“Stocked – a complete organizing solution”

REVIEW REPORT

Submitted by

PRASON POUDEL (19BCE2550)

MICKEY KUMAR ROUNIYAR (19BCE2520)

SASHANK RIJAL (19BCE2484)

Prepared For

SOFTWARE ENGINEERING CSE3001

J COMPONENT

Submitted To

SHASHANK MOULI SATAPATHY

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Table of Contents:

1. Abstract:

It is very essential to store, organise and make quickly available the necessary data required for the well organised and effective functioning of any business. From a small proprietary business like a simple shop to multi-billion dollar companies like Google, Facebook, etc., it is really necessary to organise, tabulate, store and secure the data in an efficient manner.

If we look into our community it is a general perspective that the software for keeping track should only be used by high end companies and so on. We can also see many software's are used in big shops such as supermarkets and departmental stores. But, we often see that the organization is often kept manually or sometimes not kept in the smaller shops. So, looking at the scenario we have decided to create a software that can run on a very low specifications and would be suitable for a smaller shop. It will help the shops to get insights on their organizing techniques and help for their betterment. Our "Stocked" software can help address employee count scenario and a range of other complex situations related to cost, demand, organization, tabulation, storing and security by presenting the same data to everyone in the business at the same time. We want to design a software that will help to keep track of stocks and will also provide insights about the business based on the stocks, demands, sales, recent trends, profit or loss and so on.

2. Introduction

Stocked is a standalone software which can be used in supermarket or any kind of shop as a cashier software. Our "Stocked" software can help address employee count scenario and a range of other complex situations related to cost, demand, organization, tabulation, storing and security by presenting the same data to everyone in the business at the same time. We want to design a software that will help to keep track of stocks and will also provide insights about the business based on the stocks, demands, sales, recent trends, profit or loss and so on.

In general, our objective is to include the following features on our application:

- keep a record of stock in the shop
- calculate profit or loss
- notify the shop owner when any item has crossed its expiry date
- Aware the shop owner if an item is trending
- keep record of the employee in the shop
- provide yearly review such as which item was sold more in which month.

1. Layout of the application:

Here we will create the user interface. We are going to create it in python with the help of PYQT. PYQT is an external module in python and is nowadays used more than tinkter. The user interface will be made as interactive as possible. It will be a combination of text fields, buttons and small display boxes.

2. Stock Tracker:

As soon as the shopkeeper buys some things then it will be updated in the database. The items bought has to be manually entered into the system. Similarly, as soon as any item is sold it will be reduced from the database.

3. Trending items:

In this project, we describe trending item as the item that is being rapidly sold and we define not to invest item as the item that is in the stock for a very long time and is barely sold. This application will be smart enough to classify the products into trending or not. According to the classification, the application will suggest the shopkeeper if the shopkeeper should increase the stock of items or not.

4. Profit/loss generator

Every owner should be made aware about his /her business. He /She should know whether he is in profit or in loss so that they can take necessary steps to make their business more fruitful. So, one of the modules in our application will calculate daily profit and loss which can be checked by owner. Similarly, it will calculate the yearly profit or loss.

5. Invoice Creator

It will be better if the stock management system has an invoice generator. If this happens, the shopkeeper doesn't have to manually enter the record of sales daily. So, Invoice creator will create the bills and with the help of stock tracker module, it will modify the database. Similarly, all the generated bills will be stored for future reference.

6. Expiry date checker

It is a tedious work to check the expiry date of the products in a shop. It requires extra manpower and money. As our application is already keeping record of all the products, it will also keep track of the expiry date and will check with the system calendar and automatically notify the users.

7. Employee

It is obvious that a shop will have employees. So, a system is needed to keep the data of the employee and to check if his/her salary is paid or not. This feature will be inbuilt in our system.

8. Login

There will be owner and employees in the shop. Each will have their own login credentials.

9. Revenue generator

It will access data from all the above modules and will mention which month was the most profitable, which item to be sold on which month.

3. Background / Related Work

Authors	Method	Purpose	Advantages	Disadvantages
R. Elmasri & S. B. Navathe	Implementing SQL in Python	Information on Relational Database and levels of Database	Helps to understand implementing sql in python	Requires external pyqt modules.
Thomas Connolly, Carolyn Begg	Database Design	Basic Concept on how to design an efficient database.	Helps in efficient database design.	Lacks about the information on programming the database.
Mark Summerfield	Programming and implementation.	Reference for a programmer	Helps to learn Graphical interface programming	Very lengthy code is generated.
Lockard, Robert (29 November 2010)	How a business can implement Inventory Management and how it is benefitted.	Use of Inventory Management Software in various applications.	Different businesses are benefitted using the system.	The implementation becomes highly expensive.

Utkarsha Mendhe, Ankita Lohave, Aayush Sah, Varun Pahwal, Prof. Nutan Sonwane	inventory system was made by them and studied	Bill Generator and Inventory Maintenance	Shows the way of maintaining the inventory in a systematic order in accordance to the software	It adopts a particular way of bill generation that is not suitable for very small data sets
A Rohini, Wubshetasamnew Woldeyohannshiluf and Teshometeklemichea l Walelignayimelo	Studied client server database inventory system	Inventory Management and organisation.	Gives way of organizing the inventory	It sticks to a particular organization system
Lesonsky, Rieva (1998) Entrepreneur Magazine	This paper gives a brief outline on how an inventory should work.	Tracking Inventory	Gives guidelines on how an inventory should work	-

4. Real-life Applicability

If we look into our community it is a general perspective that the software for keeping track should only be used by high end companies and so on. We can also see many software's are used in big shops such as supermarkets and departmental stores. But, we often see that the organization is often kept manually or sometimes not kept in the smaller shops. So, looking at the scenario, "Stocked" software can run on a very low specifications and would be suitable for a smaller shop.

Considering the situation all the local shop owners living in the extreme areas of Himalayas and hills have no access to internet. Also they don't have any high specification in their device. Since management of shop is important for any shop owner and they should be aware about every stock details like expiry date, price, profit and loss and other essential details like which item produces more profit and which item doesn't, this will help all the local shopkeepers to manage and run their shops efficiently.

Hence, our software and be used by any local shopkeeper who don't have any access to internet to manage their stock, manage all the employee details, manage all the customer details and to get insight about their shop based on revenue generated, trending list, profit and loss and also aware the owner about the products that is least sold and not to invest in it along with information about expired items and give warning about the items which are going to be expired within a month.

5. Individual Contributions

Register number	Name	Contribution
19BCE2550	**Prason Poudel**	**Mainly frontend design, partial backend implementation and documentation**
19BCE2520	**Mickey Kumar Rouniyar**	**Mainly backend implementation, partial frontend design and documentation**
19BCE2484	**Sashank Rijal**	**Mainly database design and connectivity, partial frontend and backend design with documentation**

6. Tools and Technologies used / Hardware – Software Requirements

Software Requirements

Operating System:

1. Windows 7 with server pack 2 (or)
2. Windows 8.1 or higher (or)
3. Linux / MacOS

Software's/Modules:

1. Python IDE
2. PyQT python module
3. DB Browser for SQlite2

Hardware Requirements

Processor: Intel Core Duo 2.0 GHz or more

RAM: 1 GB or More

Hard disk: 50GB or more

Monitor: 15" CRT, or LCD monitor

Keyboard: Normal or Multimedia

Mouse: Compatible mouse

7. Proposed System Process Flow

In this project, waterfall model will be appropriate for our overall development of software and project. The first phase of the waterfall model is a Requirements in which the developers of the system needed to know what are needed in the system or what is really the clients want to that system in order to meet their satisfaction. Once the client is satisfied for the use of waterfall model, the developers can draw up the specification document with some assurance that the product meets the client's real need. So therefore, proponents used the waterfall methodology for our project "STOCKED-A COMPLETE ORGANIZING SOLUTION" due to the following reasons:

Clarity of User Requirements

The waterfall model assures that the software meets the user requirements. The working prototype has been validated trough the interaction with the client, it is reasonable to expect that the resulting specification is based on the user requirements. It is well suited for our project as requirements are well defined.

System Reliability

With the use of waterfall model, we can develop our project in a stable and reliable manner as it is convenient model and the phases of this model will help to develop our project in a better way. Also, the project is completely dependent on project team with minimum client intervention so it gets easier for the team to develop the project.

Schedule Visibility

Since every phase of this model means completion, so this model will be suitable for the development of our project. Since waterfall model is a sequential software development process with less system complexity, where progress flows steadily toward the conclusion (like a waterfall) through the phases of a project (that is, analysis, design, development, testing). This involves fully documenting a project in advance, including the user interface, user stories, and all the features' variations and outcomes which matches with our project so this model will be appropriate for our "STOCKED" software

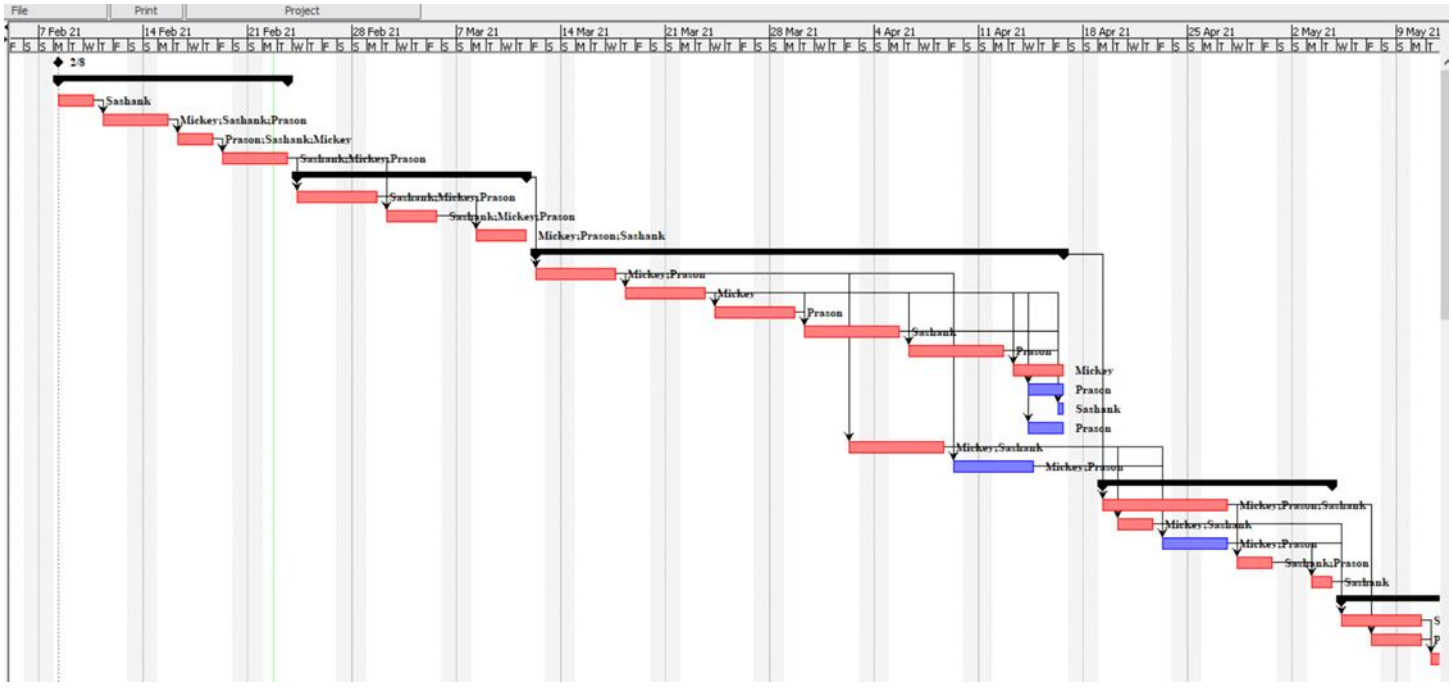
8. Working Methodology

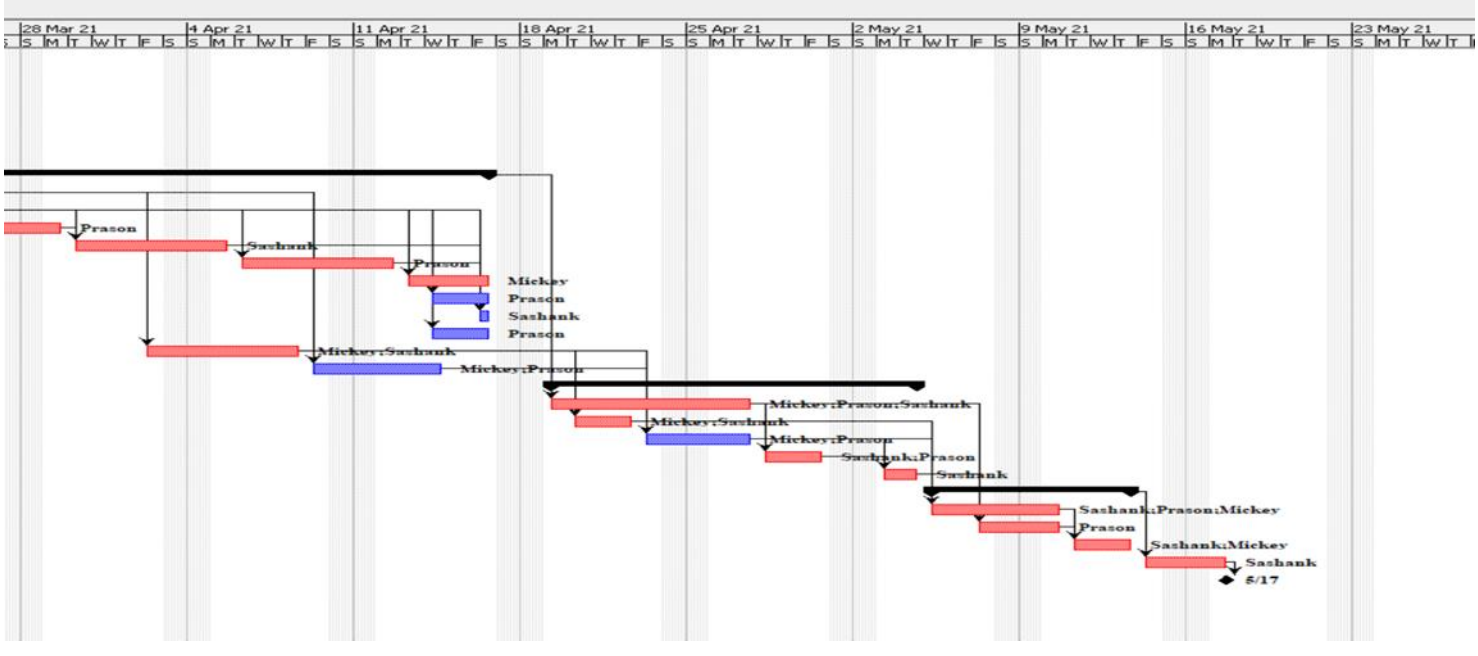
- **Activity Table**

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		Start	0 days	2/8/21 8:00 AM	2/8/21 8:00 AM		
2		**Project Analysis**	12 days	**2/8/21 8:00 AM**	**2/23/21 5:00 PM**		
3		Set up the project team	3 days	2/8/21 8:00 AM	2/10/21 5:00 PM		Sashank
4		Communicate the project plan	3 days	2/11/21 8:00 AM	2/15/21 5:00 PM	3	Mickey;Sashank;Prason
5		Develop project schedule	3 days	2/16/21 8:00 AM	2/18/21 5:00 PM	4	Prason;Sashank;Mickey
6		Project identification and problems	3 days	2/19/21 8:00 AM	2/23/21 5:00 PM	5	Sashank;Mickey;Prason
7		**Requirement Analysis**	12 days	**2/24/21 8:00 AM**	**3/11/21 5:00 PM**		
8		Analyze problem and business processes	4 days	2/24/21 8:00 AM	3/1/21 5:00 PM	6	Sashank;Mickey;Prason
9		Identify and analyze candidate solution	4 days	3/2/21 8:00 AM	3/5/21 5:00 PM	6;8	Sashank;Mickey;Prason
10		Write down the software requirements	4 days	3/8/21 8:00 AM	3/11/21 5:00 PM	8;9	Mickey;Prason;Sashank
11		**Desgin**	26 days	**3/12/21 8:00 AM**	**4/16/21 5:00 PM**		
12		Outline functional requirements	4 days	3/12/21 8:00 AM	3/17/21 5:00 PM	7	Mickey;Prason
13		login and homepage outline	4 days	3/18/21 8:00 AM	3/23/21 5:00 PM	12	Mickey
14		Addition of stock	4 days	3/24/21 8:00 AM	3/29/21 5:00 PM	13	Prason
15		check expiry and stock	5 days	3/30/21 8:00 AM	4/5/21 5:00 PM	13;14	Sashank
16		Bill creation of items	5 days	4/6/21 8:00 AM	4/12/21 5:00 PM	13;15	Prason
17		Search based on customers and items	4 days	4/13/21 8:00 AM	4/16/21 5:00 PM	13;15;16	Mickey
18		Employee Insertion and retrive function	3 days	4/14/21 8:00 AM	4/16/21 5:00 PM	13	Prason
19		Investment alerter based on trending and unsold items	1 day	4/16/21 8:00 AM	4/16/21 5:00 PM	13;15;16	Sashank
20		Total profit evaluation per day,month and year	3 days	4/14/21 8:00 AM	4/16/21 5:00 PM	13;16	Prason
21		Backend structure design	5 days	4/2/21 8:00 AM	4/8/21 5:00 PM	12	Mickey;Sashank
22		Database structural outline	4 days	4/9/21 8:00 AM	4/14/21 5:00 PM	12;21	Mickey;Prason
23		**Implementation**	12 days	**4/19/21 8:00 AM**	**5/4/21 5:00 PM**		
24		frontend functional components generation	7 days	4/19/21 8:00 AM	4/27/21 5:00 PM	11	Mickey;Prason;Sashank
25		Backend execution	3 days	4/20/21 8:00 AM	4/22/21 5:00 PM	21	Mickey;Sashank
26		Database connectivity to Backend	3 days	4/23/21 8:00 AM	4/27/21 5:00 PM	21;22;25	Mickey;Prason
27		Frontend integration to Backend	3 days	4/28/21 8:00 AM	4/30/21 5:00 PM	24;25	Sashank;Prason
28		Stocked application Impementation	2 days	5/3/21 8:00 AM	5/4/21 5:00 PM	26;27	Sashank
29		**Testing**	7 days	**5/5/21 8:00 AM**	**5/13/21 5:00 PM**		
30		Stocked application integration testing of each modules	4 days	5/5/21 8:00 AM	5/10/21 5:00 PM	28;26;25	Sashank;Prason;Mickey
31		Database accessibility and system testing	2 days	5/7/21 8:00 AM	5/10/21 5:00 PM	24;28	Prason
32		Debugging, error detection and correction	3 days	5/11/21 8:00 AM	5/13/21 5:00 PM	31;30	Sashank;Mickey
33		Software Deployment	2 days	5/14/21 8:00 AM	5/17/21 5:00 PM	29	Sashank
34		Finish	0 days	5/17/21 5:00 PM	5/17/21 5:00 PM	33	

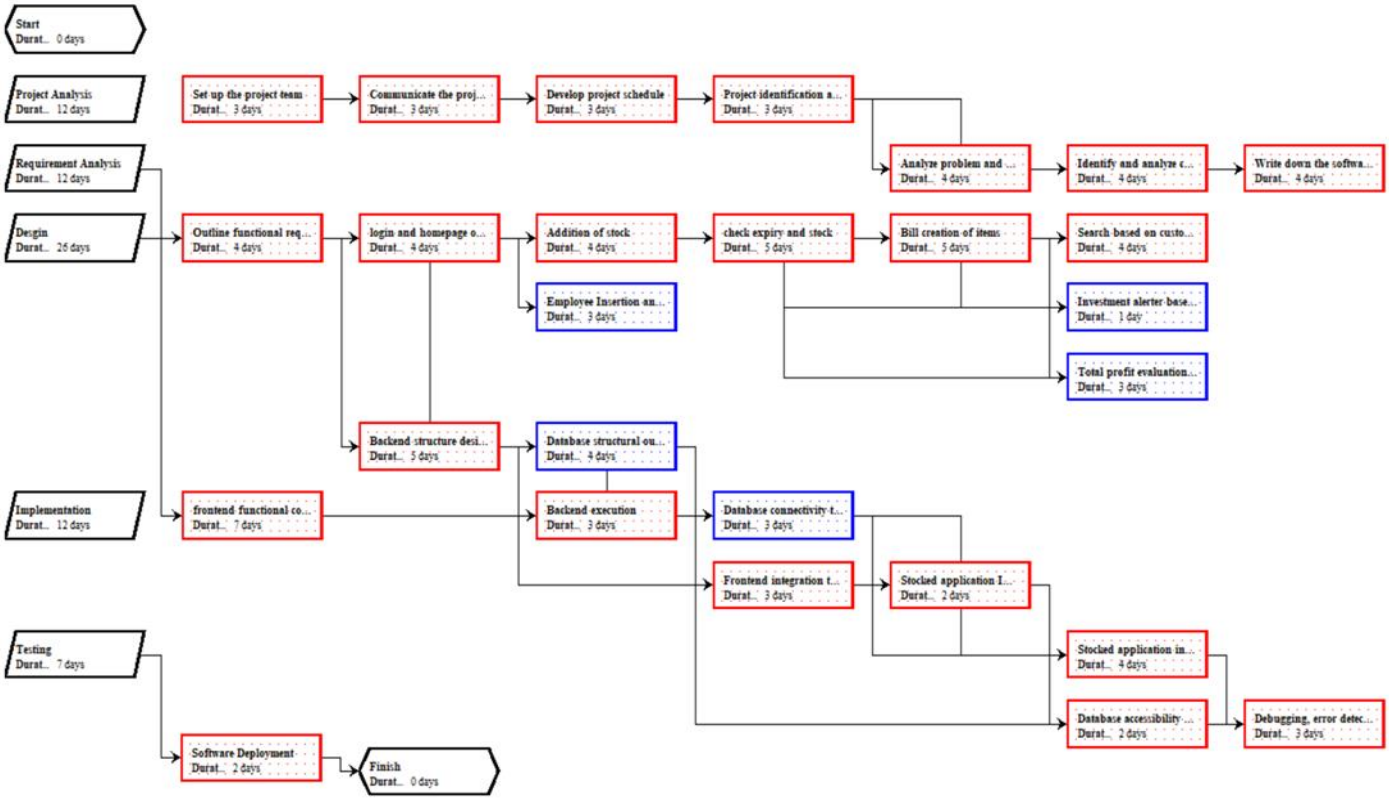
Activity	EST	EFT	LST	LFT	Slack	Critical Activity
Set up the project team	0	3	0	3	0	YES
Communicate the project plan	3	6	3	6	0	YES
Develop project schedule	6	9	6	9	0	YES
Project identification and problems	9	12	9	12	0	YES
Analyze problem and business processes	12	16	12	16	0	YES
Identify and analyze candidate solution	16	20	16	20	0	YES
Write down the software requirements	20	24	20	24	0	YES
Outline functional requirements	24	28	24	28	0	YES
login and homepage outline	28	32	28	32	0	YES
Addition of stock	32	36	32	36	0	YES
check expiry and stock	36	41	36	41	0	YES
Bill creation of items	41	46	41	46	0	YES
Search based on customers and items	46	50	46	50	0	YES
Employee Insertion and retrive function	32	35	47	50	15	
Investment alerter based on trending and unsold items	46	47	49	50	3	
Total profit evaluation per day, month and year	46	49	47	50	1	
Backend structure design	28	33	28	33	0	YES
Database structural outline	33	37	33	37	0	YES
frontend functional components generation	50	57	50	57	0	YES
Backend execution	33	36	34	37	1	
Database connectivity to Backend	37	40	57	60	20	
Frontend integration to Backend	57	60	57	60	0	YES
Stocked application Implementation	60	62	60	62	0	YES
Stocked application integration testing of each modules	62	66	62	66	0	YES
Database accessibility and system testing	62	64	64	66	0	YES
Debugging, error detection and correction	66	69	66	69	0	YES
Software Deployment	69	71	69	71	0	YES
Finish	71	71	71	71	0	YES

- **Gantt chart**

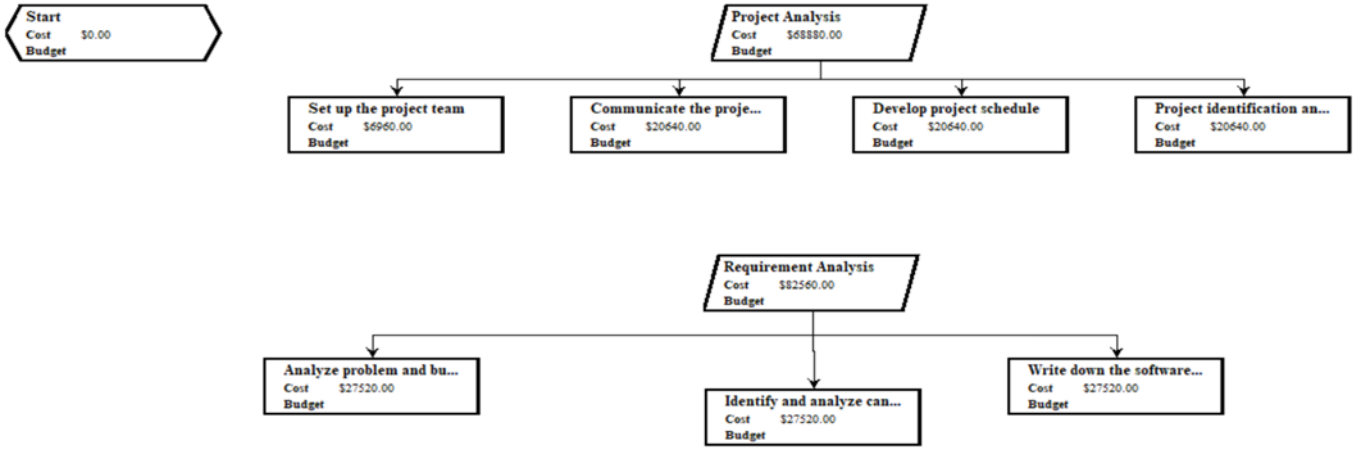

File
Print
Project
7 Feb 21
14 Feb 21
21 Feb 21
28 Feb 21
7 Mar 21
14 Mar 21
21 Mar 21
28 Mar 21
4 Apr 21
11 Apr 21
18 Apr 21
25 Apr 21
2 May 21
9 May 21
2/8
Sashank
Mickey;Sashank;Prason
Prason;Sashank;Mickey
Sashank;Mickey;Prason
Sashank;Mickey;Prason
Sashank;Mickey;Prason
Mickey;Prason;Sashank
Mickey;Prason
Mickey
Prason
Sashank
Prason
Mickey
Prason
Sashank
Prason
Mickey;Sashank
Mickey;Prason
Mickey;Prason;Sashank
Mickey;Sashank
Mickey;Prason
Sashank;Prason
Sashank

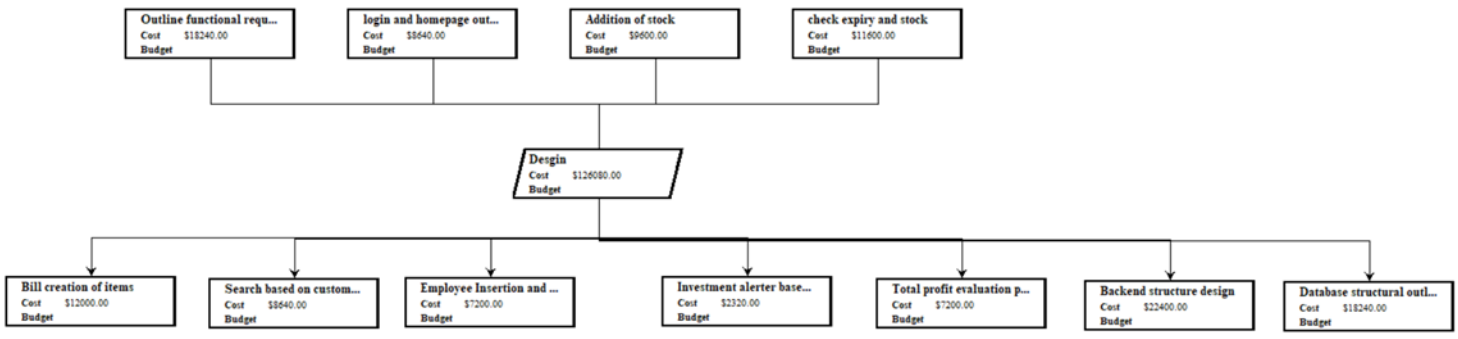

28 Mar 21
4 Apr 21
11 Apr 21
18 Apr 21
25 Apr 21
2 May 21
9 May 21
16 May 21
23 May 21
Prason
Sashank
Prason
Mickey
Prason
Sashank
Prason
Mickey;Sashank
Mickey;Prason
Mickey;Prason;Sashank
Mickey;Sashank
Mickey;Prason
Sashank;Prason
Sashank
Sashank;Prason;Mickey
Prason
Sashank;Mickey
Sashank
5/17

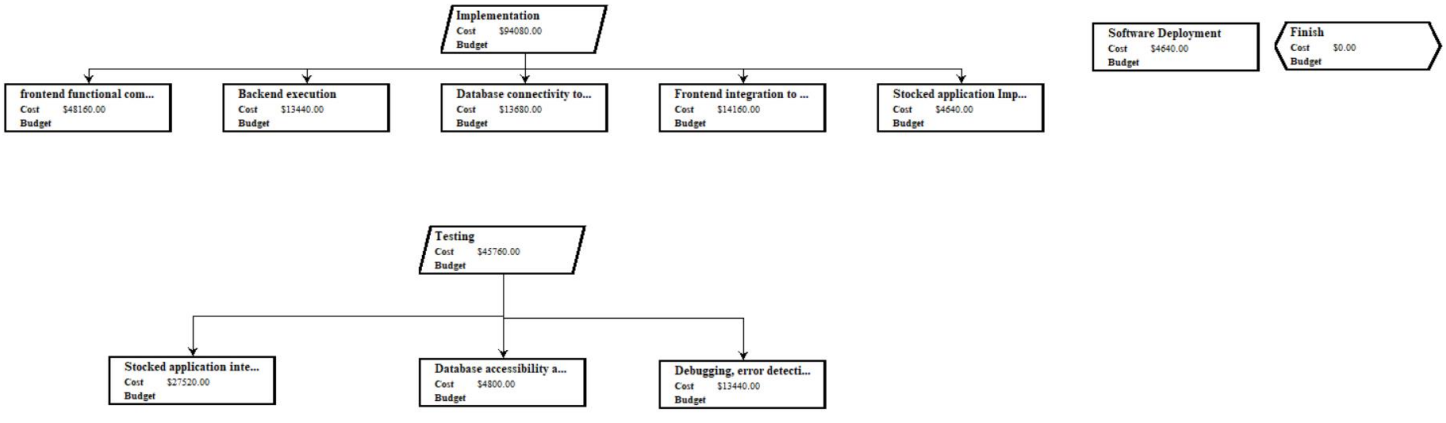
- **PERT chart**



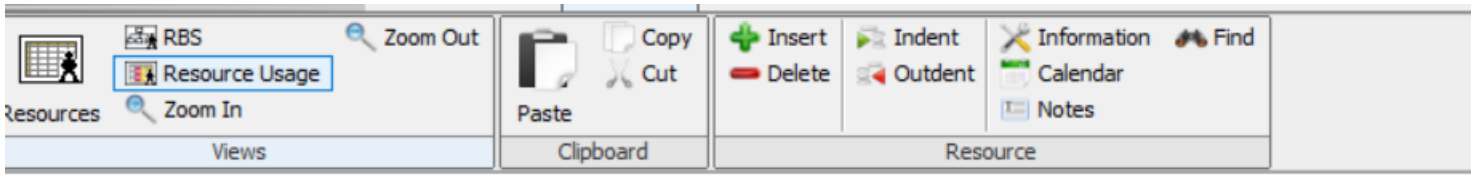
- **Work breakdown structure**

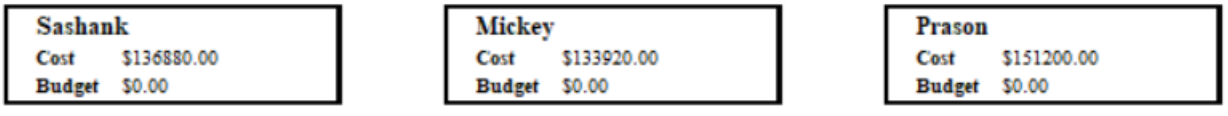



Outline functional requ...
Cost $18240.00
Budget
login and homepage out...
Cost $8640.00
Budget
Addition of stock
Cost $9600.00
Budget
check expiry and stock
Cost $11600.00
Budget
Desgin
Cost $126080.00
Budget
Bill creation of items
Cost $12000.00
Budget
Search based on custom...
Cost $8640.00
Budget
Employee Insertion and ...
Cost $7200.00
Budget
Investment alerter base...
Cost $2320.00
Budget
Total profit evaluation p...
Cost $7200.00
Budget
Backend structure design
Cost $22400.00
Budget
Database structural outl...
Cost $18240.00
Budget

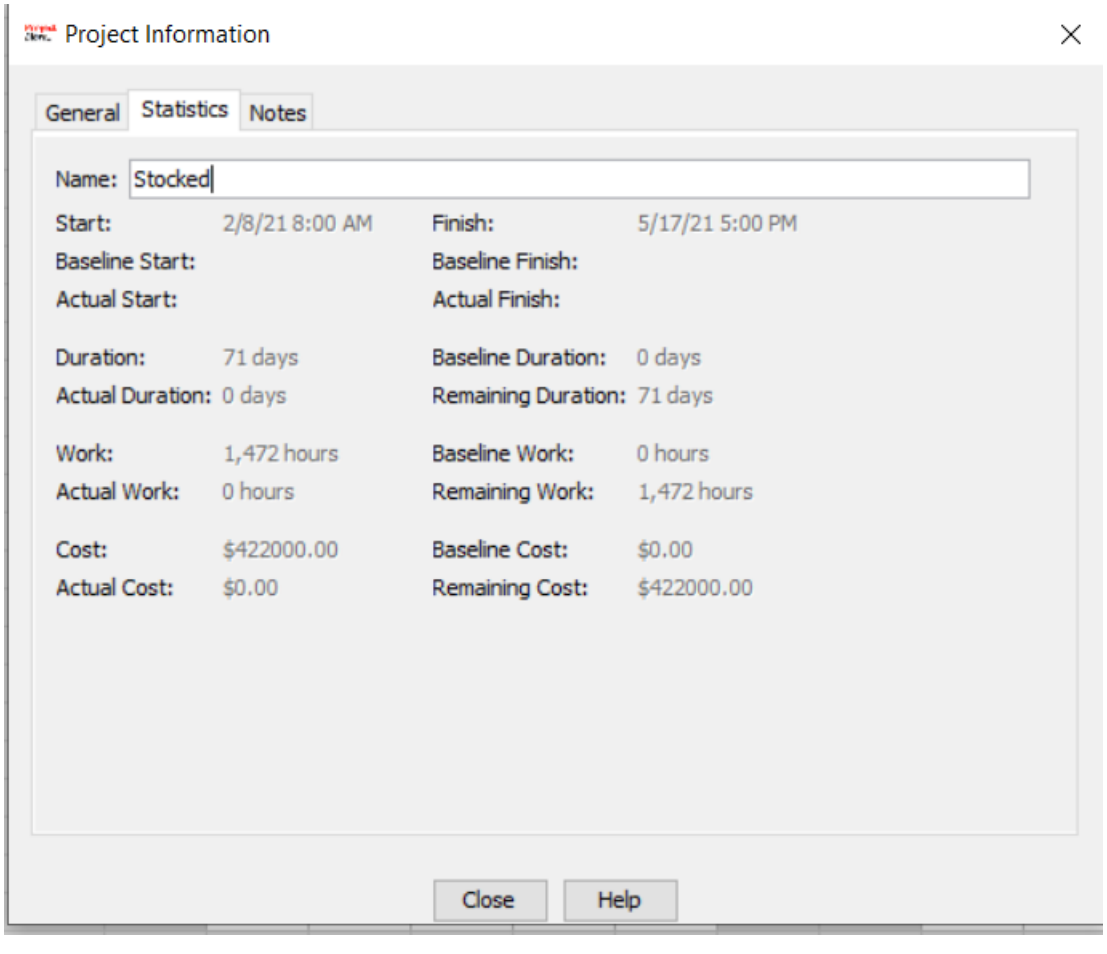

Implementation
Cost $94080.00
Budget
Software Deployment
Cost $4640.00
Budget
Finish
Cost $0.00
Budget
frontend functional com...
Cost $48160.00
Budget
Backend execution
Cost $13440.00
Budget
Database connectivity to...
Cost $13680.00
Budget
Frontend integration to ...
Cost $14160.00
Budget
Stocked application Imp...
Cost $4640.00
Budget
Testing
Cost $45760.00
Budget
Stocked application inte...
Cost $27520.00
Budget
Database accessibility a...
Cost $4800.00
Budget
Debugging, error detecti...
Cost $13440.00
Budget

- **Members and cost**



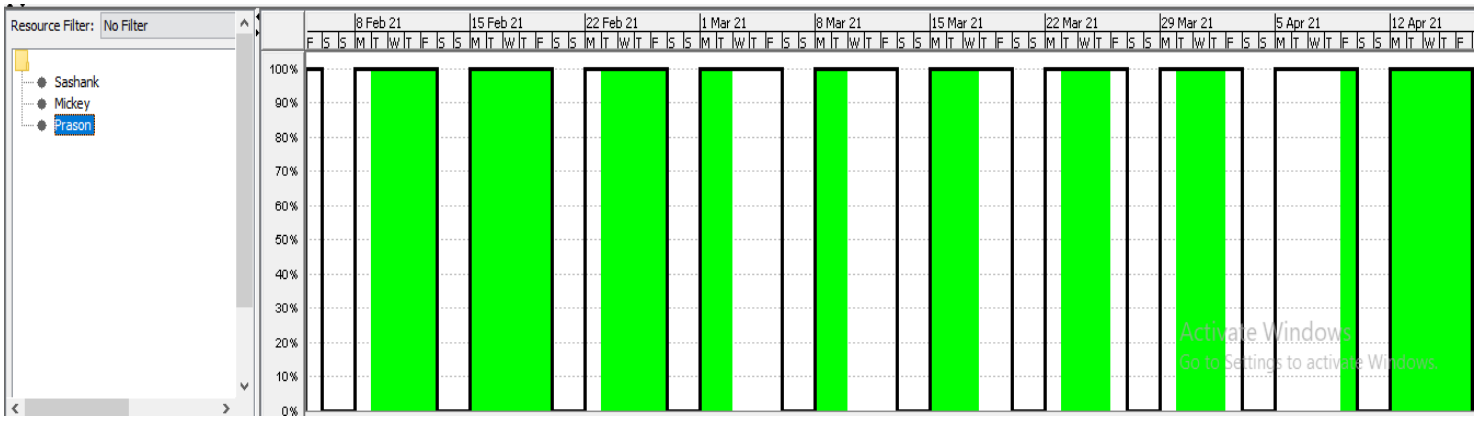


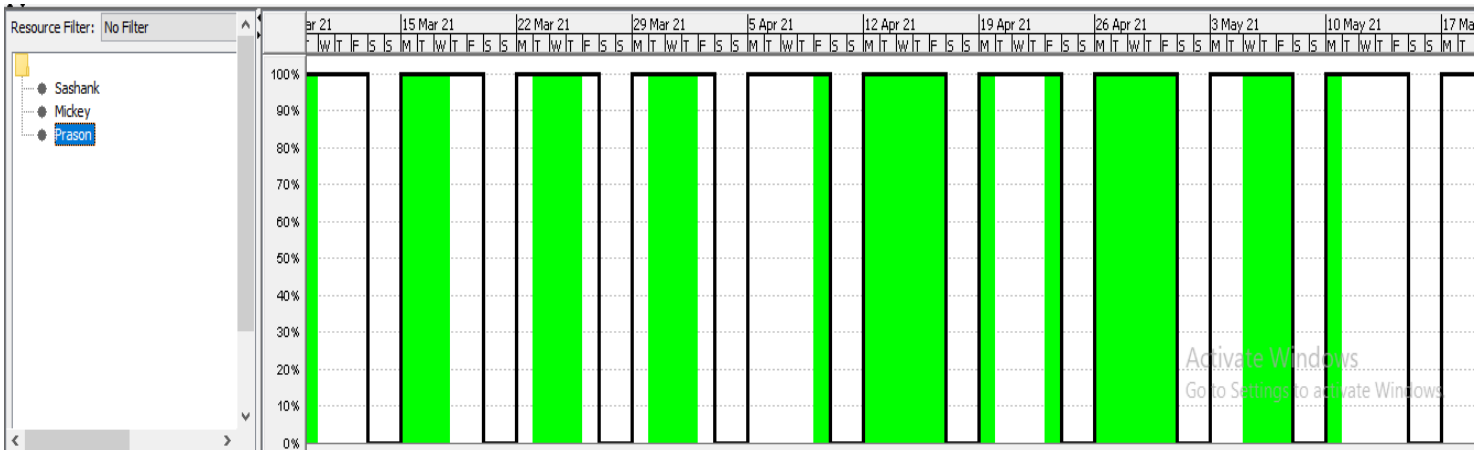
- **Final statistical cost**



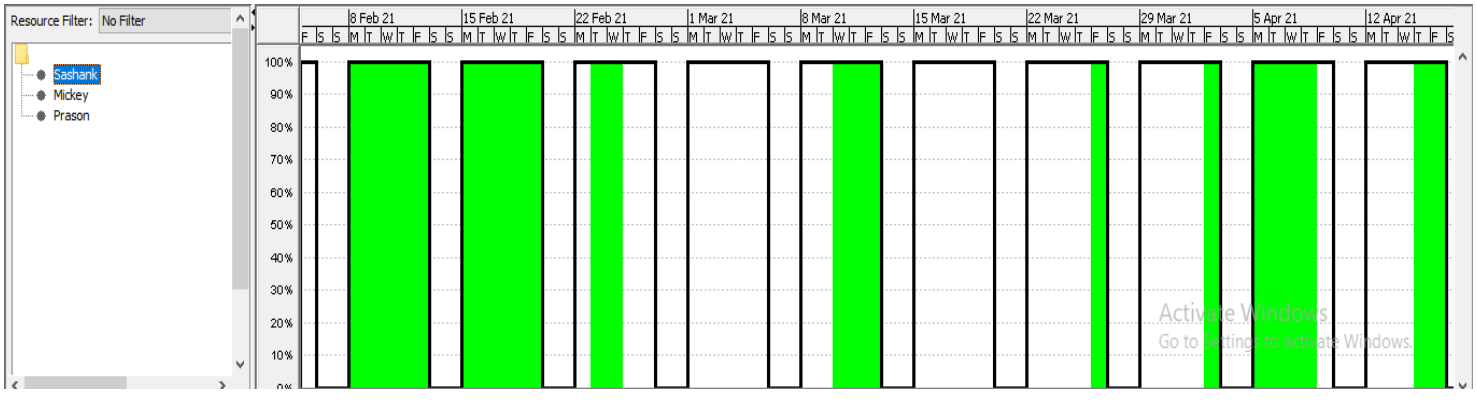
- **Histogram**

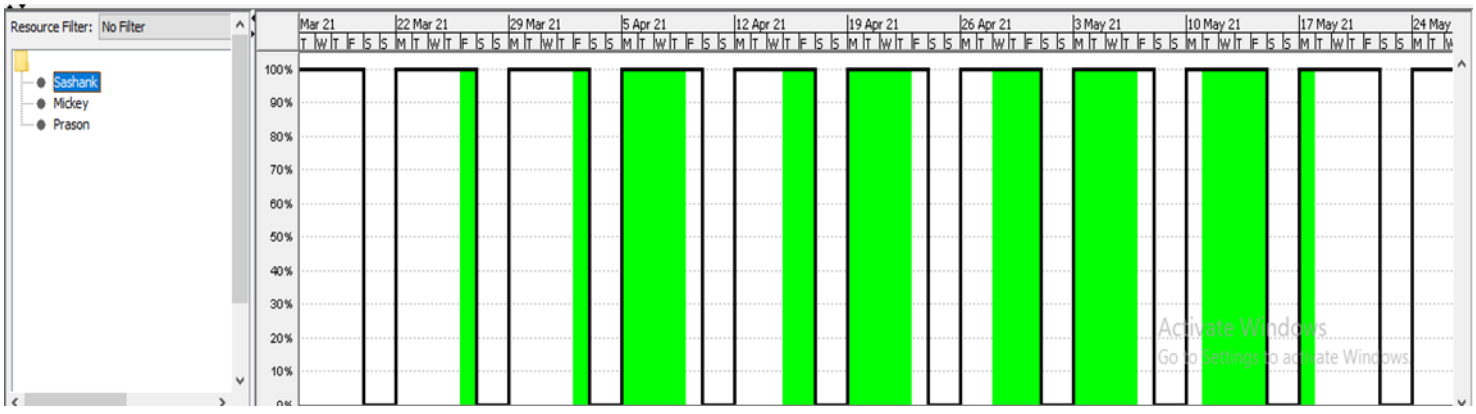
Prason



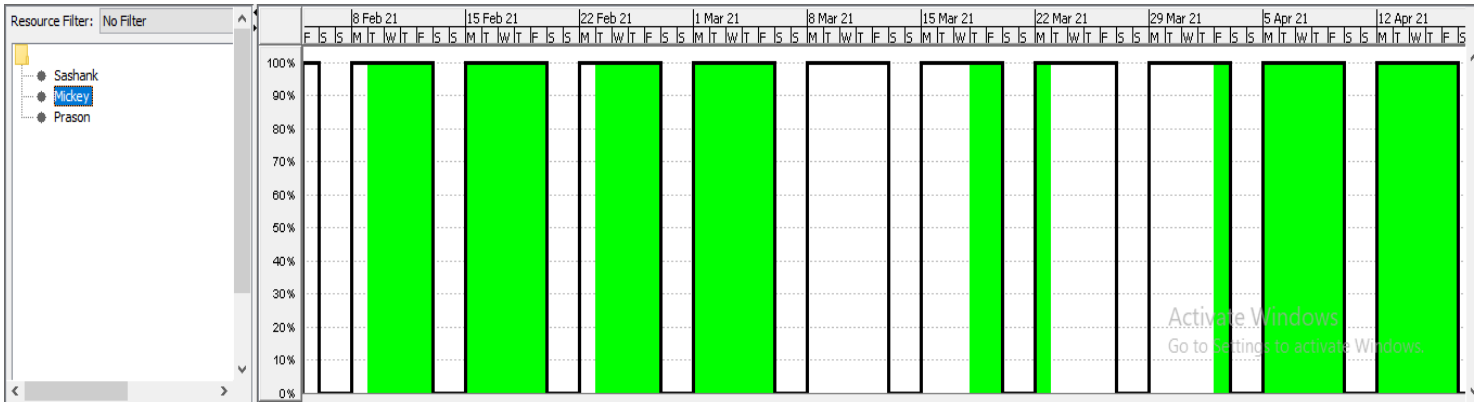


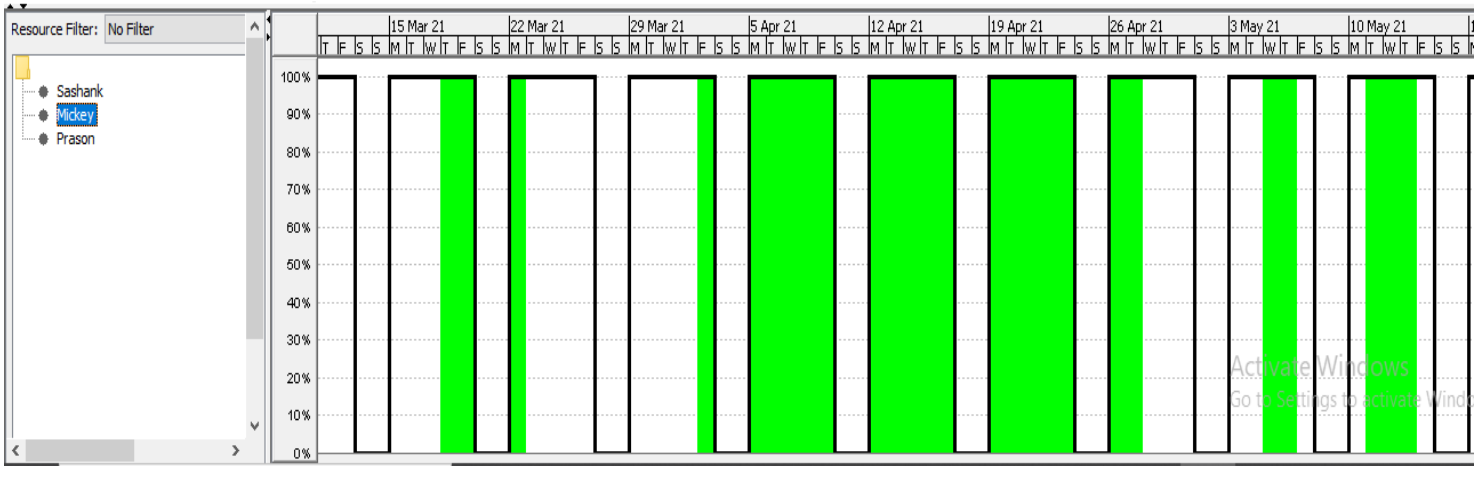
Sashank




Resource Filter: No Filter
Sashank
Mickey
Prason
Mar 21
22 Mar 21
29 Mar 21
5 Apr 21
12 Apr 21
19 Apr 21
26 Apr 21
3 May 21
10 May 21
17 May 21
24 May
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
Activate Windows
Go to Settings to activate Windows.

Mickey


Resource Filter: No Filter
Sashank
Mickey
Prason
8 Feb 21
15 Feb 21
22 Feb 21
1 Mar 21
8 Mar 21
15 Mar 21
22 Mar 21
29 Mar 21
5 Apr 21
12 Apr 21
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Activate Windows
Go to Settings to activate Windows.


Resource Filter: No Filter
Sashank
Mickey
Prason
15 Mar 21
22 Mar 21
29 Mar 21
5 Apr 21
12 Apr 21
19 Apr 21
26 Apr 21
3 May 21
10 May 21
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Activate Windows

Software Requirements Specification

for

STOCKED- A COMPLETE ORGANIZING SOLUTION

Prepared by

PRASON POUDEL 19BCE2550

MICKEY KUMAR ROUNIYAR 19BCE2520

SASHANK RIJAL 19BCE2484

VELLORE INSTITUE OF TECHNOLOGY, VELLORE

DATE: 15-03-2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

It is very essential to store, organize and make quickly available the necessary data re-quired for the well organized and effective functioning of any business. From a small proprietary business like a simple shop to multi-billion-dollar companies like Google, Facebook, etc., it is really necessary to organize, tabulate, store and secure the data in an efficient manner. If we look into our community it is a general perspective that the software for keeping track should only be used by high end companies and so on. We can also see many software's are used in big shops such as supermarkets and depart-mental stores. But we often see that the organization is often kept manually or sometimes not kept in the smaller shops. So, looking at the scenario we have decided to create a software that can run on a very low specifications and would be suitable for a smaller shop. It will help the shops to get insights on their organizing techniques and help for their betterment. Our "Stocked" software can help address employee count scenario and a range of other complex situations related to cost, demand, organiza-tion, tabulation, storing and security by presenting the same data to everyone in the business at the same time. We want to design a software that will help to keep track of stocks and will also provide insights about the business based on the stocks, demands, sales, recent trends, profit or loss and so on.

1.1 Purpose

This document lists the requirement specifications STOCKED-A COMPLETE ORGANIZING

SOLUTION. The document is subject the change as the project progresses. The given version of the document is the initial one. Further changes of the project will be recorded to the document.

1.2 Document Conventions

The document follows IEEE Format for the entirety of this document. However, a few points that should be kept while going through this paper are 1. Due to copyright and trademark reasons, the software name will not be disclosed. 2. This document is the property of the shop and any attempt to create a copy of this will lead to legal action. 3. The entire program of the application is proprie-tary and any attempt to create its copy will result in legal action. 4. The users of this software may directly view the user documentation under section 2.6 of this paper

1.3 Intended Audience and Reading Suggestions

The intended audience for the project is small businesses and shops. It has a simple graphical user interface and is really easy to interact with. A person with a minimal technical knowledge about computers and programming can easily operate the software. The project is an application that shall allow its users to:

1. Sign up and consequently sign in into the stocked software.

2. Search for items available in the shop and all new stock.
3. Generate profit and loss for the business.
4. Check the expiry date of any product.
5. Create an invoice/bill after the purchase.
6. Check the trending items in the shop
7. Check for employee's details or add a new employee.
8. Search for any item or customer
9. Check the item report based on parameters like quantity,etc

1.4 Product Scope

The main objective of the process is to fully digitalize the organizing process of the shop. Along with making the process fully digital, we intend to include some other features to the application so that it helps the shopkeeper to maximize the profits. In general, our objective is to include the following features on our application:

• keep a record of stock in the shop
• calculate profit or loss
• notify the shop owner when any item has crossed its expiry date.
• aware the shop owner if an item is being rapidly sold so that the owner can order more.
• aware the owner if any item is going to be finished in the shop
• keep record of the employee in the shop
• provide yearly review such as which item was sold more in which month

1.5 References

▪ [1] R. Elmasri & S. B. Navathe, Fundamentals of Database Systems, Addison Wesley, 7 th Edition, 2015

[2] Thomas Connolly, Carolyn Begg, Database Systems : A ‖ Practical Approach to Design,

Implementation and Management‖,6th Edition,2012

[3]. Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming by

Mark Summerfield

[4] MySQL for Python by Albert Lukaszewski

[5]. Moskowitz,Robert,"Using Your Computer for Inventory Control", Accvision Retrieved August 17, 2010.

[6] "Inventory Management Software". EGA Futura. Retrieved 23 November 2012. [7] Lockard, Robert (29 November 2010) "3 Advantages of Using Inventory Management Software". Inventory System Software Blog. Retrieved 23 November 2012.

[8] "Inventory Management Organism" by A Rohini, Wubshetasamnew Woldeyohannshiluf and Teshometeklemicheal Walelignayimelo

[9] “Bill Generator and Inventory Maintenance “ by Utkarsha Mendhe, Ankita Lohave, Aayush Sah, Varun Paliwal,Prof. Nutan Sonwane

2. Overall Description

2.1 Product Perspective

STOCKED is the digitized version of the traditional manual stock entry and management using pen and paper. Existing manual system requires tedious work and takes a lot of time to record and organize. Apart from that, the physical record is a subject to a lot of depreciation and danger and can get easily damaged by a lot of physical factors. Moreover, they are not easily accessible. The new digitized system aims to overcome the above-mentioned drawbacks of the existing system. It will allow its user to digitally perform and store all the information and data and make the data easily accessible. Furthermore, it will reduce the workload of the employees of the shop and it makes the business more efficient. The system also allows the customers to check the availability of a specific item that they want, its price, its availability and the expiry date and any discounts (if available). Administrator can modify, remove existing stocks, also add new items to the store. Furthermore, administrator can see customer requests, trading items, and decide whether to accept and reject them. The product is solely meant to provide a convenient platform to the customer as well as the store owner to perform shopping in small shops. However, in the process personal information of the user including name, contact, nationality, date of birth, address, profession and payment details may be asked.

2.2 Product Functions

Trending/Not to invest item alerter
In this project, we describe trending item as the item that is being rapidly sold and we define not to invest item as the item that is in the stock for a very long time and is barely sold. This application will be smart enough to classify the products into trending or not. According to the classification, the application will suggest the shopkeeper if the shopkeeper should increase the stock of items or not.

Invoice Creator
It will be better if the stock management system has an invoice generator. If this happens, the shopkeeper doesn’t have to manually enter the record of sales daily. So, Invoice creator will create the bills and with the help of stock tracker module, it will modify the database. Similarly, All the generated bills will be stored for future reference.

Expiry date checker
It is a tedious work to check the expiry date of the products in a shop. It requires extra manpower and money. As our application is already keeping record of all the products, it will also keep track of the expiry date and will check with the system calendar and automatically notify the users.

Profit/loss generator
Every owner should be made aware about his /her business. He /She should know whether he is in profit or in loss so that they can take necessary steps to make their business more fruitful. So, one of the modules in our application will calculate daily profit and loss which can be checked by owner. Similarly, it will calculate the yearly profit or loss.

Stock Analysis
Every owner should be made aware about the condition and state of the stocks. So the tracking analysis and recording of various stocks is necessary. In order to achieve such flexibility this module will help avoid Stock-outs and Over-stock.

Besides these functionalities' other various aspects of the project is also implemented in the software like purchase order records, search based on any features of the product like mfg, expiry date, price name and so on, employee track record, administrative privilege and so on

2.3 User Classes and Characteristics

2.4 Operating Environment

The project in its entirety is developed on python along with PyQt5 integration for the GUI features, which can be installed using easy install or pip. The user might also be required to install a database app (DB browser for SQLite) to perform the necessary database operations. Any device supporting the python version 3.4 or above should be able to run this app without any problem. However, if one faces any technical issue, the same should and can be reported at the email dev@shop.com.

2.5 Design and Implementation Constraints

The project has been developed by us to as much extent as possible. However, due to some limitations of the development platform, this app was not intended to be developed as a mobile application. Moreover, any kind of touch/pen cursors are not supported on the app.

2.6 User Documentation

2.7 Assumptions and Dependencies

The users of this app are assumed to be running a computer which support Python 3.4.1 or above. Moreover, it is assumed that users are using a mouse/touchpad/any other pointing device to access the app as the app does not support any touch sensitive screens due to design constraints.

3. External Interface Requirements

4. System Features

4.1 LOGIN

4.1.1 Description and Priority

This is a general functionality to ensure the security of the software that it is accessed through authentic person only. Once the software is opened, the user is asked to enter their login credentials. After validation, the dashboard with appropriate options is displayed.

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

REQ-1:

Input: user is asked to enter the username and password.

Output: Shows the verification and opens the dashboard.

Processing: validates the username and password and after validation opens the dashboard for the software.

4.2 ADD STOCK

4.2.1 Description and Priority

This feature allows you to add the items and materials for the specific shop. Once the user selects the “add stock” option, another window pops out with appropriate options for entering the details of the new stock.

4.2.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.2.3 Functional Requirements

REQ-1:

Input: user selects the add stock option.

Output: another window pops up with suitable options for entering the information.

REQ-2:

Input: user is asked to enter the details like name, cost price, manufacture date, expiry date, quantity and selling price.

Output: invoice is created and the details of the added item is added to the stock.

Processing: all the information regarding the stock is added to the database and all the appropriate constraints are checked and verified and after the verification, the information is displayed.

4.3 SEARCH

4.3.1 Description and Priority

This feature allows you to search for an item or for a customer. When the “search” option is selected, new window pops out and user is asked to select for the type of search to be processes. The types can be classified as item search and customer search. For item search, search can be done by using any of the parameters consisting search based on name, cost price etc. For customer search, it can also be searched by using any of the parameters including search based on name, last item purchased date, number or id assigned at the time of bill creation.

4.3.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.3.3 Functional Requirements

REQ-1:

Input: user selects the “search” option.

Output: search based on item and search based on customer is displayed.

REQ-2:

Input: user selects the “item” search option.

Output: all the options available for search based on item is displayed.

REQ-2.1:

Input: user enters any of the information related to the search. This maybe be either one of the parameters like name, cost price, selling price etc. of the item.

Output:

1> if the item is found all the information related to the searched item is displayed from the invoice generator.

2> if the item is not found

No such item error message is displayed.

Processing: the items based on any of the parameters provided by the user is taken and then searched in the system. If found and validated with appropriate constraint, the information is displayed. Else, error message is displayed.

REQ-3:

Input: user selects the “customer” search option.

Output: all the options available for search based on customer is displayed.

REQ-3.1:

Input: user enters any of the information related to the search. This maybe be either one of the parameters like customer's name, shopped date, identification number etc.

Output:

1> if the customer details are found all the details of the searched customer are displayed from the invoice generator.

2> if the customer details are not found

No such customer found error message is displayed.

Processing: the customer information based on any of the parameters provided by the user is taken and then searched in the system. If found and validated with appropriate constraint, the details of the customer like shopped date, number, name and id are displayed else the error message is shown like "no such customer found"

4.4 EMPLOYEE

4.4.1 Description and Priority

This feature helps you to maintain all your employee's information. When the "employee" option is selected, new window pops out and user is asked to select for the options available for this option. The options can be entering the information of the new employee and search for the employee who is already present. For entering the details of new employee, all the information related to it is asked like name, address, identification number, phone number, post assigned and salary. For search purposes, any of the parameters can be used such as name or id.

4.4.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.4.3 Functional Requirements

REQ-1:

Input: user selects the “employee” option.

Output: options for employees like enter for new employee and search based on employee is displayed.

REQ-2:

Input: user selects the “enter” option.

Output: all the options are available for entering the information for new employee is displayed.

REQ-2.1:

Input: user is asked to enter the details for the new employee like name, number, id, address, assigned post and salary decided.

Output: employee added successfully message is displayed and welcome message is popped.

Processing: after all the information is added, the information received is added and is checked and validated. Upon validation, invoice is generated and successful message is showed along with “welcome” message. Upon any error, error message is displayed.

REQ-3:

Input: user selects the “search” option under the employee option.

Output: all the options available for search based on employee is displayed.

REQ-3.1:

Input: user enters any of the information related to the search. This maybe be either one of the parameters like employee’s name, identification number etc.

Output:

1> if the employee details are found all the details of the searched employee are displayed from the invoice generator.

2> if the employee details are not found

No such employee found error message is displayed.

Processing: the employees information based on any of the parameters provided by the user is taken and then searched in the system. If found and validated with appropriate constraint, the details of the employees like post, salary, name, id, etc is displayed else the error message is shown like “no such employee found”

4.5 BILL CREATION

4.5.1 Description and Priority

This feature is for creating the bill for the customers. When the “bill creation” option is selected, a window for creating bills with appropriate option is displayed. The option here includes customer name, address, phone number, id, item bought and quantity.

4.5.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.5.3 Functional Requirements

REQ-1:

Input: “bill” option is selected.

Output: window for entering the details is displayed.

REQ-2:

Input: user enters various information for creating the bill like customer’s name, unique id assigned to him, address, phone number, quantity and item selected.

Output: bill is created and displayed with overall information of the customer included with the item information.

Processing: after the item and quantity is selected, the price is calculated based on the selling price of each item selected and GST taxes are added and then the total selling price is displayed and the bill is created. Also, the customer’s name and id associated with it is stored in the database for future search purposes.

4.6 CHECK EXPIRY

4.6.1 Description and Priority

This will ensure and notify the owner about his items which are going to be expired or will be expired soon within a month. Here, the after the user selects the “check expiry” option, all the information of the expired items and the item which are about to get expired are displayed.

4.6.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.6.3 Functional Requirements

REQ-1:

Input: user selects the check expiry tab.

Output: window pops out with various information of the already expired item with a separate column which displays the item which are about to get expired within the month so that the owner can put the item in for sale.

REQ-2:

Input: user selects the “remove” option.

Output: all the items which are expired are removed from the stocks.

Processing: here the system first checks for the items that are expired and then removes it from the stock.

NOTE: only the item that is expired will be removed after selecting the remove option. The items that are expiring within the month showed in another column will not be removed.

4.7 STOCK ANALYSER

4.7.1 Description and Priority

This will aware the owner about the condition and state of the stocks. When the “check stock” option is selected, the total items that are currently stored in the software along with the quantities is displayed.

.

4.7.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.7.3 Functional Requirements

REQ-1:

Input: user selects the “check” option.

Output: all the items currently present in the system is displayed along with their quantities in the new window

4.8 TRENDING/NOT TO INVEST ALERTER

4.8.1 Description and Priority

This feature will aware the owner about the item that is being rapidly sold and item that is in the stock for a very long time and is barely sold. This feature will suggest the shopkeeper if the shopkeeper should increase the stock of items or not. When the “trending” option is selected, all the most trending items is displayed so that the user and the customer know about it. Upon selecting this option, another column for the least trending item is also displayed to aware to the owner about which items to buy more and less in upcoming stages.

4.8.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.8.3 Functional Requirements

REQ-1:

Input: user selects the “trending” option.

Output: all the items which are trending will be shown. This will include the trending item based on most sold and least sold items.

Processing: the items that are sold with be looked in the system and further it will be classified based on the number of sales and items quantity based on specific dates. This will include date ranging from past date to individual month and year.

4.9 REVENUE/PROFIT LOSS GENERATOR

4.9.1 Description and Priority

This feature will aware the owner about whether he is in profit or in loss so that they can take necessary steps to make their business more fruitful. When the “revenue” is selected, it will display all the financial related information for all dates specifically categorized in terms of current date, running month and year. This option is specific to overall profit and loss of the company so to aware the user about the business.

4.9.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.9.2 Functional Requirements

REQ-1:

Input: user selects the “revenue” option.

Output: all the financial related information like profit and loss will be displayed and be categorized in terms of date, month and year.

Processing: all the sold items are taken into consideration and profit or loss (based on the sale of the specific item) is calculated at the backend first. Upon completion, it is categorized into specific month and year and is displayed for analysing purpose.

5. Other Nonfunctional Requirements

5.1 Security Requirements

5.1.1 Description and priority

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure.

▪ System should include restore and recover functions in order to prevent data loss.

▪ System should assure data integrity

▪ Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully

▪ System should have an Authentication and Authorization System (AAS) for logins. ▪ System should grant administrative privileges only to the one who logins with predefined administrative username and password.

▪ System should allow administrator to change his/her system-generated password as he/she Wishes

5.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

5.1.3 Functional Requirements

REQ-1:

Input: user enters the login credentials.

Output:

Registered message is displayed.

Processing: the credentials are first verified and then the password is first encrypted and then it is saved in the software.

5.2 Performance Requirements

System should be able to handle 1000 transactions per second.

User should be displayed acceptance message within 5 seconds, when he/she submits entered data to the system.

Warning messages about entrance data out of defined standards must remain on the screen for 3 seconds

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

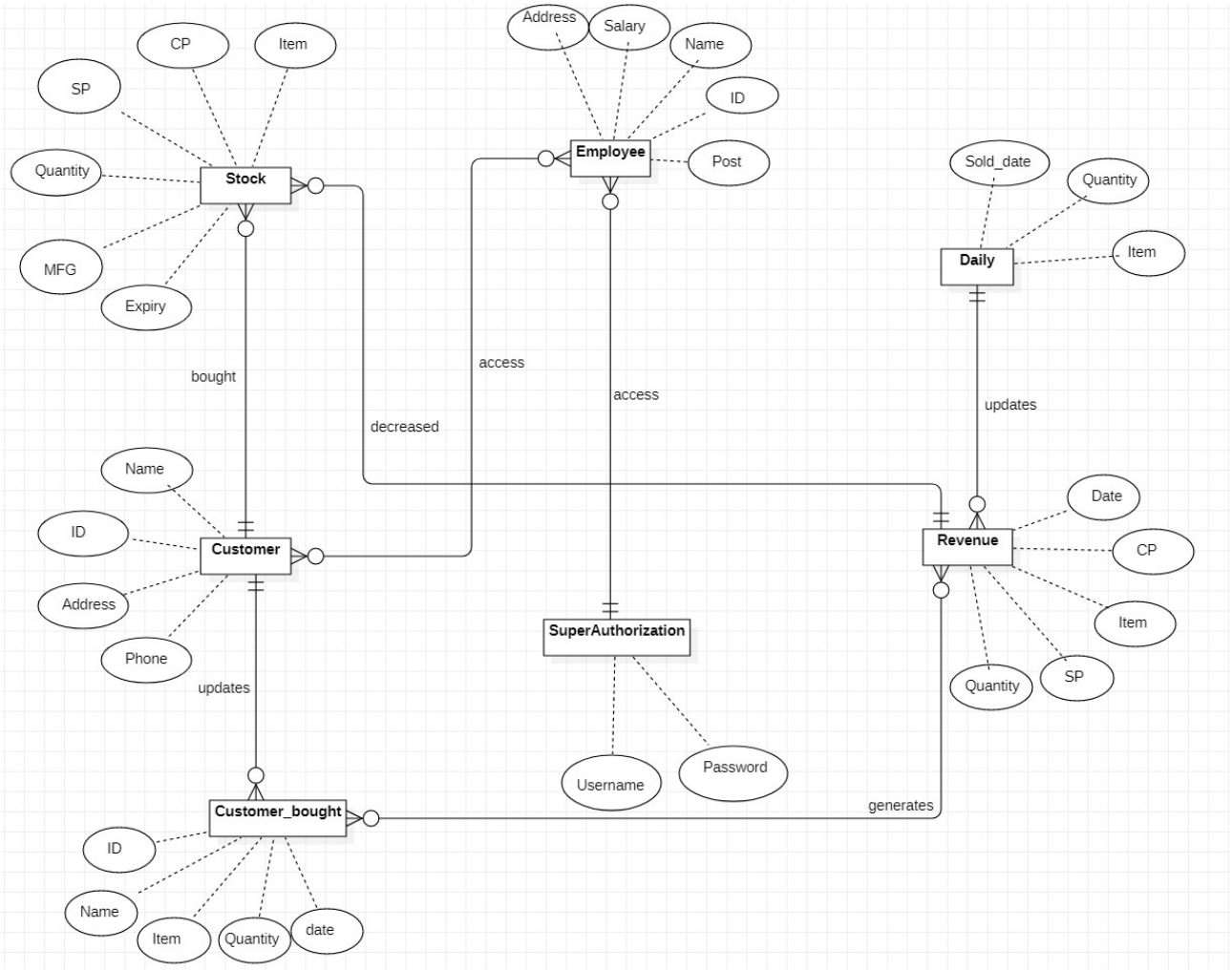
Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

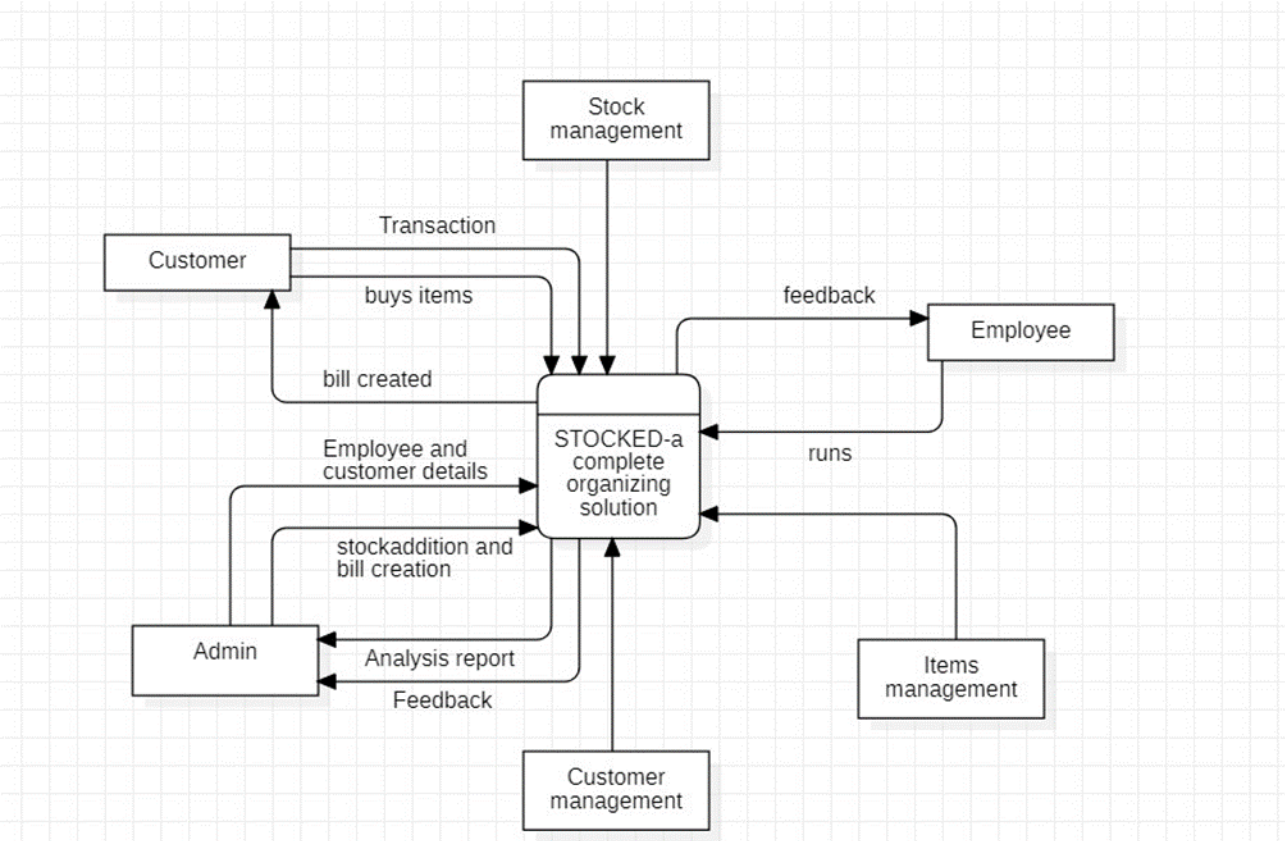
<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

- **ER diagram**


Address
Salary
Name
ID
Employee
Post
CP
Item
SP
Quantity
Stock
MFG
Expiry
bought
access
access
decreased
updates
Sold_date
Quantity
Item
Daily
Name
ID
Address
Phone
Customer
updates
SuperAuthorization
Username
Password
Date
CP
Item
Quantity
SP
Revenue
generates
Customer_bought
ID
Name
Item
Quantity
date

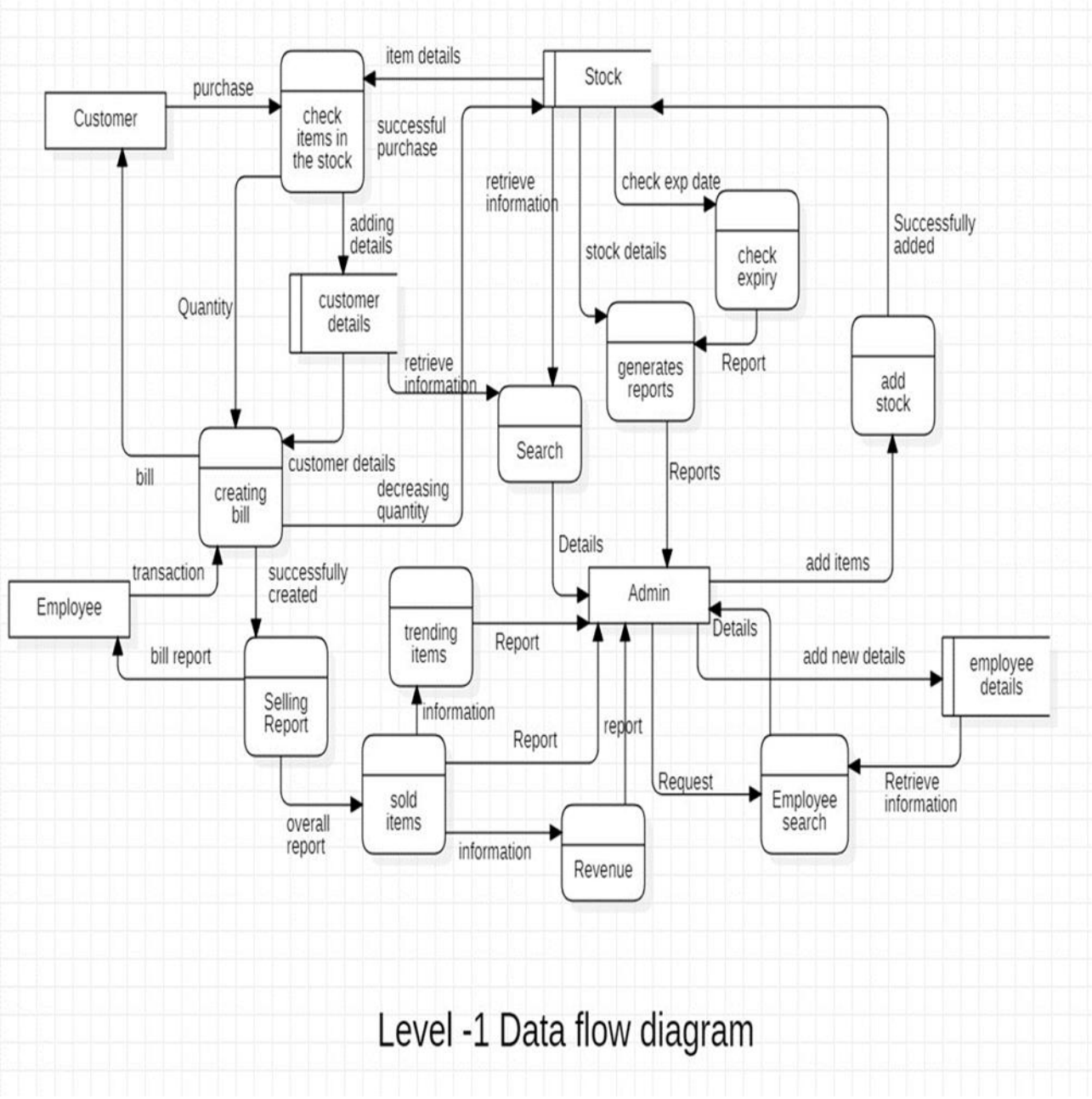
- **DATA FLOW DIAGRAM**

Level-0 DFD



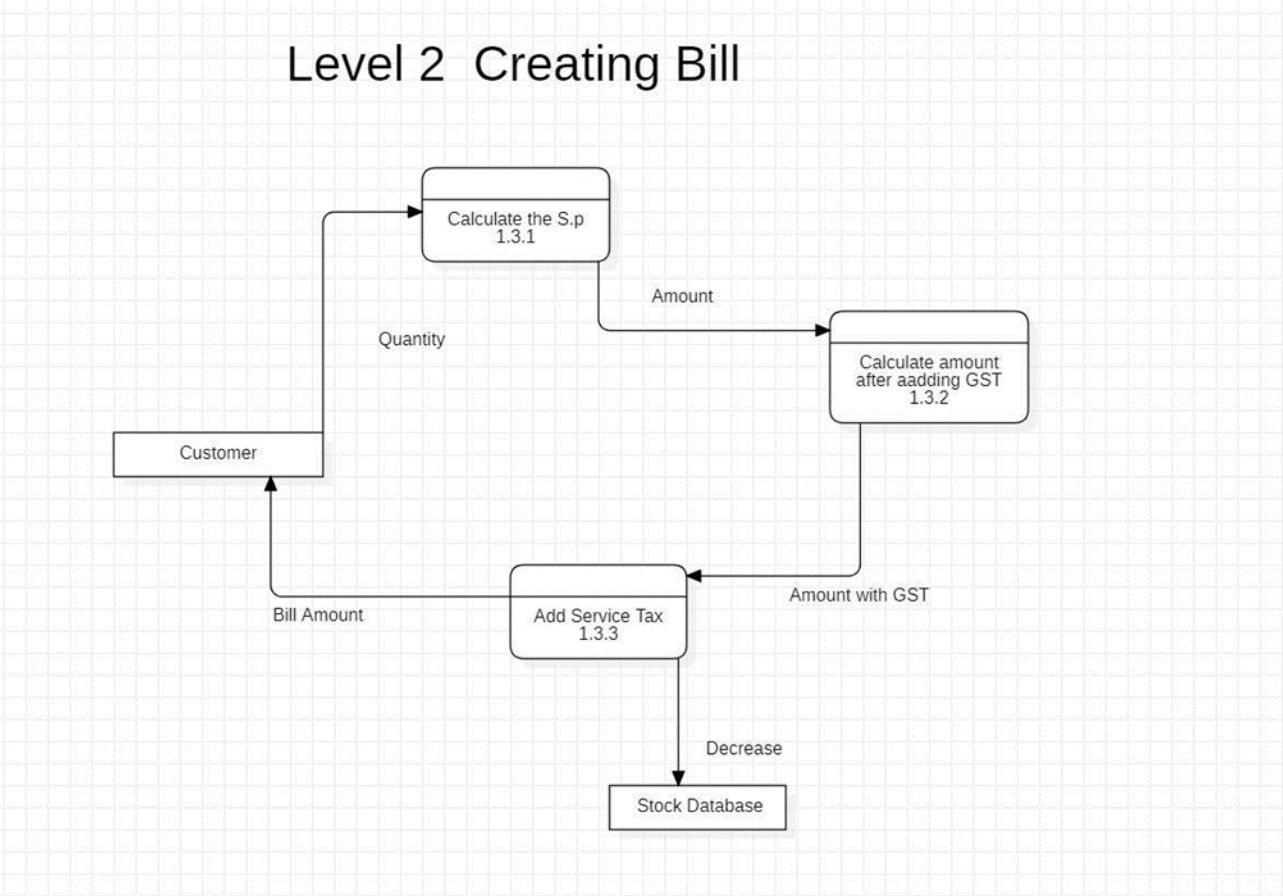
Level-0 Data flow diagram

Level-1 DFD

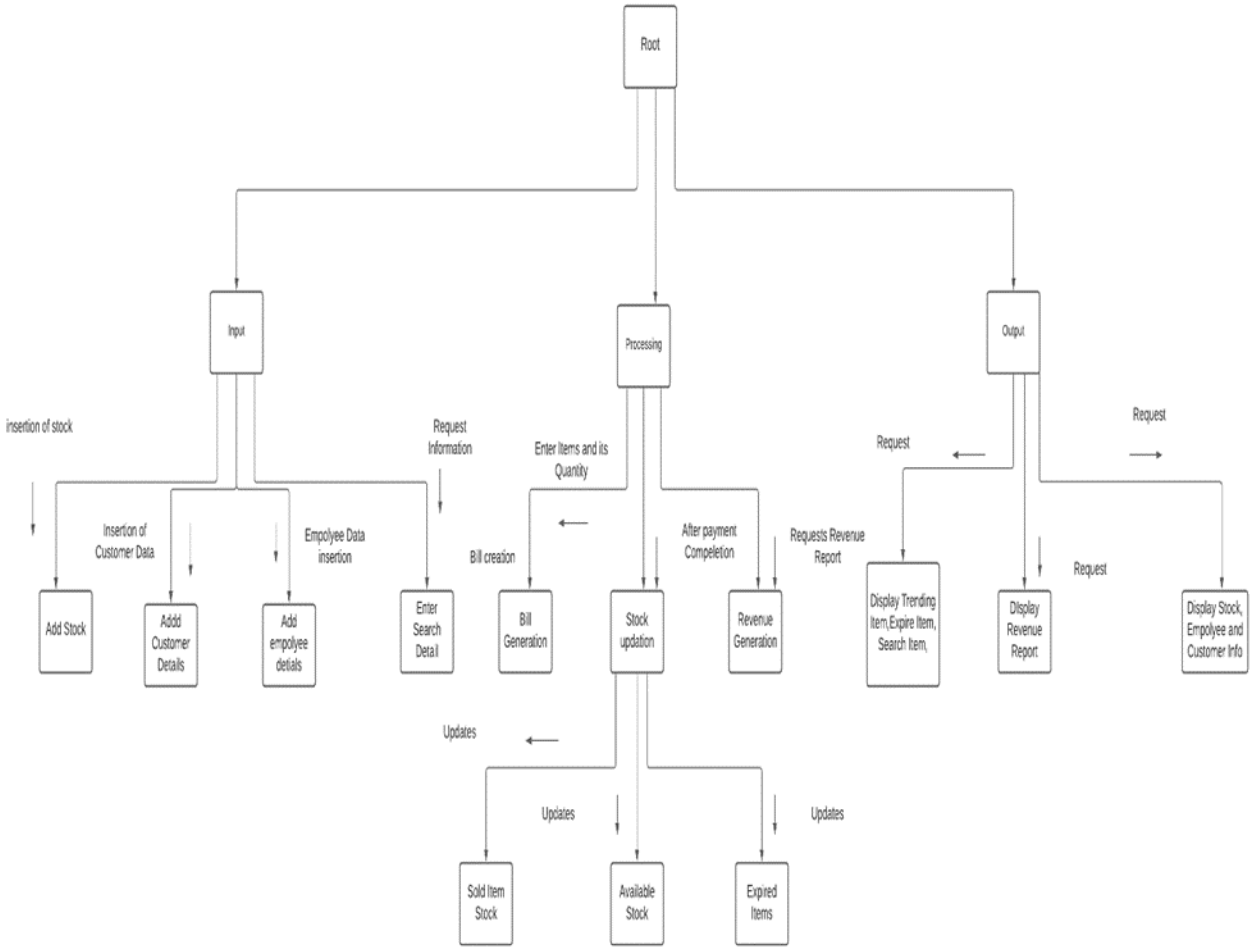


Level -1 Data flow diagram

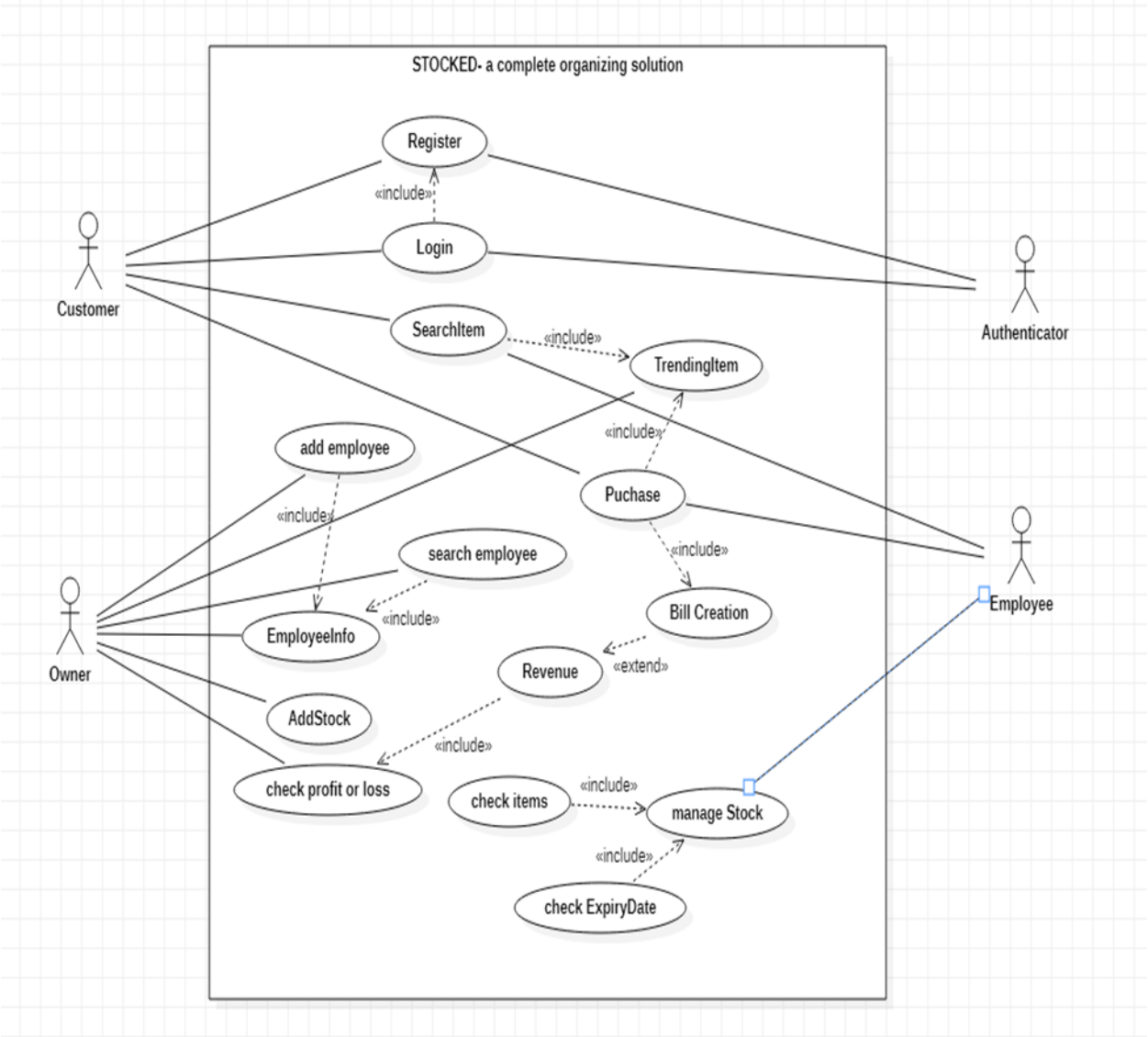
Level-2 DFD


Level 2 Creating Bill
Calculate the S.p
1.3.1
Amount
Quantity
Calculate amount
after aadding GST
1.3.2
Customer
Amount with GST
Bill Amount
Add Service Tax
1.3.3
Decrease
Stock Database

- **Structure chart**



- **Use case diagram**



- **Use Case Description for all the use cases**

Use Case ID:	1		
Use Case Name:	EmployeeInfo		
Created By:	Mickey Kumar Rouniyar	Last Updated By:	01-04-2021
Date Created:	01-04-2021	Date Last Updated:	

Actor:	owner
Description:	The owner can check the details of employees, their salaries and sales.
Preconditions:	Check whether the employee exists or not.
Postconditions:	• The list of employees will be displayed • The details about the specific employee searched will be displayed
Priority:	high
Frequency of Use:	none
Normal Course of Events:	• Owner logs in • Owner searches for employee • The system displays the employee details
Alternative Courses:	none
Exceptions:	2. Invalid search If the employee name is invalid name, system displays error message. Invalid Permission
Includes:	none
Special Requirements:	Only the owner should be able to view the employee details
Assumptions:	none
Notes and Issues:	none

Use Case ID:	2		
Use Case Name:	searchItem		
Created By:	Sashank Rijal	Last Updated By:	01-04-2021
Date Created:	01-04-2021	Date Last Up-dated:	

Actor:	Customer, Employee
Description:	The customer can check the availability of a specific item they want to buy or employee can search for a particular item.
Preconditions:	The customer or employee should be logged in
Postconditions:	The item menu will be displayed • The price and details of the item will be displayed
Priority:	High
Frequency of Use:	none

Normal Course of Events:	Customer or employee logs in • The customer or employee searches for the desired item • The system displays the details of the searched item
Alternative Courses:	Item is currently unavailable message is displayed.
Exceptions:	Invalid search
Includes:	none
Special Requirements:	none
Assumptions:	none
Notes and Issues:	none

Use Case ID:	3		
Use Case Name:	Trending Items		
Created By:	Prason Poudel	Last Updated By:	01-04-2021
Date Created:	01-04-2021	Date Last Updated:	

Actor:	Customer,employee,owner
Description:	The user can check the item with the maximum sales this week
Preconditions:	Search for the item first
Postconditions:	• The user can see the trending items • The system displays the details about the trending item • The user has the option to add the item to the cart
Priority:	high
Frequency of Use:	none
Normal Course of Events:	• When the user wants to view the trending items at a specific date • System displays the trending items and their respective week • The user selects the particular trending item they want to view the details of • System displays that specific item with their details • User confirms purchase
Alternative Courses:	No trending item is present.
Exceptions:	none
Includes:	purchase
Special Requirements:	none
Assumptions:	none
Notes and Issues:	none

Use Case ID:	4		
Use Case Name:	Add stock		
Created By:	Mickey Kumar Rouniyar	Last Updated By:	01-04-2021
Date Created:	01-04-2021	Date Last Updated:	

Actor:	owner
Description:	The owner can make an entry of new stock which is to be added.
Preconditions:	none
Postconditions:	The owner is able to view the items that is recently added.
Priority:	high
Frequency of Use:	none
Normal Course of Events:	• The menu is displayed to login for the owner. • The owner enters the details of the item he wants to add. • The owner gives the quantity • Make an entry of the item into the database
Alternative Courses:	The new item name is already present in the database
Exceptions:	Invalid authorization
Includes:	none
Special Requirements:	none
Assumptions:	none
Notes and Issues:	none

Use Case ID:	5		
Use Case Name:	Bill creation		
Created By:	Prason Poudel	Last Updated By:	01-04-2021
Date Created:	01-04-2021	Date Last Updated:	

Actor:	Employee
Description:	The employee creates the bill for the purchased item by the customer.
Preconditions:	Items should be selected by the customer
Postconditions:	Transaction is processed and amount is shown
Priority:	high
Frequency of Use:	none
Normal Course of Events:	• The customer selects the items • The customer confirms the purchase • The employee generates the bill • The system displays the total bill
Alternative Courses:	Payment declined, please pay the correct amount again.
Exceptions:	No item is selected by the customer.

Includes:	purchase
Special Requirements:	Revenue
Assumptions:	none
Notes and Issues:	none

Use Case ID:	6		
Use Case Name:	Add employee		
Created By:	Sashank Rijal	Last Updated By:	01-04-2021
Date Created:	01-04-2021	Date Last Updated:	

Actor:	owner
Description:	The owner adds the new employee for the shop
Preconditions:	none
Postconditions:	The owner is able to view the complete information of the employee added.
Priority:	Low
Frequency of Use:	none
Normal Course of Events:	• The menu is displayed to login for the owner. • The owner enters the details of the employee he wants to add. • The owner gives all the information of the employee. • Make an entry of the item into the database
Alternative Courses:	The new employee is already present in the shop
Exceptions:	none
Includes:	Employeeinfo
Special Requirements:	none
Assumptions:	none
Notes and Issues:	none

Use Case ID:	7		
Use Case Name:	Search employee		
Created By:	Mickey Kumar Rouniyar	Last Updated By:	01-04-2021
Date Created:	01-04-2021	Date Last Updated:	

Actor:	owner
Description:	The owner can search for any employee based on any parameters or information.
Preconditions:	The owner should be logged in
Postconditions:	The employee menu will be displayed • The details of the employee will be displayed

Priority:	Low
Frequency of Use:	none
Normal Course of Events:	The owner logs in • The owner searches for an employee • The system displays the details of the searched employee
Alternative Courses:	Employee is not present in the shop
Exceptions:	Invalid search
Includes:	EmployeeInfo
Special Requirements:	none
Assumptions:	none
Notes and Issues:	none

Use Case ID:	8		
Use Case Name:	Check profit or loss		
Created By:	Prason Poudel	Last Updated By:	01-04-2021
Date Created:	01-04-2021	Date Last Updated:	

Actor:	owner
Description:	The owner can check for the overall profit or loss of his shop based on each item and quantity and report is given.
Preconditions:	The owner should be logged in and at least one item should be sold.
Postconditions:	The owner is able to view the profit and loss report.
Priority:	Medium
Frequency of Use:	none
Normal Course of Events:	• The owner logs in the system • The owner requests for the report. • The system displays the profit and loss report • The owner views the report.
Alternative Courses:	• No report is generated due to no item sold.
Exceptions:	none
Includes:	Revenue
Special Requirements:	none
Assumptions:	none
Notes and Issues:	none

Use Case ID:	9		
Use Case Name:	Manage Stock		
Created By:	Sashank Rijal	Last Updated By:	01-04-2021
Date Created:	01-04-2021	Date Last Updated:	

Actor:	employee
Description:	The employee can change the information of the item
Preconditions:	Item name should be present.
Postconditions:	Update the information in the database.
Priority:	Low
Frequency of Use:	none
Normal Course of Events:	1. Employee compares the new information with the old information. 2. The employee changes the required information.
Alternative Courses:	The item is not present in the database. The item cannot be modified.
Exceptions:	Invalid details
Includes:	Check item, check expiry dates
Special Requirements:	none
Assumptions:	none
Notes and Issues:	none

Use Case ID:	10		
Use Case Name:	revenue		
Created By:	Mickey kumar Rouniyar	Last Updated By:	01-04-2021
Date Created:	01-04-2021	Date Last Updated:	

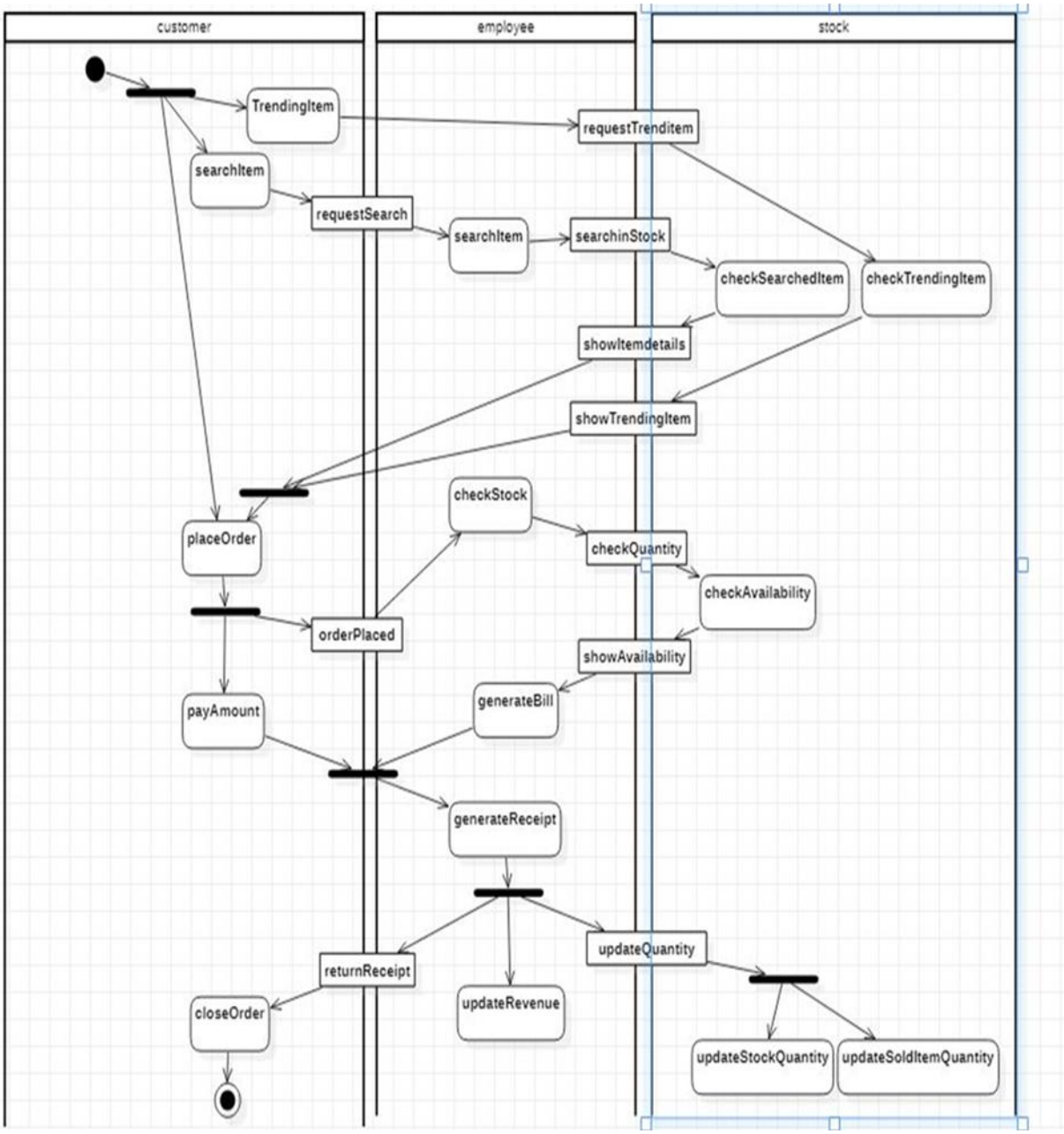
Actor:	Owner
Description:	The owner gets to view the revenue for his shop based on a particular day, week or year.
Preconditions:	The owner should be logged in.
Postconditions:	The owner can see the sold items The system displays the appropriate profit and loss generated. The owner can see the overall stock report.
Priority:	high
Frequency of Use:	none
Normal Course of Events:	• When the owner wants to view the report of his shops sales. • System displays the report based on the day, week and year. System displays the sold stock report, profit generated report and item left report overall.
Alternative Courses:	Single day report might be unavailable due to no sales on that day and only revenue report will be displayed based on week and year.
Exceptions:	none
Includes:	purchase
Special Requirements:	none

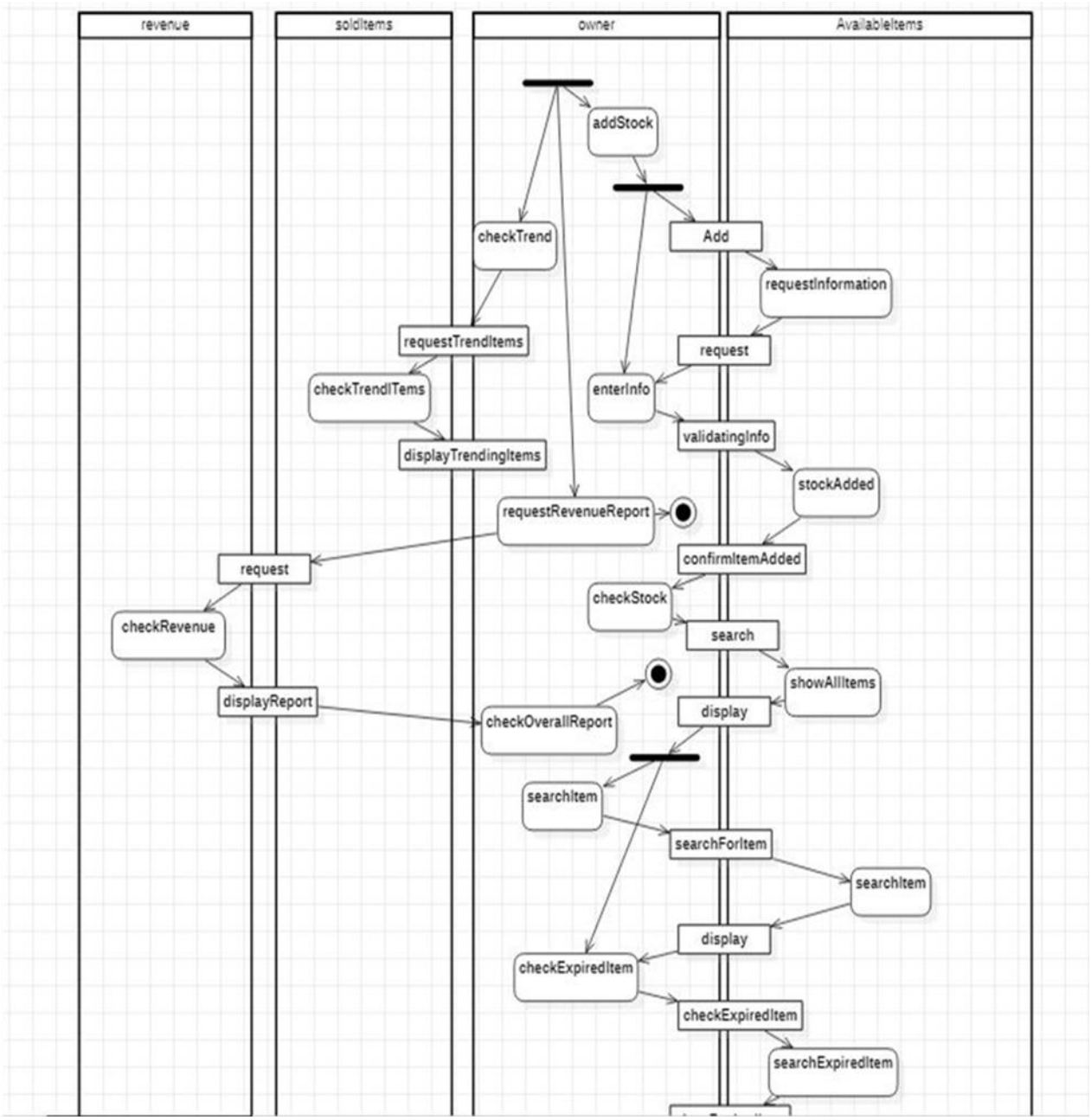
Assumptions:	none
Notes and Issues:	none

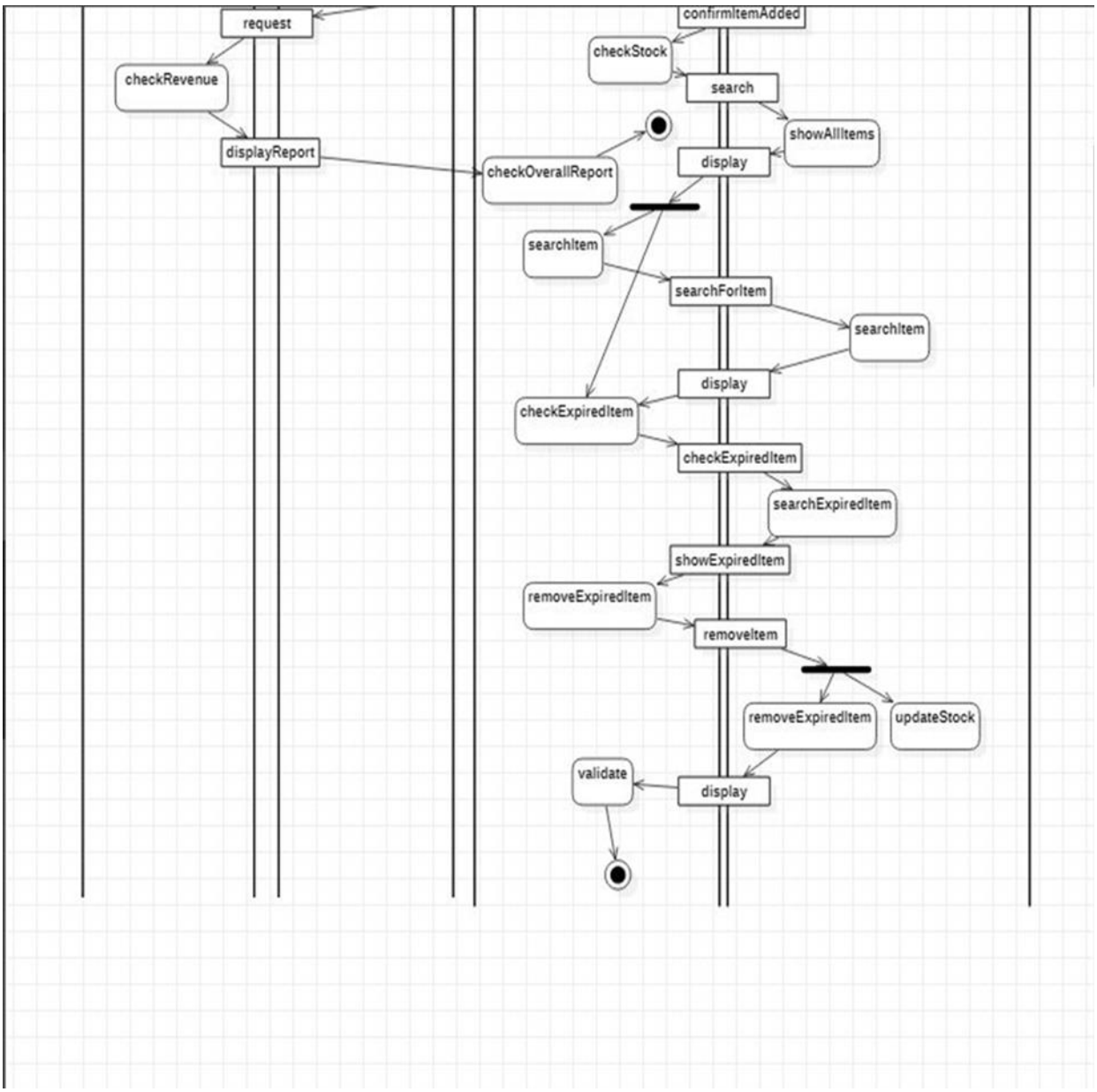
Use Case ID:	11		
Use Case Name:	login		
Created By:	Prason Poudel	Last Updated By:	01-04-2021
Date Created:	01-04-2021	Date Last Updated:	

Actor:	Customer,owner,employee
Description:	Login to use all the services for the shop.
Preconditions:	Must be a registered user
Postconditions:	Display the menu with services.
Priority:	Very high
Frequency of Use:	none
Normal Course of Events:	User is given a menu for entering the credentials User is asked to input username and password System displays the menu with appropriate services
Alternative Courses:	none
Exceptions:	Invalid user
Includes:	register
Special Requirements:	none
Assumptions:	none
Notes and Issues:	none

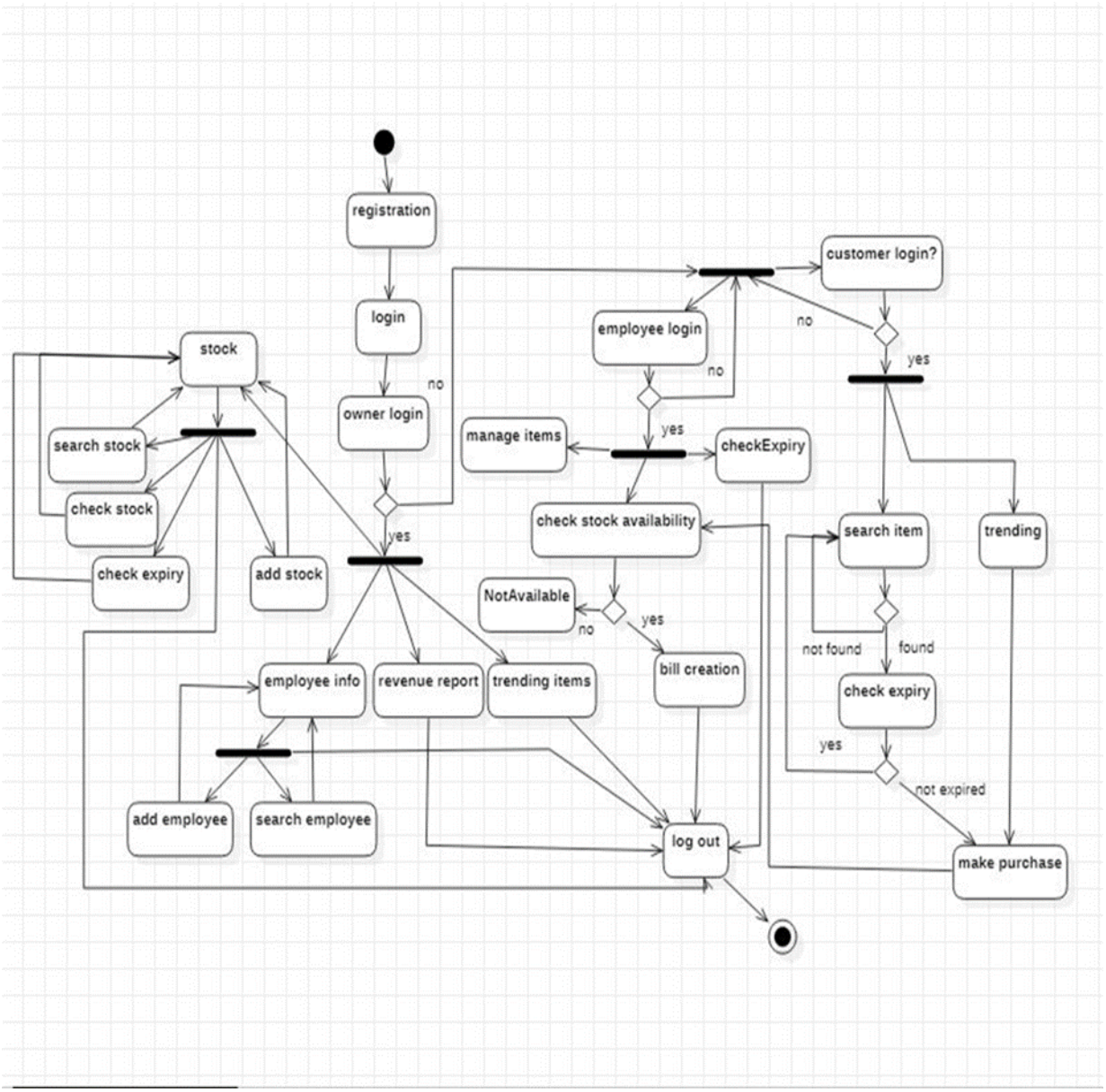
- **Activity diagram using swimlane concept**



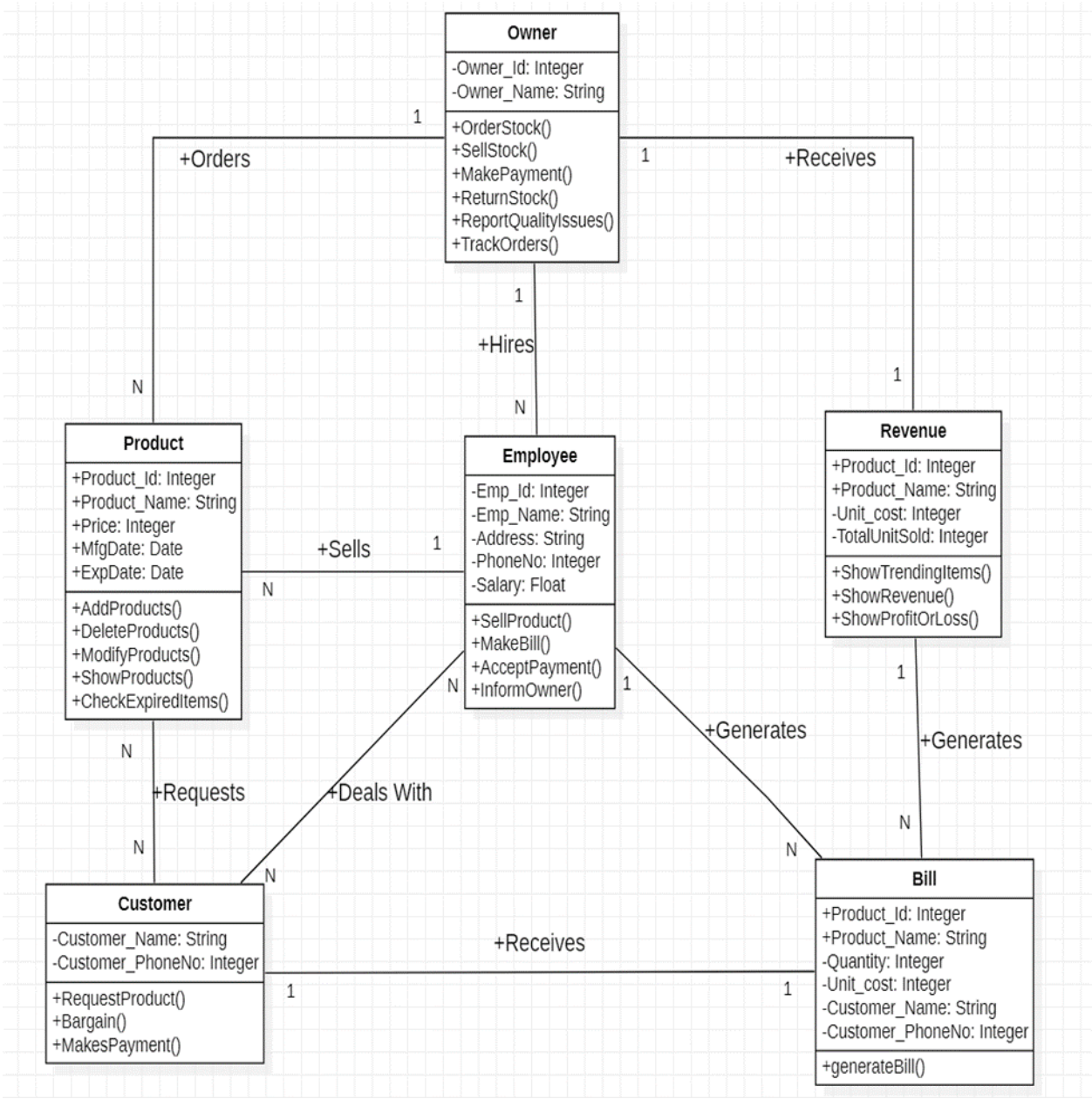

revenue
soldItems
owner
AvailableItems
addStock
checkTrend
Add
requestInformation
requestTrendItems
request
checkTrendITems
enterInfo
validatingInfo
displayTrendingItems
stockAdded
requestRevenueReport
confirmItemAdded
request
checkStock
checkRevenue
search
showAllItems
displayReport
checkOverallReport
display
searchItem
searchForItem
searchItem
display
checkExpiredItem
checkExpiredItem
searchExpiredItem


request
checkRevenue
displayReport
confirmItemAdded
checkStock
search
showAllItems
display
checkOverallReport
searchItem
searchForItem
searchItem
display
checkExpiredItem
checkExpiredItem
searchExpiredItem
showExpiredItem
removeExpiredItem
removeItem
removeExpiredItem
updateStock
validate
display

Without using swimlane concept



- **Class diagram**



- **CRC CARD**

Owner	
• OrderStock() • SellStock() • MakePayment() • ReturnStock() • ReportQualityIssues() • TrackOrders()	• Product • Employee • Revenue

Product	
• AddProduct() • DeleteProduct() • ModifyProduct() • ShowProduct() • CheckExpiredItems()	• Owner • Customer • Employee

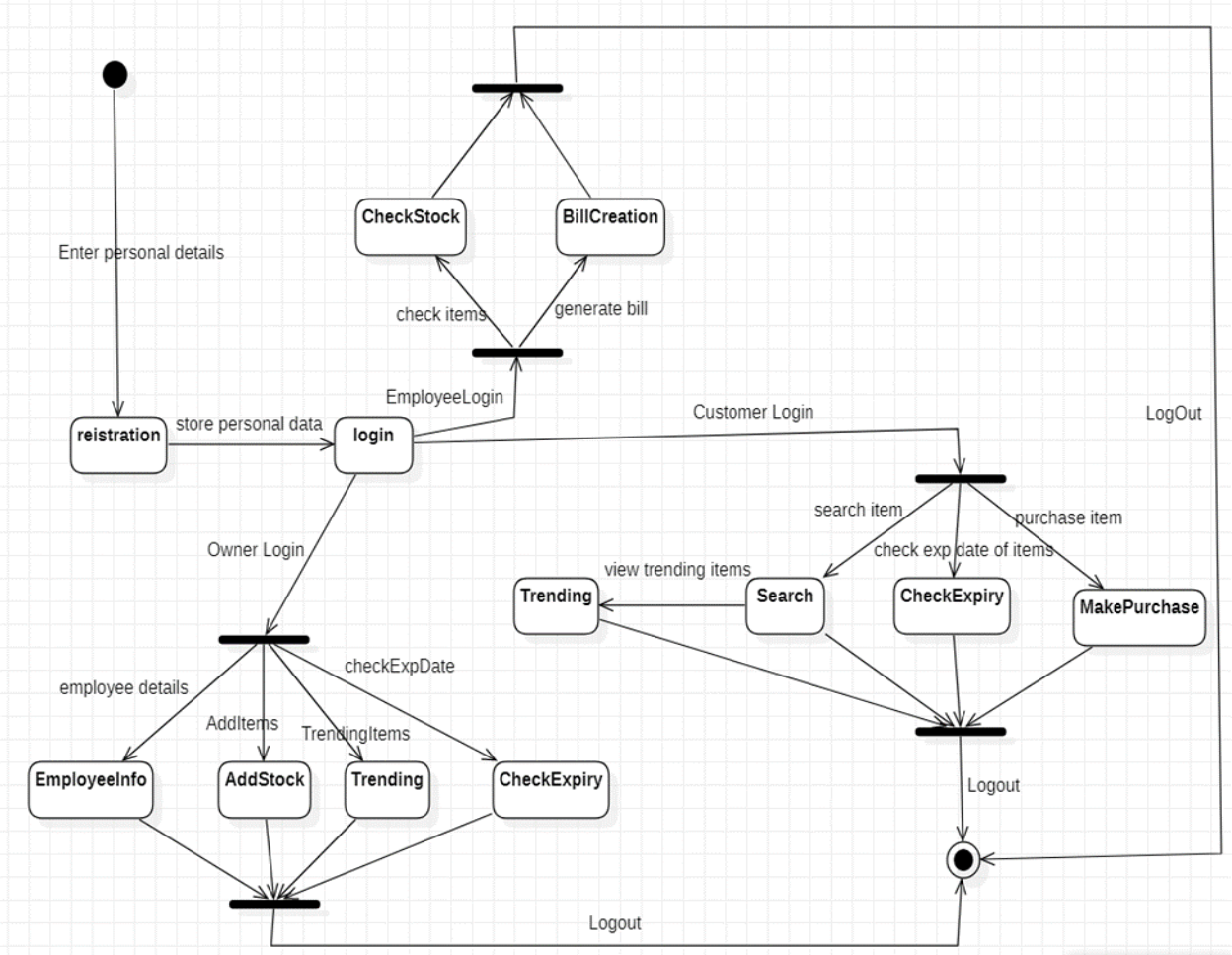
Customer	
• RequestProduct() • Bargain() • MakesPayment()	• Product • Employee • Bill

Employee	
• SellProduct() • MakeBill() • AcceptPayment() • InformOwner()	• Customer • Product • Owner • Bill

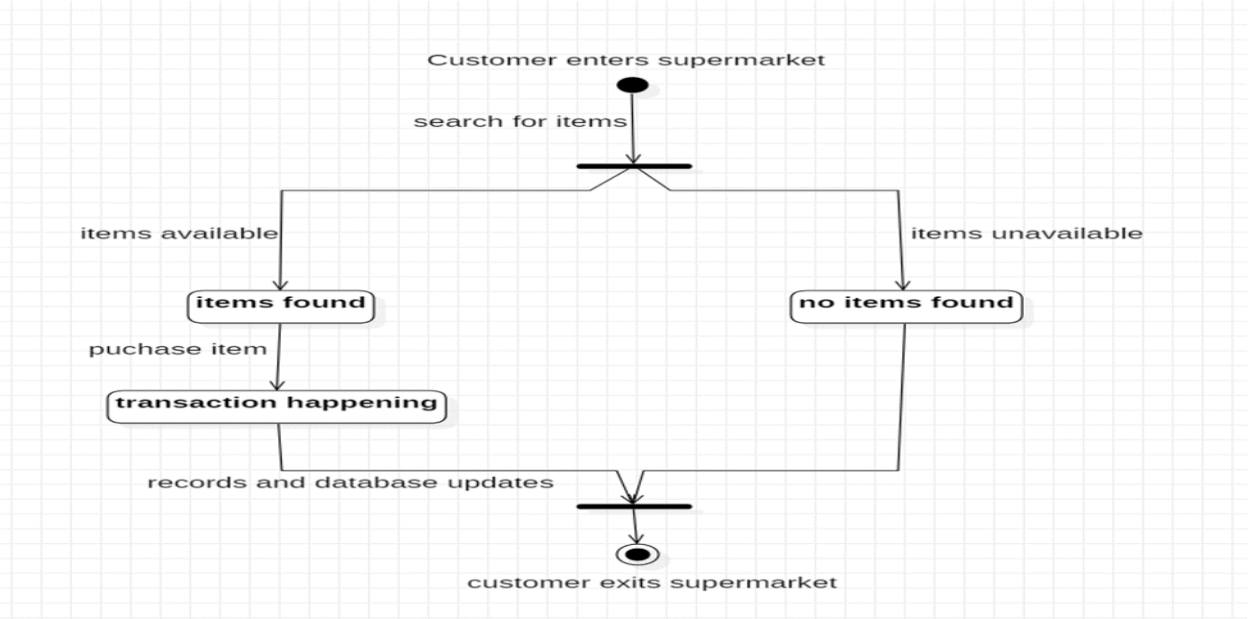
Bill	
• generateBill()	• Customer • Employee • Revenue

Revenue	
• ShowTrendingItems() • ShowRevenue() • ShowProfitOrLoss()	• Bill • Owner

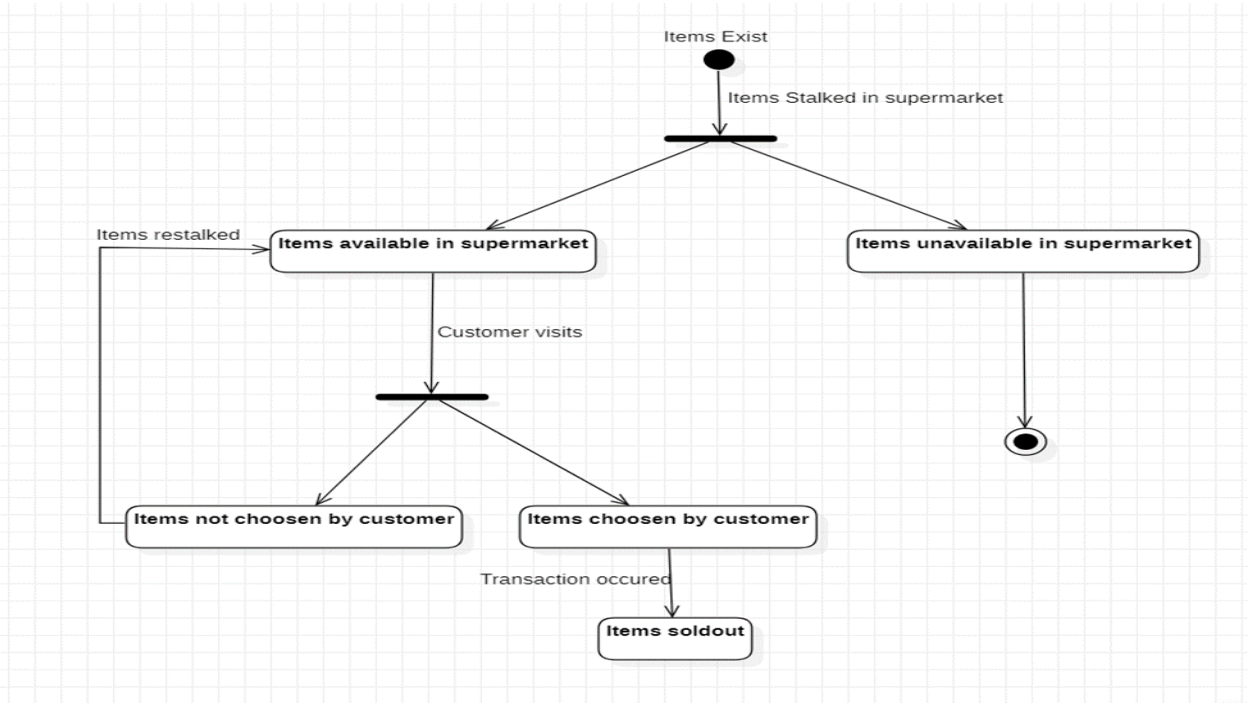
- **State chart diagram**



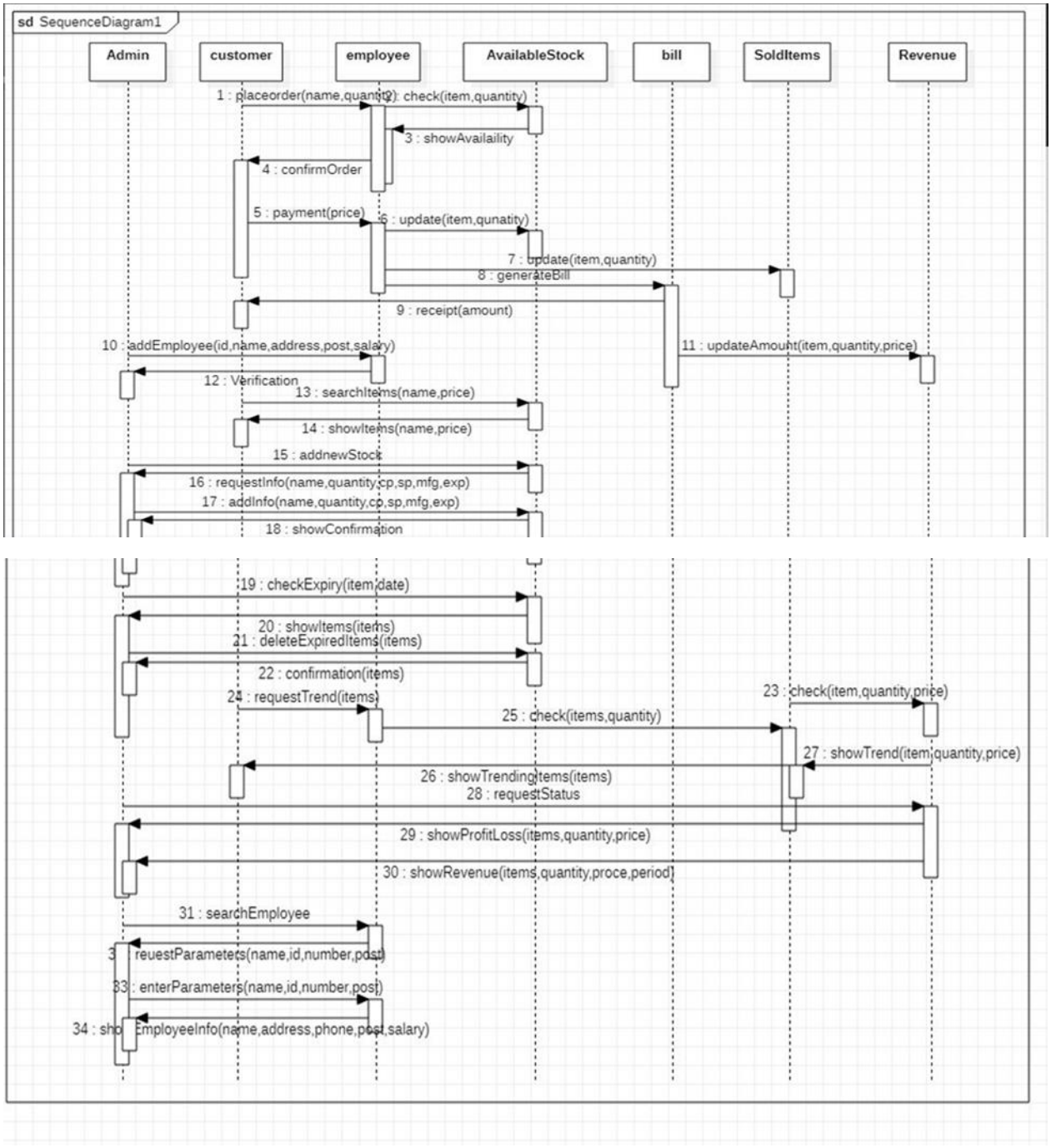
For customer class



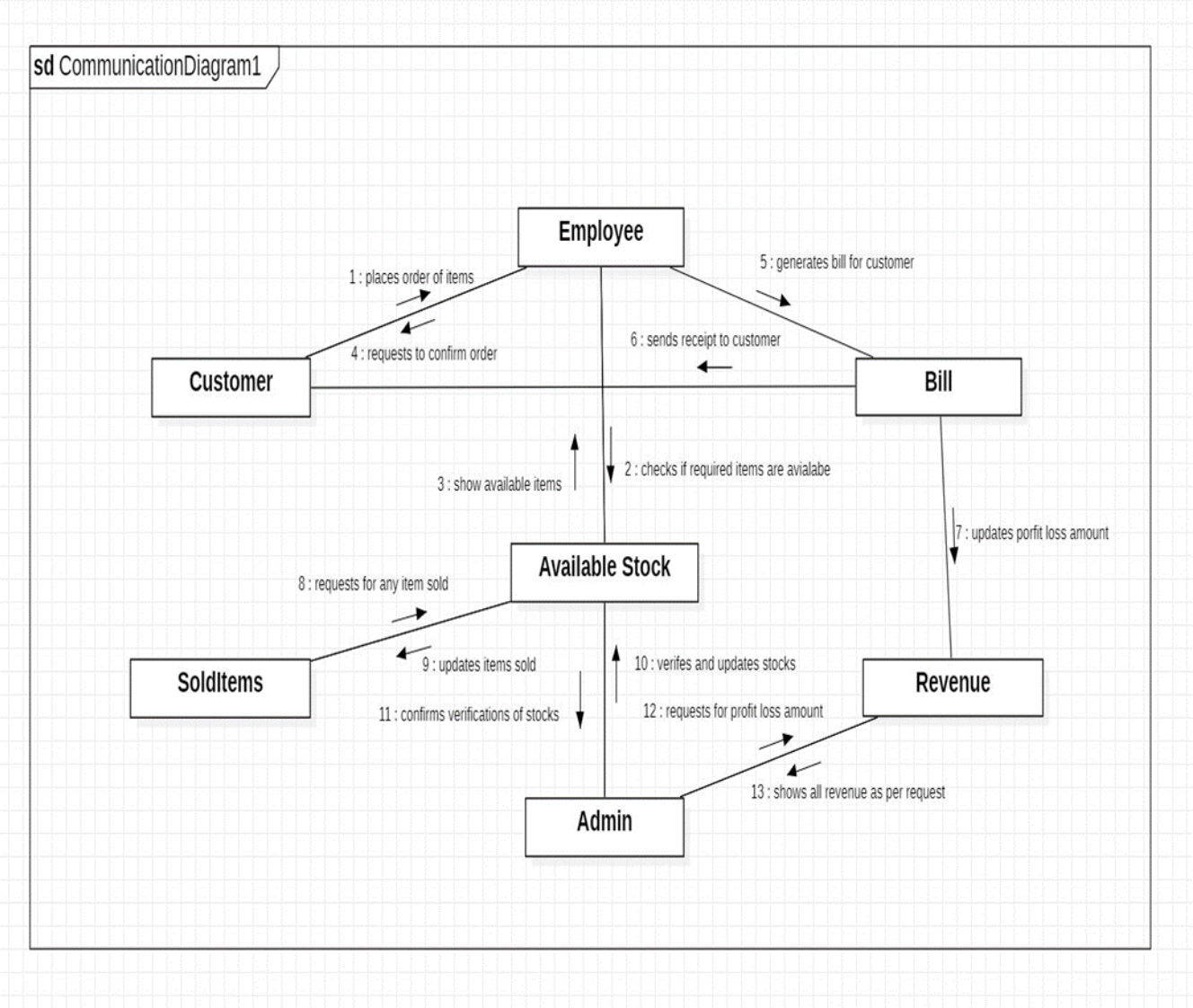
For product class



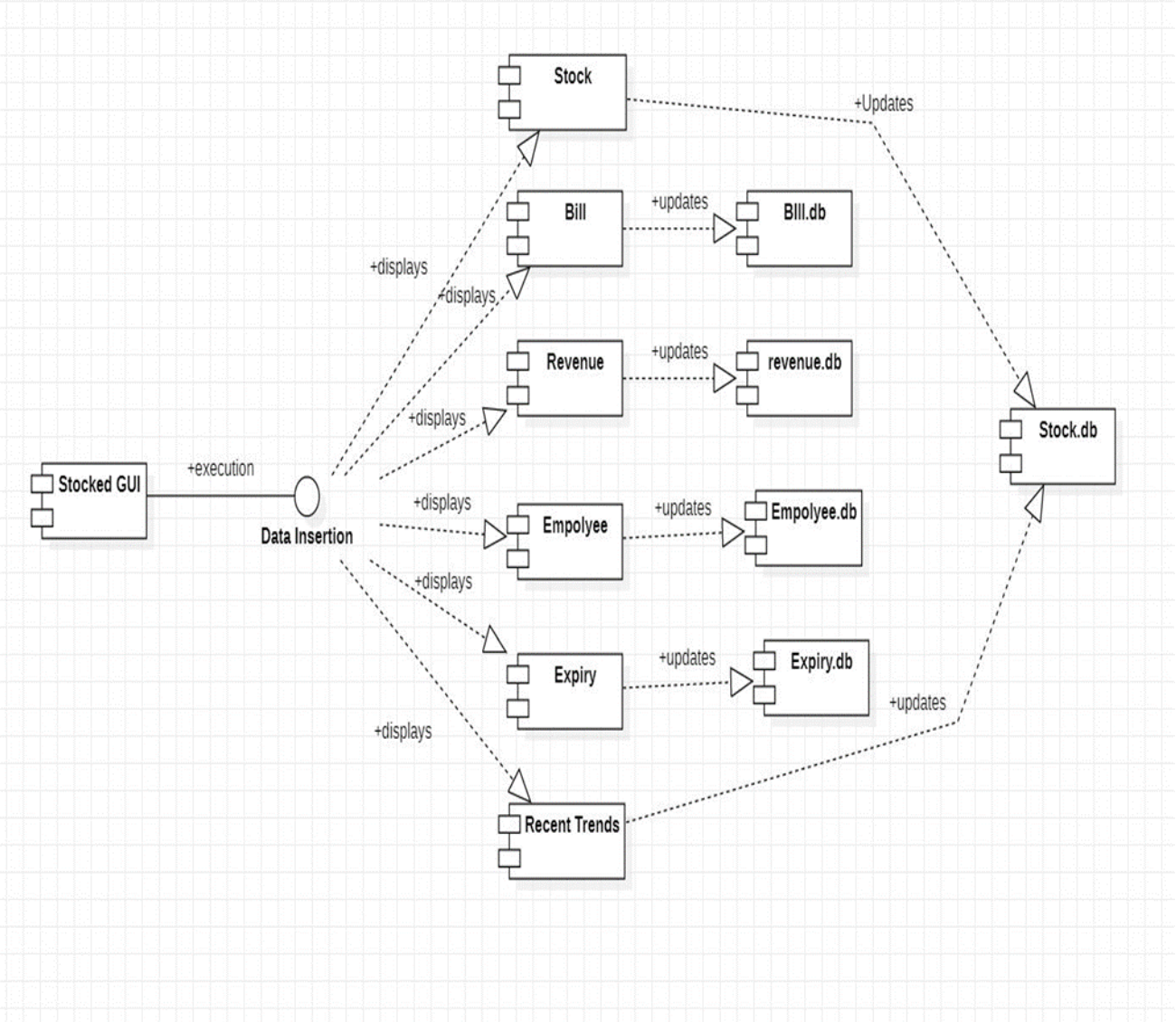
- **Sequence Diagram**



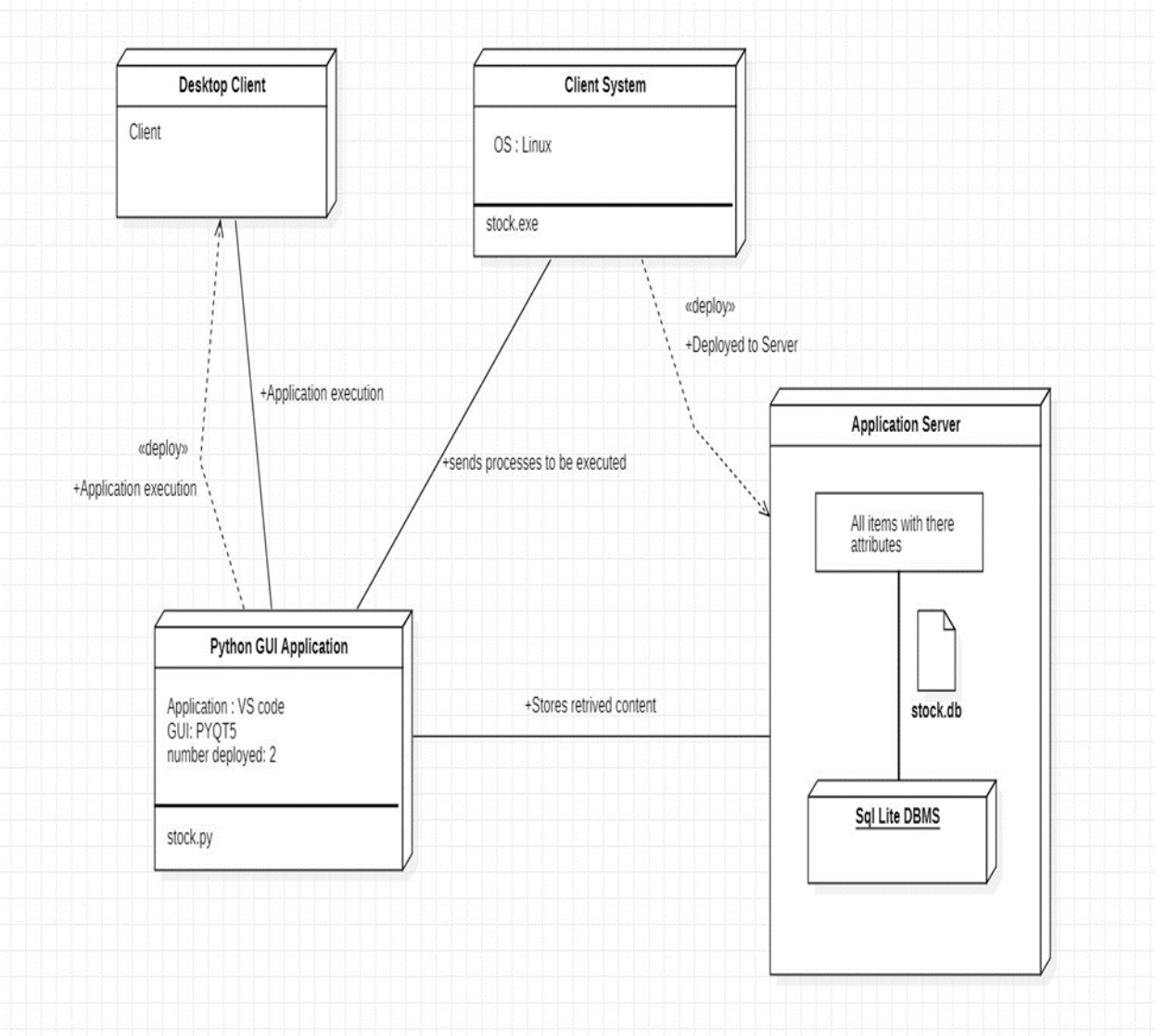
- **Collaboration/communication Diagram**



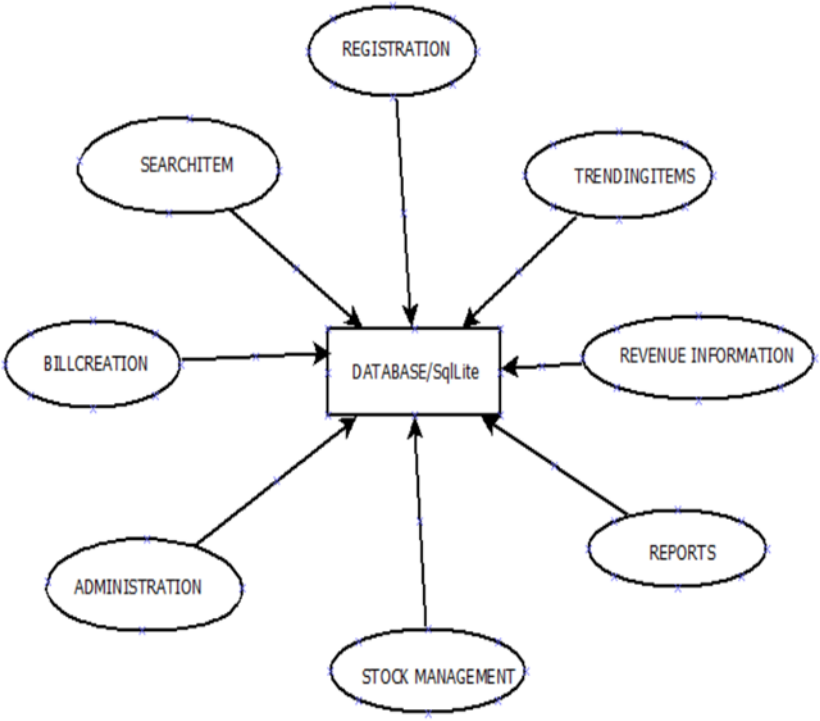
- **Component Diagram**



- **Deployment Diagram**

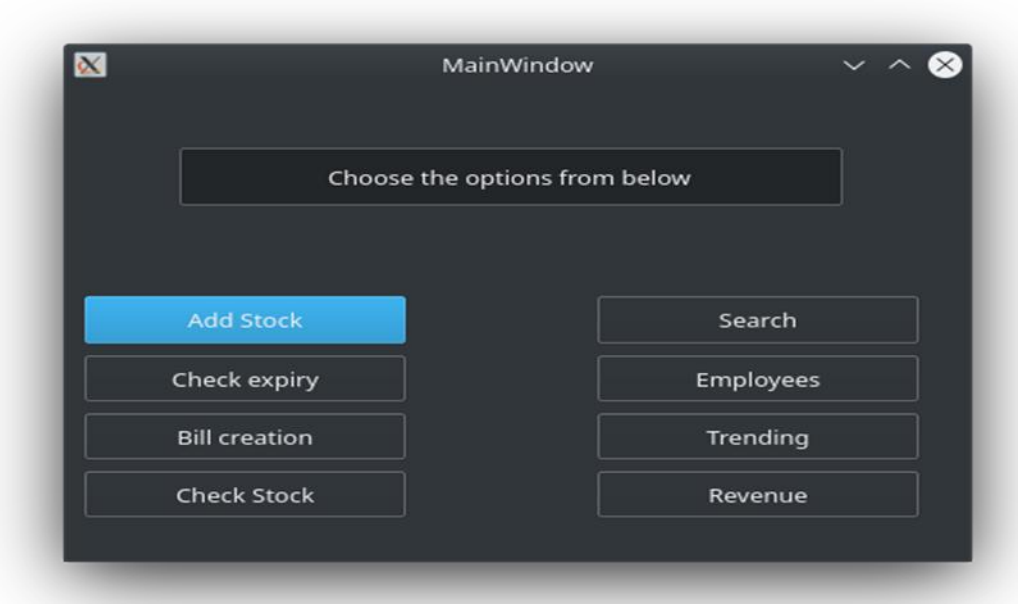


In our project, we are using **database-centric architecture** because:
→In our software, Database plays very important role such as all the information are stored in the database which further used by the functional properties like while searching item, creating bill etc.

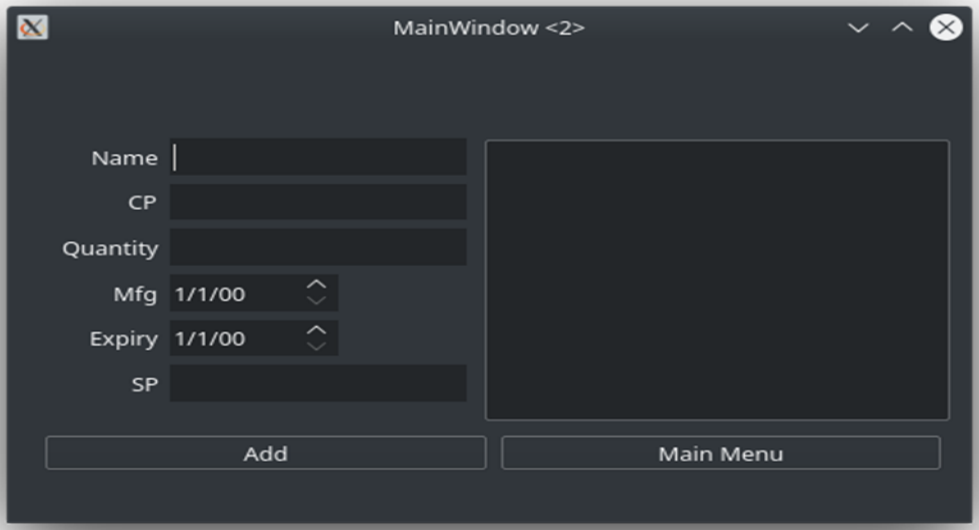


Authorization page

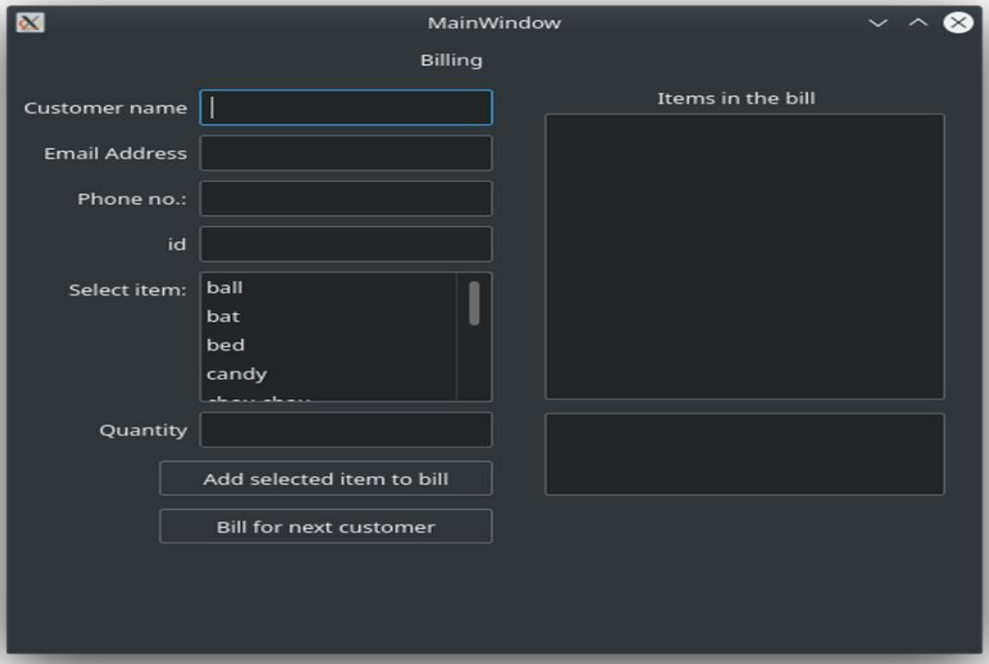
Main window or dashboard



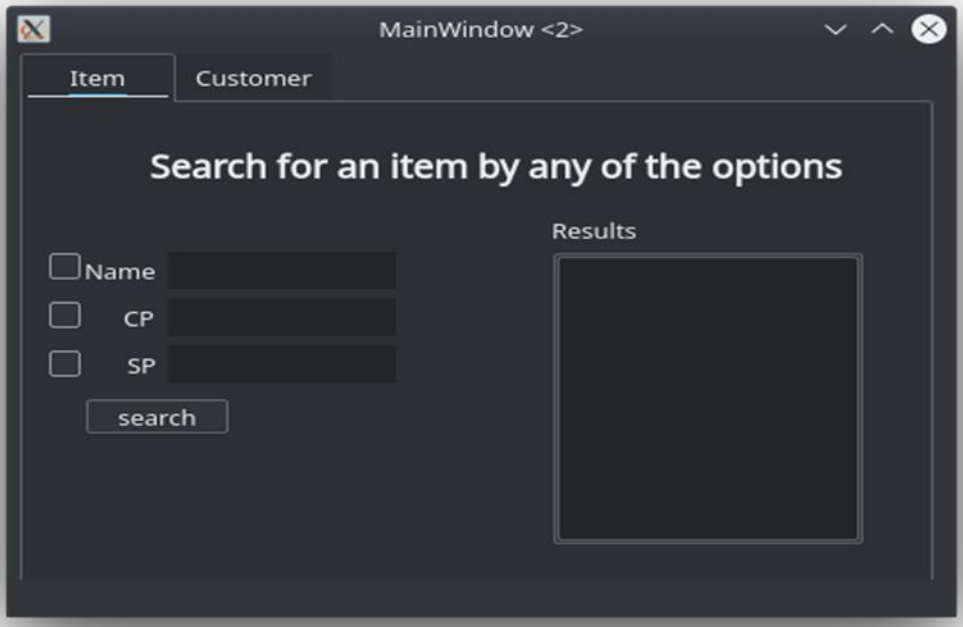
Add stock



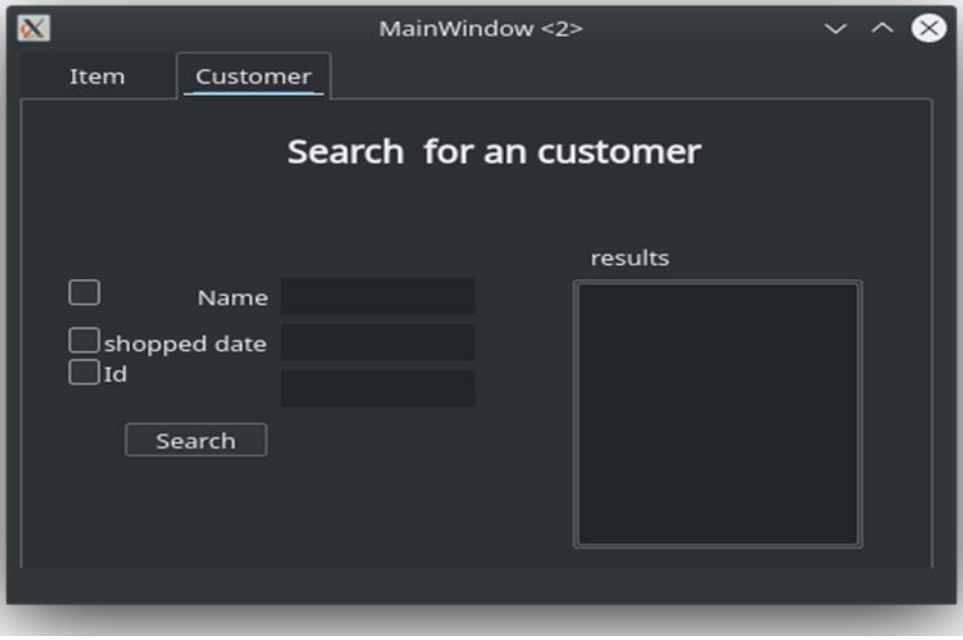
Bill creation



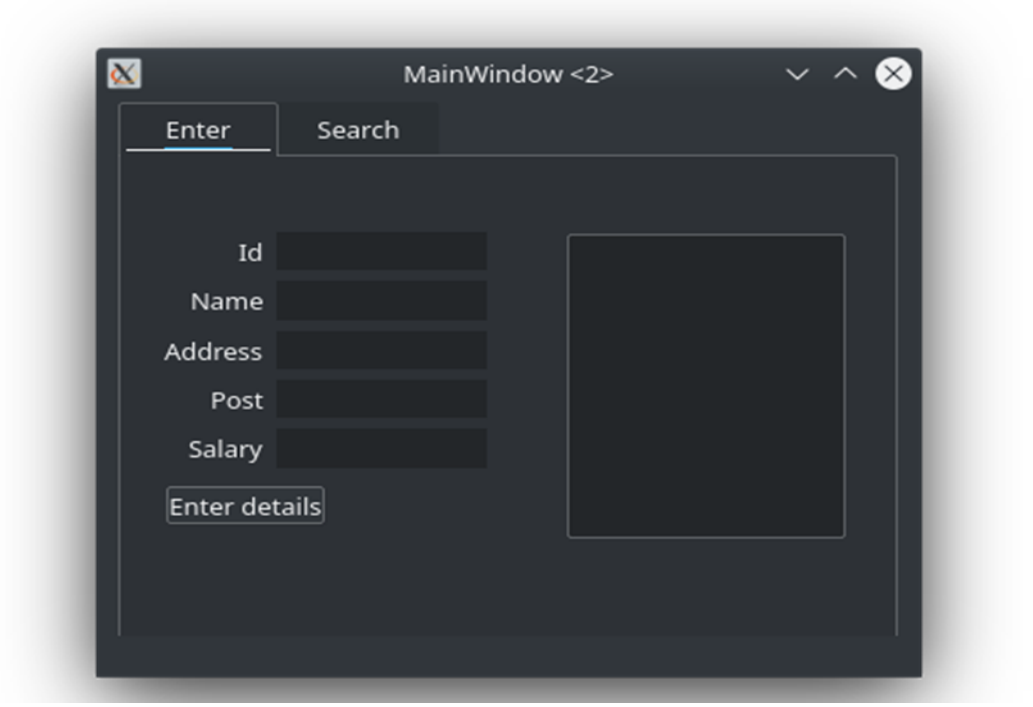
Search for an item



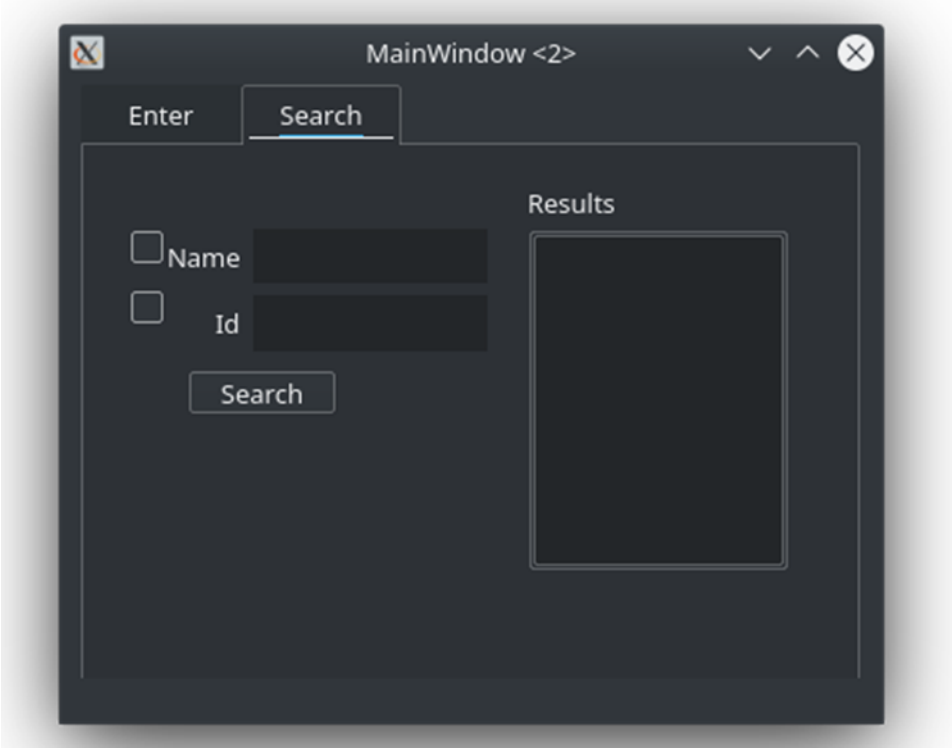
Search for a customer



Enter employee info

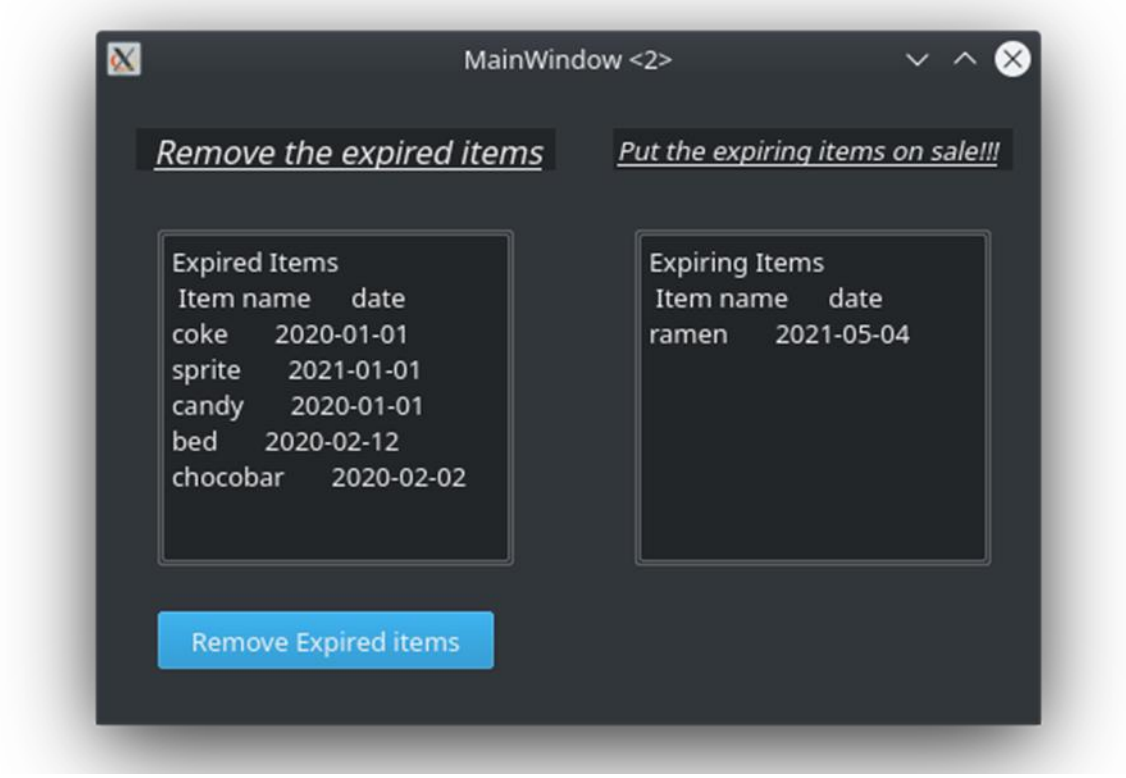


Search employee

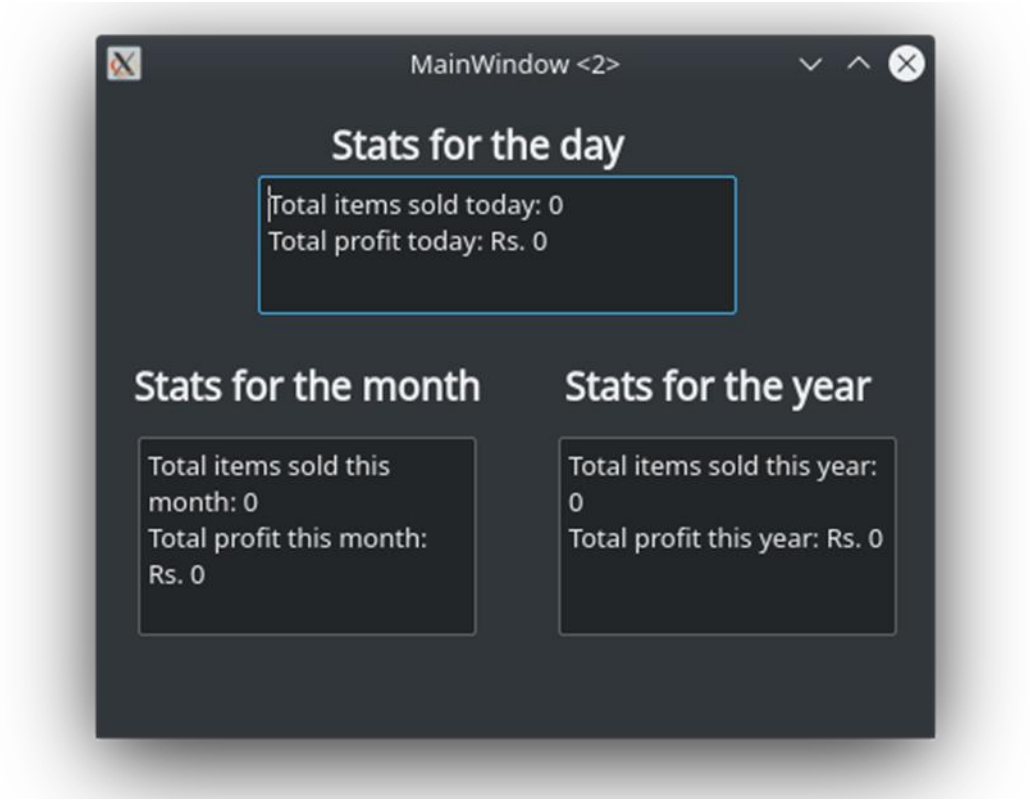


Check stock

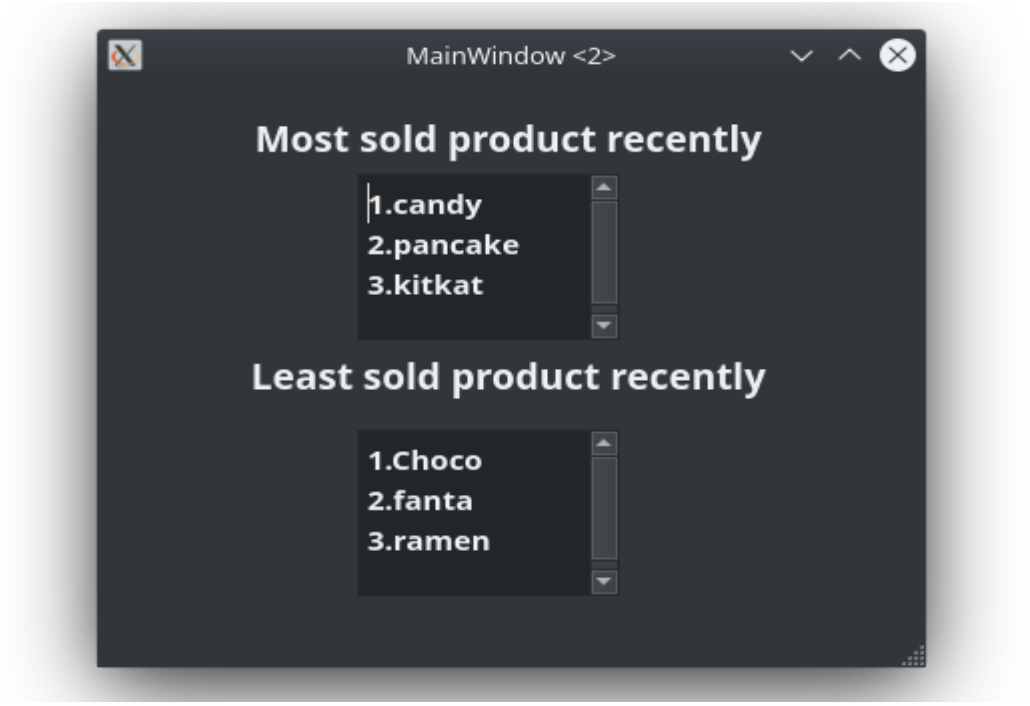
Check expiry

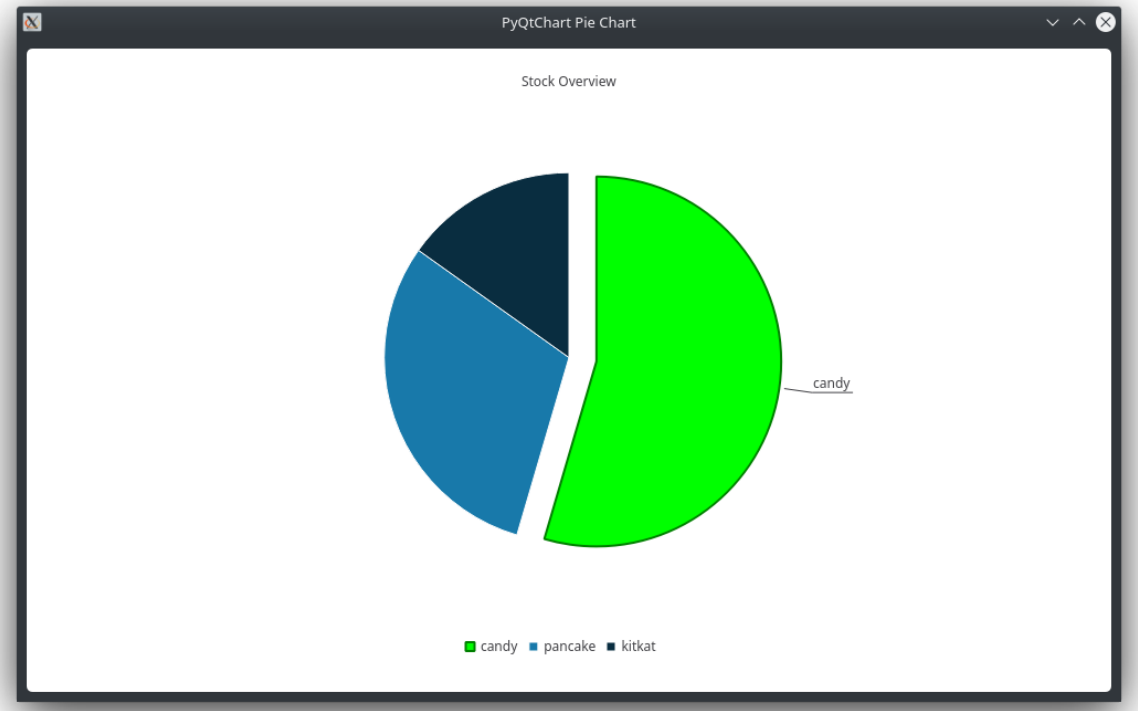


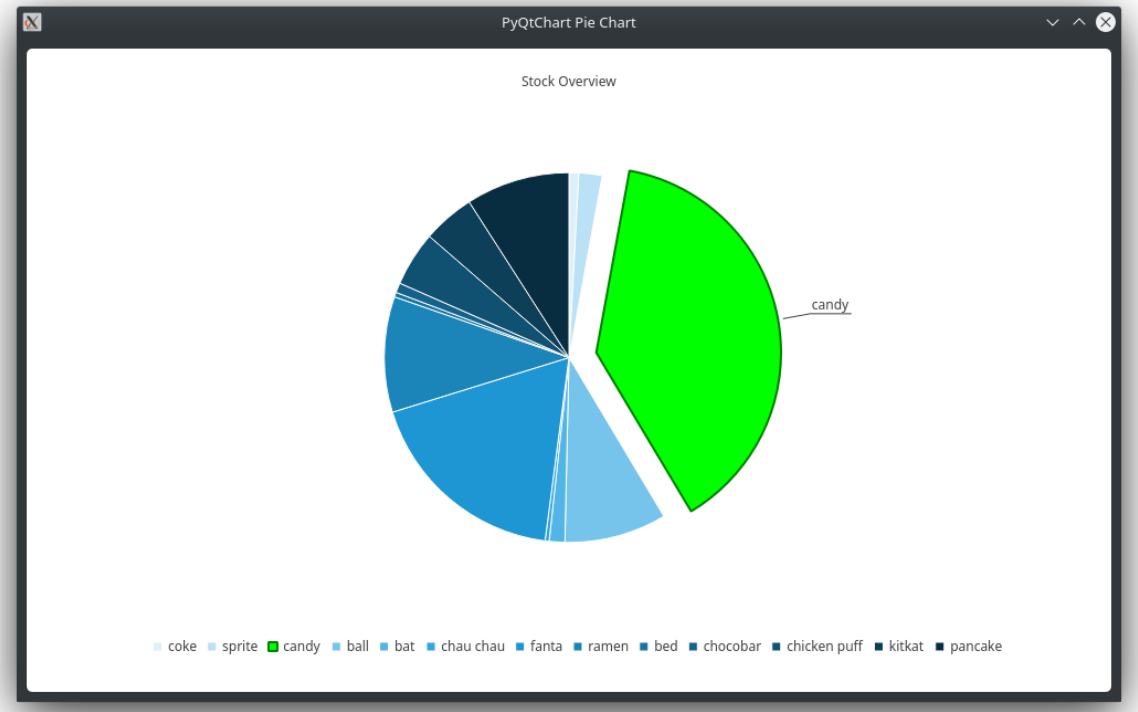
Revenue



- **Trending items/not to invest alert**




PyQtChart Pie Chart
Stock Overview
candy
candy
pancake
kitkat


PyQtChart Pie Chart
Stock Overview
candy
coke
sprite
candy
ball
bat
chau chau
fanta
ramen
bed
chocobar
chicken puff
kitkat
pancake

- **Risk Assessment**

UNIVERSITY OF MINNESOTA
Driven to Discover℠

UNIVERSITY SERVICES
PROGRAM.MANAGEMENT.OFFICE

PROJECT RISK ASSESSMENT

Project Name: Stocked - a complete organising solution	**Project Size: 3**
Project Manager: Sashank, Mickey, Prason	**Date: 22/05/2021**
Risk Score: 285	**Risk Rating:Medium RISK**

Project Risk Score & Risk Rating Guidelines	
0 to 110:	LOW RISK
115 to 290:	MEDIUM RISK
300 or higher:	HIGH RISK

The Project Risk Assessment is a self-evaluation of the project risks by identifying activities that typically represent risk. By completing this worksheet at the time of Project Origination, you can provide additional information that is relevant to understand the potential impacts this effort might have.

Review each question below and assign a risk rating of 0-40 based on the most relevant risk category item and its relevance to your project. **Risk scores of 30 or 40 must be explained in the Comments column; these specific risks must then be documented with a risk mitigation plan in the Project Charter or Project Management Plan.**

If you have any questions about completing this document, please contact the PMO (Bill Kanfield kanfi001@umn.edu or 612-626-4223).

#	Risk Category	Score	Comments
	BUSINESS RISK	60	N/A
1	**Implementation complexity and impact to processes and business areas:**	25	
	- Implementation is not complex	0	the budget of the buisness and the scale value should be analyzed before hand while running the software
	- Implementation is complex, though team has prior experience	15	
	- Implementation is complex with many areas impacted and new methods introduced	10	
2	**Business Dependencies that might impact implementation:**	15	
	- No impact, No business dependencies exist	5	to reduce the economic risk, more money should be saved and should be operated with the suitable budget for future purpose
	- Yes, some business dependencies exist but are low risk	10	
	- Yes, there are key business dependencies that might impact project implementation	0	
3	**Outages required during the implementation:**	20	
	- None required	10	
	- Outages to be planned after business hours	5	
	- Outages to be planned during business hours	5	
	TECHNOLOGY RISK	190	N/A
4	**Type of Project:**	60	
	- Maintenance (e.g., correct problems)	5	the system needs to be timely checked for maintainance purpose and to detect any anomalies in the system
	- Enhancement (e.g., add new features)	5	
	- Product introduction	5	
	- Process improvement/re-engineering	0	
	- Upgrade existing software or applications	15	
	- New development – replace existing system	0	
	- New development – develop system to support new business	0	
	- Implementation of software package in-house	0	
	- Outsourcing to external vendor	30	

5	**External Vendor Involvement:**	**40**	
	- No external or internal vendor required	0	existing external vendors like database and GUI support parties are highly involved and major part of the system relies on them
	- No, an internal partner will be used for some project activites	0	
	- Yes, an existing external vendor will be used	15	
	- Yes, a known external vendor will be used but a new contract or PO is required	20	
	- Yes, and external vendor is required but not yet identified	5	
6	**Level of vendor support necessary for the technology after implementation:**	**60**	
	- N/A (no vendors involved)	0	the support of vendors is very high
	- No subsequent vendor support necessary	0	
	- Minor support necessary (e.g., maintenance only)	30	
	- Major support necessary (e.g., programming, upgrades, etc.)	30	
7	**Newness of Technology (hardware, systems, databases, communication, etc.) used in the project:**	**10**	
	- N/A (no technology involved)	0	
	- Technology is not new and has been proven or tested	10	
	- New to the business line	0	
	- New to Services	0	
	- New to the supermarket management services industry	0	
8	**Availability of Project Team Resources**	**20**	
	- We have the resources with the appropriate skill sets	5	the team member should have familarity with the software to be used and also should be available for future maintainance and risk mitigation
	- We have resources but they will need to be trained	10	
	- The project will require external resources	5	
	IMPLEMENTATION RISK	**35**	**N/A**
9	**Will the Department of Food/product Technology and Quality Control be used during the negotiation, creation, or modification of the contract?**	**10**	
	- No contract is required for project.	0	the quality standards must be followed, consumer rights shouldn't be challenged and all the necessary taxes should be paid to the department
	- No, the existing contract will be used and no modifications or addendums will be required.	10	
	- Yes, the DFTQC will be highly involved with contract negotiation, creation, or modification.	0	
	- Yes, the DFTQC will be involved, but in a limited capacity	0	
	- No, the DFTQC will not involved with the contract negotiation, creation, or modification.	0	
10	**Has the Business Case been developed, reviewed and approved:**	**25**	
	- Yes, the Business Case/justification is documented and fully approved	5	the shop/supermarket buisness must be approved by the concerned departments and must acquire a license for buisness
	- Yes, but there are some unknowns or assumptions the business case, stakeholders or product/service deliverables	15	
	- No, there are many unknowns or assumptions around the business case, stakeholders, strategy or product/service deliverables	5	
11	**Additional project risks not previously addressed [1]:**	**0**	
	-	0	
	-	0	
	-	0	
	TOTAL PROJECT RISK SCORE	**285**	**FINAL PROJECT RISK RATING: [RATING]**

[1] Separately list and rank each additional risk not previously addressed that is unique to your business line, technology used, or project using the following scale:
Low risk = 0-10; Medium risk = 11-20; High risk = 21-40. Add additional lines as needed.

Project Risk Score & Risk Rating Guidelines
0 to 110: LOW RISK
115 to 290: MEDIUM RISK
300 or higher: HIGH RISK

#	Risk statement		(Scale 1-100%)	(Scale 1-10)	Exposure	Mitigation	Contingency	Triggers	Assignee
	Condition	Consequence	Probability	Impact					
1	Customer and employee data might require conversion. None is currently planned for.	Reduced ability of the system to access and manipulate the data as required and might result in crashing of the system and database	50%	8	6.4	Determine extent of problem. If necessary, development of scripts to standardize the format of data for entry initially.	It might be possible to develop code to access the nonstandard-format data. If not, a migration subproject will be necessary.	When a decision is made about the percentage of nonstandard data and its impact on the system	Administrator
2	Inconisitency in the support and services given to the customer.	Unsatisfied customer can reduce the reputation of the shop.	50%	3	3	A robust AI to be developed to support customer services.	Considering the factors like budget, the shop can take support from IT customer support centres.	When the AI system is turned unresponsive.	Administrator/ Employee
3	New data might not be available for searching of product or a new employee on a timely basis.	Decisions could be made on the basis of incomplete information or old information.	20%	7	6	all the new information added in the system should be scheduled in timely manner during search.	give sets of skills of skills to the actors of the search functionality like running sql queries directly from the databse.	When new information is added into the system of a shop.	Administrator
4	Employee might not be able to state the statistical data needed.	Incorrect requirements might be added.	10%	8	2	Ensure that the statistical data needs are defined in detail and is verified by the administrator.	If any data needs are not clearly stated, consider removing such requirements from the first phase of the project.	When it is known that clear requirements cannot be obtained	Administrator
5	The existing database administrator might not be available 50 percent of the time, as required.	The project could be delayed. Incorrect decisions could be made.	40%	6	1.8	Gain commitment to 50 percent of available database administrator time.	Prioritize questions to be asked of the database administrator. When possible, determine other sources of information.	Start of project	Administrator.

Manual Testing

Test Case Description	Login – Positive test case	**Test Priority**	High
Pre-Requisite	A valid admin account	**Post-Requisite**	NA

Test Execution Steps:

S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Application**	**Test Result**	**Test Comments**
1	Launch application	Run Application as Administrator	Login Page	Login Page	VsCode	Pass	[Sashank 03/05/2021 09:30 AM]: Launch successful
2	Enter correct Username & Password and hit login button	Username :Sashank Password:*****	Login success and dashboard launches	Login success and dashboard launches	VsCode	Pass	[Sashank 03/05/2021 09:31 AM]: Launch successful

Test Scenario ID	AddStock-2	**Test Case ID**	AddStock-2A
Test Case Description	Add Stock – Positive test case	**Test Priority**	High
Pre-Requisite	A Successful Login	**Post-Requisite**	NA

Test Execution Steps:

S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Application**	**Test Result**	**Test Comments**
1	Launch Window	Select AddStock Option from Window	AddStock GUI	AddStock GUI	VsCode	Pass	[Sashank 03/05/2021 09:35 AM]: Launch successful
2	Give Name CP, Quantity, MFG, Expiry and SP then Select Add button	Name: Ball CP: 15 Quantity: 200 MFG: 03/03/2021 Expiry: 03/05/2022 SP:25	Stock Insertion Successful	Stock Insertion Successful	VsCode	Pass	[Sashank 03/05/2021 09:36 AM]: Launch successful

Test Scenario ID	CheckStock-3	**Test Case ID**	CheckStock-3A
Test Case Description	CheckStock– Positive test case	**Test Priority**	Medium
Pre-Requisite	A Successful Login and item must be available in the stock	**Post-Requisite**	NA

Test Execution Steps:

S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Application**	**Test Result**	**Test Comments**
1	Launch Window	Select CheckStock option from Dashboard	CheckStock Window with Stock Information	CheckStock Window with Stock Information	VsCode	Pass	[Sashank 04/05/2021 08:40 AM]: Launch successful

Test Scenario ID	CheckExpiry-4	**Test Case ID**	CheckExpiry-4A
Test Case Description	CheckExpiry– Positive test case	**Test Priority**	Medium
Pre-Requisite	A Successful Login and item must be available in stock	**Post-Requisite**	NA

Test Execution Steps:

S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Application**	**Test Result**	**Test Comments**
1	Launch Window	Select CheckExpiry option in Dashboard	List of All Expired items	List of All Expired items	VsCode	Pass	[Sashank 04/05/2021 10:50 AM]: Launch successful
2	Remove Expired items	Select Remove Expired Items Option	All Expired Items Removed	All Expired items Removed	VsCode	Pass	[Sashank 04/05/2021 10:52 AM]: Launch successful

Test Scenario ID	Trending-5	**Test Case ID**	CheckStock-5A
Test Case Description	Trending– Positive test case	**Test Priority**	High
Pre-Requisite	Items must be sold and revenue report	**Post-Requisite**	NA

Test Execution Steps:

S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Application**	**Test Result**	**Test Comments**
1	Launch Window	Select Trending option from Dashboard	List of most and least trending Items	List of most and least trending Items	VsCode	Pass	[Mickey 05/05/2021 01:40 PM]: Launch successful

Test Scenario ID	Revenue-6	**Test Case ID**	Revenue -6A
Test Case Description	Revenue – Positive test case	**Test Priority**	High
Pre-Requisite	A Successful Login and bills should be created	**Post-Requisite**	NA

Test Execution Steps:

S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Application**	**Test Result**	**Test Comments**
1	Launch Window	Select Revenue option from Dashboard	Revenue Statistics for the day, month and year	Revenue Statistics for the day, month and year	VsCode	Pass	[Mickey 05/05/2021 02:00 PM]: Launch successful

Test Scenario ID	Bill creation-7	Test Case ID	Billcreation-7A
Test Case Description	Bill creation – Positive test case	Test Priority	High
Pre-Requisite	A Successful Login	Post-Requisite	NA

Test Execution Steps:

S.NO	Action	Inputs	Expected Output	Actual Output	Test Application	Test Result	Test Comments
1	Launch Window	Select Bill creation Option in Dashboard	Bill creation Window	Bill creation Window	VsCode	Pass	[Mickey 05/05/2021 05:35 PM]: Launch successful
2	Enter Name Email Address, Phone no, item and Quantity, then add Selected item to bill option	Name Email Address: Phone no: Item: Quantity:	Bill Insertion Successful	Bill Insertion Successful	VsCode	Pass	[Mickey 05/05/2021 05:37 PM]: Launch successful
3	Select Bill For Next Customer	Bill For Next Customer	Bill Refreshed and reset	Bill Refreshed and reset	VsCode	Pass	[Mickey 05/05/2021 05:38 PM]: Launch successful

Test Scenario ID	Add Employee-8	Test Case ID	Add Employee-8A
Test Case Description	Add Employee – Positive test case	Test Priority	Medium
Pre-Requisite	A Successful Login	Post-Requisite	NA

Test Execution Steps:

S.NO	Action	Inputs	Expected Output	Actual Output	Test Application	Test Result	Test Comments
1	Launch Window	Select Add Employee Option from Window	Add Employee GUI	Add Employee GUI	VsCode	Pass	[Mickey /05/202109:35 AM]: Launch successful
2	Give Name ID, Address ,Post, Salary then Select Add button	Name: Williams ID:1002 Address: Biratnagar Post: Cashier	Employee Insertion Successful	Employee Insertion Successful	VsCode	Pass	[Mickey 04/05/2021 09:36 AM]: Launch successful

		Salary: 90000					

Test Scenario ID	Search Employee-9	**Test Case ID**	Search Employee -9A
Test Case Description	Search Employee – Positive test case	**Test Priority**	Medium
Pre-Requisite	A Successful Login and employee must be present	**Post-Requisite**	NA

Test Execution Steps:

S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Application**	**Test Result**	**Test Comments**
1	Launch Window	Select Search Employee Option from Window	Search Employee Dashboard	Search Employee Dashboard	VsCode	Pass	[Prason 04/05/2021 10:00 AM]: Launch successful
2	Select any of the search parameter like name and ID	Name: Williams ID: 1002	Employee found and full details is displayed	Employee found and full details is displayed	VsCode	Pass	[Prason 04/05/2021 10:05 AM]: Launch successful

Test Scenario ID	Search Customer-10	**Test Case ID**	Search Customer -10A
Test Case Description	Search Customer– Positive test case	**Test Priority**	Medium
Pre-Requisite	A Successful Login	**Post-Requisite**	NA

Test Execution Steps:

S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Application**	**Test Result**	**Test Comments**
1	Launch Window	Select Search Customer Option from Window	Search Customer Dashboard	Search Customer Dashboard	VsCode	Pass	[Prason 03/05/2021 10:30 AM]: Launch successful
2	Select any of the search parameter like name, ID or	Name: Nathan ID: 1000 Shopped Date:01/05/2021	Customer found and full details of purchase is displayed along with	Customer found and full details of purchase is displayed along with	VsCode	Pass	[Prason 03/05/2021 10:35 AM]: Launch successful

	Shopped Date		customer information	customer information			

Test Scenario ID	Search Customer-11	**Test Case ID**	Search Customer -11A
Test Case Description	Search Customer– Negative test case	**Test Priority**	Medium
Pre-Requisite	A Successful Login	**Post-Requisite**	NA

Test Execution Steps:

S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Application**	**Test Result**	**Test Comments**
1	Launch Window	Select Search Customer Option from Window	Search Customer Dashboard	Search Customer Dashboard	VsCode	Pass	[Prason 03/05/2021 10:30 AM]: Launch successful
2	Select any of the search parameter like name, ID or Shopped Date and enter invalid details	Name: Sam ID: 1000 Shopped Date:2/05/2021	No such Customer found	No such Customer found	VsCode	Pass	[Prason 03/05/2021 10:35 AM]: Launch successful

Test Scenario ID	Search item-12	**Test Case ID**	Search Item - 12A
Test Case Description	Search Item– Positive test case	**Test Priority**	Medium
Pre-Requisite	A Successful Login and items should be present in database	**Post-Requisite**	NA

Test Execution Steps:

S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Application**	**Test Result**	**Test Comments**
1	Launch Window	Select Search item Option from Window	Search item Dashboard	Search item Dashboard	VsCode	Pass	[Prason 03/05/2021 10:30 AM]: Launch successful

2	Select any of the search parameter like name, CP or SP	Name: Bat CP: 1000 SP:1500	Item found and full details of item is displayed	Item found and full details of item is displayed	VsCode	Pass	[Prason 03/05/2021 10:35 AM]: Launch successful

Test Scenario ID	Search item-12	**Test Case ID**	Search Item - 12A
Test Case Description	Search Item– Negative test case	**Test Priority**	Medium
Pre-Requisite	A Successful Login and items must present in database	**Post-Requisite**	NA

Test Execution Steps:

S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Application**	**Test Result**	**Test Comments**
1	Launch Window	Select Search item Option from Window	Search item Dashboard	Search item Dashboard	VsCode	Pass	[Prason 03/05/2021 10:30 AM]: Launch successful
2	Select any of the search parameter like name, CP or SP and give invalid details	Name: window CP: 20000 SP:25000	No such Item found	No such Item found	VsCode	Pass	[Prason 03/05/2021 10:35 AM]: Launch successful

9. Comparative Analysis with other existing technologies

All the management software developed today are mainly focused for big supermarket malls and other national and multinational agency. These types of software are very critical and complex to understand and require knowledge and skill to run it. Also these software's require high end specification, internet access, additional modules and database provider generally cost very high based on the type of business and extension of it. So, generally we often see that the organization is often kept manually or sometimes not kept in the smaller shops.

So, our software "Stocked" can run on any low end specification system with very basic knowledge to run. Also, this is a standalone software, that means it can be used anywhere where there is no internet access like the areas of extreme heights where generally local shop owners run their shop and manage their stock manually.

The key features of our software can help the local shop owners to efficiently run their shop by providing automated revenue report, trending items list, not to invest alerter, expired items check, profit and loss generator, creating bills with storing all the information related to the customer as well as the employee working for them.

This software also provides a specific search option for the owner to retrieve any information whether it is related to the items or about the customer and their purchases. Though being a standalone software with low specification, it has features like providing a graphical insight about stocks and trending items, and also managing of customers as well as employees apart from stock management makes this software a complete management tool for any shop operating at areas where is no internet access.

10. Conclusion

The expected outcome of this project was to develop a software which would be able to aid the functioning of simple local shops/small businesses by helping them to track the inventory, organize the products, determine the state of the business (Profit and Loss, Most sold items, etc.) and the generation of bill during the sale and we were successfully able to achieve the expected results and thus present our fully functioning "**Stocked – a complete organizing solution"**

Further Improvements and Updates to the software will be made along with time.

Future Work

We want to integrate the following features in the future:

- Discount on products will be generated according to their sale reports.(i.e. The least sold items will discounted accordingly to improve the sales)
- Coupons / Promo Code application
- Support to handheld devices (IOS, Android, etc.)

11. References

[1] R. Elmasri & S. B. Navathe, Fundamentals of Database Systems, Addison Wesley, 7 th Edition, 2015

[2] Thomas Connolly, Carolyn Begg, Database Systems: A ‖ Practical Approach to Design,

Implementation and Management‖, 6th Edition, 2012

[3] Rapid GUI Programming with Python and Qt:The Definitive Guide to PyQt Programming by Mark Summerfield

[4] MySQL for Python by Albert Lukaszewski

[5] Moskowitz,Robert,"Using Your Computer for Inventory Control", Accvision Retrieved August 17, 2010.

[6] Lockard, Robert (29 November 2010) "3 Advantages of Using Inventory Management Software". Inventory System Software Blog. Retrieved 23 November 2012.

[7]"Inventory Management Organism" by A Rohini, Wubshetasamnew Woldeyohannshiluf and

Teshometeklemicheal Walelignayimelo

[8] "Tracking Inventory" by Lesonsky, Rieva (1998) Entrepreneur Magazine

12. Appendix:

Link to ppt:
https://drive.google.com/drive/folders/1MDBJY4jDt95rxUhuX7XviHVS4KuLGtrc?usp=sharing

Link to video: https://youtu.be/YeN6E6Q4ygA (run at 1080p for better quality)

Link to source code:
https://drive.google.com/drive/folders/1MDBJY4jDt95rxUhuX7XviHVS4KuLGtrc?usp=sharing

NOTE: BOTH PPT AND SOURCE CODE ARE IN SAME DRIVE LINK.