Store Management Sysetm

Web Application Development Project

Group Members and ID

Wondim Zekarias ID 2120226036
Alemu Ermias ID 2120226046
Anbesse Nardos ID 2120226037
Abay Wondweson ID 2120226041
Lema Fikirte ID 2120226047
.
.

Under the supervision of

Professor : B.H Wang

Date: 7/11/2023

Department of Software Engineering
Nankai Universty

Contents

Abstract

The system aims to streamline store management system inventory control,user registration , issue item, receive item add categories generate comprehensive reporting. The system also ensures data security, and offers a user-friendly interface. By implementing this tailored solution, health centers can optimize inventory management, improve processes, and meet regulatory requirements effectively and also the code is written in python using Django frame work we use MoF (Ministry of Finance)sub domain for deployed the system.

1 Introduction

The Purpose of the project is to create an automated Store Management System for Felege Melese Health Center. While developing this document, the client was interviewed and the important information has been gathered.
The reason behind is to solve and to improve the problem that exist here with current system. Store Management System will cover for a health center or institute would have specific features and functionalities tailored to the requirements of managing medical supplies, equipment, and pharmaceutical inventory and also office equipment. Here are some key components that can be included in a store management system for a health center **Inventory Management**, **Supplier Management Equipment Management** ,**Security and Compliance** and **User Roles and Permissions.**
Our store management system for health centers the software streamlines inventory control, purchase management, and equipment tracking. With secure data handling, seamless integration, and powerful reporting, it enhances operational efficiency and patient care. Experience optimized inventory management with our tailored solution

1.1 Statement of the Problem

The health center currently faces challenges in effectively managing their store inventory, medical supplies, and equipment. The existing manual or disjointed systems make it difficult to track stock levels, monitor expiration dates, and ensure timely replenishment. This leads to inefficiencies, stockouts, increased costs, and potential risks to patient care. There is a need for a comprehensive store management system that can streamline inventory control, enhance purchase management, and improve equipment tracking to overcome these challenges.

- The manual system does't have any warning mechanism for those items or equipment that should be restock.
- Checking the availability of items needed, remembering the location of modules and manual counting is so difficult. In general searching is so boring and time taking.
- Updating number of items when issued to, transferred, return and received are so difficult due to the diversity and large number of items.
- Delay in response time to prepare item status report for managements and employee due to numerous data to be manipulated in order to generate the report.
- Error in posting, balance updating and filing.
- Difficulties to search for bin card, notes when required.
- Due to the above and other reasons report generation is also difficult and not on time.

1.2 Objectives of the Project

The general objective of the project is to develop a computerized or automated item (Equipment) Store Management System that allows keeping and managing the store system of the Health Center in efficient and effective way.

1.3 Scope of the Project

On this project we will be accomplish during the system execution of Felege Melese Health Center of store management. Enable to:

- To recording item information
- Give general and specific report
- Create user account
- User registration
- Searching different kinds of item by them name and type

In general the scope of this project will concentrate on the creation of web based organization wide communication platform for Felege Melese Health center store management. The system will be capable of handling registration login new user account, new login account, user registration, and user categories registration different kinds of document and also include searching using keywords of modules.

1.4 Limitation of the Project

A limitation refers to a constraint or restriction that imposes boundaries or restrictions on the scope, functionality, or performance of a project the following are our project limitation or constraints

- **Integrating**: The store management system with other existing systems, such as Electronic Health Records (EHR)
- It does not support chat services
- Not support audio video
- not support chatbot

2 Methodologies

Methodologies refer to systematic approaches, frameworks, or sets of practices used to guide and structure the process of planning, executing, and managing projects or tasks. They provide a structured and organized way to achieve specific goals or outcomes by defining steps, roles, responsibilities, and deliverables
Among the most widely used methods, we will use the **object-oriented** approach due to the following advantages:

- Increased code re-usability because of inheritance, polymorphism and modularity.
- Enhanced extensibility. In object oriented, classes have both data and functionality hence, when adding new features we will only need to make changes in one place: the applicable class.
- Reduced maintenance burden due to modularized features of object oriented approach.

2.1 Data Gathering Methodology

In order to find relevant information about new system of Flege Melese Health ceter we will use the following data collection methods:

- Interview
- Document Review
- Observation

2.2 Implementation Tools

The implementation of the system will be user friendly and will be using the following tools and approach:-
Software part

- **Programming Language**
  Python (Django)
  Django is a Python web framework that simplifies web application development. It includes features like an ORM for database interaction, URL routing, templating, forms handling, authentication, an admin interface, security measures, internationalization support, and scalability. Django follows the "batteries included" approach, providing many built-in tools and promoting code reusability and maintainability.
- Database
  Databese SQLlite3 : SQLite3 is a lightweight, serverless, and self-contained relational database management system (RDBMS) that operates directly on a disk file. It's commonly used as an embedded database within applications.
- **Operating System**
  Ubuntu
  Ubuntu is a user-friendly and stable Linux-based operating system known for its easy installation, software center, customization options, and strong community support. It is suitable for desktop and server environments, offering security features and efficient package management.

Hardware part

- **PowerEdge R730xd Rack Server**
  The PowerEdge R730xd is a highly scalable rack server from Dell, featuring powerful processors, ample storage options, and extensive memory capacity. It offers flexible networking, management tools, and redundancy features for reliability, making it ideal for virtualization, data analytics, and storage-intensive applications.

2.3 Data Analysis Methodology

To model the user requirements for the proposed system we use UML.

- Use case modeling.
- Sequence diagram.
- Activity diagram.
- Class diagram

2.4 Project Schedule

Task Schedule

Task Name	Roles				
	Start	Finish		Duration	
Requirement Gathering	Jun 04/23	Jun 11/23		7 days	
System Design	Jun 12/23	Jun 17/23		5 days	
Implementation	Jun 18/23	July 6/23		18 days	
Testing	July 7/23	July 14/23		7 days	
Document Submission	July 15/23	July 15 /23		1 day	

3 REQUIREMENT ANALYSIS

Requirement analysis refers to the systematic process of gathering, documenting, and analyzing the needs, expectations, and constraints of stakeholders for a specific project or system. Requirement Analysis for a Store Management System in a Health Center

- Real-time inventory tracking of medical supplies, equipment, and pharmaceuticals.
- Efficient purchase management with supplier integration.
- Expiry date management and alerts for stock rotation.
- Equipment tracking, maintenance scheduling, and calibration reminders.
- Comprehensive reporting and analytics for informed decision-making.
- Integration with existing Electronic Health Records (EHR) system.
- Strong data security and role-based access control.
- User-friendly interface and training support.
- Scalability and flexibility to accommodate future growth and regulatory changes

3.1 CURRENT SYSTEM

Current system of Felege Melese Health Center Store management systems is fully manual. The responsible staffs was expected to generate analysis, processes and reports from the different stores that keeps item and documents like forms which are used as input. item registration, item issuing, item status update, report generation process and other activities are paper based. It is often common that some item recorded documents and histories were lost due to non-integrated way of record keeping. The major activities of the current system are:

3.1.1 Purchase Item

The main warehouse issues a purchase requisition which is sent to the purchaser to order further inventory of item when the current inventories run down to the level where a reorder is required.

3.1.2 Item Order

The purchaser draws up a purchase order which is sent to the supplier. The supplier may be asked to return an acknowledgement copy as confirmation of his acceptance of the order. Copies of the purchase order must be sent to the accounts division and the warehouse clerk.

3.1.3 Item Receive

The purchaser delivers of the consignment of item, and the warehouse clerk signs a delivery note for the carrier. The packages must then be checked against the copy of the purchase order, to ensure that the delivered types and quantities which were ordered. Inconsistency would be referred to the purchasing department. If the delivery is acceptable, the storekeeper prepares a item received note (IRN). A copy of the IRN is sent to the accounts department, where it is matched with the copy of the purchase order.

3.1.4 Item Issue

The Item Issue delivers of the giveing of item to the employee store clerk signs a delivery note for the carrier. The packages must then be checked against the copy of the issued order, to ensure that the delivered types and quantities which were give. Inconsistency would be referred to the store . If the delivery is acceptable, the storekeeper prepares a item received note (IRN). A copy of the IRN is sent to the accounts department, where it is matched with the copy of the issue order.

3.1.5 Generate Report

all activities are need a report by date and item type

3.2 PROPOSED SYSTEM

The proposed system is an automated system having capabilities of handling the electronic based store management of our client's users coordinators. First and for most, all the item registration, updating, deletion and report generation will be electronically captured. Secondly, any item history belonging to the branches will be kept associated to that all modules are stored in a central database of the system. Thirdly compiling reports of different kinds of items will be carried out easily as long as the items are organized in such a way that every items history starting from date of registration to the system. More over the store manager will handle the items and different forms of system users. The proposed system, which is a web based automated system, fulfills the following functional and non-functional requirements and expected to solve the current system's problems.

3.3 FUNCTIONAL REQUIREMENTS

The store management system is responsible to handle the following **major activities** that a system can perform using the different component it has. And the system describes the transaction between the system and its environment independent of the implementation. Therefore, it includes:

- New Received Item Registration
- Issued Items
- Register Item request

- Record, edit search, update and delete a particular Item
- Request Item registration
- Different levels of Report Generation

3.4 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements describe user visible aspects of the system that are not directly related with the functional behavior of the system. It is a description of other features, characteristics, and constraints that defines a satisfactory system. It includes user interface, documentation, performance characteristics, security issues, quality issues, error handling and exception of the system.

3.4.1 User Interface

The user interfaces are to be designed to meet the satisfaction of users putting them at ease while entering and viewing data. All input forms are to be designed to feet screen size. All forms should not be congested with numerous fields to be entered. Both menu and toolbar options are to be phrased properly. Additionally, both keyboard and mouse accesses are to be provided.

3.4.2 Documentation

There should be a user manual that will assist the user as a reference in the course of using the system. For developers who may be interested in the maintenance or upgrading of the system, providing a system documentation manual is important. The system is also supposed to provide online help facility for immediate support of users.

3.4.3 Performance Characteristics

The system should be fast as much as possible in storing, retrieving and accessing any database information from the server; it should allow multiple accesses to the system at the same time from different users concurrently and finally respond to the request within a reasonable period of time.

3.4.4 Security Issues

Security issue is a mechanism for controlling access to the system. In order to implement these requirements the system should a protection mechanism to prevent user's interventions from unauthorized modification to the posted data or having direct access to the database. Different interface will be designed according to user's privilege and different users must have different access rights based on the business rule. User must have proper user name and password in order to access the system. This protects the system from any unauthorized access.

3.4.5 Quality Issues

Quality issues refer the quality of system. The system should be reliable and appropriate. For this reason, the system should try to identify errors that make information in appropriate and notify for user about the error to take proper correction or the system should act accordingly.

3.5 SYSTEM MODEL

System model is a model which represents the interaction between the actors within a specific system. The system modeling also describe the general system through three phases or models:

1. Functional Model: it is used to represent the functionality needed in the eye of the actor using use case diagram.
2. Analysis object model: used to represent the structure of the system using class/object and associated in between using object/class diagram.
3. Dynamic model: it shows the interaction between objects. Represented with sequence and activity diagram.

Hence, we tried to present the Use case modeling, Sequence diagram, class diagram, activity diagram of the proposed system.

3.5.1 USE CASE MODELING

Actors Identification An actor represents an employee that interfaces with the system. The actors that we identified are:

- Stock Manager (request modules to issue)
- Warehouse clerk (Storekeeper)
- Employee (Active)

3.6 Actors Identification and Description

Store Manager is This is a person that has full authentication on all the system control and also requesting to issue item from the main warehouse.
Store keeper This is a person that handles all item transactions and stock activities.
Employee (Active) is a person that he or she request an item by them active account and also issue an item by them active account.

3.7 Use case Identification and Description

Use case is made up of a set of possible sequences of interactions between system and users in a particular environment and related to a particular goal and it proves information about what activities actor perform in the system and what role they are playing. Identifying the information to understand main tasks of users and create a change to identify use cases. The following are the major use cases recognized to make available services of the new system
Use case Identification
A use case is a behavioral related sequence of steps for the purpose of computing a task. The following are the major use cases identified to provide services of the system.

1. Log in
2. Register System Users
3. Register item by them Categorical type

4. Search item Data
5. Register Issue item
6. Update item balance by them catagorical type
7. Register Receive item
8. Register Request item
9. Generate report Receive item
10. Generate report issue item
11. Generate report fixed asset
12. Generate inventory request by a given date
13. User Remover
14. User profile update
15. add a user

The system use case diagram

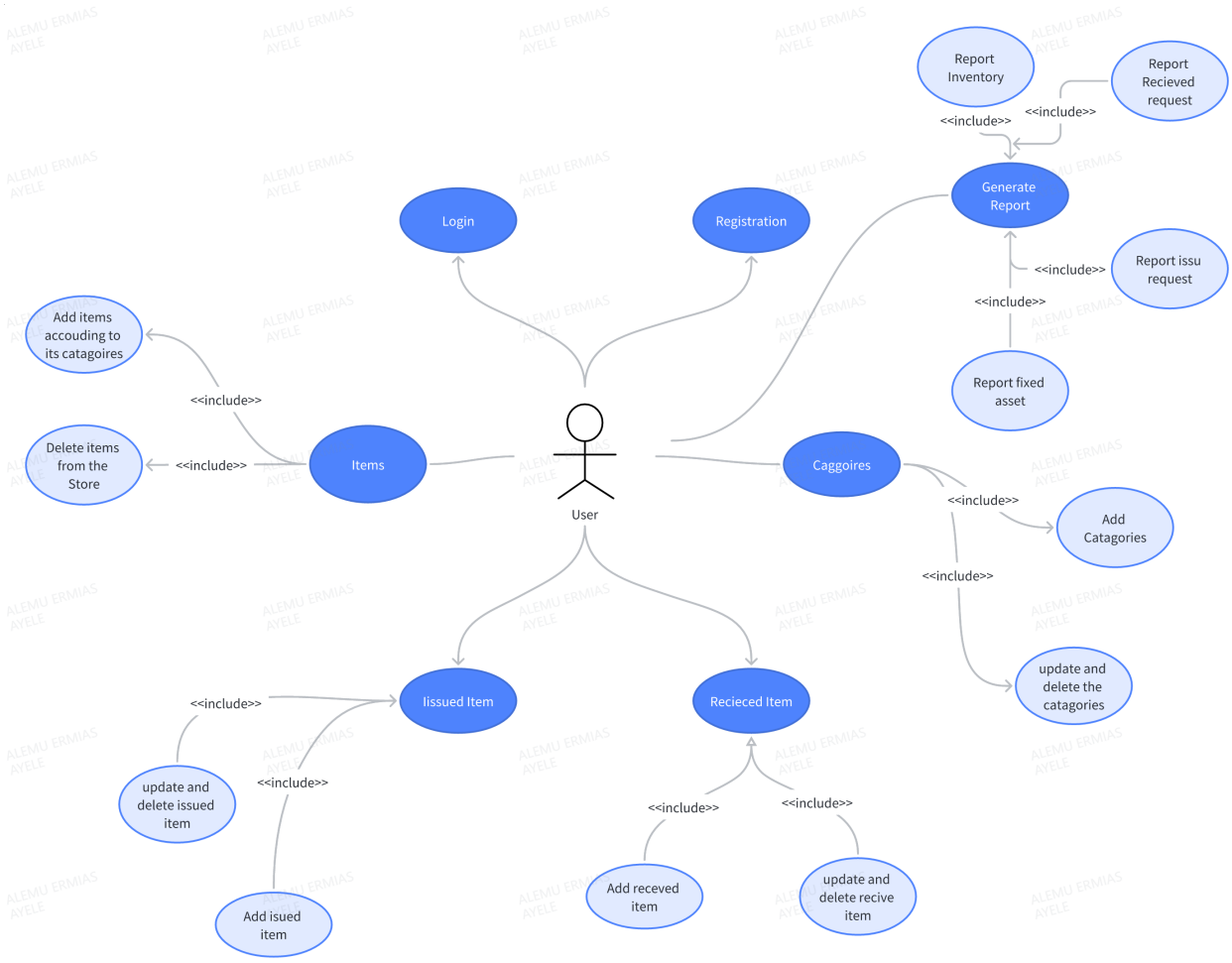


Figure 1: figure show that the use case diagram

3.8 SEQUENCE DIAGRAM

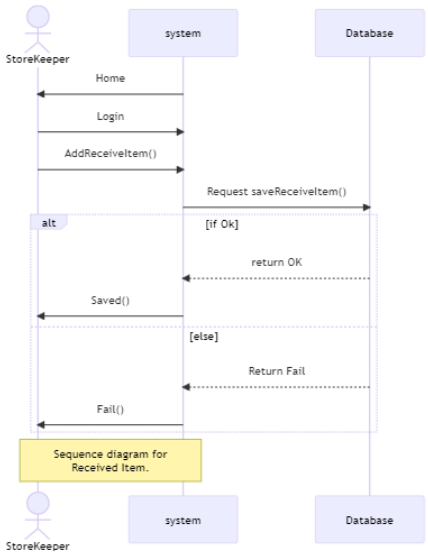
A sequence diagram is a type of interaction diagram in Unified Modeling Language (UML) that depicts the interactions and message flow between different objects or components of a system over time. It shows the sequence of actions and messages exchanged between objects, highlighting the order in which these interactions occur.

In a sequence diagram, objects are represented as vertical lifelines, and the messages exchanged between them are shown as horizontal arrows. The lifelines represent the participating objects or components, and they are arranged in a chronological order from top to bottom. The messages between the objects are labeled with the method or operation name and can include parameters and return values.

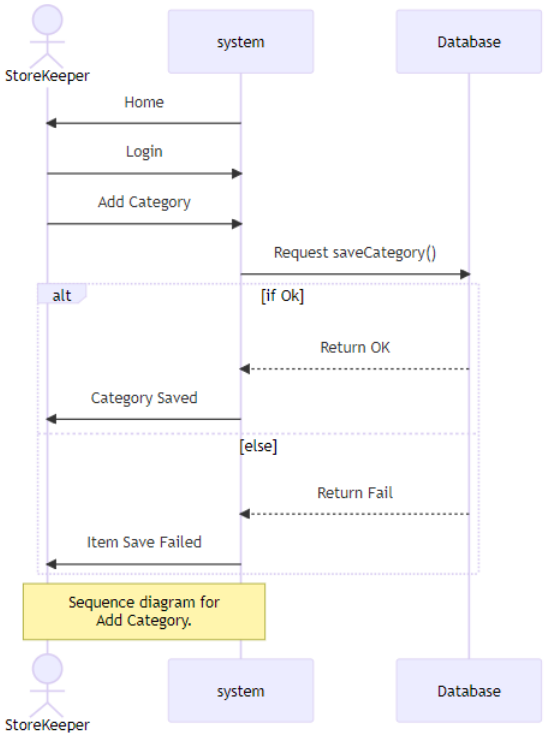
Sequence diagrams are often used to visualize the dynamic behavior of a system, capturing the flow of control and data during the execution of a particular scenario or use case. They help to understand the collaboration between objects, the order of method invocations, and the timing constraints involved in the system interactions. Sequence diagrams are commonly used in software development, particularly during the analysis and design phases, as they provide a clear and concise representation of the system's behavior.

Overall, sequence diagrams provide a valuable tool for modeling and understanding the interactions and message flow within a system, allowing developers and stakeholders to visualize and analyze the behavior and communication patterns of different system components.Here are our Swquence Diagram

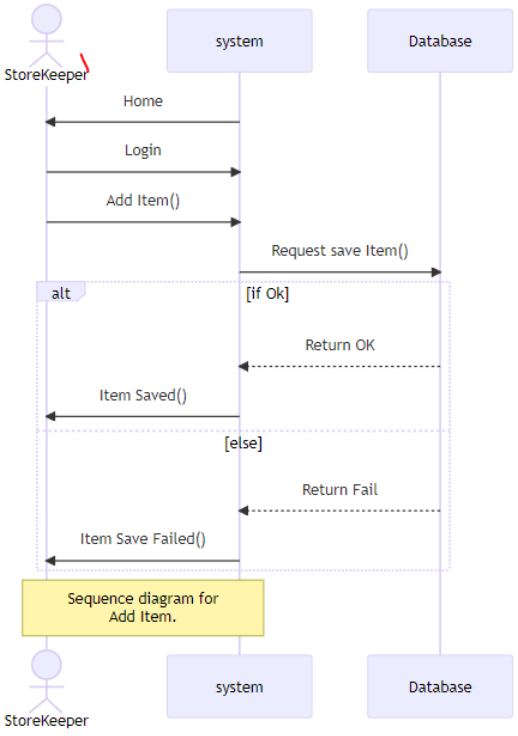
1. Add Catagories
2. Add Items
3. Add users
4. Issue Items
5. Recieve Items



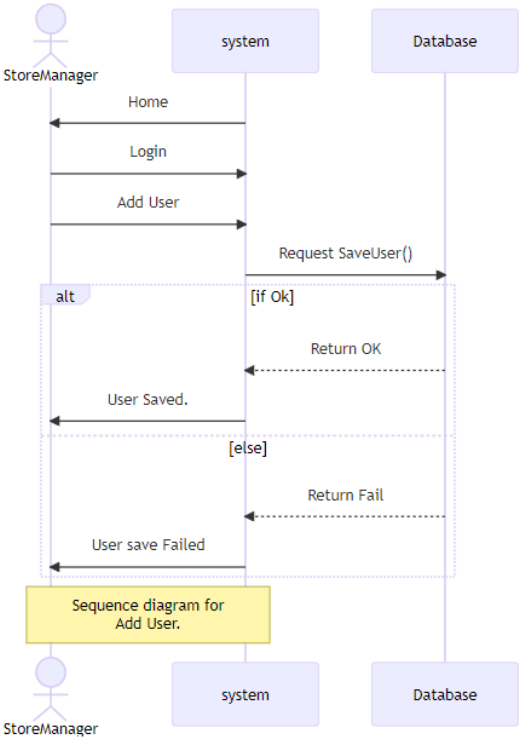
(a) Receiving an item



(b) Add categories

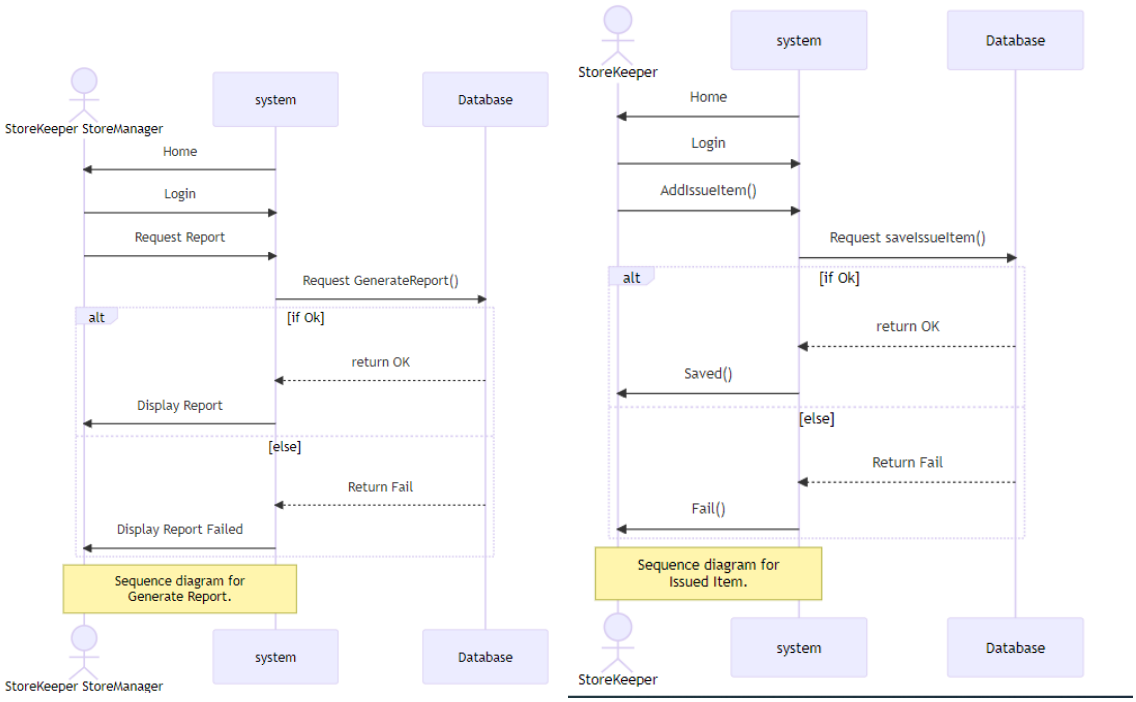


(c) Add an item



(d) Add a user

Figure 2: Sequence Diagrams



(a) Generate a report (b) Issue item

Figure 3: figure show that all Sequence Diagrams

3.9 CLASS DIAGRAM

A class diagram is a type of static structure diagram in UML (Unified Modeling Language) that depicts the structure of a system by showing the classes, their attributes, methods, and the relationships between classes. It provides a high-level view of the system's architecture and the interactions between different components.

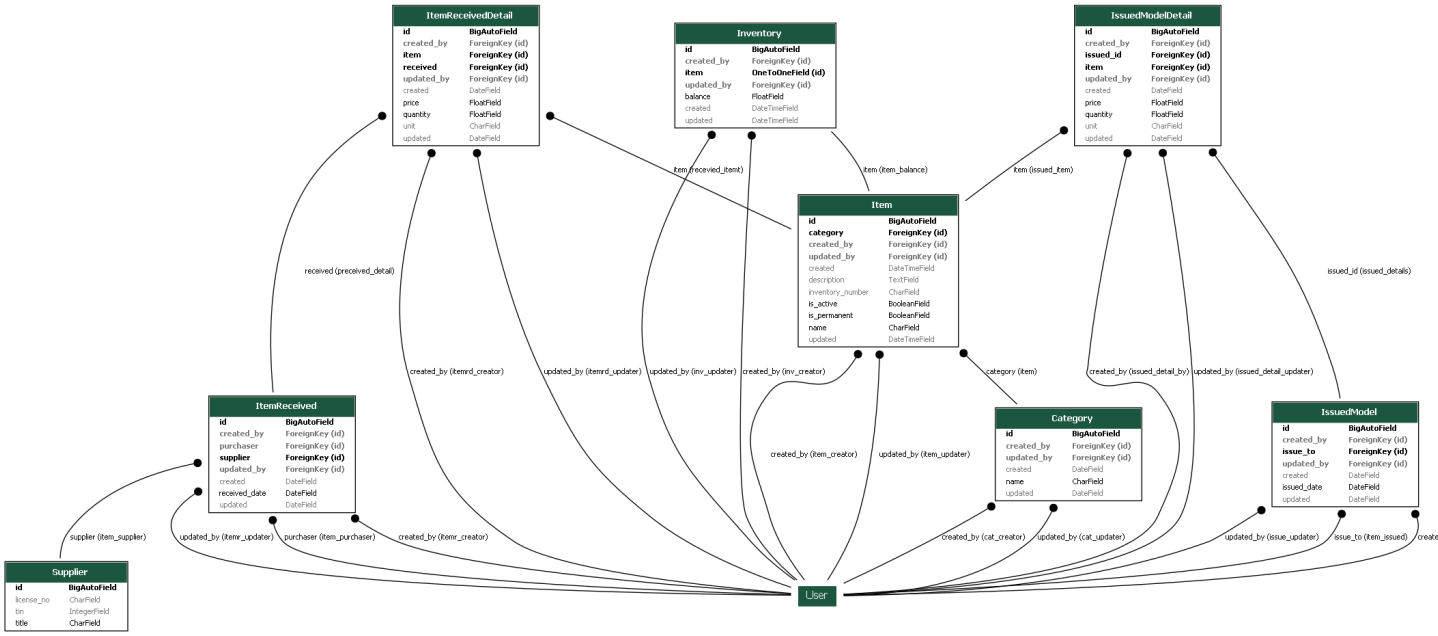


Figure 4: figure show that Class diagram of our project

4 SOFTWARE ARCHITECTURE

The proposed system is a web-based system. And it will be organized on a networked system which is server based. So, a central server is connected to clients. Therefore, in general the proposed software architecture of the system is client/server architecture. This architecture, will allow the client request data from the server and the server which hosts the database, responds accordingly. In the System as described in the

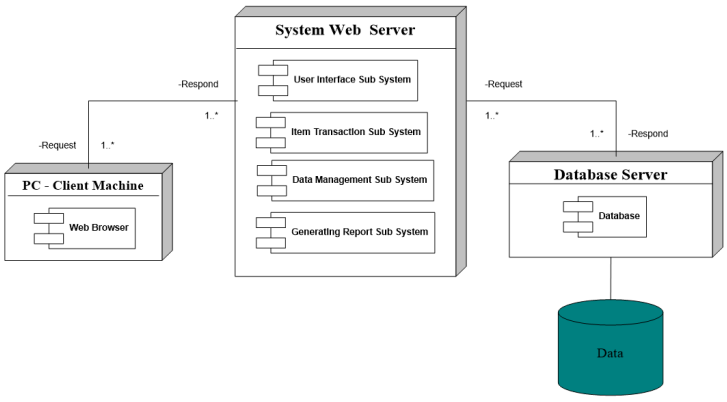


Figure 5: figure show that Deployment Diagram for the System

deployment diagram above will have three main hardware components: the client, the web server, and the database server. The web server will run PHP and will contain all subsystems. The database server will use a MySQL database and will handle all persistent data storage. Lastly, the client will access the website via their web browser.

4.1 System Decomposition

A subsystem is a collection of classes that solves a particular problem. A subsystem is relatively independent in which independency is measured with coupling and coherence. The system will divide hierarchically into set of subsystem by applying layering and partitioning. A subsystem is known by the service it is giving to other subsystem. The envisioned system has the following subsystems:-

1. User Interface subsystem
2. item (Equipment) Transactions subsystem

Items are purchased enterprise to dispatch to employee and the ware house clerk receives to warehouse. Those items are issued to employee from the warehouse.

- Item categories
- add item by them categories
- Receive item
- Issue item
- Generating Report

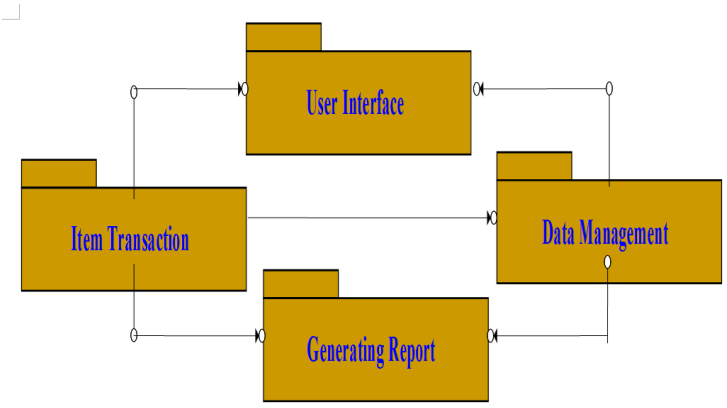


Figure 6: figure show that system decomposition diagram

4.2 Persistent Data Management

The database management system is going to use SQLite3, a relational database management system. SQLite3 is well-suited for complex queries and can handle large datasets. It offers convenient query capabilities and supports establishing relationships between tables.
The system will utilize the database to store modules and module information, as well as retrieve information from it. The data will be stored on a single database server and will not be distributed. To accommodate future data storage requirements, the database should be designed to be extensible.
The database will be stored on the server side and accessed from different client machines belonging to responsible departments and warehouse clerks. It will support concurrent usage, allowing multiple queries to be performed simultaneously.
Given these requirements, the following is a list of classes along with their corresponding tables in the database:

1. Supplier table
2. Category table
3. Item table
4. Inventory table
5. ItemReceived table
6. ItemReceivedDetail table
7. IssuedModel table
8. IssuedModelDetail table

4.3 Access Control and Security

Security is one of the primary concerns of when using computers. Security must therefore consider external environment of the system, and protect it from:

- Unauthorized access
- Malicious modification or destruction
- Accidental introduction of inconsistency

User authentication methods are used in protecting special users of documents meant to be accessed only by individual employee with the proper clearance as determined by an appropriate set of identification credentials. The data access and control mechanism is going to be functioned by providing privilege for concerned employee though authentication of their user name and password.

4.4 Access Rights for users

Access Rights for users

Operation	Roles				
	Store Manager	Store Keeper		Employee(Active)	
Login	✓	✓			
Register User	✓				
Add Items	✓	✓			
Search Item	✓	✓			
Add Issue Item	✓	✓			
Add Receive Items	✓	✓			
Update Stock Items	✓				
Delete Stock Items	✓				
Add Categories	✓	✓			
General Report	✓	✓			

4.5 Global Resource Handling

Django, web framework written in Python, provides robust support for global resource handling. In the context of Django, global resource handling refers to the mechanisms and practices employed to efficiently manage and serve static files, media files, and other resources required by a web application.

Django offers built-in features and conventions that streamline the process of handling global resources. One key component is the static files handling system. With Django's static files handling, you can define a designated directory to store your application's static files, such as CSS stylesheets, JavaScript files, and images. Django automatically takes care of serving these files efficiently, whether during development or in a production environment.

Django also provides a powerful mechanism for managing media files, such as user-uploaded images or documents. By configuring a media storage backend, you can easily handle the storage and retrieval of these files. Django's media handling capabilities include the ability to define upload paths, control file naming, and implement various storage options, such as local storage or integration with cloud-based storage services.

Furthermore, Django offers an optional third-party package called "django.contrib.staticfiles" that provides additional features and optimizations for handling static files. This package includes functionalities like automatic collection of static files from different applications, cache-busting techniques, and integration with popular front-end tools like JavaScript package managers or CSS preprocessors.

Overall, Django's global resource handling capabilities help developers efficiently manage and serve static files, media files, and other resources in their web applications. Whether it's organizing and serving static assets or handling user-uploaded files, Django provides a solid foundation for managing global resources effectively.

4.6 CLASS INTERFACES

Describe the class and their public interfaces. This includes an overview of each class, its dependencies with other classes Descriptions of the class interface and tables for attributes and the operations of the class are presented in the following detail of class article tabularx adjustbox

Table 1: Data Model for Supplier,Category and Item

Model//Class	**Field**	**Description**
Supplier	id	Primary key for the supplier
	tin	Tin number of the supplier
	title	Name of the supplier
	license_no	License number of the supplier (optional)
Category	id	Primary key for the category
	name	Name of the category
	created	Date when the category was created
	updated	Date when the category was last updated
	created_by	User who created the category
	updated_by	User who last updated the category
Item	id	Primary key for the item
	category	Category to which the item belongs
	name	Name of the item
	description	Description of the item (optional)
	is_active	Indicates if the item is active
	is_permanent	Indicates if the item is permanent or for consumers
	inventory_number	Inventory number of the item (optional)
	created_by	User who created the item
	updated_by	User who last updated the item
	created	Date and time when the item was created
	updated	Date and time when the item was last updated

Table 2: Data Model for Inventory and ItemReceived

Model/Class	**Field**	**Description**
Inventory	id	Primary key for the inventory
	item	Item associated with the inventory balance
	balance	Quantity balance of the item
	created_by	User who created the inventory
	updated_by	User who last updated the inventory
	created	Date and time when the inventory was created
	updated	Date and time when the inventory was last updated
ItemReceived	id	Primary key for the received item
	supplier	Supplier from whom the items were received
	received_date	Date when the items were received
	purchaser	User who made the purchase
	created	Date when the received item was created
	updated	Date when the received item was last updated
	created_by	User who created the received item
	updated_by	User who last updated the received item

Table 3: Data Model for ItemReceivedDetail ,IssuedMode and IssuedModelDetail

Model/Class	Field	Description
ItemReceivedDetail	id	Primary key for the received item detail
	received	Received item to which the detail belongs
	item	Item received in the detail
	quantity	Quantity of the item received
	unit	Unit of measurement for the item (optional)
	price	Unit price of the item (optional)
	created	Date when the detail was created
	updated	Date when the detail was last updated
	created_by	User who created the detail
	updated_by	User who last updated the detail
IssuedModel	id	Primary key for the issued item
	issue_to	User to whom the items were issued
	issued_date	Date when the items were issued
	created	Date when the issued item was created
	updated	Date when the issued item was last updated
	created_by	User who created the issued item
	updated_by	User who last updated the issued item
IssuedModelDetail	id	Primary key for the issued item detail
	issued_id	Issued item to which the detail belongs
	item	Item issued in the detail
	quantity	Quantity of the item issued
	unit	Unit of measurement for the item (optional)
	price	Unit price of the item (optional)
	created	Date when the detail was created
	updated	Date when the detail was last updated
	created_by	User who created the detail
	updated_by	User who last updated the detail

4.7 FUTURE WORK

There are several potential avenues for future work in the development and enhancement of a store management system for health centers:

- Integration with IoT Devices: Explore the integration of Internet of Things (IoT) devices to enable real-time monitoring of inventory levels, automated reordering, and equipment status tracking. This can further streamline inventory management and improve overall operational efficiency.
- Predictive Analytics: Implement predictive analytics algorithms to forecast demand patterns and optimize inventory levels accordingly. This can help prevent stockouts and reduce wastage while ensuring the availability of critical supplies.
- Mobile Application: Develop a mobile application to allow staff members to access the store management system on the go. This can enhance convenience and flexibility in inventory management tasks, such as inventory counts and placing purchase orders.
- Machine Learning for Supplier Performance: Utilize machine learning algorithms to analyze supplier performance data and identify trends or patterns that can help in supplier selection and negotiation for better pricing, reliability, and service.
- Enhanced Reporting and Business Intelligence: Expand reporting and business intelligence capabilities to provide more advanced analytics, visualizations, and decision support tools. This can enable health

centers to gain deeper insights into inventory trends, cost optimization opportunities, and operational performance.

- Integration with Telemedicine Platforms: Explore integration with telemedicine platforms to facilitate seamless communication between healthcare providers and the store management system. This can help in efficient medication ordering, prescription management, and remote patient monitoring.
- Enhanced Security Measures: Continuously enhance security measures to protect sensitive patient information and maintain compliance with evolving data protection regulations. This may include encryption, multi-factor authentication, and regular security audits.
- Collaboration with Other Health Centers: Foster collaboration and knowledge sharing among health centers to share best practices and lessons learned in store management. This can lead to collective improvements and standardization in inventory management processes.
- Depreciation Tracking: The system should have a dedicated module or feature to track and manage depreciation of assets. This module should allow users to input asset details such as initial cost, useful life, and salvage value.
- Automatic Calculation: The system should automatically calculate depreciation based on the entered asset details and the selected depreciation method. It should generate depreciation schedules or calculations for each asset, considering the chosen depreciation method (e.g., straight-line or reducing balance).
- Asset Register: The system should maintain an asset register or inventory that includes all assets owned by the health center. This register should include relevant details such as asset description, purchase date, initial cost, accumulated depreciation, and current book value.

By pursuing these future work areas, the store management system can evolve into a more sophisticated and robust solution that meets the evolving needs of health centers, improves operational efficiency, and ultimately enhances patient care.

4.8 CONCLUSION

In conclusion, implementing a store management system in a health center is essential to address the challenges of inventory control, purchase management, the health center can optimize inventory levels, improve efficiency, and enhance patient care. With comprehensive reporting, data security, and a user-friendly interface, the store management system provides the necessary tools to streamline operations and adapt to future needs. By investing in a tailored solution, the health center can achieve better inventory management and ensure the availability of critical supplies and equipment when needed most.