you should also specify how you model the system (OO, or structured approach) and why you chose that approach.

Structured Approach	Object-Oriented 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 and produce models using the various tools.

Refer Appendix IX for Object Oriented Design Approach.

4. Chapter 4: System analysis

4.1. Existing System

4.1.1. Existing System Description

The existing system description describes the current system of the organization as it is. This could be describing the activities they perform, how they handle information, and the drawbacks of the system.

4.1.2. Business Rules

Write the rules used by the organization currently. In the online banking case, this could be the interest rate allowed for saving account, the maximum amount of money that someone can withdraw at a time, interest rate for loan, etc. Generally, they are the rules by which the business is governed

4.2. 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.

Note: Students are therefore expected to elicit both user and system requirements.

System Requirements Specification may have the following sections and subsections.

4.2.1. 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.

4.2.2. 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.

4.2.3. 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.

4.2.4. 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.

4.2.5. 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.

4.3. System Modeling

4.4. Dataflow diagram

DFD describes the actual process that exists. The DFD that models the new system that you are going to develop may be different than this, and that shall come in the Design phase in Chapter 5.

The DFD should be followed by a data dictionary, that unambiguously describes the format of each and every piece of information both in transit as well as in repository.

1. Data Flow Diagram for existing system

The DFD for the analysis of the existing system is derived as follows:

1. The information inputs should correspond to the various forms in use, the recording of directly observed facts, the direct inputs provided by various persons and recorded by some designated official, etc.

2. The data repositories should correspond to the actual registers, or other data recording methods currently used in the process.

3. The outputs should correspond to the formal reports that are prepared in the existing system as well as the adhoc information retrievals by different persons at different stages.

4.5. Requirement Structuring

For OBJECT ORIENTED APPROACH

4.5.1. Use case diagram

Show the functionality of your system using use case diagram and how the actors interact with the system. Also show use case reusability by including <<include>>, <<extend>>, and <<inherit>> relationships between use cases.

4.5.2. Use Case Documentation

This is step by step description of the actions performed by each use case. It should contain preconditions, post conditions, main course of action, and alternate course of action.

4.5.3. Sequence diagram

Sequence diagrams should be drawn for each use case to show how different objects interact with each other to achieve the functionality of the use case.

4.5.4. Conceptual modelling: Class diagram

Create a class diagram that will be the building block the system you will develop. Class diagrams should show the objects the system is comprised of and how they are interrelated.

4.5.5. Entity relationship diagram

For STRUCTURED APPROACH

4.5.6. Data Flow Diagram(DFD) to illustrate how current system is implemented

4.5.7. Data Flow Diagram (DFD to illustrate how proposed system is implemented

4.5.8. Entity relationship diagram

5. CHAPTER 5: System Design

5.1. Architectural design

- Current software architecture
- Try to illustrate the current software architecture of your system if any.

5.2. Proposed software architecture

- Subsystem decomposition
  Subsystem diagram shows the service it provides or it accepts from other subsystems, and the coupling and the coherence between them.

5.3. Database Design

- Relational Model
- Data Description
- Data Dictionaries

5.4. **User** Interface Design

- Forms and Reports
- Interface design sample

5.5. Access control and security

- In the systems, different actors have access to different functionality and data. Define the access controls for your system.

6. CHAPTER 6: System implementation and testing

6.1. Technologies

6.2. Database implementation

6.2.1. Internal Schema of database (database schema)

6.2.2. Data Dictionary

6.3. Testing

- ✓ Give a description how you have conduct system test. The description should answer the following questions:
- ✓ Does each functionality of the system produce an expected output?
- ✓ Does the system respond with invalid data entry?
- ✓ Does the system respond with incorrect login credentials?
- ✓ Does your system restrict with user to perform non- privileged functionalities?
- ✓ Did you implement the system using the proposed techniques and development approaches?

6.4. User Interfaces – Provide a pictorial representation of most important user interfaces of the main functionalities of your system

6.5. Strength ad Limitation of the system

6.5.1. What is covered from requirements

6.5.2. What is not covered

7. CHAPTER 6: Conclusion, Recommendations Challenges and References
8. References