Assosa University

College of Computing and Informatics

Department of Computer Science

Web-Based Correction House Management System for Assosa Correction House

A Software Project Documentation Submitted to the Department of Computer Science of Assosa University in Partial Fulfillment of Requirements for the Degree of Bachelor of Science in Computer Science

Prepared by:

Birhanu Dagnaw

Dagnaneh Abuneh

Meseret Goshu

Advisors' Name:

Main Advisor ……………Nebiyat Hamza (MSc.)

07 July 2021

Assosa, Ethiopia

Software Requirements Specification

for

Web-Based Correction House Management System for Assosa Correction House

Version 1.0 approved

Prepared By: Birhanu Dagnaw, Dagnaneh Abuneh, Meseret Goshu

Assosa University

07 July 2021

Acknowledgement

First of all, we want to thank the God who permits us to do and enable us to preparing this SRS to our system. Next a special gratitude we give to our advisor Nebiyat Hamza (MSc). For their contribution in stimulating suggestions and encouragement, helped us to coordinate especially in gathering information and writing this SRS. And we would like to express our deepest appreciation to all those who provided us the possibility to complete this SRS that they helped us by providing basic information's about the idea of the system. Finally, we would like to express more to all who provide information to support, encouragement, comments and gives useful information in any time when we want to ask information they give happy face and necessary information.

Contents

Acknowledgement ii

List of Tables v

List of Figures vi

1 INTRODUCTION 1

1.1. Document Purpose 1

1.2. Product Scope 2

1.3. Intended Audience and Document Overview 2

1.4. Definitions, Acronyms and Abbreviations 2

1.5. References 4

2 OVERALL DESCRIPTION 5

2.1. Product Perspective 5

2.2. Product Functions 5

2.3. User Classes and Characteristics 6

2.4. Operating Environment 6

2.5. Design and Implementation Constraints 6

2.6. User Documentation 6

2.7. Assumptions and Dependencies 7

3 SPECIFIC REQUIREMENTS 8

3.1. External Interface Requirements 8

3.1.1. The User Interfaces 8

3.1.2. The Software Interfaces 8

3.1.3. The Hardware Interfaces 8

3.1.4. The Communication Interfaces 9

3.2. Functional Requirements 9

3.3. System Use Case Modeling 9

3.3.1. Actors and Use Cases 9

3.3.2. Use Case Diagram 11

3.3.3. System Use Case Documentation 13

4 NON-FUNCTIONAL REQUIREMENTS.......... 22

4.1. Performance Requirements 22

4.2. Safety and Security Requirements 22

4.3. Software Quality Attributes 22

4.3.1. Reliability 22

4.3.2. Portability 22

4.3.3. Availability 22

4.3.4. Maintainability 23

4.3.5. Robustness 23

4.3.6. Scalability 23

4.3.7. Usability 23

List of Tables

Table 1. Terms and their definition... 3

Table 2. Abbreviations and Acronyms 3

Table 3 user class and characteristics 6

Table 4 Hardware interfaces 8

Table 5 Display prisoners Use case Description.. 13

Table 6. Login Use case Description 15

Table 7. logout Use case Description... 15

Table 8. Modify User account Use case Description... 17

Table 9.Create new user account Use case Description... 17

Table 10. Delete user account Use case Description 18

Table 11. Register new prisoner Use case Description.. 19

Table 12. Generate prisoner report Use case Description... 20

Table 13. Post announcement Use case Description.. 21

List of Figures

Figure 1. Client Server architecture 5

Figure 2 use case diagram for WBCHMS for Assosa Correction House 12

1 INTRODUCTION

WBCHMS for Assosa correction house SRS is designed to document and describe the agreement between the stakeholders, users and developers regarding the specification of the software product requested. Its primary purpose is to provide a clear and descriptive "statement of user requirements" that can be used as a reference in further development of the software system. This document is broken into a number of sections used to logically separate the software requirements into easily referenced parts. This SRS aims to describe the Functionality, External Interfaces, Attributes and Design Constraints imposed on Implementation of the software system described throughout the rest of the document. Throughout the description of the software system, the language and terminology used are unambiguous and consistent throughout the document.

1.1. Document Purpose

WBCHMS for Assosa correction house SRS provides a detailed description of the requirements for the web-based application that means it describes each of the functional and nonfunctional requirements of the system. Since our system is going to be developed from scratch this SRS allows for a complete understanding of what is to be expected from the newly introduced system which is to be constructed. The clear understanding of the system and its' functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project. This SRS will provide the foundation for the project. From this SRS, WBCHMS for Assosa correction house can be designed, constructed, and finally tested.

This SRS will be used by the system development team which is constructing the WBCHMS and the system end users. The Project team will use the SRS to fully understand the expectations of this WBCHMS to construct the appropriate software. The system end users will be able to use this SRS as a "test" to see if the constructing team will be constructing the system to their expectations. If it is not to their expectations the end users can specify why it is not to their liking and the team will change the SRS to fit the end users' needs.

1.2. Product Scope

The introducing web-based application is designed in a way that it can be easily extended to provide more features and will be easily customizable so that it can work according to every potential user. Admins (both System and prison admins) monitors basic functionality. There are different users for our system. Those users are administrators, registrar officers, information desk offices, and storage managers can access to some system functionalities with limited restrictions.

1.3. Intended Audience and Document Overview

This SRS document submits as senior project for partial fulfillment of bachelor degree in computer science. Therefore, the audients may be project examiners and invited guests. The SRS document is organizing into different sections. The first section describes the introduction part of SRS, which primarily focuses on identifying what is going to be achieved on this project. The second section addresses the overall description of the product and related issues. It includes product overview that mainly tries to describe the context and origin of the intended software and simple diagram that shows the interaction of system components. Then the functionality of the system is roughly stated. Section 3 is generally all about functional and external interface requirements of the system. It starts with the detail description of interface requirements and proceeds to hardware requirements with software requirements. At the end of this section, one gets the use case model, which encapsulates the entire system with its stakeholders. Section 4 is all about non-functional requirement of the system, which especially focuses on performance and security and safety requirements. Software quality attributes of the system is also discussed under this section as additional part. Finally, in the last section of this SRS, other requirements will be addressed.

1.4. Definitions, Acronyms and Abbreviations

Terms	**Definition**
Actor	It is a model element that interacts with a system [1].
Bootstrap	A front-end user interface template library.
Constraints	The possible limitations, which is imposed from the side of clients.

Functional requirement	It is a description of the service that the software must offer [2].
Non-functional requirement	define system attributes such as security, reliability, performance, maintainability, scalability, and usability [3].
Requirement	It is a description of the service that the software must offer.
Stakeholders	Any direct or indirect user of the system
Use case model	It is a model of how different types of users interact with the system to solve a problem [4].

Table 1. Terms and their definition

Abbreviations and Acronyms	**Meaning**
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
HTTPS	Hypertext transfer protocol secure
MySQL	My Structured Query Language
PHP	Hypertext Preprocessor
SRS	Software requirements Specification
UML	Unified Modeling Language
WBCHMS	Web Based Correction House Management System

Table 2. Abbreviations and Acronyms

1.5. References

[1] "actor," smartpedia, [Online]. Available: https://t2informatik.de/en/smartpedia/actor/. [Accessed 27 March 2021].

[2] "functional-requirement-specification-example,"guru99,[Online].Available: https://www.guru99.com/functional-requirement-specification-example.html#:~:text=A%20Functional%20Requirement%20(FR)%20is,software%20system%20or%20its%20component.&text=It%20can%20be%20a%20calculation,system%20is%20likely%20to%20perform.. [Accessed 25 March 2021].

[3] "nonfunctional-requirements,"scaledagileframework,[Online].Available: https://www.scaledagileframework.com/nonfunctional-requirements/#:~:text=Nonfunctional%20Requirements%20(NFRs)%20define%20system,system%20across%20the%20different%20backlogs.&text=They%20ensure%20the%20usability%20and%20effectiveness%20of%20the%20entire%2. [Accessed 30 March 2021].

[4] "use_case_model_CD178AF9,"utm,[Online].Available: http://www.utm.mx/~caff/doc/OpenUPWeb/openup/guidances/concepts/use_case_model_CD178AF9.html. [Accessed 01 April 2021].

[5] "uml-use-case-diagram," javatpoint, [Online]. Available: https://www.javatpoint.com/uml-use-case-diagram. [Accessed 28 March 2021].

[6] "use-case-description-basics," thebadoc, [Online]. Available: https://thebadoc.com/ba-techniques/f/use-case-description-basics. [Accessed 26 April 2021].

2 OVERALL DESCRIPTION

2.1. Product Perspective

WBCHMS for Assosa correction house is a self-contained system and It is a replacement for the existing one that is the manual system with computerized system. The product is self-contained product intended for use on web, and implementing client-server model. Its major components, which take part in the system. The client-side component interacts with the server to make requests and to retrieve data from the database. It then displays the data retrieved from the server to the user. Also, there is a database which will keep all the records that done by user while visiting the web application.

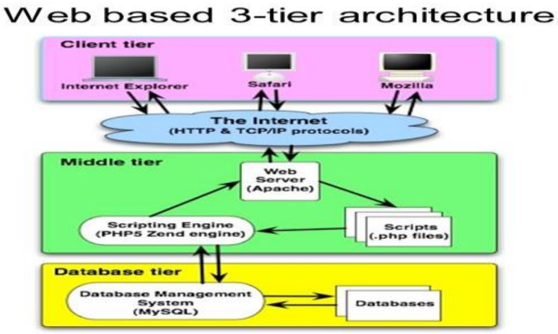

Figure 1. Client Server architecture

2.2. Product Functions

- Login
- Register new prisoner
- Register Visitors
- Prisoner information management
- User account management
- Modify prisoner Information
- Generate prisoner report
- Post announcement and so on.

2.3. User Classes and Characteristics

User Class	Description
System admin	System admin has privilege of user account management
Prison Admin	Prison admin has the ability of generating reports, post announcements, checkout prisoners and so on
Information Desk Officer	Information Desk Officer has privilege managing visitors
Registrar Officer	System admin has privilege of registering prisoners and so on.
Storage Manager	Storage manager has privilege of registering prisoners' properties.

Table 3 user class and characteristics

2.4. Operating Environment

WBCHMS for Assosa correction house can run on both computer and mobiles going on different types of internet browsers such as Chrome, Mozilla Firefox, UC browser, and other browsers. This software will be used only for Assosa correction house administration.

2.5. Design and Implementation Constraints

There are some constraints used to develop this system. If these constraints are available in the system the whole system shall be functional. These constraints are: -

- Xampp 7.1.12 server will be used as MySQL database application.
- Php is used to develop back end of WBCHMS

2.6. User Documentation

The product will include a user manual. The user manual will include product overview, complete configuration of the required software and hardware, technical details and contact information. A user manual explaining what the functionality and usage of the visualization software will be required.

Additionally, the team developing the software would potentially need to be available in case of questions or problems with the software once it starts being used by other users/administrators. Our source code will be heavily documented. This will make it easier for other people to understand and continue development on our product if needed.

2.7. Assumptions and Dependencies

The first assumption about the product is it depend on the internet connection. The product will always be used on personal computers and mobile phones that have enough performance mostly in smart phone and desktop with in the local area network. If the phone and PC or desktop does not have enough hardware resources and available broadband Wi-Fi or data connection the system does not perform its task efficiently. In this case our assumption is there will be enough internet connection for our system and for us to develop the system in order to perform our activities properly. If the lack of connection occurs, for the developer it is difficult to develop the target software in given time and difficult to use properly the system for end user.

We assume that everyone that uses the system has access to the internet at speeds of 56kb or above. Another assumption is that a user has basic computer and mobile use skill and other general knowledge of the how to use the system.

3 SPECIFIC REQUIREMENTS

3.1. External Interface Requirements

3.1.1. The User Interfaces

The user interfaces consisted of forms developed and formatted using HTML, CSS and Bootstrap. Different forms will be developed to enable the users perform the following tasks: Login to the program that takes a user where they are supposed to retrieve and submit information. Register Users which is done by the system admin by creating accounts to authorized users. Add or enter a prisoner's details. Edit Prisoner's details. Get the General Report by the data entrant. Get the Management Report for decision making. Search for a particular inmate using his identity card number or his names. Search either by crime (charge), age variance, gender, prison, remands.

3.1.2. The Software Interfaces

- Client on Internet: Web Browser, Operating System (any)
- Web Server: Apache Web server, Operating System (any)
- Data Base Server: MySQL, Operating System (any)
- Development End: XAMPP (PHP, JavaScript, HTML, CSS), Database server (MySQL), Operating System (Any), Apache Server (Web Server)

3.1.3. The Hardware Interfaces

CLIENT SIDE			
	PROCEESOR	RAM (Random Access Memory)	DISK SPACE
Web browser Any but Chrome recommended	Pentium and Above	512 MB	8 GB
SERVER SIDE			
Apache web server 7.0.2	Pentium and Above	512 MB	4 GB
MySQL 5.1.5	Pentium and Above	512 MB	2 GB

Table 4 Hardware interfaces

- Additional disk space for storing data size and additional memory may be required.

3.1.4. The Communication Interfaces

Client on Internet will be using HTTPS protocols as a communication interface to facilitate communications between the client and server.

3.2. Functional Requirements

A requirement specifies a function that a system or component must be able to perform. Functional requirements are those requirements that are explicitly stated. They are observable tasks or processes that must be performed by the system under development.

The following are some of functional requirement used to automate the system:

- The system allows registrar officer to register prisoner’s information.
- The system allows Prison admin to generate reports, to search prisoner, to transfer prisoner, to checkout prisoner.
- The system allows system admin to manage user accounts.
- The system allows Information desk officer to manage visitors.
- The system allows prison Admin to post announcements.
- The system allows Registrar Officer to assign room for prisoners.
-
- The system allows Storage manager to register and manage properties of prisoners.

3.3. System Use Case Modeling

3.3.1. Actors and Use Cases

In this section, the requirement is documented as use case, which is a list of actions or event steps typically defining the interactions between a role known as an actor in UML and a system to achieve a goal. The actor can be a human or other external system.

Use case identification is essential for simplifying the system and better understand it in simple terms as a result it will assists in the implementation of the system.

In WBCHMS the following actors are available:

I. System Admin

This actor is the administrator of the system who is in charge of the system and user accounts. The responsibilities or use cases of System admin include:

- Creating user account
- Modifying user account
- Deleting user account
- Displaying user accounts.
- Searching for user account
- Backup for data
- Activate Account
- Deactivate Account

II. Prison Admin

This actor is the administrator of the correction house, whose responsibilities on the system include:

- Displaying prisoners
- Post announcements
- Generating report of prisoners
- Checkout prisoners
- Give Amekiro service

III. Registrar Officer

This actor is an officer working in registrar office, whose duty is to manage prisoner information. The responsibilities of registrar officer include:

- Registering new prisoner
- Modifying prisoner’s information

- Displaying prisoners
- Searching for prisoner
- Assign Rooms
- Manage ketero(ቀጠሮ) cases

IV. Information desk officer

This actor is an officer who is responsible for managing visitor information. The responsibilities of information desk officer include:

- Register visitor’s information
- update visitor’s information
- Delete visitor’s information
- manage visitors’ information
- view visitors’ status

V. Storage manager

This actor is an officer who is in charge of controlling property storage area. The responsibilities of storage manager include:

- Registering new property
- Modifying property information
- Removing property
- Displaying properties list
- Searching for property

3.3.2. Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application [5].

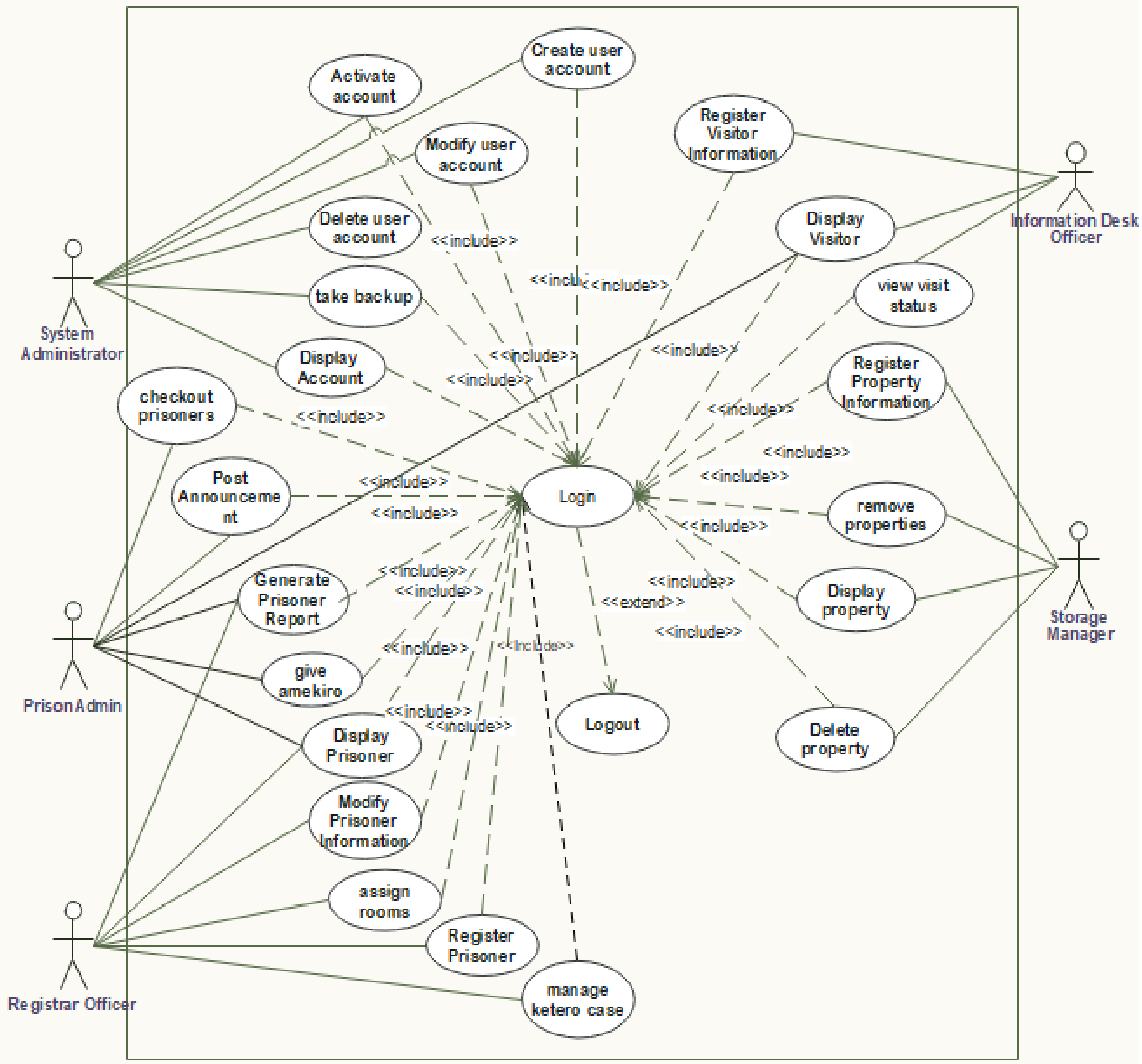

Figure 2 use case diagram for WBCHMS for Assosa Correction House

3.3.3. System Use Case Documentation

A use case description is a text-based narrative of a functionality comprised of detailed, step-by-step interaction between the actor and the system. It describes the outcomes of an action taken to accomplish a specific goal [6].

Here are the use case descriptions for WBCHMS for Assosa correction house.

Use case identifier	WBCHMS_UC01
Use case name	Display prisoners
Actor	Prison Admin, registrar officer
Description	The system provides a mechanism to list and view prisoner information
Precondition	The user must be either prison admin or registrar officer.
Postcondition	The prisoner's information displayed
Basic Flow of Events	
User action	**System response**
1. The user selects the view option. 3. The user selects what to display. 5. End use case.	2. The system displays list of things this account type can display. 4. The system displays whatever the user selects.
Exceptional Flow of Events E	
	E4, "Something went wrong! Please try later!" message displayed. E5, use case ends.

Table 5 Display prisoners Use case Description

Use case identifier	WBCHMS_UC02
Use case name	Login
Actors	Prison admin, system admin, information desk officer, registrar officer, storage manager.
Description	To get access to the system according to the authorization given by the system admin, through verification of user name and password.
Precondition	The user must have an account.
Postcondition	The user logged in to the system
Basic Flow of Events	
User action	**System response**
1.The user selects login button 3.User types in user name and password 5.User click login button.	2.Provide a log in interface. 4.The system verifies the user name and password. 6. The System authenticate the given username and password from database. 7. The System allow the user to access its own granted privileges. (Logs in user). 8.Usecase Ends
Alternative Flow of Events A	
	A6. “Please fill the user name and password correctly.” message displayed. A7. “There is no any user registered in that username! Try again” message displayed. A7. “The password is incorrect! Forget your Password?” message displayed. A8. Go to step 2.

Exceptional Flow of Events E	
	E7. “Something went wrong please try again later” message displayed. E8. Use case ends

Table 6. Login Use case Description

Use case identifier	WBCHMS_UC03
Use Case Name	Logout
Actors	All actors
Pre-Condition	User must login to the system.
Post condition	User logged out form the system.
Description	How a user logged out from The System Web.
Basic Flow of Events	
User action	**System response**
1. User clicks account option dropdown. 2. User clicks logout button.	3. The System remove sessions. 4. The System displays login page. 5. Use case ends.
Exceptional Flow of Events	
	E4. “Something went wrong please try again later” message displayed. E5. Use case ends

Table 7. logout Use case Description

Use case identifier	WBCHMS_UC04
Use case name	Modify user account
Actor	System admin
Description	The system admin updates or changes some attributes of user accounts.
Precondition	i. The user account to be modified must exist. ii. The one making the changes must be system admin.
Basic Flow of Events	
User action	**System response**
1. The user selects the manage user accounts.	2. The system provides list of how to manage user account.
3. The user selects the modify/update user account option.	4. The system provides a search engine.
5. The user types in the user name.	6. The system provides editable information in the form.
7. The user modifies the information.	
8. The user clicks saves changes.	9. The system shows success message.
	10. End use case.
Alternative Flow of Events	
	A9. “Please fill the form correctly.” message displayed. A10. Go to step 6.
Exceptional Flow of Events	
	E9. “Adding new manager failed. No changes made. Try later!” message displayed. E10. Use case ends.

Table 8. Modify User account Use case Description

	WBCHMS_UC05
Use Case Name	Create new user account
Actors	System Admin
Pre-Condition	User must logged as system Administrator.
Post condition	New user account added.
Description	How a new user added.
Basic Flow of Events	
User Action	**System response**
1. Click on add new user button. 3. Administrator fills the form. 4. Administrator clicks add button.	2. The system provides a form to be filled with all the necessary information. 5. The System validate the filled data. 6. The System add the user into the database 7. “You added a new user successfully” message displayed. 10. Use case ends.
Alternative Flow of Events A	
	A6. “Please fill the form correctly.” message displayed. A7. Go to step 2.
Exceptional Flow of Events E	
	E6. “Adding new manager failed. No changes made. Try later!” message displayed. E7. Use case ends.

Table 9.Create new user account Use case Description

Use case identifier	WBCHMS_UC06
Use case name	Delete user account
Actor	System admin
Description	The system admin removes user account in case the user is fired, deceased, or transferred.
Precondition	i. The user must be system admin. ii. The user account to be deleted must exist.
Postcondition	the user account is removed
Basic Flow of Events	
User action	**System response**
1. The user selects the manage user accounts.	2. The system provides list of how to manage user account.
3. The user selects the delete user account button.	4. The System shows a confirmation dialog.
5. The user clicks confirm button	6. The System deletes a user from users list in database 7. "You delete a user successfully" message displayed. 8. Use case ends.
Alternative Flow of Events A	
	A6. "Deleting a user canceled." message displayed. A7. Go to step 2.
Exceptional Flow of Events E	
	E6. "Deleting a user failed. No changes made. Try later!" message displayed. E7. Use case ends.

Table 10. Delete user account Use case Description

Use case identifier	WBCHMS_UC07
Use case name	Register new prisoner
Actor	Registrar officer
Description	The registrar officer records every information of the new prisoner that is required on the form provided by the system.
Precondition	The user must be a Registrar officer.
Postcondition	The prisoner information should be registered
Basic Flow of Events	
User action	**System response**
1. The user selects register new prisoner option. 3. The user types in all the required information in the form. 4. The user clicks on the register button.	2. The system brings up an empty form. 5. The System validate the filled data. 6. The System add the manager into the database 7. “You added a new prisoner successfully” message displayed. 8, Use case ends.

Table 11. Register new prisoner Use case Description

Use case identifier	WBCHMS_UC08
Use case name	Generate prisoner report
Actor	Registrar officer, Prison admin
Description	The system provides a mechanism for generating statistical report of the prisoners that are registered.
Precondition	The user must be either a registrar officer or prison admin.
Postcondition	report should be generated
Basic Flow of Events	
User action	**System response**
1. The user selects generate report option.	
	2. The system brings up available options of report.
3. The user selects prisoner report.	4. The system automatically generates the report.
	5. The system displays the report.
	6. End use case.
Exceptional Flow of Events E	
	E4. “generating a report failed. Try later!” message displayed. E5. Use case ends.

Table 12. Generate prisoner report Use case Description

Use case identifier	WBCHMS_UC09
Use case name	Post announcement
Actor	Prisoner admin
Description	The prisoner admin can post any announcement, that will show up on the home pages of every user of the system.
precondition	the prison admin must be logged in
postcondition	the announcement is posted in each user's dashboard.
Basic Flow of Events	
User action	**System response**
1. the user selects the post announcement option. 3. the user types in the announcement in the field provided. 4. the user clicks on the post button.	2. the system provides a field to type in the announcement to be made. 5. the system shows a success message. 6. end use case
Exceptional Flow of Events E	
	E5. "Posting announcement failed. Try later!" message displayed. E6. Use case ends.

Table 13. Post announcement Use case Description

4 NON-FUNCTIONAL REQUIREMENTS

4.1. Performance Requirements

The performance of the system will highly depend on the performance of the hardware and software components of the installing computer. When we consider about the timing relationships of the system the load time for user interface screens shall take no longer than three seconds. It makes fast access to system functions. The log in information shall be verified within five seconds causes' efficiency of the system. Returning query results within five seconds makes search function more accurate.

4.2. Safety and Security Requirements

There are several user levels in WBCHMS for Assosa correction house, Access to the various subsystems will be protected by a user log in screen that requires username and password. This gives different views and accessible functions of user levels through the system. Maintaining backups ensure the system database security. System can be restoring in any case of emergency.

4.3. Software Quality Attributes

4.3.1. Reliability

The system shall be highly reliable by performing all the functionalities expected by the end users. This is the most quality attribute needed by the end users.

4.3.2. Portability

This web-based application is viewable and fit with any standard web browsers, various operating systems such as Windows, Linux, Macintosh and on devices like personal computers, mobile phones and tablets.

4.3.3. Availability

Our web-based application will be accessible 24/7, anywhere and via PC, mobiles devices and tablets with an internet connection.

4.3.4. Maintainability

Whenever our system is needed to be changed or if it is needed to add some other extra functionality, the current developed system is going to be kept as it is and correct defects with making changes.

4.3.5. Robustness

Our system is strength full to handle system functions accurately and maintain the database without facing to unexpected failures and it is also error tolerant.

4.3.6. Scalability

The system will be designed to accommodate increased volumes, workloads and users.

4.3.7. Usability

Usability is the system support of the execution of user tasks (i.e., presentation of information and management of user interaction).

It is about:

- How easy it is for user to learn the system.
- How easy it is for user to memorize steps.
- How efficient it is to use the system**.**