TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

AN
OBJECT-ORIENTED ANALYSIS & DESIGN
PROJECT REPORT
ON

DEPARTMENTAL STORE MANAGEMENT SYSTEM

By:
Amit Bhandari, 070-BCT-505
Basanta Maharjan, 070-BCT-511
Brihat Ratna Bajracharya, 070-BCT-513
Kritish Pahi, 070-BCT-517
Manish Munikar, 070-BCT-520
Sushil Shakya, 070-BCT-547

A PROJECT WAS SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULLFILLMENT OF THE REQUIREMENT FOR THE BACHELOR'S DEGREE IN COMPUTER ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
LALITPUR, NEPAL

AUGUST, 2016

Acknowledgment

This report is prepared for the partial fulfillment of the course *Object Oriented Analysis and Design* of the 6th semester of Bachelors in Computer Engineering. Best efforts have been made to ensure that this report is accurate and professional as far as possible.

We are very grateful to our subject teacher **Dr. Arun Kumar Timalsina Sir**, for his excellent lectures and tutorials that helped us become familiar with the terminologies and concepts in object oriented approach in software development.

We would also like to thank the Department of Electronics and Computer Engineering (DoECE) of Central Campus Pulchowk, Institute of Engineering for providing us with this great opportunity to develop a project on the very interesting course of Object Oriented Analysis and Design. This project has greatly enhanced our knowledge of object oriented framework in software development.

Sincerely,
Amit Bhandari, *070-BCT-505*
Basanta Maharjan, *070-BCT-511*
Brihat Ratna Bajracharya, *070-BCT-513*
Kritish Pahi, *070-BCT-517*
Manish Munikar, *070-BCT-520*
Sushil Shakya, *070-BCT-547*

Abstract

The course of Object Oriented Analysis and Design (OOAD) is the continuation of the course Software Engineering with emphasis on fundamental concept of object orientation. This course mainly focuses on the software development process in object-oriented framework. It also provide exposure to Visual Object Oriented Modelling languages, specifically Unified Modeling Language (UML). This report is a formal documentation of our OOAD Project done using the theoretical as well as technical knowledge of software development procedure.

Table of Contents

List of Figures

1 Introduction

This project entitled ***Departmental Store Management System*** is a web based application that manages various activities done in department store. These activities include ordering goods/products from suppliers/producers, entering the new arrival of products into the system, managing those products in the department store and making bills while selling those products to customers. The system can store information of the products in a very structured way. Bills are created and details of each transaction can be viewed whenever required. The interface of the system will be user friendly and as simple as possible yet powerful requiring little training of operating staff.

Our system stores the detail of supplier, product, customer bills, department store staff as main entities. The department store manager regularly checks the product in stock and orders those that are out of stock. Goods and product come from many suppliers. Each supplier supply one or more products but one product is supplied by a unique supplier. These products are then classified into several categories and arranged categorically in separate racks. Details of every products (like in which category does it belong to, where it is located, available number, price) etc. are recorded and maintained in central database. An employee is assigned to enter the necessary information about the product into the database system of the store. A customer uses the system in order to find the product he is searching for and he gathers all the products he wishes to buy and brings it at the payment section of the store. An employee then enters the product codes of the customer purchase into the system and bill/invoice is prepared by the system. A customer makes the payment. Each transaction is stored in database for future references. So this is all about the workflow of our system.

As already mentioned above, this is a web based application system. A web-based application is any application that uses a website as the interface (the “front-end”). Users access the application from any computer connected to the Internet using a standard browser, instead of using an application that has been installed on their local computer. Almost any desktop software can be developed as a web-based application. With web-based applications, users access the system via a uniform environment—the web browser. While the user interaction with the application needs to be thoroughly tested on different web browsers, the application itself needs only be developed for a single operating system. There’s no need to develop and test it on all possible operating system versions and configurations. This makes development and troubleshooting much easier.

2 Objectives

We can categorize our objectives into two main categories. They are:

2.1 Academic Objectives

- To fulfill the partial requirement of our course.
- To explain and illustrate the fundamental concept of object orientation.
- To learn about Visual Object Oriented Modelling languages, specifically UML.
- To develop software/application following object oriented analysis and design.

2.2 Software Objectives

- To implement the concept of object orientation and build a software for store management system.
- To build an easy interface for our system.
- To use UML diagrams to make overview of the system and use it to implement in code.

3 Literature Review

Departmental Stores is *one spot shopping*. It is widely acceptable and fast gaining business also known as Universal Suppliers.

Unlike online virtual store, in departmental stores, the items required by every family are available under a single roof. Departmental stores specializes in satisfying a wide range of goods and products of the consumers and residential needs and at the same time offering them a choice of multiple services at variable price points. It avoids the need to visit different places to purchase different kinds of products from several places.

A typical Departmental Stores have following criteria for betterment of service to public:

- Individual department are established as per different types of goods to be sold.
- Each Department is an independent unit ie. the sections operates independently under single management.
- Departmental Stores provides maximum shopping convenience to its customers.
- It keeps huge stock of fresh goods which highlight latest fashions and trends followed by different manufacturers.

The main aim of every departmental store is to provide and fulfill all requirements of their customers at one place along with comforts and facilities, which a small scale retailer cannot provide.

Buying, Supervision, accounting, advertising and external communications are the basic operations of Departmental Stores handled directly by the central management of the store. Imagine handling all the Accounting, Supervision, Communications operated by the staffs, which is very monotonous for day to day operations of the stores.

To avoid these tedious tasks, a software system could be employed to take over these task of Accounting as Cashier, Supervise the stock as Supervisor, making the deal with all Suppliers. Cashier making bills manually for the goods purchased is a traditional approach which is outfit for modern departmental stores to handle large customers and services.

On the other hand to keep daily record of stock available for each products under all categories is almost impossible for any operators. Hence via this software system, it could keep the daily track of the goods (products) sold and the new arrival of goods from the suppliers so that there is surplus inventory to avoid being “out of stock” at the time of purchase.

4 System Development

4.1 UML Diagrams

Some of the UML diagrams that we made for this system are given below.

4.1.1 Use Case Diagram

A use case diagram of our system is given in Figure 4.1.

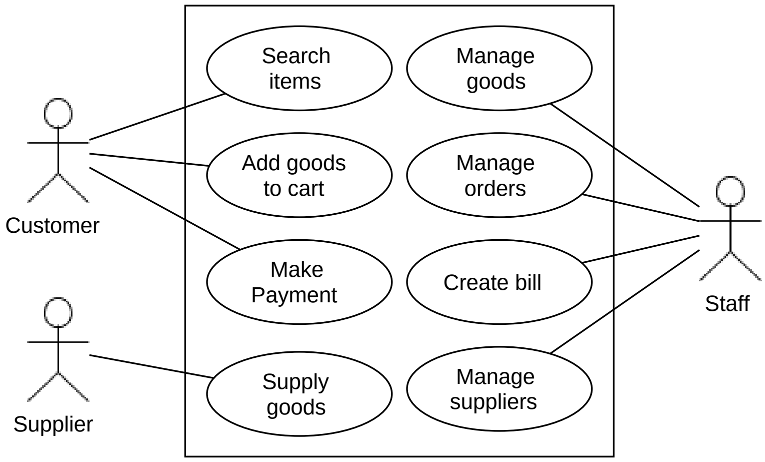


Figure 4.1: Use case diagram

4.1.2 Class Diagram

A Class Diagram in Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and relationship among objects. This diagram is considered as main building block of object-oriented modelling. The classes in a class diagram represent both the main element, interactions in the application, and the classes to be programmed. Each class comprises of three compartment, top compartment contains name of class, middle compartment consist of attributes of the class and bottom compartment contains methods that class can execute. Class diagram for DSMS is shown below. The class diagram consist of instance level relationships. Association represents a family of links and consist of role name, multiplicity, navigational direction, visibility and other properties. Ex. between product and supplier, the multiplicity is in a way that one supplier can provide one or more product. The direction is from supplier to product and role name is "supplies" indicating the statement

"Supplier provides one or more product". Similar is for other classes as well. Aggregation is "has a" association relationship. It represent that life of one class is dependent on other. Ex. Bill_detail is composite aggregate part (more strict) of Bill. Generlization is shown in between Staff and other managers in diagram below. what it signifies is that one of the two related class is considered to be specialized form of the other. Here Staff is more general and *_Manager is specialized class. Other details are self explanatory and sufficient.

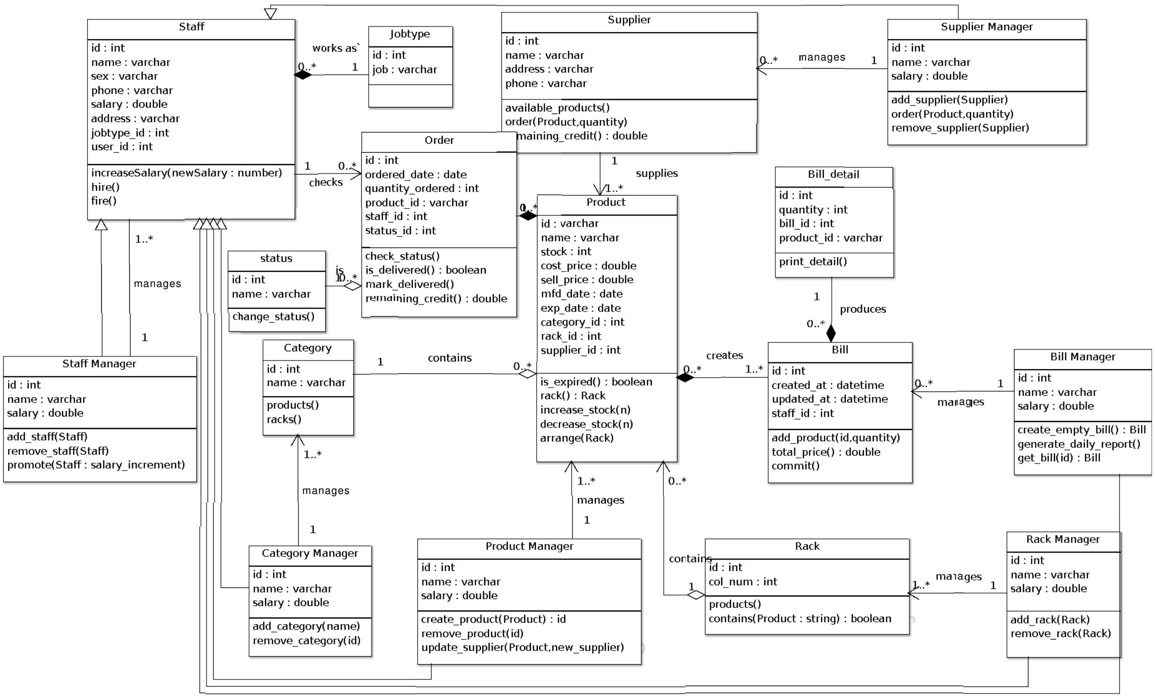


Figure 4.2: Class diagram

4.1.3 Sequence Diagram

This diagram describes sequence of steps that may commonly occur in the department store. We have five objects namely Department Store, Product, Supplier, Customer, and Bill. Department Store checks for product status for each product i.e. quantity (available stock), quality (exp date, condition during storage), cost and updates the current status. Accordingly, it orders goods/products from various suppliers and arrange them in store, Customers come, search for product and buy required product. They find cost for each product in label sticked to each product. A bill is made per transaction and product is handed to customers after bill clearance. The activities like searching product and buying product may happen several times (shown in diagram explicitly as a loop). After the transaction is complete, store updates product status and now has current status of the products.

Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It shows object interaction in time sequence. The diagram shown is arranged in a way so that horizontal axis represent object interaction (message exchange between objects) and vertical axis represent time axis (life line of object). Some objects (like Bill) is created after triggering particular event (here it is buy product) and destroyed after their job is completed. The DSMS sequence diagram consist of five objects namely Department Store, Product, Supplier, Customer, and Bill. Department Store checks for product status for each product i.e. quantity (available stock), quality (exp date, condition during storage), cost and updates the current status. Accordingly, it orders goods/products from various suppliers and arrange them in store, Customers come, search for product and buy required product. They find cost for each product in label sticked to each product. A bill is made per transaction and product is handed to customers after bill clearance. The activities like searching product and buying product may happen several times (shown in diagram explicitly as a loop). After the transaction is complete, store updates product status and now has current status of the products.

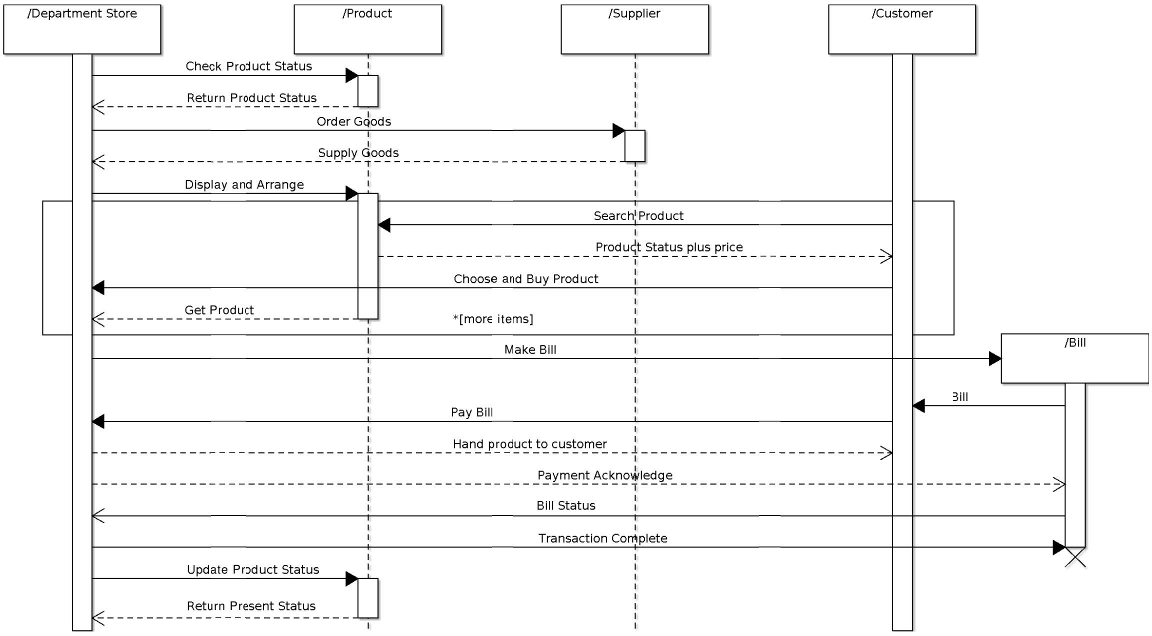


Figure 4.3: Sequence diagram

4.1.4 Activity Diagram

Activity Diagram describes the dynamic aspects of the system. It diagram shows user oriented view of system operation. We have made activity diagram using swimlanes. A swim lane is a visual element that distinguishes job sharing and responsibilities for sub-processes. In our system's activity diagram, we have three swinlanes and we have separated job/responsibilities accordingly. Each step is continuation of previous step. Decision is taken wherever necessary and fork and join is used to divide or attach work flow. The objective of making activity diagram is similar to objectives of other UML Diagrams. Only difference is that it is used to show message flow between activities. Below is the activity Diagram for DSMS.

4.1.5 Collaboration Diagram

Sequence diagram and collaboration diagram are very much similar to each other that deals with instances and not classes. Collaboration (also called communication) Diagram models the interaction between objects or parts in terms of sequenced messages. This diagram is the combination of information taken from Class, Sequence and Use Case Diagram describing both the static and dynamic aspect of the system. Sequence diagram and collaboration diagram are very much similar to each other that deals with instances and not classes and it is easier to visualize the system from this diagram. Collaboration diagram shows sequence of steps needed for operation of system in graph-like structure. Sequence of step is explicitly written before each operation association between instances. Collaboration Diagram for DSMS is shown below with five instances and sequence of message exchange is shown explicitly by a number before each message. Collaboration diagram shows interaction between element in sequence but irrespective of time.

4.1.6 State Chart Diagram

State Chart Diagram describes a state machine. It models dynamic nature of a system and defines different states of an object changed by events that occur as the system runs. In the statie chart diagram of DSMS, events (may be internal or external) is shown above each transition. Next state of the system is determined by transition event and previous state. For ex. in “Arrange Product” state, if event “display” is triggered, then it goes to “Display Product” event and if “sell” event is triggered, it goes to “Sell Product” state. The state chart diagram for DSMS is shown below.

These diagrams depict the way in which objects evolve during their life in the system. The elements of state chart diagram are:

States representing a stable condition of the object which persists for a significant time

Transitions depicting possible paths from one state to another.

Events these may be external events originating from actors, or internal events performed by other objects in the system.

Actions processing carried out as part of a transition in state

Conditions conditions governing the occurrence of a transition.

4.2 Data Flow Diagrams

Data flow diagram is used to show the flow of data from external entities into the system. It is used to represent the physical and logical area of an information system. The data flow diagrams are pictorial or graphical representation of the Departmental Store Management System is shown below. The data flow diagram covers all the processes and data storage area, which takes place during any transaction in the system. Level 0 DFD is created after identifying the logical subsystems that may exist in our system. The identified subsystems are Customer Section, Supplier Section, Account Section and Central Database. Only two external entities, namely Customer and Supplier take part in DSMS. Logical Data Flow between customer and system include product search, product info, product buy. Physical Data Flow includes Transaction receipt and purchased goods/products. Similarly, supplier takes order from system (logical data) and provide goods/product (physical data) to the system. Transaction estimate bill and payment vouchers can also be considered as physical data flow.

Level 1 DFD decomposes the processes in Level 0 DFD and identifies data stores. There are separate DFD for Customer Section and Supplier Section.

Customer Section is further divided into Counter (assist desk for customer), Account (handles financial transaction between customer and system) and database (data store for this section). Counter is only subsystem that interacts with customer directly as well as other subsystem of the section. Data Flow in this level is same as Level 0 data flow.

Similarly Supplier Section comprises of Administrator section, Account Section, and Database. Administrator is responsible to make orders as per product availability in stock and market rate. Supplier contacts this subsystem for all dataflow mentioned in Level 0 DFD. Account section handles financial tracsaction (payment—cash or check, old due, etc.) between supplier and system and database is maintained to store all relevant information.

4.3 Entity Relationship Diagrams

An Entity-Relationship (ER) model describes inter-related things of interest in a aspecific domain of knowledge. In software development, ER model ha become an abstract data model that defines a data/information structure that can be implemented in a database, typically a relational database. In our database, we have separate tables for product, product category, supplier, bill, staff details, order and other bunch of stuffs. Each table has several attributes that best describe the table. To get required information, say we need to print details of all transaction of particular day, then we need to access database and more than one tables to retrieve data. We need certain relationship between each tables in our database and a common attribute to map tuples of one table to another. ER Diagram provides visual reference to complete database at one glance. We can develop database looking at the ER Diagram and later use it as reference for further improvement. The ER Diagram for DSMS is shown below. Shown ER Diagram provides all tables, thier corresponding attributes, and relationship among them.

4.4 Project Schedule

The project schedule is shown in the Gantt chart in Figure 4.12.

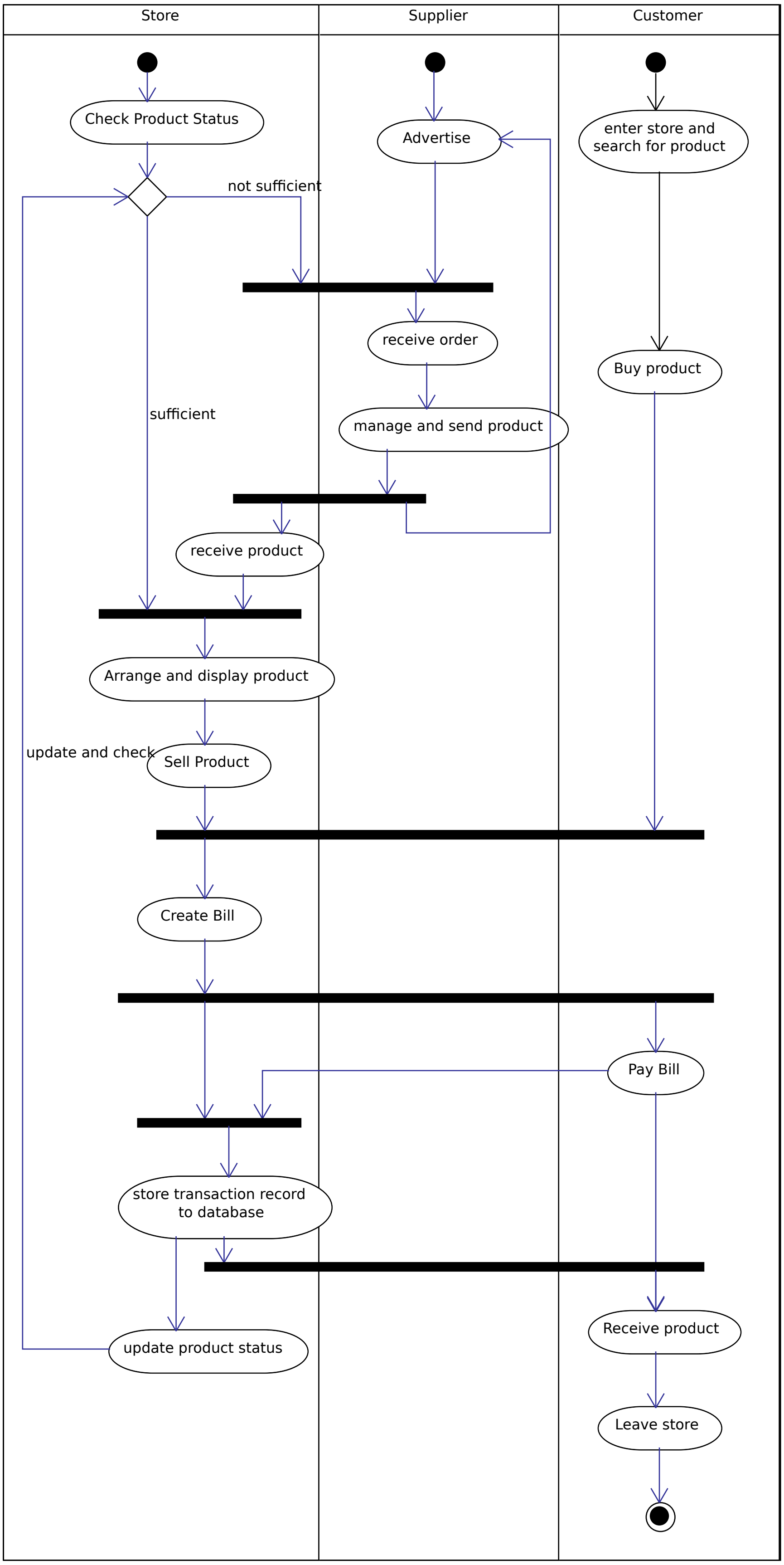


Figure 4.4: Activity diagram

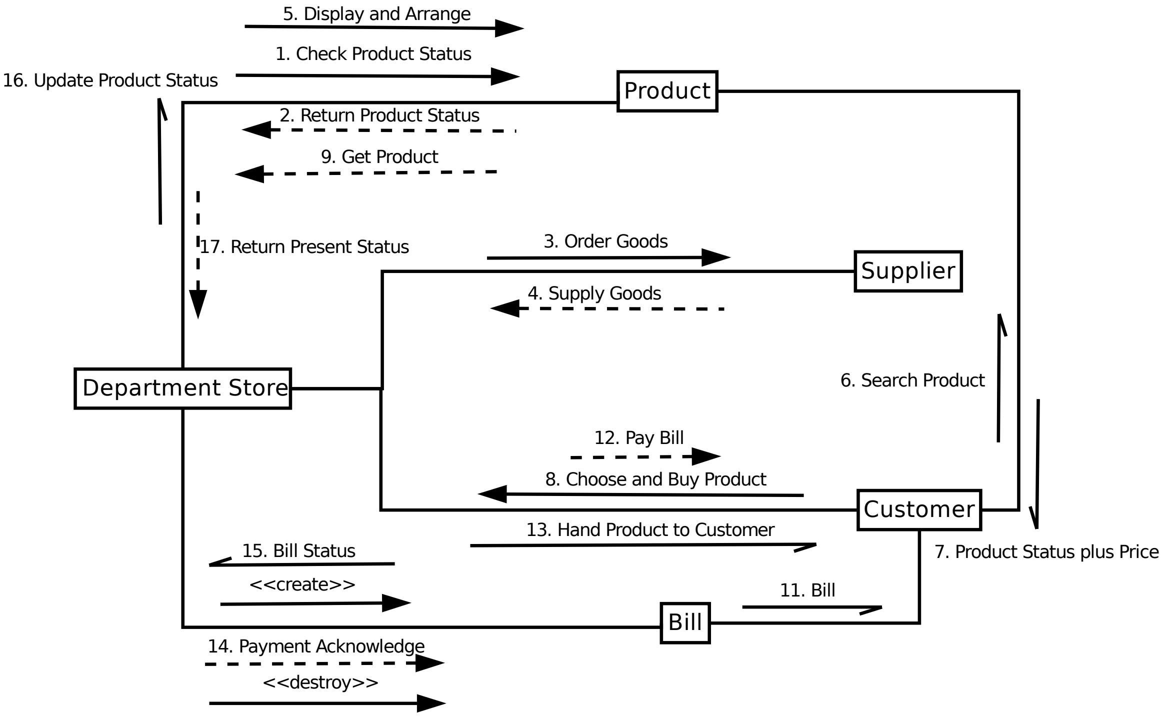


Figure 4.5: Collaboration diagram

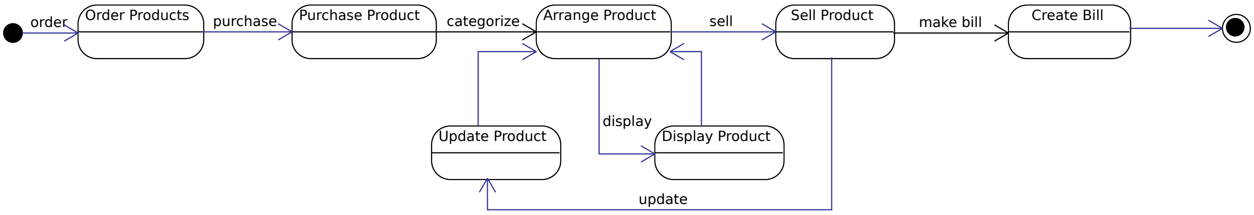


Figure 4.6: State chart diagram

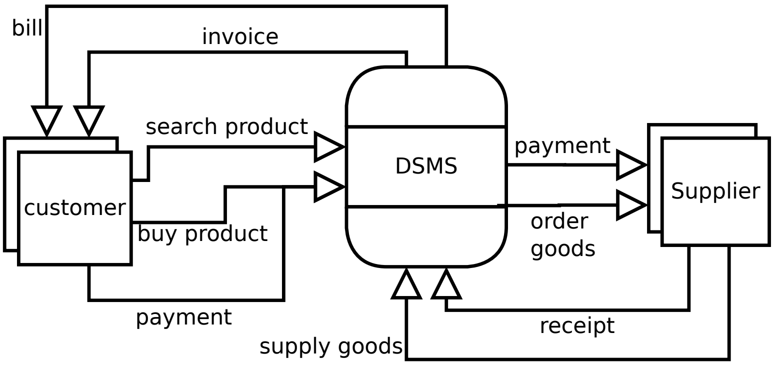


Figure 4.7: Level 0 (context) data flow diagram

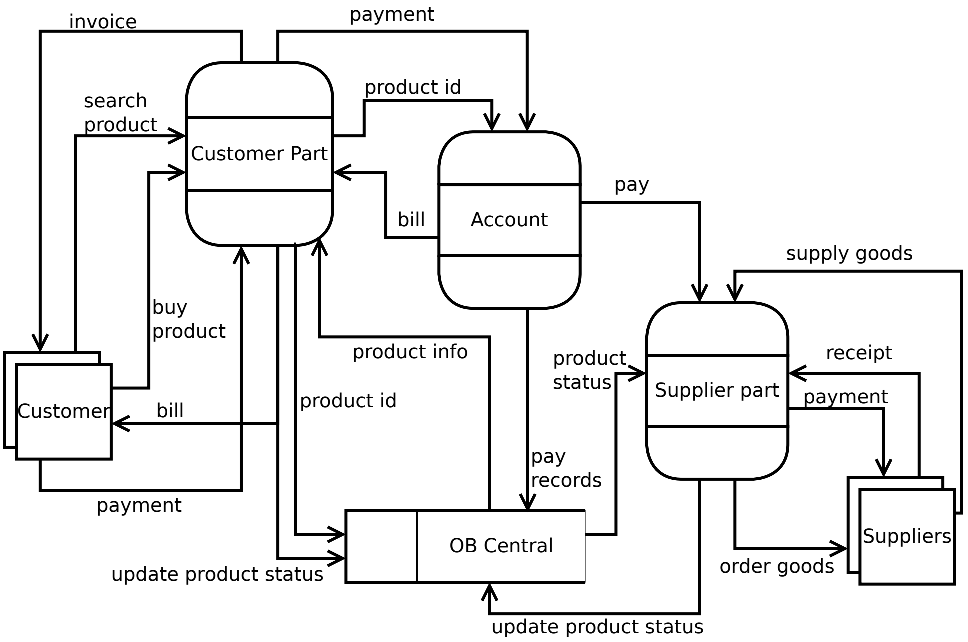


Figure 4.8: Level 1 data flow diagram

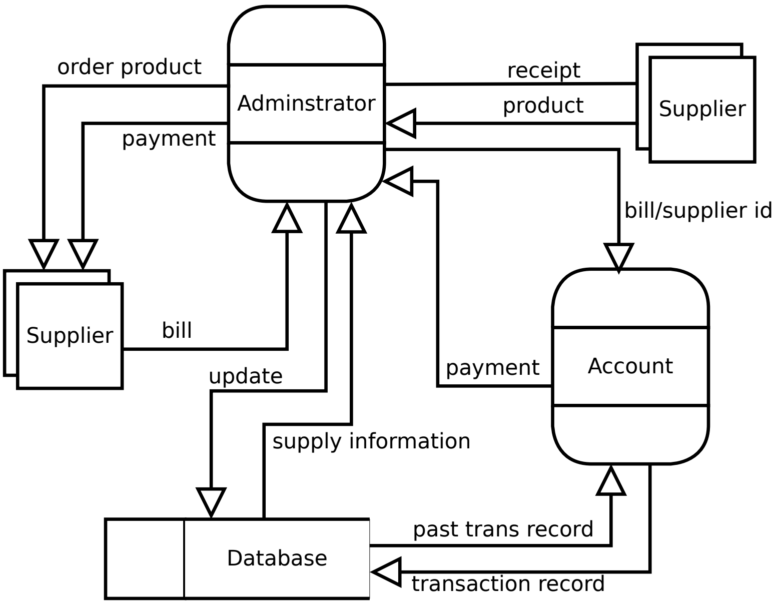


Figure 4.9: Level 2 data flow diagram

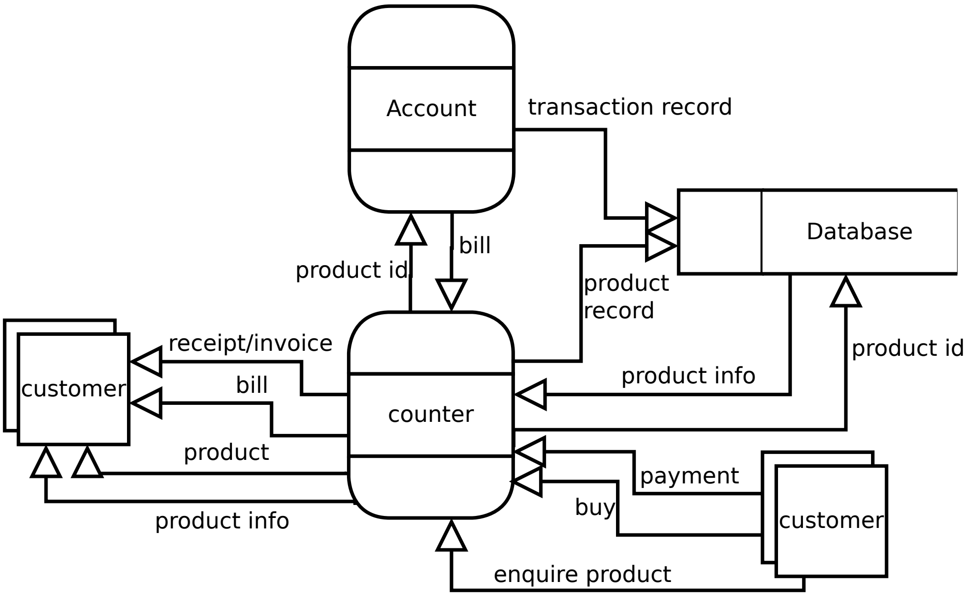


Figure 4.10: Level 2 data flow diagram (customer)

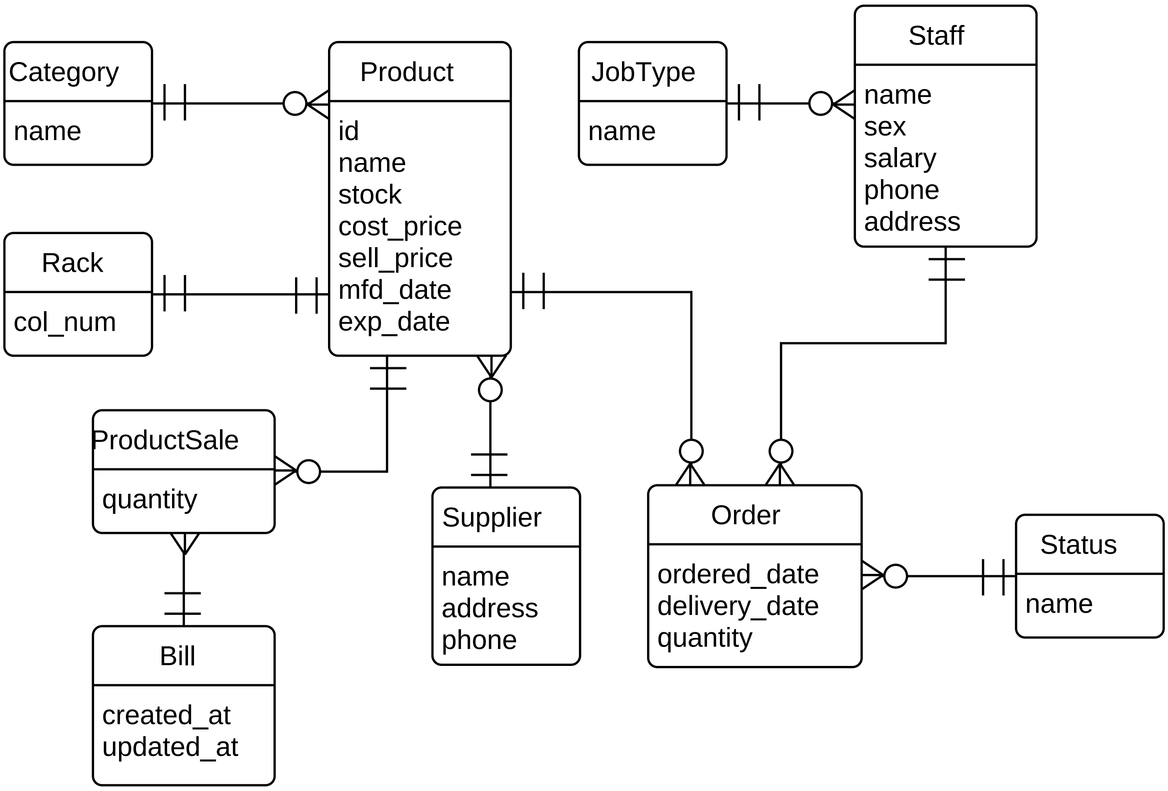


Figure 4.11: Entity relationship diagram

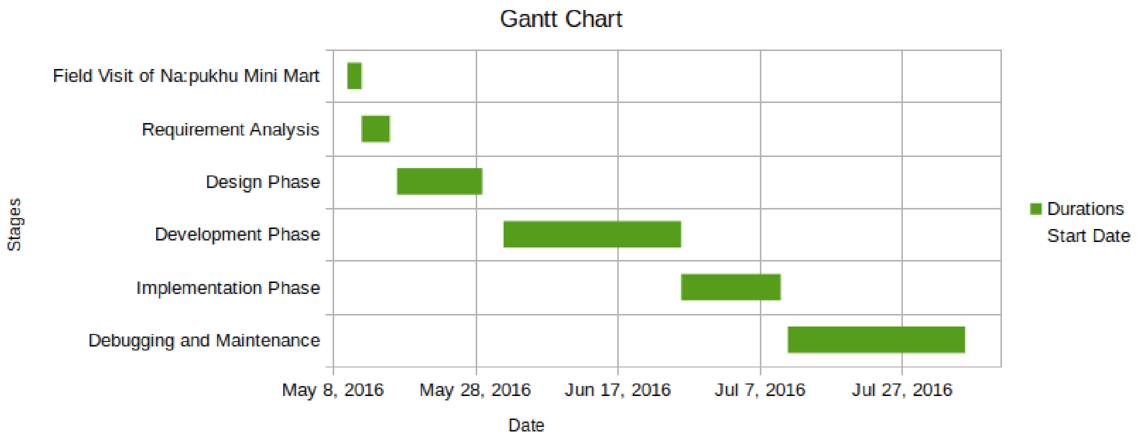


Figure 4.12: Project Gantt chart

5 Implementation Technology

5.1 Django

Django is a widely-used Python web application framework. It is a well-established open source project with tens of thousands of users and contributors spread across the planet. It is simply a collection of libraries written in the Python programming language. We have used Django version 1.10 in our current project implementation.

5.1.1 Why Django?

Django is a free and open source web application framework, written in Python.It helps us to develop websites faster and easier. It takes care of user authentication, content administration, sitemaps, RSS feeds, and many more tasks right out of the box. Its user authentication system provides a secure way to manage user accounts and passwords preventing from SQL injection cross-site scripting, cross-site request forgery and clickjacking. It is exceedingly scalable and incredibly versatile.

5.2 MySQL

MySQL is an open source relational database management system (RDBMS) based on Structured Query Language (SQL). It is the world's most popular open source database. It is developed, marketed, and supported by MySQL AB, which is a Swedish company.

5.2.1 Why MySQL?

MySQL is most often associated with web-based applications and online publishing. It supports large databases, up to 50 million rows or more in a table. It runs on many operating systems and support several development interfaces. It is easy to use, scalable, fast and secure.

5.3 Where and How Are They Used?

Since our project is mainly based on managing the products and their transaction using web based application, database is most important part of our project. Mysql database is used as the database engine of our system. The system consists of 11 major tables for different entities of our project, some of them are: Bill, Staff, Product, Bill detail, Order, Supplier, etc. Django is connected with the MySQL and models are created for each tables describing field details.

6 The Application

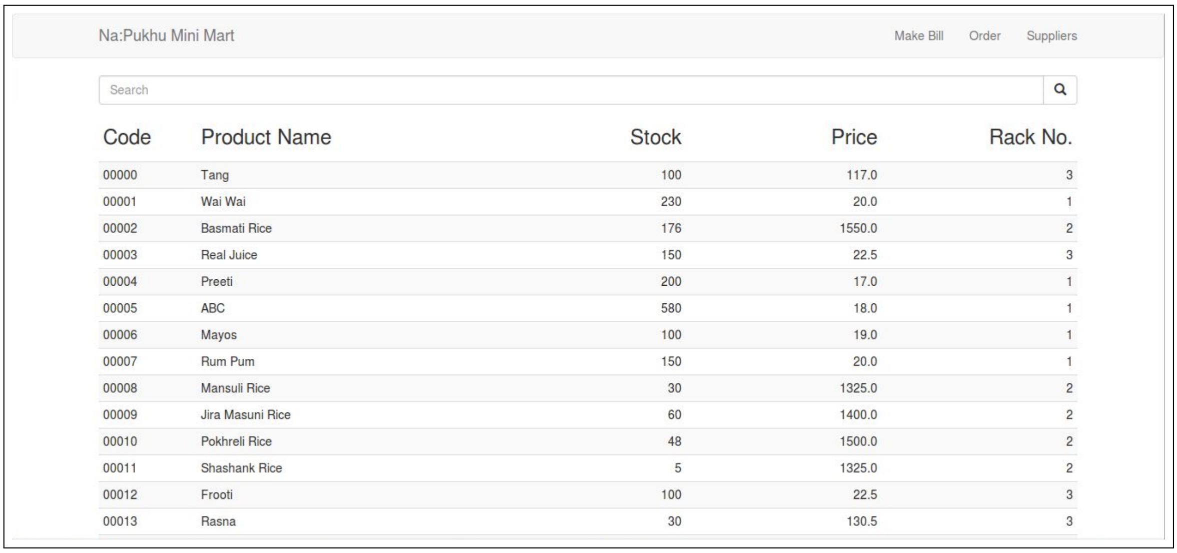
The application consists of five web pages:

1. Home page
2. Make bill page
3. Order page
4. Supplier page
5. View-bill page

Figures below show some screenshots of the application. All the functionalities are explained accordingly.

6.1 Home Page

This page displays the list of products that are available in the department store. It displays code, name, stock, price and the rack number of the product. There is a search bar where user can search for the product he is looking for. With this search bar he can find out in which rack he should look for in order to find the product.

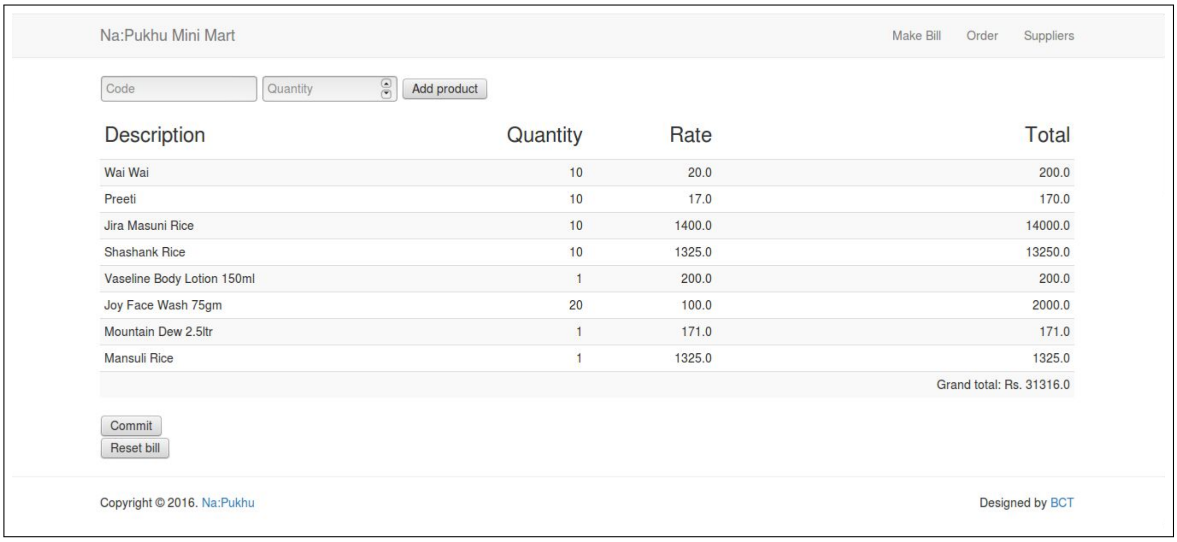


Code	Product Name	Stock	Price	Rack No.
00000	Tang	100	117.0	3
00001	Wai Wai	230	20.0	1
00002	Basmati Rice	176	1550.0	2
00003	Real Juice	150	22.5	3
00004	Preeti	200	17.0	1
00005	ABC	580	18.0	1
00006	Mayos	100	19.0	1
00007	Rum Pum	150	20.0	1
00008	Mansuli Rice	30	1325.0	2
00009	Jira Masuni Rice	60	1400.0	2
00010	Pokhreli Rice	48	1500.0	2
00011	Shashank Rice	5	1325.0	2
00012	Frooti	100	22.5	3
00013	Rasna	30	130.5	3

Figure 6.1: Home page

6.2 Make-Bill Page

This page is used in order to prepare the bill for the products bought by the customer. A staff simply enters the product code and quantity in the designated areas and the product gets added into the bill. Once all the products are added, the staff commits and the bill gets stored into the database. There is also a reset button available for resetting the whole bill.



Description	Quantity	Rate	Total
Wai Wai	10	20.0	200.0
Preeti	10	17.0	170.0
Jira Masuni Rice	10	1400.0	14000.0
Shashank Rice	10	1325.0	13250.0
Vaseline Body Lotion 150ml	1	200.0	200.0
Joy Face Wash 75gm	20	100.0	2000.0
Mountain Dew 2.5ltr	1	171.0	171.0
Mansuli Rice	1	1325.0	1325.0
			Grand total: Rs. 31316.0

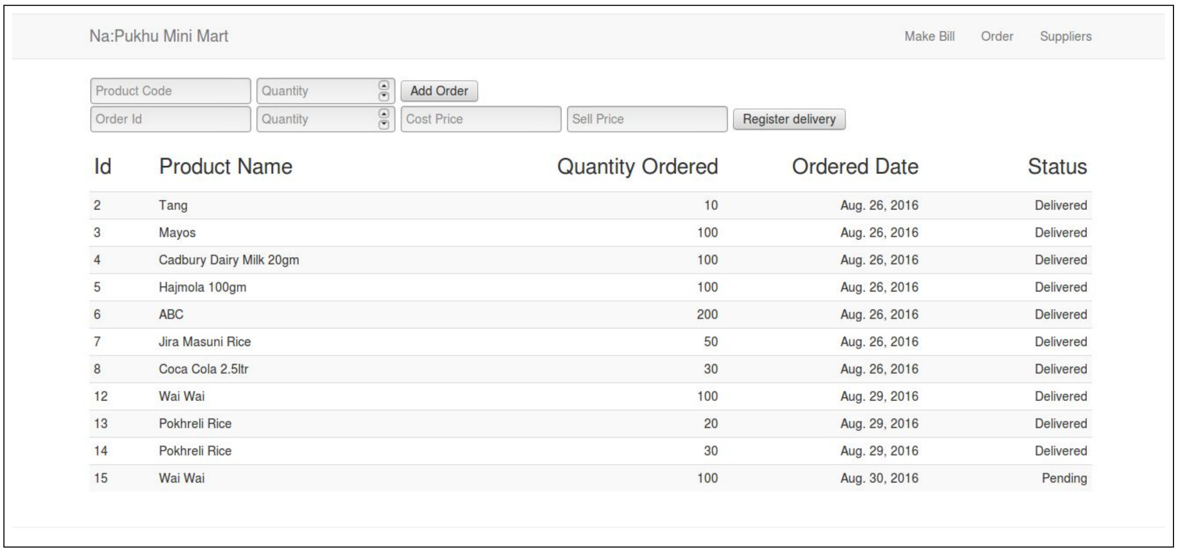
Figure 6.2: Make-bill page

6.3 Order Page

This page is used to record the orders given to the product supplier as well as the delivery made by the supplier. The page shows a list of recent orders. The order contains the id, product name, quantity ordered, ordered date and status. Once the order is recorded, it gets displayed in the list of orders and its status is “pending” by default. After the delivery is made by the supplier, the delivery detail is entered and thus the status of the pending order changes to “delivered”.

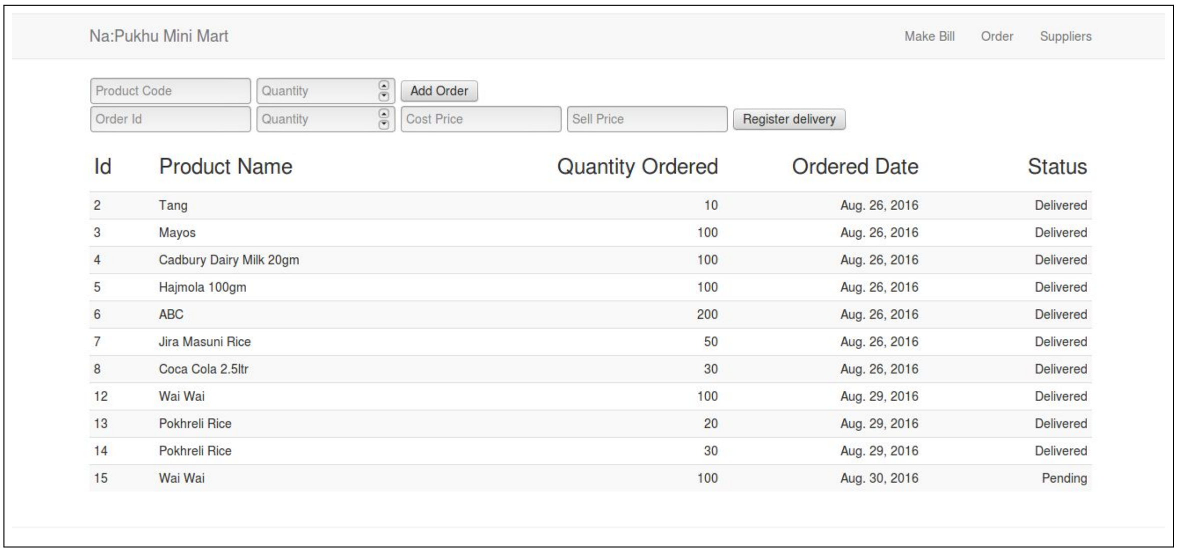
6.4 Supplier Page

This page displays the list of suppliers that are associated with the department store. It displays the name, address, contact number of the supplier.



Id	Product Name	Quantity Ordered	Ordered Date	Status
2	Tang	10	Aug. 26, 2016	Delivered
3	Mayos	100	Aug. 26, 2016	Delivered
4	Cadbury Dairy Milk 20gm	100	Aug. 26, 2016	Delivered
5	Hajmola 100gm	100	Aug. 26, 2016	Delivered
6	ABC	200	Aug. 26, 2016	Delivered
7	Jira Masuni Rice	50	Aug. 26, 2016	Delivered
8	Coca Cola 2.5ltr	30	Aug. 26, 2016	Delivered
12	Wai Wai	100	Aug. 29, 2016	Delivered
13	Pokhreli Rice	20	Aug. 29, 2016	Delivered
14	Pokhreli Rice	30	Aug. 29, 2016	Delivered
15	Wai Wai	100	Aug. 30, 2016	Pending

Figure 6.3: Order page



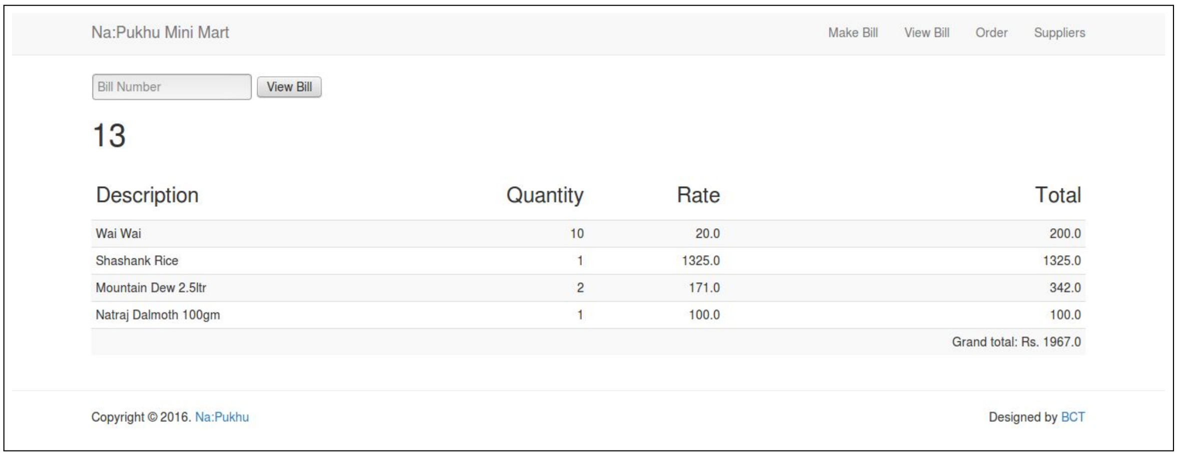
Id	Product Name	Quantity Ordered	Ordered Date	Status
2	Tang	10	Aug. 26, 2016	Delivered
3	Mayos	100	Aug. 26, 2016	Delivered
4	Cadbury Dairy Milk 20gm	100	Aug. 26, 2016	Delivered
5	Hajmola 100gm	100	Aug. 26, 2016	Delivered
6	ABC	200	Aug. 26, 2016	Delivered
7	Jira Masuni Rice	50	Aug. 26, 2016	Delivered
8	Coca Cola 2.5ltr	30	Aug. 26, 2016	Delivered
12	Wai Wai	100	Aug. 29, 2016	Delivered
13	Pokhreli Rice	20	Aug. 29, 2016	Delivered
14	Pokhreli Rice	30	Aug. 29, 2016	Delivered
15	Wai Wai	100	Aug. 30, 2016	Pending

Figure 6.4: Order page

6.5 View-Bill Page

This page is used to check out the past bills. The staff enters the bill number and the system outputs the bill details. The detail consists of each and every product bought, its rate, quantity, total and the grand total.

We actually visited *Na: Pukhu Mini Mart* (located near Bhaktapur Mini Bus Park, Itachhen) to know about managerial operations of the store and required attributes for our database. This visit was helpful in understanding practical approaches to manage a store.



13

Description	Quantity	Rate	Total
Wai Wai	10	20.0	200.0
Shashank Rice	1	1325.0	1325.0
Mountain Dew 2.5ltr	2	171.0	342.0
Natraj Dalmoth 100gm	1	100.0	100.0
			Grand total: Rs. 1967.0

Copyright © 2016. Na:Pukhu

Designed by BCT

Figure 6.5: View-bill page

Later section of this report contains Use Case and Class Diagram of the project that gives overall overview of the system. . . .

7 Limitations and Future Enhancements

Some limitations of our current applications are:

- The system is not designed to be operated by multiple cashiers at the same time.
- No automated system to notify when the stock of product is less than the threshold.
- Once the bill is made and committed, the system does not validate the confirmation of bill payment.
- The system cannot keep the track of Cash Inflow and Outflow of store yet.
- Bill printing feature isn't available yet.

Future enhancements:

- Staff login system will be added.
- A robust transaction handler system will be implemented.
- User Interface will be improved.
- Bill print feature will be added.

8 Conclusion

For this project, first we made an object oriented analysis, we prepared a use case diagram to better understand the system that we are trying to build . After that in the object oriented design part we made various UML Diagrams. Programming was done in Django (Python) using MySQL for database. Project was completed in time and we were able to fulfill mentioned objectives. Our system is solely browser based and it doesn't need any other application software to operate. Moreover, the system can run on any operating system without any hassle. In this technological era, we have built an application that helps to manage operations and activities of department store in digital form.

References

[1] Swapna Kodali. The design and implementation of an e-commerce site for online book sales. Master's thesis, Indiana University South Bend, 2007.

[2] Django: The web framework for perfectionists with deadlines. `https://www.djangoproject.com`.

[3] Programs & Notes for MCA. Uml diagrams for retail store management. `http://www.programsformca.com/2012/03/uml-retail-store-mgmt-diagrams-case.html`, 2016.