

SYSTEM DEVELOPMENT OUTLINE

CHAPTER ONE

1. INTRODUCTION

Reasons for studying the problem selected should be listed. Project works already done in that area should be mentioned. In here, you need to discuss about the significance of your focus area. Besides, the problem area and motivation to the need for your project work is described. Moreover, if your focus area leans on a specific organization you need to say so about the organization.

1.1. Background of the Organization

Here you are expected to describe mission, vision and objective of the organization for which you are going to develop the new system, organization structure etc...

1.2. Statement of the Problem

Here you are expected to describe specifically what the problem is and the problem that you intend to solve. Describe how the existing system functions (manual, semi –automated, automated) and then list and describe the problems in the existing system.

1.3. Objectives of the Project

General and specific objectives of the project to be achieved by the proposed system should be discussed. Your objective should be SMART (Specific, Measurable, Achievable, Relevant and Time bounded).

1.3.1. General Objective

Your general objective should incorporate a sentence that begins with “The general objective of this project is ...”, “The main objective of this project is ...” and include the applicable area of your project work. It should be general.

1.3.2. Specific Objectives

Here you are expected to list a number of activities which has to be done to achieve the general objective. It should be specific and systematically address the various aspect of the problem as defined under “*Statement of the Problem*” and the key factor that are assumed to influence or cause of the problem.

1.4. Feasibility Analysis

Here you are expected to assess the strengths and weaknesses of a proposed project and present directions of activities which will improve a project and achieve objectives according to section 1.4.1 – 1.4.4.

1.4.1. Technical Feasibility

Here you are expected to analysis the project in terms whether the required technology is available or not, whether the required resources are available, Manpower- programmers, testers & debuggers, and Software and hardware.

1.4.2. Operational Feasibility

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. And also, does management support the project? Are the users not happy with current business practices? Will it reduce the time (operation) considerably?

1.4.3. Economical Feasibility

For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated. Economic analysis is used for evaluating the effectiveness of the proposed system. In economic feasibility, the most important is cost-benefit analysis.

1.4.4. Other Feasibility (if any)

In this section students are expects to discuss other feasibility issues according to the project they are going to develop like political feasibility, cultural feasibility and others.

1.5. Scope and Limitation of the Project

Here you need to define specific boundaries of your project in terms of what the project does and what the project doesn't.

1.5.1. Scope of the Project

In this section you need to describe specific boundaries of your project in terms of what the project does.

1.5.2. Limitation of the Project

In this section you need to define specific boundaries of your project in terms of what the project doesn't.

1.6. Significance of the Project

The societal and technological importance of your project should be discussed in this part.

1.7. Beneficiary of the Project

Who will benefit from the system?

1.8. Methodology of the Project

State how the requirements are gathered, the tools used for analysis and design, model, implement, test etc.

1.8.1. Data Collection Tools/Techniques

In this section you may or may not include observation, questionnaire, interview and document analysis etc.

1.8.2. System Analysis and Design

In this section you should include object-oriented system analysis and design(OOSAD).

1.8.3. System Development Model

Here you are expected to mention software development life cycle models like iterative, spiral, V-model or waterfall etc. and describe why do you select the development model.

1.8.4. System Testing Methodology

In this section you should may or may not include unit testing, system testing, acceptance testing, integration testing etc.

1.8.5. Development Tools and Technologies

In this section and subsections, you should mention programing languages and its editor, database technologies, documentation tools, unified modeling language(UML) design tools, hardware tools for deployment purpose etc.

1.8.5.1. Frontend Technologies

1.8.5.2. Backend Technologies

1.8.5.3. Documentation and Modeling Tools

1.8.5.4. Deployment Environment

1.9. Budget and Time Schedule of the Project

*This section and its subsections **included only** in your proposal.*

1.9.1. Budget of the Project

Here you are expected to describe all costs (Paper cost, pen, mobile card, transport, laptop or desktop cost, etc.) by using table.

1.9.2. Time Schedule of the Project

Here you are expected to describe the time schedule of your project by using the Gant chart, pert chart or any other tool.

1.10. Team Composition

Here mention the responsibility and tasks of each group members by using table. *This section **included only** in your proposal.*

1.11. Document Organization

In this section include the chapters and what it describes about.

CHAPTER TWO

2. DESCRIPTION OF THE EXISTING SYSTEM

2.3. Introduction of Existing System

Describe the existing system how it works in detail step by step.

2.4. Users of Existing System

Here you are expected to describe the users and their responsibilities in the existing system ...

2.5. Major Functions of the Existing System

Describe the existing system major functions.

2.6. Forms and Other Documents of the Existing Systems (if any)

Here the includes sample scan image of reports, forms and other related documents that used in the existing system and the remaining forms and reports should be included in **Appendix section**.

2.7. Drawbacks of the Existing System

In this section describe the existing system disadvantages and many more difficulties to work well.

2.8. Business Rules of the Existing System

List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.

CHAPTER THREE

3. PROPOSED SYSTEM

In this chapter you are expected to discuss the overall description of your proposed system, functional requirements, and non-functional requirements.

3.3. Functional Requirements

Here you are expected to mention what your system should do. Functional requirements specify a function that a system or system component must be able to perform. Mention product features or functions that developers must implement to enable users to accomplish their tasks.

3.4. Non-functional Requirements

In this section describe nonfunctional requirements place constraints on how the system will do so. Depend on your project the nonfunctional requirement may be vary. So, list all the nonfunctional requirements which are relevant to your system like:

3.4.1. User Interface and Human Factors

What kind of interface should the system provide? What is the level of expertise of the users?

3.4.2. Hardware Consideration

Are there hardware compatibility requirements? Will the system interact with other hardware system?

3.4.3. Security Issues

Should the system be protected against internal and external intrusions or against an authorized user? To what level? What security algorithm applied for your system?

3.4.4. Performance Consideration

How responsive should the system be? How many concurrent users should it support? What is a typical or extreme load?

3.4.5. Error Handling and Validation

How should the system handle exceptions? Which exceptions should the system handle? What is the worse environment in which the system is expected to perform?

3.4.6. Quality Issues

How reliable/available/robust should the system be? What is the client's involvement in assessing the quality of the system or the development process?

3.4.7. Backup and Recovery

How, when and where to take backup? How to recover the system when your system fail?

3.4.8. Physical Environment

Where will the system be deployed? Are there external factors such as weather conditions that the system should withstand?

3.4.9. Resource Issues

What are the constraints on the resources consumed by the system?

3.4.10. Documentation

What level of document is required? Should only user documentation be provided? Should there be technical documentation for maintainers? Should the development process be documented?

CHAPTER FOUR

4. SYSTEM ANALYSIS

In this chapter, you should have discussed about use case model, object model and dynamic models.

4.3. System Model

Here you are expected to mention an overview of system model.

4.3.1. Use Case Model

In this section, you are expected to model comprised use case diagram, use case definitions, and actor definitions to document the functional requirements of a system. Also, you should have to identify each actor and use cases of the system based on the functional requirement.

4.3.1.1. Use Case Diagram

Here draws a diagram that shows system boundary, use cases, actors and their relationships by using <<include>> or <<extend>>. Also, you should have to illustrate the interaction of each actor with each use case using modelling tools like E-Draw max, Visio and others.

Actor: Someone interacts with use case (system function), and named by *noun*.

Use Case: System functions, and *named by verb + Noun (or Noun Phrase)*.

4.3.1.2. Use Case Description

Here you are expected to write the description of each use case in tabular form by using narrative style or action response style.

4.3.1.3. Use case Scenario

Scenarios are an instance (example) of a use case explaining a concrete major set of action. Scenario or use case realizations are just a sequential narrative description of events or an instance of a use case.

4.4. Object Model

In this section, discussed an overview of object model.

4.4.1. Class Diagram

In this section you are expected to illustrate (diagrammatically) the conceptual relationship among objects/classes without including visibility of data and members, association, generalization, cardinality etc.

4.4.2. Data Dictionary

In this section mention attributes, data type, data size, key constraints and constraints of the identified entities or classes by using tabular form.

4.5. Dynamic Model

The dynamic model represents the time–dependent aspects of a system. It is concerned with the temporal changes in the states of the objects in a system. In this section you are expected to document the behavior of the object model, in terms of sequence, activity and state chart diagrams.

4.5.1. Sequence Diagram

In this section, you should have to illustrate (diagrammatically) a sequential logic, in effect, and the time ordering of messages. From business process perspective. How the business model is executed?

4.5.2. Activity Diagram

In this section you are expected to illustrate graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

4.5.3. State Chart Diagram

Here you are expected to define different states of an object during its lifetime and these states are changed by events.

CHAPTER FIVE

5. SYSTEM DESIGN

In this chapter provide a brief overview of the design goals, current and proposed software architecture, Hardware/software mapping, Persistent data management and Access control and security.

5.3. Design Goals

The Design Goals specify the qualities of the system that should be achieved and addressed during the design of the system like the list down with pair manner as you listed non-functionality of your project.

- **User Interface and Human Factors**
- **Hardware Consideration**
- **Security Issues**
- **Performance Consideration**
- **Error Handling and Validation**
- **Quality Issues**
- **Backup and Recovery**
- **Physical Environment**
- **Resource Issues**
- **Documentation**

5.2. Current System Architecture (if any)

Current system architecture describes the architecture of the system being replaced.

5.3. Proposed System Architecture

This section presents a general view of your system architecture and briefly describes the assignment of functionality to each subsystem.

5.3.1. Subsystem Decomposition and Description

Subsystem decomposition draws and describes the decomposition into subsystems and the responsibilities of each. This is the main product of system design. Here, use **UML component diagram** to diagrammatically illustrate your components.

5.3.2. Hardware/Software Mapping

Hardware/software mapping describes how subsystems are assigned to hardware and customized components (if any). Here, use **UML deployment diagram** to diagrammatically illustrate the hardware/software mapping.

5.3.4. Detailed Class Diagram

In this section show, classes, attributes, methods, attribute data types, visibility ((Public (+), Private (-), Protected (#)) of attributes and methods), inheritance, association, aggregation, composition, dependency, and municipality (cardinality and optimality). Here, uses the UML class diagram to specify attributes and operations with their visibility information.

5.3.5. Persistent Data Management

Persistent data management describes the persistent data stored by the system and the data management infrastructure required for it. This section typically includes the description of data schemes, the selection of database, and the description of the encapsulation of the database. Here, use relational database or Object diagram if you are using object database. In this section, map class diagram to table.

5.3.6. Access Control and Security

Access control and security describes the user model of the system in terms of an access privilege. Utilize global access table, describing the access relation between the actors, objects and operations in the system. You can use tables to show the privilege assigned to each users of the system. This section also describes security issues, such as the selection of an authentication mechanism, the use of encryption, and the management of keys.

5.4. Packages

This section describes the decomposition of subsystems into packages and the file organization of the code. This includes an overview of each package, its dependencies with other packages, and its expected usage. Here, use UML package diagram to diagrammatically depict your packages.

5.5. Algorithm Design

For the elements found in architectural design define the algorithm required for each element to accomplish its tasks.

In this section, students required to include **pseudo code** format.

5.6. User Interface Design

Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed.

CHAPTER SIX

6. IMPLEMENTATION AND TESTING

In this chapter, discuss about implementation of database, detailed class diagram, application server, application security,

6.3. Implementation of the Database

You should choose appropriate database management system, with justified reason and should perform the following activities:

- All tables which were identified and shown as persistent model in the design document should be created with their primary keys, foreign keys, check constraints and unique constraints.
- All tables should satisfy at least third normal form
- Whenever you found appropriate, implement the following: Indexes, Views, Triggers, Stored procedures
- Configure a schedule of database backup.
- Configure database level security

6.4. Implementation of the Class Diagram

Implement all the classes on the design of your class diagram by your target programming language. Define attributes with the appropriate data type and access visibilities (private, protected, public) and implement according to the business logic. Define all methods with appropriate return type, parameters and the corresponding data types and access visibility and implement with specified logic. Document all your object code with standard comments and implement all the reports as shown on your design document.

6.5. Configuration of the Application Server

Here uses an appropriate application server for their development and clearly justified the reasons about their application server selection. Also, perform the following activities:

- Properly start and shutdown the application server.
- Organize folders and files on the server properly.

- Configure the server to work on the same machine: Able to access contents from the same machine.
- Configure the server to work on from remote machine: Able to access contents from another/remote machine.
- Separation of application server and database: E.g. Not using web development environments like WampServer.
- Configure the server to work on different port number.

6.6. Configuration of Application Security

From the security aspect of any system, students should take in to account the following tasks.

- Implement all input validations properly
- Implement encryption/deception
- Roles must be defined clearly
- User accounts must be assigned with necessary access privileges
- Sessions should be implemented.
- All none functional requirements as described in the system feature part must be checked.

6.7. Implementation of User Interface

Regarding to the user interface, students are expected to apply the following.

- The user interface must be user centered design (Place users in control of the interface)
- Reduce users' memory load
- Make the user interface consistent.

6.8. Testing

In this section and subsections, you are expected to describe which kind of testing technique you are going to apply and also should mention the test results.

- 6.8.1. Test Case**
- 6.8.2. Testing Tools and Environment**
- 6.8.3. Unit Testing**
- 6.8.4. System Testing**
- 6.8.5. Integration Testing**
- 6.8.6. Acceptance Testing**

CHAPTER SEVEN

7. CONCLUSION AND RECOMMENDATION

Matching of the conclusions with the objectives framed and fulfillment of the objectives are taking into consideration in this part. Further scope and further enhancement of the work done also indicated here.

7.3. Conclusion

The conclusion is intended to help the reader understand why your project work should matter to them after they have finished reading the paper. A conclusion is not merely a summary of the main topics covered or a re-statement of your project problem, but a synthesis of key points. For most projects, one well-developed paragraph is sufficient for a conclusion, although in some cases, a two or three paragraph conclusion may be required.

When you write a conclusion for project, always remember that you have to make a summary of the content as well as the purpose that you have in mind without looking way too wooden or dry. Most conclusions of project tend to have a couple key elements. Of course there are some tactics that you should be using if you want to write a conclusion that is effective. Plus, there are some things that you should avoid as well.

When writing the conclusion to your paper, follow these general rules:

- State your conclusions in clear, and simple language. Re-state the purpose of your study then state how your project resolve issues in the existing system and how your system is different from other related projects.
- Do not simply reiterate your results or the activities done in the entire software development life cycle.
- Indicate opportunities for future project works if you haven't already done so in the design and implementation section of your project work. Highlighting the need for further project works provides the reader with evidence that you have an in-depth awareness of the existing problem.

7.4. Recommendation

Sometimes recommendations are included with a report's conclusion, although they serve different purposes. Whereas a conclusion offers you the opportunity to summarize or review your reports main ideas, recommendations suggest actions to be taken in response to the findings of a report. You can regard recommendations as a prompt to action for your readers. As you have seen from your planning, your report structure should lead up to the recommendations and provide justification for them.

What makes a good recommendation? Effective recommendations:

- Describe a suggested course of action to be taken to solve a particular problem that can't addressed yet in the project;
- Are written as action statements without justification;
- Are stated in clear, and specific language;
- Should be expressed in order of importance (i.e. it is better to start from the most important recommendation and proceeded to others based on their significance to the clients or users of the system);
- Are based on the case built up in the body of the report; are written in parallel structure to the developed system.

A word of caution about writing recommendations: you should always consider your relationship with the reader first. If you have no authority to make recommendations, the reader may be hostile to their presence.

8. REFERENCES

List all papers, books, book sections, web sites, monographs, URLs of Internet archives or of permanent information sources, strictly as per the specific format. The references should be ordered in a convenient way. You should use consistent referencing style such as (IEEE, APA etc.).

9. APPENDICES

Appendix I: Interview and Questionnaires (if any)

Appendix II: Existing System Forms and Reports

Appendix III: Sample Source Code