Bahir Dar University

Bahir Dar Institute of Technology

Faculty of Computing

Requirement Analysis Document

for

Industrial project on [Your Project title here]

Submitted to the faculty of computing in partial fulfillment of the requirements for the degree of Bachelor of Science in [your program of study here]

Group members

	Name	ID Number
1.	_____	_____
2.	_____	_____
3.	_____	_____
4.	_____	_____
5.	_____	_____

Advisor :_____

[Year]

Bahir Dar University, Bahir Dar Institute of Technology

1. Declaration

The Project is our own and has not been presented for a degree in any other university and all the sources of material used for the project have been duly acknowledged.

----------	----------
Name	***Signature***
----------	----------
Name	***Signature***
----------	----------
Name	***Signature***
----------	----------
Name	***Signature***
----------	----------
Name	***Signature***

Faculty: Computing

Program: _____

Project Title: _____

This is to certify that I have read this project and that in my supervision and the students' performance, it is fully adequate, in scope and quality, as a project for the degree of Bachelor of Science.

---------- ----------

Name of Advisor ***Signature***

Examining committee members		**signature**	**Date**
1. Examiner1	.....	_____	.....
2. Examiner2	.....	_____	.....
3. Examiner2	.....	_____	.....

It is approved that this project has been written in compliance with the formatting rules laid down by the faculty.

Roles and Responsibilities of the Group Members

Fill the following role assignment matrix and put a tick mark(√) under each member in line with each task to indicate who has participated in carrying out the activities to produce the draft deliverable for discussion to the group so that they will discuss on the issue and come to consensus. Finally each group member will well understand the entire work of the project by sharing experiences among the colleagues.

List of Tasks	List members			
	Student1	Student 2	Student 3	Student 4
Task1				
Task2				
Task3				
Task4				
.				
.				
.				

Acknowledgment

List of acronyms

Write Expand form of abbreviations and short hand notations. You should put them in alphabetical order and all abbreviations that are included in the document must be written.

List of Figures

Generate a list of figures used in the document with their figure sequence by indicating their page number

List of Tables

Generate a list of tables used in the document with their table sequence by indicating their page number

Table of Contents

1. Introduction 3
1.1 Purpose 3
1.2 Document Conventions (Optional) 3
1.3 Intended Audience and Reading Suggestions 3
1.4 Project Scope 3
1.5 Definitions, Acronyms, & Abbreviations 3
1.6 Overview of the Document 3
2. Overall Description 4
2.1 Product Perspective 4
2.2 Product Features/Functions 4
2.3 User Classes and Characteristics 4
2.4 General Constraints 4
2.5 User Documentation (Optional) 5
2.6 Assumptions and Dependencies 5
3. Specific Requirements 6
3.1 User Requirements 6
3.2 System Requirements 7
3.2.1 Use case Diagram 8
3.2.2 Use Case documentation 8
3.2.3 *Business Rule Documentation* 10
4. External Interface Requirements 10
4.1 User Interfaces 10
4.2 Hardware Interfaces 10
4.3 Software Interfaces 10
4.4 Communications Interfaces 11
5. Analysis Models 11
5.1 Sequence Diagrams 11
5.2 Activity Diagrams 11
5.3 Data Flow Diagrams 11
Appendix 12
References 13

1. Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

1.2 Document Conventions (Optional)

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Project Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here. An SRS that specifies the next release of an evolving product should contain its own scope statement as a subset of the long-term strategic product vision.>

1.5 Definitions, Acronyms, & Abbreviations

1.6 Overview of the Document

<This section should

- *Describe what the of the SRS contains*
- *Explain how the SRS is organized>*

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Features/Functions

<Summarize the major features the product contains or the significant functions that it performs or lets the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or a class diagram, is often effective.>

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy.>

2.4 General Constraints

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.5 User Documentation (Optional)

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.6 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. Specific Requirements

<This template illustrates organizing the functional and non-functional requirements for the product by level of details. You may prefer to organize this section by system features use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

3.1 User Requirements

3.1.1. Functional User Requirements

< Group the requirements. Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind. Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. >

Grouping 1

Each group shall have Id like FR1, FR2…

Each requirement should be written using the format below

Requirement ID	XXX	Priority	Medium/Hign/Low
Name	*<The name of the requirement>*		
Description	*<Short description >*		
Source	*Source of the requirement eg(forms, rules, laws…*		
Related Requirements	*FR-YYY,FR-XYZ*		

FR –XYZ(X –refers to group ID, and YZ refers to specific requirement)

You can use the following example

FR1. User Management

Req. ID: 101	**Source:**	**Priority: high**
Name:	***New User registration***	
Description:	The system shall allow the system administrator to registers new users.	
Source		
Related Req.		

Grouping 2.

……

<Don't really say "Grouping 1." State the group name in just a few words.>

3.1.2. Non -Functional Requirements

Use the format for writing functional requirement to write this types of requirements too .Give them id to each NFR like NFR-1 , NFR-2,

<Specify quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: performance requirements, safety requirements, security adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>
Each requirements must also be measureable

3.2 System Requirements

<Take each functional requirement from section 3.1.1. and detailed the functional requirements. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary.>

3.2.1 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. The requirements specified in section 2.3.1 should be modeled using use case diagram.

3.2.2 Use Case documentation

This section should include a use case documentation by showing use case number, name, actor, description, pre-condition, post condition, priority, basic course of action and alternate course of actions. In the basic course of action and alternate course of action, you have to indicate the user interface and the business rule if needed.

System Use Case Template- All rows might not be important

Use Case ID:	*UC-01*
Use Case Name:	*Heritage Promotion*
Created By:	*Betemariam*
Date Created:	*01/01/2019*

Updated By:	*Samuel, Mizan, Azeb*
Date Updated:	*03/01/2019*

Actors:	*Tourism Development Directorate*
Description:	*[Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case. Use not more than three statements.]*
Trigger:	*[Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.]*
Preconditions:	*[List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:* *1. User's identity has been authenticated.* *2. All required files are loaded on the memory.*

	]
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. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, "How do I <accomplish the task stated in the use case name>?" This is done as a numbered list of actions performed by the actor, alternating with responses provided by the system.]*
Post conditions:	*[Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:* *1. A file is opened for the customer.* *2. Customer is registered.* *]*
Alternative Flows:	*[Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form "X.Y", where "X" is the number in the normal flow and Y is a sequence number for the alternative flow. For example, "5.3" would indicate the third alternative flow line number 5 in the normal flow.]*
Exceptions:	*[Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception.*
Priority:	*[The relative priority of the service provided by the given use case. Three categories of priority values can be assigned: High, Intermediate or Low]*
Frequency of Use:	*[How frequently is the use case used – every five minutes, once in a day, every other day, once in a week, etc]*
Business Rules:	*[Write any business rule of the organization that directly or indirectly affect the use case]*
Special Requirements:	*[Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.]*
Assumptions:	*[List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.]*
Notes and Issues:	*[List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due*

	date, and what the resolution ultimately is.]

3.2.3 *Business Rule Documentation*

Write relevant business rules for the system which will include (Identifier, name, description and reference, revision history).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. External Interface Requirements

4.1 User Interfaces

<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. Details of the user interface design should be documented in a separate user interface specification.>

4.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

4.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be

implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

4.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

5. Analysis Models

<List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements. Some of the models that you could use includes>

5.1 Sequence Diagrams

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. Show how objects operate with one another and in what order (chronological) to respond to the actors.

5.2 Activity Diagrams

Draw activity diagrams to show the operations/activities performed by use cases to achieve their functionality. Activity diagrams are drawn for each use case. Show the visual representation of activities/processes for use cases or methods or business processes.

5.3 Data Flow Diagrams

Draw state chart diagram for each class showing the states the object (class) passes through in its life. This models the how objects change from one state to another and when the change happens.

Appendix

Example: questionnaires used to gather requirement …

References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

You should you either APA or IEEE citation techniques.