



FINAL SEMESTER ASSESSMENT (FSA)
B.TECH. (CSE)
VI SEMESTER

**UE18CS355 – OBJECT ORIENTED ANALYSIS AND DESIGN
WITH SOFTWARE ENGINEERING LABORATORY**

PROJECT REPORT
ON
WAREHOUSE MANAGEMENT SYSTEM

SUBMITTED BY

NAME

SRN

1) Navya Parameshwar Hegde

PES220180007

2) Nayana C S Reddy

PES2201800510

3) Shloka Lakka

PES2201800502

JANUARY – MAY 2021
**DEPARTMENT OF COMPUTER SCIENCE &
ENGINEERING**
EC CAMPUS,
BENGALURU – 560100, KARNATAKA, INDIA

TABLE OF CONTENTS		
Sl.No	TOPIC	PAGE No
	ABSTRACT	1
1.	SOFTWARE REQUIREMENTS SPECIFICATION	2
2.	PROJECT PLAN	15
3.	DESIGN DIAGRAMS	20
4.	MODULE DESCRIPTION	26
5.	TEST CASES	27
6.	SCREEN SHOTS OF OUTPUT	36

ABSTRACT

In our project titled Warehouse Management System, we tackle everyday activities of a warehouse and try to automate most of the activities in order to make it easier for the users by reducing manual effort and precious time.

This project caters to three types of users- Suppliers, Customers and Pickers. Suppliers are the ones that supply items to the warehouse. Customers are the users that visit the warehouse to buy items that they require. Pickers are the employees of the warehouse that aid the suppliers and customers in their tasks.

Each unique user has a specific menu that caters to the tasks they are allowed to perform. The common functionality in the system for all three types of users would be the login(albeit having different passwords) and exit. The menu for the supplier consists of adding new products, displaying the stock, refilling stock, removing items, and exiting the menu. The customer's menu consists of purchasing items, displaying stock, and exiting the menu. The picker's menu consists of displaying stock, refilling the stock, and exiting the menu.

Software Requirements Specification

for

Warehouse Management System

Version 1.0 approved

Prepared by Varun Seshu, Hritik Shanbhag, Shashwath Kumar S, Ajay

**Varun Seshu contribution: Sections 1, 3 and 5 Section 4 Use case
diagram**

PES University

2/2/2021

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Intended Audience and Reading Suggestions	1
1.3 Product Scope	1
1.4 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	3
2.4 Operating Environment	3
2.5 Design and Implementation Constraints	3
2.6 Assumptions and Dependencies	3
3. External Interface Requirements	4
3.1 User Interfaces	4
3.2 Software Interfaces	5
3.3 Communications Interfaces	5
4. Analysis Models	5
5. System Features	6
5.1 Login for supplier, customer and picker	6
5.2 Customer ordering and item	7
5.3 Supply of items to warehouse	7
5.4 Pickers picking items	8
6. Other Nonfunctional Requirements	9
6.1 Performance Requirements	9
6.2 Safety Requirements	9
6.3 Security Requirements	9
6.4 Software Quality Attributes	9
6.5 Business Rules	10
Appendix A: Glossary	10
Appendix B: Field Layouts	11
Appendix C: Requirement Traceability matrix	11

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to capture at a functional level, the description and requirement of a warehouse management system. The focus here is to provide services to the customer, the supplier and the warehouse employee (the picker). This product allows the customers and suppliers to place orders, which are then executed by the system via the pickers (the employees) who take the inbound/outbound product in/from the warehouse to the destination.

1.2 Intended Audience

This document can be used by developers as a guide to building the product in accordance to the features mentioned in section 5. It can also be used by project managers to track the progress of the product. It can also be used by documentation writers as a reference to navigate the product and provide adequate documentation. The rest of this document contains a comprehensive description of the product and its target users, capabilities, shortcomings, design decisions, interfaces requirements and features.

1.3 Product Scope

The main goal of the product is to provide a solution to ensure smooth functioning of a warehouse (described in section 2.6), ensuring both optimal allocation of space in the warehouse of inbound items and the optimal path that a picker needs to take in order to move items within the warehouse. This allows for maximum items to be stored in the warehouse and minimum time required to move objects within the warehouse. In the long run, this approach allows for minimum costs in the functioning of the warehouse and lower costs to customers.

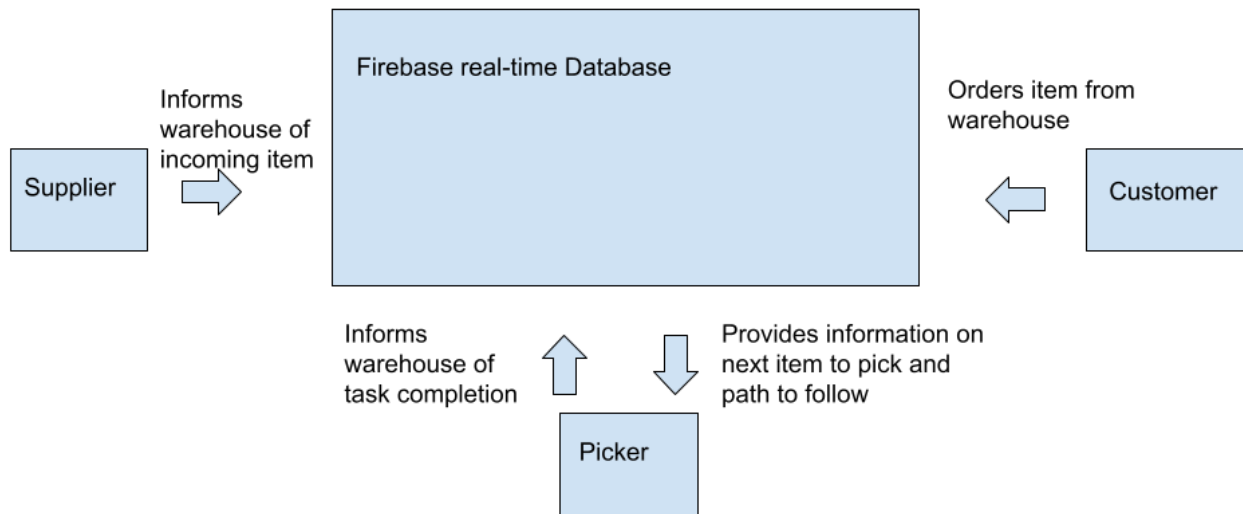
1.4 References

- Vue js styling standards for the user interface - <https://vuejs.org/v2/style-guide/>
- Warehouse Management system - https://aspnet.cob.ohio.edu/isms/upload/documents/10867_635355682482963914_Ferrari_Order_Receiving_SDLC_Document.pdf
- <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>

2. Overall Description

2.1 Product Perspective

This product is a new, self-contained system which allows for the efficient functioning of a warehouse. It consists of 3 different user classes, each interacting with the database as shown in the diagram below.



*All communication via RESTful Api calls

2.2 Product Functions

The major functions the product the product must perform are:

- Handling incoming item requests according to priority.
- Handling Inbound items efficiently in a time bound manner.
- Allocate space in the warehouse items in an optimal manner.
- Finding an Optimal path for the closest free picker to transport the item.
- Notifying the picker of the item and path
- Efficiently handling outgoing requests based on priority.
- Handling Outbound items efficiently in a time bound manner.

2.3 User Classes and Characteristics

There are 3 types of user classes:-

- Supplier :-
 - The supplier informs the warehouse about incoming items that were requested by the warehouse.
- Customer:-
 - The customer will order an item or request a service from the warehouse based on his or her requirements .
- Picker:-
 - Pickers are the middle men between the interaction of customer and supplier in the warehouse.
 - They will inform the warehouse of task completion.
 - They receive information from the system on what item to pick and path to follow in the warehouse.

2.4 Operating Environment

The backend of our system is built on the firebase cloud services platform. We use the firebase real-time database to store our data. Further, our application uses cloud functions for computations in the cloud and interaction with the database. Also, the frontend of our website is hosted on the firebase hosting service. Thus, our product operates on different services provided by the firebase cloud platform which must interact to provide a flawless user experience.

2.5 Design and Implementation Constraints

The following are the constraints:

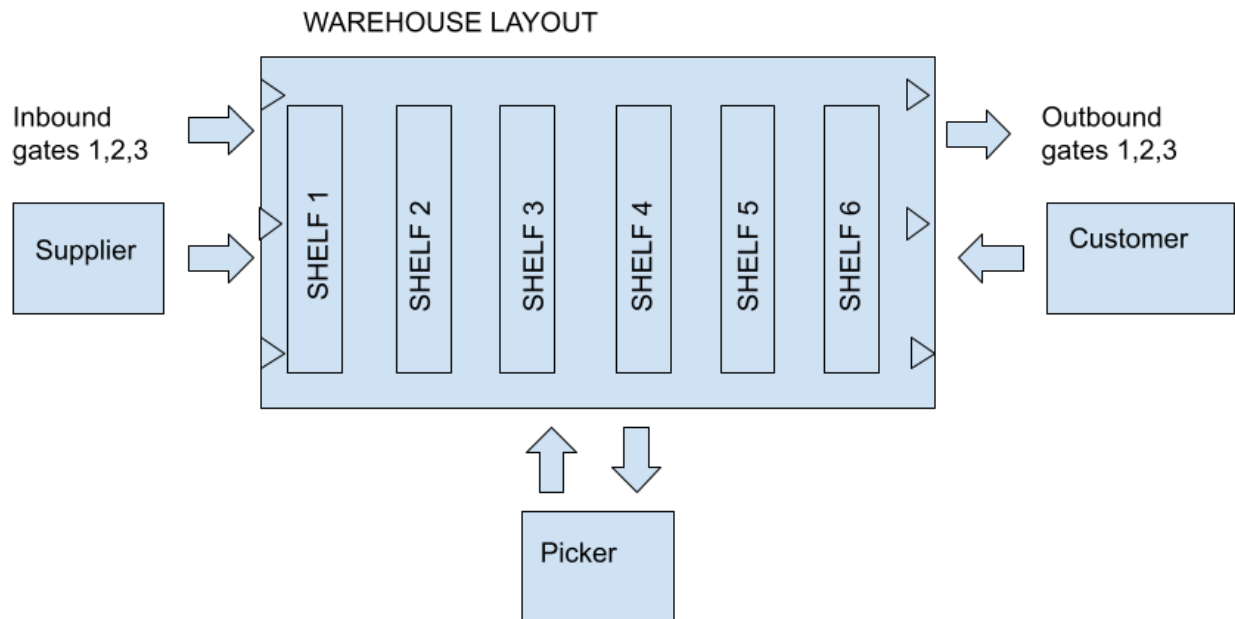
- Languages to be used in the backend - TypeScript, JavaScript
- Firebase Real-time database
- All communications encrypted through HTTPS
- All communications via RESTful API calls
- Many processes are integrated into a system hence an error in one place will involve errors in others.
- Web site response time < 2 seconds for any action performed.
- There must be high reliability for the web app.
- Vue js styling standard followed for the User Interface
- Object Oriented Programming principles to be followed
- Login and signup secured via firebase authentication

2.6 Assumptions and Dependencies

Assumptions:

- Supplier informs the warehouse on the items arriving at a particular inbound gate.
- The system can deny an item from the supplier.
- Picker can only pick one item at a time.
- No limit on the dimension of items a picker can pick.

- No cancellation of orders after being placed.
- The items in the warehouse are non-perishable i.e. they have very long shelf life so we do not need to worry about discarding degrading items.
- Items are classified as large and small volume and put in different shelves according to their dimensions.
- Assumption of layout of the warehouse according to the following diagram



3. External Interface Requirements

3.1 User Interfaces

As mentioned in section 2.1, the 3 user perspectives have their own abstractions in the website. We follow the Vue.js standards for styling (mentioned in section 1.4) throughout the site. For each user, we have a user interface tailored for them, but 'log out' is a standard button that is present for all users. The user interface for the customer allows for placement of orders to the warehouse, for the employee it notifies when an order is placed along the optimal path to pick up and drop the item. Finally, it allows the supplier to inform the warehouse of incoming items.

3.2 Software Interfaces

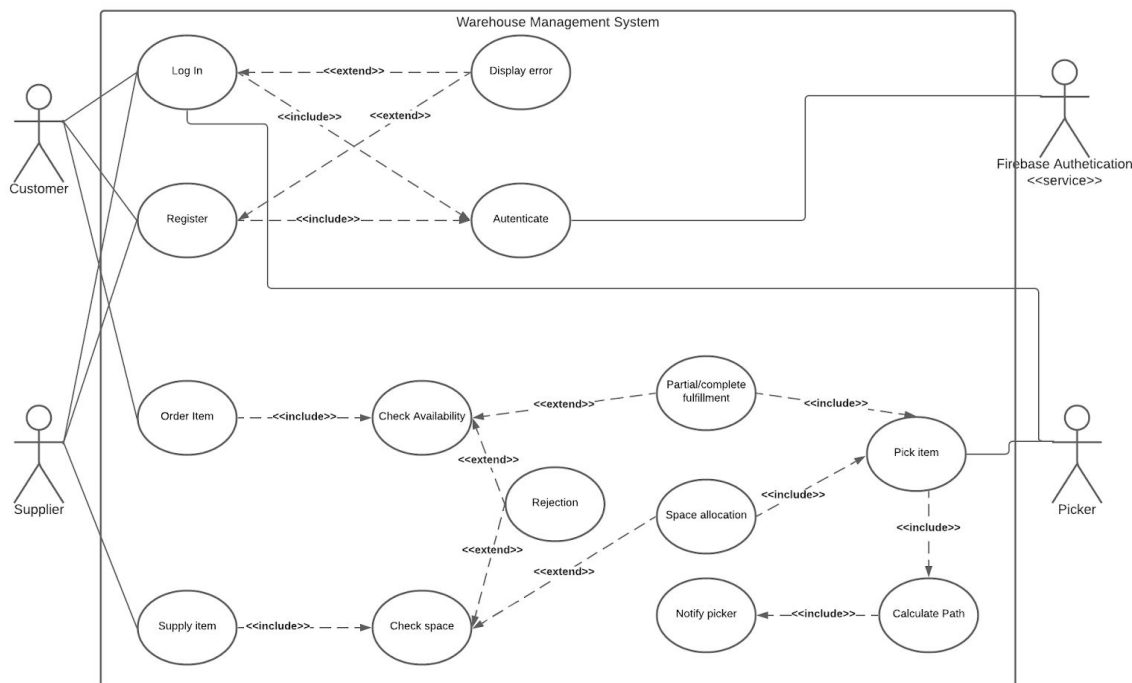
Since the backend for the product is hosted on firebase cloud services, we integrate with a number of firebase tools in our product. We use the CLI interface 'firebase-tools' to deploy our product and update it. For the cloud functions, we use the 'firebase' library to upload and execute our code on a API call. Further, we use the firebase 'real-time database' as our storage solution and communicate with the database and perform actions via an API call to firebase cloud functions. We also host the system via firebase hosting. The firebase real-time database acts as a central repository of information, the 2 way communication between the UI on the browser and the database happens via RESTful APIs, more information is provided in section 3.3.

3.3 Communications Interfaces

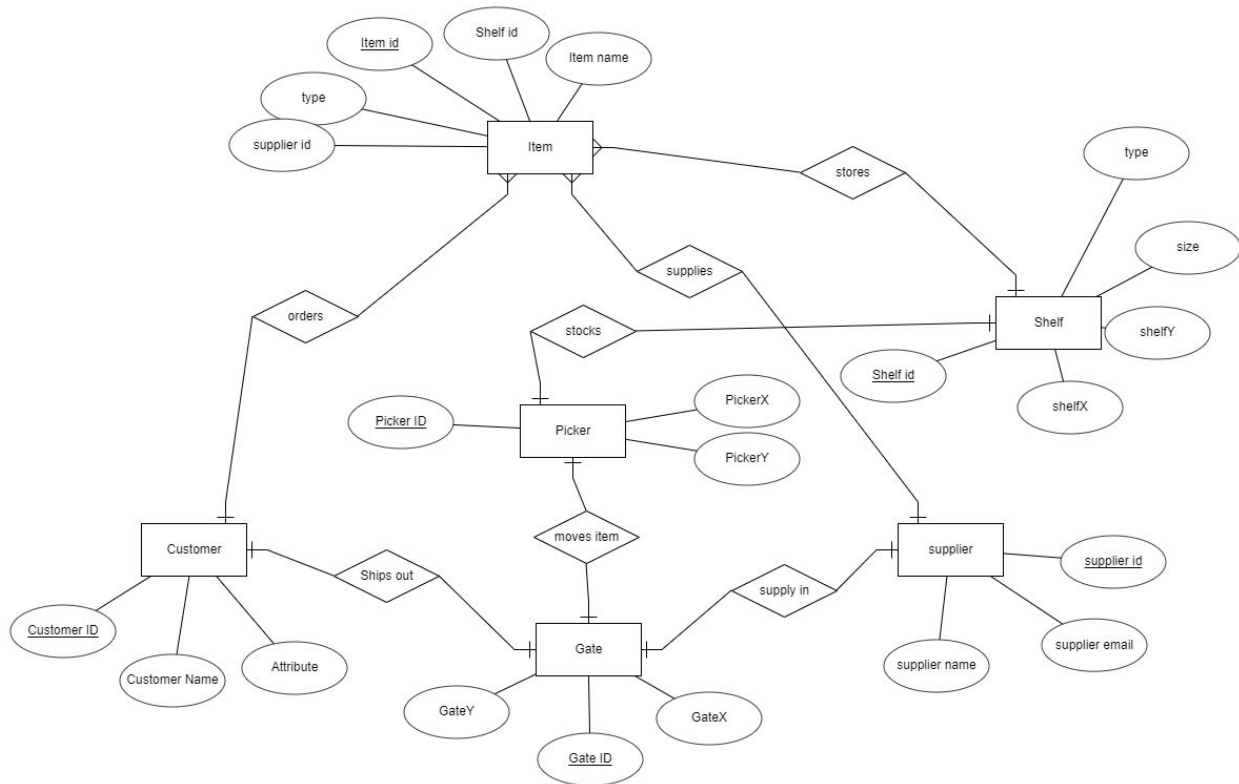
To access the web site of our product, we require the chrome browser. All communications between the server (hosting the backend) and the frontend (User Interface) is through a RESTful API - which uses the JSON format and where the data is converted to a string before transmission. For data transmission, the site uses the HTTPS protocol. A network connection of at least 5 Mbps is required for a smooth user experience. The firebase real-time database automatically synchronizes data across all active users.

4. Analysis Models

Use Case Diagram



ER Diagram



5. System Features

5.1 Login for supplier, customer and picker

5.1.1 Description and Priority

Login and signup feature provided for both supplier, customer and picker. Authentic email id must be provided for safety reasons. The priority of this feature is High.
 Benefits - easier maintenance of warehouse demands and requests.
 Risk - Fraud email id entering as supplier or customer may manipulate requests.

5.1.2 Stimulus/Response Sequences

- Login customer/supplier/picker when correct email and password provided.
 - Else provide option to sign up
- Registration
 - Get supplier/customer sign up information.
 - Authenticate information such as email id.
 - Register the supplier/customer to the database.
 - Route back to login page for authentication.

5.1.3 Functional Requirements

REQ-1: Different Signup/Login Page for customer, supplier and picker.

REQ-2: Authentication of registered individuals must be done through firebase authentication.

REQ-3: Registration of individuals must be done through firebase authentication.

REQ-4: Registration allowed only for customers and suppliers.

5.2 Customer ordering an item

5.2.1 Description and Priority

This feature allows the customer to place an order for items in the warehouse. The warehouse may reject or partially fulfill the order based on availability of items in the warehouse. Further the customer needs to provide adequate information in the order about the item bought, the quantity and the outbound gate to send the item to.

5.2.2 Stimulus/Response Sequences

- Customers place an order including information of the item, the quantity and the outbound gate the item must be delivered to.
- The system check for availability of item and responds in 3 possible manners
 - If the item is available, the customer is notified of order fulfillment.
 - If the item is available, but not in sufficient quantity, the customer is notified of partial fulfillment of order.
 - If the item is unavailable, the customer is notified of non-availability and the order is rejected.

5.2.3 Functional Requirements

REQ-5: Customers place orders via a blue 'order' button in the user interface.

REQ-6: Customers can select items via a dropdown menu.

REQ-7: Customers can also type in the name of the item they want to order.

REQ-8: Customers are notified in case of order fulfillment, partial fulfillment and rejection.

REQ-9: Customers must provide information of the item name, quantity to buy and the outbound gate to send the item to in an order.

5.3 Supply of items to warehouse

5.3.1 Description and Priority

The suppliers ensure there is a flow of items into the warehouse through the inbound gates. This feature handles this supply of items. The supplier needs to provide adequate information of the dimensions of the supplied item and inbound gate the item has arrived at.

5.3.2 Stimulus/Response Sequences

- Supplier sends an item, which arrives at an inbound gate in the warehouse.
- The supplier notifies the warehouse of the item, providing information of the item such as name, quantity, dimensions and inbound gate.
- The system may then respond in 2 ways
 - If the warehouse does not have adequate space to allocate for the item, the item is rejected and the warehouse notifies the supplier.
 - If there is adequate space in the warehouse for only a partial quantity of items, then a picker is notified to pick the item and the supplier is notified that the item is partially received.
 - If there is adequate space in the warehouse, then a picker is notified to pick the item and the supplier is notified that the item is successfully received.

5.3.3 Functional Requirements

REQ-10: Suppliers notify the system via a blue 'notify' button in the user interface.

REQ-11: Suppliers can select item attributes via a dropdown menu.

REQ-12: Suppliers are notified in case of item rejection and partial or successful fulfillment.

REQ-13: Suppliers must provide information of the item name, quantity, dimensions and inbound gate.

REQ-14: The system must optimally allocate space in order to maximise space usage and minimise empty space in the warehouse.

5.4 Pickers picking items

5.4.1 Description and Priority

Pickers are assigned to transport items from one part of the warehouse to another. The pickers are sent the information on the item, the quantity to pick and the gate to transport to/from the warehouse.

The pickers are also sent the information on the optimal path to follow in order to pick the item.

5.4.2 Stimulus/Response Sequences

- System notifies a picker who is free and is closest to the gate where the item is present.
- System informs the picker on the item, the quantity, the gate to transport to/from the warehouse and the optimal path to follow.
- After successfully picking an item, the picker notifies the system.

5.4.3 Functional Requirements

REQ-15: Notification to picker of item information, gate and path to follow.

REQ-16: Finding the closest picker for each transport.

REQ-17: Tracking pickers and maintaining information on their current position.

REQ-18: Pickers must notify the system via the blue 'task completed' button after each transport.

REQ-19: System must find an optimal path for the picker to follow to save time and energy.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

- Latency of website
 - Must have a latency of maximum latency of < 500 ms
- Concurrency
 - Must accommodate at least 15 active users concurrently
- Scalability
 - Compute must scale quickly and on demand
 - Storage must scale up from Gigabytes to terabytes

6.2 Safety Requirements

Since we are using Firebase for hosting it has very low down-times and is very reliable. Precautions that are needed to be taken care of are only in edge case conditions and bugs.

Incase of a catastrophic failure and the database is unsalvageable, there should be an option to recover the previous version of the database, with all the details. There should also be an option to record each operation so that these operations can be performed on the old database to get an updated version.

6.3 Security Requirements

To prevent misuse of data and make sure that data is regularly updated, a strong and secure database must be used. This database should be capable of handling multiple transactions at once. The entire system must be completely secure to ensure that there is no theft or misuse of items in the inventory. There must also be a system that takes stock of the inventory at the end of the day to ensure nothing is missing. The system should not be accessible by all and different people should have different database accessibilities. For example, the manager should have complete access to the database whereas for the warehouse pickers, it would be enough for them to have just writing and reading permissions, and not editing the database.

Firebase Security Rules leverage extensible, flexible configuration languages to define what data your users can access for Realtime Database, Cloud Firestore, and Cloud Storage.

6.4 Software Quality Attributes

- **Availability:** The item should be available and pickers should be free at a given time.
- **Adaptability:** Can work on most devices since it is a web app.

- **Correctness:** The item must be labelled with the correct supplier and shipped out to the correct customer.
- **Maintainability:** The application should use continuous integration so that features and bug fixes can be deployed quickly without downtime.
- **Usability:** The warehouse should have maximum throughput of items coming in and going out.

6.5 Business Rules

1. If the manager is missing, the head of the warehouse workers can handle the database.
2. Maintain safety stock levels.
3. Control the flow of goods.
4. Maintain quality control.

Domain requirements:

1. Tracking of items
2. Inbound and outbound items management
3. Order allocation
4. Storage management

Appendix A: Glossary

1. **Supplier** - Class which supplies items to the warehouse
2. **Customer** - Class which orders items from the warehouse for shipping
3. **Item** - Any object transported in and out of the warehouse
4. **Gates** - Where items are brought in or sent out
5. **Shelf** - Where items are stored
6. **Picker** - Class which is used to move items in warehouse from gates to shelf or vice
7. **Firebase** - Platform developed by Google for creating mobile and web applications
8. **JSON** - open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types
9. **REST API** - standard for a software architecture for interactive applications that typically use multiple Web services
10. **Cloud Resource** - availability of computer system resources, especially data storage and computing power, without direct active management by the user

Appendix B: Field Layouts

Field	Length	Data Type	Description	Is Mandatory
Email id	20	alphanumeric	email id of customer/supplier	Y
password	20	alphanumeric	password of the corresponding email id	Y
item name	20	alphanumeric	name of the item requested by the customer	Y
Location	40	alphanumeric	gate number and its location	Y
Dimension	20	numeric	volume of item	Y
supplier	20	alphanumeric	supplier name	Y
customer	20	alphabets	customer name	Y
quantity	20	numeric	quantity of items	Y

Appendix C: Requirement Traceability Matrix

Sno.	Requirement ID	Brief Description of Requirement	Design Reference	Code File Reference	Test Case ID	System Test Case ID		
1	REQ-1	Different Signup/Login Page for customer, supplier and picker.						
2	REQ-2	Authentication of registered individuals must be done through firebase authentication.						
3	REQ-3	Registration of individuals must be done through firebase authentication.						
4	REQ-4	Registration allowed only for customers and suppliers.						
5	REQ-5	Customers place orders via a blue 'order' button in the user interface.						
6	REQ-6	Customers can select items via a dropdown menu.						
7	REQ-7	Customers can type in the name of the item they want to order.						
8	REQ-8	Customers are notified in case of order fulfillment, partial fulfillment and rejection.						
9	REQ-9	Customers must provide information of the item name, quantity to buy and the outbound gate to send the item to in an order.						
10	REQ-10	Suppliers notify the system via a blue 'notify' button in the user interface.						
11	REQ-11	Suppliers can select item attributes via a dropdown menu.						
12	REQ-12	Suppliers are notified in case of item rejection and partial or successful fulfillment.						
13	REQ-13	Suppliers must provide information of the item name, quantity, dimensions and inbound gate.						
14	REQ-14	The system must optimally allocate space in order to maximise space usage and minimise empty space in the warehouse.						
15	REQ-15	Notification to picker of item information, gate and path to follow.						
16	REQ-16	Finding the closest picker for each transport.						
17	REQ-17	Tracking pickers and maintaining information on their current position.						
18	REQ-18	Pickers must notify the system via the blue 'task completed' button after each transport.						
19	REQ-19	System must find an optimal path for the picker to follow to save time and energy.						

Project Planning

Individual contributions:

Nayana CS Reddy :Designing the lifecycle for the execution,tools required for the life cycle.

Shloka Reddy Lakka:Determining the deliverables and classifying them into build or reuse,Creating the WBS for the entire functionality.

Navya Parameshwar Hegde:A rough estimation of the efforts required to complete the tasks,creating a gantt chart for scheduling them.

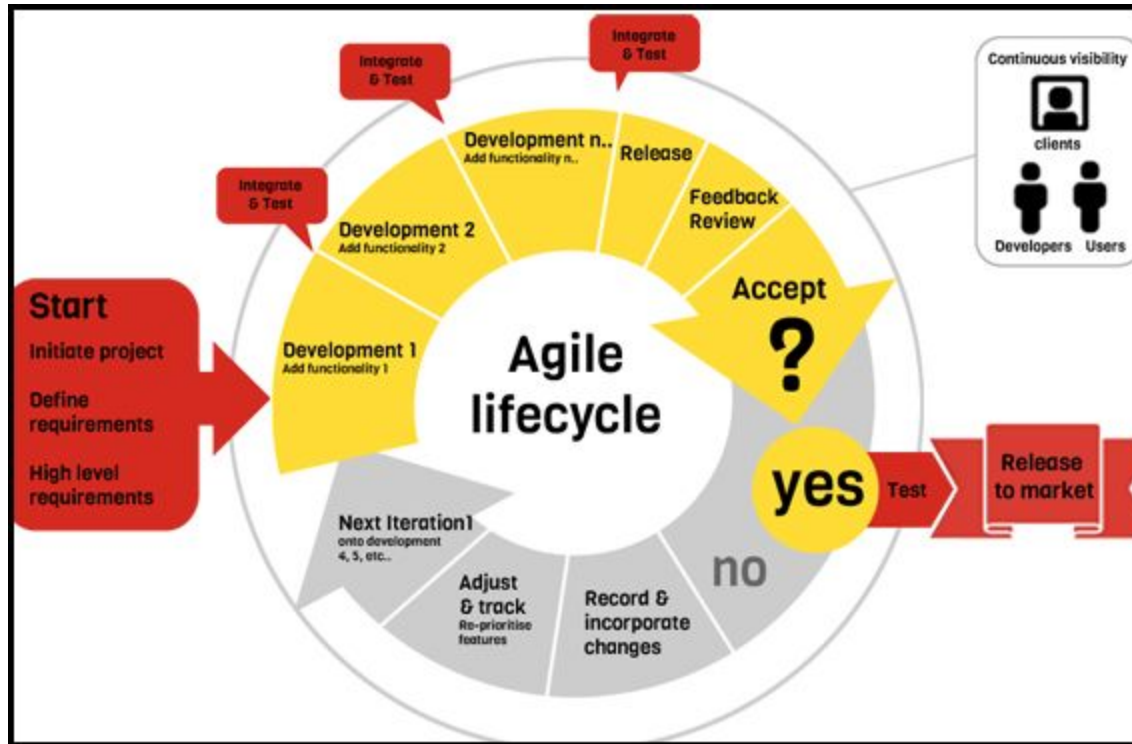
1: Identify the lifecycle to be followed for the execution of your project and justify why you have chosen that model.

After some background research, we decided that the agile model best suited our requirements. Agile seemed an obvious choice for the following reasons :

- Easily and quickly adaptable to change : Teams not only adapt to change in Agile, they are encouraged to embrace the practice. Agile acknowledges that customer needs change and that teams must be able to adapt. Working in time-boxed iterations means the team does not need to wait on a lengthy requirement change, review and approval process. Any change or maintenance item is added to the backlog and allotted to an upcoming sprint based on priority and business need.
- Higher quality product : On an agile project, we wouldn't have to start developing all features at once. We can assign a smaller subset of features at each sprint (The Agile process consists of short, time-boxed iterations known as sprints).
- Better stakeholder engagement : For a project implementing an agile model to be successful, it is important for the owner to be engaged throughout the process.
- Greater customer satisfaction : When using an agile model, the owner will be working very closely with the team. They will get to see a finished working product at the end of each sprint and will be pleased that the team can deliver releases more quickly and frequently.
- Better project control : The team works together, along with the product owner, to determine what goes into each sprint. That way, the team is on the same page about what needs to be delivered. There is less of a chance of surprises or unplanned features making it into the build.

- Reduced risk : Agile project management essentially eliminates the chance of absolute project failure. Working in sprints from the very beginning allows the team to develop a working product from the beginning, or fail fast and take another approach.

A basic diagram to understand the agile lifecycle model :



2: Identify the tools which you want to use throughout the lifecycle like planning tool, design tool, version control, development tool, bug tracking, testing tool.

Planning tool : Gantt Chart (lucidchart)

Design tool : Adobe XD (template), StarUML (diagrams), Google Docs, Slides

Version control : Git

Development tool : Apache Tomcat, JDK, JRE / VSCode

3: Determine all the deliverables and categorise them as reuse/build components and justify the same.

1. User Registrations: Registration details are stored in the database. This must be built.
2. User logins: successful login if the ID and password entered are correct. This must be built.
3. Book an Appointment: Patients can book an appointment with the doctor of their choice as well as choose a time. This must be built.

4. Prescription: Details of the medicines to be taken by the patient with the doses along with the next appointment date is provided. This must be built.

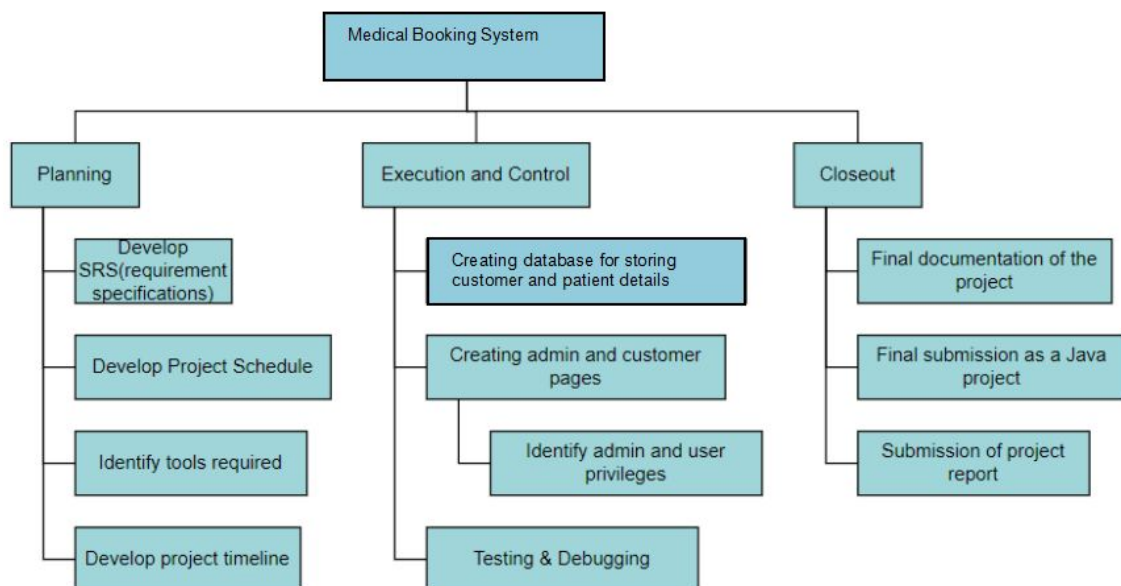
5. Payment: Fees to be paid to each doctor. This must be built.

6. Online Report: Patients lab and health reports are shown. This must be built.

The above components will be included under build because they have to be built from scratch.

Any libraries or APIs we may have to use to maintain records will be under reuse.

4: Create a WBS for the entire functionalities in detail.



Functionalities :

CUSTOMER

- Entering the website
 - Sign up - User must sign up before booking
 - Name - It is an important field and cannot be skipped.
 - Username - This will be used for logging in once the user profile is created.
 - Password - Required for signing up safely.
 - Email-id - Required for registration

Phone Number - Required for registration.

- Log in
Username- For logging in
Password - For logging in safely
- Booking an appointment
Enter doctor name - Entering the name of the doctor they wish to consult
Select consultation time - Choosing a time slot convenient for their consultation..

ADMIN

- Login - To gain the admin permissions
Username - Unique to the admin
Password - Unique to the admin
- Approve/reject appointment requests - Approving or rejecting the appointment requests depending on the availability of the doctors.
- Add/remove users - Admin has the permission to add or delete users.
- Grant permission to doctors - Admin has the power to grant permission to doctors to upload the prescription.
- React to complaints - An admin has the permissions to view user complaints and take necessary actions.

DOCTOR

- Login - To gain access to the consultation platform
Username - Unique to the doctor
Password - Unique to the doctor
- Consult patients - Consult patients at the time allocated and help them identify and treat their ailments.
- Write and upload prescription - Upload a prescription to the website that will be made available only to that particular patient.

5: Do a rough estimate of effort required to accomplish each task in terms of person months.

We will be using CoCoMO (Constructive Cost Model) for estimating the effort required to accomplish each task in terms of person months. It is known as procedural cost estimate model for software oriented projects and often used for predicting various parameters associated with the project such as effort, cost, size, quality and time. We have **3 kilo lines of code (KLOC)**.

$$E = a(KLOC)^b$$

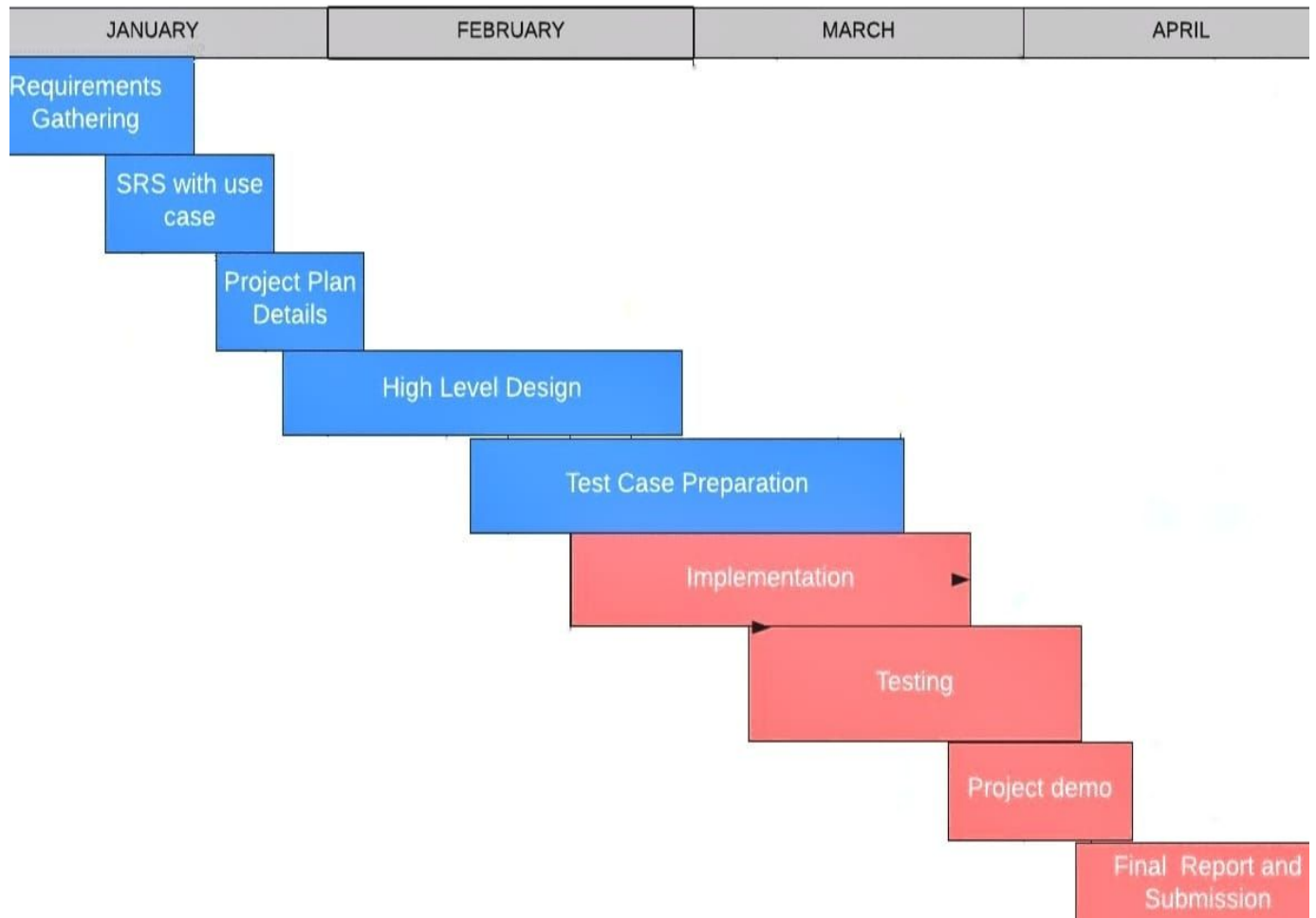
Our project falls under the basic and organic type of CoCoMo model as our team size is small (3 members) and here we are doing rough estimation. For the above equation, **a = 2.4, b = 1.05**

Substituting all the values in the above formula, we get

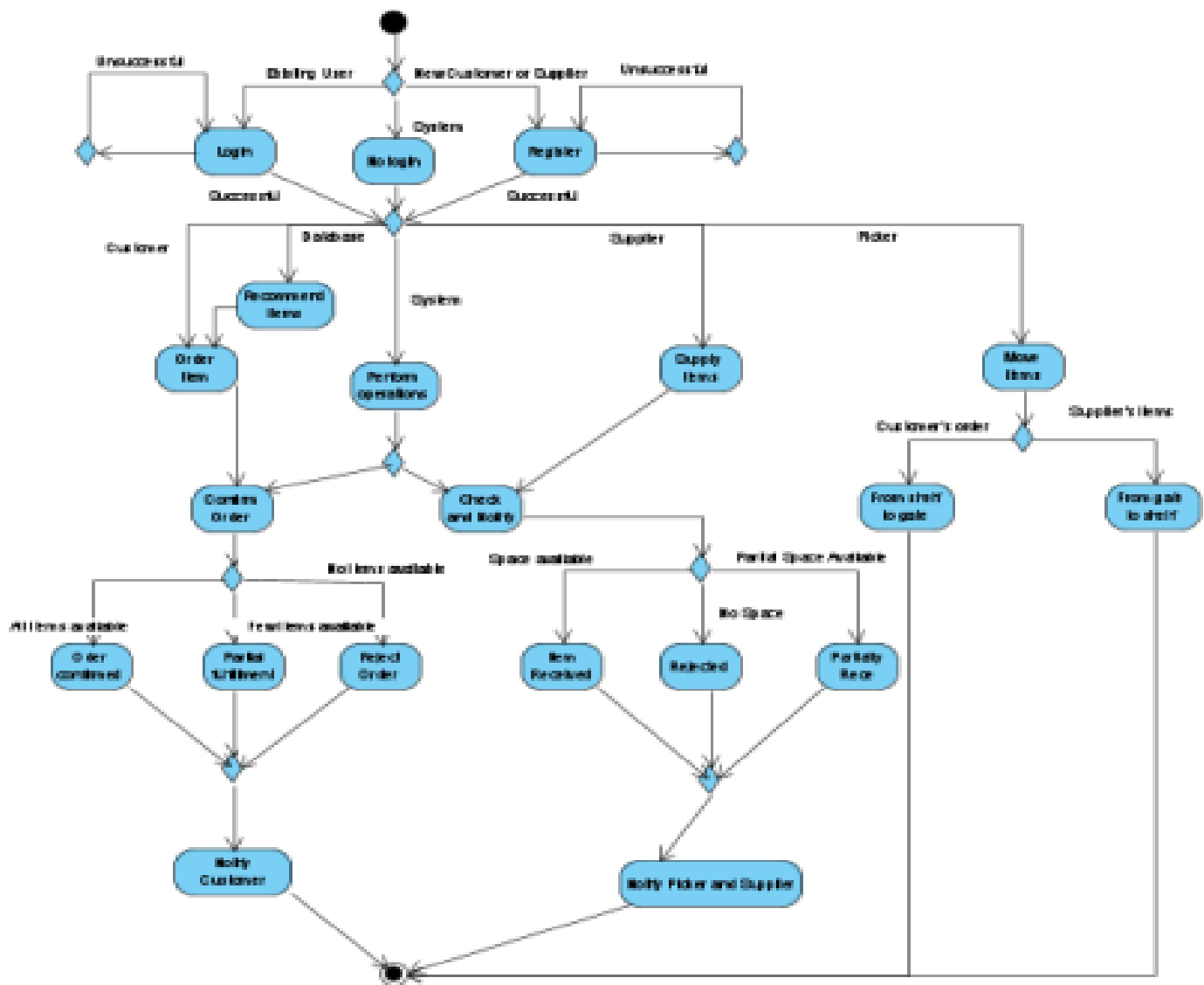
$$E = 2.4 * (3 ^ 1.05) = 7.606 \text{ person-months}$$

So the total estimated effort is 7.6 person-months.

6: Create the Gantt Chart for scheduling.

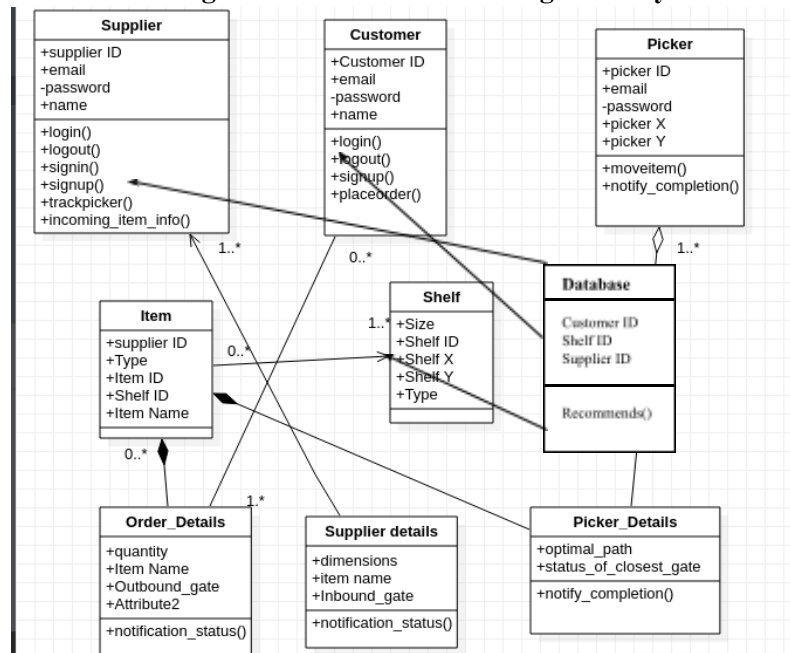


Activity Diagram for Warehouse management system



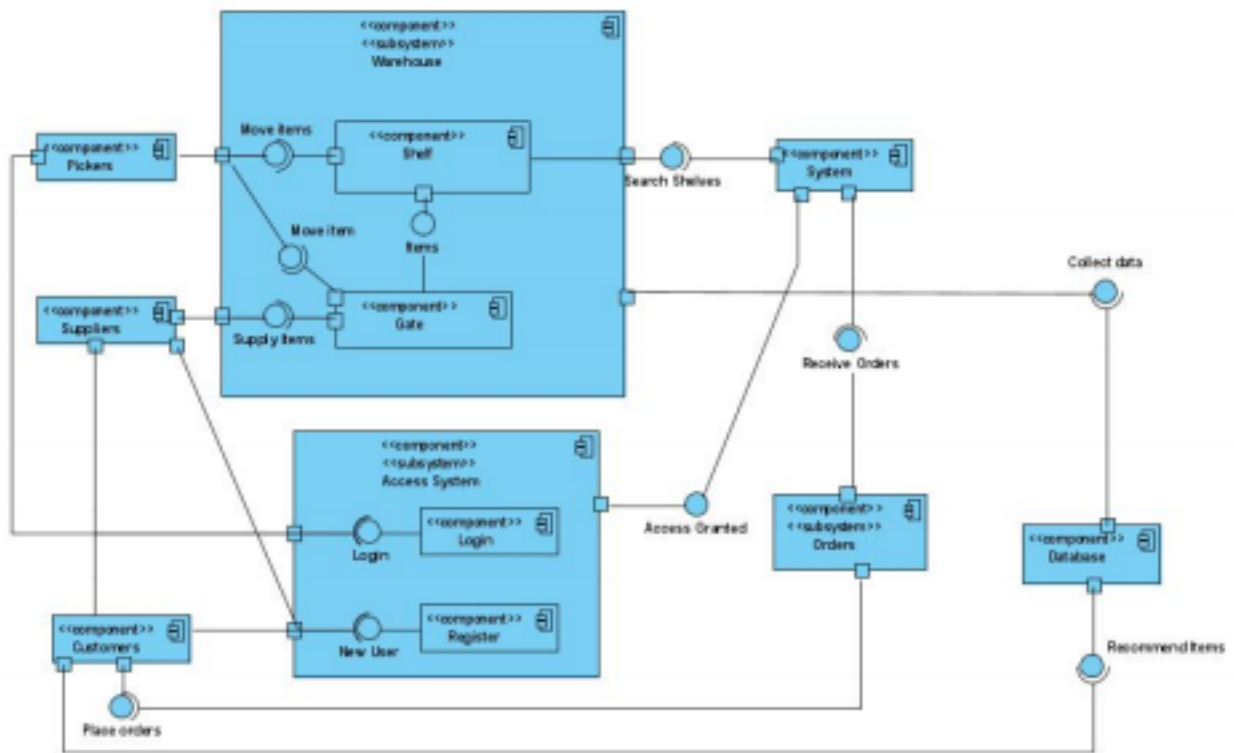
An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state).

Class Diagram for Warehouse Management System



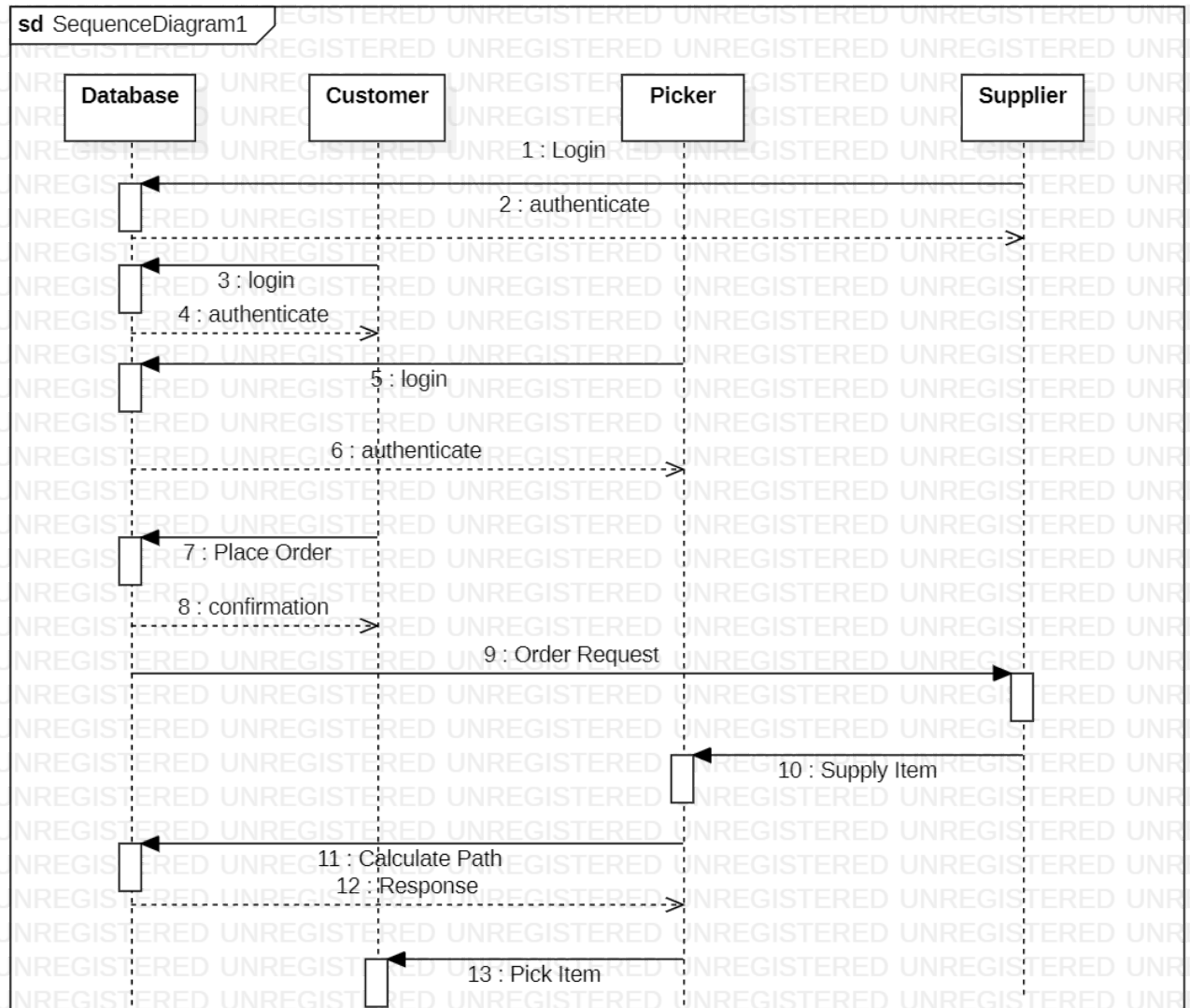
The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.

Component Diagram for Warehouse Management System



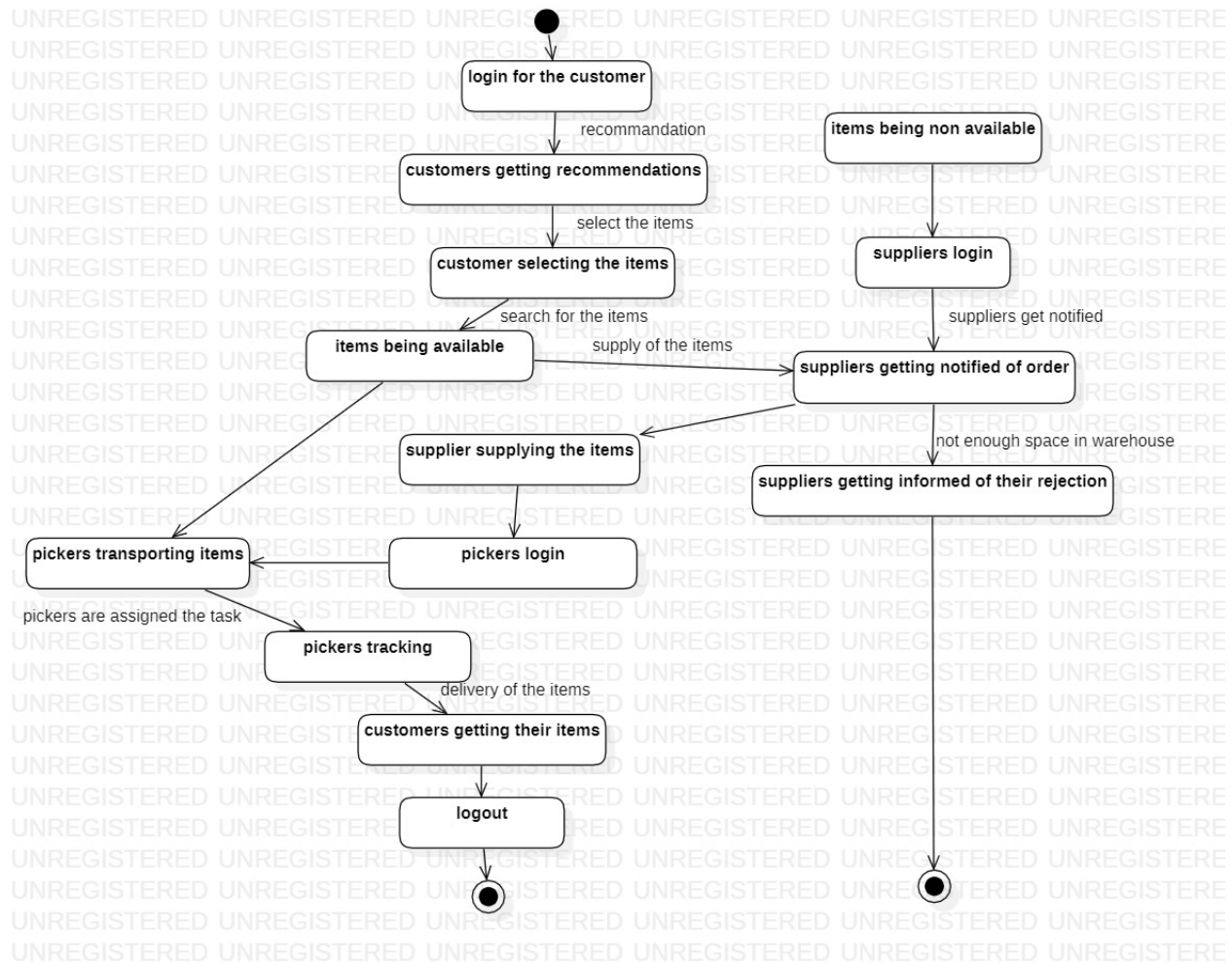
A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

Sequence Diagram for Warehouse Management System



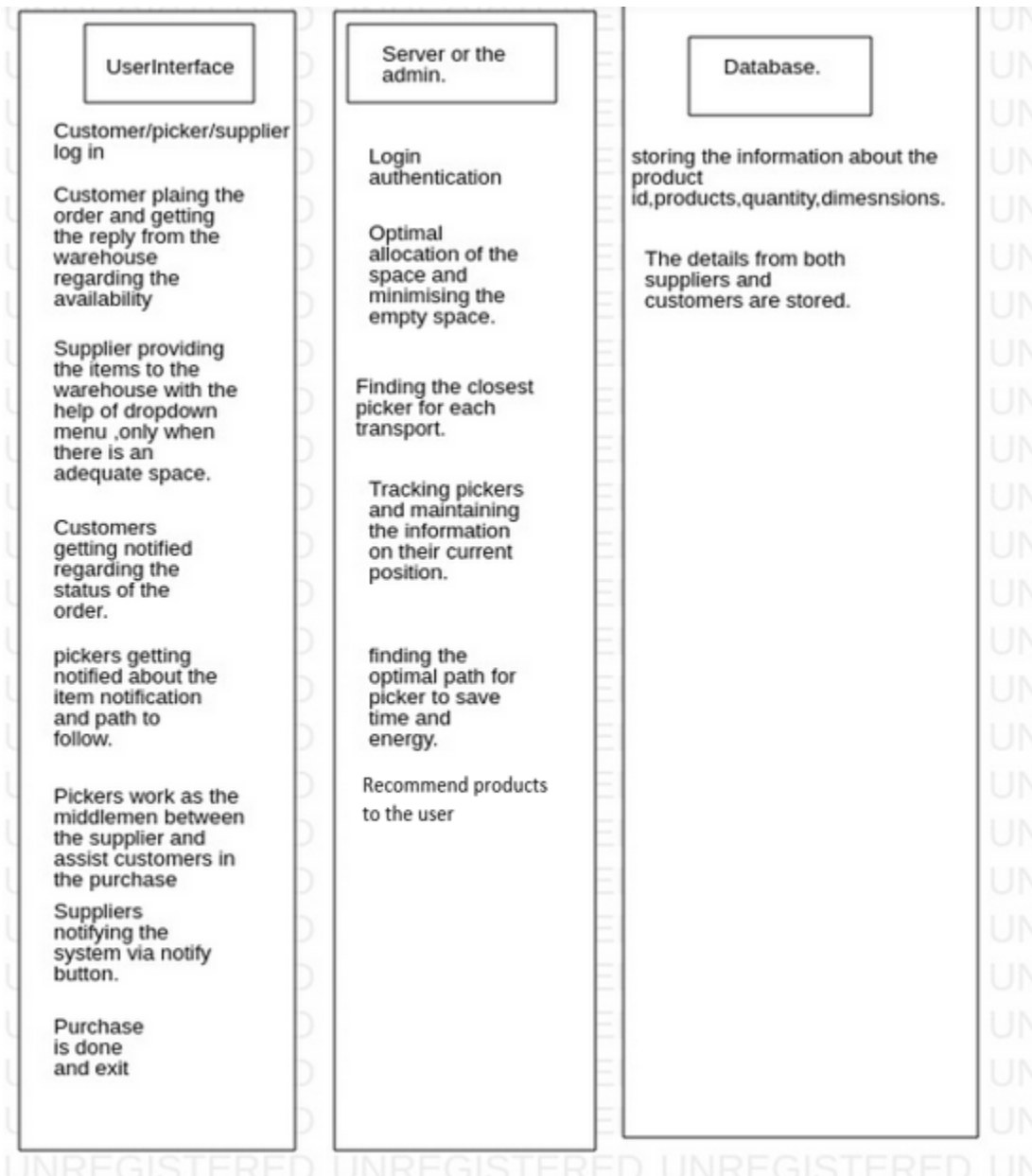
Sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.

State-chart diagram for warehouse management system.



A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction.

System Architecture Diagram for Warehouse Management System



System architecture gives an overall view of the project. It depicts the system behavior. It helps to understand the behavior of all the components and also helps to develop the project further.

A brief description of each use case implemented

Supplier:

Login: Supplier needs to enter into the warehouse management desktop app and he should be knowing the required password.

Add new product: Supplier has to enter new items into the warehouse based on customer demand.

Display Stock: Supplier needs to show/see the items present in the warehouse to verify if item addition/removal has taken place successfully.

Refill: Supplier needs to refill the products if the products get finished in the warehouse to meet the user demands.

Remove an item: Supplier has the right to remove a particular item after the product is not in demand or if the product is damaged.

Exit: At any time, the user can exit from the supplier module.

Customer:

Login: Customers who want to enter into the warehouse need to log in to the system and are given a password for the purpose of security.

Purchase: Customers can purchase the items based on their requirements.

Display stock: Customers can view stocks before purchasing to check if the item they need is present in the warehouse.

Exit: At anytime user can exit from the customer module.

Picker:

Login: Every employee of the warehouse needs to log in to the system and is given a password for security purposes.

Display Stock: Picker should display the stock present in the warehouse for convenience.

Refill: Since the picker is the middleman between the supplier and customers, he/she needs to refill the items based on the user demand.

Exit: At anytime user can exit from the picker module.

Recommendation System: Based on the items sold or the products demanded by the customers, the picker recommends the products to the user.

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UT_01-login.	supplier	User needs to enter into the warehouse management desktop app and he should be knowing the required password	supplier should have a proper login credentials.	After entering the designed desktop app ,user needs enter the password after selecting the choice as supplier so that he is given all the privileges .	A password provided for the supplier module.	The user should be able to enter the desktop app if he has the correct password .	The user was able to enter the desktop app with the right password .	Pass
UT_02- Add item	supplier	Supplier has to enter the items based on customer demand.	The supplier must have a good knowledge of the price,quantity of the products to be added.	After logging in as supplier, the user needs to select the option-1 to add items.	Any item with its price and quantity.	The user as a supplier should be able to add items to the warehouse.	The user with access to the supplier module could add the items to the warehouse with the quantity as well as the price.	Pass
UT_03-display stock	supplier	Supplier needs to show the items present in the warehouse.	Database should be up to date.	The user needs to select the option 2 in the supplier module.	The information of the items which have been added	The supplier must be able to display the available stocks on	The supplier was able to display the stock on screen.	Pass

					previo usly.	the screen.		
UT_04-Refill	supp lier	Supplier needs to refill the products if the products get finished in the warehous e to meet the user demands.	The supplier should have a proper knowledg e of the items to be added	After entering the designed desktop app ,user needs to select option 3 of the supplier module.	Any item in dema nd with the price and quanti ty.	The supplier must be able to refill the items in the warehou se correspo nding to the user demands .	The user was able to find out the user demands and refill the stocks.	Pas s
UT_05-Remov e an item.	supp lier	Supplier has the right to remove a particular item after the product is not in demand or if the product is damaged.	Supplier should have a proper understa nding of the warehou se trends.	After entering the designed desktop app ,user needs to select option 4 of the supplier module.	Any item which is not requir e for the custo mers.	The supplier needs to remove an item which is not in demand so that other items can be filled in the warehou se.	The supplier was able to remove a particular item at a time.	Pas s
UT_06-Exit	supp lier	At anytime user can exit from the supplier module.	None.	The user can exit from the supplier module at any point of time by selecting choice of the supplier module.	None.	The supplier needs to take an exit from the module.	The supplier could easily exit from the module	Pas s
UT_07-login.	picke r	Every employee of the warehous e needs to log in to the	Picker should have the login credentia ls.	The user needs to enter into the desktop app and select	A passw ord provid ed for the picker	The picker should able to login to the picker	The user with the correct password was able to enter the	Pas s

		system and is given a password for the security purpose.		the picker module and enter the correct password to get into the module.	module.	module with the correct password .	picker module.	
UT_08_Recommenadtion of the products to the customers.	picker	Based on the items sold or the products demanded by the customer s,picker recommends the products to the user.	Sales must happen and all the details should be entered in the database.	Based on the data stored in the database, the system on picker side should suggest the top products to the customer .	The items stored in the database.	The picker should recommend the products to the customer based on the past history.	The picker was able to recommend the products to the customer s based on the quantity sold.	Pas s
UT_09-Display stock.	picker	Picker should display the stock present in the warehouse for convenience.	Database must be up to date	After entering into the picker module of the desktop app,one needs to select option 1 of the module.	The items present in the database at that particular time along with the price and quantity.	The picker should be able to display the stocks so that the customer s can get a fair idea.	The picker was able to display the stocks to the customer s.	Pas s
UT_10-refill	picker	Since picker is the middleman between the	Picker must be knowing the ground situations and	User needs to select option 2 of the picker module	The items in demand with their	As the picker is the middleman between the	The picker was able to refill the stocks.	Pas s

		supplier and customer s,he needs to refill the items based on the user demand.	should be capable of analysing the sales.	to refill the stocks.	quantity and price.	supplier and customer he needs to understand the situation on both the sides and refill the stocks according to the demand.		
UT_11-Exit	picker	At anytime user can exit from the picker module	None.	The user can exit from the picker module at any point of time by selecting choice 3 of the picker module.	None.	The picker needs to take an exit from the module.	The picker could easily exit from the module	Pas s
UT_12-login	customer	Customer s who want to enter into the warehouse needs to log in to the system and is given a password for the security purpose	Customer should have proper login credentials.	After entering the desktop app,One needs to select customer module and enter the password .	A password provided for the customer module.	The customer must be able to log in to the system using the password .	The login option is not provided here.	fail
UT_13-purchase.	customer	Customer s can purchase the items based on their	The customer must be given a chance to look at	After selecting option 2 of the customer module	Items available in the database	The customer must be able to purchase the items	The customer could purchase the items by	Pas s

		requirements.	the various items available and then make a decision.	one needs to enter the products and their quantity.	along with their price and quantity.	in the warehouse.	entering the quantity of the items needed.	
UT_14- Looking at the available stocks	customer	The customers are given a privilege to view the stocks.	The database must be up to date.	The user needs to select option 2 of the customer module.	The items stored in the database.	The customer must be able to view the stocks.	The customer was able to look at the available stocks.	Pass
UT_15-payment	customer	Various payment methods are provided to pay for the items purchased.	Customer must purchase the products to pay.	After entering into the desktop app, the user needs to select the option in the customer module.	The prices for the items purchased along with the quantity.	User needs to pay the amount to the warehouse by any of the payment methods.	The payment option is not enabled.	Fail.
UT_16-exit	customer	At anytime user can exit from the customer module	None.	The user can exit from the customer module at any point of time by selecting choice 3 of the customer module.	None.	The customer needs to take an exit from the module.	The customer could easily exit from the module	Pass
UT_17-Feeding input	picker	The data stored in warehouse database is fed into the recommendation system.	Sufficient data must be given as the input to the recommendation system.	The data stored in the csv file should be given as input to the recommendation	The items with their price and quantity which are	The data stored in the database should be fed into the recommendation system.	The data stored in the database was fed into the recommendation system.	Pass

				system code.	stored in the database.			
UT_18-Collecting the user reviews	picker	The customer is asked to give the ratings on the products he/she has purchased	Customer must buy the products.	The customer is offered a google form to be filled.	Ratings given on a scale of 1-10 for the items purchased.	The picker should be able to collect reviews from the customers.	No such interactive option was provided in the picker module.	Fail.
UT_19 - Recommending based on the quantity of products sold	Picker	High sale of a particular product implies that other users are likely to buy that item more in future. So this can be used as a measure to recommend the products to the users	Products sold must be stored in a database.	The results of the sales are stored in the database. we need to give the database as an input to the recommendation system written in a .ipynb file.	The items present in the database along with their quantity and price.	The picker should recommend the products based on the quantity of the products sold.	The picker was able to recommend the relevant products to the customers based on the number of products sold.	Pass
UT_20- Recommending the optimal path for the picker.	supplier	Out of all possible paths for the picker, the system recommends the optimal path to increase	Supplier must have a good knowledge of the all possible paths that a picker may take.	After logging in to the supplier module, the supplier should look at the list of paths available.	The list of all paths that a picker can follow to reach the custo	The supplier should choose the optimal path for the picker to follow.	The supplier could not choose any optimal path for the picker to follow.	Fail.

		the efficiency.			mer from the supplier.			
UT_21-recommending the least popular items.	Picker	The user also needs to know about the products which are least likely to be recommended.	Database should be up to date.	After storing the data in the database, it is given as an input to the recommendation system in a .ipynb file. The items which are at the bottom of the recommended products indicate that they are least recommended items.	The items present in the database along with their price and quantity.	The picker should also be able to recommend the least popular items to the customer.	The picker was able to show the products which are unlikely to be purchased.	Pass
UT_22-Recommendation of the items based on a huge dataset provided the warehouse is not overfilled.	picker	Recommendation of products are decided by considering a huge data set in order to get the better results.	Presence of a dataset.	After storing the data in the database, it is given as an input to the recommendation system in a .ipynb file. There is no limit on the size of database	The items stored in a dataset.	The picker should recommend the products based on the data taken from a huge data set to yield better results.	The picker was able to recommend the products even when there was a huge data set.	Pass

				that can be stored until the warehouse is full.				
UT_23-Recommendin g top 5 items.	Picke r	User tries to look at the top suggested items so that he/she can get a clear cut idea.	Presence of a updated database.	After storing the data in the database, it is given as an input to the recommen dation system in a .ipynb file.	The items which have the highest sales.			Pas s
UT_24-providi ng various discounts.	supp lier	The customer s are provided with various discounts from the supplier.	None.	After logging into the supplier module one needs to look at the choice given to provide any sort of discounts to the customer .	The items along with the wareh ouse trends .	The supplier should occasion ally provide discounts on a particular group of items.	No such provision for discount was given.	Fail .
UT_25-Remov ing more than one product at a time.	Supp lier.	After logging in to the supplier module,t he user tries to remove more than one item at once.	Items should be there in the warehou se.	After logging into the supplier module one needs to select option 4 to remove items.	The items not requir e	The supplier should be able to remove more than one item at a time to increase the efficiency .	The supplier could not remove more than one item at a time.	Fail .

UT_26								
UT_27								
UT_28								
UT_29								

This document for warehouse management was prepared by Navya Parameshwar
Hegde-PES2201800017(test cases 1-8)

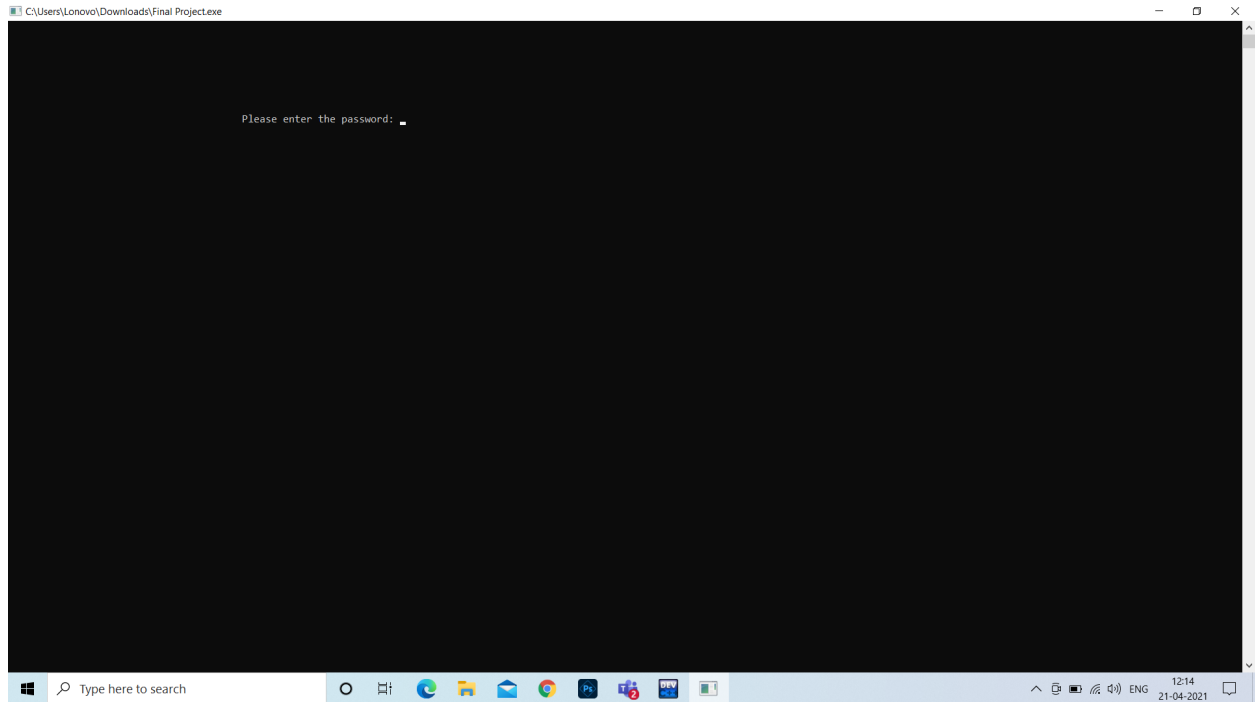
Nayana C S reddy-PES2201800510(test cases 9-16)

Shloka lakka-PES2201800502(test cases 17-24)

Relevant Outputs

Supplier

Login:



Supplier Menu:

```
C:\Users\Lonovo\Downloads\Final Project.exe

=====
SUPPLIER MENU
1. Add new product
2. Display stock
3. Refill
4. Remove an item
5. Exit:

=====END OF MENU=====

Enter your Choice :
```

Add Item:

```
C:\Users\Lonovo\Downloads\Final Project.exe

Enter the No. of Products that you wish to add: 5

Input the name, price and the quantity of item respectively
pen 5 4

item updated

Input the name, price and the quantity of item respectively
pencil 8 7

item updated

Input the name, price and the quantity of item respectively
notebook 9 10

item updated

Input the name, price and the quantity of item respectively
candy 15 17

item updated

Input the name, price and the quantity of item respectively
chips 45 67

item updated
Stock Updated!!!
```

Display Stock:

```
C:\Users\Lonovo\Downloads\Final Project.exe

=====
THE STOCK ITEMS ARE
=====

PARTICULARS   STOCK AVAILABLE   PRICE
=====
pen           4                 5
pencil        7                 8
notebook      10                9
candy         17                15
chips         67                45
=====
```

Refill:

```
C:\Users\Lonovo\Downloads\Final Project.exe

=====
THE STOCK ITEMS ARE
=====

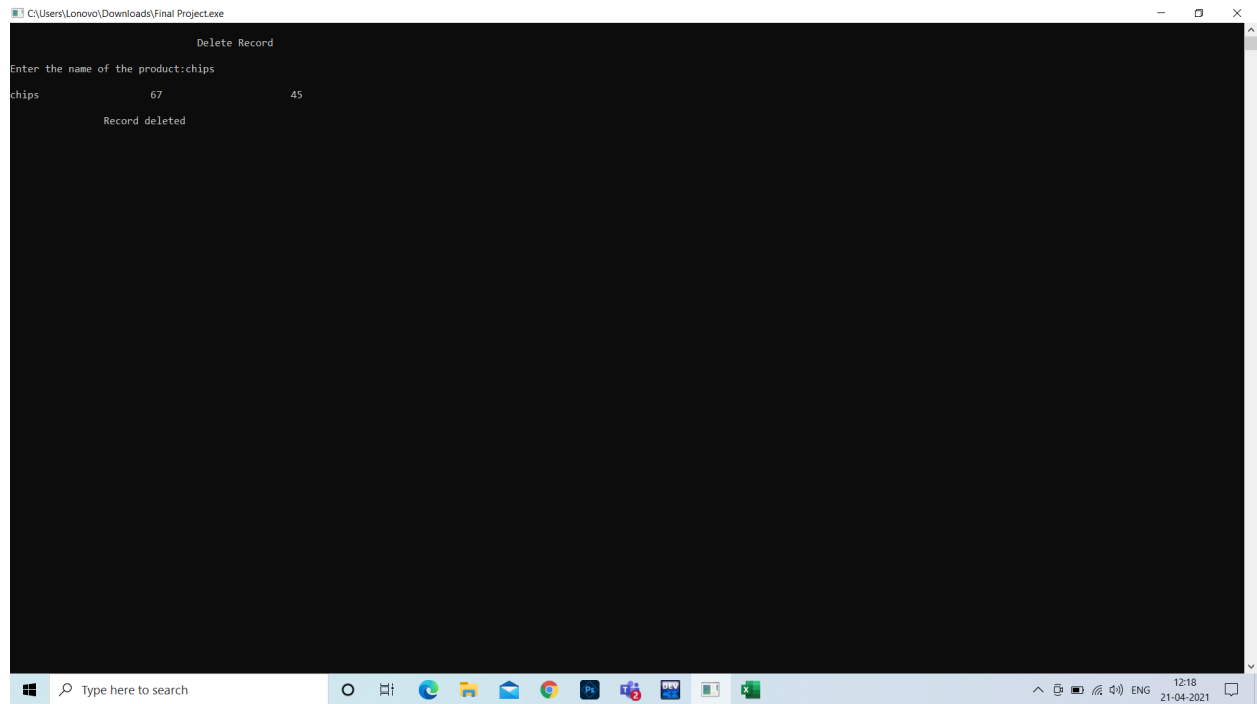
PARTICULARS   STOCK AVAILABLE   PRICE
=====
pen           4                 5
pencil        7                 8
notebook      10                9
candy         17                15
chips         67                45
=====

Enter the products name
pen

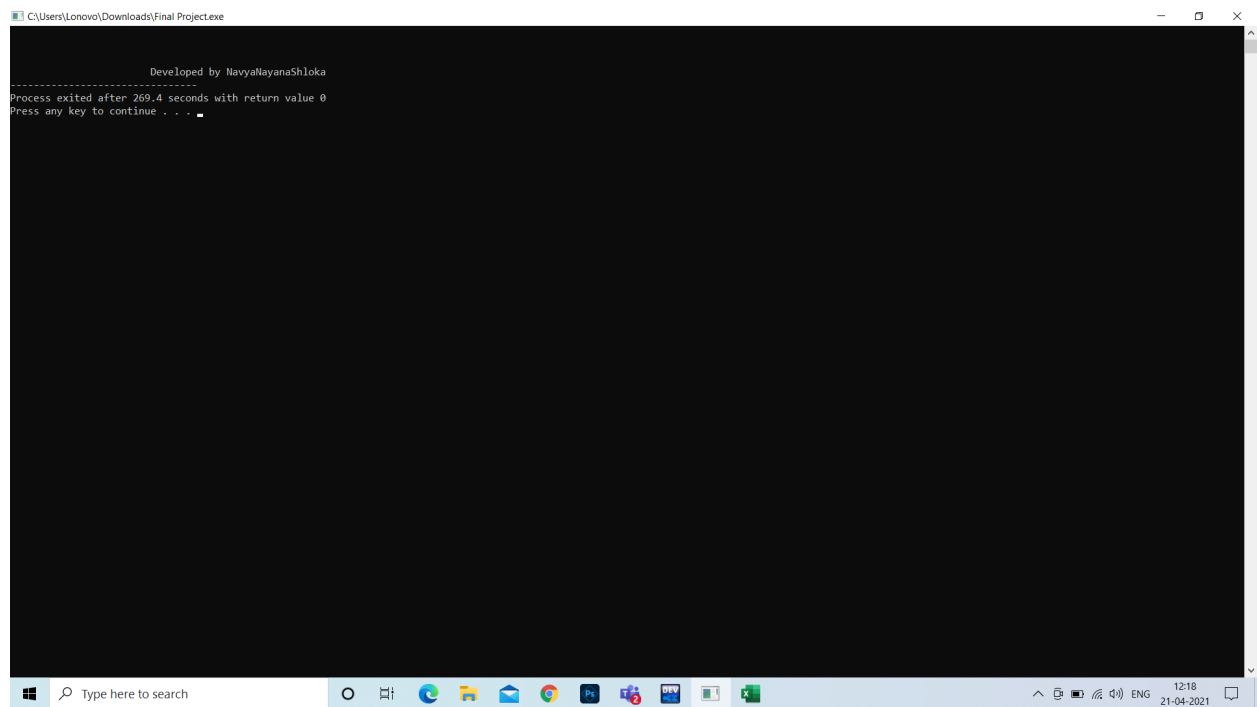
Enter quantity:
5

Stock updated.
```

Remove Item:

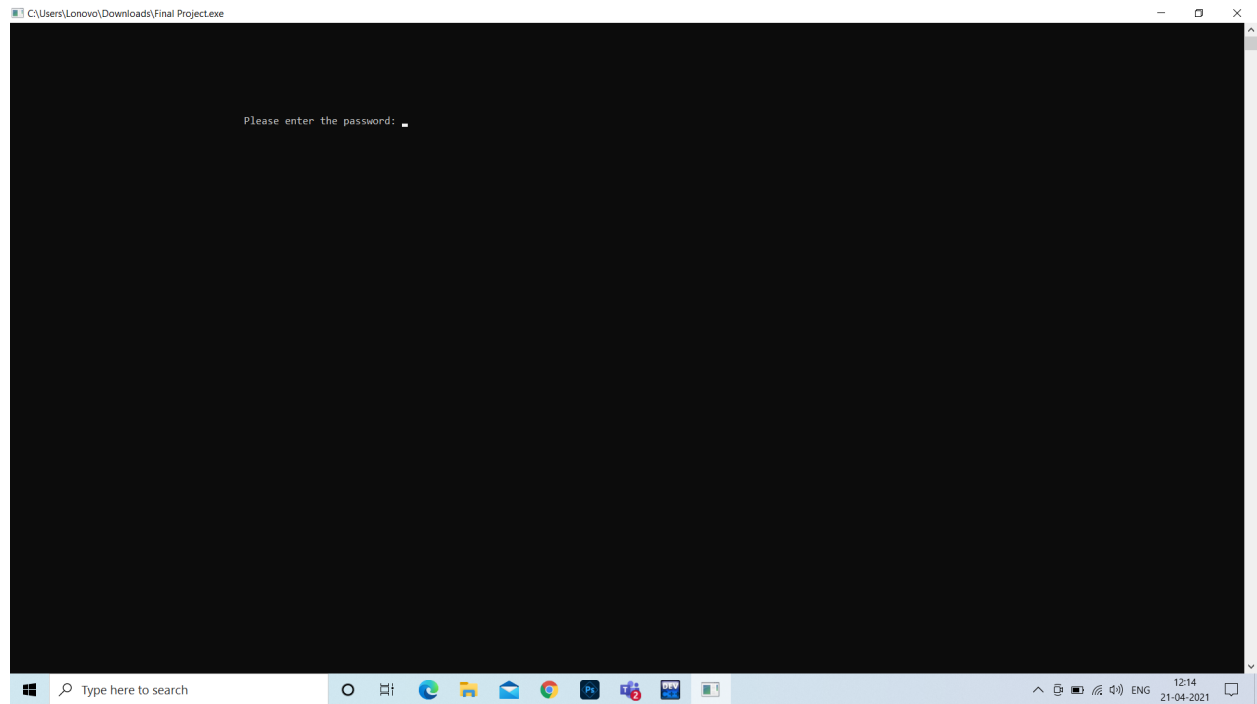


Exit:

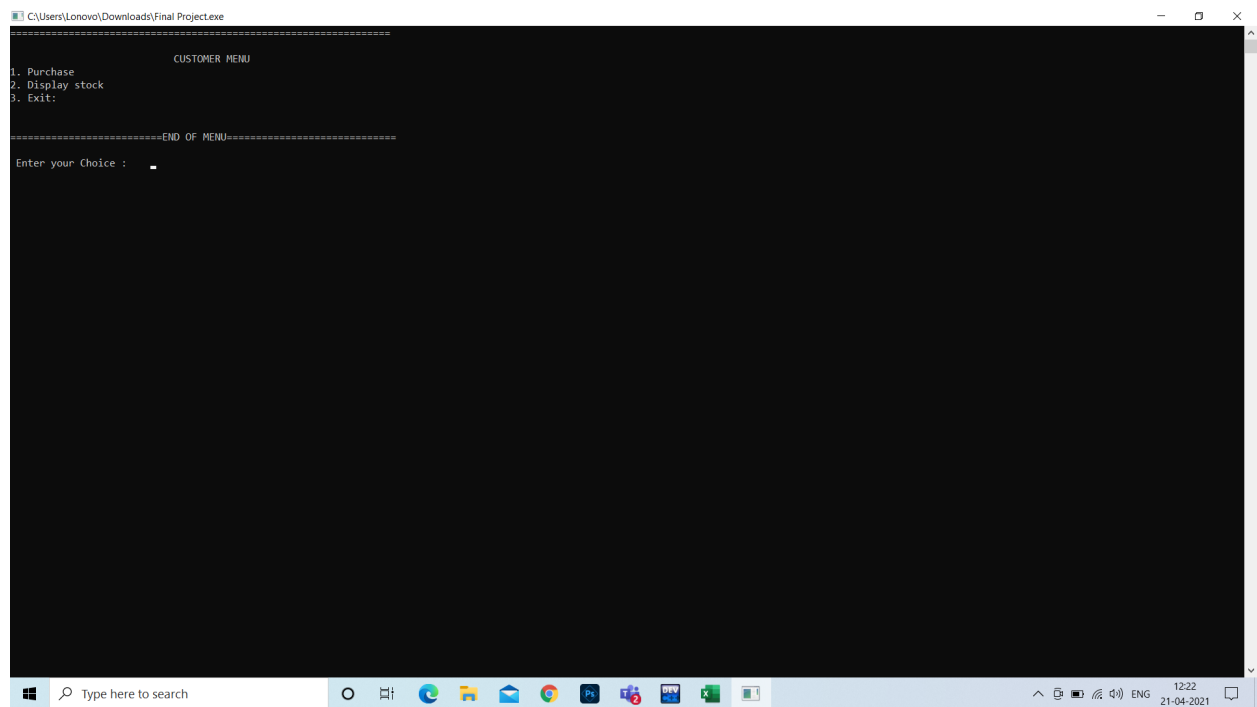


Customer

Login:



Customer Menu:



Purchase:

```
C:\Users\Lonovo\Downloads\Final Project.exe

=====
THE STOCK ITEMS ARE
=====

PARTICULARS   STOCK AVAILABLE   PRICE
-----
pen           9                 5
pencil        7                 8
notebook      10                9
candy         17                15

Enter the product's name
pen

Enter quantity:
4

Stock updated.

Total price to be paid:20
```

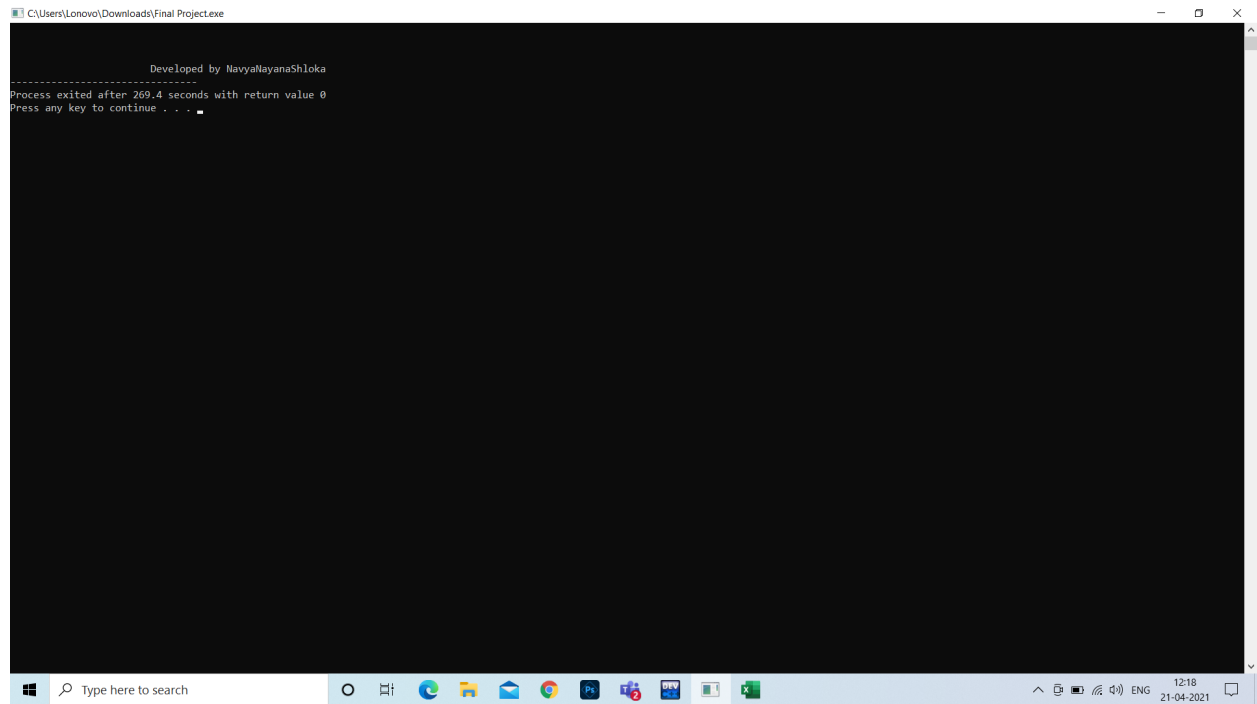
Display Stock:

```
C:\Users\Lonovo\Downloads\Final Project.exe

=====
THE STOCK ITEMS ARE
=====

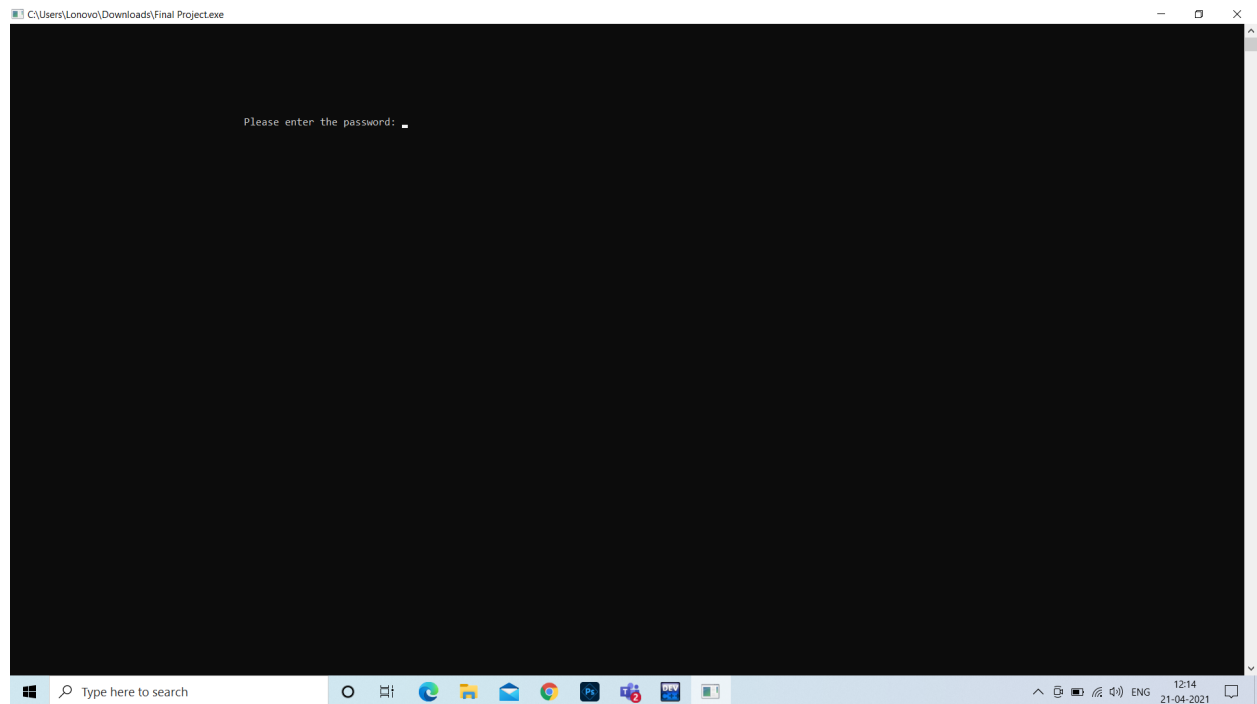
PARTICULARS   STOCK AVAILABLE   PRICE
-----
pen           4                 5
pencil        7                 8
notebook      10                9
candy         17                15
chips         67                45
```

Exit:



Picker

Login:



Picker Menu:

```
C:\Users\Lonovo\Downloads\Final Project.exe

=====
PICKER MENU
1. Display stock
2. Refill
3. Exit
=====END OF MENU=====

Enter your Choice : 1
```

Display Stock:

```
C:\Users\Lonovo\Downloads\Final Project.exe

=====
THE STOCK ITEMS ARE
=====

PARTICULARS    STOCK AVAILABLE    PRICE
=====
pen            5                5
pencil        7                8
notebook      10              9
candy         17             15
```

Refill:

```
C:\Users\Lonovo\Downloads\Final Project.exe

=====
THE STOCK ITEMS ARE
=====

PARTICULARS    STOCK AVAILABLE    PRICE
-----
pen            4                    5
pencil         7                    8
notebook       17    10            15    9
candy          67                    45
chips          67                    45

Enter the products name
pen

Enter quantity:
5

Stock updated.
```

Exit:

```
C:\Users\Lonovo\Downloads\Final Project.exe

Developed by NavyaMayanaShloka
-----
Process exited after 269.4 seconds with return value 0
Press any key to continue . . .
```

Recommender System:

shop - Excel

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

Clipboard: Paste, Cut, Copy, Format Painter

Font: Calibri, 11, Bold, Italic, Underline, Text Color, Fill Color

Alignment: Wrap Text, Merge & Center

Number: General, Percentage, Decimal, Fraction

Styles: Conditional Formatting, Format as Table, Cell Styles

Cells: Insert, Delete, Format, Clear

Editing: AutoSum, Fill, Sort & Filter, Find & Select

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Items	Quantity	Price																				
2	pen	5	4																				
3	pencil	8	7																				
4	notebook	9	10																				
5	candy	15	17																				
6																							
7																							
8																							
9																							
10																							
11																							
12																							
13																							
14																							
15																							
16																							
17																							
18																							
19																							
20																							
21																							
22																							
23																							
24																							
25																							
26																							
27																							
28																							
29																							

shop

shop - Notepad

File Edit Format View Help

```
pen 5 4
pencil 8 7
notebook 9 10
candy 15 17
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8 | 12:27 21-04-2021

Home Page - Select or create a notebook | X | converting_dat_csv - Jupyter Notebook | X | +

localhost:8888/notebooks/converting_dat_csv.ipynb

jupyter converting_dat_csv Last Checkpoint: 14 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [3]: import csv

with open("shop.dat") as infile, open("shop.csv", "w") as outfile:
    csv_writer = csv.writer(outfile)
    prev = ''
    csv_writer.writerow(['Items', 'Quantity', 'Price'])
    for line in infile.read().splitlines():
        csv_writer.writerow([line, prev])
        prev = line
```

In []:

Type here to search

12:29 21-04-2021

Home Page - Select or create a notebook | X | recommendation_system - Jupyter Notebook | X | +

localhost:8888/notebooks/recommendation_system.ipynb

jupyter recommendation_system Last Checkpoint: 14 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [2]: import numpy as np
import pandas as pd
df=pd.read_csv("shop.csv")
df.head()
df.groupby('Items')['Quantity'].mean().sort_values(ascending=True).head()
```

Out[2]:

Items	Quantity
pen	5
pencil	8
notebook	9
candy	15

Name: Quantity, dtype: int64

In []:

Type here to search

12:29 21-04-2021