

**Woldia University**

**Institute of Technology**

**School of Computing**

**Department of Computer Science**

<Font Size 22><BOLD><Centered>

**TITLE OF PROJECT**

<Font Size 14>><BOLD><Centered>

***A project submitted in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science (Regular) - Part 1***

<Font Size 14> <1.5-line spacing> <Italic> <BOLD> <Centered>

**Submitted by:**

**Student’s First Name Father Name Student ID**

<Font Size 14> <BOLD><Centered>

**Advised by:**

**Advisor First Name Father Name (MSc)**

<Font Size 14> <BOLD><Centered>

**Jan, 2023**

**Woldia, Ethiopia**

<Font Size 12> <BOLD> <right>

**CERTIFICATE**

The project titled

“\_\_\_\_\_\_\_\_\_\_\_”

*Compiled* *by*

Name of Student ID

As partial fulfilment of the requirements for degree of

**Bachelor of Science**

*in*

**Computer Science**

*from*

**Woldia University**

**Head of the Department** **Examiner**

**<Name of the Head> <Name of the Examiner >**

<Signature> <Signature>

<Designation> <Designation>

Department of Computer Science Department of Computer Science

Date Date

**DECLARATION**

This is to declare that the project entitled “<**Title of The Project Work>”** is an original work done by us, undersigned students in the Department of Computer Science, School of Computing, Institute of Technology, Woldia University. The reports are based on the project work done entirely by us and not copied from any other source.

**Advisor**

**<Name of the Advisor>**

<Signature>

Department of Computer Science

Date

<**Name of the Student > ID Signature**

**<Name of the Student > ID Signature**

<**Name of the Student > ID Signature**

**<Name of the Student > ID Signature**

**System Requirement Specification Preparation Format**

1. **Acknowledgment**
2. **Table of Content**
3. **List of Figures**
4. **List of Tables**
5. **List of Acronyms**
6. **Abstract**
7. **Chapter 1: Introduction**
   1. **Background of the organization/project**

*// Here you are expected to describe background of the origination and introduce your project in a coherent manner. Describe the objective of the organization for which you are going to develop the new system, organization structure, etc.*

*The background also introduces your project. Identify and describes the nature of your projects in a well-defined manner with reference to the existing related work (related projects). To describe your project, first, you should review related works. Related works are the works/projects related to the problem you are attempting to solve. This provides you information directly related to your topic, problem, or solution. Reviewing others' work provides you an important understanding that should result in avoiding past mistakes, taking advantage of previous successes, and most importantly, potentially improving your solution or the technique in general when applied in your context and others. This also helps you to avoid implementing systems that are already developed. In addition to the obvious purpose indicated, the related work section also can serve to justify that the problem exists by example and argument, motivate interest in your work by demonstrating relevance and importance, and identify the important issues, and provide the background to your solution. After reviewing others work, start writing the background of the organization (if), state what the project is, presenting an overview of related projects, make general statements about the topic, oppose the existing projects, highlight the importance of the topic, state the intent of your project, etc.*

*A reader of the background should be able to answer the following questions.*

*What is the project about? To whom the project will be developed? Why is it relevant or important? What are the issues or problems? What is the proposed solution or approach?*

* *Note: Texts in your project writing must be coherent*
  1. **Introduction about the project**

**//** try to introduce your project here.

* 1. **Statement of the problem**

*// Here you are expected to describe specifically what the problem is and the problem that you intend to solve. Statement of the problem briefly addresses the question: What is the problem that the project will address? The goal of a statement of the problem is to transform a generalized into targeted, well-defined problems; one that can be resolved through computer software. Writing a statement of the problem helps you to clearly identify the purpose of the project. The statement of the problem also serves as the basis for the introductory section of your project directing your reader’s attention quickly to the issues that your proposed project will address and providing the reader with a concise statement of the proposed project itself. A statement of the problem need not be long and elaborate: one page is more than enough for a good statement of the problem.*

* 1. **Objective of the project**
     1. **General Objective**

*//This part describes the goal of the project. It is usually written in a single sentence. 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.*

* + 1. **Specific Objective**

*// Here you are expected to list several activities that must be done to achieve the general objective. Avoid using very general and obvious activities as specific objectives. The objectives should be specific and realistic in terms of capacity, resources, and time. It should be specific and systematically address the various aspect of the problem as defined under “Statement of the Problem” and the key factors that are assumed to influence or cause the problem. During the actual work, it is possible that these objectives may be modified.*

* 1. **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 does not.*

* + 1. **Scope of the project**

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

* + 1. **Limitation of the project**

*// This is intended to help the reader understand the context in which the project should be interpreted and applied. This lists the shortcomings of your project, which may be based on several reasons such as the unavailability of required resources, inefficient project design, the method used, or lack of access to advanced instruments, etc.*

* 1. **Methodology of the Project**
     1. **Data gathering methodology**

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

* + 1. **System Analysis and Design methodology**

*// In this section, you should include object-oriented system analysis and design (OOSAD), or mention other design methods that you use.*

* + 1. **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 you select the development model.*

* + 1. **System Testing Methodology**

*// In this section, you may include unit testing, system testing, acceptance testing, integration testing, etc*

* + 1. **Development Environment and Programming Tools**

*// 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.* ***Front-end frameworks*** *or* ***Technologies****,* ***Backend frameworks*** *or* ***Technologies****,*

* 1. **Feasibility study**
     1. **Technical Feasibility**

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

* + 1. **Operational Feasibility**

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

* + 1. **Economic 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. **Other Feasibility (if any)**

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

* 1. **Significance of the Project**

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

* 1. **Team Composition**

// Here mention the responsibility and tasks of each group member by using a table.

1. **Chapter 2: Existing system**
   1. **Introduction**

// Describe the excising system how it works in a more generalized form and highlight what is presented/discussed in the next subsections in a more generalized form.

* 1. **Description of the existing system**

// Here describes the existing system. This includes the description of the manual system or computerized (if any) processes. Description of the users and their responsibilities in the existing system is required.

* 1. **Limitation of the Existing System**

// This describes the limitation of the current system. The current system can be the manual system or computerized system. When the system that you are intended to develop replaces a manual system, here you are required to describe the major limitations of the manual system with respect to the organization. If the new system will replace an existing computerized system, this section describes the functionality and the problems of the current system. You can organize the limitations of the current system interms of performance (response time), input (inaccurate/redundant/flexible) and output (inaccurate), security and controls, efficiency, etc.

Note: do not describe the limitations of the manual system that everybody guesses. For example, searching, deletion, and update problems, etc.

**2.4. Model of the Existing System**

**2.4.1. Actor**

// An actor is a user that interacts with the system. Therefore, you are expected to mention the actors in the existing system and how they interact with the system.

**2.4.2. Essential use-case**

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

**2.5. Business rule**

// List any operating principles about the product (if any), 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.

1. **Chapter 3 Proposed system**
   1. **Overview of the proposed system**

*// In this chapter you are expected to discuss the overall description of your proposed system, functional requirements, and non-functional requirements. Here, introduces your proposed system in a more general form and highlight what the following subsections of this part discuss.*

**3.2. Requirement specification**

**3.2.1. Functional Requirement**

*// Functional requirements express the services, tasks or functions the system is required to perform. They are fundamental actions that must take place in the software in accepting, processing, and generating the outputs. Functional requirements include the following:*

*- Validity checks on the inputs*

*- The exact sequence of operations*

*- Responses to an abnormal situation, including overflow, communication facilities, and Error handling and recovery*

*- Effect of parameters*

*- Relationship of outputs to inputs, including input/output sequences and formulas for input to output conversion*

**3.2.2. Non-functional requirement**

*// Non-functional requirements are constraints on the system or the development process. Some of the Non-functional requirements are presented as follows:*

*-User Interface and Human Factors: What kind of interface should the system provide? What is the level of expertise of the users?*

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

*- Hardware Consideration: Are there hardware compatibility requirements? Will the system interact with other hardware systems?*

*- Performance Characteristics: How responsive should the system be? How many concurrent users should it support? What is a typical or extreme load?*

*- Error Handling and Extreme Conditions: 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? Are there safety requirements on the system?*

*- 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? System Modifications: What is the anticipated scope of future changes? Who will perform the changes?*

*- Physical Environment: Where will the system be deployed? Are there external factors such as weather conditions that the system should withstand?*

*- Security Issues: Should the system be protected against external intrusions or against malicious users? To what level?*

*- Resource Issues: What are the constraints on the resources consumed by the system?*

**3.2.3. System Requirement**

*// System requirements are a statement that identifies the functionality that is needed by a system in order to satisfy the customer's requirements.*

* 1. **User interface specification and description**

*// This part describes the logical characteristics of each interface between the software product and the users. The specification covers all possible actions that an end-user may perform.*

1. **Chapter 4 System Modeling**
   1. **Introduction**

*// In this chapter, you are expected to mention an overview of the system model. The UML uses mostly graphical notations to express the OO analysis and design of software projects. Write an overview of the use case model, object model, and dynamic models.*

* 1. **Functional/Use-case model**

*// In this section, you are expected to model the 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.*

**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 modeling tools like E-Draw max, Visio, and others.*

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

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

**Use Case Description**

*Here you are expected to write the description of each use case in tabular form by using a narrative style or action response style. The table below indicates a comprehensive use case template. [ table label, the table will be on one page]*

|  |  |
| --- | --- |
| Use Case ID: | Enter a unique numeric identifier for the Use Case. e.g., UC-1 |
| Use Case Name: | Enter a short name for the Use Case using an active verb phrase. e.g., Order a Meal |
| Actors: | [An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks.] |
| Description: | [Provide a brief description of the reason for and outcome of this use case.] |
| Trigger/Precondition: | [Identify the event that initiates the use case.] |
| Post conditions: | [Describe the state of the system at the conclusion of the use case execution. |
| Normal Flow: | [Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. |
| Alternative Flows: | [Document legitimate branches from the main flow to handle special conditions (also known as extensions). For each alternative flow reference, the branching step number of the normal flow and the condition must be true for this extension to be executed. |
| Business Rules | Use cases and business rules are intertwined. Some business rules constrain which roles can perform all or parts of a use case. Perhaps only users who have certain privilege levels can perform specific alternative flows. That is, the rule might impose preconditions that the system must test before letting the user proceed. Business rules can influence specific steps in the normal flow by defining valid input values or dictating how computations are to be performed |
| Assumptions: | List any assumptions |

**Use case Scenario**

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

* 1. **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.*

* + 1. **Sequence Diagram**

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

* + 1. **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.*

* + 1. **State Chart Diagram**

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

* 1. **Object Model**

*// The object model, represented in UML with class diagrams, describes the structure of a system in terms of objects, attributes, associations, and operation.*

* + 1. **Class Diagram**

*// Class diagram describe the structure of the system in terms of class and object. Class are abstraction that specify the attributes and behavior of a set of objects, whereas objects are entities that encapsulate state and behavior.*

* 1. **Database Specification**

*// A database specification consists of formal specifications of all data types used, objects declared, and operations over database objects defined. The database specification is stored in the database and can be used for the specification of an application program operating with it.*

* + 1. **E-R Diagrams**

*// An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education, and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships, and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.*

**Reference**

*// Use appropriate IEEE Reference format as described below.*

**Appendixes**

**N.B. Documentation Formation**

1. **Paper Size** (orientation): **A4** (portrait)
2. **Font** **Type**: Times New Roman
3. **Margins**: **1-inch** *top / bottom / right* and **1.3-inch** *left*
4. **Font** **Size**: **16** bold, all caps and centered for **heading 1** (chapter names, Appendix and references), **14** bold, left for **heading 2** and **12** bold, left aligned for **heading 3** and **12** for **normal** text of the contents.
5. **Line** **Spacing**: **1.5** throughout the contents
6. **Justification:** All paragraphs should be justified
7. **Bullets:** use only three types of bullets for major and sub topics
8. **Tables:** Table name and number should be displayed above the table and generated automatically
9. **Figures:** Figure name and number should be displayed bottom of the figure and generated automatically
10. The table/figure is numbered using chapter number, a full stop, and serial number, such as 2.1, this means 2 indicates the table is found in chapter 2 and 1 indicate that the first table in the chapter. For every chapter, the table/figure numbering should be reseated.
11. Each chapter must start on a new page.
12. Avoid copy/paste. Instead, take ideas and use your own English.
13. Never use I or my. Instead, use We or our.
14. **Page** **Numbering**: All pages shall be numbered consecutively in the bottom, center position. Uses 1 2 3 4 5 …

N.B. All pages prior to the Table of Contents page do not have page numbers.

**Follow the IEEE format for quoting References**

**Journal Paper:**

[1] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, 1997.

[2] A.E. Eiben, R. Hinterding, and Z. Michalewicz, “Parameter control in evolutionary algorithms”, *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124-141, 1999.

**Conference Paper:**

[3] C.W. Sung, S.Y. Yuen, “On the analysis of the (1+1) evolutionary algorithm with short-term memory,” in *Proc. IEEE Congress on Evolutionary Computation* (CEC), June 2008, pp. 235-241.

**Book:**

[4] J.H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.

**Link:**

[5] Regal, R. (2012, April 7). Globalizing Variables. KFU News. Retrieved from <http://www.kfu.news.sa>

[6] Al-Mulhem, K. (2009). Delphi Guide [Online Video]. Delphi Videos. Retrieved from http://www.youtube.com/watch?v=asd5thw.