****

**ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY**

**SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING**

**Computer science and engineering department**

**FSE Mini project**

**Store Inventory Management System**

(WEB APPLICATION)

BY:

* Mustefa Badri ……………R/0904/08
* Robel Haileselassie ……………R/1005/08
* Milkesa Abera ……………R/0859/08

Submitted to: -instructor Anteneh

Submission Date: April 2 2019 GC

Acknowledgment

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to Adama Science and Technology University for the opportunity it provided us to work on this project and explore further knowledge.

We would like to express our deepest gratitude towards our advisor Mr. Anteneh for his guidance and constant supervision as well as his kind co-operation and encouragement, which helped us in completion of this project, which is a Software Requirement Specification (SRS) document.

Our thanks and an appreciation go to all of those who have willingly supported and helped us out with their abilities.

**Abstract**

Store Inventory Management System (SIMS) is the overall planning, coordination, and control of a project from inception to completion aimed at meeting Shop or supermarket requirements in order to produce a functionally and financially visible project. This paper contains starting from data gathering, system analysis, and system design and implementation part of construction Project management system. The system analysis phases show that what the existing system does and what the problems are. However, the design phase shows the new proposed system and it shows the solutions to the problems of the existing system and finally implementation part shows what look like and how it works and shows the user interface.

**Acronym**

SIMS……………………Store Inventory Management System

DB……………………… database

HTML……………………Hyper Text Markup Language

CSS………………………Cascading Style Sheet

GB………………………Giga Byte

RAM…...........................Random Access Memory

SQL……………………. Structural Query Language

PHP……………………. Hypertext Processor

UML…..........................Unified Modeling Language

Contents

[Chapter one 1](#_Toc5068002)

[1.1 Introduction 1](#_Toc5068003)

[1.2. Background 1](#_Toc5068004)

[1.3 Statement of the Problem 1](#_Toc5068005)

[1.4 Purpose of the project 1](#_Toc5068006)

[1.5 Team composition 2](#_Toc5068007)

[1.6 Objectives of the project 2](#_Toc5068008)

[1.6.1 General Objective 2](#_Toc5068009)

[1.6.2. Specific objective 2](#_Toc5068010)

[1.7. Feasibility study 2](#_Toc5068011)

[1.7.1. Technical feasibility 2](#_Toc5068012)

[1.7.2. Operational feasibility 3](#_Toc5068013)

[1.7.3 Economical Feasibility 3](#_Toc5068014)

[1.8 Scope and limitation 3](#_Toc5068015)

[1.8.1 Scope 3](#_Toc5068016)

[1.8.2 Limitation 3](#_Toc5068017)

[1.9 Significance of the project 3](#_Toc5068018)

[1.10. Methodology 4](#_Toc5068019)

[1.10.1. Data collection methodology 4](#_Toc5068020)

[1.11. Development tools 5](#_Toc5068021)

[1.12. Test plan 5](#_Toc5068022)

[1.12.1 Requirement testing 5](#_Toc5068023)

[1.12.2. Unit testing 5](#_Toc5068024)

[1.12.3. Integration testing 6](#_Toc5068025)

[1.12.4. System testing 6](#_Toc5068026)

[1.12.5. Acceptance Testing 6](#_Toc5068027)

[1.13. Project Execution Phase 7](#_Toc5068028)

[1.14. task and schedule 8](#_Toc5068029)

[Chapter Two 9](#_Toc5068030)

[Description of existing system 9](#_Toc5068031)

[2.1 Major function of existing system 9](#_Toc5068032)

[2.2. Users of current system 9](#_Toc5068033)

[2.3. Drawback of current system 9](#_Toc5068034)

[2.4. Business rule 10](#_Toc5068035)

[Chapter Three 11](#_Toc5068036)

[Proposed System 11](#_Toc5068037)

[3.1 Overview 11](#_Toc5068038)

[3.2 Functional Requirement 11](#_Toc5068039)

[3.2.1 Functional Process Requirements 11](#_Toc5068040)

[3.3. Non-functional requirement 11](#_Toc5068041)

[3.4 SYSTEM MODEL 12](#_Toc5068042)

[3.4.1 Scenarios 12](#_Toc5068043)

[3.4.2. Use case model 17](#_Toc5068044)

[3.5 Object Model 23](#_Toc5068045)

[3.5.1 Data dictionary 23](#_Toc5068046)

[3.5.2 Class diagram 24](#_Toc5068047)

[3.6. Dynamic model 25](#_Toc5068048)

[3.6.1. Sequence diagram 25](#_Toc5068049)

[Chapter Four 42](#_Toc5068050)

[4.1. Overview of system design 42](#_Toc5068051)

[4.1.1. Purpose of the system design 42](#_Toc5068052)

[4.1.2 Design goal 42](#_Toc5068053)

[4.2. Proposed system architecture 43](#_Toc5068054)

[4.2.1. System process 44](#_Toc5068055)

[4.2.2. Subsystem decomposition 47](#_Toc5068056)

[4.2.3 Hardware/Software Mapping 48](#_Toc5068057)

[4.2.4. Persistent data management 49](#_Toc5068058)

[4.2.5. Component diagram 53](#_Toc5068059)

[4.2.6. Deployment diagram 54](#_Toc5068060)

[4.2.7. Database design 55](#_Toc5068061)

[4.2.8 Access control 56](#_Toc5068062)

[4.2.9 User interface design 57](#_Toc5068063)

[Reference 58](#_Toc5068064)

# **Chapter one**

# **1.1 Introduction**

Nowadays, we have many shops in the world including Ethiopia. People buy and sell materials. sellers supply what people wants. But, the seller or the owner of the shop cannot control all things that what they have, what is happening in his/her shop (for owner) because of may be some materials lost in the shop, some person borrows money etc…

Store inventory management system is a record of all items available for use in your daily business operation. It is all about having the right inventory at the right quantity, the right place, and at the right time you will be looking for space to store these items.

# **1.2. Background**

Lemi shop founds in Adama Science and Technology university (ASTU) around Caffe students. This shop purchases a lot of materials, arranges this material in to the shelve, records the price of material on notebook, records who borrowed material. Lemi shop uses manually rather than web-based application in his whole life. So, our Store Inventory management system changes a manual record into a web-based system record and also records a daily profit of the shop, Controls the export and import materials and so on.

# **1.3 Statement of the Problem**

People, mostly having a big shopping area faces with these problems:

* The owner of the shop doesn’t know what kind of actions happens in the shop individually.
* Doesn’t know exactly how much quantity does he/she sell and how much the quantity left.
* Doesn’t know how much money does he/she earns per day?
* They did not recognize all materials that will sell and did not recognize some materials placement.
* Some materials that expires fast did not recognize easily before some buyers tell them.
* Don’t know what the other branch wants to import and export?

# **1.4 Purpose of the project**

The main purpose of the project is overcome the problems of the existing system or current system that exist in operations sellers and changing the whole current system in to computerized system without affecting the structure of the existing system. In addition to this the owner can access easily and controls at any place.

# **1.5 Team composition**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Team Composition |  |  |
| Name | Responsibility | Main activity | Other |
| Mustefa Badri | Team leader | Coordinate and lead the Members, Design, Architecture, Implementation and  Documentation | Participate in all activity |
| Milkesa abera | Secretary General | Requirement Analysis,  Design, Architecture,  Implementation and  Documentation | Participate in all activity |
| Robel Haile Selassie | Secretary General | Requirement Analysis,  Design, Architecture,  Implementation and  Documentation | Participate in all activity |

Table 1.1 Team composition

# **1.6 Objectives of the project**

## **1.6.1 General Objective**

The general objective of this project is to develop web based for store inventory management system and to change the manual system to computerized

system in order to solve problems.

## **1.6.2. Specific objective**

In order to achieve the main objective, we have developed the following specific objectives:

* Studying about problem of the existing system of the store inventory management.
* Analyzing the gathered information.
* Designing the proposed system.
* Implementing the system.
* Testing the system.

# **1.7. Feasibility study**

Feasibility study is essential to evaluate the cost and benefits of the new system. On the basis of

the feasibility study decision is taken on whether to proceed or to cancel the project. It includes

the following.

## **1.7.1. Technical feasibility**

Technical Feasibility study is about testing if the existing technology has a potential to develop or unable to acquire the proposed system. Implementation of the proposed system isgoing to be developed by following the Object-Oriented System Development technique. The project members don’t have enough knowledge about PHP, Java script, CSS, MySQL database and Enterprise architect to design the system. The system has definitely a positive impact on sellers by providing easy controlling on his/her supermarket on day to day business operation. Therefore, we can say the project is technically feasible.

## **1.7.2. Operational feasibility**

Operational feasibility is a measure of how well a proposed system solves the problems. It reduces the time for searching the materials by where the materials are placed and how much quantities left. And also, it helps to know how much profit the system has. Therefore, we can say the project is operational feasible.

## **1.7.3 Economical Feasibility**

Economic feasibility evaluates the cost of the system development against the ultimate income or a benefit gets from the developed system. So, the project is economically feasible because this system reduces the number of sellers and other extra workers such as auditor of the supermarket. In addition to this the customers satisfies by the service because all materials are ready to sell. So, the supermarket gets more profit.

# **1.8 Scope and limitation**

## **1.8.1 Scope**

* To know if he/she earns good or not.
* Helps to recognize the seller all the materials what he/she have, where it placed, and is that expire or not.
* Helps how much quantity lefts to sell.
* Helps to know a customer borrows or not.
* Easily controls supermarkets branch export and import.

## **1.8.2 Limitation**

* Since, it is a web-based system the service given by the system is not accessed by users as long as there is no connection.
* Sellers cannot sell on the internet.
* Because of the shortage of time we cannot perform our activity effectively.

# **1.9 Significance of the project**

Significances of this store inventory management System are:

* Minimizing the time wasted of auditing your money.
* better recording on quantity of your materials.
* Better controlling your shop.
* Easily know your profit.
* Better self-confidence on your shop.
* Improve buyers’ satisfaction.
* Man power reduction (for a big shop).

# **1.10. Methodology**

First, we gather data like written below then we decide what methodology to choose.

## **1.10.1. Data collection methodology**

To propose this project, we have used the following data collection method.

**a. Interviewing**

We have made an interview with owners of shop and workers at super markets to get an overview of the store inventory management System and the problem of it.

**b. Observation**

we have observed different supermarkets.

**c. Document analysis**

we have analyzed document we get a name of many persons that borrow materials and names of materials with its corresponding cost. This shows the overview of the existing system.

**implementation methodology**

Implementation is coding of all functions specified by requirement analysis and design. To

perform this we will use client server architecture. The server provides service to the client and

the client requests service from server. So we will have servers and clients (computers) on the

system will be hosted.

**System development methodology**

Since the system requirement is well known, we will use water fall model for the system development. i.e. starting from requirement analysis till testing of the project.

We choose water fall model, because of the following reason:

* It is very simple to understand and use.
* In a waterfall model, each phase must be completed fully before the next phase can begin.
* Requirements are well known, clear and fixed.
* There are no ambiguous requirements.
* The project is short.

# **1.11. Development tools**

These are tools that will help us to develop the application and to document the system facts.

|  |  |  |
| --- | --- | --- |
| Activities | Tools | Purpose |
|  | Hypertext markup language | For configuration |
| front end implementation software tools | Cascading style sheet | For layout design, content decoration in user interface  design and to give the style of  the interface |
|  | Java script/JQuery | For validating client side monitoring language |
| back end implementation software tools | Hypertext preprocessor | Backend implementation  of the project. |
|  | MySQL Server | For database |
|  | xampp sever | To run the project |
| Diagram tools | Enterprise Architect | To develop UML diagram of the project |
| Browsers | All Internet Browsers | to display web application via internet. |
| Documentation | MS Word | To Document our project |
| Editors | Notepad++ | For editing the source code |
| Computers | Pc | To write and run our programs. |
| Flash | Flash | For transfer. |

Table 1.2. Development tools

# **1.12. Test plan**

Testing is the process of executing a program with the intension of finding errors.

## **1.12.1 Requirement testing**

Whether there is an error or not in our system, we will perform the following testing

strategies; these strategies are used to test the functionalities and the working style of the

system.

## **1.12.2. Unit testing**

To perform these, we will use two approaches/techniques of unit testing

**A. Black box testing**

In this technique, we will test to see if the function of the system is fully operational or error free. This includes, testing the interface of the system rather than the logical structure of the system. We used this testing technique for the following reasons:

* More effective on larger units of code.
* Tester needs no knowledge of implementation, including specific programming languages.
* Tester and programmer are independent of each other.
* Tests are done from a user's point of view.
* Will help to expose any ambiguities or inconsistencies in the specifications.

**B. White box testing**

We will use this approach to know the internal working style of the system, test that all internal operations are performed according to specifications and all internal components have been exercised and the logical path of the system are correct.

We use this testing technique for the following reasons:

* It is easy to find out which type of input/data can help in testing the application effectively.
* To optimize the code.
* It helps in removing the extra lines of code, which can bring in hidden defects.
* Early detection of errors during software development.

## **1.12.3. Integration testing**

In these strategies the group member will focuses on testing the design and construction of the software architecture. To do the two techniques of integration testing have considered.

**1. Top –Down integration testing**

This will perform starting from the top module up to the last or bottom module individually (tests were run as each individual module is integrated).

**2. Bottom-up integration**

We will begin with the lowest –level modules which are combined to cluster, or build that perform a specific software sub-function (top-level).

## **1.12.4. System testing**

After all of the above testing is checked we will test our system by other peoples and we will conduct some comments how they get our system.

## **1.12.5. Acceptance Testing**

This testing is done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It includes:

* Alpha- Conducted by users to ensure they accept the system.
* Beta- Users use real data, not test data.

# **1.13. Project Execution Phase**

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Required time | Required tools | Function |
| Data collection  and gathering | Dec20-21 | Pen Paper  Ms word  Computers  Flash | Used to get good information about existing system structure and develop proposed system. |
| Requirement  elicitation and  analysis | Dec20-21 | Pen  paper  computers  flash  MSWORD. | Identify requirements and define the functionalities of the project (elicitation). Analyze, validate and understand the requirement (analysis). |
| Design | Dec22-24 | Pen  paper  computers  flash  SQL server  MSWORD. | Transform user requirement into implementation suitable form. To identify and understand the class in order to reuse, refine and remove vague classes. |
| Coding and implementation | Mar27-Apr2 | Html JavaScript CSS PHP XAMP server Browser MYSQL server Computers Flash | Software coding phase is typing code within the selected programming language in order to implement the functionalities of the project. |
| Testing | Mar27-Apr2 | Computers Browsers  servers | Identify faults and errors in the project. |

**T**able 1.3 project execution phase

The application contains the following features:

* The system focuses record the Company project information in to the database system, can control and manage your profit, the borrowed person and quantity of materials left and sold ones.
* Seller and the owner of the shop have own login ID, update and delete their materials and anything they want.

# 1.14. task and schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Months |  |  |  |
| Phases | Dec20-21 | Dec22-24 | Mar27-Apr2 | Mar27-Apr2 |  |
| Requirement gathering and analysis |  |  |  |  |  |
| Design |  |  |  |  |  |
| Implementation |  |  |  |  |  |
| Testing |  |  |  |  |  |
| Maintenance |  |  |  |  |  |

Table1.4. task and schedule

# **Chapter Two**

# **Description of existing system**

Currently in shops is fully manual system. It is necessary to know the existing system to develop a better system. The target area of this proposed system is web-based Store Inventory Management System (i.e. change manual system in to computerized system).

# **2.1 Major function of existing system**

The major function of the existing system of Inventory management includes the following:

* Monitor quantity
* Monitor borrowing
* Calculate profit
* Record the materials
* Check the expiry date
* Control export and import materials

# **2.2. Users of current system**

This illustrates the actors involved in the current system:

**1.seller**

* Record materials.
* Arrange materials at their places.
* record the quantity of materials.
* Buy materials(optional)
* Lend to buyers.
* Sell materials.
* Export and import material

**2.owner(admin)**

* Controls the system.
* Monitor seller

# **2.3. Drawback of current system**

They face the following problems:

* Lack of time management
* don’t know what kind of actions happens in the supermarket individually.
* Doesn’t know exactly how much quantity does he/she sell and how much the quantity left.
* Doesn’t know is that profitable or not.
* Lack of recognize all materials and their placement.
* Lack of recognizing expiry date.
* Difficult to generate report

# **2.4. Business rule**

The following are business rules of system.

1. The seller who is not registered does not allowed to perform any activity to the system.
2. The admin is the only user allowed to manage users and any activity related.
3. The user must be authenticated to access the system.
4. Materials must be registered before any kind of selling began.
5. The seller must arrange the materials.
6. Report must be written.

# **Chapter Three**

# **Proposed System**

# **3.1 Overview**

The proposed system is designed to replace the manual system of store inventory management system in to computerized system and also designed to control all materials in the shop by computerized system easily rather than a manual system.

The system we are going to develop will provide a well-organized data storage system and will bring efficient, safe and easy access data and information by secured online network throughout the store inventory and improve the profits of business.

# **3.2 Functional Requirement**

The functional requirements descry be the core functionality of the application. This section includes the data and functional process requirements.

## **3.2.1 Functional Process Requirements**

Process requirements describe what the application must do. Process requirements relate the entities and attributes from the data requirements to the users. State the functional process requirements in a manner that enables the reader to see broad concepts decomposed into layers of increasing detail.

* manage account (create, update, delete, view account)
* record material and its price
* register sellers (add, delete, update)
* generate report about daily activities
* generate borrowing table
* allows import and export materials among branches
* Notifies expire date and quantity left

# 3.3. Non-functional requirement

Non-functional requirements are requirement, which has no essential for the system, but it can support and give more quality for the system.

1. **Users interface requirement**

* Our system is menu driven and attractive.
* It’s user friendly.
* The system should support error-handling mechanism that display graphic approach and the system guide the user what will be the next action.

**B. Authentication Requirement**

* The system support user name and password to authentic.
* The system has different privilege to protect intruding.

**C. Robustness (Error handling requirement):** The system has error handling mechanisms that is, as errors occur it will not stop functioning rather provide error manages and back to the previous page to give chance to reenter data and process the task by beyond the error.

**D. Resources:** The system is compatible with specified hardware and software environment

**E. Usability**

The system is user friendly. The new system provides web application user interfaces that are compatible with any browsers.

* The system shall provide the easy access
* The system should be easy to understand.
* Unauthorized person should not use the system; rather just view the main page.
* No one can change the password without login to the system

**F. Hardware consideration:** the following sub-sections discuss the various aspect of hardware requirement.

* COMPUTER
* Android running smart phone
* Server

**G. Software consideration:**

* Network connection
* Server connection

**H. Reliability:** The system should be reliable. Appropriate error messages will be provided to users whenever incorrect information is inserted and handle the occurrence of that error.

# **3.4 SYSTEM MODEL**

## **3.4.1 Scenarios**

**Scenario**: 1

**Name of scenario**: register

**Participating actors:** admin

**Entry condition:**

 Internet connection should be available.

 They have to navigate to SIMS’s URL

**Flow of events:**

1. Admin clicks REGISTER button

2.choose either for Admin or seller to register

3.Registration form will be displayed

4. Admin fills required and optional fields

5. Click NEXT button

6. User’s detail displayed at this page with edit tab.

7. If user wants to edit his/her information, then click EDIT tab

8. Edit incorrect information and click SUBMIT button

9. Success message will be displayed

**Exceptional flow:** if the user enters wrong data, the system displays a message to

enter correct data.

**Scenario**: 2

**Name of scenario**: login

**Participating actors:** all users

**Entry condition:**

 Internet connection should be available.

 They have to navigate to SIMS’s URL

 The user should have valid username and password

**Flow of events:**

1. The user opens the system

2. Enters username, password and branch

3. Clicks LOGIN button

4. The system displays appropriate page for the user

5. If the user forgets his/her username/password, then click forgot password/username link.

6. Password recovery page will be displayed

**Exceptional flow:**

if the user enters invalid username and password, the system notifies to enter the correct

one.

**Scenario**: 3

**Name of scenario**: manage material

**Participating actor:** Seller

**Entry condition:**

 Internet connection should be available.

 They have to navigate to SIMS’s URL

 The user should have valid username and password and must login

**Flow of events:**

1. The user opens the system

2. Login to his/her page

3. Click manage material tab

4.chooses View, Add, Delete or update tabs

**Exceptional flow:** if the user enters wrong data, the system displays a message to

enter correct data.

**Scenario**: 4

**Name of scenario**: manage account

**Participating actors:** admin

**Entry condition:**

 Internet connection should be available.

 They have to navigate to SIMS’s URL

 The user should have valid username and password and must login

**Flow event:**

1. Admin open and login to the system.

2. Admin select manage account tab.

3. chooses Create seller account, add account and update account tab.

4. He/she fills for the chosen tab.

5. Click the choose button

6. The system displays successful message

**Exceptional flow:** if the user enters wrong data, the system displays a message to

enter correct data.

**Scenario**: 5

**Name of scenario**: report about daily activities

**Participating actor**: seller

**Entry condition:**

 Internet connection should be available.

 They have to navigate to SIMS’s URL

 The user should have valid username and password and must login

**Flow event**:

1.seller open & log in to the system

2.user page is displayed to the user

3.the user selects daily activity tab

4.choose report, export, import tabs

4.the system display form to fill the chosen tab.

5.the user fills the form

6. click submit

**Exceptional flow:** if the user enters wrong data, the system displays a message to

enter correct data.

**Scenario:** 6

**Name of scenario**: generate borrowing table

**Participating actor**: seller

**Entry condition:**

 Internet connection should be available.

 They have to navigate to SIMS’s URL

 The user should have valid username and password and must login

**Flow event**:

1.seller open & log in to the system

2.user page is displayed to the user

3.the user select borrowing table

4.the system display form to fill borrowing table

5.the user fills the form & click submit

**Exceptional flow:** if the user enters wrong data, the system displays a message to

enter correct data.

**Scenario:** 7

**Name of scenario**: Import and Export

**Participating actor**: seller

**Entry condition:**

 Internet connection should be available.

 They have to navigate to SIMS’s URL

 The user should have valid username and password and must login

**Flow event**:

1.seller open & log in to the system

2.user page is displayed to the user

3.the user select Export and Import tab

4.Chooses either Export or Import tab.

5.the user fills the required material in the form for chosen tab & click submit

**Exceptional flow:** if the user enters wrong data, the system displays a message to

enter correct data.

**Scenario:** 8

**Name of scenario**: Request

**Participating actor**: seller

**Entry condition:**

 Internet connection should be available.

 They have to navigate to SIMS’s URL

 The user should have valid username and password and must login

**Flow event**:

1.seller open & log in to the system

2.user page is displayed to the user

3.the user select request tab

4.the system display form to fill request.

5.the user fills the form & click submit

**Exceptional flow:** if the user enters wrong data, the system displays a message to

enter correct data.

## **3.4.2. Use case model**

1. **actor identification**

* register
* Login

Owner

* View materials
* Manage account
* Change password
* View borrowing table
* View Daily Activity
* View Expired material
* View Profit
* Log out

* Manage materials

Seller

* login
* Change password
* Log out
* borrowing table
* Generate Daily Activity
* Export and Import
* Request
* Logout

1. **Use case identification**

Our system includes the following use cases:

* login
* Manage account
* Add account
* Create Seller account
* Delete account
* Update account
* Manage materials
* Add material
* Delete material
* Update material
* View material
* Generate Daily Activity
* Report sold and purchased material
* Exported materials report
* Imported material report
* Export and Import
* Request
* borrowing table
* record borrowing table
* view borrowing table
* delete borrowing table
* Logout

1. **USE CASE DIAGRAM**

****

Figure 1 use case diagram

1. **USECASE DISCRIPTION**

|  |  |
| --- | --- |
| Use case name | manage account |
| Use case ID | 01 |
| Use case description | To register/delete/approve/view users in to the database |
| Actor | Owner(admin) |
| Pre-condition | owner must be login to the system and connection must be available. |
| Post-condition | System has successfully register/delete/update/ and display account information to the owner. |
| Main flow | a. Admin select create/delete/update/view account tab from his/her page.  b. The system displays create/delete/update / view form.  c. Admin fills account information to create /delete/update/view account.  d. Admin click create/delete/update/search button. |
| Exceptional flow | a. When there is not filled the account information correctly  “please fill account data correctly” message will be displayed.  b. When the admin fills the form incorrect manner, the message “Invalid data Entered please fill in appropriate format” will be displayed |
| Include | Create, Update, delete, view account |
| Business rule | 3 |
| Frequency of use | Depends on the users of the system |
|  |  |
| Use case name | Login |
| Use case ID | 02 |
| Use case description | To authenticate users |
| Actor | All users |
| Pre-condition | Users must have username and password and internet connection is available. |
| Post-condition | Login to the actor page/leave from page. |
| Main flow | a. Actor opens the system  b. The system displays login form.  c. User fills correct username and password.  d. Click login button  e. The system redirect to the user’s homepage. |
| Exceptional flow | When the user fills invalid username and password the system displays incorrect message and notifies the user to enter the correct data. |
| Include | Create account, forgot password |
| Business rule | - |
| Frequency of use | Once on every user. |
|  |  |
| Use case name | Upload materials |
| Use case ID | 03 |
| Use case description | To upload necessary materials, its price purchased, it will buy and expiry date. |
| Actor | Seller |
| Pre-condition | User must log into seller's page |
| Post-condition | Display categories available on users page |
| Main flow | a. Actor open the system  b. The system displays category choice page.  c. User chooses material list tab.  d. Chooses specific material he/she wants to upload for  e. Click upload when finished  f. The system uploads to the stack of information it accumulated before. |
| Exceptional flow | If the user enters incorrect type of information, the system notifies to validate it. |
| Include | Materials list, name, price, expiry date, shelf number. |
| Business rule |  |
| Frequency of use | Sellers wants to update |
|  |  |
| Use case name | Import and Export Material |
| Use case ID | 04 |
| Use case description | Used for seller to exchange the materials among the branch |
| Actor | seller |
| Pre-condition | seller must have username and password. |
| Post-condition | Record import and export to the database. |
| Main flow | a. seller selects import and export tab from his or her page.  b. chooses either export or import tabs  b. The system displays form.  c. seller fill the form with correct data.  d. Click export or import button. |
| Exceptional flow | When he/she fills invalid data in to the form, the system  displays incorrect message and notifies the user to enter  the correct data. |
| Include | View export and import |
| Business rule |  |
| Frequency to use | When it is needed.. |
|  |  |
| Use case name | Generate Report |
| Use case ID | 05 |
| Use case description | Used to generate report for the daily activities. |
| Actor | Seller |
| Pre-condition | Users must be login to the system and the connection must be available. |
| Post-condition | Fetch the report from different tables of the database. |
| Main flow | a. Actor select daily activities tap from his/her page.  b. The system displays report form.  c. The actors fill the data correctly.  d. The actor clicks the report button.  e. The system display report. |
| Exceptional flow | If the actors fill invalid detail to the form, the system displays  incorrect message and notifies the contractor to enter the correct data. |
| Include | User login |
| Business rule |  |
| Frequency to use | Seller Reports daily |
|  |  |
| Use case name | upload borrowing table |
| Use case ID | 06 |
| Use case description | Used to generate borrowing table |
| Actor | Seller |
| Pre-condition | User must log into seller's page |
| Post-condition |  |
| Main flow | Display categories available on users page |
| Exceptional flow | a. Actor open the system  b. The system displays category choice page.  c. User chooses borrowing table tab.  d. upload who is borrowing on the given table.  e. Click upload when finished  f. The system uploads to the stack of information it accumulated before. |
| Include | Borrower’s name ,phone no. |
| Business rule |  |
| Frequency to use | When a person borrows material. |
|  |  |
| Use case name | Logout |
| Use case ID | 07 |
| Use case description | Used to leave from the page. |
| Actor | All users |
| Pre-condition | Actors must have username and password. |
| Post-condition | The actor must be leave from the page. |
| Main flow | a. Actors clicks logout link.  b. The actor leaves the page. |

Table 3.1 use case description

# **3.5 Object Model**

## **3.5.1 Data dictionary**

Object model is a description of an object-oriented architecture, including the details of the object structure, interfaces between objects and other object-oriented features and functions.

|  |  |  |  |
| --- | --- | --- | --- |
| Classes | Attributes | Operations | Description |
| AdminLogin | Username (Phone), password | check (), forgot () | To log the admin into the system |
| ForgotPassword | Username (Phone), password | check (), sendPassword () | To restore forgotten password |
| Notification | Shelfno, expiry date, materialsname | sendNotification () | To notify expired materials |
| Sellerlogin | Username (Phone), password | check (),forgot () | To log the seller into the system |
| Register | Mname, Fname, Lname, age, sex, address, email, username(phone), password | generatePassword (),  generateusername (),  register () | To register Sellers |
| Upload materials | materials type, materials quantity, materials price, Shelf no, material name, materials expiry date | registerMaterial() view material()  view expiry date()  view profit()  deletematerial () | Allow users to add material |
| Seller | FName, MName, LName, address, Age, Sex, phone, email. | registerMaterial (),  deletematerial ()  View materials(),  Give comment(),  Viewcomments(), delete comments(),  Recordborrowing table (), Changepassword(), Viewborrowing table(),delete borrowing table() | To add, view and delete the materials,give comments,record and view borrowing table, |
| Feedback | Time, date, username, comments | Givecomment (),  Viewcomment (),delete comments() | Allow users to give and view comments |
| Account | Fname, Mname, Lname, age,sex,email, ,username, password, location, phone. | Login ()  Create account() | Allow to create  account for user |
| Delete account | Username(phone), | deleteaccount(), | The admin delete accounts of users  from the current data  base. |
| Admin | FName, MName, LName, address, Age, Sex, phone, email | View materials(),  Give comment(),  Viewcomment(),  Deletecomment(),  Changepassword(), Viewborrowing table(), manageaccount(), | To view materials and borrowing table,give comments, manage account. |
| Upload borrowing table | FName, MName, LName, address, Sex, phone, | Viewborrowingtable(),  ,record borrowingtable(),delete borrowing table | To view and record borrowing table |

Table 3.2 data dictionary

## **3.5.2 Class diagram**

The Class diagram captures the logical structure of the system; the classes and things that make

up the model. It is a static model, describing what exists and what attributes and behavior it has,

rather than how something is done. Class diagrams are most useful to illustrate relationships

between classes and interfaces.

****

Figure 2 class diagram

# **3.6. Dynamic model**

## **3.6.1. Sequence diagram**

Sequence diagrams in the UML are primarily used to model the interactions between the actors and the objects in a system and the interactions between the objects themselves. As the name implies, a sequence diagram shows the sequence of interactions that take place during a particular use case or use case instance. The objects and actors involved are listed along the top of the diagram, with a dotted line drawn vertically from these. Interactions between objects are indicated by annotated arrows. The rectangle on the dotted lines indicates the lifeline of the object concerned. The annotations on the arrows indicate the calls to the objects, their parameters, and the return values.

**1-sequential diagram for admin login**



Figure3 sequential diagram for admin login

**2- sequential diagram for seller login**



Figure 4 sequential diagram for seller login

**3- sequential diagram for Register Seller**



Figure 5 sequential diagram for register seller

**4-sequential diagram for Register Material**



Figure 6 sequential diagram for register material

**5-sequential diagram for Register Borrow**



Figure 7 sequential diagram for register borrow

**6-sequential diagram for Generating report**



Figure 8 sequential diagram for generating report

**7-sequential diagram for Export and Import materials**

****

Figure 9 sequential diagram for export and import material

**Activity diagram**

An activity diagram is used to understand the flow of work that an object or component

performs. It can also be used to visualize the interaction between different use cases. One of the

strengths of activity diagrams is the representation of concurrent activities. Some of the activity

diagrams of our system are listed below.

**1.Login**



Figure 10 activity diagram for login

**2. Create account**

 Figure 11 activity diagram for create account

**3.Register seller**

 Figure 12 activity diagram for register seller

**4. Material registration**

 Figure 13 activity diagram for material registration

**5.borrowing table**

**** Figure 14 activity diagram for borrow registration

**6.Generate report**



Figure 15 activity diagram for generate report

**State chart diagram**

It is used to describe the externally visible behavior of a system or of an individual object. Some

of the state chart diagrams of our system are described below.

1. **Login**



Figure 16 state chart diagram for login

1. **Register material**



Figure 17 state chart for generate report

1. **Register seller**



Figure 18 state chart for register seller

# **Chapter Four**

# **4.1. Overview of system design**

This is the system design document of Store inventory management system for shop or super markets. This document describes the design issues of the overall system, such as design goal, subsystem decomposition, hardware/software mapping, and persistent data management. It provides the complete architectural overview of the proposed system.

## **4.1.1. Purpose of the system design**

The purpose of designing is to show the direction how the system is built and to obtain clear and enough information needed to drive the actual implementation of the system. The purpose of this document is to describe the implementation of Store Inventory management system depending on the shop or supermarket requirements.

## **4.1.2 Design goal**

The design goals describe the qualities of the system that are derived from the non-functional requirements which can lead to decisions of developers.

Goal achievement can be measured by the following properties: -

• Reliability • Robustness

• Modifiability • Maintainability

• Understandability • Adaptability

• Reusability • Efficiency

• Portability • Traceability of requirements

• Fault tolerance • Backward-compatibility

• Cost-effectiveness • Low-cost

• Flexibility • Robustness

• High-performance • Good documentation

• Well-defined interfaces • User-friendliness

• Reuse of components • Rapid development

• Minimum number of errors • Readability

• Ease of learning • Ease of remembering

• Ease of use • Increased productivity

Among all of this properties Store inventory management system should achieve some it that is derived from nonfunctional requirements.

**Specially: -**

**Reliability:** The system should be reliable. Appropriate error messages will be provided to users whenever incorrect information is inserted and handle the occurrence of that error.

**Security:** In our system security is not an area of hesitation at all. It is the only choice to be fulfilled in designing this kind of system.

**Robustness:** There are always anomalous and may crashing happen. Our system should be able to recover from all anomalous events.

**Reusability:** Our system cannot be used in another application but it is usable in all reasonably related applications without modification.

**Flexibility:** Our system will have to be replaced entirely if specification changes. And should be easily adaptable to reasonable changes.

**Scalability:** Our system cannot be used as the basis of a larger version. It is an outstanding basis and can be upgraded and updated in the future.

# **4.2. Proposed system architecture**

In this project, the team uses a three-tier architecture, which has three layers. These three layers are the Application or Presentation layer, the business layer and the data access layer.

* Application or presentation layer is the form, which provides the user interface to either programmer or end user.
* The business layer is the class, which the team uses to write the function, which works as a mediator to transfer data from application layer or presentation layer to data layer.
* The data access layer which is only interacts with the database.

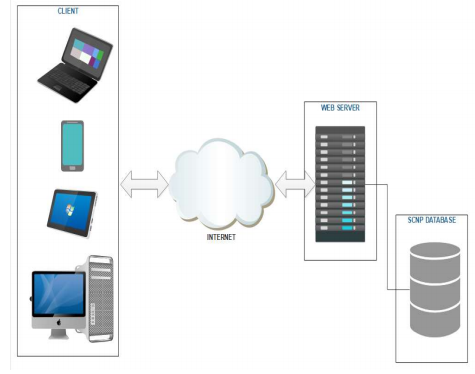
SIMS database

Figure19 system architecture

## **4.2.1. System process**



Figure20 System activity diagram

Admin

This section explores in detail admin system architecture design. A top-level overview of the  
system, a brief overview of system processes or functionality are described in the below  
section.

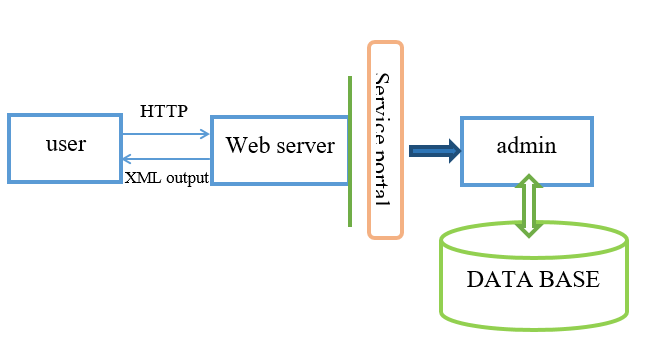


Figure21 admin system overview



Figure 22 admin system process

Seller

This section explores in detail seller system architecture design. A top-level overview of the  
system, a brief overview of system processes or functionality are described in the below  
section.

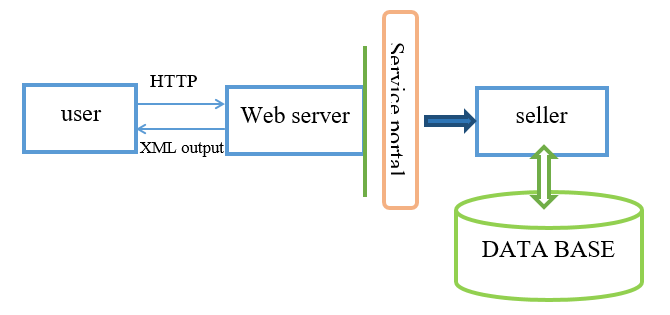


Figure 23 seller system overview



Figure 24 seller system process

## **4.2.2. Subsystem decomposition**

To reduce the complexity of the solution domain, we decompose a system into simpler parts, called subsystems. The subsystems can be considered as packages holding related classes or objects.

SIMS System

Seller

Admin

Figure 25 Subsystem decomposition

|  |  |  |
| --- | --- | --- |
| Subsystem | purpose | class |
| Admin | * Register Seller * Manage account * View material * View profit * Change password * login * logout | * AdminLogin * ForgotPassword * GetPassword * RegisterSeller * Viewmaterial |
| Seller | * Register Material * Register borrow * Importing and Exporting Material * Request Material * Change Material * Change password * Login * Logout | * SellerLogin * ForgotPassword * Registermaterial * Registerborrow * Generatereport |

Table 4.1 Subsystem Decomposition

## **4.2.3 Hardware/Software Mapping**

When we say hardware/software mapping for the system, it describes how subsystems are assigned to hardware and off-the-shelf components. In this system design mainly there are three hardware components. The client side, web/application server and data server. Network should be installed between each side.

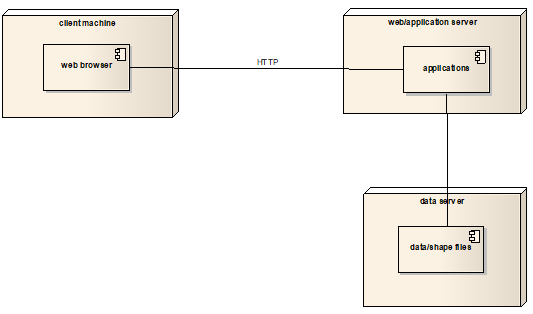


Figure 26 Hardware and software mapping

## **4.2.4. Persistent data management**

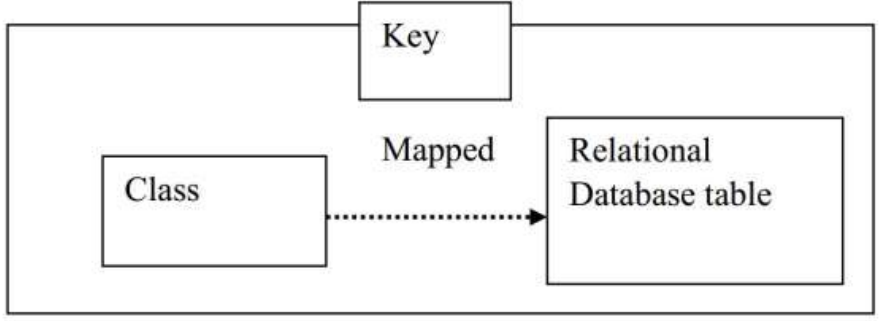
The purpose of this section is to show the mapping of the objects/classes of the system, identified during the analysis stage, in to the corresponding relat

Figure 27 Class to relational database mapping



Figure 28 Register mapping



Figure 29 Borrowing table mapping



Figure 30 Account mapping



Figure 31 Seller mapping



Figure32 Owner mapping



Figure 33 Material mapping



Figure 34 Daily activity mapping



Figure 35 Request mapping



Figure 36 Notification mapping

## **4.2.5. Component diagram**

Component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

The purpose of the component diagram can be summarized as

* Visualize the components of a system.
* Construct executables by using forward and reverse engineering.
* Describe the organization and relationships of the components.



Figure 37 Component diagrams

## **4.2.6. Deployment diagram**

The Deployment model describes how and where the system will be deployed. Deployment diagrams are used to visualize the topology of the physical components of a system,

where the software components are deployed. And used to describe the static deployment view

of a system. Component diagrams and deployment diagrams are closely related. Component

diagrams are used to describe the components and deployment diagrams shows how they are

deployed in hardware.

The purpose of deployment diagrams can be described as:

Visualize the hardware topology of a system.

* Describe the hardware components used to deploy software components.
* Describe the runtime processing nodes.



Figure 38 Deployment Diagram

## **4.2.7. Database design**

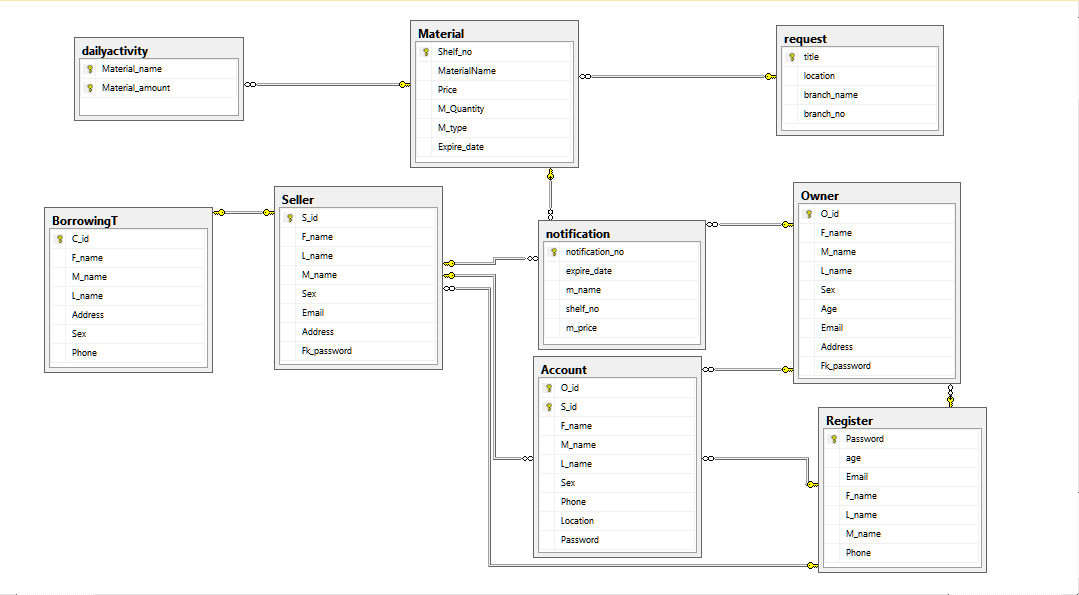


Figure 39 Database design

## **4.2.8 Access control**

In multiuser systems, different actors have access to different functionality and data. We modeled these distinctions by associating different use cases to different actors. This is described by the following access matrix.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Objects Actors | account | materials | Activity | Export and Import | borrowing | Request |
| Admin | Add account ()  Create Seller account ()  Delete account ()  Update account () | View material () | view daily activities () |  | View borrowing table () |  |
| Seller |  | Add material ()  Delete material ()  Update material () | Record sold and purchased material ()  Record Export and import material () | Export and import material() | Record borrowing table ()  Delete borrowing table() | Send request() |

Table 4.2 Access Control

## **4.2.9** **User interface design**

Navigational paths



Figure 40 User interface design

# **Reference**

*[1] Teklebirhan Ambaye Construction Project Management System*

*[2] An Sommerville. software engineering ninth edition*

*[3]* [*www.google.com*](http://www.google.com)

*.*