UNIT 1 PROJECT STRUCTURE

Structure **Page Nos.**

1.0 INTRODUCTION

The project report should be documented with an engineering approach to the solution of the problem that you have sought to address. The project report should be prepared in order to solve the problem in a methodical and professional manner, making due references to appropriate techniques, technologies and professional standards. You should start the documentation process from the first step of software development so that you can easily identify the issues to be focused upon in the ultimate project report. You should also include the details from your project notebook, in which you would have recorded the progress of your project throughout the course. The project report should contain enough details to enable examiners to evaluate your work. The details, however, should not render your project report as boring and tedious. The important points should be highlighted in the body of the report, with details often relegated to appendices. This unit covers all the details on the structure of mini project report contents; it also contains detailed explanations on each of these contents.

1.1 OBJECTIVES

After going through this unit, you should be able to:

- demonstrate a systematic understanding of project contents;
- understand methodologies and professional way of documentation;
- know the meaning of different project contents, and
- understand established techniques of project report development.

1.2 IMPORTANCE OF THE MINI PROJECT

The Mini Project is not only a part of the coursework, but also a mechanism to demonstrate your abilities and specialisation. It provides the opportunity for you to demonstrate originality, teamwork, inspiration, planning and organisation in a software project, and to put into practice some of the techniques you have been taught throughout the previous courses. The Mini Project is important for a number of reasons. It provides students with:

- opportunity to specialise in specific areas of computer science;
- future employers will most likely ask you about your project at interview;
- opportunity to demonstrate a wide range of skills and knowledge learned, and
- encourages integration of knowledge gained in the previous course units.

The project report is an extremely important aspect of the project. It serves to show what you have achieved and should demonstrate that:

- You understand the wider context of computing by relating your choice of the project, and the approach you take, to existing products or research.
- You can apply the theoretical and practical techniques taught in the course to the problem you are addressing and that you understand their relevance to the wider world of computing.
- You are capable of objectively criticising your own work and making constructive suggestions for improvements or further work based on your experiences so far.
- You can explain your thinking and working processes clearly and concisely to others through your project report.

1.3 MINI PROJECT: TABLE OF CONTENTS

The project report should contain a full and coherent account of your work. Although there will be an opportunity to present your work verbally, and demonstrate the software, the major part of the assessment will be based on the written material in your project report. You can expect help and feedback from your MCS-044 course counsellor, but ultimately it's your own responsibility. The suggestive structure of a project report should be as given below; however, you should be guided by your counsellor in selecting the most appropriate format for your project.

Title Page

Original Copy of the Approved Proforma of the Project Proposal
Certificate of Authenticated work
Role and Responsibility Form

Abstract

Acknowledgement
Table of Contents
Table of Figures

CHAPTER 1: INTRODUCTION

1.1 Background
1.2 Objectives
1.3 Purpose, Scope, and Applicability
  1.3.1 Purpose
  1.3.2 Scope
  1.3.3 Applicability
1.4 Achievements
1.5 Organisation of Report

CHAPTER 2: SURVEY OF TECHNOLOGIES

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem Definition
3.2 Requirements Specification
3.3 Planning and Scheduling
3.4 Software and Hardware Requirements
3.5 Preliminary Product Description
3.6 Conceptual Models

CHAPTER 4: SYSTEM DESIGN

4.1 Basic Modules

1.4 EXPLANATION OF CONTENTS

Title Page

Sample format of *Title page* is given in Appendix 1 of this block. Students should follow the given format.

Original Copy of the Approved Proforma of the Project Proposal

Sample *Proforma of Project Proposal* is given in Appendix 2 of this block. Students should follow the given format.

Certificate of Authenticated work

Sample format of *Certificate of Authenticated work* is given in Appendix 3 of this block. Students should follow the given format.

Role and Responsibility Form

Sample format for *Role and Responsibility Form* is given in Appendix 4 of this block. Students should follow the given format.

Abstract

This should be one/two short paragraphs (100-150 words total), summarising the project work. It is important that this is not just a re-statement of the original project outline. A suggested flow is background, project aims and main achievements. From the abstract, a reader should be able to ascertain if the project is of interest to them and, it should present results of which they may wish to know more details.

Acknowledgements

This should express your gratitude to those who have helped you in the preparation of your project.

Table of Contents: The table of contents gives the readers a view of the detailed structure of the report. You would need to provide section and subsection headings with associated pages. The formatting details of these sections and subsections you will find in unit 2 of this block.

Table of Figures: List of all Figures, Tables, Graphs, Charts etc. along with their page numbers in a table of figures.

Chapter 1: Introduction

The introduction has several parts as given below:

Background: A description of the background and context of the project and its relation to work already done in the area. Summarise existing work in the area concerned with your project work.

Objectives: Concise statement of the aims and objectives of the project. Define exactly what you are going to do in the project; the objectives should be about 30 /40 words.

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

- *Purpose:* Description of the topic of your project that answers questions on why you are doing this project. How your project could improve the system its significance and theoretical framework.

- *Scope:* A brief overview of the methodology, assumptions and limitations. You should answer the question: What are the main issues you are covering in your project? What are the main functions of your project?

- *Applicability:* You should explain the direct and indirect applications of your work. Briefly discuss how this project will serve the computer world and people.

Achievements: Explain what knowledge you achieved after the completion of your work. What contributions has your project made to the chosen area? Goals achieved - describes the degree to which the findings support the original objectives laid out by the project. The goals may be partially or fully achieved, or exceeded.

Organisation of Report: Summarising the remaining chapters of the project report, in effect, giving the reader an overview of what is to come in the project report.

Chapter 2: Survey of Technologies

In this chapter *Survey of Technologies* you should demonstrate your awareness and understanding of Available Technologies related to the topic of your project. You should give the detail of all the related technologies that are necessary to complete your project. You should describe the technologies available in your chosen area and present a comparative study of all those Available Technologies. Explain why you selected the one technology for the completion of the objectives of your project.

Chapter 3: Requirements and Analysis

Problem Definition: Define the problem on which you are working in the project. Provide details of the overall problem and then divide the problem in to sub-problems. Define each sub-problem clearly.

Requirements Specification: In this phase you should define the requirements of the system, independent of how these requirements will be accomplished. The Requirements Specification describes the things in the system and the actions that can be done on these things. Identify the operation and problems of the existing system.

Planning and Scheduling: Planning and scheduling is a complicated part of software development. Planning, for our purposes, can be thought of as determining all the small tasks that must be carried out in order to accomplish the goal. Planning also takes into account, rules, known as constraints, which, control when certain tasks can or cannot happen. Scheduling can be thought of as determining whether adequate resources are available to carry out the plan. You should show the Gantt chart and Program Evaluation Review Technique (PERT).

Software and Hardware Requirements: Define the details of all the software and hardware needed for the development and implementation of your project.

- *Hardware Requirement:* In this section, the equipment, graphics card, numeric co-processor, mouse, disk capacity, RAM capacity etc. necessary to run the software must be noted.

- *Software Requirements:* In this section, the operating system, the compiler, testing tools, linker, and the libraries etc. necessary to compile, link and install the software must be listed.

Preliminary Product Description: Identify the requirements and objectives of the new system. Define the functions and operation of the application/system you are developing as your project.

Conceptual Models: You should understand the problem domain and produce a model of the system, which describes operations that can be performed on the system, and the allowable sequences of those operations. Conceptual Models could consist of complete Data Flow Diagrams, ER diagrams, Object-oriented diagrams, System Flowcharts etc.

Chapter 4: System Design

Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudocode and other documentation.

Basic Modules: You should follow the divide and conquer theory, so divide the overall problem into more manageable parts and develop each part or module separately. When all modules are ready, you should integrate all the modules into one system. In this phase, you should briefly describe all the modules and the functionality of these modules.

Data Design: Data design will consist of how you organise, managing and manipulate the data.

- *Schema Design:* Define the structure and explanation of schemas used in your project.

- *Data Integrity and Constraints:* Define and explain all the validity checks and constraints you are providing to maintain data integrity.

Procedural Design: Procedural design is a systematic way for developing algorithms or procedurals.

- *Logic Diagrams:* Define the systematical flow of procedure that improves its comprehension and helps the programmer during implementation. e.g., Control Flow Chart, Process Diagrams etc.

- *Data Structures:* Create and define the data structure used in your procedures.

- *Algorithms Design:* With proper explanations of input data, output data, logic of processes, design and explain the working of algorithms.

User Interface Design: Define user, task, environment analysis and how you intend to map those requirements in order to develop a "User Interface". Describe the external and internal components and the architecture of your user interface. Show some rough pictorial views of the user interface and its components.

Security Issues: Discuss Real-time considerations and Security issues related to your project and explain how you intend avoiding those security problems. What are your security policy plans and architecture?

Test Cases Design: Define test cases, which will provide easy detection of errors and mistakes with in a minimum period of time and with the least effort. Explain the different conditions in which you wish to ensure the correct working of your software.

Chapter 5: Implementation and Testing

Implementation Approaches: Define the plan of implementation, and the standards you have used in the implementation.

Coding Details and Code Efficiency: Students not need include full source code, instead, include only the important codes (algorithms, applets code, forms code etc). The program code should contain comments needed for explaining the work a piece of code does. Comments may be needed to explain why it does it, or, why it does a particular way.

You can explain the function of the code with a shot of the output screen of that program code.

- *Code Efficiency:* You should explain how your code is efficient and how you have handled code optimisation.

Testing Approach: Testing should be according to the scheme presented in the system design chapter and should follow some suitable model – e.g., category partition, state machine-based. Both functional testing and user-acceptance testing are appropriate. Explain your approach of testing.

- *Unit Testing:* Unit testing deals with testing a unit or module as a whole. This would test the interaction of many functions but, do confine the test within one module.

- *Integrated Testing:* Brings all the modules together into a special testing environment, then checks for errors, bugs and interoperability. It deals with tests for the entire application. Application limits and features are tested here.

Modifications and Improvements: Once you finish the testing you are bound to be faced with bugs, errors and you will need to modify your source code to improve the system. Define what modification you implemented in the system and how it improved your system.

Chapter 6: Results and Discussion

Test Reports: Explain the test results and reports based on your test cases, which should show that your software is capable of facing any problematic situation and that it works fine in different conditions. Take the different sample inputs and show the outputs.

User Documentation: Define the working of the software; explain its different functions, components with screen shots. The user document should provide all the details of your product in such a way that any user reading the manual, is able to understand the working and functionality of the document.

Chapter 7: Conclusions

Conclusion: The conclusions can be summarised in a fairly short chapter (2 or 3 pages). This chapter brings together many of the points that you would have made in the other chapters.

Limitations of the System: Explain the limitations you encounterd during the testing of your software that you were not able to modify. List the criticisms you accepted during the demonstrations of your software.

Future Scope of the Project describes two things: firstly, new areas of investigation prompted by developments in this project, and secondly, parts of the current work that were not completed due to time constraints and/or problems encountered.

REFERENCES

It is very important that you acknowledge the work of others that you have used or adapted in your own work, or that provides the essential background or context to your project. The use of references is the standard way to do this. Please follow the given standard for the references for books, journals, and online material.

GLOSSARY

If you use any acronyms, abbreviations, symbols, or uncommon terms in the project report then their meaning should be explained where they first occur. If you go on to use any of them extensively then it is helpful to list them in this section and define the meaning.

APPENDICES

These may be provided to include further details of results, mathematical derivations, certain illustrative parts of the program code (e.g., class interfaces), user documentation etc.

In particular, if there are technical details of the work done that might be useful to others who wish to build on this work, but that are not sufficiently important to the project as a whole to justify being discussed in the main body of the project, then they should be included as appendices.

1.5 SUMMARY

Project development usually involves an engineering approach to the design and development of a software system that fulfils a practical need. Projects also often form an important focus for discussion at interviews with future employers as they provide a detailed example of what you are capable of achieving. In this course you can choose your project topic from the lists supplied in *Unit 4: Category-wise Problem Definition.* The next Unit *Guidelines and Suggestions* will provide you detailed guidelines and suggestions, which will be useful for you during project development and the preparation of the report.

1.6 FURTHER READINGS

1. *Modern Systems Analysis and Design*; Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich; Pearson Education; Third Edition; 2002.

2. ISO/IEC 12207: *Software Life Cycle Process (http://www.software.org/quagmire/descriptions/iso-iec12207.asp).*

3. IEEE 1063: *Software User Documentation (http://ieeexplore.ieee.org).*

4. ISO/IEC: 18019: *Guidelines for the Design and Preparation of User Documentation for Application Software.*

5. http://www.sce.carleton.ca/squall.

6. http://en.tldp.org/HOWTO/Software-Release-Practice-HOWTO/documentation.html.

7. http://www.sei.cmu.edu/cmm/