



National University of Computer and Emerging Sciences



MedPy ERP

Yameen Javaid.....16L-4344
Muhammad Shehriyar.....17L-4358
Hamza Riaz.....18L-1004
Ghasif Javed.....18L-2181

Supervisor: Ms. Sana Fatima

B.S. Computer Science
Final Year Project
Month Year

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the:

Title: MedPy ERP

is the sole contribution of the author(s) and no part hereof has been reproduced on **as it is** basis (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: 1st June 2022

Student 1

Name: Yameen Javaid

Signature: _____

Student 2

Name: Muhammad Shehriyar

Signature: _____

Student 3

Name: Hamza Riaz

Signature: _____

Student 4

Name: Ghasif Javed

Signature: _____

Table of Contents

Table of Contents	i
List of Tables	vi
List of Figures	vii
Abstract	9
Chapter 1: Introduction	10
1.1 Goals and Objectives	10
1.2 Scope of the Project	10
Chapter 2: Literature Survey / Related Work	11
2.1 Oracle ERP.....	11
2.2 SAP ERP	11
2.3 Microsoft Dynamics.....	11
2.4 Odoo.....	11
Chapter 3: Requirements and Design	13
3.1 Functional Requirements	13
3.1.1 General Functional Requirements.....	13
3.1.2 Functional Requirements for Admin	13
3.1.3 Functional Requirements for Production Officer	14
3.1.4 Functional Requirements for Accounts Officer.....	15
3.1.5 Functional Requirement for Sales Officer	16
3.1.6 Functional Requirements for Purchase Officer.....	16
3.2 Non-Functional Requirements	17
3.2.1 Performance	17
3.2.2 Usability	18
3.2.3 Sustainability.....	18
3.2.4 Reliability.....	18
3.2.5 Flexibility.....	18
3.3 Hardware and Software Requirements	18
3.3.1 Hardware Requirements.....	18
3.3.2 Software Requirements.....	18
3.4 System Architecture.....	19
3.4.1 Admin Component.....	19
3.4.2 Search Component	20
3.4.3 Account Component	20
3.4.4 Attendance Component.....	21
3.4.5 Authentication Component	21
3.4.6 Customer Component	22
3.4.7 Dashboard Component.....	22
3.4.8 Raw Material Component	23
3.4.9 Product Component	23
3.4.10 Production Component	24
3.4.11 Purchase Component	24
3.4.12 Sale Component	25
3.4.13 Staff Component	25
3.4.14 Supplier Component	25
3.5 Architectural Strategies.....	26
3.5.1 Use of a Particular Product	26
3.5.2 Reuse of Existing Software Components	26
3.5.3 Future Plans	26
3.5.4 Memory Management.....	26

3.5.5 Centralized Data over Distributed	26
3.5.6 User Interface Paradigms	26
3.6 Use Cases	27
3.6.1 Login	27
3.6.2 Manage Password	28
3.6.3 Logout	28
3.6.4 View Staff	29
3.6.5 Add Staff	30
3.6.6 Remove Staff	31
3.6.7 Edit Staff	31
3.6.8 Search Staff	32
3.6.9 View Attendance	32
3.6.10 Add Attendance	33
3.6.11 Remove Attendance	33
3.6.12 Edit Attendance	34
3.6.13 Search Attendance	35
3.6.14 Generate Reports	36
3.6.15 View To-Do List	36
3.6.16 Add Task	37
3.6.17 Remove Task	37
3.6.18 Edit Task	38
3.6.19 Search Task	38
3.6.20 Start Production	39
3.6.21 Remove Production	40
3.6.22 View COA	40
3.6.23 Add COA	41
3.6.24 Remove COA	42
3.6.25 Edit COA	42
3.6.26 Search COA	43
3.6.27 View Voucher	43
3.6.28 Add Voucher	44
3.6.29 Remove Voucher	45
3.6.30 Edit Voucher	45
3.6.31 Search Voucher	46
3.6.32 View Supplier	46
3.6.33 Add Supplier	47
3.6.34 Remove Supplier	48
3.6.35 Edit Supplier	48
3.6.36 Search Supplier	49
3.6.37 View Purchase	49
3.6.38 Add Purchase	50
3.6.39 Remove Purchase	51
3.6.40 Edit Purchase	51
3.6.41 Search Purchase	52
3.6.42 View Raw Material	52
3.6.43 Add Raw Material	53
3.6.44 Remove Raw Material	54
3.6.45 Search Raw Material	54
3.6.46 Add Packing Material	55
3.6.47 Remove Packing Material	56

3.6.48 Edit Packing Material	56
3.6.49 Add Sale.....	57
3.6.50 Remove Sale	58
3.6.51 Edit Sale	58
3.6.52 Add Product	59
3.6.53 Remove Product.....	60
3.6.54 Edit Product	60
3.6.55 Add Customer	61
3.6.56 Remove Customer.....	62
3.6.57 Edit Customer	62
3.7 GUI	63
3.8 Database Design.....	72
3.8.1 ER Diagram	72
3.9 System Requirements.....	72
3.9.1 Hardware Requirements.....	72
3.9.2 Software Requirements.....	72
3.10 Design Considerations	72
3.10.1 Assumptions and Dependencies	72
3.10.2 General Constraints.....	73
3.11 Development Methods	74
3.12 Class diagram.....	74
3.13 Sequence diagram	75
3.13.1 Add Staff.....	75
3.13.2 Add Supplier	76
3.13.3 Add Sale.....	77
3.13.4 Add Purchase	78
3.13.5 Add COA	79
3.13.6 Add Packing Material	80
3.13.7 Add Payroll	81
3.13.8 Add Attendance	82
3.13.9 Add Customer	83
3.13.10 Add Product	84
3.13.11 Add Production	85
3.13.12 Add To-Do List.....	86
3.13.13 Edit Purchase	87
3.13.14 Edit Attendance.....	88
3.13.15 Edit Customer	89
3.13.16 Edit Product	90
3.13.17 Edit Sale	91
3.13.18 Edit Staff.....	92
3.13.19 Edit Supplier	93
3.13.20 View COA	94
3.13.21 View Payroll	95
3.13.22 View Voucher	96
3.13.23 View Attendance.....	96
3.13.24 View Staff.....	97
3.13.25 View To-Do List	98
3.13.26 View Production	98
3.13.27 View Purchase	99
3.13.28 View Supplier	100

3.13.29 View Customer	100
3.13.30 View Sales	101
3.13.31 Delete Staff	102
3.13.32 Delete Attendance	103
3.13.33 Delete Production.....	104
3.13.34 Delete Customer.....	105
3.13.35 Search COA	106
3.13.36 Search Staff.....	107
3.13.37 Search Production	108
3.13.38 Search Purchase	109
3.13.39 Search Customer	110
3.14 Policies and Tactics.....	110
3.14.1 Standards and Guidelines.....	110
3.14.2 Data Collection	110
3.14.3 Protocol.....	110
3.14.4 Algorithm(s).....	111
3.14.5 Testing.....	111
Chapter 4: Implementation and Test Cases	112
4.1 Implementation	112
4.1.1 Prototype Description	112
4.1.2 Authentication.....	112
4.1.3 Managing Staff.....	112
4.1.4 Add and Modify Staff	112
4.1.5 Database.....	113
4.2 Test Case Design and Description	113
4.2.1 Login.....	113
4.2.2 View Dashboard.....	114
4.2.3 View Products List.....	115
4.2.4 View Purchases List.....	116
4.2.5 View Suppliers List.....	117
4.2.6 View Customer List	118
4.2.7 View Tasks List	119
4.2.8 View Production List	120
4.2.9 View Staff List.....	121
4.2.10 View Sales List	121
4.2.11 View Accounts List.....	122
4.2.12 Edit Product	123
4.2.13 Edit Purchases	124
4.2.14 Edit Supplier	125
4.2.15 Edit Customer	126
4.2.16 Edit Tasks.....	127
4.2.17 Edit Production	128
4.2.18 Edit Staff.....	129
4.2.19 Edit Sales	130
4.2.20 Edit Accounts.....	131
4.2.21 Add a new Purchase.....	132
4.2.22 Add a new Product.....	133
4.2.23 Add a new Supplier.....	134
4.2.24 Add a new Customer.....	135
4.2.25 Add a new Tasks.....	136

4.2.26 Add a new Production.....	137
4.2.27 Add a new Staff	138
4.2.28 Add a new Sales.....	139
4.2.29 Add a new Account.....	140
4.2.30 Delete a In Process Purchase	141
4.2.31 Delete a Product.....	142
4.2.32 Delete a Supplier.....	143
4.2.33 Delete a Customer.....	144
4.2.34 Delete a Task.....	145
4.2.35 Cancel a Production	146
4.2.36 Delete an Account.....	147
4.2.37 Delete Sale Order.....	148
4.2.38 Search a Record	149
4.2.39 Generate Report	150
4.3 Test Metrics	150
Chapter 5: Experimental Results and Analysis.....	151
Chapter 6: Conclusion.....	151
References.....	152

List of Tables

Table 2-1: Comparison between different ERPs	12
--	----

List of Figures

Fig 3-1: General System Architecture of Web Application.....	19
Fig 3-2: Admin Component	19
Fig 3-3: Search Component	20
Fig 3-4: Account Component.....	20
Fig 3-5: Attendance Component	21
Fig 3-6: Authentication Component	21
Fig 3-7: Customer Component.....	22
Fig 3-8: Dashboard Component.....	22
Fig 3-9: Raw Material Component	23
Fig 3-10: Product Component.....	23
Fig 3-11: Production Component.....	24
Fig 3-12: Purchase Component.....	24
Fig 3-13: Sale Component	25
Fig 3-14: Staff Component	25
Fig 3-15: Supplier Component.....	25
Fig 3-16: Admin Signup	63
Fig 3-17: Admin Login	63
Fig 3-18: View Attendance.....	64
Fig 3-19: Add User	64
Fig 3-20: Chart of Accounts	65
Fig 3-21: Remove User.....	65
Fig 3-22: View Sales.....	66
Fig 3-23: View Purchase.....	66
Fig 3-24: View Production.....	67
Fig 3-25: View Tasks.....	67
Fig 3-26: Assign Task.....	68
Fig 3-27: Add Voucher	68
Fig 3-28: Add Order.....	69
Fig 3-29: Add COA	69
Fig 3-30: Add Customer	70
Fig 3-31: Add Sale	70
Fig 3-32: Generate Reports.....	71
Fig 3-33: ER Diagram.....	72
Fig 3-34: UML Class Diagram	74
Fig 3-35: Add Staff	75
Fig 3-36: Add Supplier	76
Fig 3-37: Add Sales	77
Fig 3-38: Add Purchase	78
Fig 3-39: Add COA	79
Fig 3-40: Add Purchase	80
Fig 3-41: Add Payroll	81
Fig 3-42: Add Attendance.....	82
Fig 3-43: Add Customer	83
Fig 3-44: Add Product	84
Fig 3-45: Add Production	85
Fig 3-46: Add To-Do List.....	86
Fig 3-47: Edit Purchase.....	87
Fig 3-48: Edit Attendance.....	88
Fig 3-49: Edit Customer.....	89

Fig 3-50: Edit Product.....	90
Fig 3-51: Edit Sale	91
Fig 3-52: Edit Staff	92
Fig 3-53: Edit Supplier.....	93
Fig 3-54: View COA.....	94
Fig 3-55: View Payroll.....	95
Fig 3-56: View Voucher	96
Fig 3-57: View Attendance.....	96
Fig 3-58: View Staff	97
Fig 3-59: View To-Do List	98
Fig 3-60: View Production.....	98
Fig 3-61: View Purchase.....	99
Fig 3-62: View Supplier.....	100
Fig 3-63: View Customer.....	100
Fig 3-64 : View Sales.....	101
Fig 3-65: Delete Staff.....	102
Fig 3-66: Delete Attendance	103
Fig 3-67: Delete Production.....	104
Fig 3-68: Delete Customer.....	105
Fig 3-69: Search COA	106
Fig 3-70: Search Staff	107
Fig 3-71: Search Production	108
Fig 3-72: Search Purchase	109
Fig 3-73: Search Customer	110

Abstract

The purpose of this project is to provide the target company with a web application that provides them with the ease of managing their day to day operations and make the company's working more efficient. Our project, MedPy ERP is an Enterprise Resource Planning System that helps the target company to collaborate within its different departments and that way all of them can work together on a single platform. Moreover, the admin can manage all the employees, view different reports as provided by the other departments and along with that we provide an easy to use interface to do all this. The other managers like the production manager can provide the entire working of his department to the admin as well as add, edit or remove productions. Similarly, the accounts manager can manage salaries using the information provided by the admin as well as create different accounting reports. Furthermore, the other departments like sales and purchase can also work in harmony and keep a record of all the ongoing transactions easily in our database. For our backend we used the python Django framework which provides a default admin menu so that we can view all the backend data as well. We used ReactJS for the implementation of our frontend whereas we used the cloud based MongoDB Atlas for keeping record of our data. We studied different research papers to understand how the ERPs work and how they can boost a company's business by having efficient system to automate their day to day operations. For the scope of this project, we used an example of a pharmaceutical manufacturing company in which we can produce a drug while keeping the sales, purchase, material used, cost calculations automatic. If we want to work in a much larger scope or for a different type of product, our ERP is efficient enough to be used there as well. After conducting the thorough research we came up with the functional and non-functional requirement to develop our product. After completing the development of our project we conducted testing on the app using multiple test cases and Jira as well as manual testing. In short, we will be providing our client with a single platform to perform their daily operations that are entirely automated and more efficient than the manual system.

Chapter 1: Introduction

MedPy ERP is an Enterprise Resources Planning System for a pharmaceutical organization. An ERP refers to a type of software that organizations use to manage day-to-day business-activities such as accounting, procurement, project management, risk management and, compliance, and supply chain operations. Our ERP will include features like enterprise performance management, and software that helps plan, budget, predict, and report on an organization's financial results. We will be providing a web application to provide all the features of an ERP for a use case of a pharmaceutical company. So basically, the application will provide all the ERP features a pharmaceutical organization might need.

This report will cover the Requirement and Design of our project including Functional and non-Functional requirements, Hardware and Software requirements, the system architecture, use cases, a sample GUI, the database design, class diagram and the sequence diagrams. Moreover, we will be discussing the implementation and testing of our project. In conclusion, we will address the future work and how our project will progress.

1.1 Goals and Objectives

We aim on achieving the following goals:

- Improve Service Experience
- Enhance Competitiveness amongst Pharmaceuticals
- Modernize Medical Processes and Systems
- Automate Medical Solutions
- Increase Operating Efficiency
- Provide Access to Standardized Medical Data
- Replace inefficient systems with one that is functionally integrated
- Provide a state-of-the-art technical infrastructure to better serve pharmaceutical staff, other members of the medical community
- Empower users with more control over system function, service, and scheduling
- Utilize a completely web-based system
- Provide easy access to data and information without compromising security and regulatory requirements

1.2 Scope of the Project

The motive of our project is business process acceleration, to gain the competitive advantage, and integration of financial information and its management. Our ERP system will provide the following:

- Automate the processes and handle administration complexity by centralizing the administration processes across programs
- Efficiency improvements in administration

- Provide shared access to data for decision support
- **Finance** – Financial and asset management, enterprise and budget control
- **Logistics** – Production, project, and material management etc.
- **Human resource** – Training, development and personnel management, etc.
- **Supply Chain** – Sale and purchase control, planning, and inventory control, etc.
- **Work-flow** – Flexible allocation of tasks and responsibilities to positions and jobs etc.

Chapter 2: Literature Survey / Related Work

Over the past few years, the use of technology is uplifted due to different factors. In past, manual entry systems and the manual calculations and record keeping was used. Then the record keeping became much easier with the excel sheets but all the information was to be entered all by the human effort and the formulas were to be applied but then there was an evolution of the ERP systems to manage the workflow of the organizational unit. All the transactions and computations are automated using the ERP systems. The Finance management and the production unit is also very effective part of the ERP systems. But still there are some vulnerabilities such as widespread data breaches, theft, and loss.

2.1 Oracle ERP

In 2012, the Oracle corporation introduced an ERP software called Oracle Cloud Enterprise Resource Planning. It manages the functionalities like accounting, financial management, project management and procurement.

2.2 SAP ERP

SAP ERP was developed by a German company named as SAP SE. It incorporates the business functions of an organizations. It has operational, financial, Human Capital Management and corporate Services. It was built on base of the former SAP R/3 software.

2.3 Microsoft Dynamics

Microsoft Dynamics is a line of enterprise resource planning (ERP) and customer relationship management (CRM) software applications. Microsoft Dynamics forms part of "Microsoft Business Solutions". Dynamics can be used with other Microsoft programs and services, such as SharePoint, Yammer, Office 365, Azure and Outlook. The Microsoft Dynamics focus-industries are retail, services, manufacturing, financial services, and the public sector. Microsoft Dynamics offers services for small, medium, and large businesses.

2.4 Odoo

The Odoo is a suite of business management software. The source code for the framework and core ERP modules is curated by the Belgium-based Odoo S.A. Odoo is available for both on-premise and ready to use SaaS environment. The source code for the Open Object framework and the core ERP modules are provided by Belgium-based Odoo S.A. managed.

	Hosted/Self-Hosted	Latest Version in Market	Organizational Size	Extended Functionality with 3 rd party apps
SAP ERP	Hosted	V.6.0	All	Yes
Odoo	Hosted	V.15.0	All	Yes
Microsoft Dynamics	Hybrid	Dynamics 365	All	Yes
Oracle ERP	Self-Hosted	R12. 2.11	All	Yes

Table 2-1: Comparison between different ERPs

This table shows the comparisons of the different ERPs that are available in the market

Chapter 3: Requirements and Design

3.1 Functional Requirements

This section includes all the functional requirements that must be implemented for the development of MedPy ERP. The functional requirements can be divided among different roles that are Admin, Production Officers, Accounts Officers, Sales Officers, Purchase Officers. These functions will be the functionalities of the MedPy ERP system.

3.1.1 General Functional Requirements

3.1.1.1 Login and Logout

- i. User shall provide the Email ID in order to log into the system.
- ii. Combination of Email and Password will be used to Authentication.
- iii. User shall be able to Logout of its Profile.

3.1.1.2 Manage Password

- i. If user forgets his password, he can request to change the password.
- ii. User shall be able to change password at any instance of time.

3.1.2 Functional Requirements for Admin

3.1.2.1 Add or Remove User

- i. Admin must enter the Name of the user (staff member).
- ii. Admin shall enter Chart of Accounts information of the user.
- iii. Admin shall assign a role to user.
- iv. Admin shall be able to remove already existed user.

3.1.2.2 Update Profile

- i. Admin shall be able to edit the User Profile.
- ii. Admin shall be able to save the changes at any time.

3.1.2.3 View Users List

- i. Admin shall be able to View the List of Users.
- ii. Admin shall be able to sort the list on Date they joined or on other attributes.
- iii. Admin shall be able to Search a specific user.

3.1.2.4 Mark Attendance

- i. Admin shall be able to mark the attendance on daily basis.
- ii. Admin must fill the attendance status of every employee.

3.1.2.5 View Attendance

- i. Admin shall be able to see the list of attendance sheets by date.
- ii. Admin shall be able to search for an attendance sheet.

3.1.2.6 View Reports

- i. Admin shall be able to see the list of available reports.
- ii. Admin shall be able to generate the Accounts reports.
- iii. Admin shall be able to generate Attendance reports.
- iv. Admin shall be able to generate Purchase reports.
- v. Admin shall be able to generate Sales reports.
- vi. Admin shall be able to generate Production reports.

3.1.2.7 View List of Sales

- i. Admin shall be able to see the list of all sales.
- ii. Admin shall be able to search a specific sale record.
- iii. Admin shall be able to sort the list by any attribute.

3.1.2.8 View List of Purchase

- i. Admin shall be able to see the list of all purchase.
- ii. Admin shall be able to search a specific purchase record.
- iii. Admin shall be able to sort the list by any attribute.

3.1.2.9 Check Insights on Dashboard

- i. Admin can view the dashboard.
- ii. Admin can search for specific data on dashboard.

3.1.2.10 View To-Do List

- i. Admin must be able to see the list of tasks assigned to the employees.
- ii. Admin shall be able to search a specific task assignment.
- iii. Admin shall be able to sort the list by task type.

3.1.2.11 Assign or Remove Task

- i. Admin must be able to assign tasks to the employees.
- ii. Admin shall be able to remove the assigned task from To-do List.

3.1.3 Functional Requirements for Production Officer

3.1.3.1 Start Production

- i. Production Officer shall be able to start a production of a product.
- ii. Production Officer will need to name the product.
- iii. Production Officer will have to provide or select the product composition.
- iv. Production Officer must specify the Quantity of the product to be manufactured.

3.1.3.2 View Production Related Reports

- i. Production Officer can view the list of reports only related to Production process.
- ii. Production Officer shall be able to generate any of the listed reports.

3.1.3.3 View Production Insights on Dashboard

- i. Production Officer can view the dashboard.
- ii. Production Officer can search for specific data on dashboard.

3.1.3.4 View Products

- i. Production Officer shall be able to see the list of all products.
- ii. Production Officer shall be able to search a specific product.
- iii. Production Officer shall be able to sort the list by any attribute.

3.1.3.5 View Raw Materials

- i. Production Officer shall be able to see the list of all raw materials.
- ii. Production Officer shall be able to search a specific raw material.
- iii. Production Officer shall be able to sort the list by any attribute.

3.1.3.6 View Packing Materials

- i. Production Officer shall be able to see the list of all packing materials.
- ii. Production Officer shall be able to search a specific packing material.
- iii. Production Officer shall be able to sort the list by any attribute.

3.1.3.7 View Product Composition

- i. Production Officer shall be able to see the list of all products.
- ii. Production Officer shall be able to search a specific product.
- iii. Production Officer shall be able to sort the list by any attribute.

3.1.4 Functional Requirements for Accounts Officer

3.1.4.1 Generate Accounting Reports

- i. Accounts Officer can see list of accounting related reports.
- ii. Accounts Officer can generate Salary slips.
- iii. Accounts Officer can generate voucher reports.
- iv. Accounts Officer can generate Payroll and many other reports.

3.1.4.2 Add Chart of Accounts

- i. Accounts Officer must enter the Title of the Account.
- ii. Accounts Officer must enter the account number.
- iii. Accounts Officer shall enter the account type.
- iv. Accounts Officer shall be able to freeze already existed account.

3.1.4.3 View Chart of Accounts

- i. Accounts Officer shall be able to see the list of all COA.
- ii. Accounts Officer shall be able to search a specific COA.
- iii. Accounts Officer shall be able to sort the list by any attribute.

3.1.4.4 Add a Voucher

- i. Accounts Officer must enter the Voucher number.
- ii. Accounts Officer must enter the account number.
- iii. Accounts Officer shall enter the voucher type.

3.1.4.5 View a Voucher

- i. Accounts Officer shall be able to see the list of all Voucher.

- ii. Accounts Officer shall be able to search a specific Voucher.
- iii. Accounts Officer shall be able to sort the list by any attribute.

3.1.5 Functional Requirement for Sales Officer

3.1.5.1 Add Orders

- i. Sales Officer must enter the enter the Customer reference number.
- ii. Sales Officer must enter the account number.
- iii. Sales Officer shall enter the Order type.
- iv. Sales Officer must enter the ordered product.
- v. Sales Officer has to specify the quantity of the product.

3.1.5.2 View Sales

- i. Sales Officer shall be able to see the list of all Sales.
- ii. Sales Officer shall be able to search a specific Sale.
- iii. Sales Officer shall be able to sort the list by any attribute.

3.1.5.3 Edit Sales

- i. Sales Officer shall be able to Edit the specific sale.
- ii. Sales Officer shall be able to save the changes at anytime

3.1.5.4 View Products

- i. Sales Officer shall be able to see the list of all products.
- ii. Sales Officer shall be able to search a specific product.
- iii. Sales Officer shall be able to sort the list by any attribute.

3.1.5.5 Add Customer

- i. Sales Officer must enter the enter the Customer reference number.
- ii. Sales Officer must enter the Customer account number.
- iii. Sales Officer shall enter the Address.
- iv. Sales Officer must enter the Contact information.

3.1.5.6 View Customer

- i. Sales Officer shall be able to see the list of all Customers.
- ii. Sales Officer shall be able to search a specific Customer.
- iii. Sales Officer shall be able to sort the list by any attribute.

3.1.5.7 Edit Customer

- i. Sales Officer shall be able to Edit the specific Customer data.
- ii. Sales Officer shall be able to save the changes at any time.

3.1.6 Functional Requirements for Purchase Officer

3.1.6.1 Add Raw Material

- i. Purchase Officer must enter the enter the name of material.
- ii. Purchase Officer must enter the all-relative information.

3.1.6.2 View Raw Material

- i. Purchase Officer shall be able to see the list of all Raw materials.
- ii. Purchase Officer shall be able to search a specific Raw material.
- iii. Purchase Officer shall be able to sort the list by any attribute

3.1.6.3 Edit Raw Material

- i. Purchase Officer shall be able to Edit the Raw material.
- ii. Purchase Officer shall be able to save the changes at any time.

3.1.6.4 Add Packing Material

- i. Purchase Officer must enter the name of material.
- ii. Purchase Officer must enter the all-relative information.

3.1.6.5 View Packing Material

- i. Purchase Officer shall be able to see the list of all Packing materials.
- ii. Purchase Officer shall be able to search a specific Packing material.
- iii. Purchase Officer shall be able to sort the list by any attribute.

3.1.6.6 Edit Packing Material

- i. Purchase Officer shall be able to Edit the Packing material.
- ii. Purchase Officer shall be able to save the changes at any time.

3.1.6.7 Add Purchase

- i. Purchase Officer must enter the Supplier reference number.
- ii. Purchase Officer must enter the account number.
- iii. Purchase Officer shall enter the Order type.
- iv. Purchase Officer must enter the ordered material.
- v. Purchase Officer have to specify the quantity of the material.

3.1.6.8 View Purchase

- i. Purchase Officer shall be able to see the list of all Purchases.
- ii. Purchase Officer shall be able to search a specific Purchase.
- iii. Purchase Officer shall be able to sort the list by any attribute.

3.1.6.9 Edit Purchase

- i. Purchase Officer shall be able to Edit the specific purchase.
- ii. Purchase Officer shall be able to save the changes at any time.

3.2 Non-Functional Requirements

This section addresses the non-functional requirements of our system.

3.2.1 Performance

Our application can be accessed by various users at their systems since it's a web-based system. It should be able to cater at least 100 users at a time.

3.2.2 Usability

The system will be easy to use and the UI will be attractive and will provide guidelines to the users.

3.2.3 Sustainability

The application can be accessed from any popular web browser with a system having the mentioned specifications in hardware requirements.

3.2.4 Reliability

The system will perform all the above-mentioned functionalities without failure.

3.2.5 Flexibility

The application will be flexible and extensible because it will be scalable for adding other features as well. Therefore, our approach will be more of a generalized one rather than a specific approach.

3.3 Hardware and Software Requirements

This section lists the hardware and software requirements that we will use to deploy and develop our system.

3.3.1 Hardware Requirements

These requirements are a baseline for the system to perform its core functionalities. These requirements are mentioned below:

1. The computer system should at least have a dual-core processor and 4 GBs of RAM will be well enough to perform the tasks we need.
2. The system must have a browser (one of the popular ones) to run the application.

3.3.2 Software Requirements

1. We will be using React JS for the front end of our application.
2. Moreover, we will be using Django 3.2 for our web framework.
3. For our database we will be using SQL as our language.

3.4 System Architecture

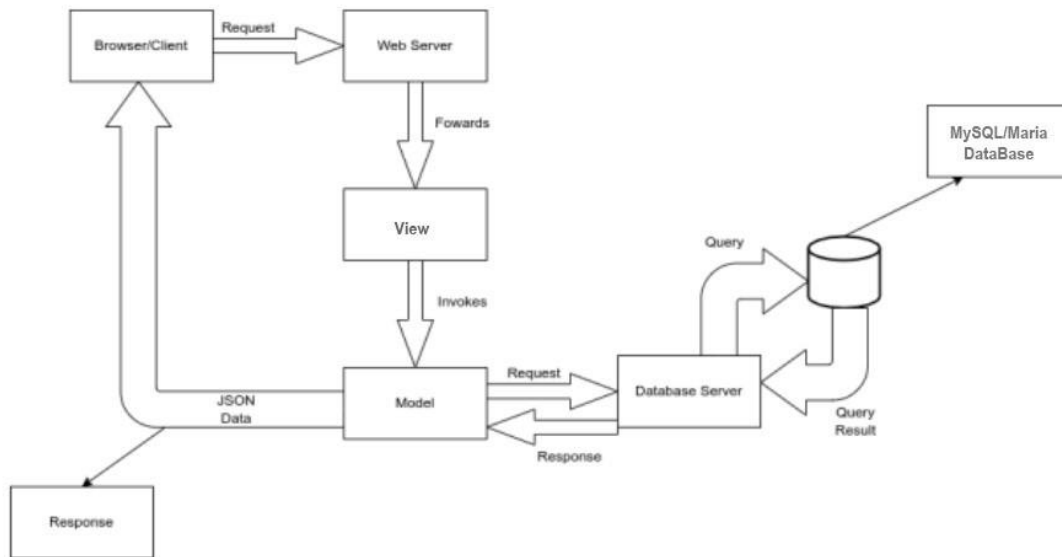


Fig 3-1: General System Architecture of Web Application
Detailed Architectural Diagram of MedPy ERP

The main architecture of our system is the MVT architecture [1]. The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a MySQL database. The View is the user interface — what you see in your browser when you render a website. It is represented by HTML/CSS/JavaScript and Jinja files. A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted. After the data is fetched the model and view interact and the front-end will represent the data to the user.

3.4.1 Admin Component

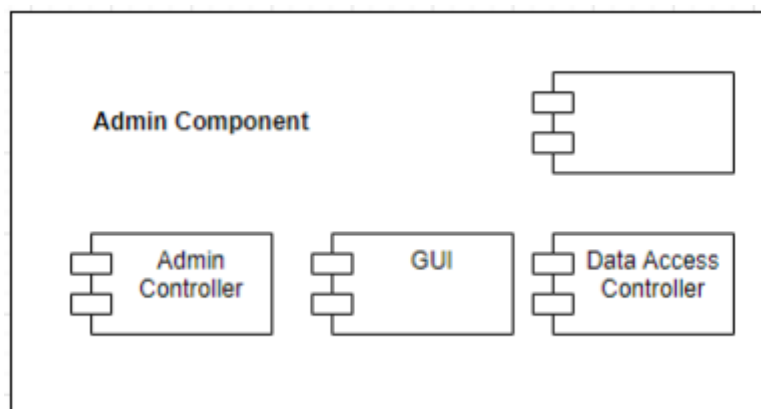


Fig 3-2: Admin Component
Component Required to implement Admin Functionality

This component handles all of the requests related to the admin related functionalities. First it will collaborate with Data Access Layer to get/store the required information and then using GUI controller it will display the required results.

3.4.2 Search Component

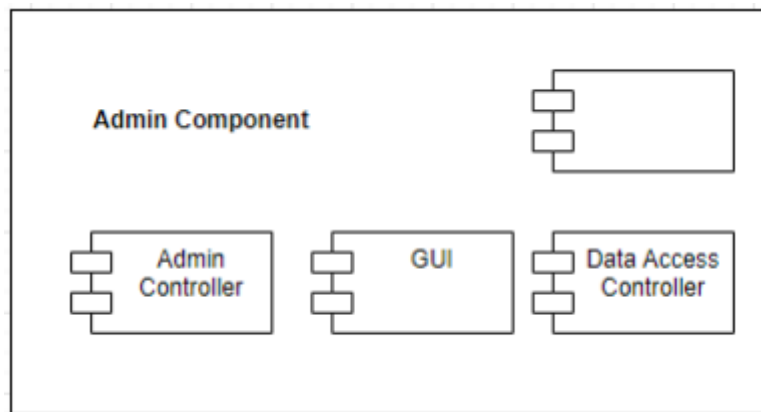


Fig 3-3: Search Component
Component Required to implement Search Functionality

This component handles all search queries i.e., whenever a user search for anything like item or any other thing that component will handle that request and generate result. Search-Controller collaborates with Data Access Controller to get the required results and then it will collaborate with GUI to display those results.

3.4.3 Account Component

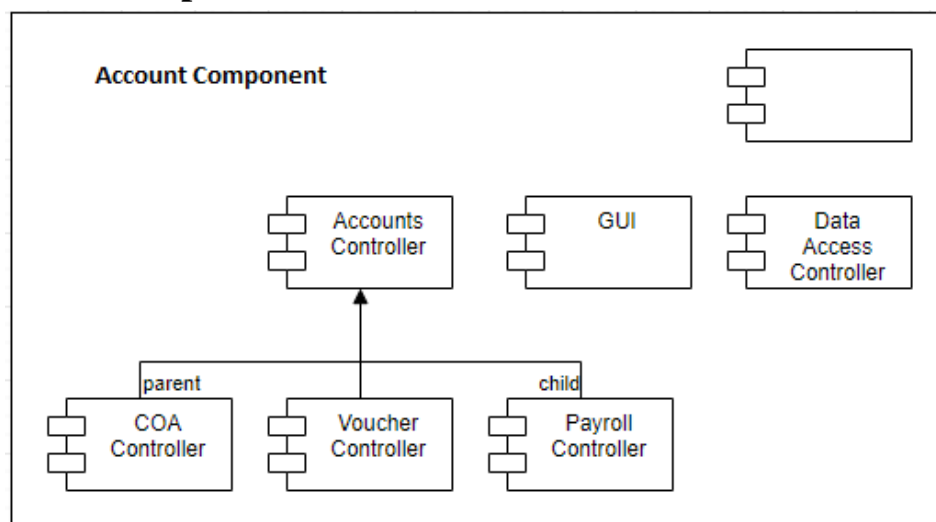


Fig 3-4: Account Component
Component Required to implement the Accounts Functionality

This component handles all of the requests related to the account related functionalities. First, it chooses between which of the controller is to use COA, Payroll, or Voucher controller then it will collaborate with Data Access Layer to get/store the required information and then using GUI controller it will display the required results.

3.4.4 Attendance Component

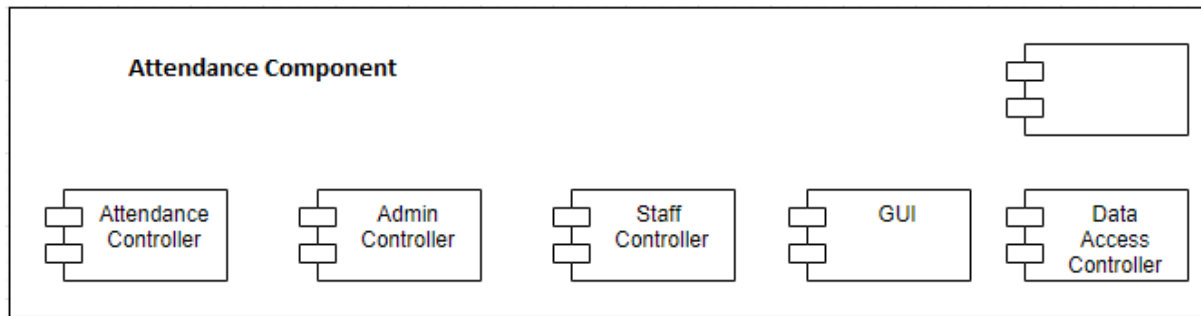


Fig 3-5: Attendance Component

Component Required to implement the Attendance Functionality

This component handles all of the requests related to the attendance related functionalities. First, the admin controller is used to access the attendance controller and then it uses the staff controller to mark the attendance. Then it will collaborate with Data Access Layer to get/store the required information and then using GUI controller it will display the required results.

3.4.5 Authentication Component

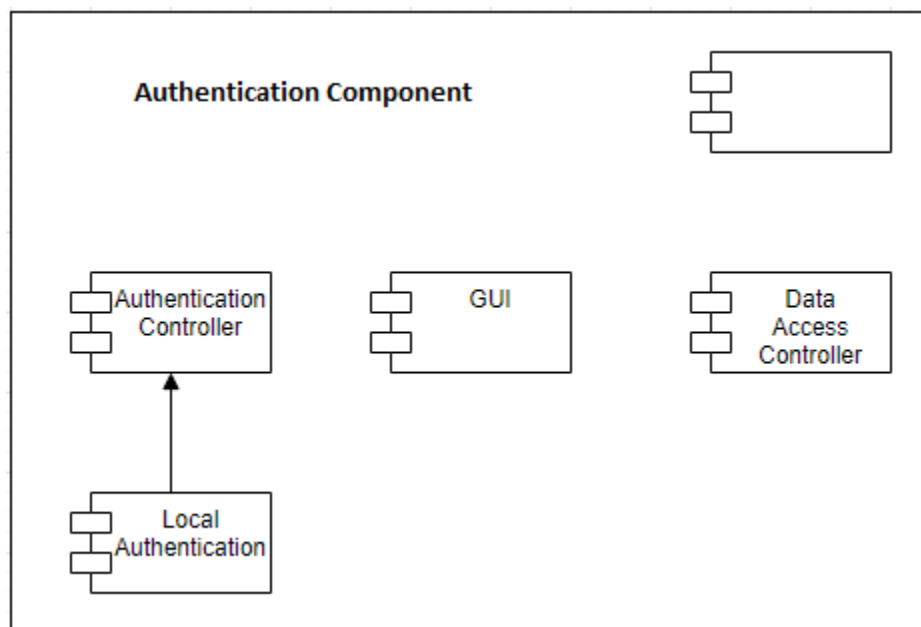


Fig 3-6: Authentication Component

Component Required to implement the Authentication Functionality

This component handles all the requests related to authentication. Then it collaborates with Data Access Controller and display the results using GUI controller.

3.4.6 Customer Component

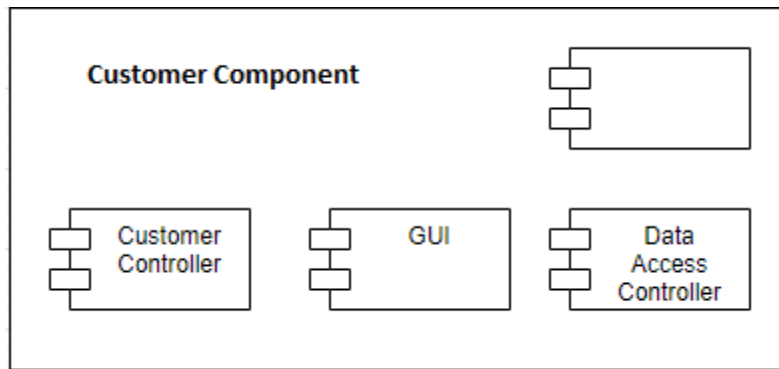


Fig 3-7: Customer Component

Component Required to implement the Customer Functionality

This component handles all of the requests related to the customer of some specific store. Uses Data Access Controller to get/store the information and display the result using GUI controller.

3.4.7 Dashboard Component

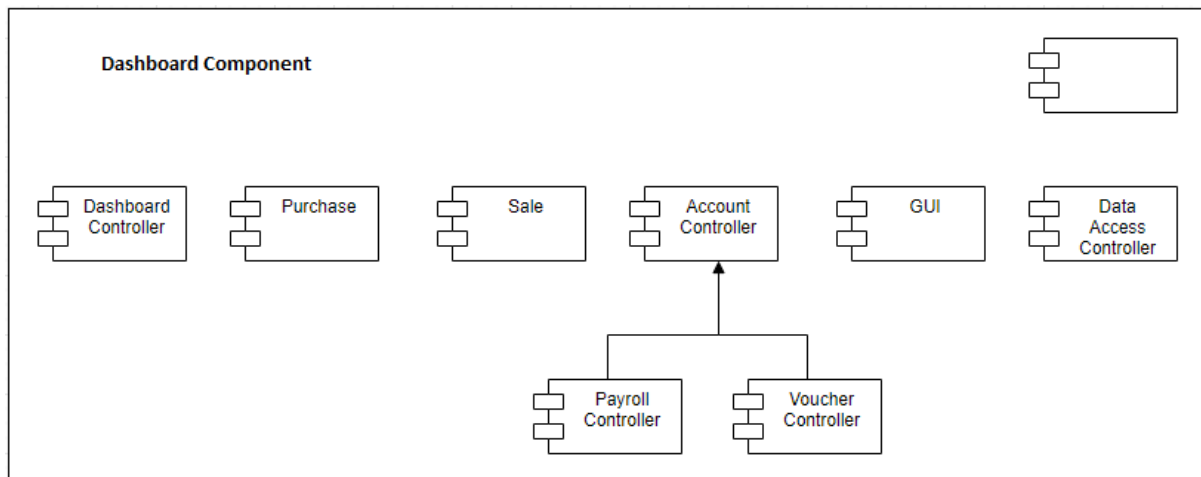


Fig 3-8: Dashboard Component

Component Required to implement the Dashboard Functionality

This component handles all of the requests related to the dashboard related functionalities. First, it collaborates with purchase sale controller and choose between account controllers and then it will collaborate with Data Access Layer to get/store the required information and then using GUI controller it will display the required results.

3.4.8 Raw Material Component

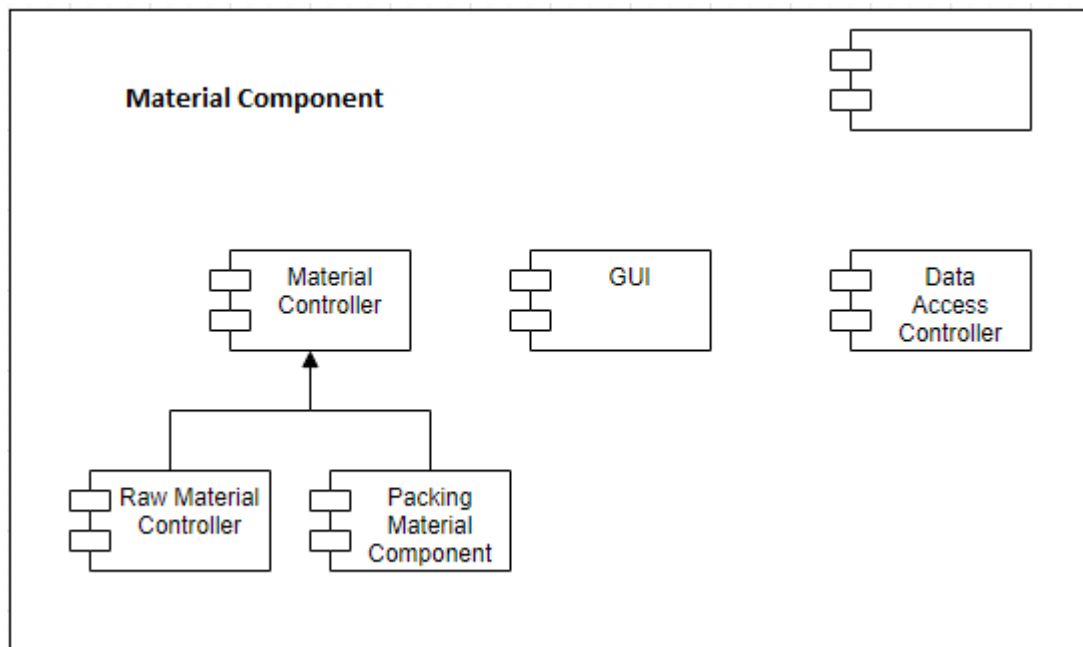


Fig 3-9: Raw Material Component

Component required to implement the Raw Material Functionality

This component handles all of the requests related to the material related functionalities. First, it chooses between which of the controller is to be used between raw material or packing material controller then it will collaborate with Data Access Layer to get/store the required information and then using GUI controller it will display the required results.

3.4.9 Product Component

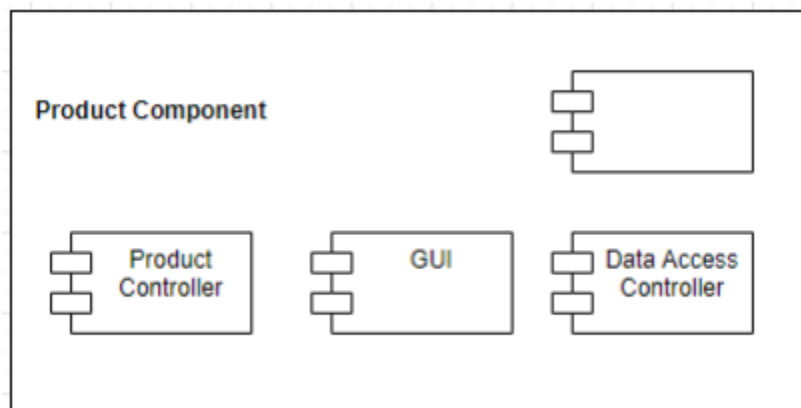


Fig 3-10: Product Component

Component Required to implement Product Functionality

This component handles all the requests related to Products like create new product, edit it and so on. It will collaborate with Data Access Controller and display the results using GUI controller.

3.4.10 Production Component

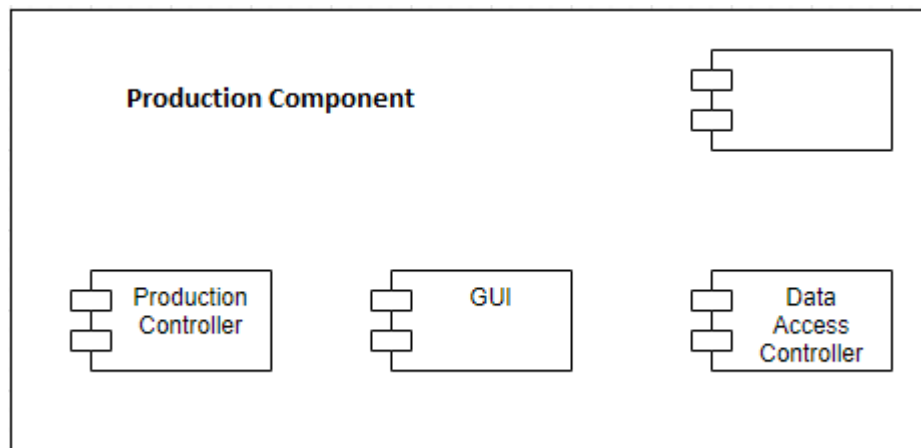


Fig 3-11: Production Component

Component Required to implement Production Functionality

This component handles all of the requests related to the Production of some specific store. First, collaborates with material components then uses Data Access Controller to get/store the information and display the result using GUI controller.

3.4.11 Purchase Component

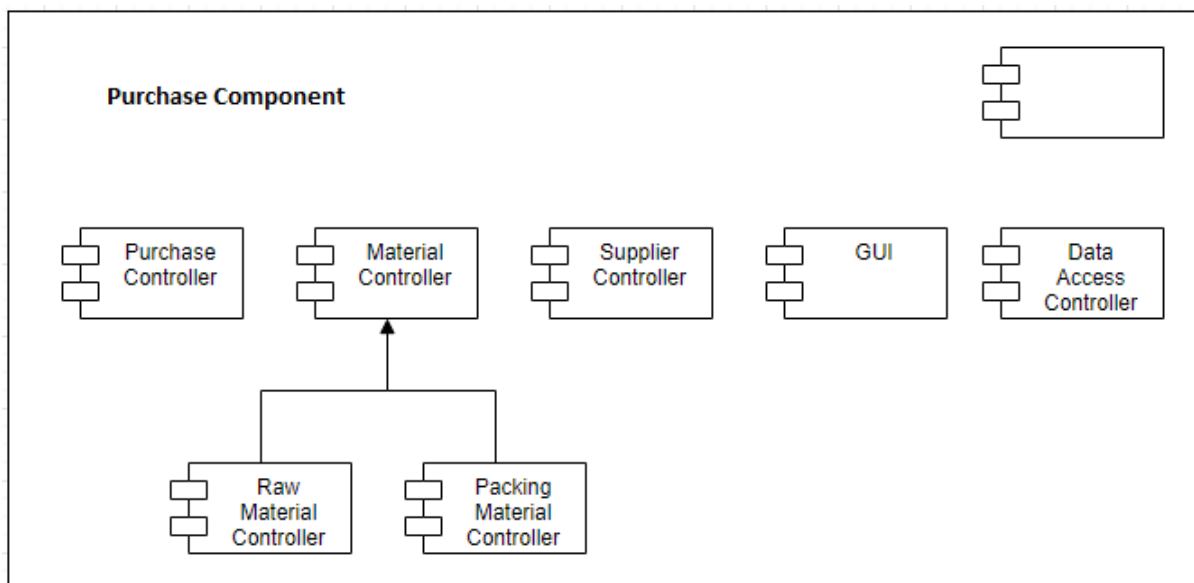


Fig 3-12: Purchase Component

Component Required to Purchase Functionality

This component handles all of the requests related to the purchase of some specific Material. It chooses which type of material is to be bought from supplier using supplier and material controller and then uses Data Access Controller to get/store the information and display the result using GUI controller.

3.4.12 Sale Component

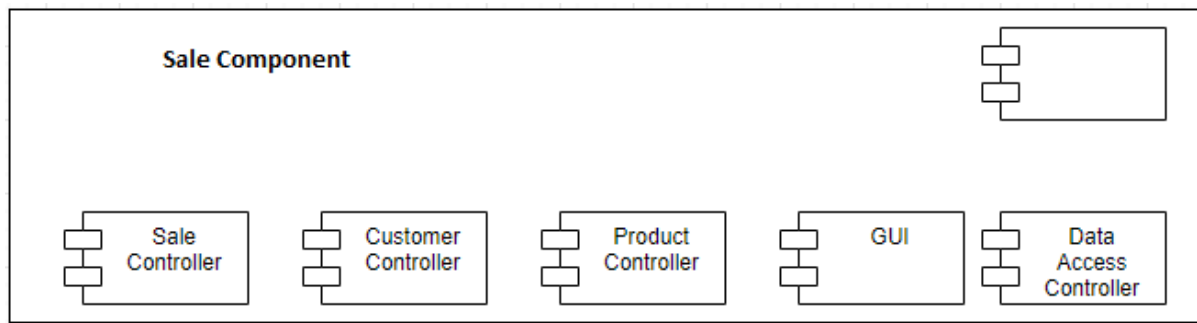


Fig 3-13: Sale Component
Component Required to implement the Sale Functionality

This component handles all of the requests related to the sale of some specific product. First, some customer controller wants to buy product using product controller which is done by the use of sale controller and then it uses Data Access Controller to get/store the information and display the result using GUI controller.

3.4.13 Staff Component

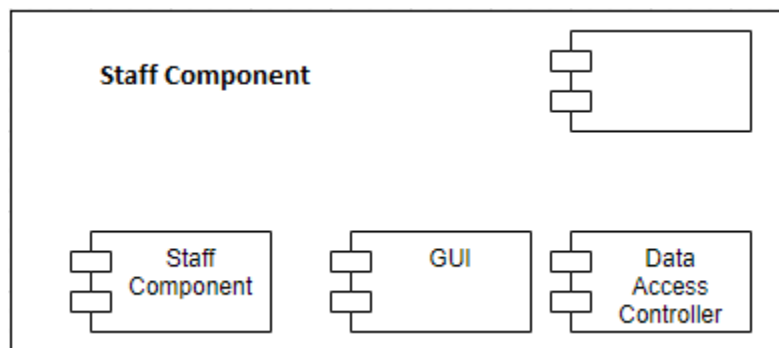


Fig 3-14: Staff Component
Component Required to implement the Staff Functionality

This component handles all of the requests related to the staff of some specific store. Uses Data Access Controller to get/store the information and display the result using GUI controller.

3.4.14 Supplier Component

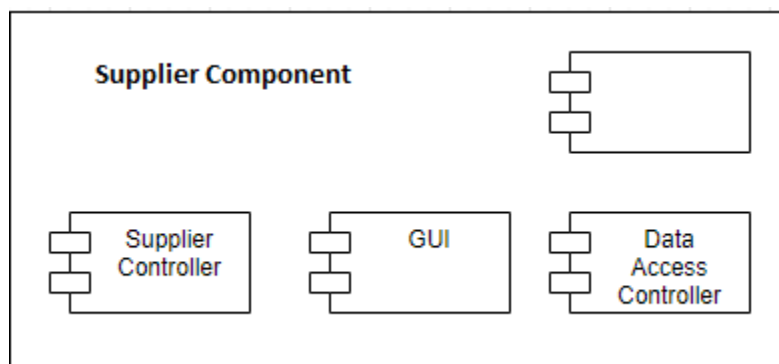


Fig 3-15: Supplier Component
Component Required to implement Supplier Functionality

This component handles all of the requests related to the supplier of some specific Material. Uses Data Access Controller to get/store the information and display the result using GUI controller.

3.5 Architectural Strategies

3.5.1 Use of a Particular Product

- Since it is a web application, we will use three main parts; server, client and database.
- We will use MySQL for our database that is hosted by MySQL server. We will be using ReactJS, Bootstrap, CSS for our Interface and the client side.
- Moreover, we will use Django for our server side. Furthermore, we will be using Gunicorn HTTP for our connectivity purposes.

3.5.2 Reuse of Existing Software Components

We will be using built-in Django libraries and services for this project thus reusing already available components. We will be using built-in CSS and JS classes provided by Bootstrap [3].

3.5.3 Future Plans

For our FYP, it is restricted to only one pharmaceutical company but in future we can extend it to other medical related firms as well as make changes that could be used in other industries as well.

3.5.4 Memory Management

The primary data will be fetched from the database on runtime. Moreover, some of the data will also be stored in cookies for a better performance.

3.5.5 Centralized Data over Distributed

All our data will be stored in the MySQL database.

3.5.6 User Interface Paradigms

Our user interface will be simple, self-explanatory and easy to use and learn. Overall design will be user friendly, so that any kind of user can interact with our application with minimum difficulty.

3.6 Use Cases

3.6.1 Login

Name		Login	
Actors		Admin, Accounts Manager, Sales Manager, Purchase Manager	
Summary		User will login using email and password to use the platform and the session will be created	
Pre-Conditions		User is on the login page and is registered	
Post-Conditions		Redirect to the dashboard and session will be created.	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	User enters Email		
2	User Enters Password		
3	User Press Login button	4	System will authenticate, identify user and redirect it to the dashboard and successfully create session for user
Alternative Flow			
1-A	User enters invalid Email	4-A	The system responds with an error message: <i>Incorrect email entered</i>
2-A	User enters invalid Password	4-B	The system responds with an error message: <i>Incorrect password entered</i>

3.6.2 Manage Password

Name		Manage Password	
Actors		Admin, Accounts Manager, Sales Manager, Purchase Manager	
Summary		User can change password	
Pre- Conditions		User is on the Manage Password page and user is registered.	
Post- Conditions		Password is changed	
Special Requirements		Password should fulfill criteria.	
Basic Flow			
Actor Action		System Response	
1	User enters new Password according to criteria.	2	Password is changed
Alternative Flow			
1-A	User enters new Password not according to criteria.	2-A	Password is not changed.

3.6.3 Logout

Name		Logout	
Actors		Admin, Accounts Manager, Sales Manager, Purchase Manager	
Summary		User can logout of website	
Pre-Conditions		User is logged in	
Post-Conditions		Session expires and successfully logged out	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	User presses logout button	2	System redirects to login page

3.6.4 View Staff

Name		View Staff	
Actors		Admin	
Summary		Admin is able to see the list of all active/inactive staff members	
Pre-Conditions		Admin is logged in	
Post-Conditions		List of Staff members is shown	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	User presses Staff Button	2	System redirects to Staff page

3.6.5 Add Staff

Name		Add Staff	
Actors		Admin	
Summary		Admin is able to add a new staff member	
Pre-Conditions		Admin is logged in and on Staff page	
Post-Conditions		Staff member is added in database along with its details	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Admin clicks on add User button	2	System redirects to the add user page
3	Admin shall provide a username		
4	Admin shall provide a CNIC		
5	Admin shall enter the role of the User		
6	Admin shall provide the contact		
7	Admin clicks the Submit button	8	The Staff member is added and system redirects to the dashboard
Alternative Flow			
3-A	Amin did not provide the username	8-A	Display “Field must not be empty”
4-A	Admin did not provide a CNIC	8-B	Display “Field must not be empty”
5-A	Admin did not enter the role of the User	8-C	Display “Field must not be empty”
6-A	Admin did not provide the contact	8-D	Display “Field must not be empty”

3.6.6 Remove Staff

Name		Remove Staff	
Actors		Admin	
Summary		Admin can remove a particular Staff member from list	
Pre-Conditions		Admin is on Staff page	
Post-Conditions		Staff member is removed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Admin presses remove button	2	Record is removed

3.6.7 Edit Staff

Name	Edit Staff		
Actors	Admin		
Summary	Admin can edit the information of a particular staff member		
Pre-Conditions	Admin is on Staff page		
Post-Conditions	Record of staff member is updated		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Admin clicks on edit button of a particular staff member from list	2	System will display edit record page
3	Admin enters updated record details		
4	Admin clicks on submit button	5	System will verify the data and update the record information
Alternative Flow			
3-A	Admin enters invalid data in a field.	5-A	System prompts invalid data error.
3-B	Discard button is clicked	5-B	The previous record is saved by default

3.6.8 Search Staff

Name		Search Staff	
Actors		Admin	
Summary		Admin can Search a particular staff member	
Pre-Conditions		Admin is on Staff page	
Post-Conditions		The specific record is filtered out from the list	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Admin enter the Staff member information in search bar	2	System will filter out the searched staff member
Alternative Flow			
1-A	Admin enters non-existed staff member	2-A	System prompts no data found

3.6.9 View Attendance

Name		View Attendance	
Actors		Admin	
Summary		Admin is able to see the list of all the marked attendance sheets	
Pre-Conditions		Admin is logged in	
Post-Conditions		List of attendance sheets is shown	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Admin presses Attendance Button	2	System redirects to Attendance page

3.6.10 Add Attendance

Name	Add Attendance		
Actors	Admin		
Summary	Admin is able to add a new Attendance sheet		
Pre-Conditions	Admin is logged in and on Attendance page		
Post-Conditions	Attendance sheet is added in database along with its details		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Admin clicks on add sheet button	2	System redirects to the add sheet page
3	Admin selects the attendance status for every staff member in sheet		
4	Admin clicks the Submit button	5	The Attendance sheet is added and system redirects to the Attendance page

3.6.11 Remove Attendance

Name		Remove Attendance	
Actors		Admin	
Summary		Admin can remove a particular Attendance sheet	
Pre-Conditions		Admin is on attendance page	
Post-Conditions		Attendance sheet is removed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Admin presses remove Sheet button	2	Sheet is removed

3.6.12 Edit Attendance

Name	Edit Attendance		
Actors	Admin		
Summary	Admin can edit the information of a particular Attendance sheet		
Pre-Conditions	Admin is on Attendance page		
Post-Conditions	Record of attendance sheet is updated		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Admin clicks on edit button of a particular attendance sheet from list	2	System will display edit sheet page
3	Admin enters updated record details		
4	Admin clicks on submit button	5	System will verify the data and update the record's information
Alternative Flow			
3-A	Admin enters invalid data in a field.	5-A	System prompts invalid data error.
4	Discard button is pressed	5-B	Previous data is saved by default

3.6.13 Search Attendance

Name		Search Attendance	
Actors		Admin	
Summary		Admin can Search a particular Attendance sheet	
Pre-Conditions		Admin is on Attendance sheet	
Post-Conditions		The specific record is filtered out from the list	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Admin enter the Attendance sheet information in search bar	2	System will filter out the searched record
Alternative Flow			
1-A	Admin enters non-existed Attendance sheet	2-A	System prompts no data found

3.6.14 Generate Reports

Name	Generate Reports		
Actors	Admin, Accounts Manager, Sales Manager, Purchase Manager		
Summary	User can generate report of any kind		
Pre-Conditions	User is logged in		
Post-Conditions	Report is shown in pdf		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	User clicks on the Reports button	2	The Reports portal is opened
3	User selects the time period in which wants the reports		
4	User fills the required fields to generate the report		
5	User enters the report type		
6	User selects the report to generate from dropdown	7	The report is generated and shown as a pdf
Alternative Flow			
4-A	User does not fill the required fields	7-A	Display “Field must not be empty”
5-A	User does not enter the report type	7-B	Display “Field must not be empty”

3.6.15 View To-Do List

Name	View To-do List		
Actors	Admin		
Summary	Admin is able to see the list of all active/inactive tasks		
Pre-Conditions	Admin is logged in		
Post-Conditions	List of tasks is shown		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	User presses To-do list Button	2	System redirects to To-do list page

3.6.16 Add Task

Name	Add Task		
Actors	Admin		
Summary	Admin is able to add a new task		
Pre-Conditions	Admin is logged in and on To-do list page		
Post-Conditions	Task is added in database along with its details		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Admin clicks on add User button	2	System redirects to the add task page
3	Admin shall provide a task name		
4	Admin shall provide a task description		
5	Admin shall enter the assigned to information		
6	Admin clicks the Submit button	7	The task is added and system redirects to the To-do list
Alternative Flow			
3-A	Amin did not provide the task name	7-A	Display “Field must not be empty”
4-A	Admin did not provide a description	7-B	Display “Field must not be empty”
5-A	Admin did not enter the assigned to information	7-C	Display “Field must not be empty”

3.6.17 Remove Task

Name		Remove Task	
Actors		Admin	
Summary		Admin can remove a particular task from list	
Pre-Conditions		Admin is on To-do list page	
Post-Conditions		Task is removed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Admin presses remove button	2	Task is removed

3.6.18 Edit Task

Name	Edit Task		
Actors	Admin		
Summary	Admin can edit the information of a particular task		
Pre-Conditions	Admin is on To-do list page		
Post-Conditions	Record of task is updated		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Admin clicks on edit button of a particular task from list	2	System will display edit record page
3	Admin enters updated record details		
4	Admin clicks on submit button	5	System will verify the data and update the record information
Alternative Flow			
3-A	Admin enters invalid data in a field.	5-A	System prompts invalid data error.
3-B	Discard button is clicked	5-B	The previous record is saved by default

3.6.19 Search Task

Name		Search Task	
Actors		Admin	
Summary		Admin can Search a particular task	
Pre-Conditions		Admin is on To-do list page	
Post-Conditions		The specific record is filtered out from the list	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Admin enter the task information in search bar	2	System will filter out the searched task
Alternative Flow			
1-A	Admin enters non-existed task	2-A	System prompts no data found

3.6.20 Start Production

Name		Start Production	
Actors		Production Officer	
Summary		Production Officer is able to start a new production of a product required by the customer using appropriate raw materials and packing materials	
Pre-Conditions		Production Officer is on Production page	
Post-Conditions		Production record is added in database along with its details	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Production Officer clicks on Start Production button	2	System redirects to the Start production page
3	Production Officer shall provide a product name		
4	Production Officer shall provide a list of raw materials		
5	Production Officer shall provide a list of packing materials		
6	Production Officer needs to enter the status of the production		
7	Production Officer clicks the Submit button	8	The Production is started and system redirects to the Production
Alternative Flow			
3-A	Production Officer did not provide the task name	8-A	Display “Field must not be empty”
4-A	Production Officer did not provide a description	8-B	Display “Field must not be empty”
5-A	Production Officer did not enter the assigned to information	8-C	Display “Field must not be empty”
6-A	Production Officer did not enter the status of the production	8-D	Display “Field must not be empty”

3.6.21 Remove Production

Name		Stop Production	
Actors		Production Officer	
Summary		Production Officer can remove a particular task from list	
Pre-Conditions		Production is already in the database	
Post-Conditions		Production is Stopped	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Production Officer presses remove button	2	Production is removed

3.6.22 View COA

Name	View COA		
Actors	Accounts Officer		
Summary	Accounts Officer is able to see the list of all active/inactive COA		
Pre-Conditions	Accounts Officer is logged in		
Post-Conditions	List of COA is shown		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	User presses COA Button	2	System redirects to COA page

3.6.23 Add COA

Name		Add COA	
Actors		Accounts Officer	
Summary		Accounts Officer is able to add a new COA	
Pre-Conditions		Accounts Officer is logged in and on COA page	
Post-Conditions		COA is added in database along with its details	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Accounts Officer clicks on add User button	2	System redirects to the add COA page
3	Accounts Officer shall provide an account name		
4	Accounts Officer shall provide an account number		
5	Accounts Officer shall enter the account type		
6	Accounts Officer shall provide the contact		
7	Accounts Officer clicks the Submit button	8	The COA is added and system redirects to the COA page
Alternative Flow			
3-A	Accounts Officer did not provide the account name	8-A	Display “Field must not be empty”
4-A	Accounts Officer did not provide a account number	8-B	Display “Field must not be empty”
5-A	Accounts Officer did not enter the account type	8-C	Display “Field must not be empty”
6-A	Accounts Officer did not provide the contact	8-D	Display “Field must not be empty”

3.6.24 Remove COA

Name		Remove COA	
Actors		Accounts Officer	
Summary		Accounts Officer can remove a particular COA from list	
Pre-Conditions		COA already exists in the list	
Post-Conditions		COA is removed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Accounts Officer presses remove button	2	Record is removed

3.6.25 Edit COA

Name	Edit COA		
Actors	Accounts Officer		
Summary	Accounts Officer can edit the information of a particular COA		
Pre-Conditions	COA already exists		
Post-Conditions	Record of COA is updated		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Accounts Officer clicks on edit button of a particular staff member from list	2	System will display edit record page
3	Accounts Officer enters updated record details		
4	Accounts Officer clicks on submit button	5	System will verify the data and update the record information
Alternative Flow			
3-A	Accounts Officer enters invalid data in a field.	5-A	System prompts invalid data error.
3-B	Discard button is clicked	5-B	The previous record is saved by default

3.6.26 Search COA

Name	Search COA		
Actors	Accounts Officer		
Summary	Accounts Officer can Search a particular COA		
Pre-Conditions	Accounts Officer is on COA page		
Post-Conditions	The specific record is filtered out from the list		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Accounts Officer enter the COA information in search bar	2	System will filter out the searched COA
Alternative Flow			
1-A	Accounts Officer enters non-existed COA	2-A	System prompts no data found

3.6.27 View Voucher

Name		View Voucher	
Actors		Accounts Officer	
Summary		Accounts Officer is able to see the list of all Vouchers	
Pre-Conditions		Accounts Officer is logged in	
Post-Conditions		List of Vouchers is shown	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	User presses Voucher Button	2	System redirects to Voucher page

3.6.28 Add Voucher

Name		Add Voucher	
Actors		Accounts Officer	
Summary		Accounts Officer is able to add a new Voucher	
Pre-Conditions		Accounts Officer is logged in and on Voucher page	
Post-		Voucher is added in database along with its details	
Conditions			
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Accounts Officer clicks on add Voucher button	2	System redirects to the add Voucher page
3	Accounts Officer shall provide a Voucher name		
4	Accounts Officer shall provide a Voucher number		
5	Accounts Officer shall enter the Voucher type		
6	Accounts Officer shall provide the Voucher date		
7	Accounts Officer clicks the Submit button	8	The Voucher is added and system redirects to the Voucher page
Alternative Flow			
3-A	Accounts Officer did not provide the Voucher name	8-A	Display “Field must not be empty”
4-A	Accounts Officer did not provide a Voucher number	8-B	Display “Field must not be empty”
5-A	Accounts Officer did not enter the account type	8-C	Display “Field must not be empty”
6-A	Accounts Officer did not provide the Voucher date	8-D	Display “Field must not be empty”

3.6.29 Remove Voucher

Name		Remove Voucher	
Actors		Accounts Officer	
Summary		Accounts Officer can remove a particular Voucher from list	
Pre-Conditions		Voucher already exists in the list	
Post-Conditions		Voucher is removed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Accounts Officer presses remove button	2	Record is removed

3.6.30 Edit Voucher

Name	Edit Voucher		
Actors	Accounts Officer		
Summary	Accounts Officer can edit the information of a particular Voucher		
Pre-Conditions	Voucher already exists		
Post-Conditions	Record of Voucher is updated		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Accounts Officer clicks on edit button of a particular Voucher from list	2	System will display edit record page
3	Accounts Officer enters updated record details		
4	Accounts Officer clicks on submit button	5	System will verify the data and update the record information
Alternative Flow			
3-A	Accounts Officer enters invalid data in a field.	5-A	System prompts invalid data error.
3-B	Discard button is clicked	5-B	The previous record is saved by default

3.6.31 Search Voucher

Name		Search Voucher	
Actors		Accounts Officer	
Summary		Accounts Officer can Search a particular Voucher	
Pre-Conditions		Accounts Officer is on Voucher page	
Post-Conditions		The specific record is filtered out from the list	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Accounts Officer enter the Voucher information in search bar	2	System will filter out the searched Voucher
Alternative Flow			
1-A	Accounts Officer enters non-existed Voucher	2-A	System prompts no data found

3.6.32 View Supplier

Name	View Supplier		
Actors	Purchase Officer		
Summary	Purchase Officer is able to see the list of all Suppliers		
Pre-Conditions	Purchase Officer is logged in		
Post-Conditions	List of Suppliers is shown		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	User presses Supplier Button	2	System redirects to Supplier page

3.6.33 Add Supplier

Name		Add Supplier	
Actors		Purchase Officer	
Summary		Purchase Officer is able to add a new Supplier	
Pre-Conditions		Purchase Officer is logged in and on Supplier page	
Post-Conditions		Supplier is added in database along with its details	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Purchase Officer clicks on add Voucher button	2	System redirects to the add Supplier page
3	Purchase Officer shall provide a supplier name		
4	Purchase Officer shall provide a Supplier contact number		
5	Purchase Officer shall enter the Supplier type		
6	Purchase Officer shall provide the Supplier address		
7	Purchase Officer clicks the Submit button	8	The Supplier is added and system redirects to the Supplier page
Alternative Flow			
3-A	Purchase Officer did not provide the Supplier name	8-A	Display “Field must not be empty”
4-A	Purchase Officer did not provide a Supplier number	8-B	Display “Field must not be empty”
5-A	Purchase Officer did not enter the Supplier type	8-C	Display “Field must not be empty”
6-A	Purchase Officer did not provide the Supplier address	8-D	Display “Field must not be empty”

3.6.34 Remove Supplier

Name		Remove Supplier	
Actors		Purchase Officer	
Summary		Purchase Officer can remove a particular Supplier from list	
Pre-Conditions		Supplier already exists in the list	
Post-Conditions		Supplier is removed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Purchase Officer presses remove button	2	Record is removed

3.6.35 Edit Supplier

Name	Edit Supplier		
Actors	Purchase Officer		
Summary	Purchase Officer can edit the information of a particular Supplier		
Pre-Conditions	Supplier already exists		
Post-Conditions	Record of Supplier is updated		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Purchase Officer clicks on edit button of a particular Supplier from list	2	System will display edit record page
3	Purchase Officer enters updated record details		
4	Purchase Officer clicks on submit button	5	System will verify the data and update the record information
Alternative Flow			
3-A	Purchase Officer enters invalid data in a field.	5-A	System prompts invalid data error.
3-B	Discard button is clicked	5-B	The previous record is saved by default

3.6.36 Search Supplier

Name		Search Supplier	
Actors		Purchase Officer	
Summary		Purchase Officer can Search a particular Supplier	
Pre-Conditions		Purchase Officer is on Supplier page	
Post-Conditions		The specific record is filtered out from the list	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Purchase Officer enter the Supplier information in search bar	2	System will filter out the searched Supplier
Alternative Flow			
1-A	Purchase Officer enters non-existed Supplier	2-A	System prompts no data found

3.6.37 View Purchase

Name	View Purchase		
Actors	Purchase Officer		
Summary	Purchase Officer is able to see the list of all Purchase		
Pre-Conditions	Purchase Officer is logged in		
Post-Conditions	List of Purchase is shown		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	User presses Purchase Button	2	System redirects to Purchase page

3.6.38 Add Purchase

Name		Add Purchase	
Actors		Purchase Officer	
Summary		Purchase Officer is able to add a new Purchase	
Pre-Conditions		Purchase Officer is logged in and on Purchase page	
Post-Conditions		Purchase is added in database along with its details	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Purchase Officer clicks on add Purchase button	2	System redirects to the add Purchase page
3	Purchase Officer shall provide a Purchase date		
4	Purchase Officer shall provide a Purchase number		
5	Purchase Officer shall enter the Purchase type		
6	Purchase Officer shall provide the Purchase Item		
7	Purchase Officer clicks the Submit button	8	The Purchase is added and system redirects to the Purchase page
Alternative Flow			
3-A	Purchase Officer did not provide the Purchase date	8-A	Display “Field must not be empty”
4-A	Purchase Officer did not provide a Purchase number	8-B	Display “Field must not be empty”
5-A	Purchase Officer did not enter the Purchase type	8-C	Display “Field must not be empty”
6-A	Purchase Officer did not provide the Purchase Item	8-D	Display “Field must not be empty”

3.6.39 Remove Purchase

Name		Remove Purchase	
Actors		Purchase Officer	
Summary		Purchase Officer can remove a particular Purchase from list	
Pre-Conditions		Purchase already exists in the list	
Post-Conditions		Purchase is removed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Purchase Officer presses remove button	2	Record is removed

3.6.40 Edit Purchase

Name	Edit Purchase		
Actors	Purchase Officer		
Summary	Purchase Officer can edit the information of a particular Purchase		
Pre-Conditions	Purchase already exists		
Post-Conditions	Record of Purchase is updated		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Purchase Officer clicks on edit button of a particular Purchase from list	2	System will display edit record page
3	Purchase Officer enters updated record details		
4	Purchase Officer clicks on submit button	5	System will verify the data and update the record information
Alternative Flow			
3-A	Purchase Officer enters invalid data in a field.	5-A	System prompts invalid data error.
3-B	Discard button is clicked	5-B	The previous record is saved by default

3.6.41 Search Purchase

Name		Search Purchase	
Actors		Purchase Officer	
Summary		Purchase Officer can Search a particular Purchase	
Pre-Conditions		Purchase Officer is on Purchase page	
Post-Conditions		The specific record is filtered out from the list	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Purchase Officer enter the Purchase information in search bar	2	System will filter out the searched Purchase
Alternative Flow			
1-A	Purchase Officer enters non-existed Purchase	2-A	System prompts no data found

3.6.42 View Raw Material

Name		View Raw Material	
Actors		Purchase Officer	
Summary		Purchase Officer is able to see the list of all Raw Material	
Pre-Conditions		Purchase Officer is logged in	
Post-Conditions		List of Raw Material is shown	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	User presses Raw Material Button	2	System redirects to Raw Material page

3.6.43 Add Raw Material

Name		Add Raw Material	
Actors		Purchase Officer	
Summary		Purchase Officer is able to add a new Raw Material	
Pre-Conditions		Purchase Officer is logged in and on Raw Material page	
Post-Conditions		Raw Material is added in database along with its details	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Purchase Officer clicks on add Raw Material button	2	System redirects to the add Raw Material page
3	Purchase Officer shall provide a Raw Material name		
4	Purchase Officer shall provide a Raw Material Price		
5	Purchase Officer shall enter the Raw Material type		
6	Purchase Officer shall provide the Raw Material Quantity		
7	Purchase Officer clicks the Submit button	8	The Raw Material is added and system redirects to the Raw Material page
Alternative Flow			
3-A	Purchase Officer did not provide the Raw Material name	8-A	Display “Field must not be empty”
4-A	Purchase Officer did not provide a Raw Material Price	8-B	Display “Field must not be empty”
5-A	Purchase Officer did not enter the Raw Material type	8-C	Display “Field must not be empty”
6-A	Purchase Officer did not provide the Raw Material Quantity	8-D	Display “Field must not be empty”

3.6.44 Remove Raw Material

Name		Remove Raw Material	
Actors		Purchase Officer	
Summary		Purchase Officer can remove a particular Raw Material from list	
Pre-Conditions		Raw Material already exists in the list	
Post-Conditions		Raw Material is removed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Purchase Officer presses remove button	2	Record is removed

3.6.45 Search Raw Material

Name	Search Raw Material		
Actors	Purchase Officer		
Summary	Purchase Officer can Search a particular Raw Material		
Pre-Conditions	Purchase Officer is on Raw Material page		
Post-Conditions	The specific record is filtered out from the list		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Purchase Officer enter the Raw Material information in search bar	2	System will filter out the searched Raw Material
Alternative Flow			
1-A	Purchase Officer enters non-existed Raw Material	2-A	System prompts no data found

3.6.46 Add Packing Material

Name		Add Packing Material	
Actors		Purchase Officer	
Summary		Purchase Officer is able to add a new Packing Material	
Pre-Conditions		Purchase Officer is logged in and on Packing Material page	
Post-Conditions		Packing Material is added in database along with its details	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Purchase Officer clicks on add Packing Material button	2	System redirects to the add Packing Material page
3	Purchase Officer shall provide a Packing Material name		
4	Purchase Officer shall provide a Packing Material Price		
5	Purchase Officer shall enter the Packing Material type		
6	Purchase Officer shall provide the Packing Material Quantity		
7	Purchase Officer clicks the Submit button	8	The Packing Material is added and system redirects to the Packing Material page
Alternative Flow			
3-A	Purchase Officer did not provide the Packing Material name	8-A	Display “Field must not be empty”
4-A	Purchase Officer did not provide a Packing Material Price	8-B	Display “Field must not be empty”
5-A	Purchase Officer did not enter the Packing Material type	8-C	Display “Field must not be empty”
6-A	Purchase Officer did not provide the Packing Material Quantity	8-D	Display “Field must not be empty”

3.6.47 Remove Packing Material

Name		Remove Packing Material	
Actors		Purchase Officer	
Summary		Purchase Officer can remove a particular Packing Material from list	
Pre-Conditions		Packing Material already exists in the list	
Post-Conditions		Packing Material is removed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Purchase Officer presses remove button	2	Record is removed

3.6.48 Edit Packing Material

Name		Edit Packing Material	
Actors		Purchase Officer	
Summary		Purchase Officer can edit the information of a particular Packing Material	
Pre-Conditions		Packing Material already exists	
Post-Conditions		Record of Packing Material is updated	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Purchase Officer clicks on edit button of a particular Packing Material from list	2	System will display edit record page
3	Purchase Officer enters updated record details		
4	Purchase Officer clicks on submit button	5	System will verify the data and update the record information
Alternative Flow			
3-A	Purchase Officer enters invalid data in a field.	5-A	System prompts invalid data error.
3-B	Discard button is clicked	5-B	The previous record is saved by default

3.6.49 Add Sale

Name		Add Sale	
Actors		Sale Officer	
Summary		Sale Officer is able to add a new Sale	
Pre-Conditions		Sale Officer is logged in and on Sale page	
Post-Conditions		Sale is added in database along with its details	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Sale Officer clicks on add Sale button	2	System redirects to the add Sale page
3	Sale Officer shall provide a Sale date		
4	Sale Officer shall provide a Sale number		
5	Sale Officer shall enter the Sale type		
6	Sale Officer shall provide the Sale Item		
7	Sale Officer clicks the Submit button	8	The Sale is added and system redirects to the Sale page
Alternative Flow			
3-A	Sale Officer did not provide the Sale date	8-A	Display “Field must not be empty”
4-A	Sale Officer did not provide a Sale number	8-B	Display “Field must not be empty”
5-A	Sale Officer did not enter the Sale type	8-C	Display “Field must not be empty”
6-A	Sale Officer did not provide the Sale Item	8-D	Display “Field must not be empty”

3.6.50 Remove Sale

Name		Remove Sale	
Actors		Sale Officer	
Summary		Sale Officer can remove a particular Sale from list	
Pre-Conditions		Sale already exists in the list	
Post-Conditions		Sale is removed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Sale Officer presses remove button	2	Record is removed

3.6.51 Edit Sale

Name	Edit Sale		
Actors	Sale Officer		
Summary	Sale Officer can edit the information of a particular Sale		
Pre-Conditions	Sale already exists		
Post-Conditions	Record of Sale is updated		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Sale Officer clicks on edit button of a particular Sale from list	2	System will display edit record page
3	Sale Officer enters updated record details		
4	Sale Officer clicks on submit button	5	System will verify the data and update the record information
Alternative Flow			
3-A	Sale Officer enters invalid data in a field.	5-A	System prompts invalid data error.
3-B	Discard button is clicked	5-B	The previous record is saved by default

3.6.52 Add Product

Name		Add Product	
Actors		Sale Officer	
Summary		Sale Officer is able to add a new Product	
Pre-Conditions		Sale Officer is logged in and on Product page	
Post-Conditions		Product is added in database along with its details	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Sale Officer clicks on add Product button	2	System redirects to the add Product page
3	Sale Officer shall provide a Product name		
4	Sale Officer shall provide a Product number		
5	Sale Officer shall enter the Product type		
6	Sale Officer shall provide the Product expiry		
7	Sale Officer clicks the Submit button	8	The Product is added and system redirects to the Product page
Alternative Flow			
3-A	Sale Officer did not provide the Product name	8-A	Display “Field must not be empty”
4-A	Sale Officer did not provide a Product number	8-B	Display “Field must not be empty”
5-A	Sale Officer did not enter the Product type	8-C	Display “Field must not be empty”
6-A	Sale Officer did not provide the Product expiry	8-D	Display “Field must not be empty”

3.6.53 Remove Product

Name		Remove Product	
Actors		Sale Officer	
Summary		Sale Officer can remove a particular Product from list	
Pre-Conditions		Product already exists in the list	
Post-Conditions		Product is removed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Sale Officer presses remove button	2	Record is removed

3.6.54 Edit Product

Name	Edit Product		
Actors	Sale Officer		
Summary	Sale Officer can edit the information of a particular Product		
Pre-Conditions	Product already exists		
Post-Conditions	Record of Product is updated		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Sale Officer clicks on edit button of a particular Product from list	2	System will display edit record page
3	Sale Officer enters updated record details		
4	Sale Officer clicks on submit button	5	System will verify the data and update the record information
Alternative Flow			
3-A	Sale Officer enters invalid data in a field.	5-A	System prompts invalid data error.
3-B	Discard button is clicked	5-B	The previous record is saved by default

3.6.55 Add Customer

Name		Add Customer	
Actors		Sale Officer	
Summary		Sale Officer is able to add a new Customer	
Pre-Conditions		Sale Officer is logged in and on Customer page	
Post-Conditions		Customer is added in database along with its details	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Sale Officer clicks on add Customer button	2	System redirects to the add Customer page
3	Sale Officer shall provide a Customer name		
4	Sale Officer shall provide a Customer contact number		
5	Sale Officer shall enter the Customer type		
6	Sale Officer shall provide the Customer address		
7	Sale Officer clicks the Submit button	8	The Customer is added and system redirects to the Customer page
Alternative Flow			
3-A	Sale Officer did not provide the Customer name	8-A	Display “Field must not be empty”
4-A	Sale Officer did not provide a customer contact number	8-B	Display “Field must not be empty”
5-A	Sale Officer did not enter the Customer type	8-C	Display “Field must not be empty”
6-A	Sale Officer did not provide the Customer address	8-D	Display “Field must not be empty”

3.6.56 Remove Customer

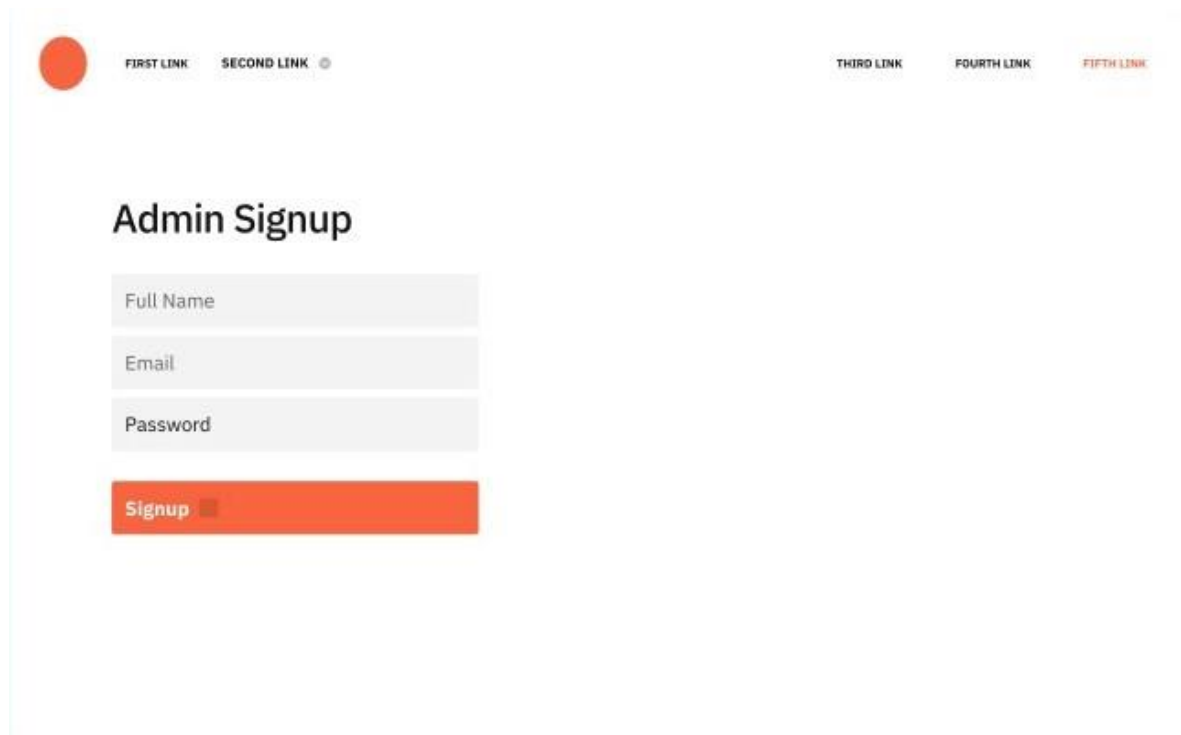
Name		Remove Customer	
Actors		Sale Officer	
Summary		Sale Officer can remove a particular Customer from list	
Pre-Conditions		Customer already exists in the list	
Post-Conditions		Customer is removed	
Special Requirements		None	
Basic Flow			
Actor Action		System Response	
1	Sale Officer presses remove button	2	Record is removed

3.6.57 Edit Customer

Name	Edit Customer		
Actors	Sale Officer		
Summary	Sale Officer can edit the information of a particular Customer		
Pre-Conditions	Customer already exists		
Post-Conditions	Record of Customer is updated		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Sale Officer clicks on edit button of a particular Customer from list	2	System will display edit record page
3	Sale Officer enters updated record details		
4	Sale Officer clicks on submit button	5	System will verify the data and update the record information
Alternative Flow			
3-A	Sale Officer enters invalid data in a field.	5-A	System prompts invalid data error.
3-B	Discard button is clicked	5-B	The previous record is saved by default

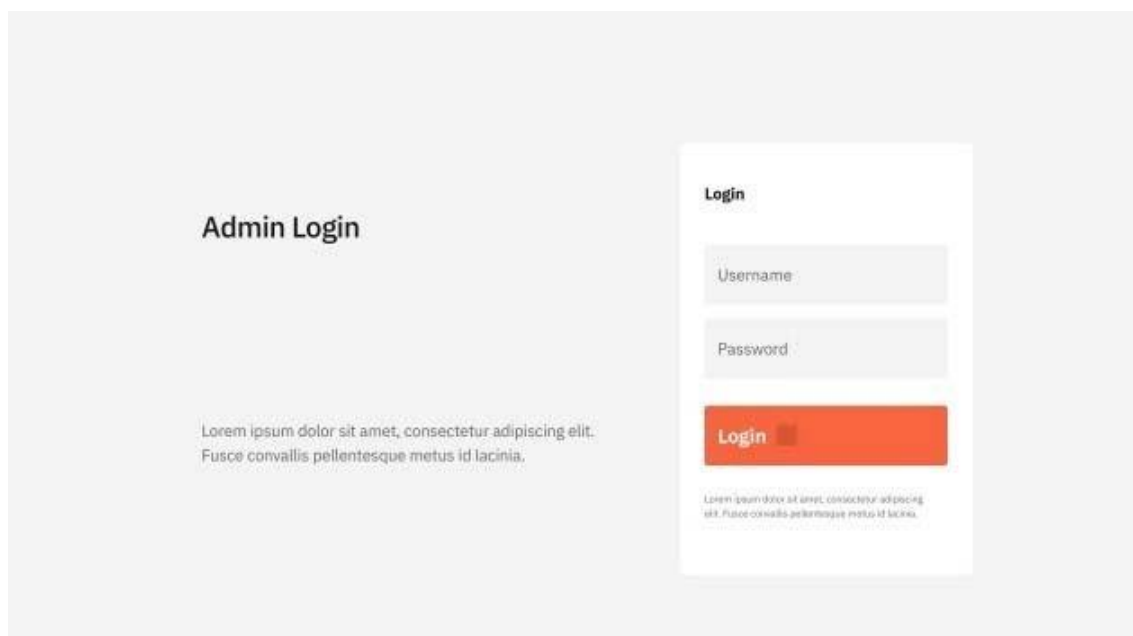
3.7 GUI

A basic depiction of what our system will look like is entailed below.



The image shows a web interface for an 'Admin Signup' page. At the top, there is a navigation bar with a red circular logo on the left and five links: 'FIRST LINK', 'SECOND LINK' (which is active and highlighted with a blue circle), 'THIRD LINK', 'FOURTH LINK', and 'FIFTH LINK'. Below the navigation bar, the main heading 'Admin Signup' is displayed. Underneath the heading are three input fields: 'Full Name', 'Email', and 'Password'. At the bottom of the form is a red 'Signup' button with a small red square icon to its right.

Fig 3-16: Admin Signup
The signup page for the Admin User



The image shows a web interface for an 'Admin Login' page. The background is a light gray. On the left side, the heading 'Admin Login' is displayed. Below the heading is a block of placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce convallis pellentesque metus id lacinia.' On the right side, there is a white box containing the login form. The form has a heading 'Login' at the top, followed by two input fields: 'Username' and 'Password'. At the bottom of the form is a red 'Login' button with a small red square icon to its right. Below the button is another block of placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce convallis pellentesque metus id lacinia.'

Fig 3-17: Admin Login
The Login page for the Admin User

Employee Attendance

March 2021

Admin: Ali Ahmad

Employee Namee	Mon 7	Tue 8	Wed 9	Thu 10	Fri 11	
	P					Add Date
		A				
						Save Attendance

Fig 3-18: View Attendance
The attendance view of the admin and managers

Pharmaceutical ERP System

MANAGE Employee

Add Employee

Name: Haider

Joining Date: 10/11/2022

Phone: 03054724675

Address: 16 A OLD OFFICER COLONY SADAR CANTT

Add Employee

Successful Employee Data Save Successfully.

All Employee Data

#ID	Name	Joining Date	Phone	Address	Added On	Action
3	Haider	2022-10-11	03054724675	16 A OLD OFFICER COLONY SADAR CANTT	10/19/2022, 10:56:41 PM	VIEW

© 2022 - 2023 MedPy ERP. Version: 1.0.5

Fig 3-19: Add User
The Add user view of the admin

Pharmaceutical ERP System

MANAGE COMPANY ACCOUNT

Add Company Account Bill

Company: ShalimarPharma Transaction Type: Debit Amount: Enter Amount

Transaction Date: mm/dd/yyyy Payment Mode: Enter Payment Mode

Add Company Transaction

All Companies Account Transactions

#ID	Company Name	Company ID	Transaction Type	Amount	Date	Payment Mode	Added On
-----	--------------	------------	------------------	--------	------	--------------	----------

© 2022 - 2023 MedPy ERP. Version: 1.0.5

Fig 3-20: Add Company Bill
The view of adding a company bill

Pharmaceutical ERP System

Manage Medicine

All Medicine

#ID	NAME	Medical Type	Buy Price	Sell Price	Batch No	Shelf No	Expire Date	Mfg Date	In Stock	Company	Added On	Action
-----	------	--------------	-----------	------------	----------	----------	-------------	----------	----------	---------	----------	--------

Manage Medicine

Name: Enter Name

Medicine Type: Enter Medicine Type

Buy Price: Enter Buy Price

Sell Price: Enter Sell Price

C GST: Enter C-GST

S GST: Enter S-GST

Batch No: Enter Batch No

© 2022 - 2023 MedPy ERP. Version: 1.0.5

Fig 3-21: View all Medicine
The view for viewing all the medicine in production

Pharmaceutical ERP System

MedPy ERP
MedPy_ERP@icloud.com

- Home
- Company
- Add Medicine
- Manage Medicine
- Manage Company Account
- Manage Employee
- Generate Bill
- Customer Request

Enter Batch No.

Shelf No.
Enter Shelf No.

Expire Date
mm/dd/yyyy

Mfg Date
mm/dd/yyyy

Description.
Enter Description

In Stock Total
Enter In Stock

Qty. in Strip
Enter Description

Company
ShalimarPharma

Add Details Remove Details

Salt Name Salt Qty Salt Qty Type Description
Enter Salt name Enter Salt Qty Enter Salt Qty Type Enter Description

Add Medicine

© 2022 - 2023 MedPy ERP.
Version: 1.0.5

Fig 3-22: Add production
The view for adding a new production

Pharmaceutical ERP System

MedPy ERP
MedPy_ERP@icloud.com

- Home
- Company
- Add Medicine
- Manage Medicine
- Manage Company Account
- Manage Employee
- Generate Bill
- Customer Request

Name
Enter Company Name

License No.
Enter License No.

Address
Enter Company Address

Contact No.
Enter Contact No.

Email
Enter Company Email

Description
Enter Description

Add Company

All Companies

ID	NAME	License NO.	Address	Contact	Email	Description	Added On	Action
11	ShalimarPharma	1234	OLD OFFICER COLONY SADAR CANTT	03121212123	yameenjavaid287@gmail.com	All Pharma Products	10/19/2022, 10:54:11 PM	View

© 2022 - 2023 MedPy ERP.
Version: 1.0.5

Fig 3-23: View Sales
The sales view for managers and admin

Pharmaceutical ERP System

MedPy ERP
MedPy_ERP@ic@gmail.com

- Home
- Company
- Add Medicine
- Manage Medicine**
- Manage Company Account
- Manage Employee
- Generate Bill
- Customer Request

Enter S-GST

Batch No.
Enter Batch No.

Shelf No.
Enter Shelf No.

Expire Date
mm/dd/yyyy

Mfg Date
mm/dd/yyyy

Description.
Enter Description

In Stock Total
Enter In Stock

Qty. in Strip
Enter Description

Company
ShalimarPharma

Add Details Remove Details

Edit Medicine

© 2022 - 2023 MedPy ERP.
Version: 1.0.5

Fig 3-24: View Production
The production view for admin and production officer

View Tasks Admin: Ali Ahmad

March 2021 Sort Tasks Search Add

Task ID	Task Title	Description	Assigned Date	Assigned To	Assigned By
Open					
Open					
Open					
Open					
Open					
Open					

Delete

Fig 3-25: View Tasks
Viewing tasks for employees and managers

Assign Task

Admin: Ali Ahmad

Task ID	
Task Title	
Description	
Assigned Date	
Assigned To	
Assigned By	

Add Task →

Fig 3-26: Assign Task
The view for assigning task for admin and managers

Add Voucher

Admin: Ali Ahmad

Voucher ID	
Voucher Number	
Account Number	
Voucher Type	

Add Voucher

Fig 3-27: Add Voucher
View for adding a voucher

Add Order

Admin: Ali Ahmad

Order ID	
Supplier Ref No.	
Account Number	
Order Type	
Ordered Product	
Qty.	

Add Order

Fig 3-28: Add Order
View for adding a new order into the system

Add Chart of Account

Admin: Ali Ahmad

Chart of Account ID	
Account Title	
Account Number	
Account Type	
Account Status	

Add Account

Fig 3-29: Add COA
View for adding the chart of accounts

Pharmaceutical ERP System

MedPy ERP
MedPy_ERP@icloud.com

Home
Company
Add Medicine
Manage Medicine
Manage Company Account
Manage Employee
Generate Bill
Customer Request

MANAGE COMPANY

Add Company

Name
Enter Company Name

License No.
Enter License No.

Address
Enter Company Address

Contact No.
Enter Contact No.

Email
Enter Company Email

Description
Enter Description

Add Company

All Companies

#ID	NAME	License NO.	Address	Contact	Email	Description	Added On	Action
-----	------	-------------	---------	---------	-------	-------------	----------	--------

© 2022 - 2023 MedPy ERP.
Version: 1.0.5

Fig 3-30: Add Customer
View for adding a new customer

Add Sale

Admin: Ali Ahmad

Sale ID

Production Id

Sale Type

Sale Tax

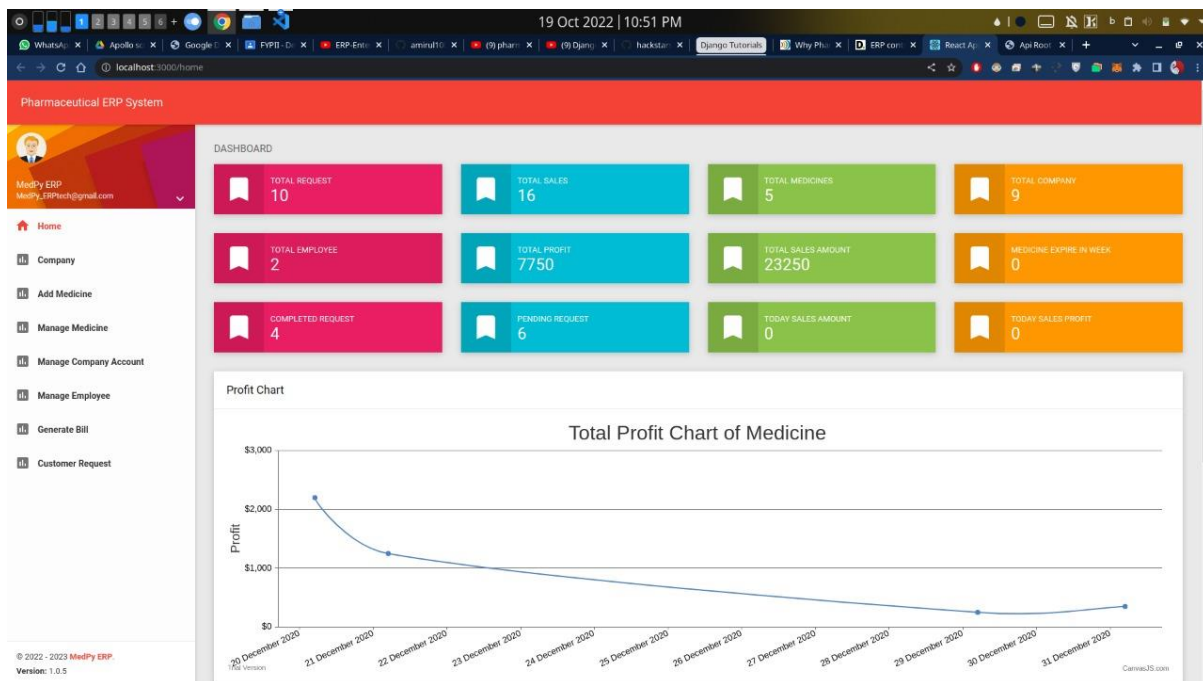
Due Date

Com.Date

Qty

Add Sale

Fig 3-31: Add Sale
View for adding a new sale to the system

**Fig 3-32: Dashboard**

View for checking various reports of the company's working

3.8 Database Design

3.8.1 ER Diagram

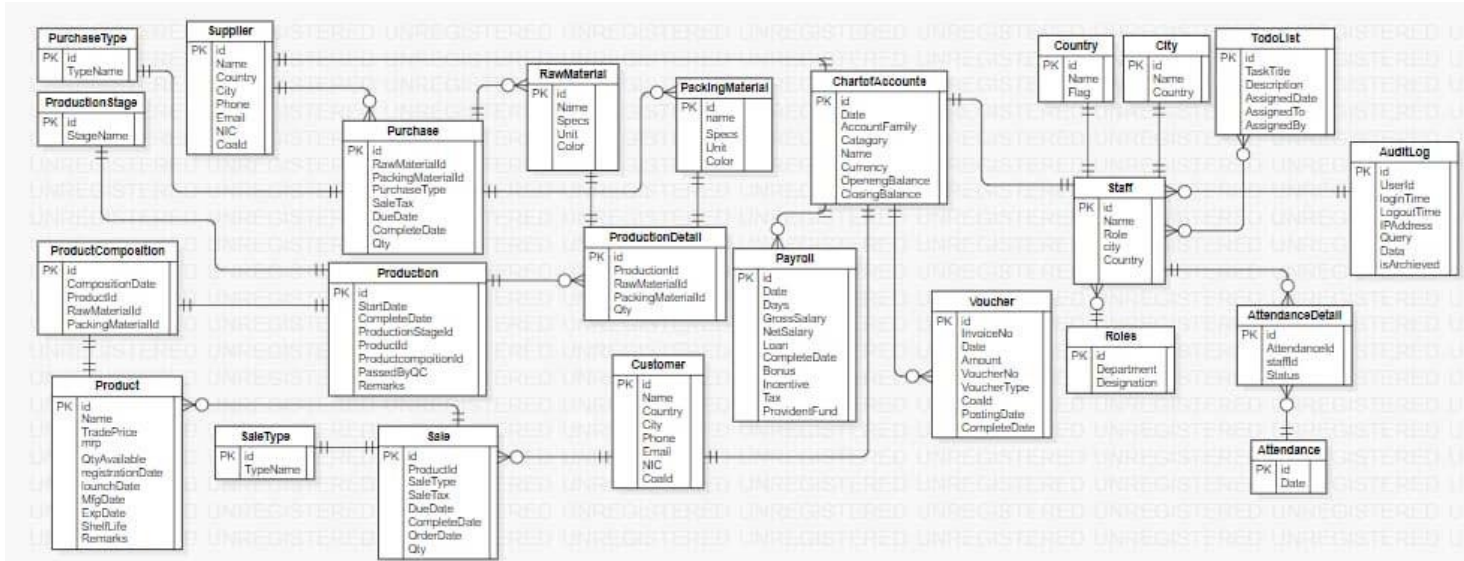


Fig 3-33: ER Diagram

The entity relationship diagram of our system that depicts our database

3.9 System Requirements

3.9.1 Hardware Requirements

These requirements are a baseline for the system to perform its core functionalities. These requirements are mentioned below:

1. The computer system should at least have a dual-core processor and 4 GBs of RAM will be well enough to perform the tasks we need.
2. The system must have a browser (one of the popular ones) to run the application.

3.9.2 Software Requirements

1. We will be using React JS for the front end of our application.
2. Moreover, we will be using Django 3.2 for our web framework.
3. For our database we will be using SQL as our language.

3.10 Design Considerations

3.10.1 Assumptions and Dependencies

- The user's system must have Windows 7 or higher, Ubuntu, MacOS
- The system should have a modern browser like Safari, Chrome, Edge, Firefox, Opera etc.
- IDE for Django application, MySQL server, ReactJS and text editors like VS Code.
- Users should be smart enough to use and interact with the application.

- Internet availability.
- Availability of online server for database and application.
- The User's device must be able to run the application.

3.10.2 General Constraints

- To provide an excellent response time, the application will be optimized.
- The browser on the target system should be either Safari, Edge, Mozilla Firefox, Chrome and Opera.
- The system must have a dual core processor, 1GB of RAM with one of the mentioned browsers installed and an internet connection enabled.
- The cache of the browser should not be altered or disabled. As it will slow down the overall performance of the website.
- The browser must support the latest CSS libraries like Bootstrap [3].
- Users should accept the cookies to get a better overview and performance.
- Users must have a web browser with cache enabled to store the temporary resources of the application.
- Our system will be following the MVT architecture [1]. The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database, the view layer will be used to interact with the user and the template will consist of the static parts of the desired HTML output.
- We will use MySQL to host our MySQL database. It will be the primary centralized database [2].
- Our system will be intelligent enough to prevent the common attacks.
- Cache should have enough memory to store the resources of the application and the memory of the system should be enough to keep the browser running.
- The performance of the application highly depends on the internet connectivity.
- Only the users who had created an account for the application will be allowed to use the application and all of its functionalities. It has a login system that authenticates the user.

3.11 Development Methods

We will be using the modular approach. There are 6 kinds of users that are sales, purchase, accounts, production and admin. All of them will have different modules. We will develop these modules in parallel to save time and resources. After completing these modules, we will test them locally. Moreover, we will be using both agile and prototyping process models. Development using modules will allow us to test the application along with the development. Every developer will write their own unit test cases to thoroughly test the system. Once the module is completed, we will use black box testing, and then top-down integration testing.

3.12 Class diagram

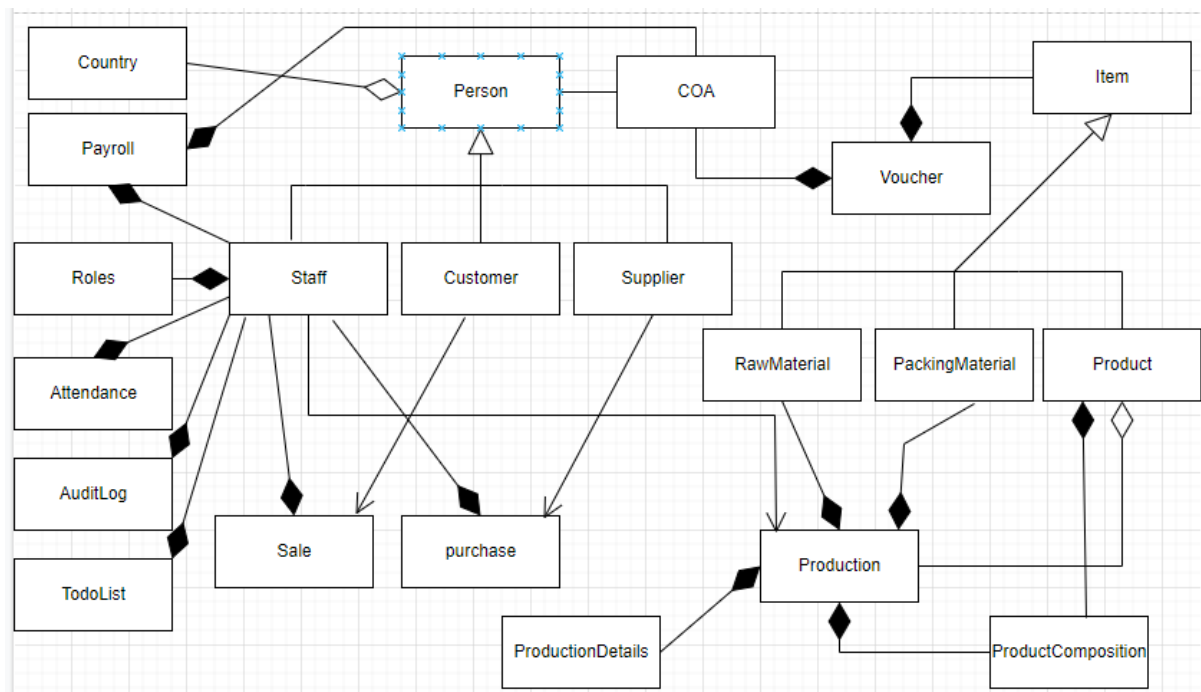


Fig 3-34: UML Class Diagram
The logical class diagram for the system

3.13 Sequence diagram

3.13.1 Add Staff

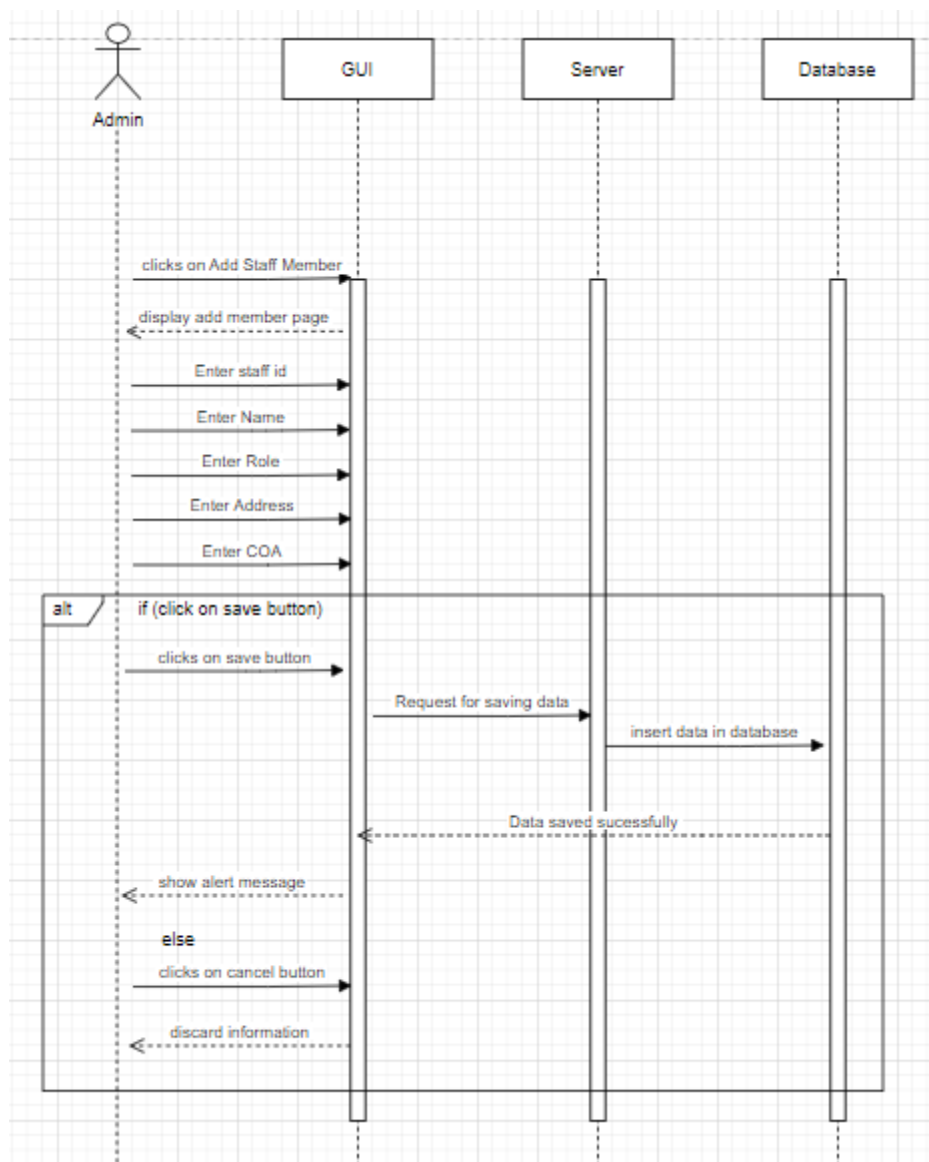


Fig 3-35: Add Staff
Sequence Diagram for add staff functionality

3.13.2 Add Supplier

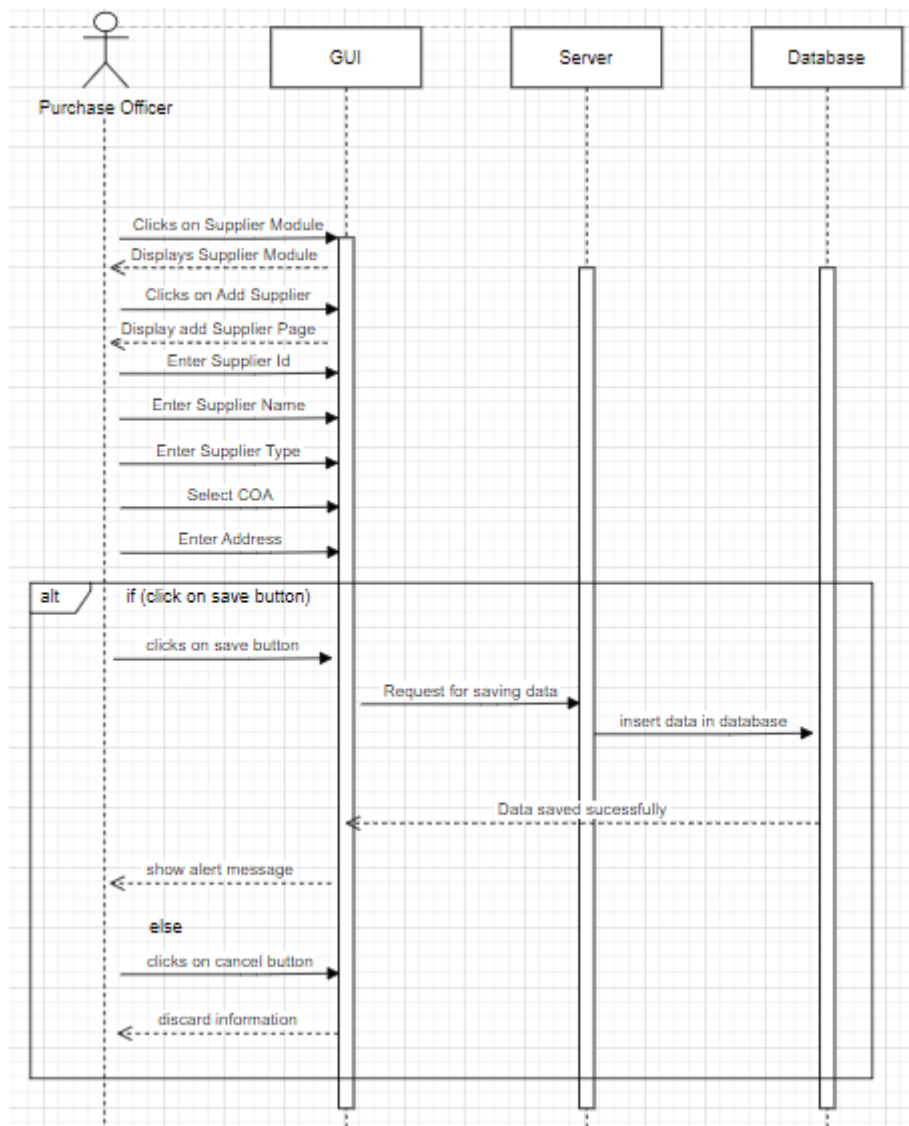


Fig 3-36: Add Supplier
Sequence for add supplier functionality

3.13.3 Add Sale

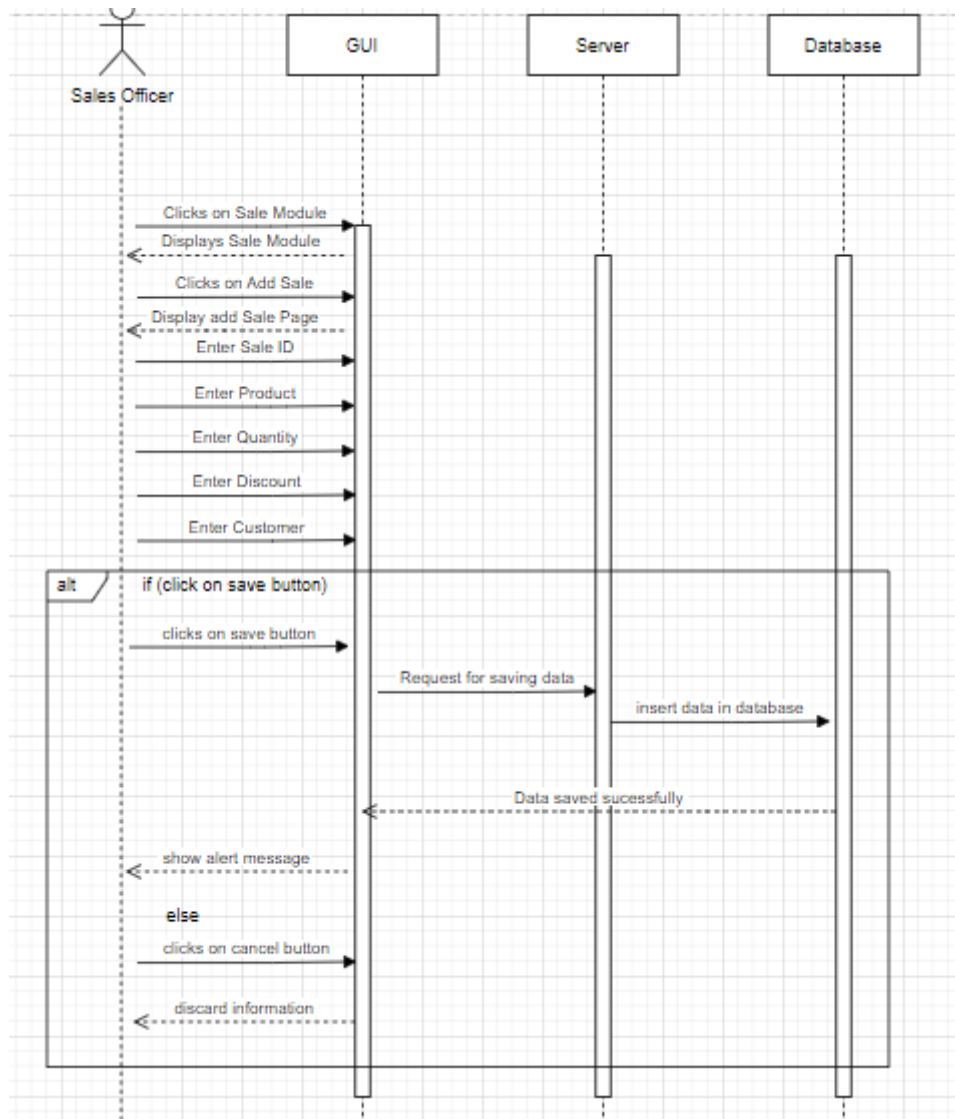


Fig 3-37: Add Sales
Add sales sequence diagram

3.13.4 Add Purchase

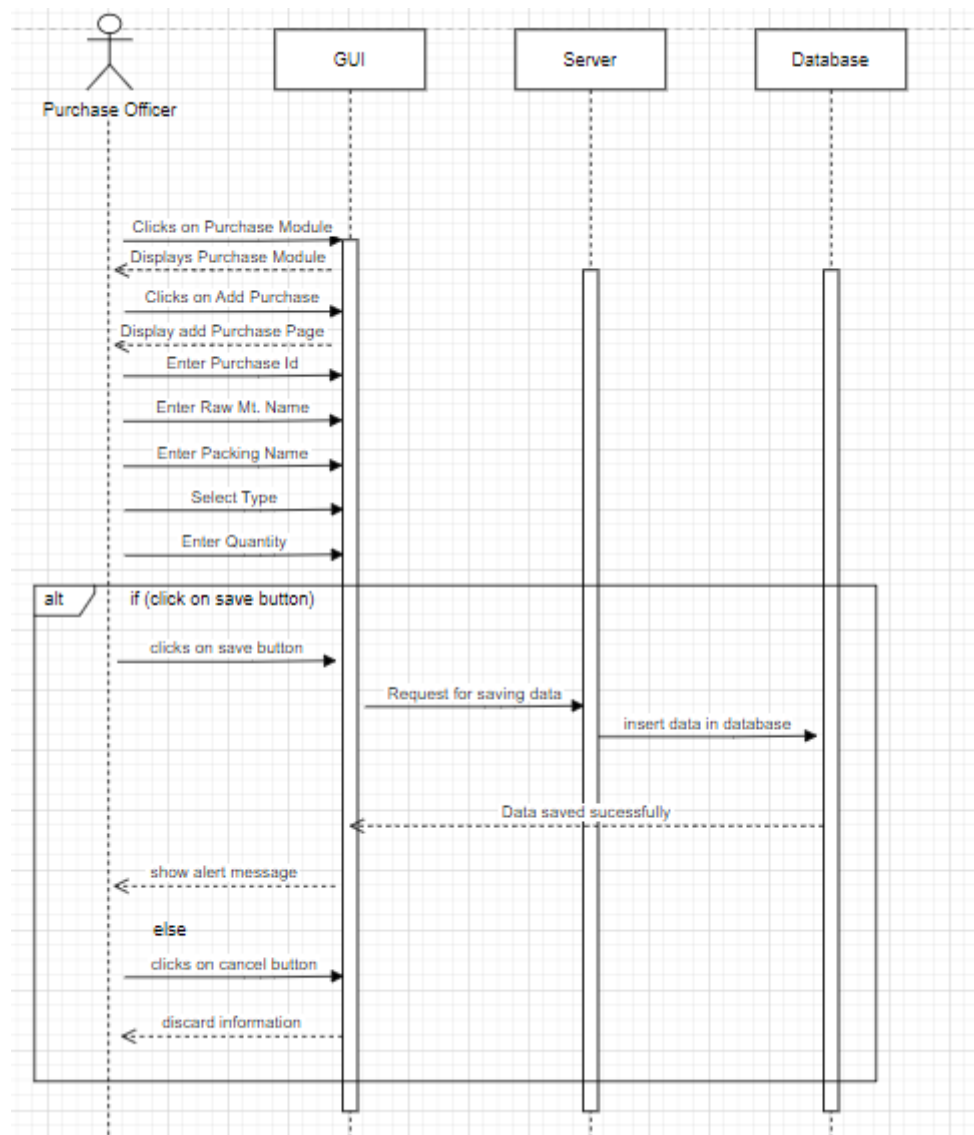


Fig 3-38: Add Purchase
Sequence Diagram for adding a purchase

3.13.5 Add COA

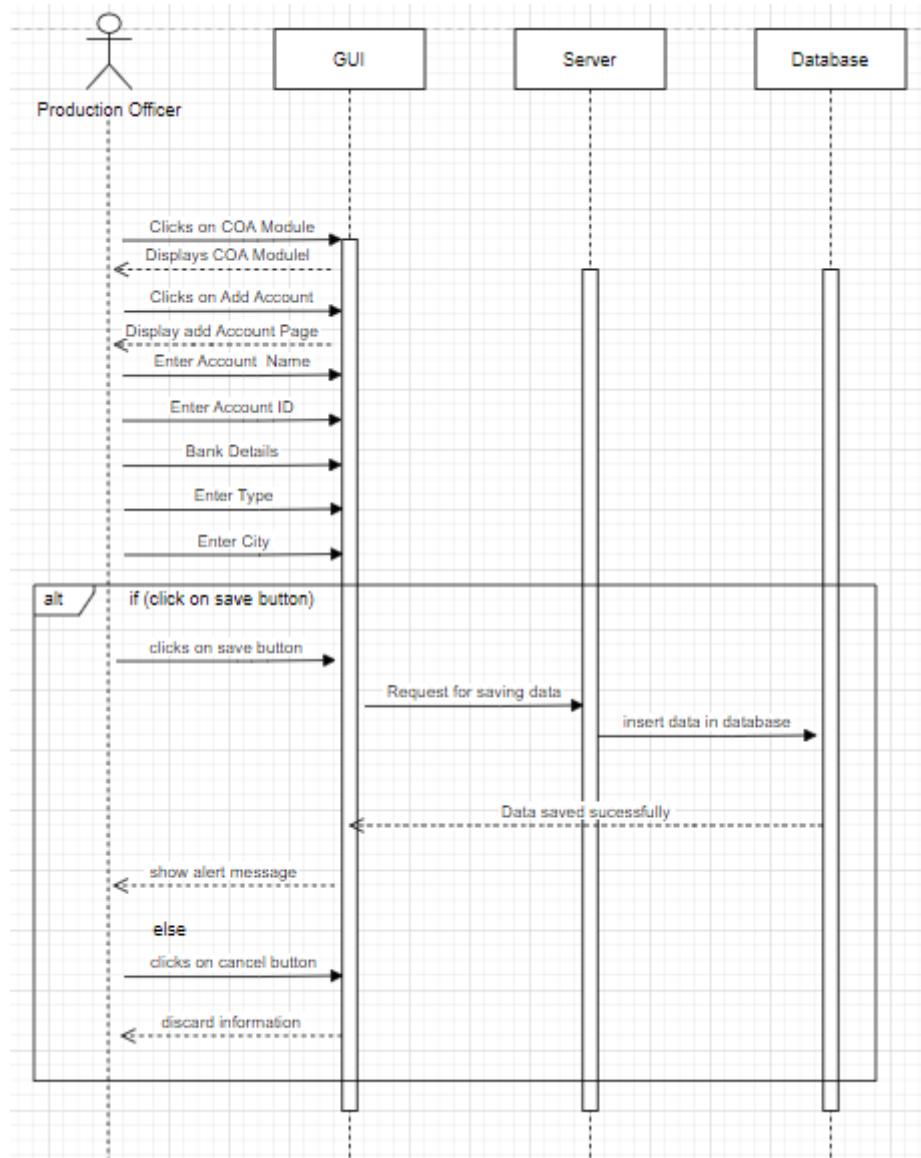


Fig 3-39: Add COA
Sequence diagram for adding a chart of accounts

3.13.6 Add Packing Material

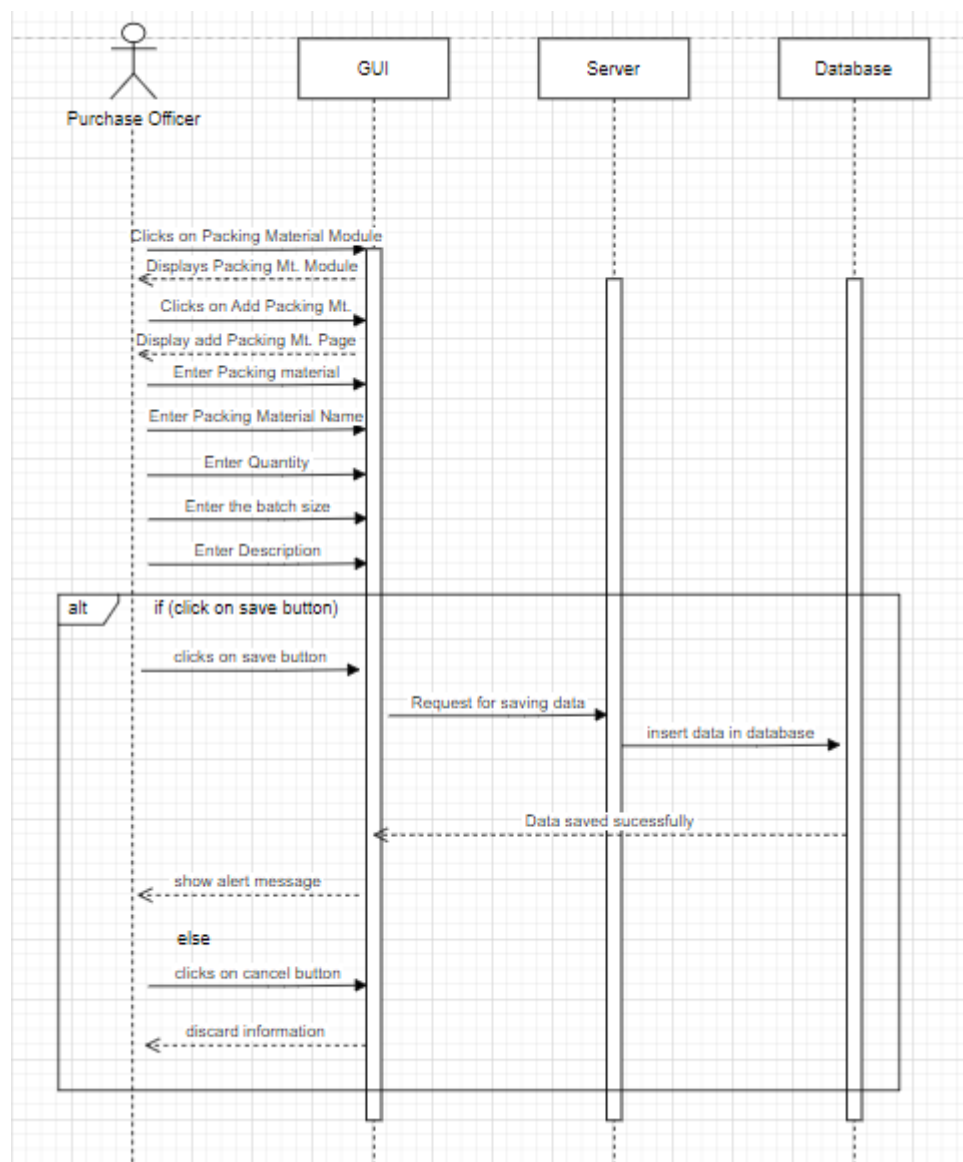


Fig 3-40: Add Purchase
Sequence diagram for adding a purchase

3.13.7 Add Payroll

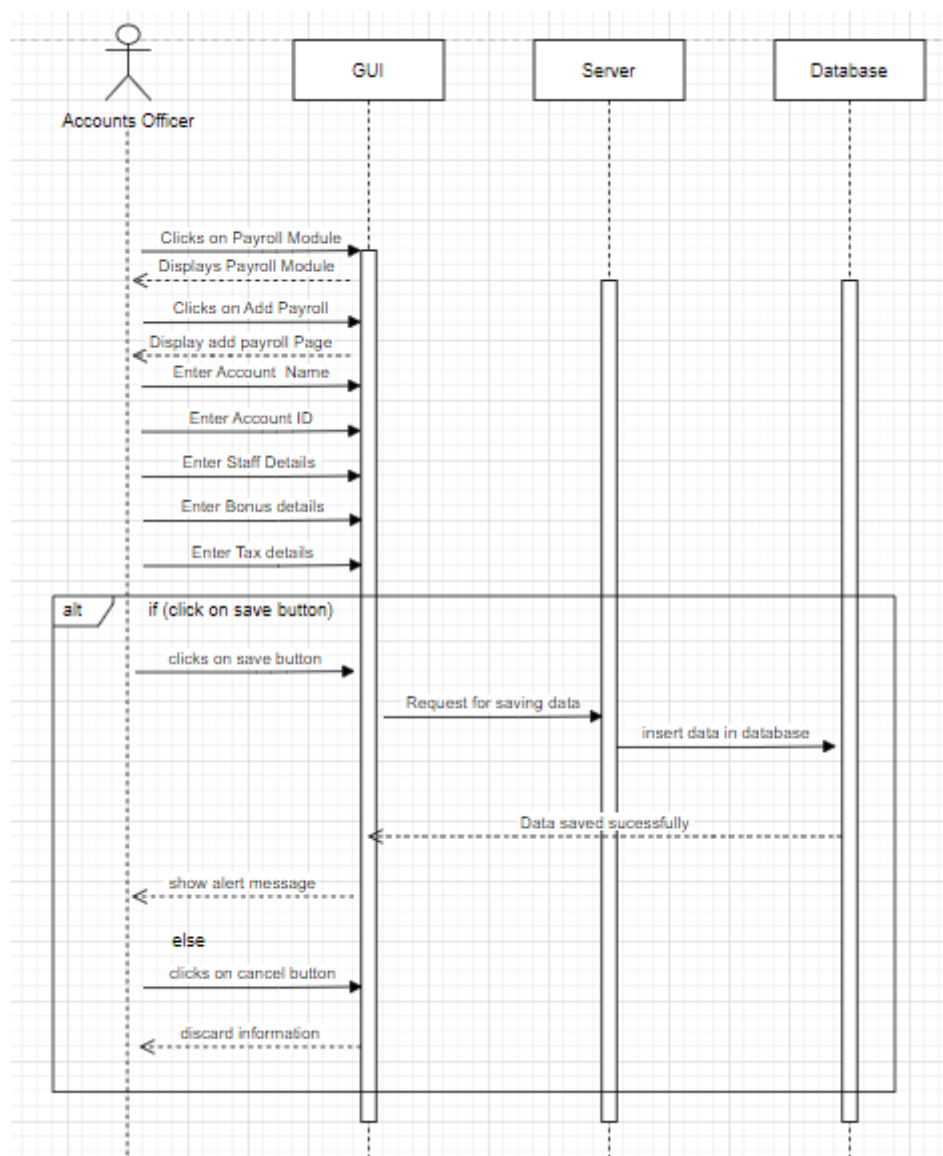


Fig 3-41: Add Payroll
Sequence diagram for adding payroll for an employee

3.13.8 Add Attendance

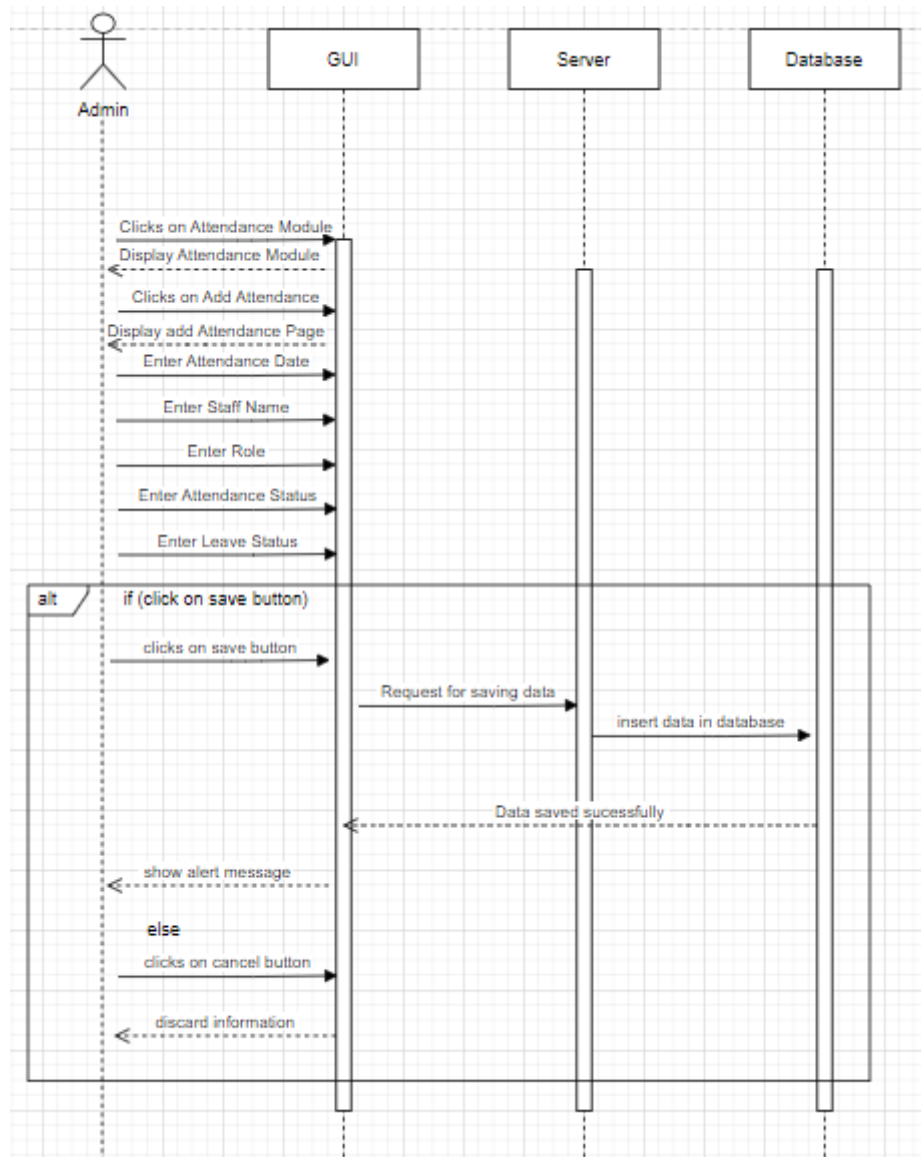


Fig 3-42: Add Attendance
Sequence diagram for adding attendance of an employee

3.13.9 Add Customer

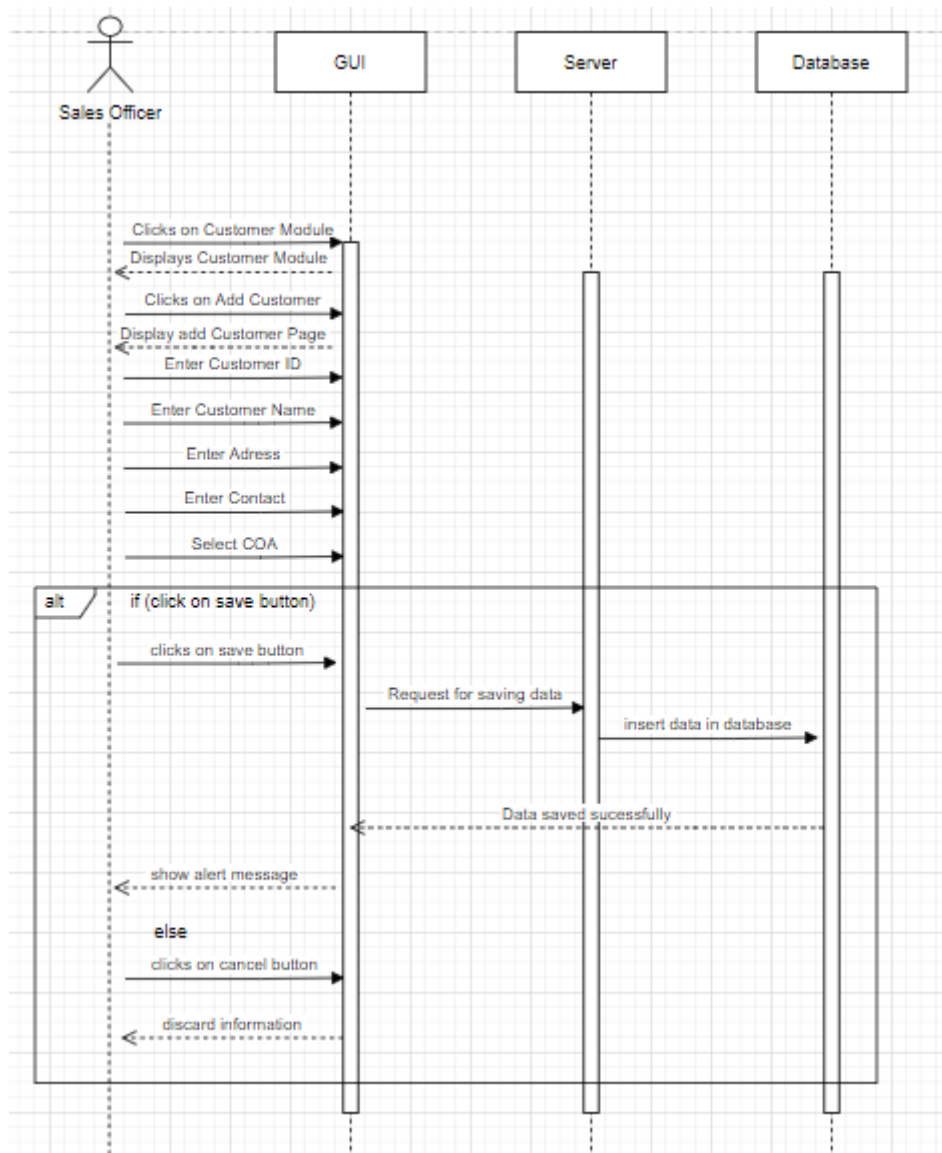


Fig 3-43: Add Customer
Sequence for adding a customer to the system

3.13.10 Add Product

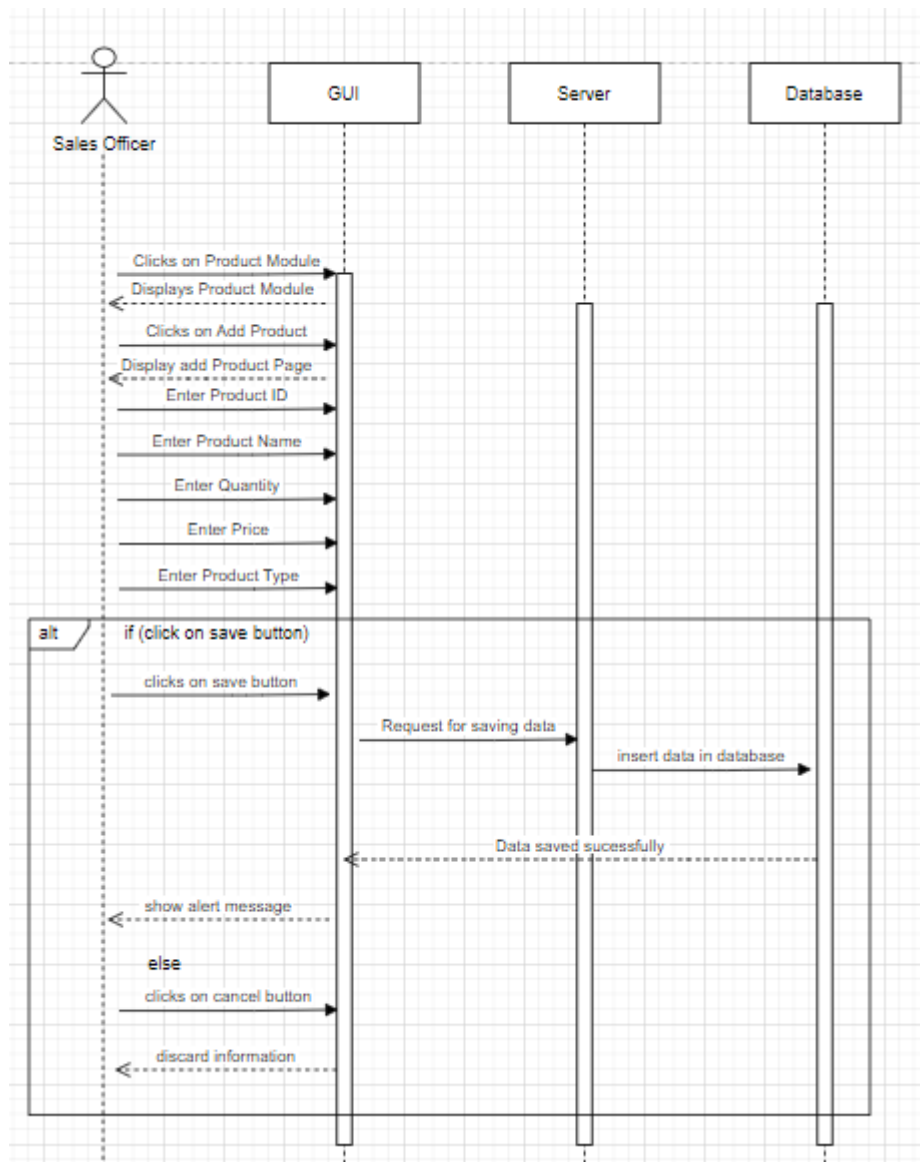


Fig 3-44: Add Product
Sequence for adding a new product

3.13.11 Add Production

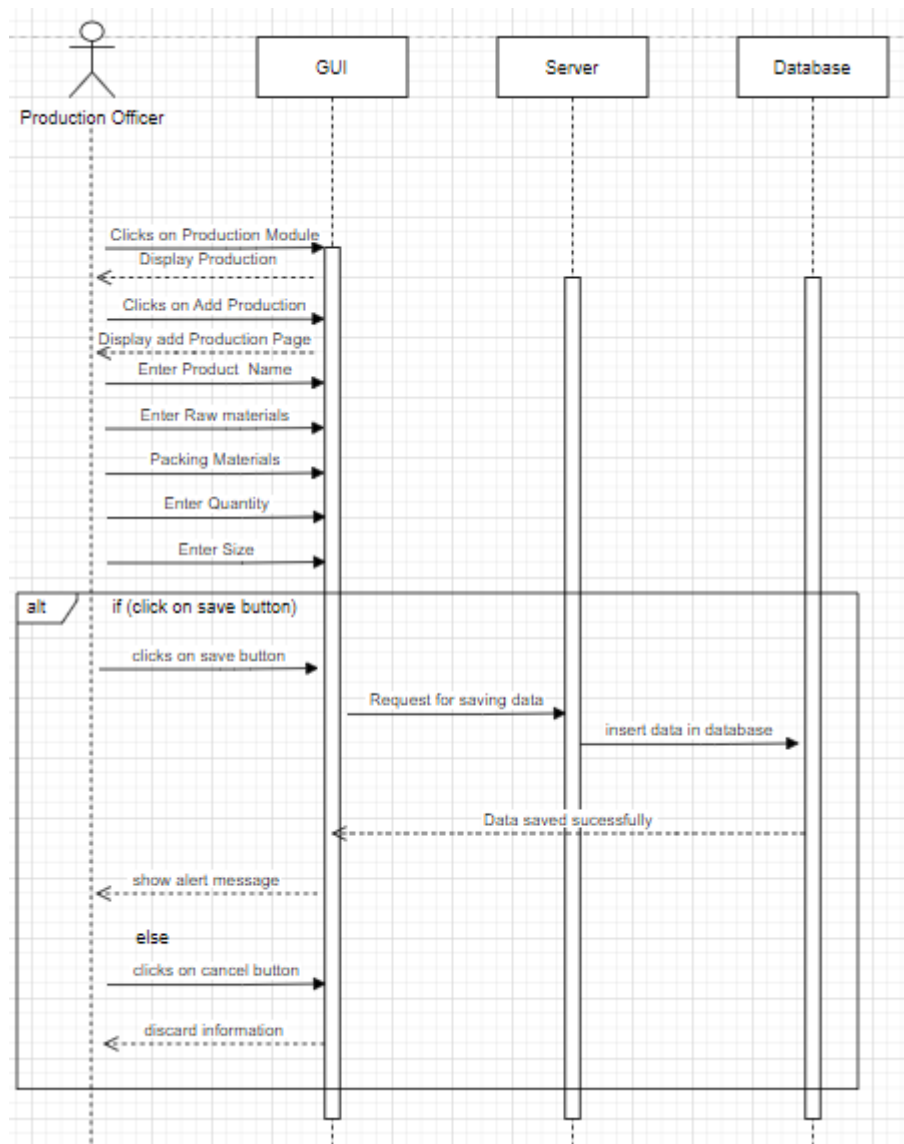


Fig 3-45: Add Production
Sequence for adding a new production

3.13.12 Add To-Do List

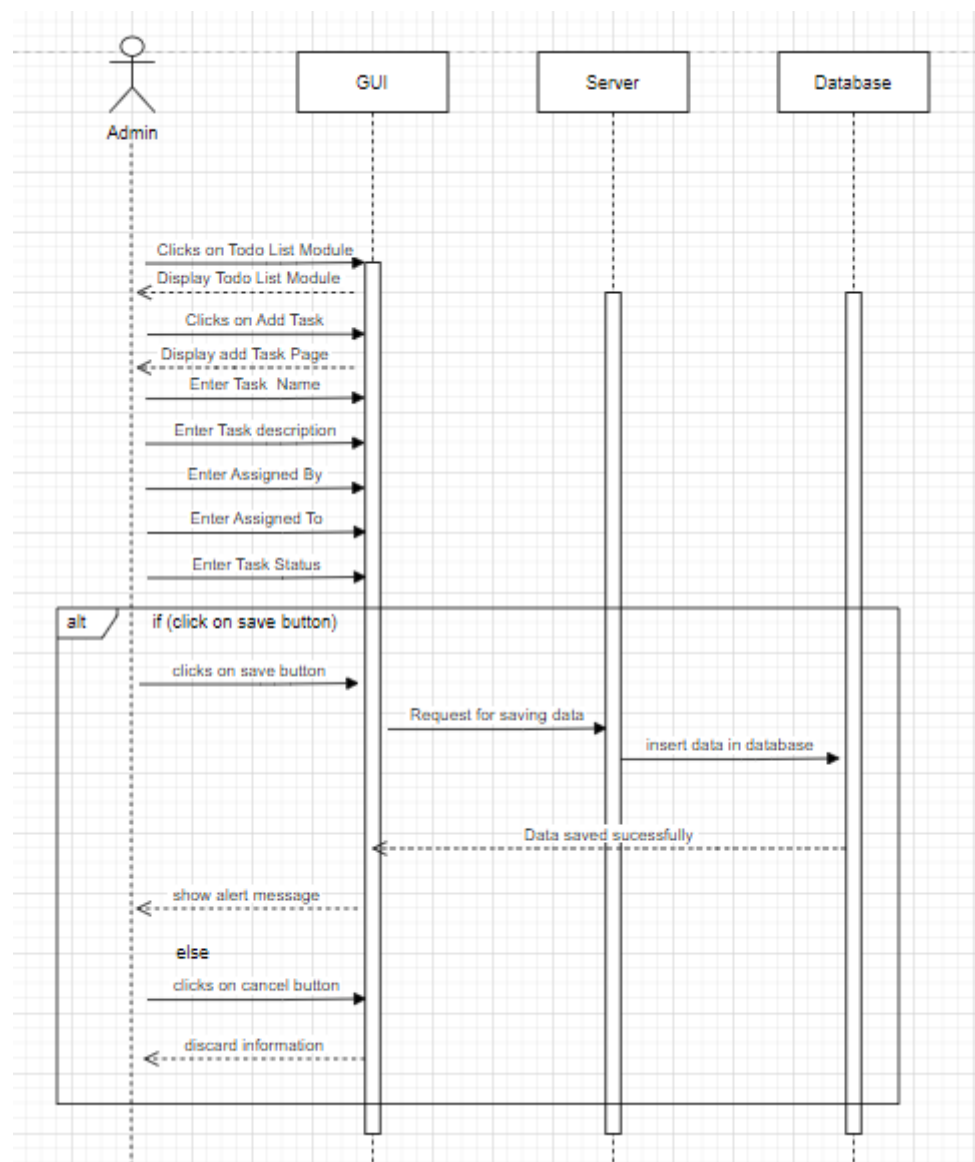


Fig 3-46: Add To-Do List
Sequence Diagram for adding to the To-Do List

3.13.13 Edit Purchase

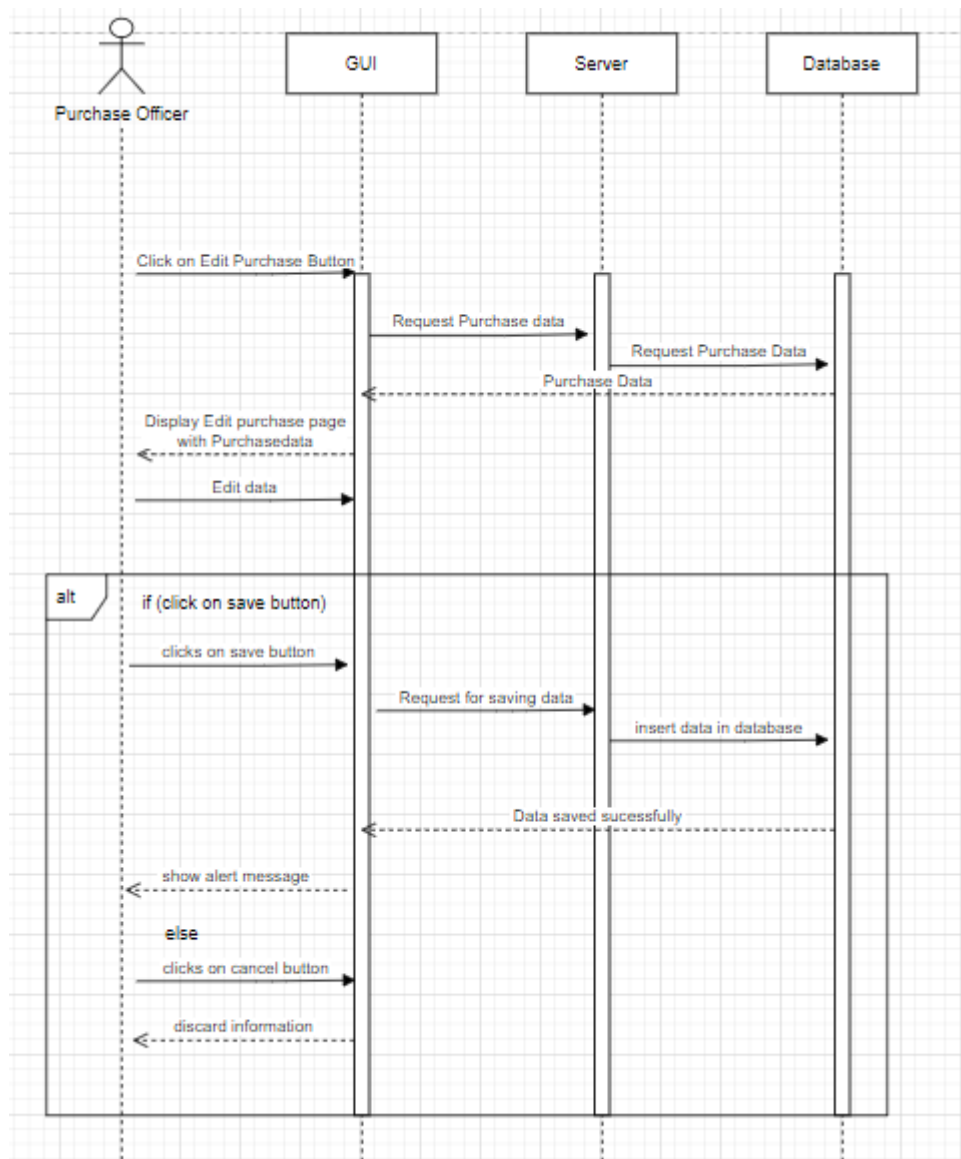


Fig 3-47: Edit Purchase
Sequence diagram for editing an already existing purchase

3.13.14 Edit Attendance

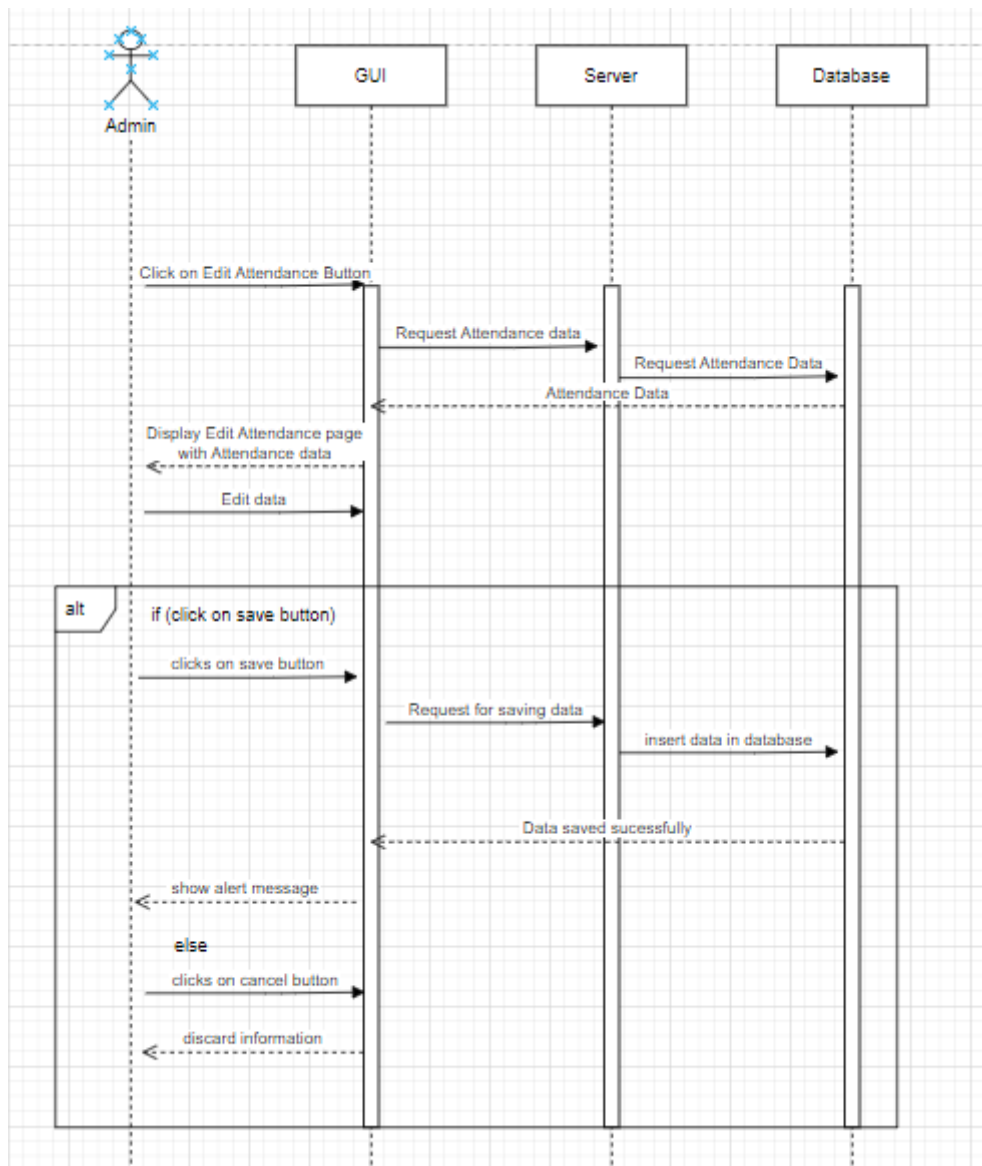


Fig 3-48: Edit Attendance
Sequence diagram for editing an employee's attendance

3.13.15 Edit Customer

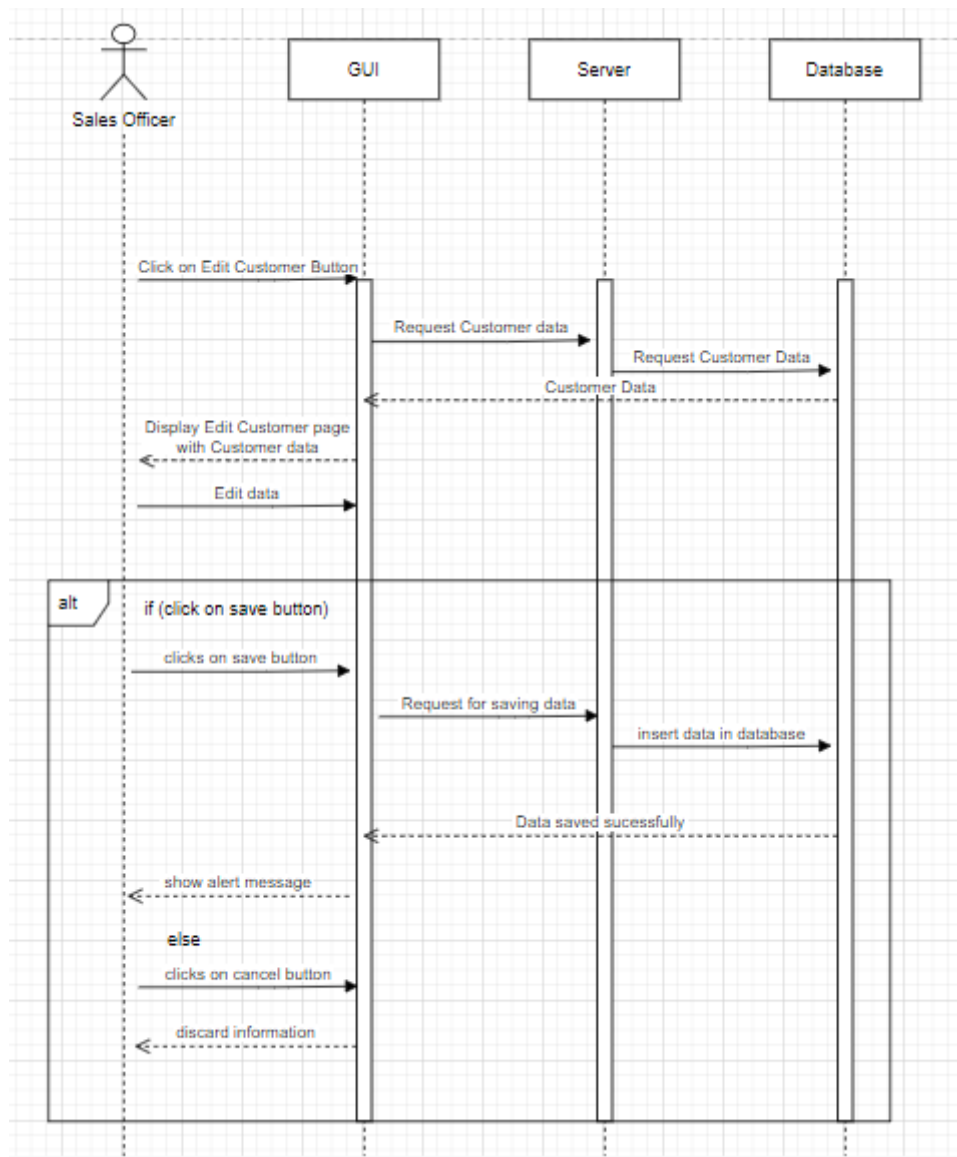


Fig 3-49: Edit Customer
Sequence diagram for editing an already existing customer

3.13.16 Edit Product

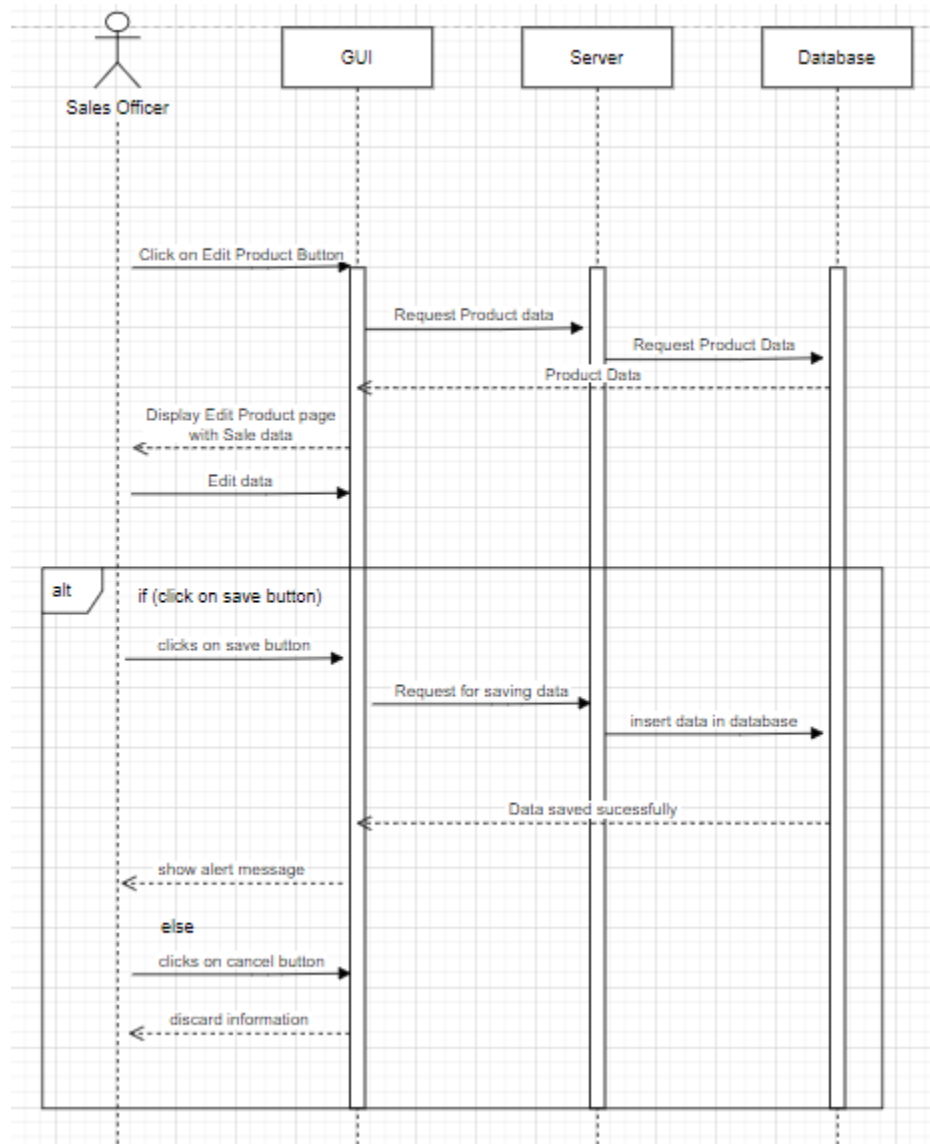


Fig 3-50: Edit Product

Sequence diagram for editing an already existing product

3.13.17 Edit Sale

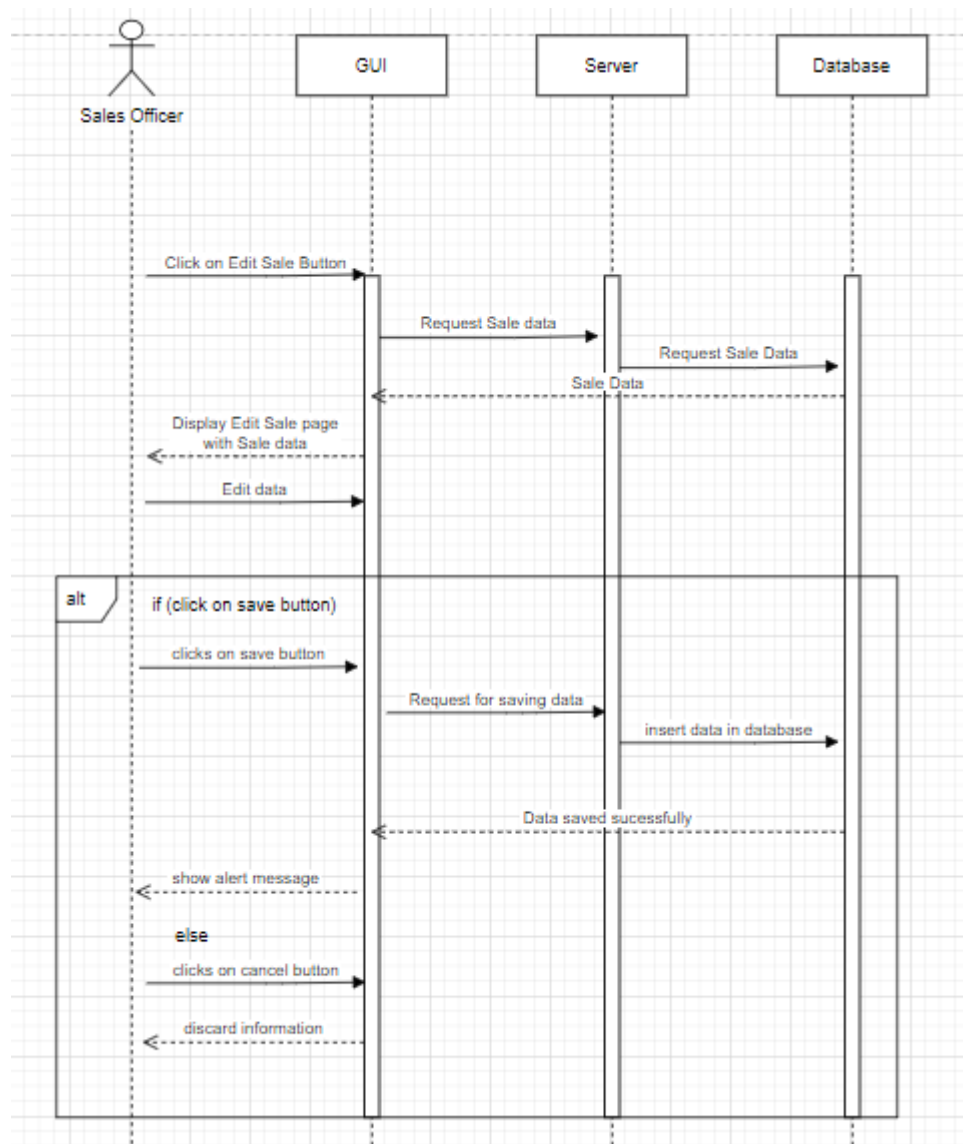


Fig 3-51: Edit Sale
Sequence diagram for editing an already existing sale

3.13.18 Edit Staff

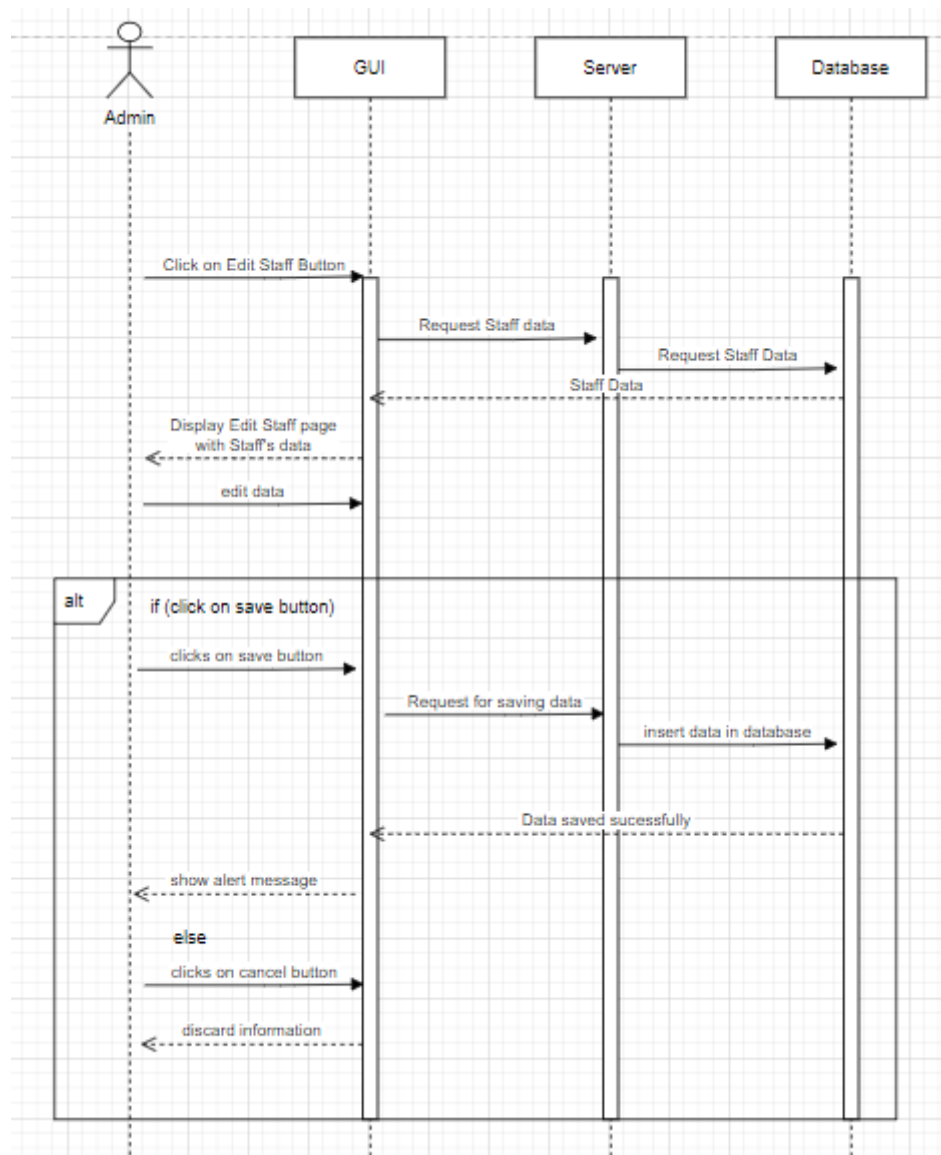


Fig 3-52: Edit Staff
Sequence Diagram for editing a staff member

3.13.19 Edit Supplier

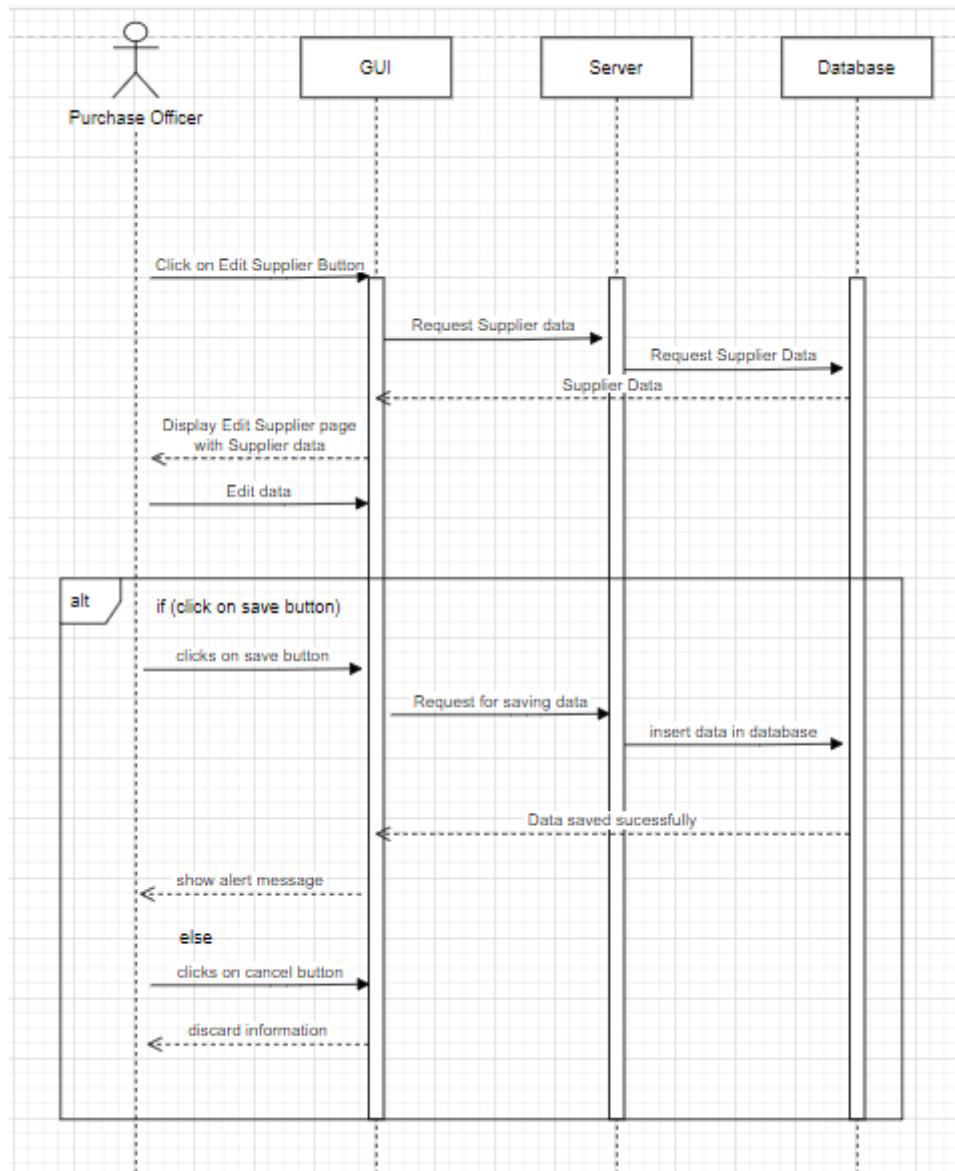


Fig 3-53: Edit Supplier
Sequence diagram for editing an already existing supplier

3.13.20 View COA

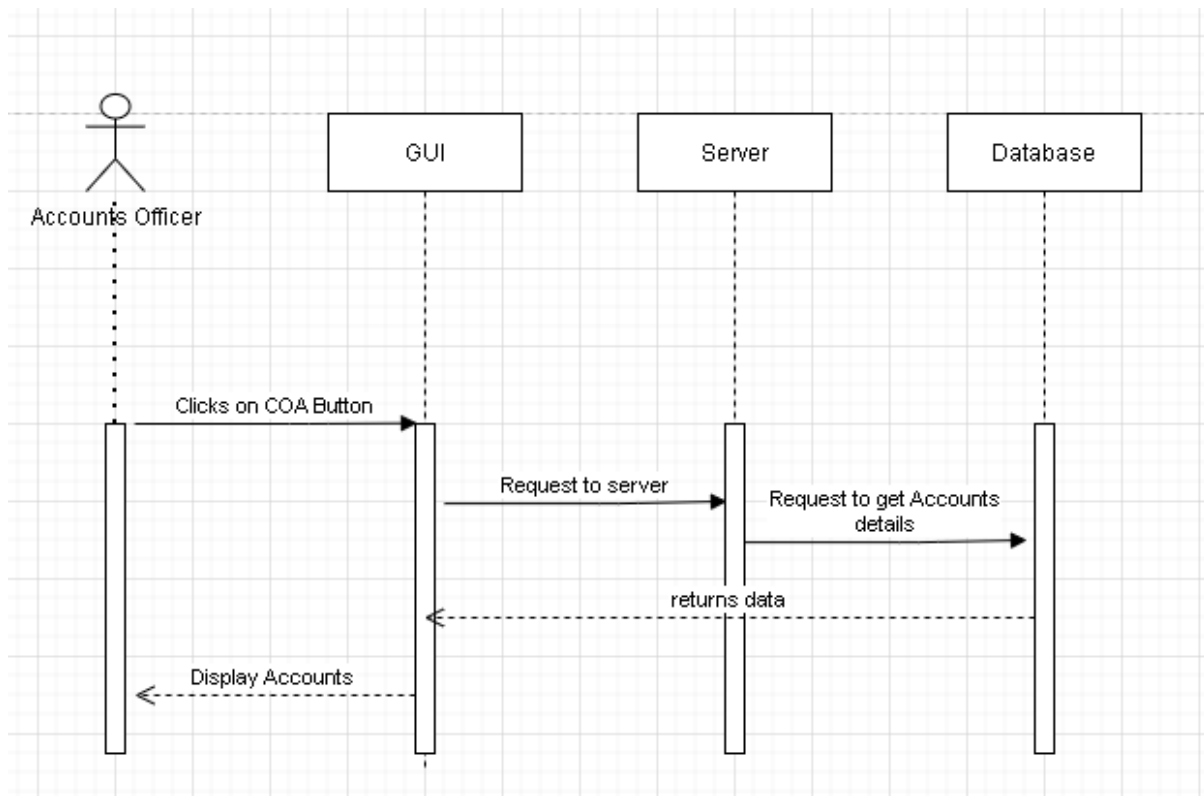


Fig 3-54: View COA
Sequence Diagram for viewing the chart of accounts

3.13.21 View Payroll

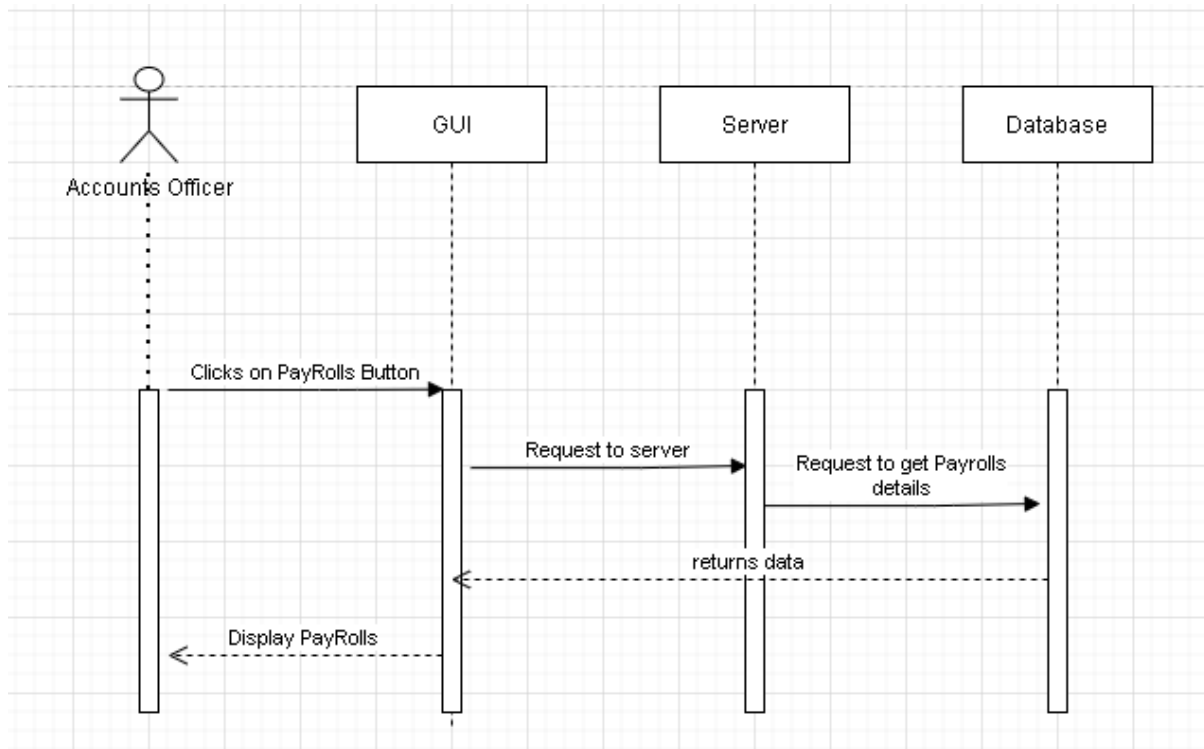


Fig 3-55: View Payroll
Sequence Diagram for viewing an employee's payroll

3.13.22 View Voucher

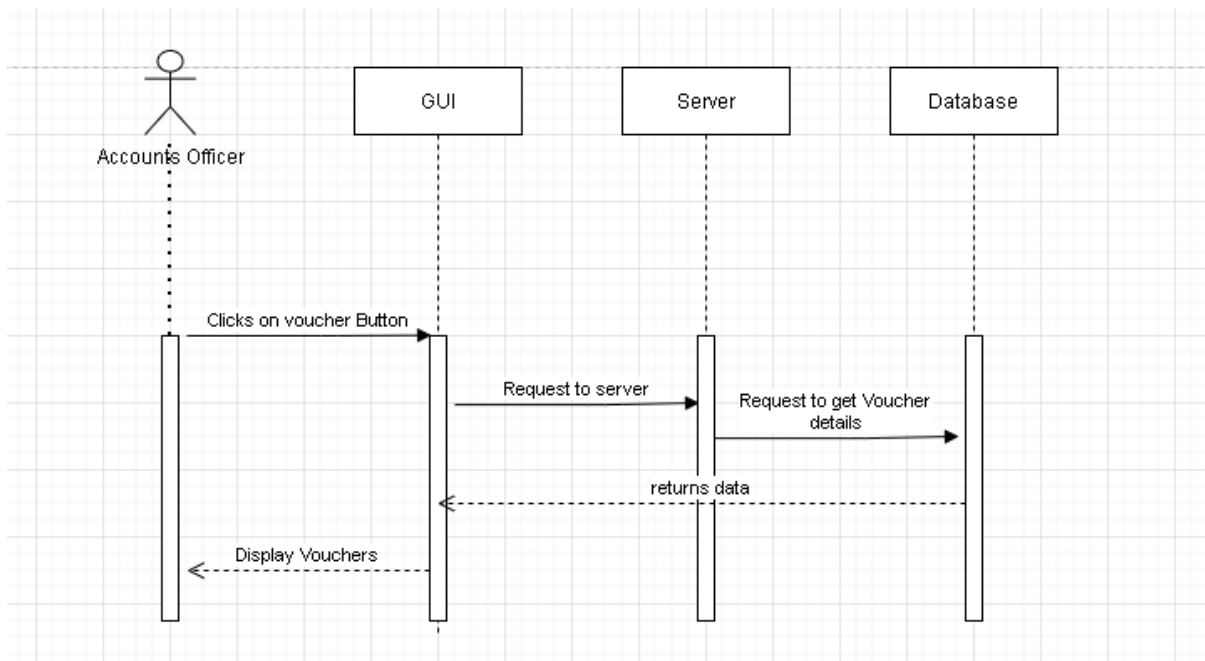


Fig 3-56: View Voucher
Sequence diagram for viewing a voucher

3.13.23 View Attendance

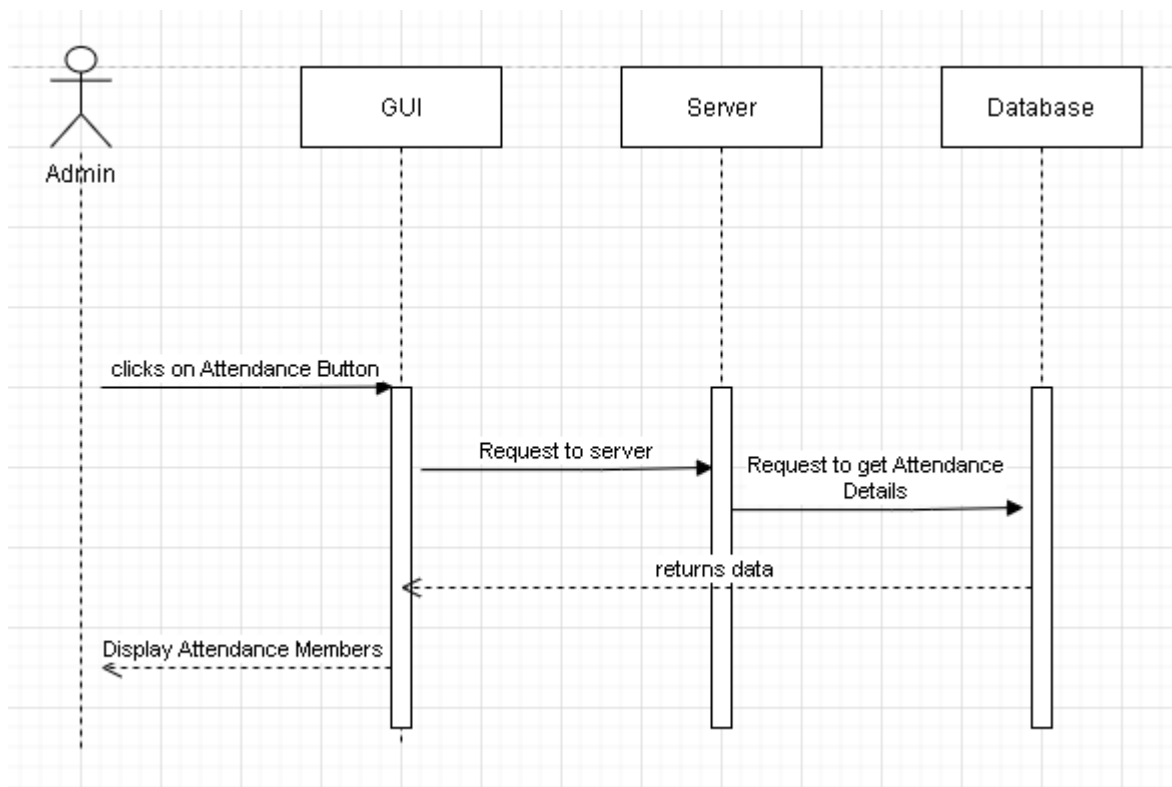


Fig 3-57: View Attendance
Sequence Diagram for viewing an employee's attendance

3.13.24 View Staff

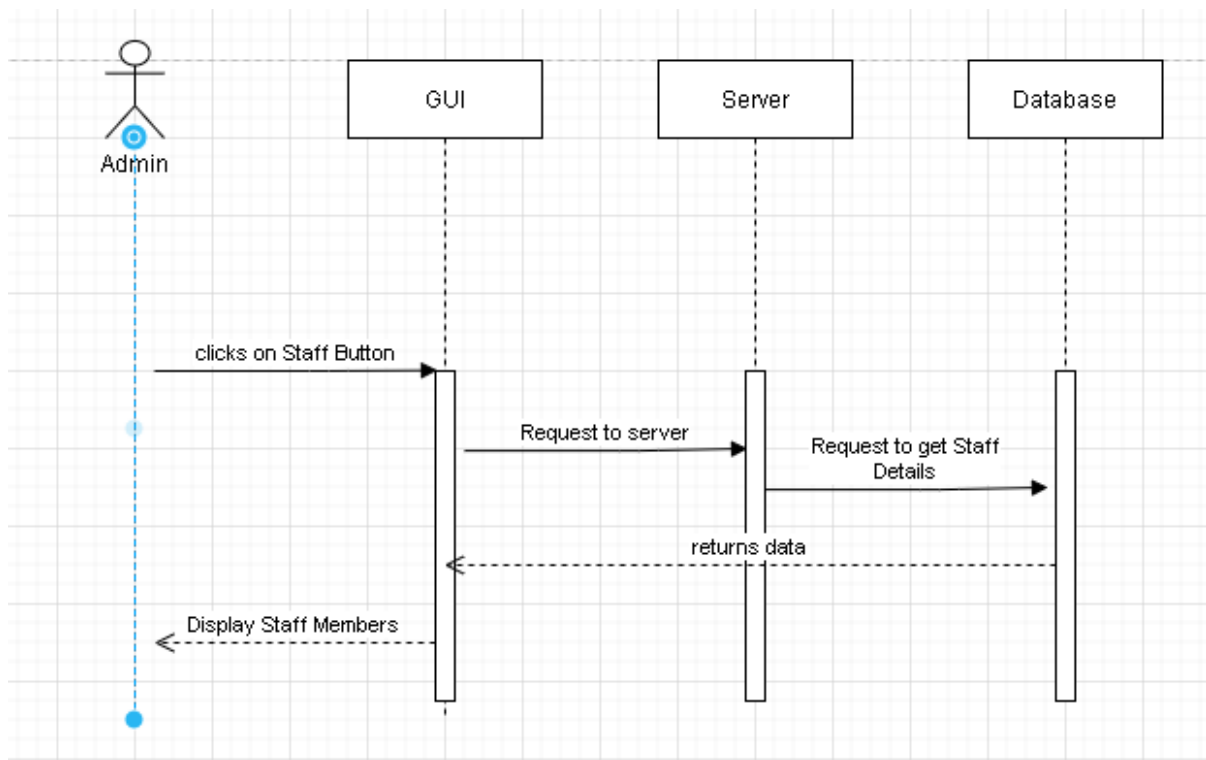


Fig 3-58: View Staff
Sequence diagram for viewing the available staff's information

3.13.25 View To-Do List

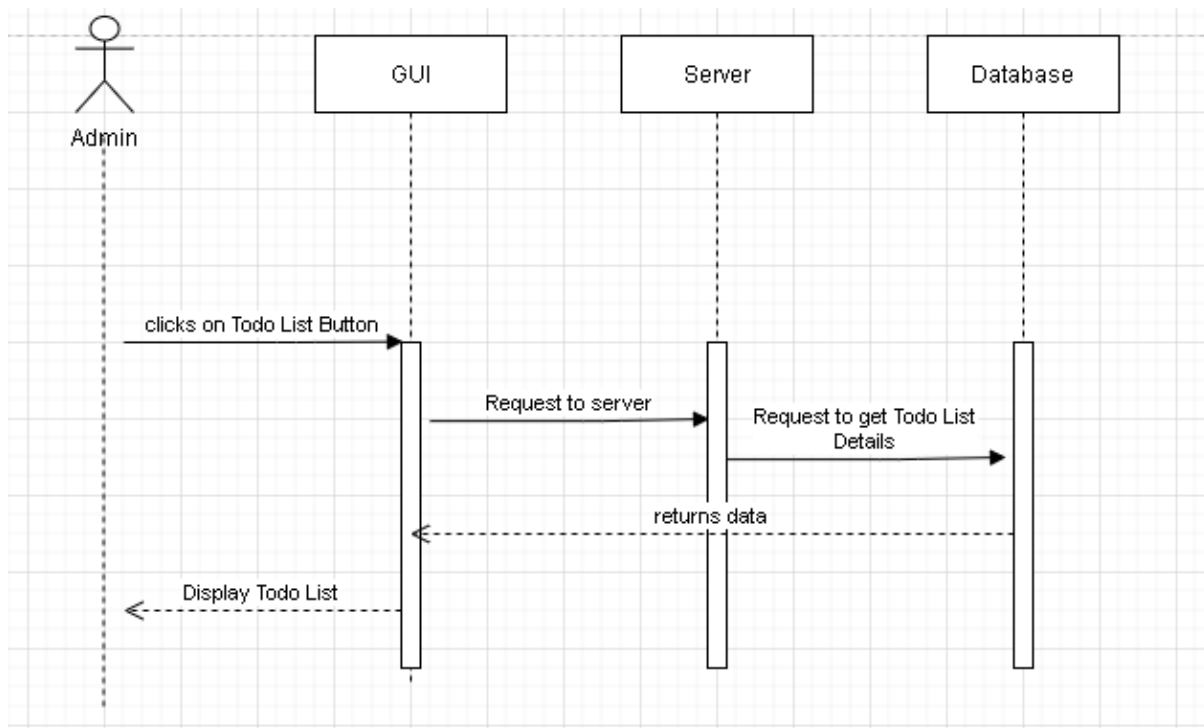


Fig 3-59: View To-Do List
Sequence diagram for viewing the To-Do List

3.13.26 View Production

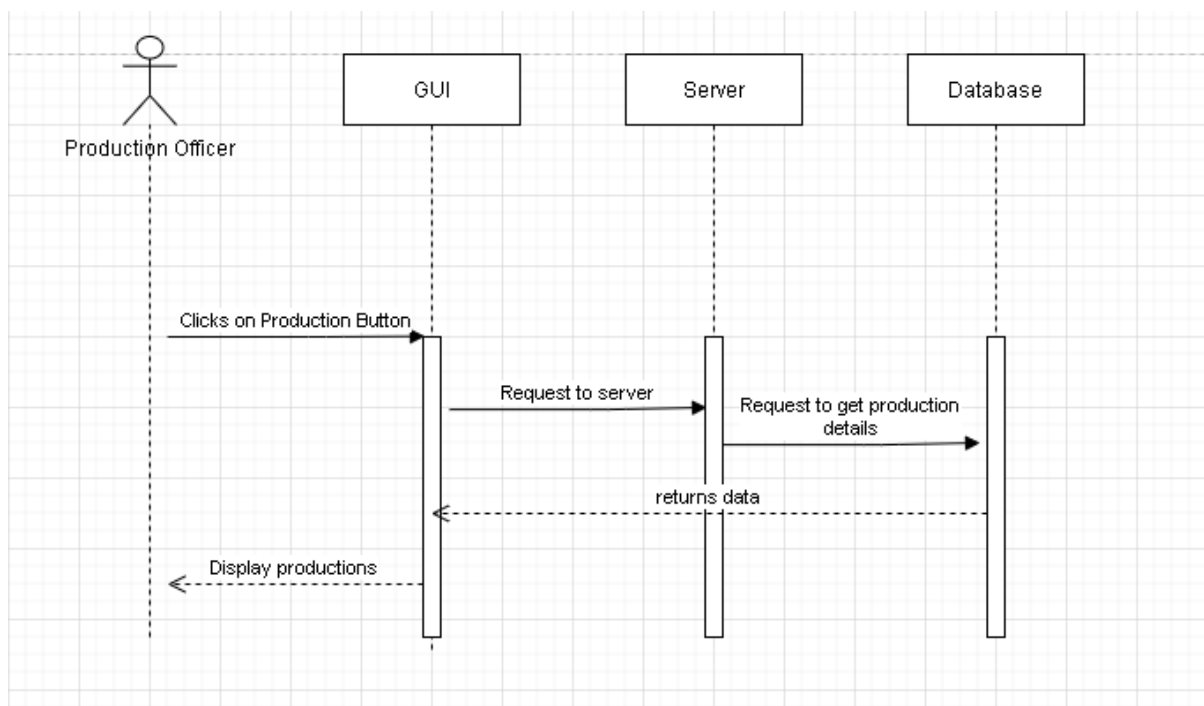


Fig 3-60: View Production
Sequence Diagram for viewing an ongoing or completed production

3.13.27 View Purchase

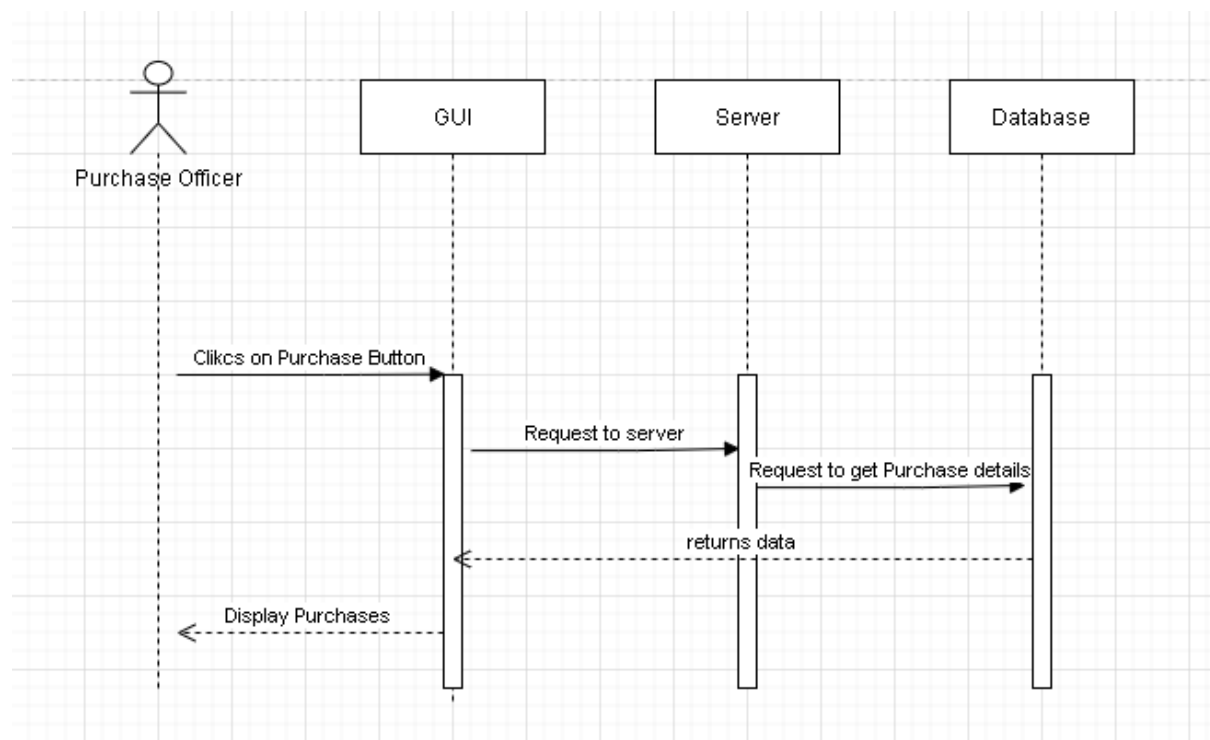


Fig 3-61: View Purchase
Sequence diagram for viewing an already existing purchase

3.13.28 View Supplier

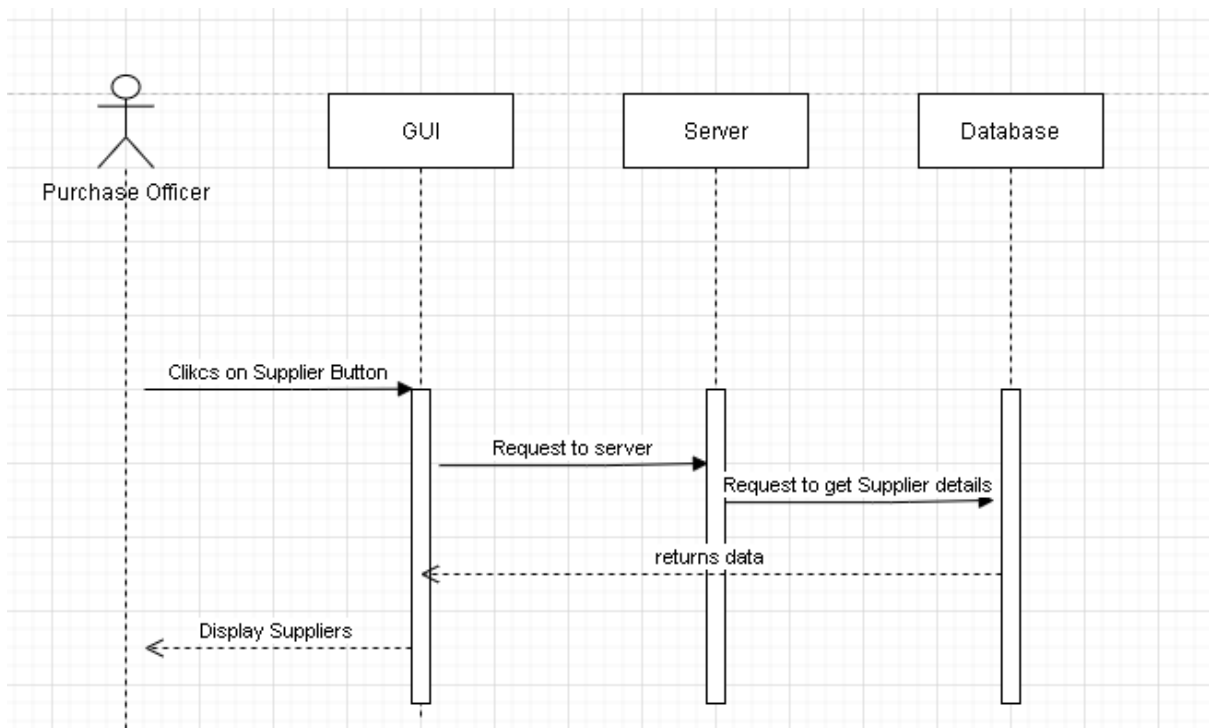


Fig 3-62: View Supplier

Sequence diagram for viewing an already existing supplier

3.13.29 View Customer

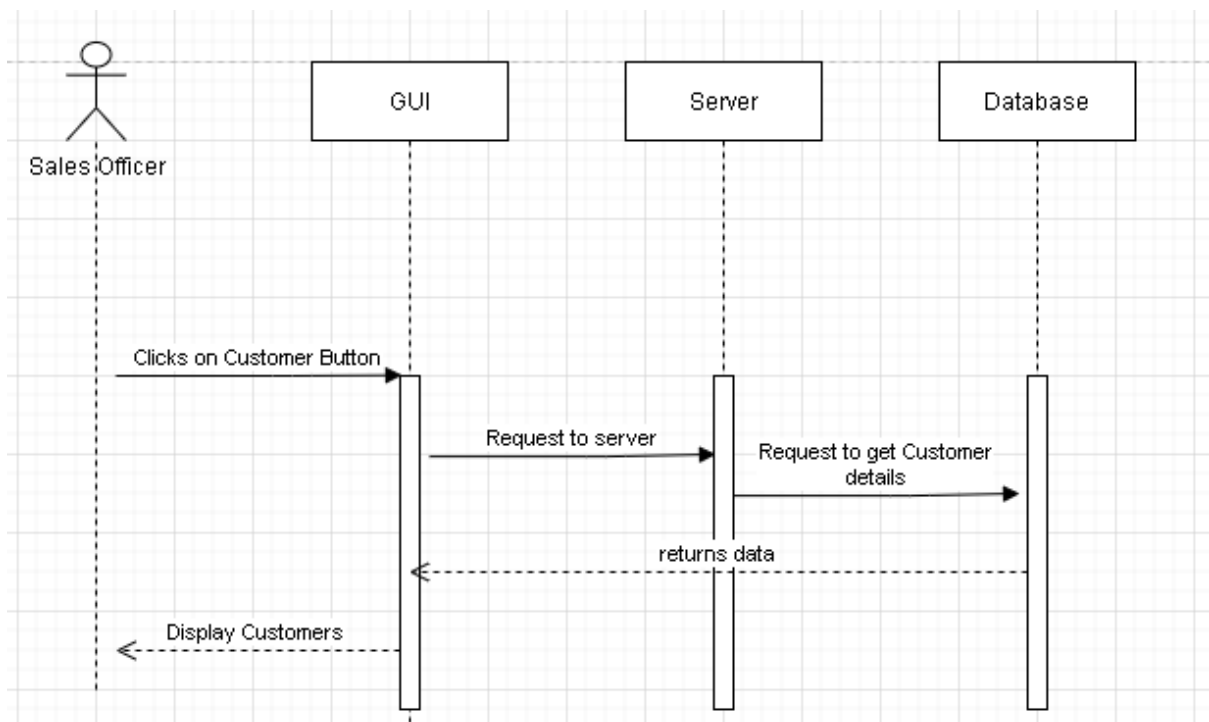


Fig 3-63: View Customer

Sequence diagram for viewing a customer's information

3.13.30 View Sales

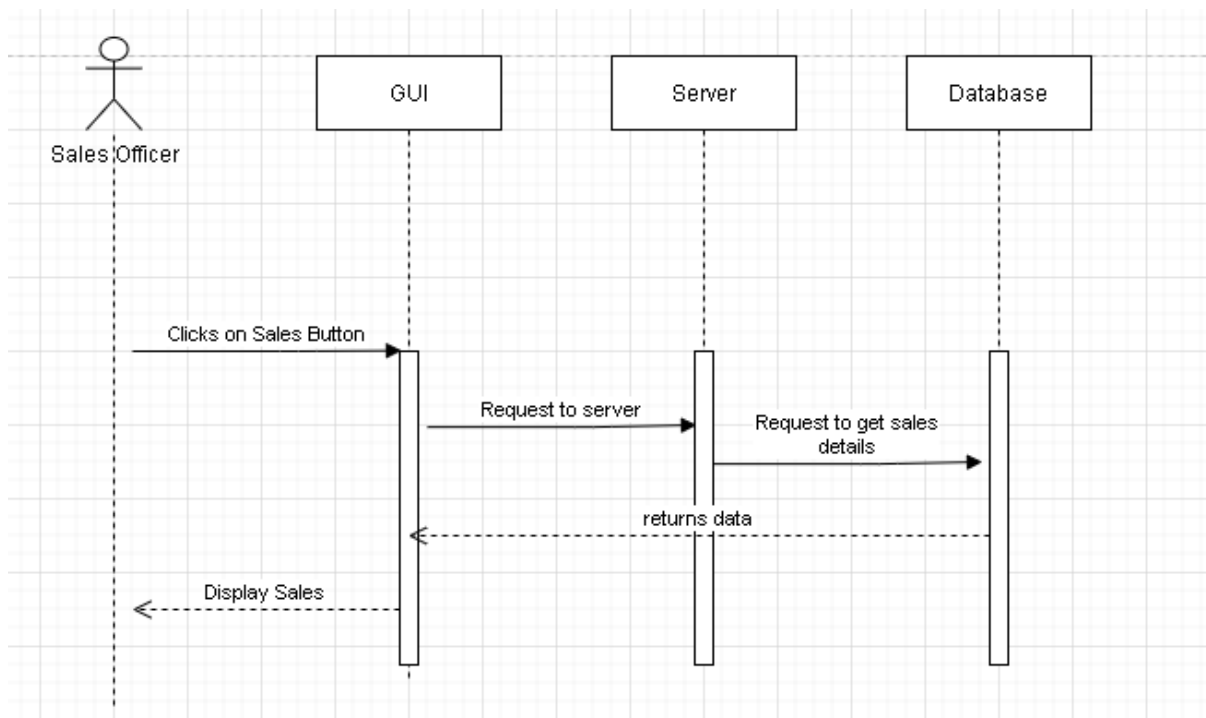


Fig 3-64 : View Sales
Sequence diagram for viewing already made sales

3.13.31 Delete Staff

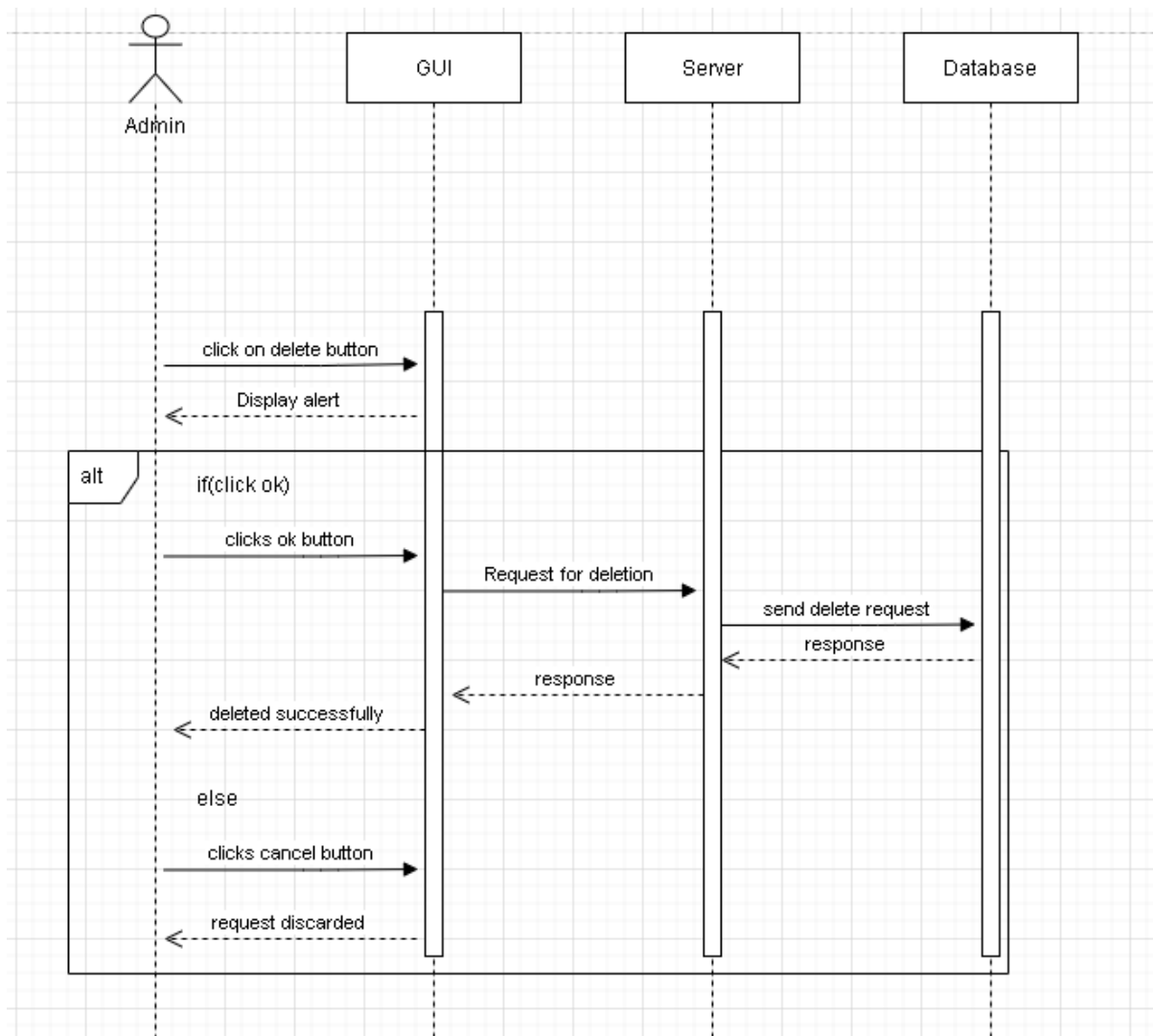


Fig 3-65: Delete Staff
Sequence diagram for deleting a staff member

3.13.32 Delete Attendance

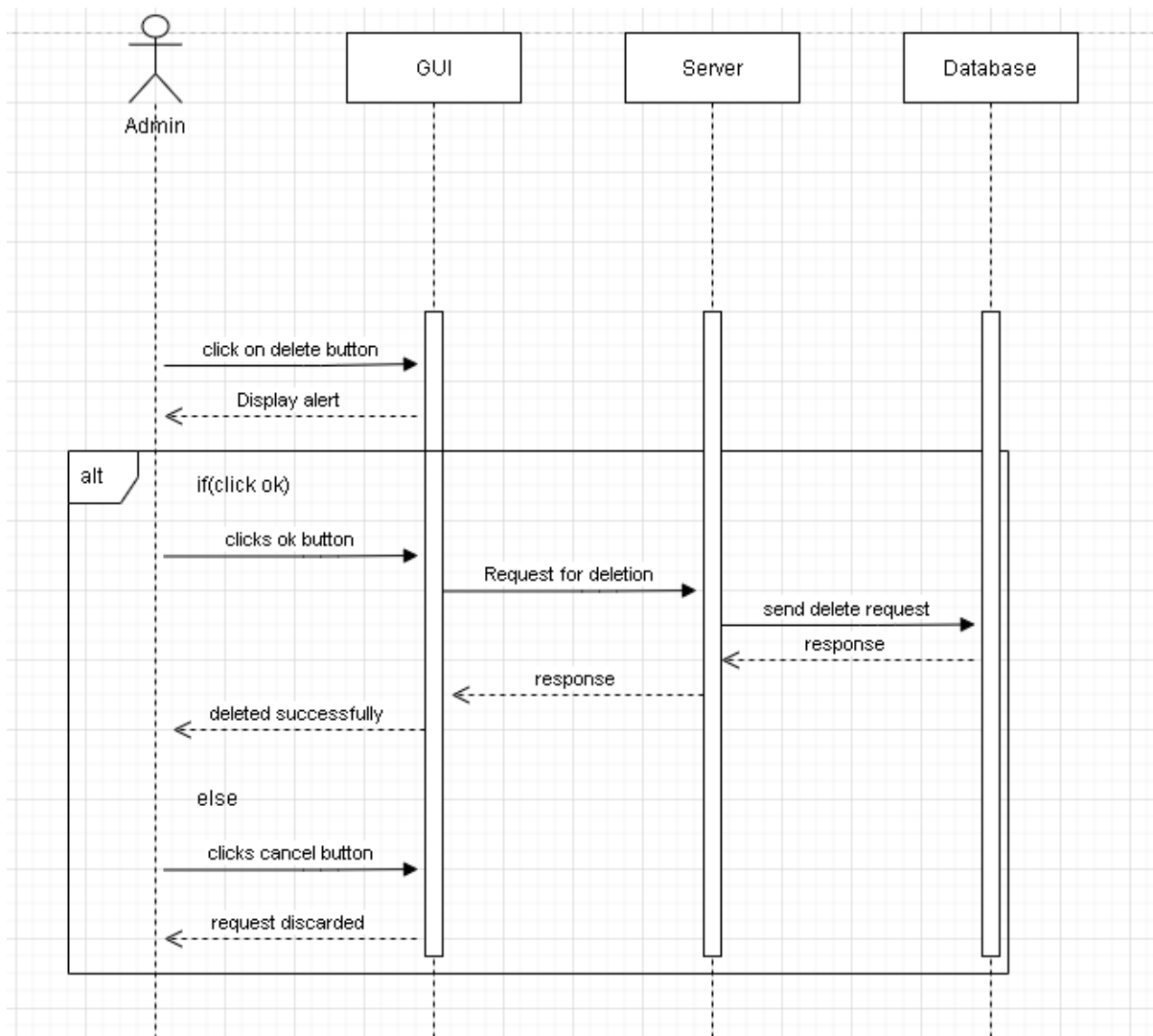


Fig 3-66: Delete Attendance

Sequence diagram for deleting an employee's attendance

3.13.33 Delete Production

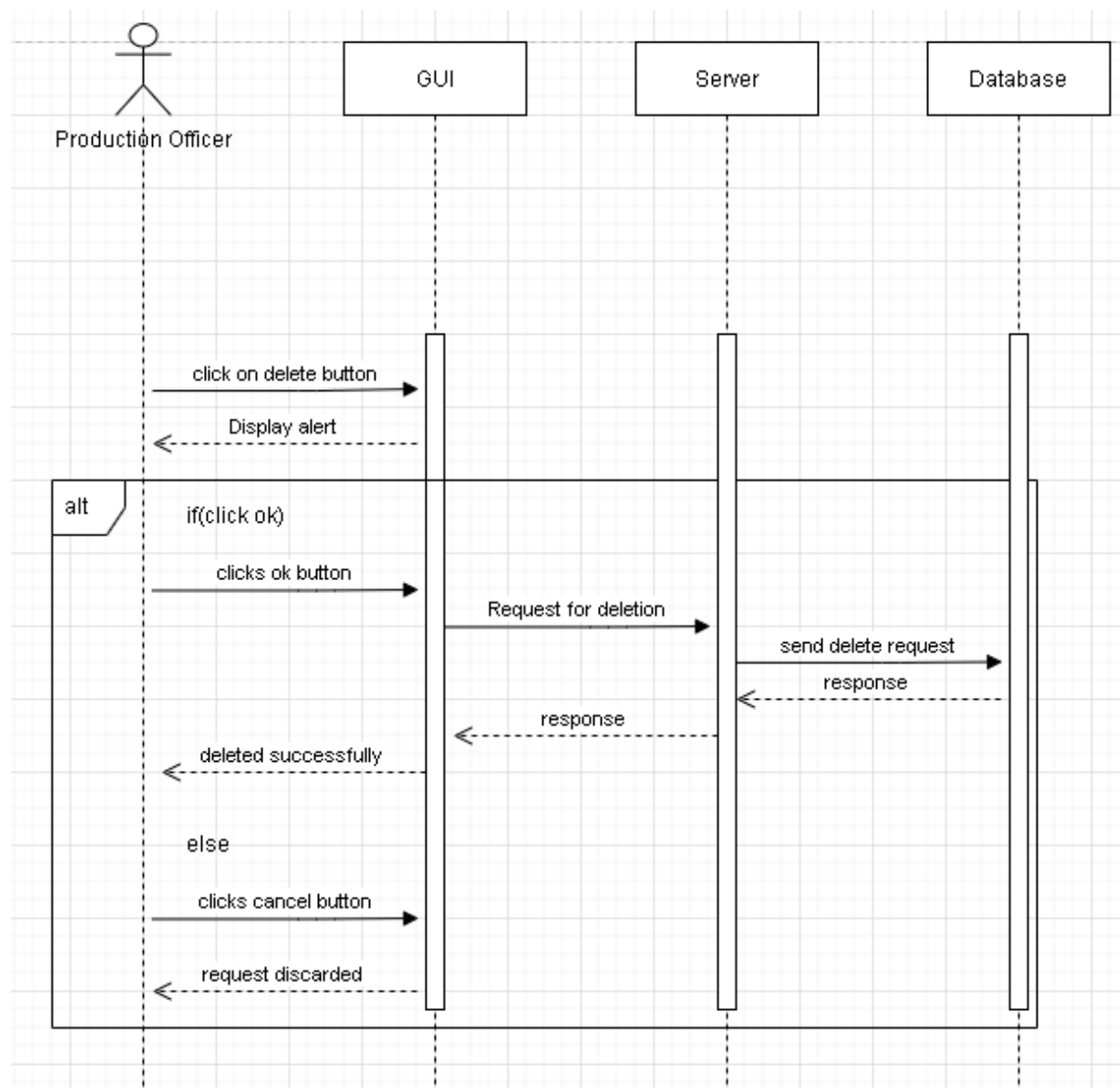


Fig 3-67: Delete Production
Sequence diagram for deleting an existing production

3.13.34 Delete Customer

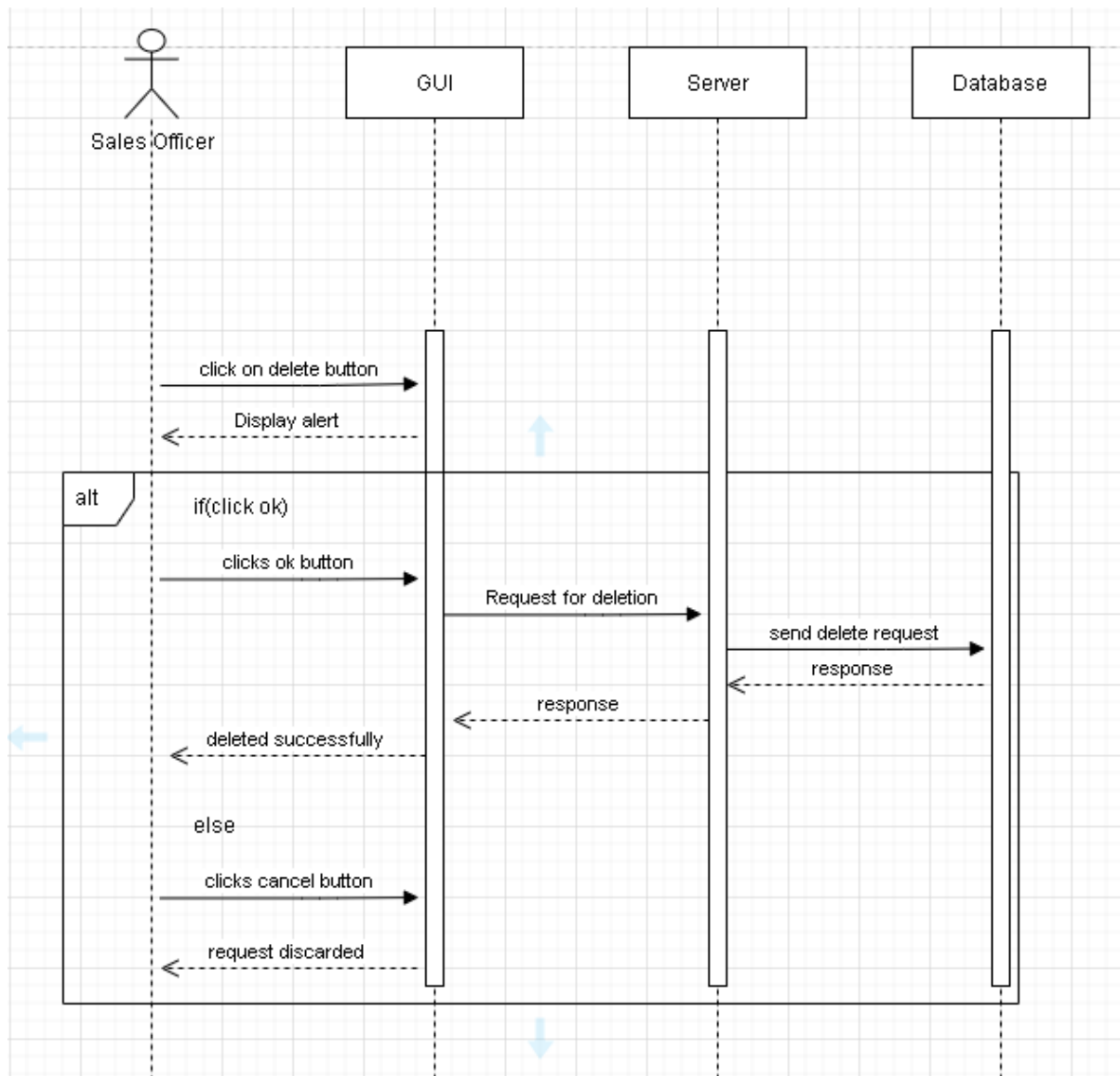


Fig 3-68: Delete Customer
Sequence diagram for deleting a customer

3.13.35 Search COA

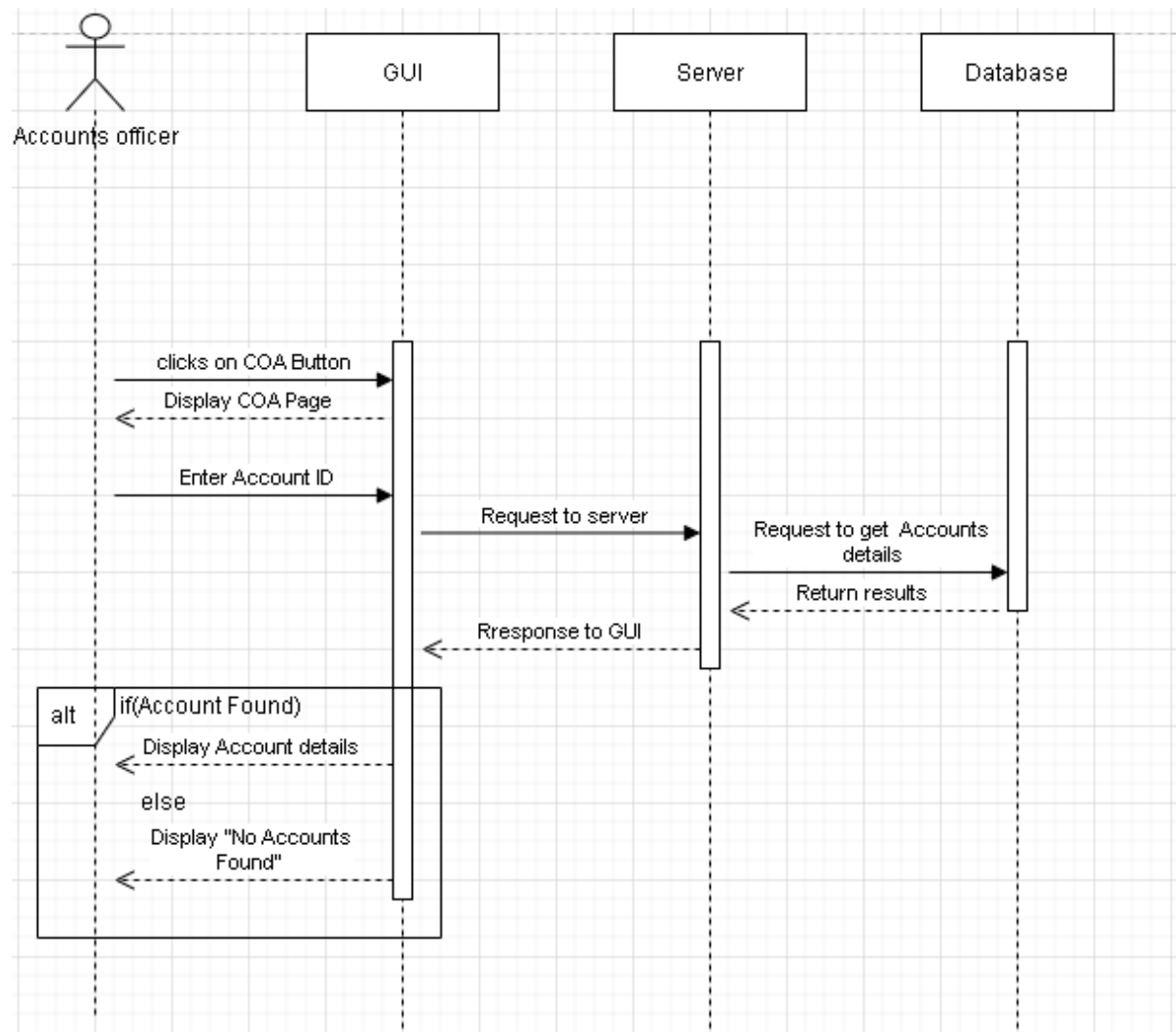


Fig 3-69: Search COA
Sequence diagram for searching the chart of accounts

3.13.36 Search Staff

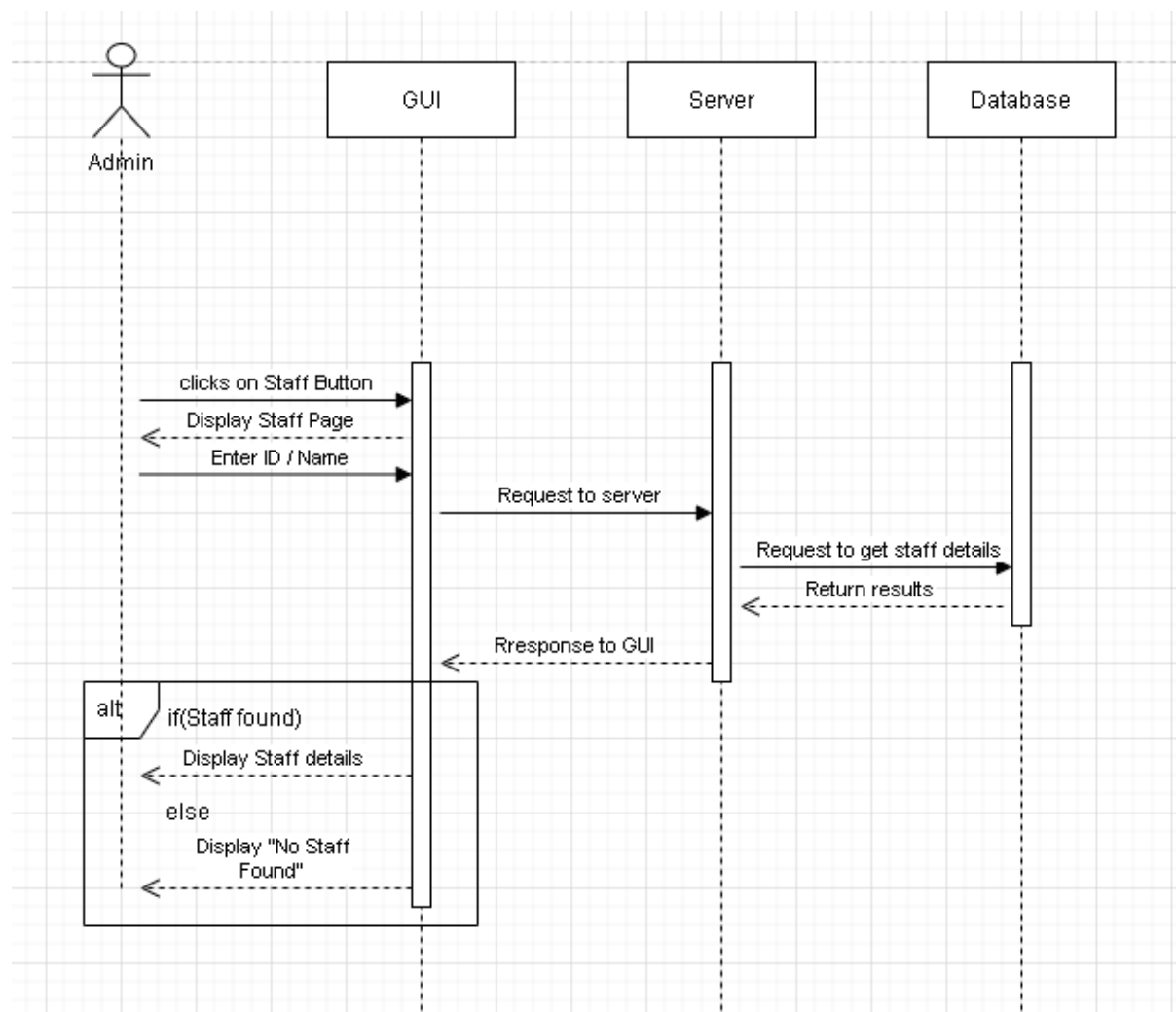


Fig 3-70: Search Staff
Sequence diagram for searching a staff member

3.13.37 Search Production

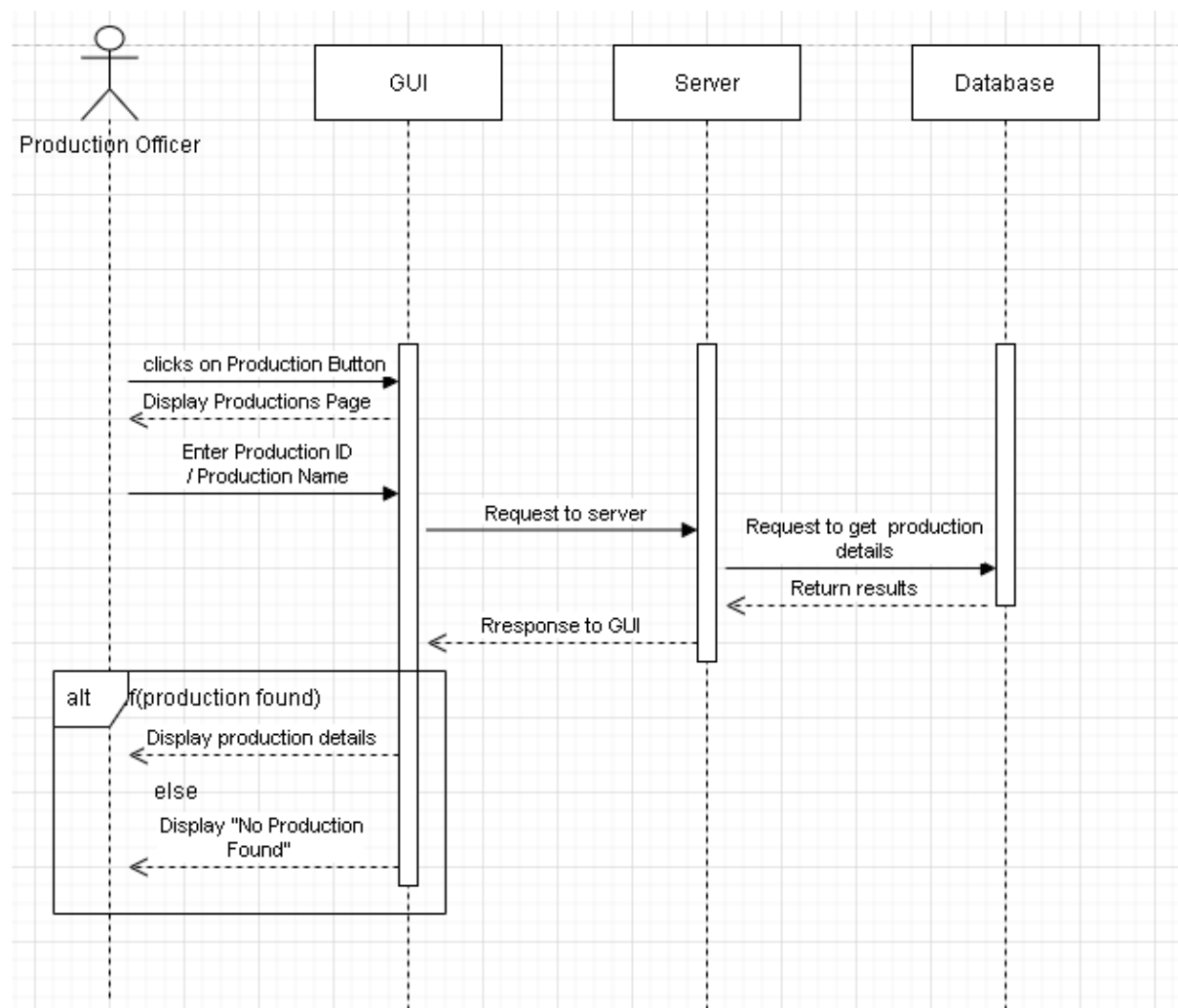


Fig 3-71: Search Production
Sequence diagram for searching an already existing production

3.13.38 Search Purchase

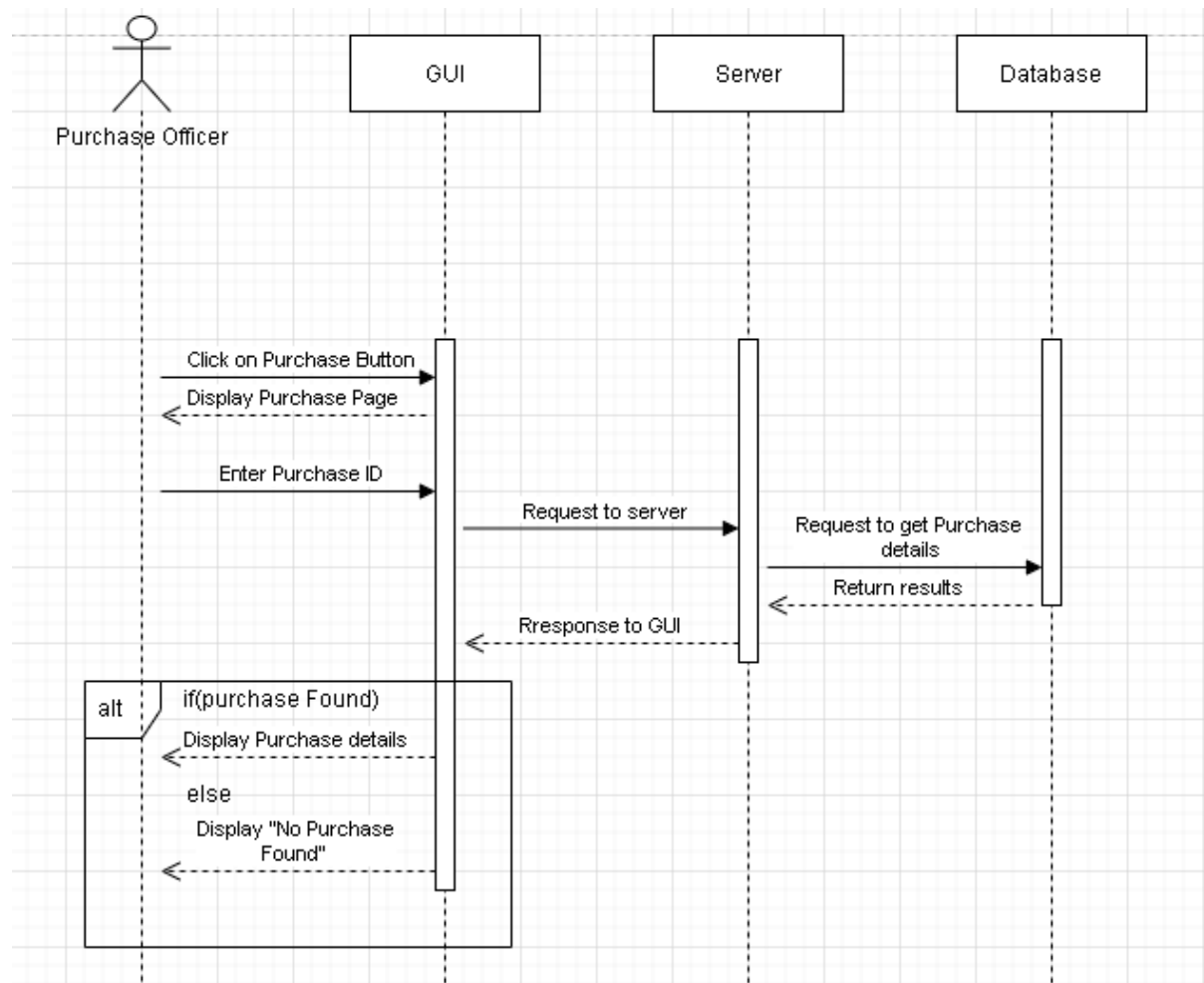


Fig 3-72: Search Purchase
Sequence Diagram for searching an existing purchase

3.13.39 Search Customer

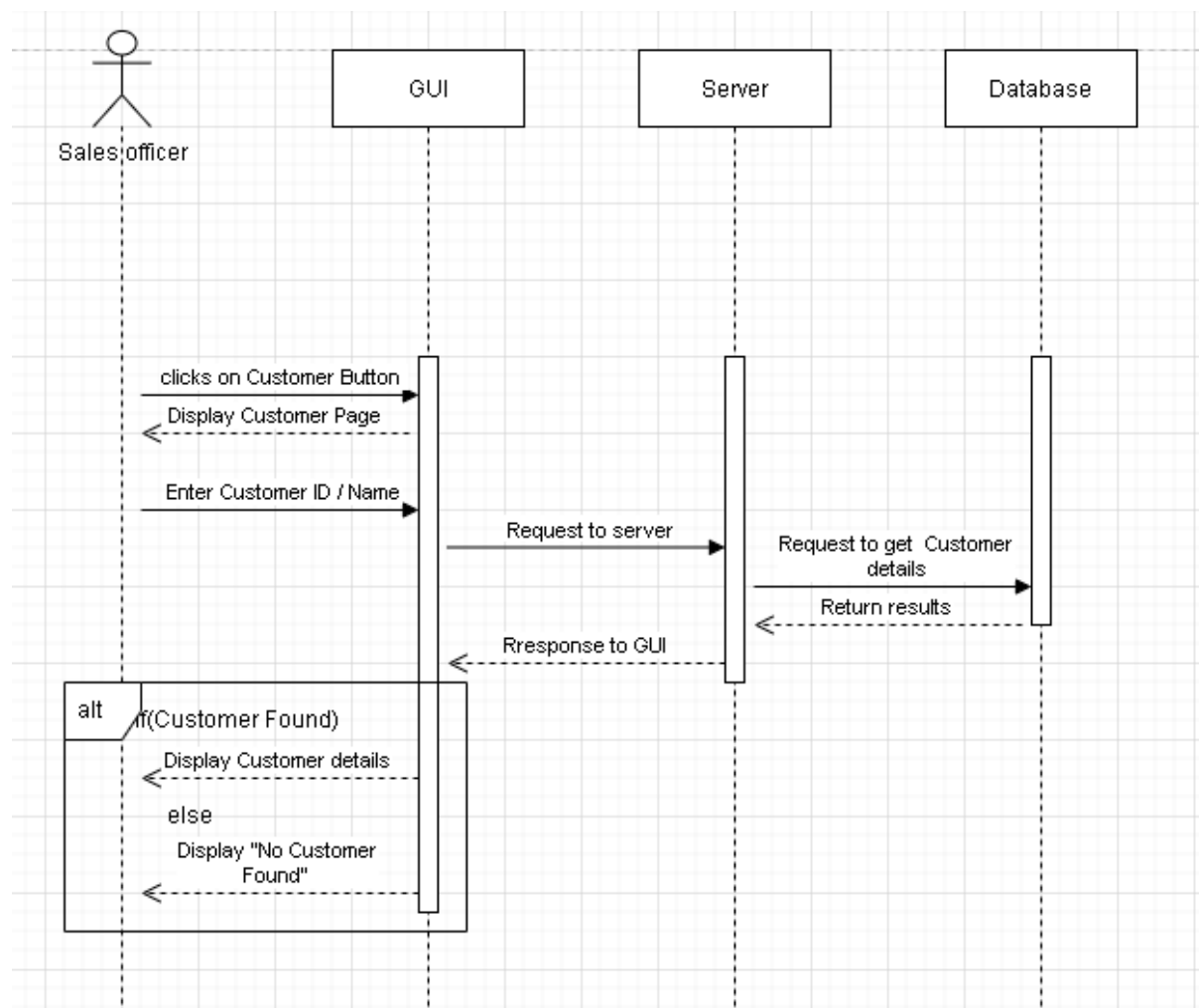


Fig 3-73: Search Customer
Sequence diagram for searching a customer

3.14 Policies and Tactics

3.14.1 Standards and Guidelines

We will be using the standard practices of ReactJS, Django, and MVT.

3.14.2 Data Collection

The data will be collected from the users and some data might be pre-stored as per the company's requirement.

3.14.3 Protocol

We will be using:

- HTTP
- JSON
- FTP

3.14.4 Algorithm(s)

We will be using several algorithms for different functionalities and for tracking the user activities.

3.14.5 Testing

For testing, we will be using Notion and Selenium for writing and running tests [5].

Chapter 4: Implementation and Test Cases

4.1 Implementation

The implementation that we have done so far is entailed below.

4.1.1 Prototype Description

The prototype of our system is created to demonstrate the functionality of our project. It is a web-based application which allows the users like admin, and other officers to perform their tasks. Admin can add the new Staff members with their roles in organization. The admin can see the list of total staff and can also modify the information of a specific staff member, can also search for a member and view his information in detail. Moreover, the staff member can also be deleted from the list. Then after Admin has added the members, they have been allocated the email and password for the user and then they can login to the system. The customer asks for the product, if the required quantity of product is not available then the Production Officer start the production using the raw materials bought from supplier by the Purchase officer.

Our System's Front-end is based totally on the JavaScript. The backend of this module is built using the python's Django framework using the MVT. It enabled us to build our server from scratch and meet the different ends to answer the different requests. Ultimately, the front-end of our system is completed in ReactJS and at the back-end we've used the Django to connect with the Database. To store the information or data we've used the MySQL Database.

4.1.2 Authentication

The Log in page is created from where the admin can sign in to access the modules available. Through this functionality the user will be able to access the underlying main functionality of our web application. The forgot password functionality is added in case the user forgets the password. The new password is automatically generated by the system and sent to the user.

4.1.3 Managing Staff

The admin will be able to manage the staff of the organization. Different services are provided in the module like adding a new staff member modifying the data of the existing staff member deleting the member and viewing the list of members. The staff member list can be seen by the Admin. Different search filters will be available for the admin to search for some specific kind of the members. The criteria for adding the new members are to be followed and the deletion of some staff members will be prohibited.

4.1.4 Add and Modify Staff

After the Module of the staff member is accessed by the admin than the new members can be added and the existing can be modified. As our Project is based on modules, the other modules will be having the similar functionalities so this module explains all the functionalities that will be provided by the system. The new member can be added if the certain data or information that is required is filled like the role of that member must be added. Then the admin will be able to save the changes as well.

4.1.5 Database

The database is one of the most important parts of any system. We have already designed the database for our system, we have connected the MySQL database with our system server to store the data and we have already deployed it on Heroku. All the data will be stored on the database which will be used in our system. There is also the backup for the database available offline as well. In case of any anomaly, the backup will be used to cope up with the problems of data loss.

4.2 Test Case Design and Description

4.2.1 Login

Login			
1			
Test Case ID:	1	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Login and Logout
Revision History:	-		
Objective	Test Login functionality with correct email and password		
Product/Ver/Module :	Login Module		
Environment:	The app is up and running and system is online		
Assumptions:	User is on login page		
Pre-Requisite:	User is already registered as a Staff		
Step No.	Execution description	Procedure result	
1	User enters email, password and clicks on login button	User is logged in and redirected to main dashboard page.	
Comments: The test case is passed and system is working as desired.			
☑Passed ☐Failed ☐Not Executed			

4.2.2 View Dashboard

View Dashboard			
2			
Test Case ID:	2	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	View dashboard
Revision History:	-		
Objective	Test the functionality of dashboard graphs and stats		
Product/Ver/Module :	Login Module		
Environment:	The app is up and running and system is online		
Assumptions:	Valid user		
Pre-Requisite:	User is already registered as a Staff and logged in		
Step No.	Execution description	Procedure result	
1	Clicks on Dashboard button	Redirected to main dashboard page.	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.3 View Products List

View Item List			
3			
Test Case ID:	3	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	View Product
Revision History:	-		
Objective	Test the functionality of listing the existing records of Product modules		
Product/Ver/Module :	Products		
Environment:	The app is up and running and system is online		
Assumptions:	User is on View List Page		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the Module button to go that module	The user is redirected to the module'	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.4 View Purchases List

View Purchases List			
4			
Test Case ID:	4	QA Test Engineer:	Ghasif Javed
Test case Version:	1.3	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	View Purchases
Revision History:	-		
Objective	Test the functionality of listing the existing records of all modules		
Product/Ver/Module:	Purchase		
Environment:	The app is up and running and system is online		
Assumptions:	User is on View List Page		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the Purchase Module button	The user is redirected to the Purchase module'	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.5 View Suppliers List

View Suppliers List			
5			
Test Case ID:	5	QA Test Engineer:	Ghasif Javed
Test case Version:	1.3	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	View Supplier
Revision History:	-		
Objective	Test the functionality of listing the existing records of Supplier module		
Product/Ver/Module:	Supplier		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the list view of Supplier		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the Supplier Module button	The user is redirected to the Supplier module'	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.6 View Customer List

View Customer List			
6			
Test Case ID:	6	QA Test Engineer:	Ghasif Javed
Test case Version:	1.3	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	View Customer
Revision History:	-		
Objective	Test the functionality of listing the existing records of Customer module		
Product/Ver/Module:	Customer		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the list view of Customer		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the Customer Module button	The user is redirected to the Customer module'	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.7 View Tasks List

View Tasks List			
7			
Test Case ID:	7	QA Test Engineer:	Ghasif Javed
Test case Version:	1.3	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	View Tasks
Revision History:	-		
Objective	Test the functionality of listing the existing records of tasks module		
Product/Ver/Module:	Tasks		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the list view of Tasks		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the Tasks Module button	The user is redirected to the Tasks module'	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.8 View Production List

View Production List			
8			
Test Case ID:	8	QA Test Engineer:	Ghasif Javed
Test case Version:	1.3	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	View Production
Revision History:	-		
Objective	Test the functionality of listing the existing records of Production module		
Product/Ver/Module:	Production		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the list view of Production		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the Production Module button	The user is redirected to the Production module'	
Comments: The test case is passed and system is working as desired.			
<input type="checkbox"/> Passed <input checked="" type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.9 View Staff List

View Staff List			
9			
Test Case ID:	9	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	View Staff
Revision History:	-		
Objective	Test the functionality of listing the existing records of Staff module		
Product/Ver/Module:	Staff		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the list view of Staff		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the Staff Module button	The user is redirected to the Staff module'	
Comments: The test case is passed and system is working as desired.			
<div>✓Passed <input type="checkbox"/>Failed <input type="checkbox"/>Not Executed</div>			

4.2.10 View Sales List

View Sales List			
10			
Test Case ID:	10	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	View Sales
Revision History:	-		
Objective	Test the functionality of listing the existing records of Sales module		
Product/Ver/Module:	Sales		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the list view of Sales		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the Sales Module button	The user is redirected to the Sales module'	
Comments: The test case is passed and system is working as desired.			
<div>✓Passed <input type="checkbox"/>Failed <input type="checkbox"/>Not Executed</div>			

4.2.11 View Accounts List

View Accounts List			
11			
Test Case ID:	11	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	View Accounts
Revision History:	-		
Objective	Test the functionality of listing the existing records of Accounts module		
Product/Ver/Module:	Accounts		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the list view of Accounts		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the Accounts Module button	The user is redirected to the Accounts module'	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.12 Edit Product

Edit Product			
12			
Test Case ID:	12	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Edit Product
Revision History:	-		
Objective	Test the functionality of Editing the existing records of Product modules		
Product/Ver/Module :	Products		
Environment:	The app is up and running and system is online		
Assumptions:	User is on Edit Record Page		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Modifies the the data in the record and press submit button	The data is modified and the data is saved and redirected to the list page	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.13 Edit Purchases

Edit Purchases			
13			
Test Case ID:	13	QA Test Engineer:	Ghasif Javed
Test case Version:	1.3	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Edit Purchases
Revision History:	-		
Objective	Test the functionality of Editing the existing records of all modules		
Product/Ver/Module:	Purchase		
Environment:	The app is up and running and system is online		
Assumptions:	User is on Edit form View		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Modifies the the data in the record and press submit button	The data is modified and the data is saved and redirected to the list page	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.14 Edit Supplier

Edit Supplier			
14			
Test Case ID:	14	QA Test Engineer:	Ghasif Javed
Test case Version:	1.3	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Edit Supplier
Revision History:	-		
Objective	Test the functionality of Editing the existing records of Supplier module		
Product/Ver/Module:	Supplier		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the Edit view of Supplier		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Modifies the the data in the record and press submit button	The data is modified and the data is saved and redirected to the list page	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.15 Edit Customer

Edit Customer			
15			
Test Case ID:	15	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Edit Customer
Revision History:	-		
Objective	Test the functionality of Editng the existing records of Customer module		
Product/Ver/Module:	Customer		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the Edit view of Customer		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Modifies the the data in the record and press submit button	The data is modified and the data is saved and redirected to the list page	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.16 Edit Tasks

Edit Tasks			
16			
Test Case ID:	16	QA Test Engineer:	Ghasif Javed
Test case Version:	1.3	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Edit Tasks
Revision History:	-		
Objective	Test the functionality of Editing the existing records of tasks module		
Product/Ver/Module:	Tasks		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the edit view of Tasks		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Modifies the the data in the record and press submit button	The data is modified and the data is saved and redirected to the list page	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.17 Edit Production

Edit Production			
17			
Test Case ID:	17	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Edit Production
Revision History:	-		
Objective	Test the functionality of Editing the existing records of Production module		
Product/Ver/Module:	Production		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the Edit view of Production		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Modifies the the data in the record and press submit button	The data is modified and the data is saved and redirected to the list page	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.18 Edit Staff

Edit Staff			
18			
Test Case ID:	18	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Edit Staff
Revision History:	-		
Objective	Test the functionality of Editing the existing records of Staff module		
Product/Ver/Module:	Staff		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the Edit view of Staff		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Modifies the data in the record and press submit button	The data is modified and the data is saved and redirected to the list page	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.19 Edit Sales

Edit Sales			
19			
Test Case ID:	19	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Edit Sales
Revision History:	-		
Objective	Test the functionality of Editing the existing records of Sales module		
Product/Ver/Module:	Sales		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the Edit view of Sales		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Modifies the data in the record and press submit button	The user is redirected to the Sales module'	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.20 Edit Accounts

Edit Accounts			
20			
Test Case ID:	20	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Edit Accounts
Revision History:	-		
Objective	Test the functionality of Editing the existing records of Accounts module		
Product/Ver/Module:	Accounts		
Environment:	The app is up and running and system is online		
Assumptions:	User is on the Edit view of Accounts		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Modifies the the data in the record and press submit button	The data is modified and the data is saved and redirected to the list page	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.21 Add a new Purchase

Add a new Purchase			
21			
Test Case ID:	21	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Make a new Purchase
Revision History:	-		
Objective	Test the functionality of adding new records of Purchase modules		
Product/Ver/Module :	Purchase		
Environment:	The app is up and running and system is online		
Assumptions:	User is on Create Form of Purchase		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the add new button to open a form to enter details	The app is redirected to the create form page	
2	User enters the details click the submit button	The new record is created	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.22 Add a new Product

Add a new Product			
22			
Test Case ID:	22	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Add a new Product
Revision History:	-		
Objective	Test the functionality of adding new records of Product modules		
Product/Ver/Module:	Product		
Environment:	The app is up and running and system is online		
Assumptions:	User is on Create Form view of Products		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the add new button to open a form to enter details	The app is redirected to the create form page	
2	User enters the details click the submit button	The new record is created	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.23 Add a new Supplier

Add a new Supplier			
23			
Test Case ID:	23	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Add a new Supplier
Revision History:	-		
Objective	Test the functionality of adding new records of Supplier modules		
Product/Ver/Module:	Supplier		
Environment:	The app is up and running and system is online		
Assumptions:	User is on Create Form view of Supplier		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the add new button to open a form to enter details	The app is redirected to the create form page	
2	User enters the details click the submit button	The new record is created	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.24 Add a new Customer

Add a new Customer			
24			
Test Case ID:	24	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Add a new Customer
Revision History:	-		
Objective	Test the functionality of adding new records of customer modules		
Product/Ver/Module :	Customer		
Environment:	The app is up and running and system is online		
Assumptions:	User is on Create Form view of Customer		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the add new button to open a form to enter details	The app is redirected to the create form page	
2	User enters the details click the submit button	The new record is created	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.25 Add a new Tasks

Add a new tasks			
25			
Test Case ID:	25	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Add a new Tasks
Revision History:	-		
Objective	Test the functionality of adding new records of tasks modules		
Product/Ver/Module:	Tasks		
Environment:	The app is up and running and system is online		
Assumptions:	User is on Create Form view of Tasks		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the add new button to open a form to enter details	The app is redirected to the create form page	
2	User enters the details click the submit button	The new record is created	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.26 Add a new Production

Add a new Production			
26			
Test Case ID:	26	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Add a new Production
Revision History:	-		
Objective	Test the functionality of adding new records of Production modules		
Product/Ver/Module :	Production		
Environment:	The app is up and running and system is online		
Assumptions:	User is on Create Form view of Production		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the add new button to open a form to enter details	The app is redirected to the create form page	
2	User enters the details click the submit button	The new record is created	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.27 Add a new Staff

Add a new Staff			
27			
Test Case ID:	27	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Add a new Staff
Revision History:	-		
Objective	Test the functionality of adding new records of Staff modules		
Product/Ver/Module:	Staff		
Environment:	The app is up and running and system is online		
Assumptions:	User is on Create Form view of Staff		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the add new button to open a form to enter details	The app is redirected to the create form page	
2	User enters the details click the submit button	The new record is created	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.28 Add a new Sales

Add a new Sales			
28			
Test Case ID:	28	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Add a new Sales
Revision History:	-		
Objective	Test the functionality of adding new records of Sales modules		
Product/Ver/Module:	Sales		
Environment:	The app is up and running and system is online		
Assumptions:	User is on Create Form view of Sales		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the add new button to open a form to enter details	The app is redirected to the create form page	
2	User enters the details click the submit button	The new record is created	
Comments: The test case is passed and system is working as desired.			
<input type="checkbox"/> Passed <input checked="" type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.29 Add a new Account

Add a new Account			
29			
Test Case ID:	29	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Add a new Account
Revision History:	-		
Objective	Test the functionality of adding new records of Acoounts modules		
Product/Ver/Module:	Accounts		
Environment:	The app is up and running and system is online		
Assumptions:	User is on Create Form view of Accounts		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the add new button to open a form to enter details	The app is redirected to the create form page	
2	User enters the details click the submit button	The new record is created	
Comments: The test case is passed and system is working as desired.			
☑Passed ☐Failed ☐Not Executed			

4.2.30 Delete a In Process Purchase

Delete a In Process Purchase			
30			
Test Case ID:	30	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Delete a Purchase
Revision History:	-		
Objective	Test the functionality of deleting selected record of a Purchase		
Product/Ver/Module :	Purchase		
Environment:	The app is up and running and system is online		
Assumptions:	User is on List View of Purchase		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the deleted icon on a record that is to be deleted	The record is deleted and removed from list page.	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.31 Delete a Product

Delete a Product			
31			
Test Case ID:	31	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Delete a Product
Revision History:	-		
Objective	Test the functionality of deleting selected record of a Product		
Product/Ver/Module:	Product		
Environment:	The app is up and running and system is online		
Assumptions:	User is on List View of Product		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the deleted icon on a record that is to be deleted	The record is deleted and removed from list page.	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.32 Delete a Supplier

Delete a Supplier			
32			
Test Case ID:	32	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Delete a Supplier
Revision History:	-		
Objective	Test the functionality of deleting selected record of a Supplier		
Product/Ver/Module:	Supplier		
Environment:	The app is up and running and system is online		
Assumptions:	User is on List View of Supplier		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the deleted icon on a record that is to be deleted	The record is deleted and removed from list page.	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.33 Delete a Customer

Delete a Customer			
33			
Test Case ID:	33	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Delete a Customer
Revision History:	-		
Objective	Test the functionality of deleting selected record of a Customer		
Product/Ver/Module :	Customer		
Environment:	The app is up and running and system is online		
Assumptions:	User is on List View of Customer		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the deleted icon on a record that is to be deleted	The record is deleted and removed from list page.	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.34 Delete a Task

Delete a Task			
34			
Test Case ID:	34	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Delete a Task
Revision History:	-		
Objective	Test the functionality of deleting selected record of a Task		
Product/Ver/Module :	Task		
Environment:	The app is up and running and system is online		
Assumptions:	User is on List View of Tasks		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the deleted icon on a record that is to be deleted	The record is deleted and removed from list page.	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.35 Cancel a Production

Cancel a Production			
35			
Test Case ID:	35	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Delete a Production
Revision History:	-		
Objective	Test the functionality of Cancel selected Production		
Product/Ver/Module :	Production		
Environment:	The app is up and running and system is online		
Assumptions:	User is on List View of Production		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the deleted icon on a record that is to be deleted	The record is deleted and removed from list page.	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.36 Delete an Account

Delete an Account			
36			
Test Case ID:	36	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Delete an Accounts
Revision History:	-		
Objective	Test the functionality of deleting selected record of a Accounts		
Product/Ver/Module :	Accounts		
Environment:	The app is up and running and system is online		
Assumptions:	User is on List View of Account		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the deleted icon on a record that is to be deleted	The record is deleted and removed from list page.	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.37 Delete Sale Order

Delete Sale Order			
37			
Test Case ID:	37	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Delete a Sale Order
Revision History:	-		
Objective	Test the functionality of deleting selected record of a Sale		
Product/Ver/Module :	Sale		
Environment:	The app is up and running and system is online		
Assumptions:	User is on List View of Sale		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User Clicks the deleted icon on a record that is to be deleted	The record is deleted and removed from list page. The Record must not be removable.	
Comments: The test case is failed and system is not working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.38 Search a Record

Search a Record			
38			
Test Case ID:	38	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	List of Products, Purchases, Sales, Customers, Staff
Revision History:	-		
Objective	Test the functionality of searching a specific record by any of its attributes		
Product/Ver/Module:	Purchase, Supplier, Customer, Staff, Production, Tasks, Sales Accounts		
Environment:	The app is up and running and system is online		
Assumptions:	User is on View List Page		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User enters a key value in the search bar	The User gets the list of records containing that search parameter	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.2.39 Generate Report

Generate a Report			
39			
Test Case ID:	39	QA Test Engineer:	Ghasif Javed
Test case Version:	1.0	Reviewed By:	Ghasif Javed
Test Date:	13-11-2022	Use Case Reference(s):	Generate Report
Revision History:	-		
Objective	Test the functionality of Creating a Report of selected Parameters		
Product/Ver/Module :	Report Module		
Environment:	The app is up and running and system is online		
Assumptions:	User is on Report Module		
Pre-Requisite:	User has authority to access that module		
Step No.	Execution description	Procedure result	
1	User sets the parameters to print the report		
2	User presses the button to open a report	The report is generated as a downloadable pdf file	
Comments: The test case is passed and system is working as desired.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

4.3 Test Metrics

Metric:	Purpose
Number of Test Cases:	39
Number of Test Cases Passed:	37
Number of Test Cases Failed:	2
Test Case Defect Density:	5.128
Test Case Effectiveness:	100
Traceability Matrix:	Included in a separate file

Chapter 5: Experimental Results and Analysis

First of all, we implemented different modules in our FYP. Sales, Purchases, Staff, Attendance, Production, Accounts, Customers, Suppliers, Tasks, and Admin. Their main functionalities consist of adding product, starting production of a product upon request, purchasing materials, Selling Products, managing Staff, and several Accounting functionalities will be done there through interactive interface using graphs. Talking about results and analysis, our main concern here is performance, so most of the functionalities are done given the fact of performance, so our system usually responds in milli-seconds. But in case of analytics for all managers and admin, it usually takes some time to load graphs. We also did take security into consideration, and our system is secure for most of the attacks. If we talk about UI, we have a good user interface and user experience and it is mostly easy to use. So, if we talk about some numbers, we did a lighthouse test and our performance score was 50% for the whole system and we did follow 85% of the best practices. Our page loading time was also pretty fast. So, that's it till right now, currently we are doing load testing and we are constantly trying to make our system better.

Chapter 6: Conclusion

Our Project is about providing the Enterprise Resource Planning system to a Medicine production organization where they can manage their work flow of their unit. The document starts with the abstract and introduction of our system. Then Functional and Non-Functional requirements are included. Furthermore, the class diagrams, sequence diagrams and architectural diagram are added. Also, the basic initial version of the GUI is represented in this document. These things will help in understanding the functionality and methodology of our system. Moreover, we have developed the deep knowledge about the project we are implementing. The basic objectives of our project are already being achieved. The prototype of our project shows the implementation of one module of our project whereas other modules will be pretty similar to the prototype. The Uses cases are also given in this document and the ER diagram is also added here in this document.

Since it is the end of our FYP, we have finished development of the application. We used a Django backend and a React JS frontend. Finally Medpy ERP has achieved to provide the user with the above mentioned functionalities. We can shift a company to the ERP now, once a user signup for the application they can perform day to day operations on the ERP. We have completed the admin, production, sales, purchase, and the accounting modules. The relevant employees can signin with their provided emails and passwords and start using the ERP.

References

- [1] Addo-Tenkorang, R., Helo, P. 2019. Enterprise Resource Planning (ERP): A Review Literature Report. *Proceedings of the World Congress on Engineering and Computer Science (WCECS)*, Vol. II.
- [2] Al-Fawaz, K., Al-Salti, Z., Eldabi,T. 2017. Critical Success Factors in ERP Implementation: A Review. European and Mediterranean Conference on Information Systems (EMCIS2008).
- [3] Rohit Kenge. 2020. A Research Study on the ERP System Implementation and Current Trends in ERP.
- [4] Sreekumar Menon. 2019. Critical Challenges in Enterprise Resource Planning (ERP) Implementation. *International Journal of Business and Management; Vol. 14, No. 7; 2019*
- [5] Ahmed Elragal, Ayman M Al-Serafi. 2017. The Effect of ERP System Implementation on Business Performance: An Exploratory Case-Study.