

Islington College



*i*slington college

FC6PO1NI Mobile Store Management System Final Documentation

Submitted By:

Student Name: Bibek Shrestha

Student Id Number: 16033491

Group: L3C3

Date: _2018/05/09

Submitted To:

Mr. Dhruba Sen.

Mr. Prakash Shrestha.

Lecturer, IT Faculty.

Word Count: 12094

Semester: Spring/Autumn

Contents

Plagiarism Declaration:.....	5
Abstract	6
Chapter 1: Introduction:.....	7
1.1. Introduction:.....	7
1.1.1. Web Based Application:.....	7
1.1.2. Management Systems:	7
1.1.3. Current System:	8
1.1.4. Problem Domain:	8
1.1.5. Project as a solution:	8
1.1.6. Project Aims:.....	8
1.1.7. Project Objectives:	9
1.1.8. Project Deliverables:	9
1.1.9. Motivational Factor:.....	10
Summary of this chapter:	10
Chapter 2: Background:	11
2.1. Introduction.....	11
2.2. Research Works:	11
2.3. Client:.....	11
2.4. System Architecture:	12
2.5. System Components:.....	13
2.6. History of Management system:	14
2.7. Similar Systems:.....	15
2.7.1. Mobile Store Management System (By Park D. Patel):	15
2.7.2. Medical clinic Management System (Raval Dhrub and Suthar Hardik):	16
2.7.3. Online Music Store Management System:	17
2.7.4. Comparative Analysis:	18
2.8. Proposed Tools and Techniques:	19
2.8.1. Programs which would be used:	19
2.8.2. Personal Techniques:.....	19
Chapter 3: Development:	20
3.1. Methodology Considerations:.....	20
3.1.1. Throw away Prototyping:	20

3.1.2. Evolutionary Prototype:	22
3.1.3. Waterfall Methodology	24
3.1.4. Selected Methodology:	26
3.1.5. Reasons for selecting this Methodology:	27
3.2. Methodology stage 1: Planning:	28
3.3. Methodology stage 2: Analysis:	28
3.3.1. Software Requirements Specification:.....	28
3.3.1.1.	28
3.3.2. Project Features:.....	29
3.4. Methodology stage 3: Logical Development:	33
3.4.1. Introduction:.....	33
3.4.2. Use Case Diagram of the system:	34
3.4.3. Class Diagram:.....	35
3.4.4. Classes:	35
3.4.5. Activity diagrams of the System:.....	49
3.5. Methodology Stage 4: Physical Development:.....	60
3.5.1. Physical Development:	60
3.5.2. Interface Design:	60
3.5.3. Screen Outputs of interface and Functionality:	61
3.5.4. Data Design:.....	74
3.5.5. The coding part:	80
Chapter 4: System Testing & Implementation:	85
4.1. System Testing:	85
4.1.1. Testing Types:	85
4.2. Test Cases:.....	86
4.2.1. Output Screen of Test cases:	92
4.3. Implementation Plan:	103
4.4. Development Tool Adopted:	104
4.5. Database Adopted:	104
4.6. Hardware Specification:.....	104
4.7. Software Specifications:	104
4.8. Installation of the System:	104
4.9. Training:	105

4.10.System Changeover:	105
Chapter 5: Conclusion and Review:	107
5.1. Conclusion Outline	107
5.2. Project Achievements:	107
5.3. Future Enhancement:	108
5.4. Evaluation Introduction:	108
5.5. Evaluation of the project Deliverables:.....	108
5.6. System Evaluation:	108
References:.....	109
Appendix:.....	111

Plagiarism Declaration:

I confirm that the enclosed written work (including application code) is entirely in my own words except where explicitly stated otherwise. I further declare that wherever I used copying, paraphrasing, summarizations or other appropriate mechanism related to another author's work it has been properly acknowledged in accordance with normal scholarly conventions. I further understand that wherever 6 or more consecutive words are extracted from a source they must be acknowledged and 10 consecutive words that are unacknowledged will be regarded as proof of plagiarism.

Abstract

This report is done for the completion of the Final Report which was assigned by London Metropolitan University. The major aim of this report is to summarize all the steps, requirements and other necessary details about the system.

The initial part of this report is all about the definition and introduction of the project. The introduction part includes the information regarding management system and use of different technology, its problem domain, project as a solution and also its aims and objectives. The approaches undertaken, project deliverables and project timeline is also included in detail. The second part of the report is about the literature review of the project with its subtopics which includes the research data and information and comparison of the project to other similar systems with chosen methodology and its pros and cons. The third part of the report is about the requirement analysis which includes the different structures, facts and figures and design of the system. The further part of the report will provide information about the system's testing installation, requirements evaluation and finally the conclusion.

Chapter 1: Introduction:

1.1. Introduction:

1.1.1. Web Based Application:

A web-based application is a program which is accessed directly on a network connection using internet protocols like HTTP, rather than a program existing in the memory of a device. Web-based applications usually operate in a web browser. In some cases web application may also be client based in which a slight part of the program is downloaded on the desktop of the user while the processing is performed using the internet on external server.

Web-based applications are also considered as web apps.

1.1.2. Management Systems:

A web based management system is known as the collection of programs, information and processes which can be used to manage the various phases of a project that is accessible on the internet. The essential process of project management like scheduling, calculation critical paths, resource management, controlling documents, building timelines etc. can be automated and controlled the project management solutions. There are mainly two types of inventory management systems. They are as follows:

➤ Perpetual Inventory Management System:

A permanent inventory account for tracking the inventory purchases and uses is kept by perpetual inventory management system. The system process is very simplified because when the system purchases different items during the business operation, the purchase record increase the balance sheet of inventories. Likewise the sales process in the system decreases the balance of inventories. A continuous record of inventories is maintained into the perpetual inventory management system.

➤ Periodic Inventory Management System:

A temporary purchase amount and inventory is used on a periodic basis on the periodic inventory management system. During the purchasing process, the purchased information is saved on a temporary purchase account which doesn't instantly affect the inventory account. At the time of sales, the inventory and reduction to the inventory account is not tracked. At the time of closing business cycle, the account of purchase is closed and the balance of purchase is added to the initial inventory. A physical inventory count is required in order to calculate the ending inventory.

Thus in the context of the current system that created. The current system falls under the Perpetual Inventory management system where the inventory is affected directly during the business cycle, with the title Mobile Store Management System.

1.1.3. Current System:

A survey was undertaken to find out the existing inventory systems used in the stores by the different merchants. Each of these stores is using different software based on their requirements. In the context of Nepal, although there are several benefits, many small businesses don't track their inventory using software. Some of the small businesses don't track their inventory manually at all but some merchants keep the track of their inventories in paper based systems. According to the survey, most of the merchants believe that digital inventory management system is better than the paper based management systems but 66.7% of people using the current management system are not much satisfied. 75% of the people believe that a digital management system should be implemented in every store. Hence, the new system is proposed in order to overcome all the problems of the current system and make the trend of using an inventory management system by every store.

1.1.4. Problem Domain:

In the context of Nepal, there are some desktop based store management systems available in the market. But still some owners of the stores prefer using ledger based system for billing reports of inventories. Paper based systems are highly vulnerable to various environmental factors. Due to repetitive access of data, there is very high chance of theft or loss of data. The available management systems are not very appropriate and possess some major limitations or faults. Due to these limitations the end users are unable to rely on the system. Hence they prefer to use paper based systems. A good store management system necessary in the current context that overcome all the issues of the current system and replace paper based system with digital system.

1.1.5. Project as a solution:

Nowadays, the changing technology should be quickly adopted by the marketers in order to reduce the overhead cost, cost of operation and stay competitive in the market. Through the use of this system, a user can transfer the information regarding the sales, purchases, customer information, supplier and company data from paper based to digital data. This will also help to access data and information regarding other stores so that marketers can be aware of each other's inventories. This information may be very much useful while ordering new purchases and to avoid overstocking of products. The proposed new system will overcome all the limitations of the current system and make the system more effective and reliable.

1.1.6. Project Aims:

The ultimate aim of this project is to develop a web based mobile store management system that will help the user of the system to manage the resources of the store. The application will also help the user to keep records of the customers and suppliers as well.

1.1.7. Project Objectives:**General Objectives:**

1. Learning ASP.Net through various sites.
2. Getting up with different web application development trends and designs.
3. Prepare plans to make Gantt Charts with milestones for each task.
4. Gathering ideas and being familiar with necessary aspects of web application development.

Specific Objectives:

1. To keep proper records of the inventory of the store.
2. To store necessary details of the suppliers.
3. To store necessary details of the customers
4. To maintain the record of imported items from the supplier.
5. To store identical details of the customers and their respective mobile phones.
6. To produce a proper dataset of the inventories those are currently present in the shop.
7. To maintain the record of sales and amount of the store.
8. To make calculation and inventory management easy and fast.
9. To calculate number of purchases and sales in a daily basis.
10. To keep record of cash or credit sales and purchases.
11. To improve task efficiency.
12. To reduce the cost of operation.
13. To increase in retention of customers.
14. To increase in number of sales and level of profit.
15. To increase in number of satisfied customers.
16. To make the store accessible to many users.

1.1.8. Project Deliverables:

- A Standard documentation of the system including Project Proposal, Requirement Specification, Test reports etc.
- Fully working model of the project
- Data entry Modules
- Data transportation Modules
- User Manual and Training Manual.

1.1.9. Motivational Factor:

The main motivation to do this project comes from the problems that we face in our family business namely “Newa Mobile Store”. Keeping records of different customers and their mobile phones along with its model and problems details was a major problem. When many customers provide their phones for repair service, it was so messy to identify their respective phones since different persons may have the same mobile phone. It was also difficult to determine the number of socks, imported items and sales records. Hence these problems motivated me to develop a system what would keep records of every imported inventory, current stocks; sales details and repairs records. The motivation to do this project comes from my eagerness to learn the web app and site design and get a deep understanding of different management systems.

Summary of this chapter:

This chapter provided brief introduction of the topic, how the idea of the project was initiated, the motivation factors behind the project, current scenario, the problem domain, approach to be adopted, some basic aims and objectives of the proposed system.

Chapter 2: Background:

2.1. Introduction

Inventory is the amount of goods, products or services that a store offers to its customers. These goods are itemized and recorded in the store management system. In a mobile store, there are many mobile phones along with its related accessories. Keeping the record of each and every item is a very challenging task for a store keeper. The management process of the mobile store starts with keeping the records of inventories available in the store. Storing details about purchases, sales, purchase returns and sales returns is another aspect of management process. Forecasting, demand planning and inventory controlling are some of the aspects of store management that I am considering for this project.

2.2. Research Works:

In today's word, the Internet and technology has become an important part of human life. Life without technology is very hard to imagine. The world without internet is incomplete. The idea of innovation in business organizations means doing things differently on a better and smarter way to create a positive difference. The key to success in digital life is considered as Innovation. There can be a lot of positive changes in terms of quantity, productivity, quality, reachability and so on through the use of technology in business measures. Use of technology has dramatically changed the life of individuals and organizations. The Online shopping, digital marketing, digital communication social networking etc. are the examples of change that came through the usage of technology for business purposes.

Proper management of resource plays a vital role in the success of business. In case of Large and medium business organizations, the role of resource management by the top level management is very hard. The resource of a business may include human, physical, tangible, financial resource etc. The problem of resource management has been solved by the usage of information technology by introducing user friendly solutions. Nowadays many large and multinational companies like Microsoft, Apple, Amazon, and ALIBABA etc. user cloud based solutions in order to manage their virtual or physical resources.

One of the most important components of the Business environment nowadays is web based applications. Through the use of these systems, the resource management and business transactions are simplified and are more likely to achieve their objectives.

2.3. Client:

“Suraj Mobile Zone” is the client that I have chosen for the application of this project. The store was established in 2070 BS. The proprietor of the store is Mr.Suraj Shrestha. The store has been providing people with different type of mobile phones and related accessories. The store imports its inventories from Pokhara city. The store provides all kinds of repair facilities including unlocking and software related problems. Accessories like charger, earphones, tempered glass, speakers, selfie stick etc are available in the store.

2.4. System Architecture:**Front End**

- Html 5, Css, Javascript
- Bootstrap
- .Net Components

Back End

- My Sql
- IIS Server

2.5.System Components:

- **Users:**
Users include Admin and End users of the system.
- **Customers**
They are customers of the business who purchase different products from the store.
- **Supplier:**
Suppliers are the one to supply various goods to the store.
- **Products**
It represents the inventories of the store.
- **Repair**
It represents records of all the phones to be repaired.
- **Purchase**
It represents all the purchased goods to the system.
- **Sales**
It represents the selling of items to the customers
- **Returns**
It represents the products that are returned to the store or that are returned to the supplier.
- **Receipt**
It is the receipt to the store by customer for credit purchases.
- **Payment**
It is the payment done to supplier by the store.

2.6.History of Management system:

A modern digital Mobile Store Management System must have the ability to keep the record of sales and inventories and also provide some communication means to contact the suppliers as per the need. In the early days the marketers or merchants used to write the details of the inventory and sales on a paper. They had to spend a lot of time to estimate their needs of future and retrieve their sales and inventory information. They had to spend a lot of their time for this process every day. As time passed by, different factors like efficiency and accuracy became the major factors of business along with the positive care of customers for increasing the sales. In the early 1930s, at Harvard University a team designed the first check-out system. This system would collect data and information for different punch cards which was associated with details of the items. At that time this system was rather too expensive for merchants. This was the very first time when a store management system was digitally designed into a computerized system. In spite of the system being too expensive, it created a new dimension to store management system. Further different ideas of store management were introduced such as Bar-code scanner, RFID-based scanning system etc. In the late 1940s, researchers created a better forerunner of the barcoding system. These barcode scanners used ultraviolet ink and arc reader in order to detect different items at the time of sale as it required significant contribution of computer which was also too expensive. Furthermore different affordable technology like laser technology was developed in the 1960. Laser technology used smaller, faster and cheaper readers. IN 1970s the modern barcode was developed these systems were cheaper and were affordable. They could manage inventory systems having many significant improvements or updates. During mid-late 1990s, retail shopkeepers began to actually implement the inventory management systems as the advancement in the computer technology. A good inventory management system insures to meet the requirements of the customers so that merchants or marketers can get their profit.

2.7.Similar Systems:

2.7.1. Mobile Store Management System (By Park D. Patel):



Figure 1: Mobile store management system demonstration.

The Mobile Store Management System is developed for desktop systems to facilitate mobile shop owners' management of customer details and inventory data, which will include mobile phones and accessories. It can be used efficiently for physically separated shops in different locations. This software will provide a simple and easy-to-operate user interface, which can be managed by any user without having prior in-depth knowledge of the computer system. One can use this software to get a sales report. Administrators can pull data from any location from the server. This application allows stores to manage customer details, keep inventory of all products and purchase information, in a very simple way, using a state-of-the-art software application. It will automatically generate invoices and update inventory. Keeping records of multiple stores is the key feature that has inspired me from this system. The system is a complete management package but yet is a desktop-based system which can be considered as its major limitation. The store doesn't itemize any related accessories of the mobile or the IMEI number of mobile phones for unique identification. It also doesn't include sales and purchase returns records.

2.7.2. Medical clinic Management System (Raval Dhrub and Suthar Hardik):

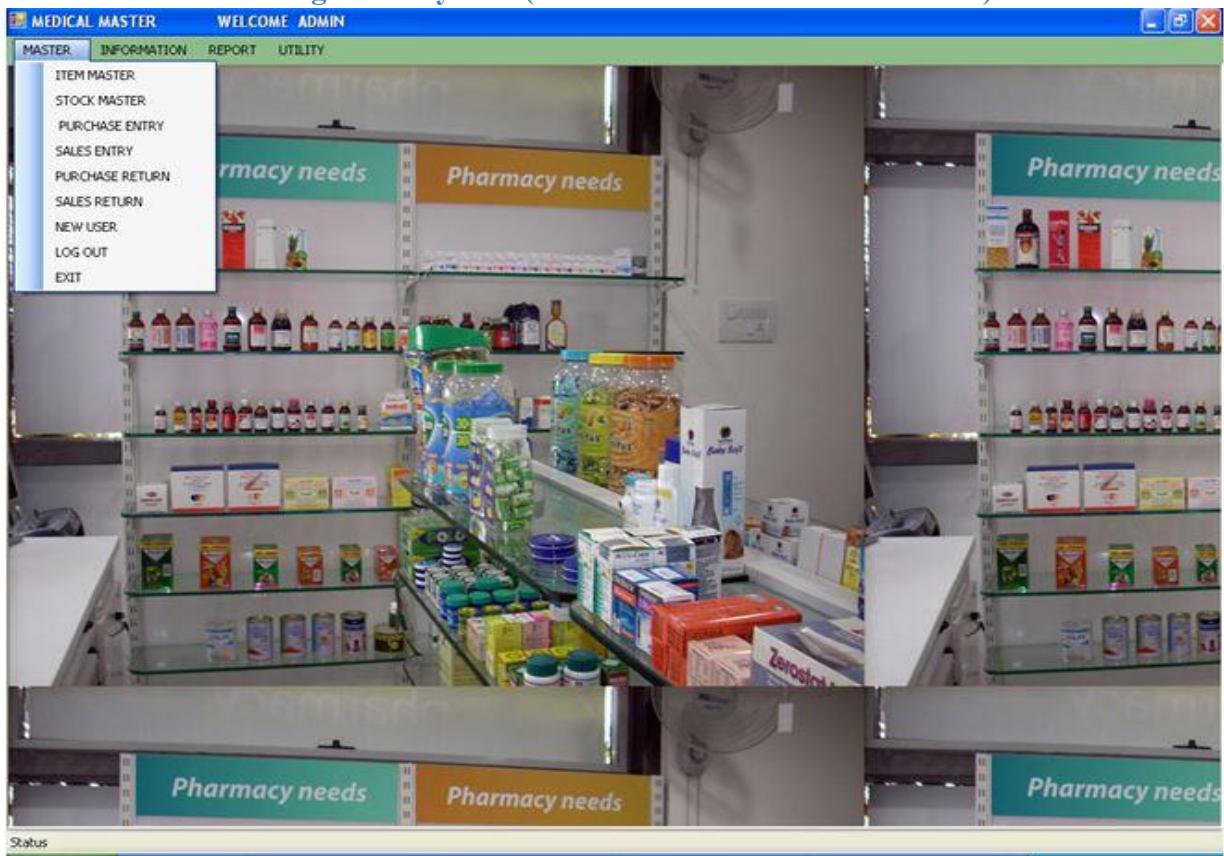


Figure 2: Medical clinic management system demonstration

The Medical clinic Management system was built for desktop systems using VB.Net and SQL server in 2008. This system was developed by Suthar Hardik J and Raval Dhrub Y. This Medical store management system was developed for the Hari-Om medical store. The major aim of this project was to build a modern computer based medical clinic's inventory management system. This system focused on different medical related items and its Sales. The system is simple and was very user-friendly. This system stored the data and related information of different medical items presented on the shop, the customers of the store, and the suppliers or dealers of the medical store. The system is a complete package for medical clinic management. The system stores data relating to sales, purchases, purchase and sales returns. The idea of recording purchase and sales return information regarding different medicines are some features of this system that I liked the most. The interface of the system can also be more attractive.

2.7.3. Online Music Store Management System:

Figure 3: Music store management system demonstration.

The Music Store Management System was developed by Reshma D. Sawant submitted to Kansas State University in 2008. The system is developed as a web based application which facilitate browsing, searching, and providing recommendation on different music items as the choice of customers. The system was build using ASP.Net 2.0 and SQL server. This system provided an online product catalog along with a custom shopping cart. The system stored different kinds of customer details like credit card numbers, usernames, passwords and other necessary details. The system also made some music recommendations to the customers by storing some preference based data of the customers. Music recommendation in this system is one of the major features that I liked the most about this project. The system is a web based application and is a very good system for music store management. The user interface of the system can be made more attractive and the system of customer login can also be embedded in the system to keep the record of customers.

2.7.4. Comparative Analysis:

The features of different Systems are compared as below:

S.N	Features	Mobile Store Management System (Park D. Patel)	Medical Clinic Management system	Online Music Store Management System	Mobile Store Management System
1.	Platform	Desktop based	Desktop based	Web based	Web Based
3.	Record of current inventory	Available	Available	Available	Available
4.	Record of Customer Details	Available	Not Available	Available	Available
5.	Recommendations	Not Available	Not Available	Available	Available
6.	Supplier Information	Not Available	Not Available	Not Available	Available
7.	Purchase Returns Entries	Not Available	Available	Not Available	Available
8.	Sales Return Entries	Not Available	Not Available	Not Available	Available
9.	Order receive	Not Available	Not Available	Not Available	Available
10.	Different User Levels	Not Available	Not Available	Not Available	Available
11.	Order supplier	Not Available	Not Available	Not Available	Available
12.	Password Recovery	Available	Not Available	Available	Available
13.	Built in	Visual Basic	Visual Basic	ASP.Net	ASP.Net
14.	Store Location	Not Available	Not Available	Not Available	Available

The Proposed system (Mobile Store Management System) will be built is ASP.Net framework. The System will provide different recommendations of products which are running low on stock in the store and which should be imported to the store. It will also provide details of high quantity products to avoid overstocking. Details of purchase returns are also available in the proposed system.

2.8. Proposed Tools and Techniques:

2.8.1. Programs which would be used:

- Microsoft Visual Studio 2017:

Microsoft Visual studio is IDE Integrated Development Environment from Microsoft. Visual studio is mostly used to build different computer programs for Microsoft Windows, web sites, web applications, and mobile applications. It utilizes the Microsoft software development platforms like Windows Forms, Windows API, Windows presentation Foundation, Windows Store, Microsoft Silverlight etc. It can also generate both managed and native code.

- Asp.Net Framework

The ASP.NET is the next version of the Microsoft's Active Server Pages (ASP) which is a feature of the Internet Information Server (IIS). The framework allows dynamic building of web pages by entering queries to a relational database in the webpage. It is a integrated web development model unified with .NET framework in order to provide services for creating web applications and web services. It supports the code written in compiled languages which includes C#, C++, Perl etc.

- Bootstrap:

Bootstrap is a free and open source front end development framework for the creation of websites and web apps. The Bootstrap framework is built on HTML, CSS, and JavaScript (JS) to facilitate the development of responsive, mobile-first sites and apps.

- Html5/CSS3/Java Script

- Microsoft SQL Server:

Microsoft SQL server is a RDBMS relational database management system. It supports a wide range of business intelligence, transaction processing, and analytics applications in corporate IT environments. The Microsoft SQL server is one of the most market-leading database technologies including Oracle Database ad IBM's DB2.

- IIS server:

IIS stands for "Internet Information Services." IIS is a web server software package designed for Windows Server. It is used for hosting websites and other content on the Web.

2.8.2. Personal Techniques:

- To Get familiar with Visual Studio 2017 and ASP.Net Framework:
- To have proper knowledge on web application and .Net programming.

Chapter 3: Development:

3.1. Methodology Considerations:

3.1.1. Throw away Prototyping:

The Throwaway prototyping is such type of methodology in which a standard prototype of the system is developed at the first. After the development of the prototype, the final users are requested to make evaluations of the software. After getting the feedbacks from the end users, the developed prototype is thrown away. A new system is further developed as per the requirements of end user. The process is repeated until a ultimate system is developed.

Throw-away prototyping

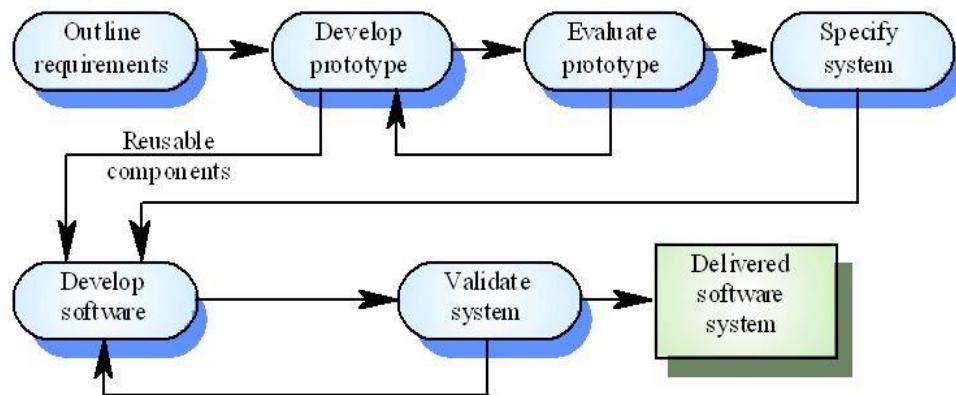


Figure 4: Throwaway Prototyping

The advantages of this methodology are as follows:

- 1) Evaluation of the user is considered.
- 2) Feedbacks are mainly focused on this methodology.
- 3) At the first, a prototype is developed so that the developer can change the requirements and designs in accordance to the provided feedbacks.

The disadvantages of this methodology are as follows:

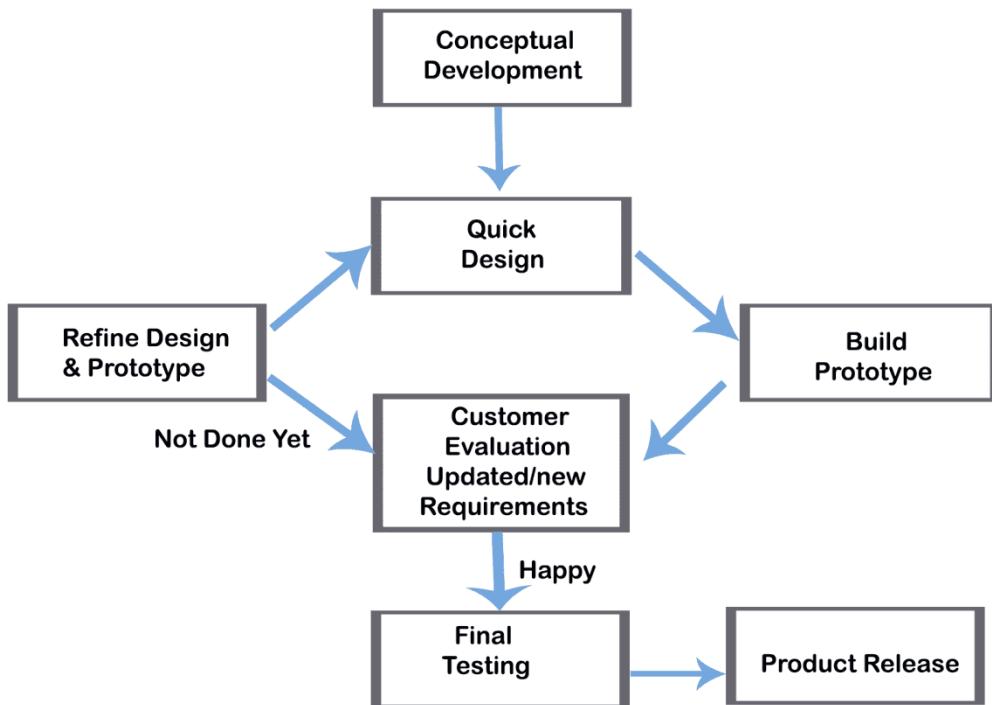
- 1) Time consuming method,
- 2) Changing requirements.
- 3) Diversion of the developer from the project.
- 4) Suitable for big and expensive projects.

This methodology is not considered appropriate for his project as it consist some major disadvantages. The developer can be diverted even before the completion of the project. This methodology is suitable for the project with high number of resources and funds. It is also not appropriate as the time constraints is very less and it may take much determination to finish the project.

3.1.2. Evolutionary Prototype:

The Evolutionary Prototype methodology is such type of methodology in which a prototype of the system is developed at the first. After the development of the prototype, the final users are requested to test and evaluate the prototype. After taking the evaluation and feedbacks from the customers, the same prototype is utilized to build another prototype system. This system is also repeated until a final system is developed.

e-Tech3.com



Evolutionary prototyping process model:

Figure 5: Evolutionary Prototype.

The advantages of this methodology are as follows:

- 1) Gathering of feedbacks form evaluation.
- 2) No distraction of the developer.

-
- 3) Use of the developed prototype to build another prototype.

The disadvantages of this methodology are as follows:

- 1) Time consuming method.
- 2) Change in requirements.
- 3) Appropriate for large and expensive projects.

This methodology is also not considered the best methodology for the development of this system because the system could be more and more improved. In accordance to the development trends, many changes to the system can be made and the ultimate exposure of the system is also not sure.

3.1.3. Waterfall Methodology

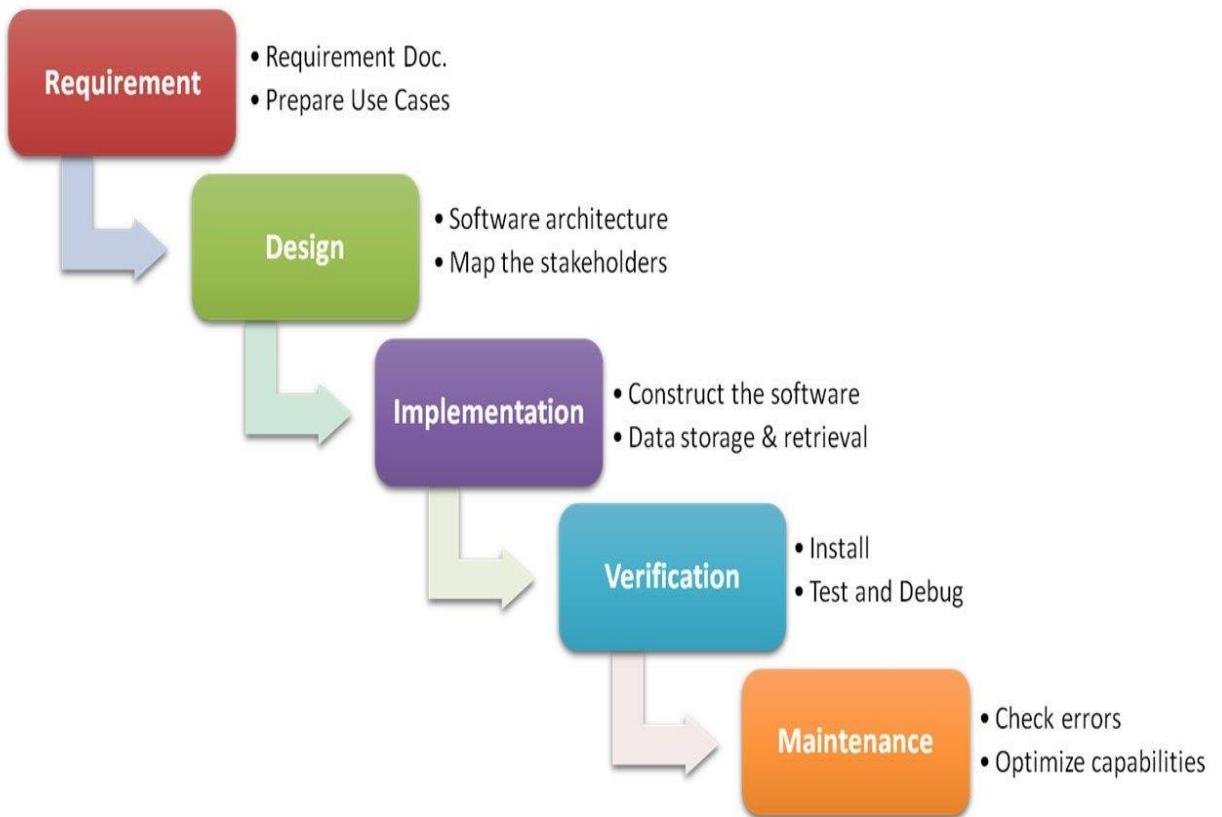


Figure 6: Waterfall Model

The waterfall model is also known as linear sequence life cycle model. This software development methodology is very simple and easy to user. In this model, every phase must be fully completed before moving on to the next phase. The next phase can only begin after the completion of the previous phase so there is no overlapping of the phases. This model is considered as the earliest method of SDLC which was used for software development. This model illustrates the process of software development in a linear sequential flow. Phases in the waterfall model don't overlap.

Steps of waterfall model:

- Requirement Analysis
- System Design
- Implementation
- Verification
- Maintenance

Advantages of waterfall model:

- Simplified and easy to understand.
- Easier management due to rigidity of model
- Phases don't overlap.
- Suitable for small projects where there is clear understanding of requirements.

Disadvantages of waterfall model:

- Impossible to go back to the previous stage to change something.
- High risk or uncertainty.
- No working prototype is developed until late during the life cycle.
- Not suitable for large and complex projects.
- Not suitable for the projects where are at moderate to high risk of change.

When to use the waterfall model:

- Clear understanding of requirements.
- Definition of the product is stable.
- Understanding of technology.
- No ambiguous requirements.
- Project is short.
- Resources are sufficient.

The waterfall methodology is not exactly appropriate for the development process of my project. This methodology fulfills most of the requirements of my project. The project is small; the requirements of the project are clear and fixed. But in this methodology, once a phase is completed, the developer cannot go back to the previous phase. This is one of the main reasons why I am not applying this methodology for the development of my project.

3.1.4. Selected Methodology:

Enhanced Waterfall Model:



3. Enhanced Waterfall SDLC

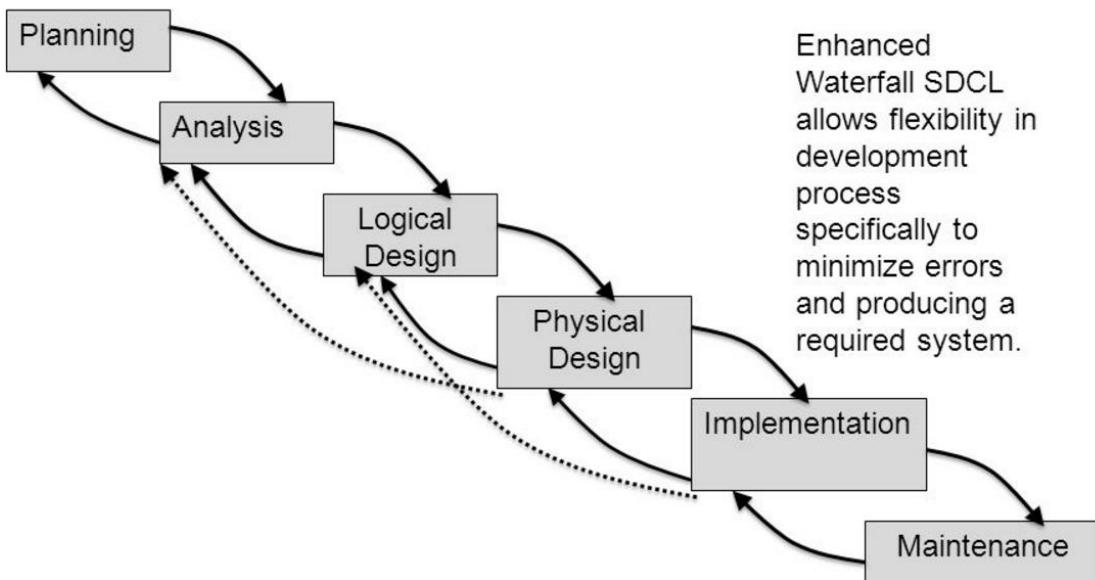


Figure 7: Enhanced Waterfall Methodology.

The Enhanced Waterfall methodology is considered as an upgraded version of the traditional waterfall methodology. It is one of the popular versions of software development methodology which is adopted in an environment where all the requirements of the system are clear, fixed and fully understood. This methodology is somehow different from the classic waterfall methodology as the developer of the system can take a step back to the previous phase in case of some bug or problem occurrence which was not possible in the traditional waterfall methodology.

The advantages of this methodology are:

1. Simplified, easy to use and understand.
2. Appropriate for small projects where there are clear understandings of requirements.
3. Model phases are processes and completed one at a time.
4. Ability of rolling back changes in some cases.

The disadvantages of this methodology are:

1. Not appropriate for huge and complex projects
2. Not suitable for projects with changing requirements
3. High Risks.

3.1.5. Reasons for selecting this Methodology:

The reasons for selecting the Enhanced waterfall methodology are as follows:

1. Simple phases and easy to understand and use.
2. Clear understandings of the requirements of my project.
3. The project is small and matches the requirement of the methodology.
4. Ability of going back to previous phase in case of problem occurrence.
5. Requirements of the system are fixed.

Steps:

Planning: I will create different time plans, project plans, select development tools etc.

Analysis: I will create different requirements using SRS and analyze those requirements.

Logical Design: I will create different charts and diagrams of the system(ER, Class diagram).

Physical Design: Actual development of the system i.e. coding and interface and database design.

Implementation: I will implement the engineered system to the end user site.

Maintenance: I will test and main the system after implementation for future use.

The Enhanced Waterfall Methodology is considered best methodology for the development of this project. This methodology is simple, easy to understand and use. The methodology is appropriate for small projects like this one where requirements are easily understood and fix that is as required for my project. The management process is very easy. And the developer can go a step back to the previous phase in case of problem occurrence.

3.2. Methodology stage 1: Planning:

Planning is the process of creating a set of plans that guides through the execution and different of a project. Project plan is the first part of every project. The plans that are created during this stage will help in managing time cost quality and risk issues. As per the methodology I followed, Firstly I selected an appropriate project, created a proposal and submitted it to the supervisor. Different project time plans, development plans were created in this stage.

3.3. Methodology stage 2: Analysis:

In the analysis stage, different requirements of the system are captured and developed. At the first the design function and purpose are clearly understood and the input output and the final product are specified and studied.

3.3.1. Software Requirements Specification:

Software requirement specification can be defined as a complete description of the planned purpose and the environment for the software under the development. The major objective of software requirement specification is to describe the objective of software and how it will be expected to perform. It helps in minimizing the time and effort that is needed by the developers to gain desired objectives and it also reduces the cost of development. A good software requirement specification should properly define the interaction of an Application with system hardware and human users in real world situations.

3.3.1.1. Functional Requirements:

These requirements define the goal of a system or a product. In other words, these requirements specify what the product should actually do. These requirements help in relating the actions that should be carried out by a product for satisfying the essential reasons for its existence.

- The application shall have all required functionality which is necessary for mobile store inventory and billing system management.
- The application shall have ability to suggest the high stocked and low stocked products.
- The application shall have functionality to be used by different users.
- The application shall have ability to maintain the repair record uniquely, storing the IMEI number.
- The application shall have ability to store all the necessary details of customers and suppliers.
- The application shall have ability to keep records all the products available in the store.
- The application shall have functionality to calculate the total amount of sales done.
- The application shall have ability to calculate the total amount of purchases sales and orders done in current day.
- The application shall have ability to provide the location of the store to its customers.
- The application shall have functionality to reset the password of the users.
- The application shall have user name and password protected security system

3.3.1.2 Technological Requirements:

Backend database system and frontend programming tools are included in the technological requirements of a system. The application should be easily used and managed by any person or user having a little knowledge of computers. The database of the system should also be very easy to install manage and use. The database should be very portable and independent so that it can be used in various devices like laptop and pc. The application should also be flexible and versatile. It should be able to operate on various kinds of devices and it shouldn't require high level on configuration on any machine.

3.3.1.3. User Interface Requirements:

- The application should be simple and easy to understand and use.
- The interface should have very attractive user interface.
- The application should fulfill all of the requirements of a general user interface.
- The application should take very few inputs from user.
- The system should be understood by the user in very short time.

3.3.2. Project Features:

➤ Repair records:

Keeping the records of the phones that are given to the shop for repair service is very challenging. The customers may have the same mobile phone of the same model and colour. The system will have ability to store the IMEI number which is unique to every phone. The records will also consists information about the customers and their respective phone model. Thus the repair record of the customers can be uniquely identified.

➤ Record of inventories:

The record of the inventories of the store which includes, mobile phones, covers, tempered glasses, earphones and other related accessories are properly maintained in the database of the system. Different types of inventories with their respective prices, image and available quantity are set on the database of the system.

➤ Contact details of suppliers:

The details of the supplier along with their contact information are very important to any business. The system will have ability to store the necessary details of supplier and also store the links of their sites if available.

➤ Customer Records:

Customers are one of the main aspects of every business. There can be many-many customers of a certain store. Some might be a regular customer and some might not be so regular. The system has the ability to store various customer details like name, age, contact, image, address, email etc.

➤ Order:

Receiving orders of different products directly from the customers will increase the business transactions of any store. If ordering of products is made easily accessible and easier to customers of the store, the customer will no longer have to constantly visit the store to order any sort of product. This feature will gradually increase the number of sales of the business along with satisfied customers.

➤ Order Supplier:

The system allows the store keeper to order different types of products from the supplier. The orders that are received from the customers can be seen and analysed and the products that are not available or are in insufficient quantity can be recorded in the database of the system.

➤ Location information:

A customer of the business can get the actual location of the business through the location feature of the system. An unknown customer can use the system and get instant location and directions of the store through Google maps.

➤ Password recovery:

If the users of the system accidentally forgot their password, they can also recover their password using the recovery keys that are provided to the users by the admin during the registration process.

➤ Password change:

The users of the system can easily change their passwords as per their need and their recovery keys for security purposes.

➤ Cash and Credit purchase:

The system stores the purchase details of the store in two different ways that are cash and credit purchase. The purchases that are done cash to cash are labelled as cash purchase in which the balance of the customers are not recorded or updated. In credit purchase, the balance of the customers is updated with the total credit purchases.

➤ Record of Receipt and Payment:

The transactions of purchases and sales that are done in credit basis are responsible for the update in balance of the customers or suppliers. The receipt feature allows receiving of payment from different customers to their respective credit purchases and the payment feature allows payment to the suppliers for the respective credit purchases.

➤ Record of sales:

Keeping the record of sold products is a very important aspect for any business organization. The system will maintain a systematic record of every sold item from the shop in the database of the system. Sales data and information which includes the customer details, product quantity and discount details are stored in the system.

➤ Recommendation of products:

Overstocking and Low stocking is one of the major problems of every business. To minimize the risk of overstocking the system will have capability to read the product with high quantity from the database. The system will have the ability to recommend the product with low quantity that should be imported by the shop owner into the business.

➤ Record of purchase and returns:

Some of the products can be damaged or may not work properly in the long run. The system will also have the ability to keep records of the damaged good and the purchased good that may not have worked properly and are to be returned to the supplier.

➤ Record of Sales returns:

In the long run of the business, some of the good may not work properly or may not work at all after the products are sold to the customers. Some products may stop functioning even before the expiry date of the warranty and guarantee service. In such cases some of the goods should be returned to the store from the customers. The system consist the ability to store the sales returns details and update the balance.

➤ Calculation of Total Sales:

The sales transactions that are done by the system are stored in the respective database. The total amount and number of sales are calculated and presented to the user of the system. The number and total amount of sales of the current day is also calculated by the system.

➤ Calculation of Total Purchases:

The transactions of purchases are stored in the respective table of the system. The total amount and quantity of the purchase entries are calculated and presented to the user automatically by the system. The total purchase of the current day is also calculated by the system.

➤ Calculation of Total Orders:

The total number of orders, either it is order done by customers or it is order to suppliers are also calculated and present to the user as a whole or as current day.

3.4. Methodology stage 3: Logical Development:

3.4.1. Introduction:

A logical design can be known as a conceptual design or an abstract design. In this phase the actual physical implementation is not initiated. Definition of different types of useful information is done in this stage.

Logical development process includes arrangement of data into different series of conceptual relationships called entities and attributes. In a relational database, an entity refers to a table and an attribute refers component of an entity. An attribute marks a column in a relational database. The logical database is generally created using a pen and paper or it can be designed using different tools and software over the internet.

Logical design is very important for the implementation of corporate database. An incomplete logical design or a faulty design may be very problematic and costly to the system later while making changes to the means of data collection, protection and storage. A good test database can be created easily with a good preliminary design. For the successful implementation of a system, a sound database or logical design is essential.

An accurate and complete logical design for a system helps to ensure the followings:

➤ Data independence:

Development of a logical design creates a model of database which is independent of program or requirements of physical storage. The developed model displays the structure of data that are represented to the users. Maintenance and processing of data structures are not specified in this process.

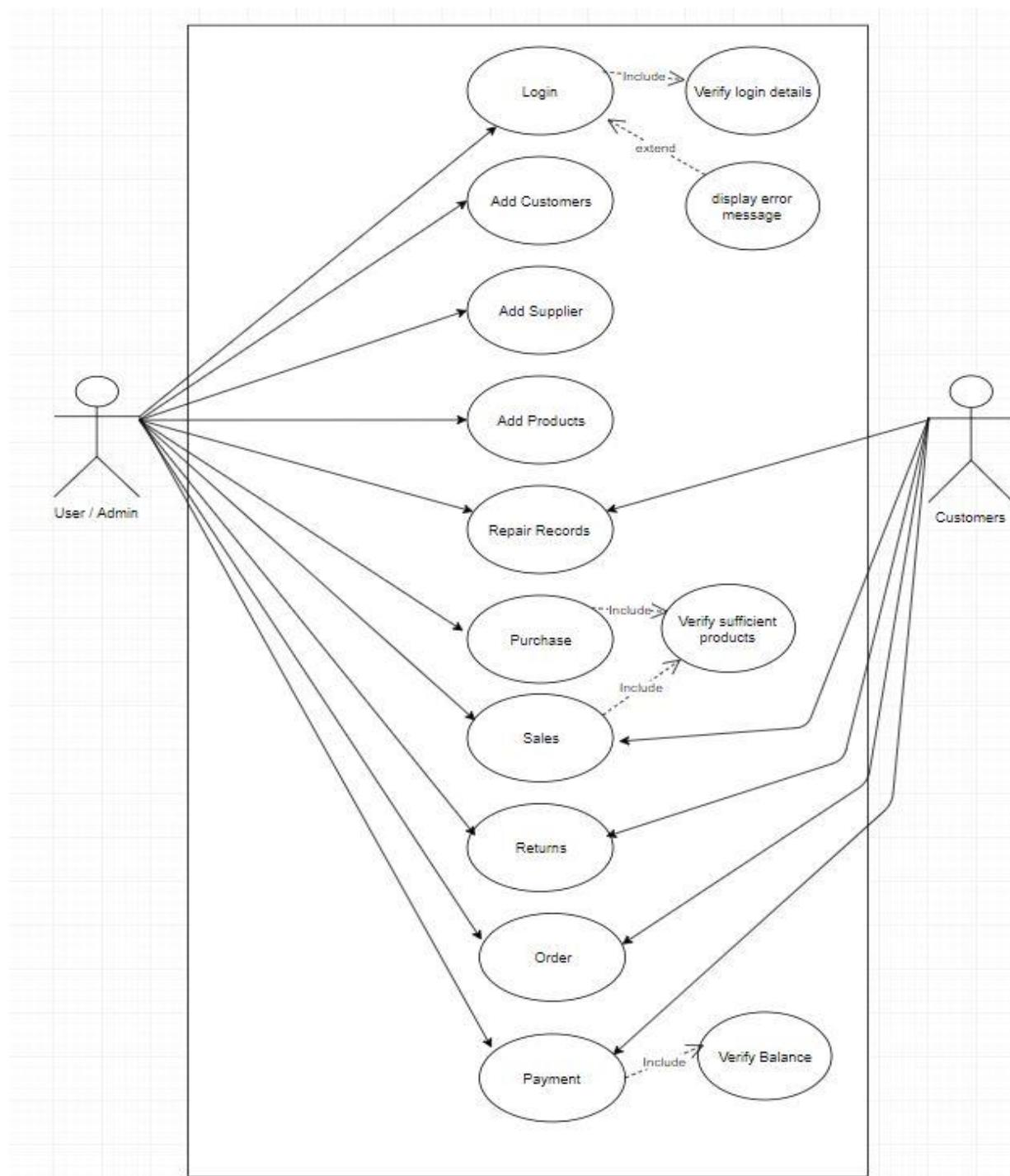
➤ Physical database flexibility:

The logical design is independent of the storage and performance requirements. It can be easily used for the implementation of a database with any sort of hardware and software system. During the physical design process, the logical design can be used to satisfy the needs of particular users or data processing environment.

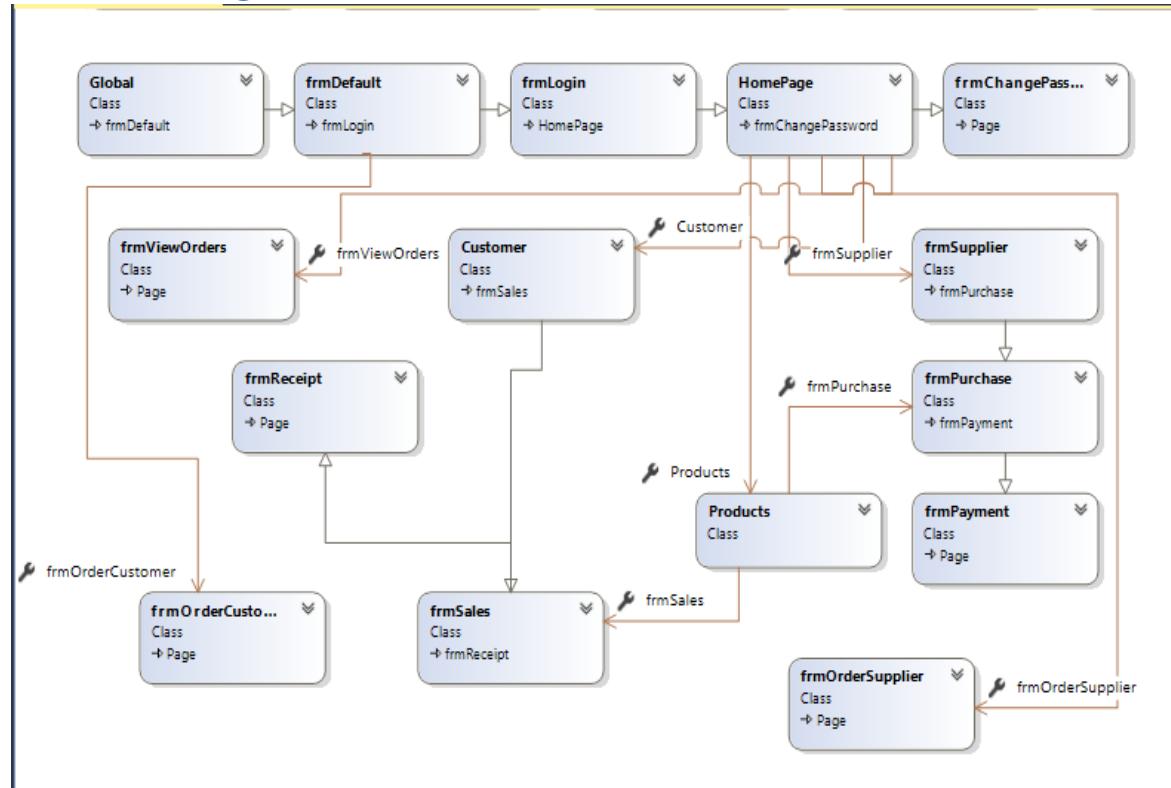
➤ Integrity:

The data maintained in the system and the rules of business are identified by the logical design of the system. To define integrity rules for physical design, the business rules can be used.

3.4.2. Use Case Diagram of the system:



3.4.3. Class Diagram:



3.4.4. Classes:

Global Class:

Default Class:

1

Login Class:

1

Order Class:

]

Class Homepage:

]

Form Customers:

]

From Supplier:

]

Class Product:

]

Class Purchase:

1

Class: Sales:

1

Class Purchase returns:

1

Class Sales Returns:

]

Class Password Change:

1

Class Password Recovery:

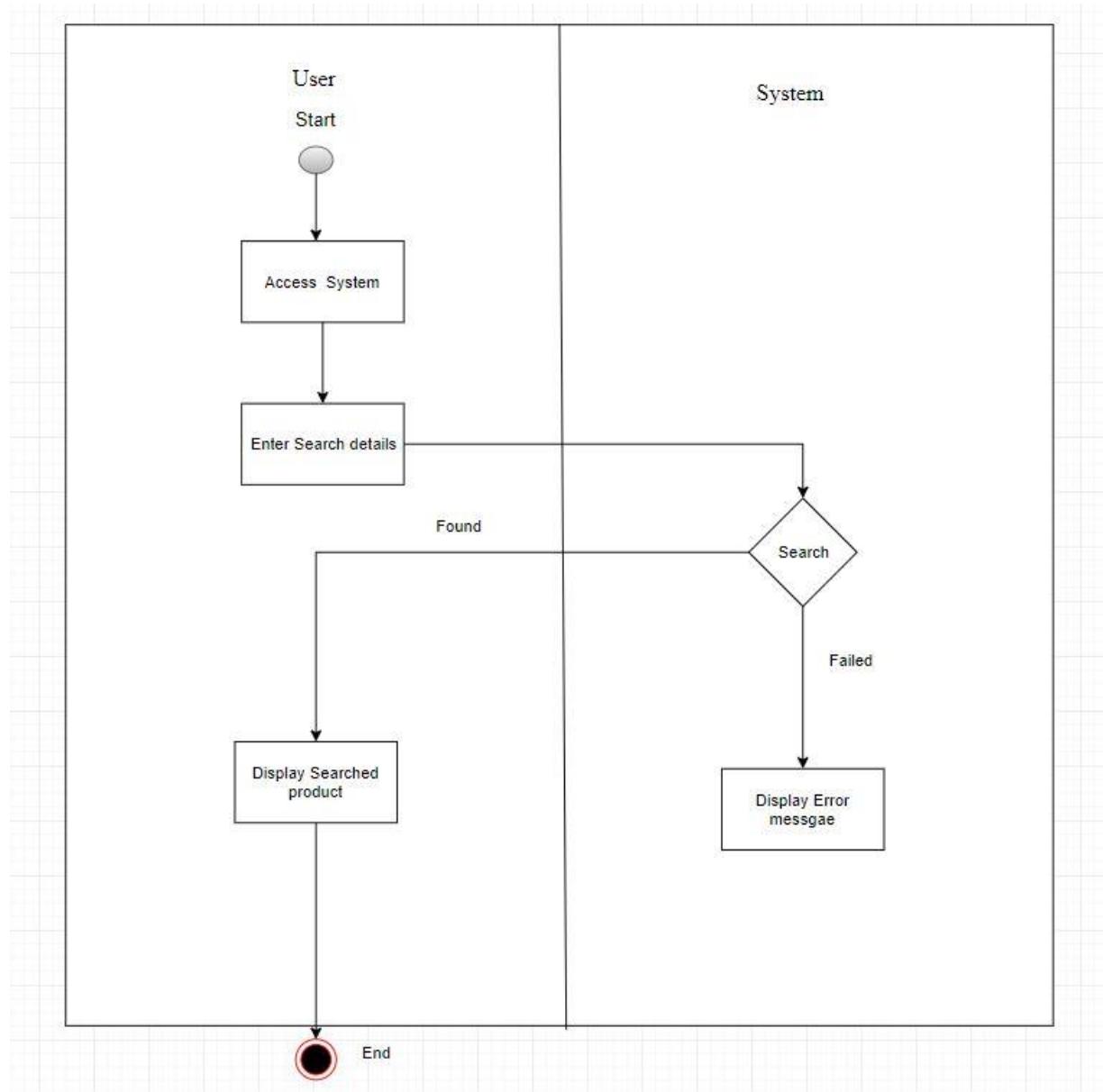
1

Class Register:

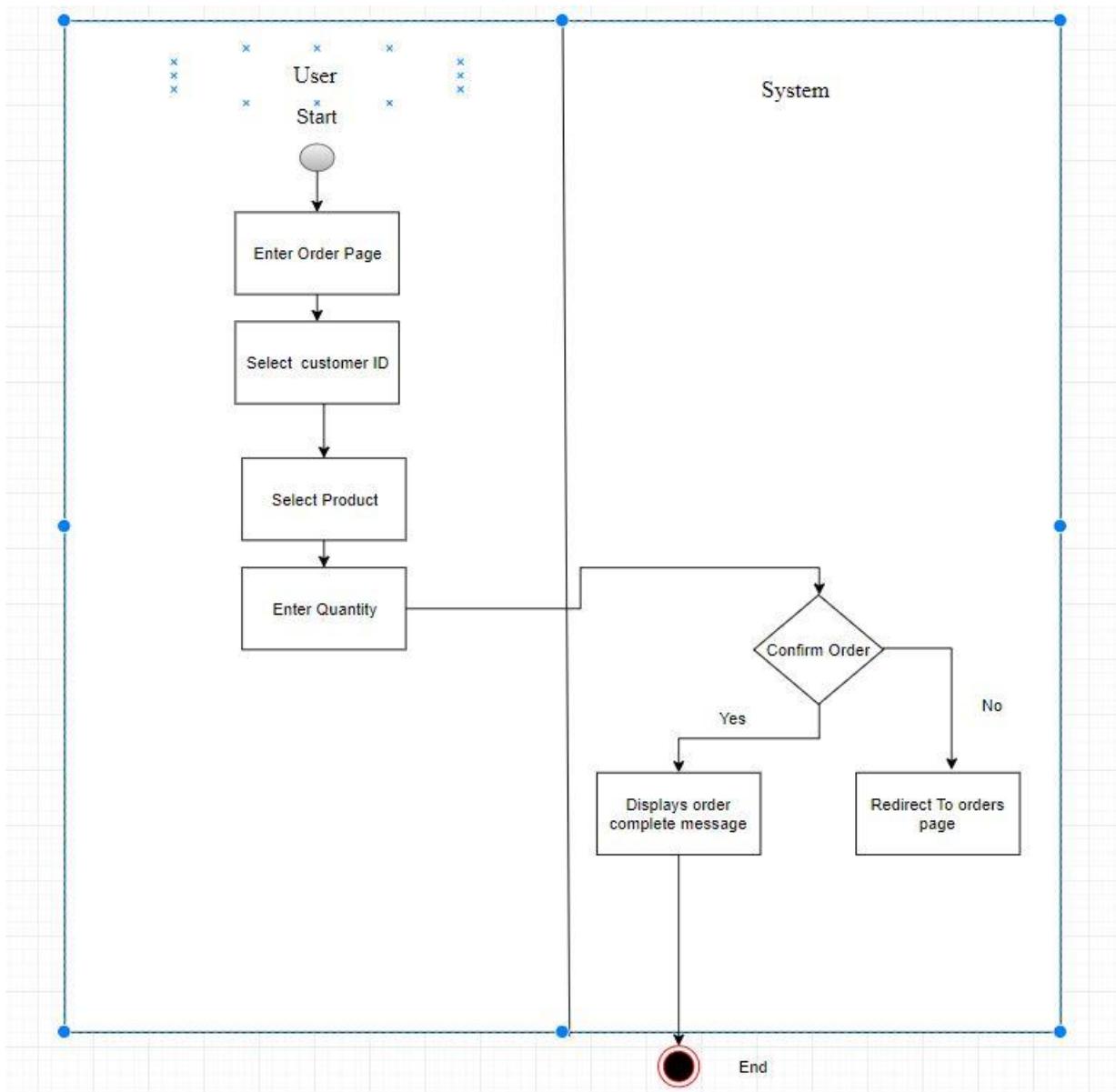
]

3.4.5. Activity diagrams of the System:

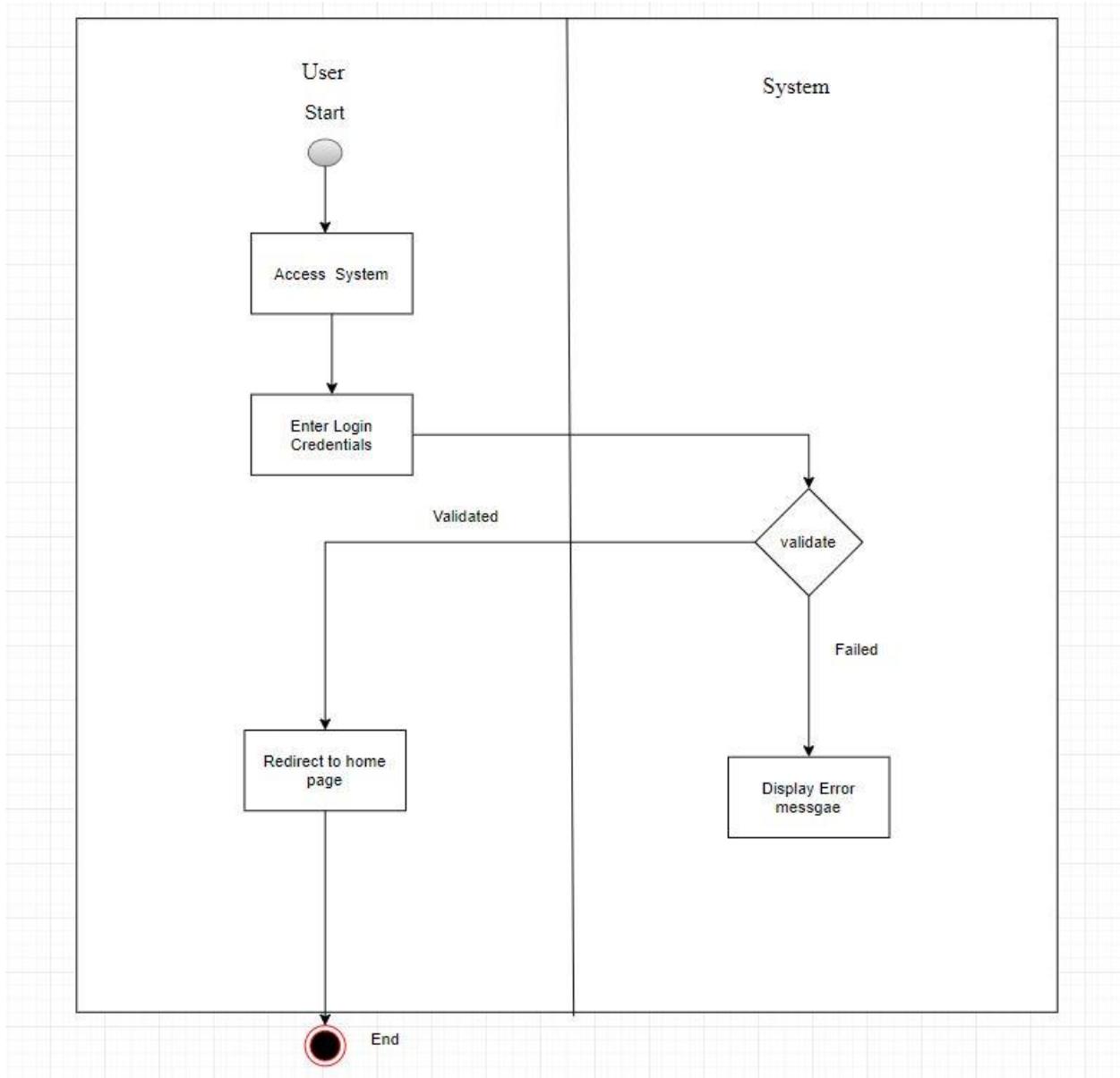
Searching Products in default page



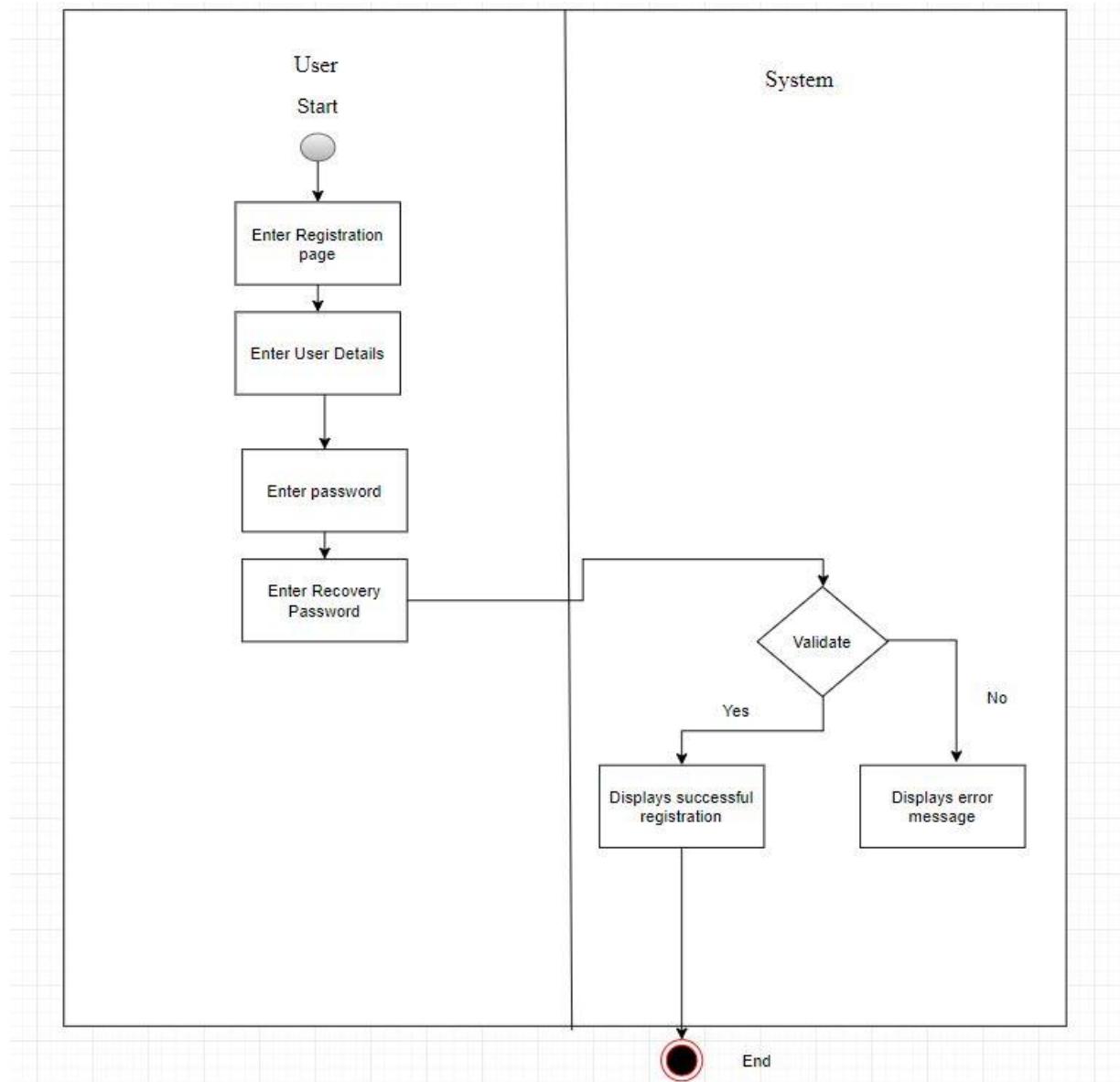
Order Products By Customers:



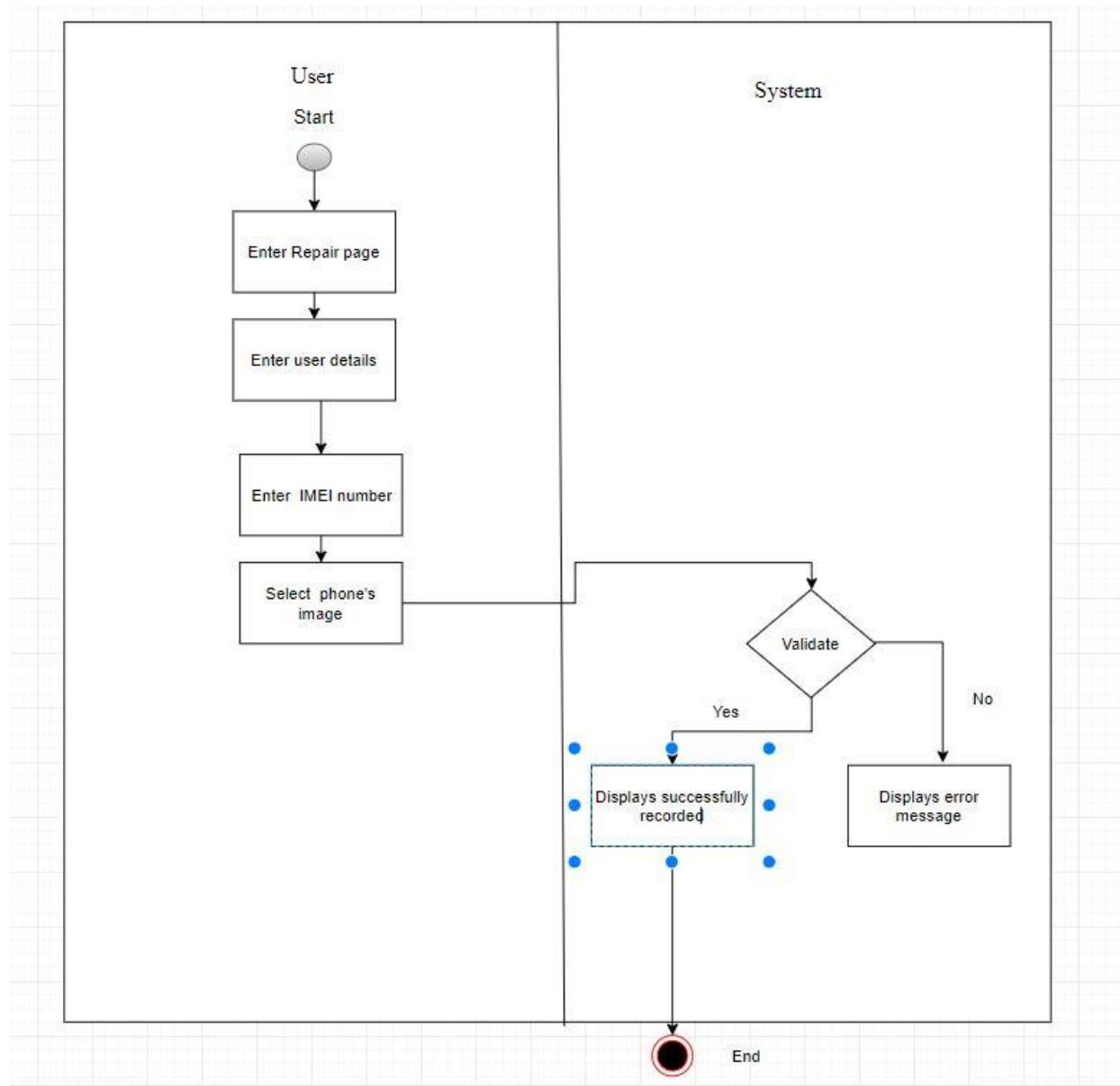
Login Process



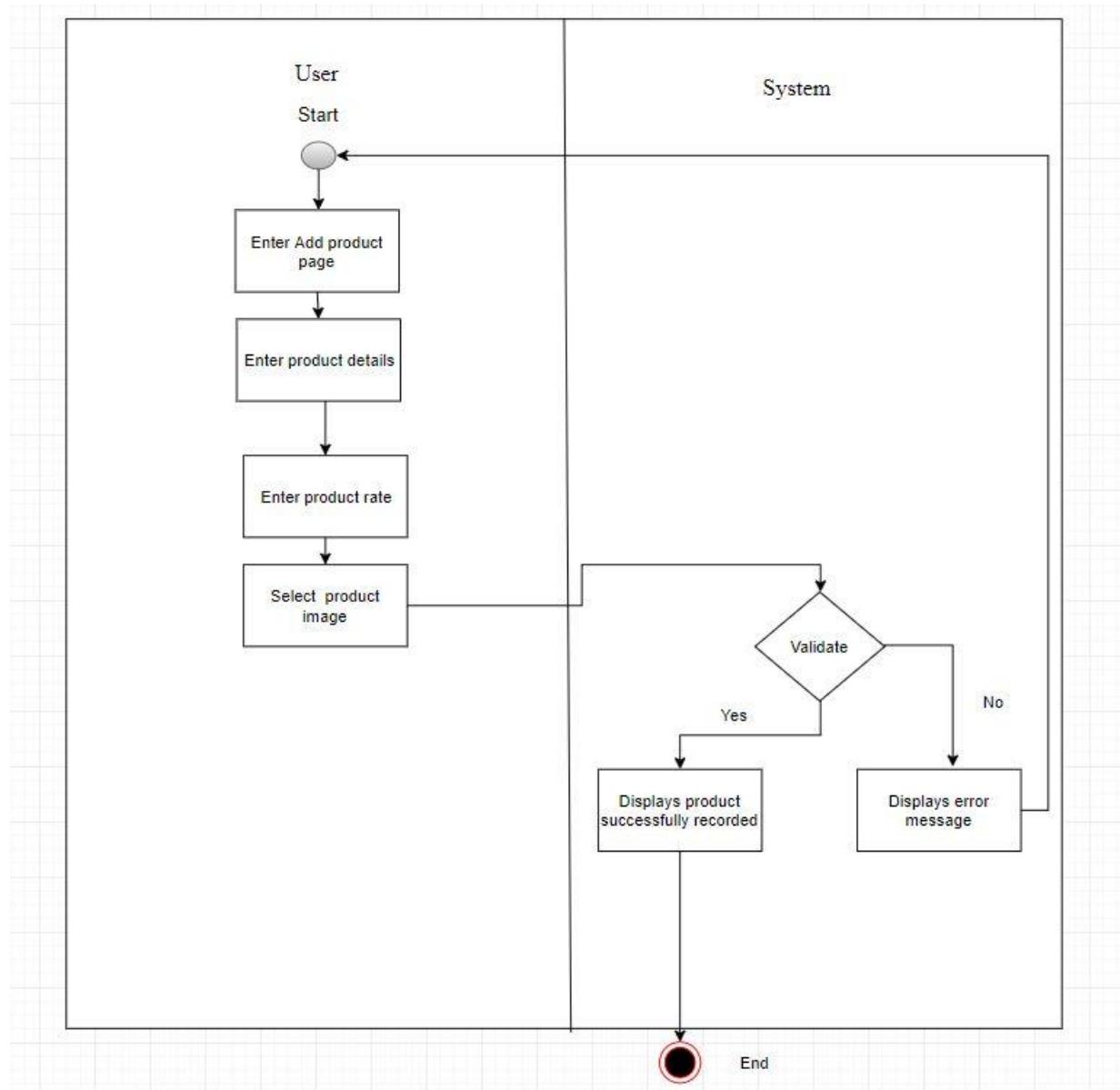
Registration Process:



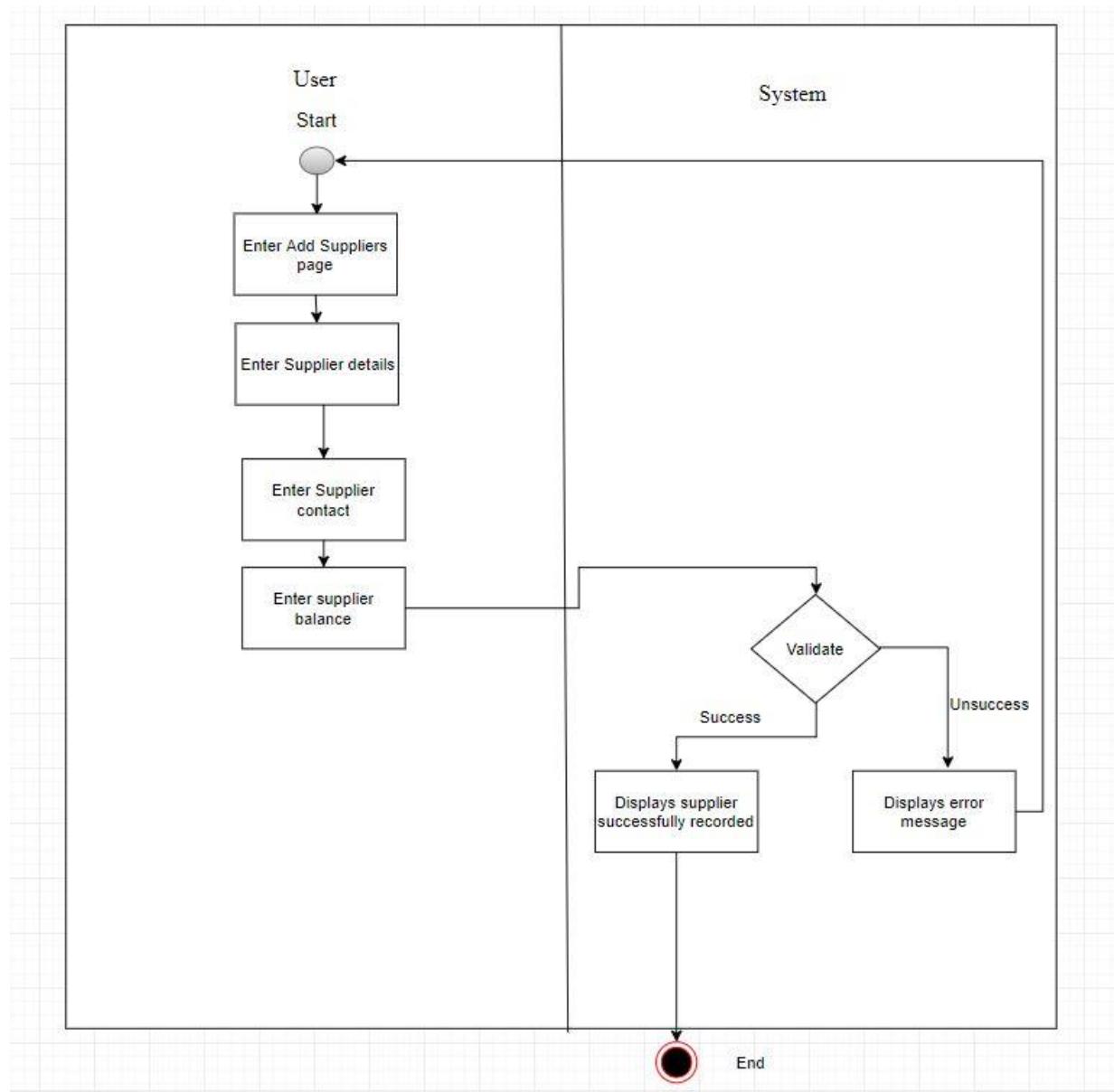
Repair Records:



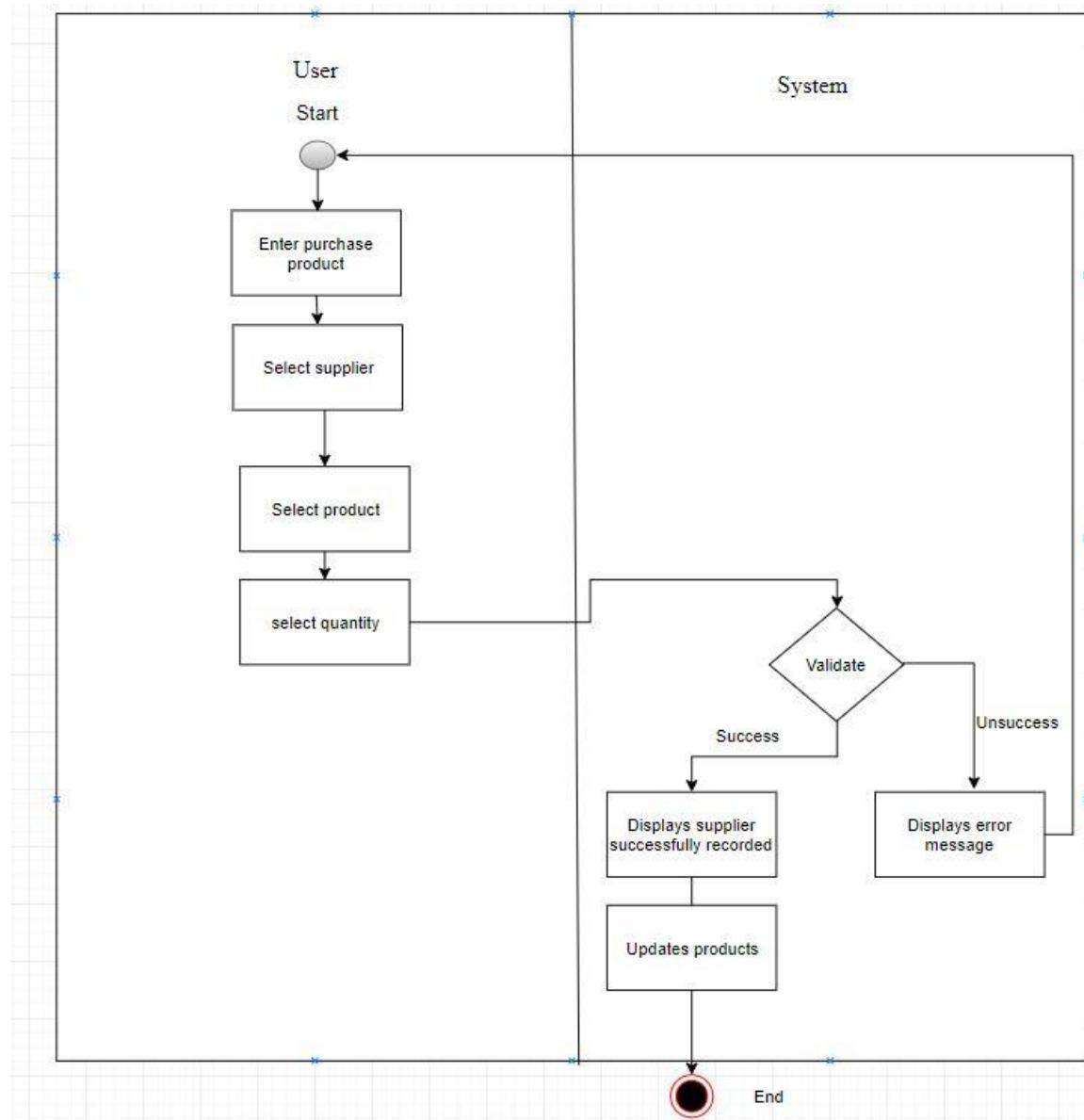
Adding Product:



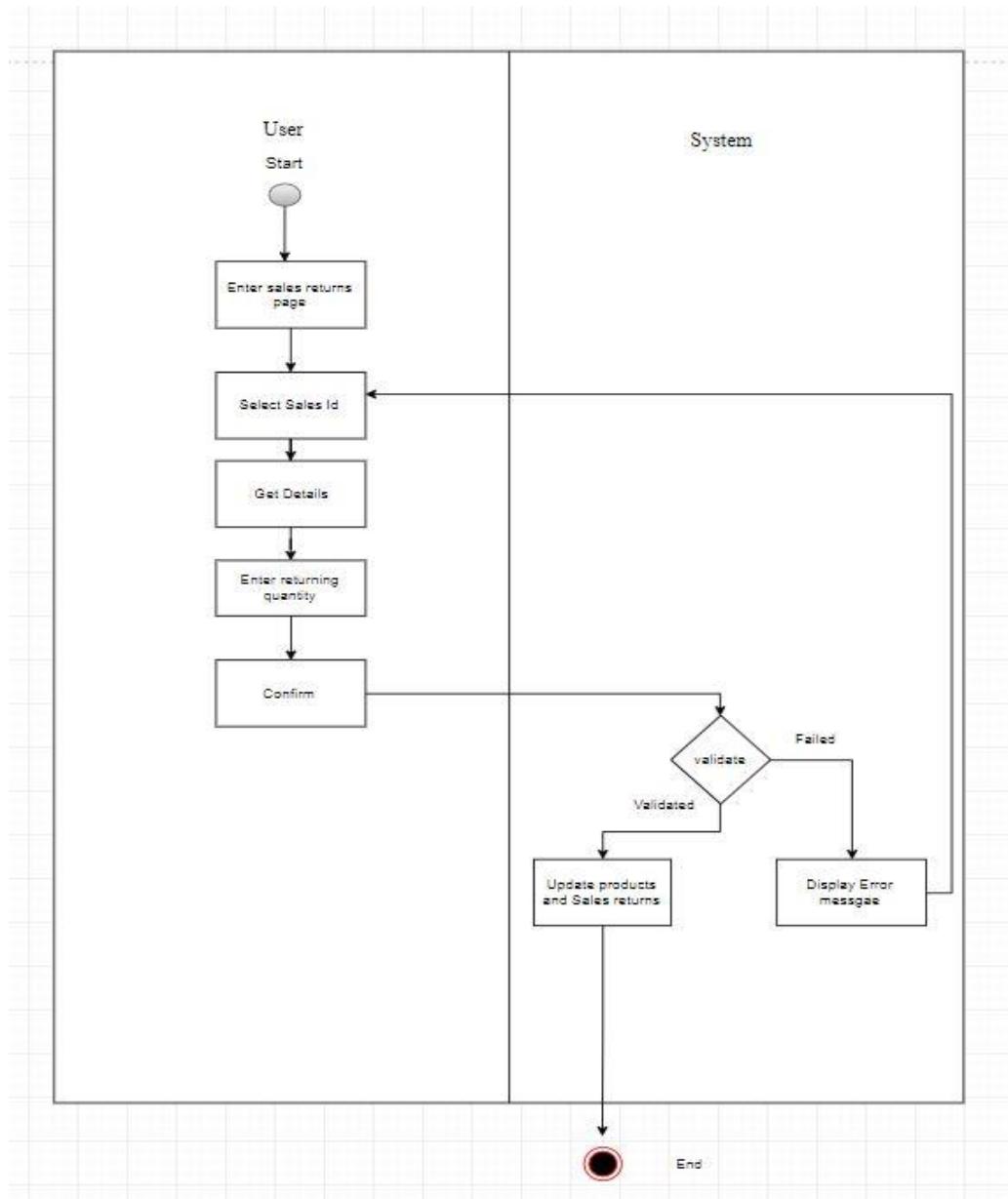
Adding Supplier:



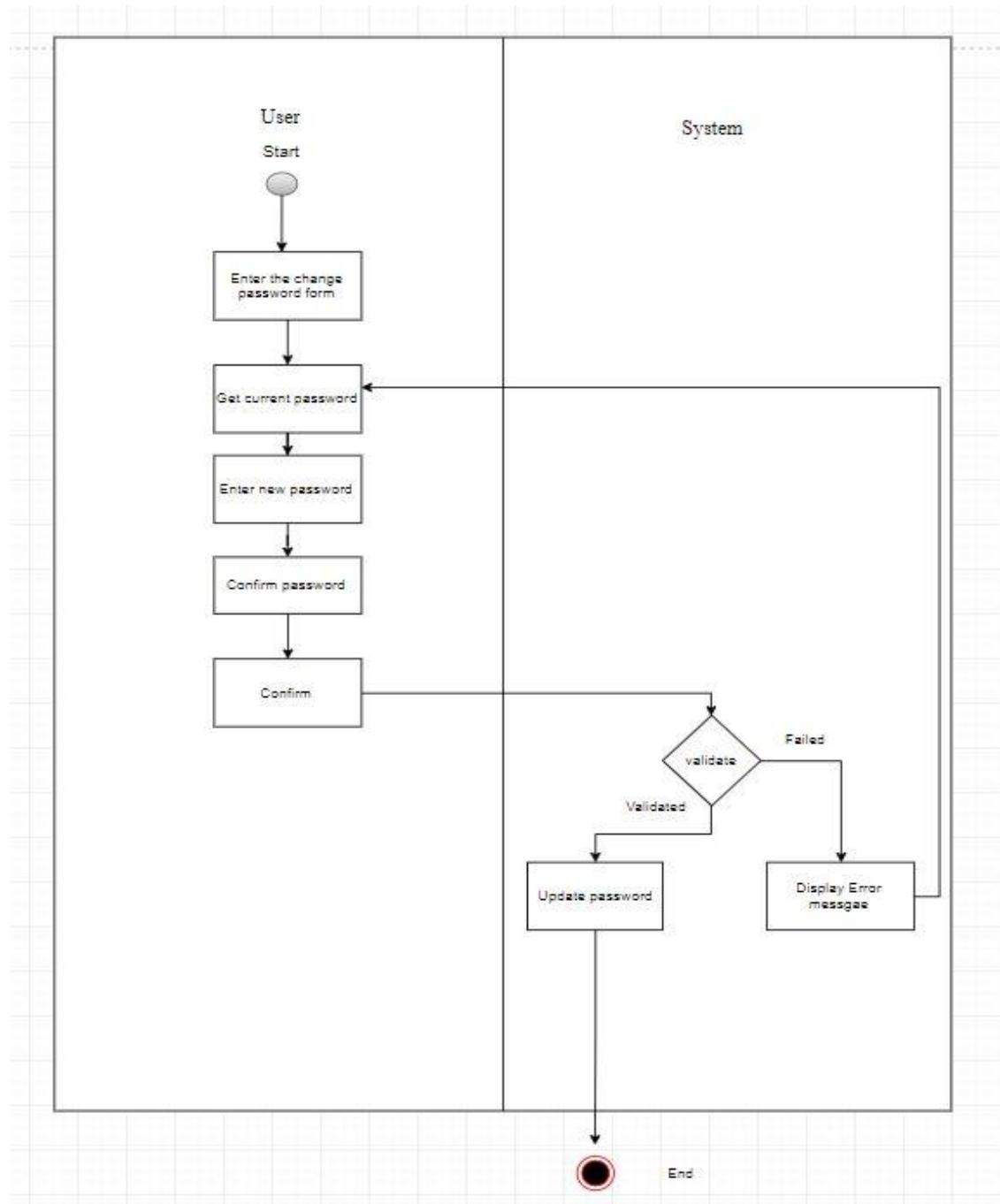
Purchase Process:



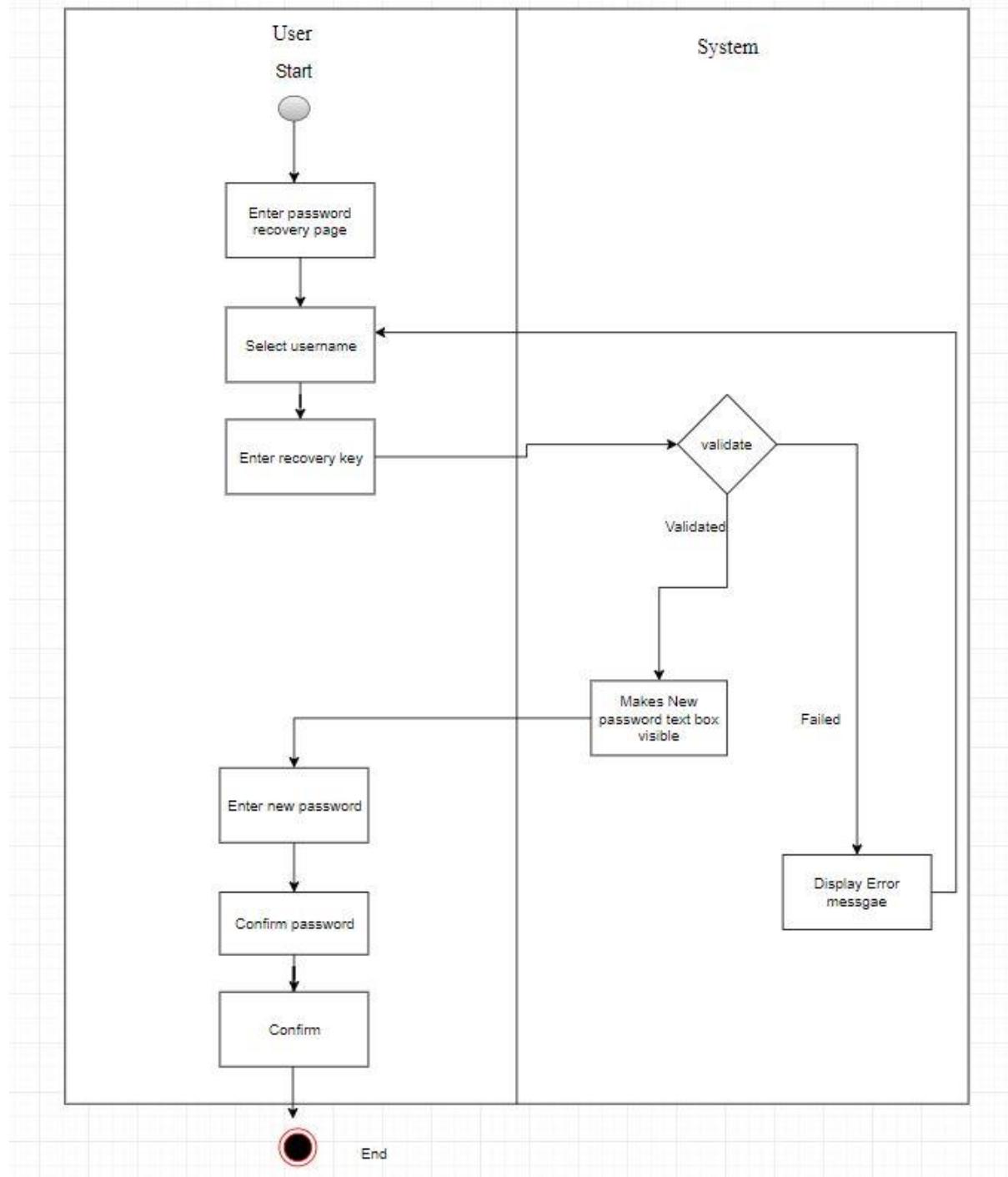
Sales Returns:



Password Change:



Password Recovery:



3.5. Methodology Stage 4: Physical Development:

3.5.1. Physical Development:

Physical development is that development that relates the input and actual output processes of the system. The major focus of physical development is in ensuring how the data and information are entered, verified, processed and output is generated into a system.

An actual working system is developed by the definition of design specification that specifies the requirements of the system. The physical design is considered with interface design, process design and the data design.

Physical design includes the following steps:

- Specification of the input or output media, database design and backup procedures.
- Planning of system implementation.
- Planning of a test implementation plan and specification of hardware and software.
- Updating benefits, costs, system constraints and conversion dates.

3.5.2. Interface Design:

Interface design is one of the most important considerations to be made in the front end design process. Usability issue is a major issue of the current systems. The design of proposed system is planned to be more attractive and user-friendly. The User interface design of the system should include the following factors:

- Simple but attractive.
- Less time consuming
- Less input from user.
- Easier functionalities.
- Artificial intelligent to some extent.

3.5.3. Screen Outputs of interface and Functionality:

Design 1: Default page:

The screenshot shows a web browser window for 'localhost:52946/WebForm3.aspx'. The title bar reads 'Mobile Store Management System'. The navigation bar includes links for 'Login', 'Order', and 'Location'. A search bar at the top right contains the placeholder 'Enter your search here' with 'REFRESH' and 'SEARCH' buttons. Below the search bar is a large promotional image for the 'SAMSUNG Galaxy S9 | S9+'. The image features four phones: two black S9 models and two blue S9+ models, arranged in a staggered fashion. To the left of the phones is a 'Concept Creator' logo. Below the banner is a grid of mobile phone images. The grid includes a Samsung Galaxy S9, a Nokia 3, an HP Envy Laptop, a Samsung G530, an iPhone X, a Nokia 2, and a Nokia 6. Each phone has its name labeled below it. At the bottom of the page is a table displaying product information:

Name	Quantity	Sellingrate
Samsung Galaxy S9 Mobile	1483	120000
Nokia 3 Mobile	1503	13500
Hp Envy Laptop	13	94000
Samsung G530 Mobile	1587	12500
Iphone X	9	170000
Nokia 2 Mobile	40	11600
Nokia 6 Mobile	41	25000

The above presented form is the Default form of Mobile store management system. In this form, customer can see the products, their available quantity and their respective selling rates. The customers can also search the products using the search button in the search bar and refresh the page using the refresh button. The customers can order necessary goods from the order page which is accessible through the order link at the navbar of the page. The customers can also get the location of the store through the location link in the navbar. Users of the system can login to the system through the login link.

Design 2:

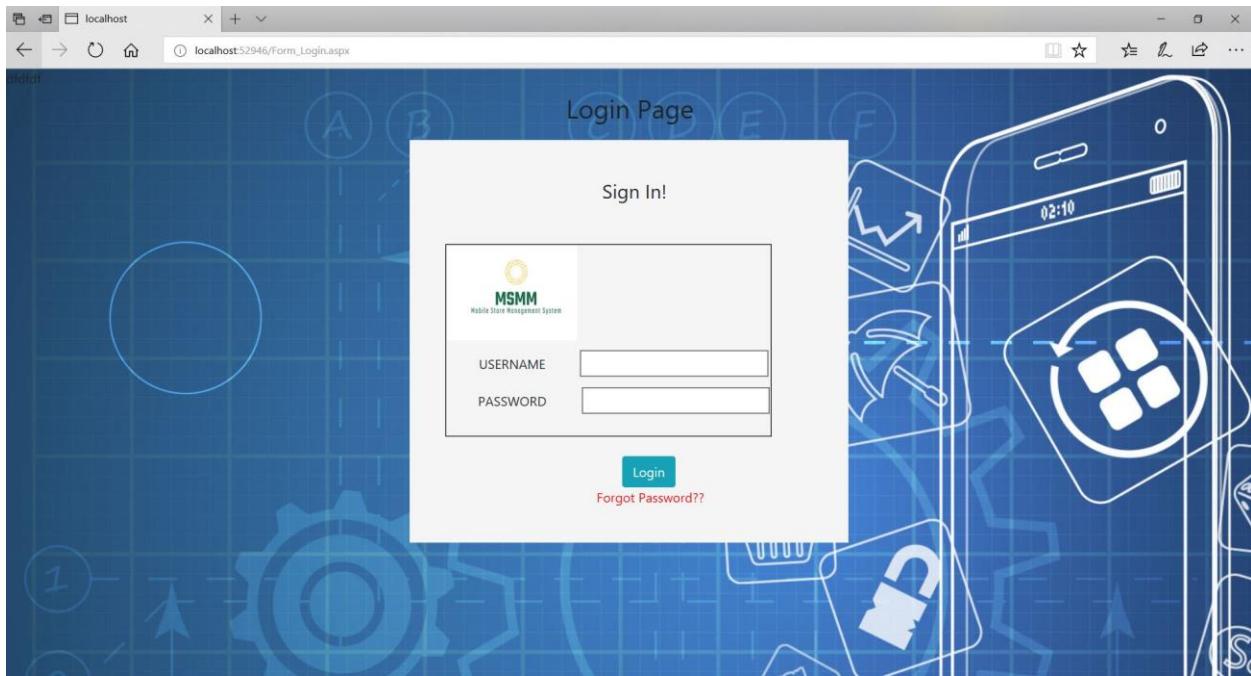
Order

The screenshot shows a web browser window titled "Mobile Store Management System" with the URL "localhost:52946/Form_Order.aspx". The page has a dark background with a blurred image of a smartphone. At the top, it says "Please select the ordering product". Below this, there is a form with the following fields and buttons:

- New Order Id (text input field)
- Get Order ID (button)
- Order Date (text input field with placeholder "mm/dd/yyyy")
- Customer ID (dropdown menu showing "33444203-1dd1-4f2f-8979-aaba")
- Get Info (button)
- Name: Label (text input field)
- Ordering Product (dropdown menu showing "Samsung Galaxy S9 Mobile")
- Info (button)
- Rate (text input field)
- Quantity (text input field)
- ADD (button)
- Confirm Order (button at the bottom right)

The above presented form is the UI design of orders page. This page enables customers to order different products from the store. A registered customer can order the product without visiting the store. Here a customer has to get an order id through the getorderId button select date, select his/her id, select product and desirable quantity and add the product and confirm the order process through the ConfirmOrder button.

Design 3:



Form Usage: Verification of users through the registered username and password.

Satisfied Functionalities:

- Username and password protected system.
- The design is attractive and fresh.

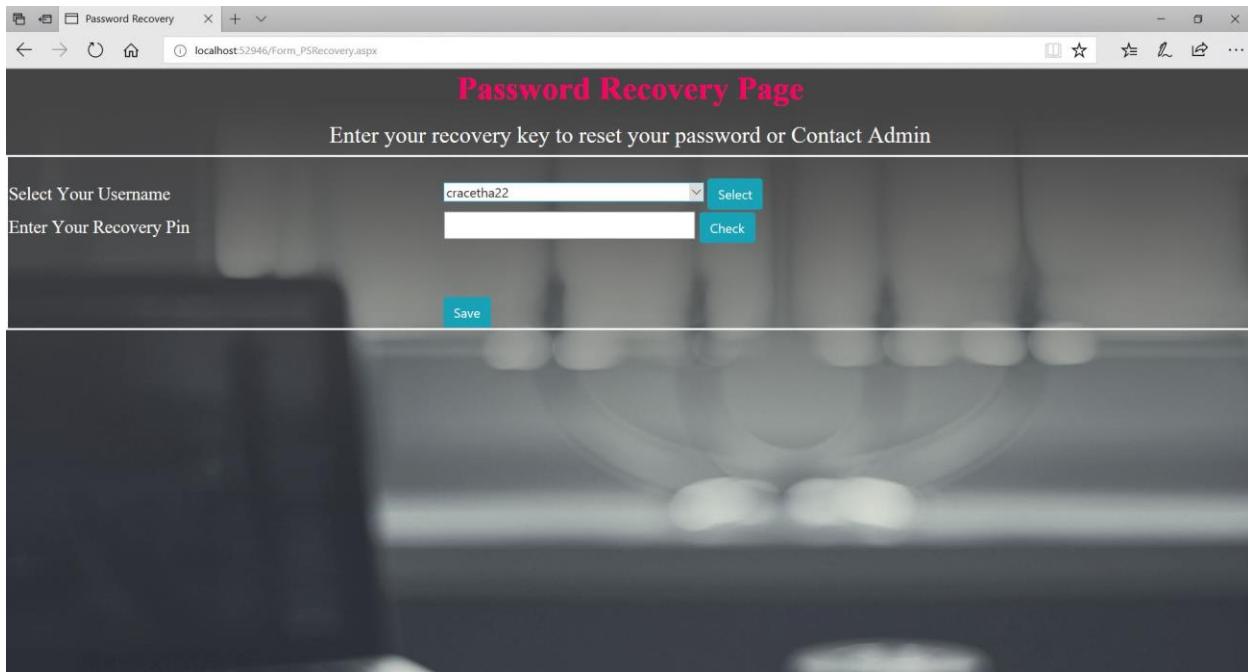
Form Item:

- Login Button:
 - It verifies the username and password entered by the user. It redirects to home page if the information are verified or it is set displays error message in case the information are not verified.

- Forgot password link:

This link redirects the system to the password recovery page.

Design 4: Password Recovery.



Form Usage: Resetting the password of the registered users through the validation of their respective recovery key.

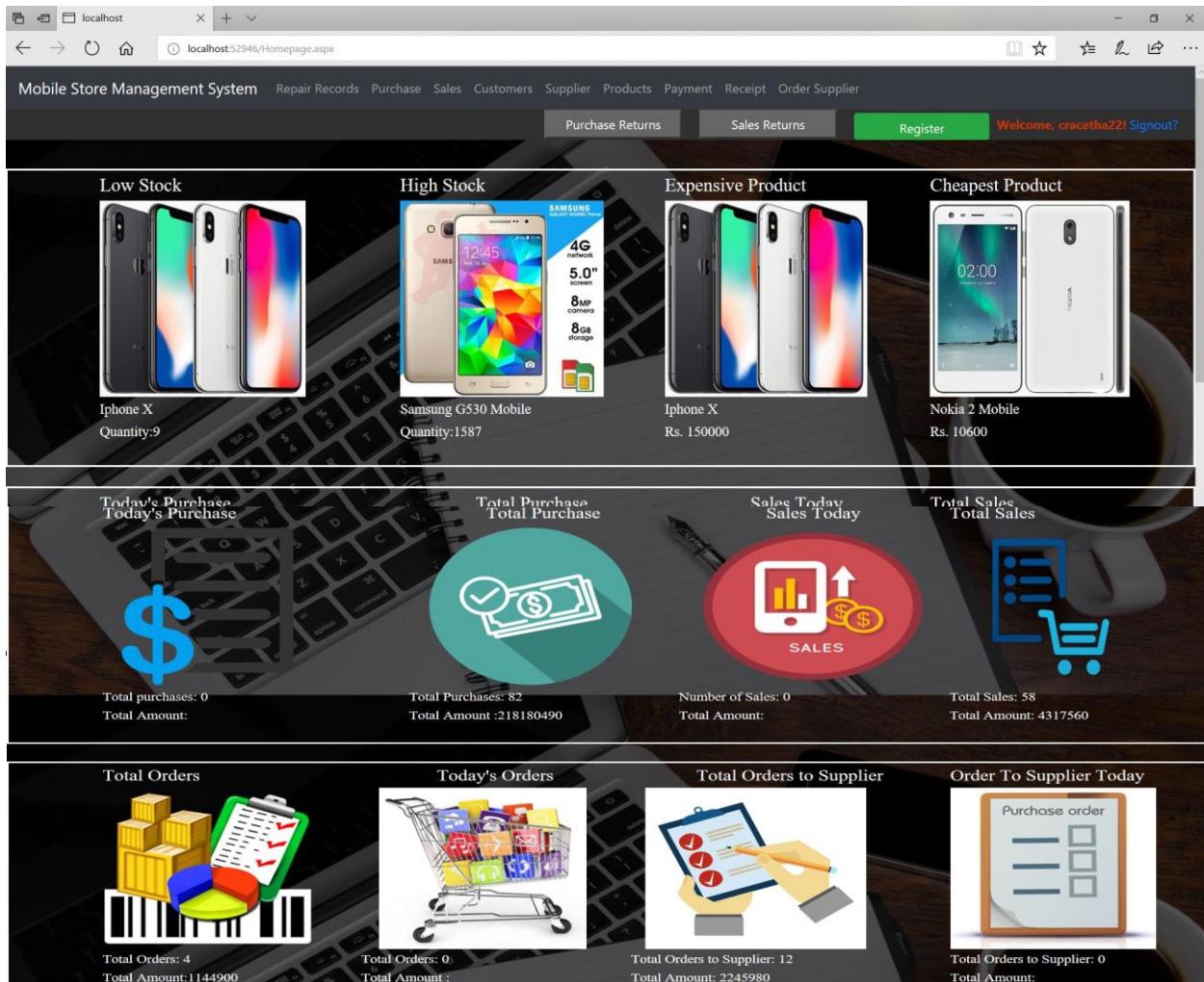
Satisfied Functionality:

- The password of the users can be recovered with the help of recovery key.

Form Items:

- Select Button: It retrieves the recovery key of selected user in the drop down to an invisible textbox on the form.
- Check Button: It compares the recovery key value to the retrieved key. The field for entering new password is set to be visible if the recovery key is correct.

Design 5: Homepage:



Form Usage:

Homepage provides links to all of the different forms and provides some essential information of the system to the users.

Satisfied Functionalities:

- Notifies the user with products with low stock, high stock, expensive and cheapest product.
- Notifies the user about the total number of purchase and sales of a current day or as in total till date.
- Notifies the user about the total number of orders from customers and orders made to supplier of a current day or as in total till date.
- The system doesn't require any input for the calculation of these entities.

Form Items:

- Register Button: This button redirects the system to Registration page. This button is only visible if the user level is 1.
- Purchase Returns: This button redirects the system to purchase returns page. This button is only visible if the user level is 1.
- Sales Return: This button redirects the system to the sales returns page. This button is also only visible if the user level is 1.
- Welcome, @user: The link redirects the system to the password recovery page.
- Repair Records: This link redirects to the Repair Records page.
- Purchase: This link redirects to the Purchase products page of the system.
- Sales: This link redirects to the Sales products page of the system.
- Customer: This link redirects to the Customers page of the system.
- Supplier: This Link redirects to the Supplier page of the system.
- Products: This Link redirects to the Add Product page of the system.
- Payment: This Link redirects to the Payment to Suppliers page of system.
- Receipt: This link redirects to the Receive payment page of the system.
- Order Supplier: This link redirects to the order supplier page of the system.

Design 6: Repair Form:

The screenshot shows a web browser window for the 'Repair Records Page' at 'localhost:52946/Form_Repair.aspx'. The page title is 'Mobile Store Management System'. The main content area has a header 'Enter Details For Repair Service!' and a form with fields for Customer Name (dropdown menu), Address, Phone Model, IMEI Number, Problem Description (text area), Given Date (date input), and Select Image (file input). There are 'Get Info' and 'Customer Name: Customer ID' buttons. A 'Save' button is at the bottom. Below the form is a table with columns: Id, Customer Name, Customer ID, Address, Phone Model, IMEI Number, Problem, Date, and Image. The first row of the table shows a profile picture of a man.

Id	Customer Name	Customer ID	Address	Phone Model	IMEI Number	Problem	Date	Image
								

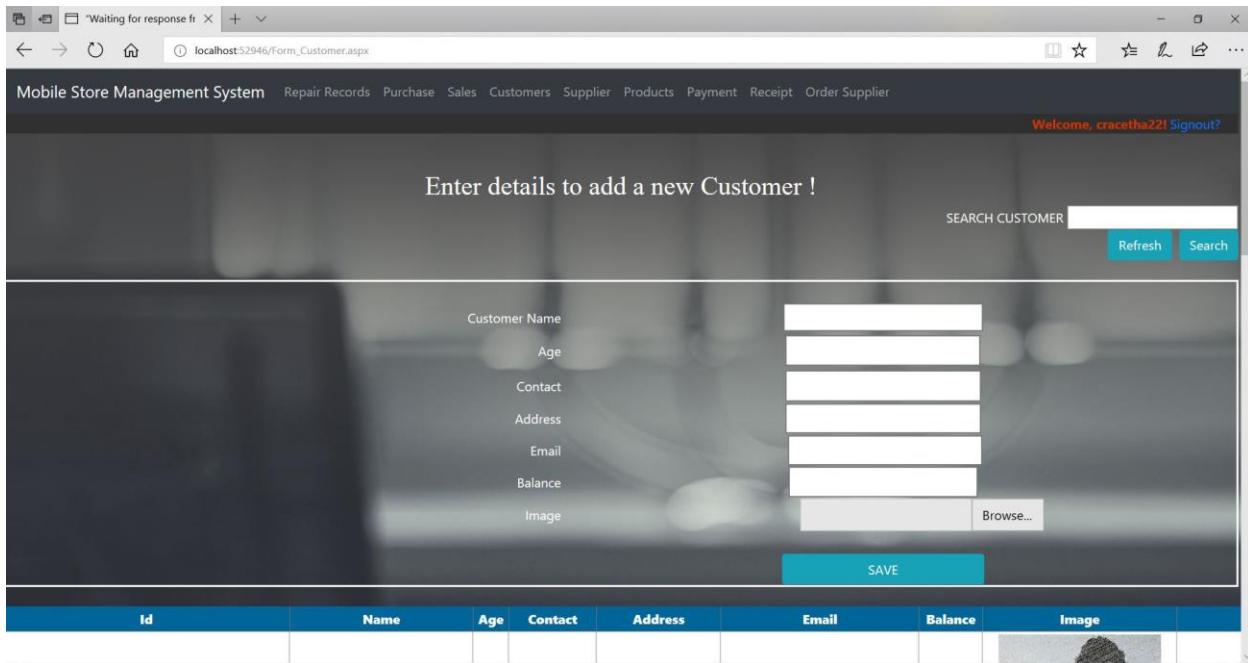
Form Usage: Registering repair record of the phones of the customers.

Satisfied Functionalities:

- Storage of repair records with IMEI number.
- The UI is simple and easy to understand.

Form Items:

- Navbar: The contents of the navbar are already explained in the home design.
- Get info button: This button retrieves all the necessary information of the selected id of customers from the customers table and sets the value to the respective fields in the form.
- Browse button: This button is a file upload button. This button is used to upload image files into the system.
- Save button: This button is used to store all the provided information into the repair table of the system.

Design 7: Customers:

Form usage: The customer form is used to register new customers in the system.

Satisfied Functionality:

- Ability to search customers by name.
- Image uploading of customers.
- Simple form

Form items:

- Navbar: The contents of the navbar are already explained in the homepage section.
- Search button: This button is set to match the name of customers entered into the textbox to the customers registered into the database.
- Refresh button: The button is used to refresh the customer page.
- Browse button: This button is used to browse image files.

- Save button: This button is used to save all the provided information into the customers table.

Design 8: Products:

Product ID	Name	Quantity	Rate	Sellingrate	Image
398da4f9-45f5-4aa4-8280-ef677d08485a	Samsung Galaxy S9 Mobile	1483	99990	120000	

Form usage: This form is used to register new products into the system.

Satisfied functionalities:

- Registering of new products into the system.
- Requires less input from the user.

Form Elements:

- Navbar: The elements of the navbar are already explained in the homepage section.
- Browse button: This button is used to browse image files of the device.
- Add button: This button is used to save all the entered information into the total product table of the system.

Design 9: Supplier:

The screenshot shows a web application window titled 'Mobile Store Management System'. The URL in the address bar is 'localhost:52946/Form_Supplier.aspx'. The page header includes a welcome message 'Welcome, cracetha22! Signout?' and a navigation menu with links: Repair Records, Purchase, Sales, Customers, Supplier, Products, Payment, Receipt, and Order Supplier.

The main content area is titled 'Enter Supplier Details'. It contains five input fields labeled 'Supplier Name', 'Address', 'Contact', 'Email', and 'Balance'. Below these fields is a blue 'Save' button. At the bottom of the page, there is a table with a single row showing columns for 'Id', 'Suppliername', 'Address', 'Contact', 'Email', and 'Balance'.

Form Usage: This form is used to store the supplier information into the system.

Satisfied Functionality:

- Storage of necessary information of different suppliers.
- Less time consuming.
- Simple but attractive.

Form Items:

- Navbar: The contents of the navbar are already explained in the homepage section.
- Save button: The save button is used to insert all the entered supplier information in the supplier table.

Design 10: Purchase:

Form Usage: This form is used to purchase different items into the system from different suppliers.

Satisfied Functionality:

- Update of product quantity through the purchase process.
- Cash purchase and credit purchase.

From Elements:

- Navbar: The contents of the navbar are already explained in the homepage section.
- Button Purchase Id: This button is used to control the purchase id on the form.
- Button Get info: This button is used to retrieve rate and quantity information from the product table to the purchase page.
- Add item button: This button is used to add the selected item in the grid view.
- Payment on cash radio button: This button is used to specify the purchase is being done on cash basis.
- Payment on credit radio button: This button is used to specify the purchase is being done on credit basis.
- Purchase button: This button is used to store the entered information into the purchase database and update the product quantity in the total product table.

Design 11: Sales:

The screenshot shows a web-based application for managing mobile store sales. The main title is 'Enter details for SALES!'. The form includes fields for Sales ID (with a dropdown for 'New Sales Id' containing 'Aabish Shrestha'), Customer Name (dropdown), Sales Date (text input), Item (dropdown containing 'Samsung Galaxy S9 Mobile'), Item Purchased Rate (text input), Available Quantity (text input), Selling Quantity (text input), Selling Rate (text input), Discount in % (text input), and Remaining Quantity (text input). There are also buttons for 'Get Sales ID', 'Get Info', 'Select Item', 'Calculate', 'Add Item', and 'Make Sales'.

Form Usage: This form is used to sell the products to the registered customers of the system.

Satisfied functionality:

- Storage of sales records.
- Calculation of discount.
- Validation of product quantity.
- Payment on cash and credit.

Form Items:

- Navbar: The contents of the navbar are already explained in the homepage section.
- Button Sales Id: This button is used to control sales id on the form.
- Button Get info: This button is used to retrieve customer information from the customer table to respective form elements.
- Select item: This button is used to retrieve selected product information from the total product table to the respective form elements.
- Calculate button: This button is used to calculate the given amount of discount on the selling rate of product.
- Add item: This button is used to store the sales information into the sales grid view.
- Payment on cash radio button: This button is used to confirm that the sale is being done on cash basis.
- Payment on credit radio button: This button is used to confirm that the sale is being done on credit basis.
- Make sales: This button is used to confirm the sales item in the sales data grid view and update the balance and quantity of customers and products.

Design 12: Order Supplier:

Order ID	Date	Product ID	Product Name	Quantity	Rate	Amount	Supplier ID	Operations
6a1fc87a-a339-4f64-9c33-038c966b9258	4/29/2018 12:00:00 AM	d54af377-1085-4386-a3c5-f80544d5b163	Nokia 6 Mobile	10	24000	240000	a3a0ff39-17e4-48dc-bca6-2792a1648d70	Edit Delete
898f2cb4-d801-4800-aeb9-63ae42eff8e6	4/29/2018 12:00:00 AM	94e3d15d-52a4-47cc-8efc-9efe1488e0f7	Samsung G530 Mobile	10	11500	115000	a3a0ff39-17e4-48dc-bca6-2792a1648d70	Edit Delete

Page Usage: This form is used to see the orders made by the customers and make necessary orders to the suppliers.

Satisfied Functionality:

- Registration of different orders made to the suppliers.
- View of orders made by customers.

Form Elements:

- Navbar: The contents of the navbar are already explained in the homepage section.
- Get Order Id button: This button is used to control order id in the order supplier form.
- Get name: This button is used to retrieve supplier name from the supplier table to name label in the form in accordance to the selected supplier id in the dropdown.
- Get Rate: This button is used to retrieve rate from the total product.
- Add order: This button is used to save all the entered information into the order supplier table.

Design 13: Password Change:

The screenshot shows a web browser window for 'localhost' at port 52946, displaying the page 'Form_Pwchange.aspx'. The title bar says 'Mobile Store Management System'. The top navigation menu includes 'Repair Records', 'Purchase', 'Sales', 'Customers', 'Supplier', 'Products', 'Payment', 'Receipt', and 'Order Supplier'. A welcome message 'Welcome, cracetha22! Signout?' is visible. The main content area has a heading 'Enter Your New Password or Recoverykey to Continue'. It contains two sections: 'Password Change' and 'Change Recovery Key'. In the 'Password Change' section, there are four input fields: 'Current Password' (containing '00292'), 'New Password', 'Confirm Password', and a 'Save' button. In the 'Change Recovery Key' section, there are four input fields: 'Current Recovery Key' (containing '12345'), 'New Recovery key', 'Confirm Recovery Key', and another 'Save' button. The background of the form features a faint image of a smiling person.

Form usage: This form is used to change the password and recovery key of the registered users.

Satisfied Functionality:

- Change of password of users.
- Change of Recovery key of users.
- The form is simple easy to understand and use.

Form elements:

- Navbar: The contents of the navbar are already explained in the homepage section.
- Save button (Left): This button is used to update the current password of the user with the newly entered password.
- Save button (right): This button is used to update the current recovery key to new recovery key.

3.5.4. Data Design:

The data design is concerned with the storage and representation of data in the system.

Login Design:

The screenshot shows the SQL Server Management Studio (SSMS) interface. The top pane displays the table structure for 'dbo.Login'. The bottom pane shows the T-SQL script for creating the 'Login' table.

Table Structure (dbo.Login):

Name	Data Type	Allow Nulls	Default
Id	varchar(50)	■	
Username	varchar(50)	■	
Password	varchar(50)	■	
Firstname	nchar(10)	■	
Lastname	nchar(10)	■	
Email	varchar(50)	■	
Phone	bigint	■	
userlevel	int	■	
Recovery	varchar(50)	■	

Keys (1):
<unnamed> (Primary Key, Clustered: Id)

Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

T-SQL Script:

```

1 CREATE TABLE [dbo].[Login] (
2     [Id]      VARCHAR (50) NOT NULL,
3     [Username] VARCHAR (50) NOT NULL,
4     [Password] VARCHAR (50) NOT NULL,
5     [Firstname] NCHAR (10) NOT NULL,
6     [Lastname] NCHAR (10) NOT NULL,
7     [Email]    VARCHAR (50) NOT NULL,
8     [Phone]    BIGINT NOT NULL
)

```

Customers:

The screenshot shows the SQL Server Management Studio (SSMS) interface. The top pane displays the table structure for 'dbo.customers'. The bottom pane shows the T-SQL script for creating the 'customers' table.

Table Structure (dbo.customers):

Name	Data Type	Allow Nulls	Default
Id	varchar(50)	■	
Name	varchar(50)	■	
Age	nchar(10)	■	
Contact	nchar(10)	■	
Address	varchar(50)	■	
Email	varchar(50)	■	
Balance	int	■	
Image	varchar(MAX)	■	

Keys (1):
<unnamed> (Primary Key, Clustered: Id)

Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

T-SQL Script:

```

1 CREATE TABLE [dbo].[customers] (
2     [Id]      VARCHAR (50) NOT NULL,
3     [Name]    VARCHAR (50) NOT NULL,
4     [Age]     NCHAR (10) NOT NULL,
5     [Contact] NCHAR (10) NOT NULL,
6     [Address] VARCHAR (50) NOT NULL,
7     [Email]   VARCHAR (50) NOT NULL,
)

```

Supplier:

The screenshot shows the SQL Server Management Studio interface. On the left, the 'Design' tab is selected, displaying the schema of the `dbo.Supplier` table. The table has six columns: `Id`, `Suppliername`, `Address`, `Contact`, `Email`, and `Balance`. All columns are of type `varchar(50)` except for `Contact` which is `bigint`. The `Allow Nulls` column contains several small square checkboxes. The `Default` column is empty. On the right, the properties pane shows:

- Keys (1)**: A primary key named <unnamed> (Primary Key, Clustered: `Id`)
- Check Constraints (0)**
- Indexes (0)**
- Foreign Keys (0)**
- Triggers (0)**

Below the design view, the 'T-SQL' tab is selected, showing the `CREATE TABLE` script:

```

1 CREATE TABLE [dbo].[Supplier] (
2     [Id]         VARCHAR (50) NOT NULL,
3     [Suppliername] VARCHAR (50) NOT NULL,
4     [Address]    VARCHAR (50) NOT NULL,
5     [Contact]    BIGINT      NOT NULL,
6     [Email]      VARCHAR (50) NOT NULL,
7     [Balance]    INT         NOT NULL,
PRIMARY KEY CLUSTERED ([Id] ASC)

```

Total Product:

The screenshot shows the SQL Server Management Studio interface. On the left, the 'Design' tab is selected, displaying the schema of the `dbo.Totalproduct` table. The table has six columns: `PId`, `Name`, `Quantity`, `Rate`, `Sellingrate`, and `Image`. `PId` is of type `varchar(50)`, while the others are `int` or `nvarchar(MAX)`. The `Allow Nulls` column contains several small square checkboxes. The `Default` column is empty. On the right, the properties pane shows:

- Keys (1)**: A primary key named <unnamed> (Primary Key, Clustered: `PId`)
- Check Constraints (0)**
- Indexes (0)**
- Foreign Keys (0)**
- Triggers (0)**

Below the design view, the 'T-SQL' tab is selected, showing the `CREATE TABLE` script:

```

1 CREATE TABLE [dbo].[Totalproduct] (
2     [PId]        VARCHAR (50) NOT NULL,
3     [Name]       VARCHAR (50) NOT NULL,
4     [Quantity]   INT        NOT NULL,
5     [Rate]       INT        NOT NULL,
6     [Sellingrate] INT        NOT NULL,
7     [Image]      NVARCHAR (MAX) NOT NULL,
PRIMARY KEY CLUSTERED ([PId] ASC)

```

Repair:

Script File: dbo.Repair.sql*

Name	Data Type	Allow Nulls	Default
[Id]	nvarchar(50)	■	
Customername	varchar(50)	■	
CustomerId	varchar(50)	■	
Address	varchar(50)	■	
Phonemodel	varchar(50)	■	
Imei	bigint	■	
Problem	nvarchar(MAX)	■	
Date	date	■	
Image	varchar(MAX)	■	

Keys (1)
<unnamed> (Primary Key, Clustered: Id)

Check Constraints (0)

Indexes (0)

Foreign Keys (0)

Triggers (0)

Design T-SQL

```

1 CREATE TABLE [dbo].[Repair] (
2     [Id] NVARCHAR (50) NOT NULL,
3     [Customername] VARCHAR (50) NOT NULL,
4     [CustomerId] VARCHAR (50) NOT NULL,
5     [Address] VARCHAR (50) NOT NULL,
6     [Phonemodel] VARCHAR (50) NOT NULL,
7     [Imei] BIGINT NOT NULL,
8     [Problem] NVARCHAR (MAX) NOT NULL
9 )

```

Purchases:

Script File: dbo.Purchase.sql*

Name	Data Type	Allow Nulls	Default
[Id]	varchar(50)	■	
Date	date	■	
Sname	varchar(50)	■	
Pname	varchar(50)	■	
Quantity	int	■	
Rate	int	■	
Amount	int	■	

Keys (0)

Check Constraints (0)

Indexes (0)

Foreign Keys (0)

Triggers (0)

Design T-SQL

```

1 CREATE TABLE [dbo].[Purchase] (
2     [Id] VARCHAR (50) NOT NULL,
3     [Date] DATE NOT NULL,
4     [Sname] VARCHAR (50) NOT NULL,
5     [Pname] VARCHAR (50) NOT NULL,
6     [Quantity] INT NOT NULL,
7     [Rate] INT NOT NULL,
8     [Amount] INT NOT NULL
9 )

```

Sales:

The screenshot shows the SQL Server Management Studio interface. The top pane displays the table structure for 'dbo.Sales' with columns: Id, Name, CID, Item, Date, Purchaserate, Salesrate, Sellingquantity, and Total. The bottom pane shows the T-SQL script for creating the 'Sales' table.

Name	Data Type	Allow Nulls	Default
Id	varchar(50)	■	
Name	varchar(50)	■	
CID	varchar(50)	■	
Item	varchar(50)	■	
Date	datetime	■	
Purchaserate	int	■	
Salesrate	int	■	
Sellingquantity	int	■	
Total	int	■	

```

1 CREATE TABLE [dbo].[Sales] (
2     [Id]           VARCHAR (50) NOT NULL,
3     [Name]          VARCHAR (50) NOT NULL,
4     [CID]          VARCHAR (50) NOT NULL,
5     [Item]          VARCHAR (50) NOT NULL,
6     [Date]          DATETIME    NOT NULL,
7     [Purchaserate] INT          NOT NULL,
8     [Salesrate]     INT          NOT NULL,
9     [Sellingquantity] INT          NOT NULL,
10    [Total]         INT          NOT NULL
)

```

Orders:

The screenshot shows the SQL Server Management Studio interface. The top pane displays the table structure for 'dbo.Orders' with columns: Id, OrderDate, CustomerName, OrderingProduct, PID, Quantity, Rate, Total, and CID. The bottom pane shows the T-SQL script for creating the 'Orders' table. It also lists two foreign key constraints: FK_Orders_ToTable (PID) and FK_Orders_ToTable_1 (Id).

Name	Data Type	Allow Nulls	Default
Id	varchar(50)	■	
OrderDate	date	■	
CustomerName	varchar(50)	■	
OrderingProduct	varchar(50)	■	
PID	varchar(50)	■	
Quantity	nchar(10)	■	
Rate	int	■	
Total	int	■	
CID	varchar(50)	■	

```

1 CREATE TABLE [dbo].[Orders] (
2     [Id]           VARCHAR (50) NOT NULL,
3     [OrderDate]    DATE        NOT NULL,
4     [CustomerName] VARCHAR (50) NOT NULL,
5     [OrderingProduct] VARCHAR (50) NOT NULL,
6     [PID]          VARCHAR (50) NOT NULL,
7     [Quantity]     NCHAR (10)  NOT NULL,
8     [Rate]          INT         NOT NULL,
9     [Total]         INT         NOT NULL
)

```

Order Supplier:

The screenshot shows the SQL Server Management Studio (SSMS) interface. The top pane displays the table structure for 'dbo.Ordersupplier' with columns: Name, Data Type, Allow Nulls, and Default. The bottom pane shows the T-SQL script for creating the table:

```

1 CREATE TABLE [dbo].[Ordersupplier] (
2     [OID]          VARCHAR (50) NOT NULL,
3     [Date]         DATE        NOT NULL,
4     [PID]          VARCHAR (50) NOT NULL,
5     [Pname]        VARCHAR (50) NOT NULL,
6     [Quantity]    INT         NOT NULL,
7     [Rate]         INT         NOT NULL,

```

Purchase Returns:

The screenshot shows the SQL Server Management Studio (SSMS) interface. The top pane displays the table structure for 'dbo.Purchasereturns' with columns: Name, Data Type, Allow Nulls, and Default. The bottom pane shows the T-SQL script for creating the table:

```

1 CREATE TABLE [dbo].[Purchasereturns] (
2     [PRId]          VARCHAR (50) NOT NULL,
3     [Date]          DATE        NOT NULL,
4     [PID]           VARCHAR (50) NOT NULL,
5     [Pname]         VARCHAR (50) NOT NULL,
6     [Quantity]     INT         NOT NULL,
7     [Rate]          INT         NOT NULL,

```

Sales Returns:

Update Script File: dbo.Salesreturns.sql

Name	Data Type	Allow Nulls	Default
SRId	varchar(50)	<input checked="" type="checkbox"/>	
Date	date	<input checked="" type="checkbox"/>	
SId	varchar(50)	<input checked="" type="checkbox"/>	
Pname	varchar(50)	<input checked="" type="checkbox"/>	
Quantity	int	<input checked="" type="checkbox"/>	
Rate	int	<input checked="" type="checkbox"/>	
CId	varchar(50)	<input checked="" type="checkbox"/>	
Amount	int	<input checked="" type="checkbox"/>	

Keys (1)
 <unnamed> (Primary Key, Clustered: SRId)
 Check Constraints (0)
 Indexes (0)
 Foreign Keys (0)
 Triggers (0)

Design T-SQL

```
1 CREATE TABLE [dbo].[Salesreturns] (
2     [SRId]      VARCHAR (50) NOT NULL,
3     [Date]       DATE        NOT NULL,
4     [SId]        VARCHAR (50) NOT NULL,
5     [Pname]      VARCHAR (50) NOT NULL,
6     [Quantity]   INT         NOT NULL,
7     [Rate]       INT         NOT NULL,
8     [CId]        VARCHAR (50) NOT NULL,
```

Receipt:

Update Script File: dbo.Purchase.sql*

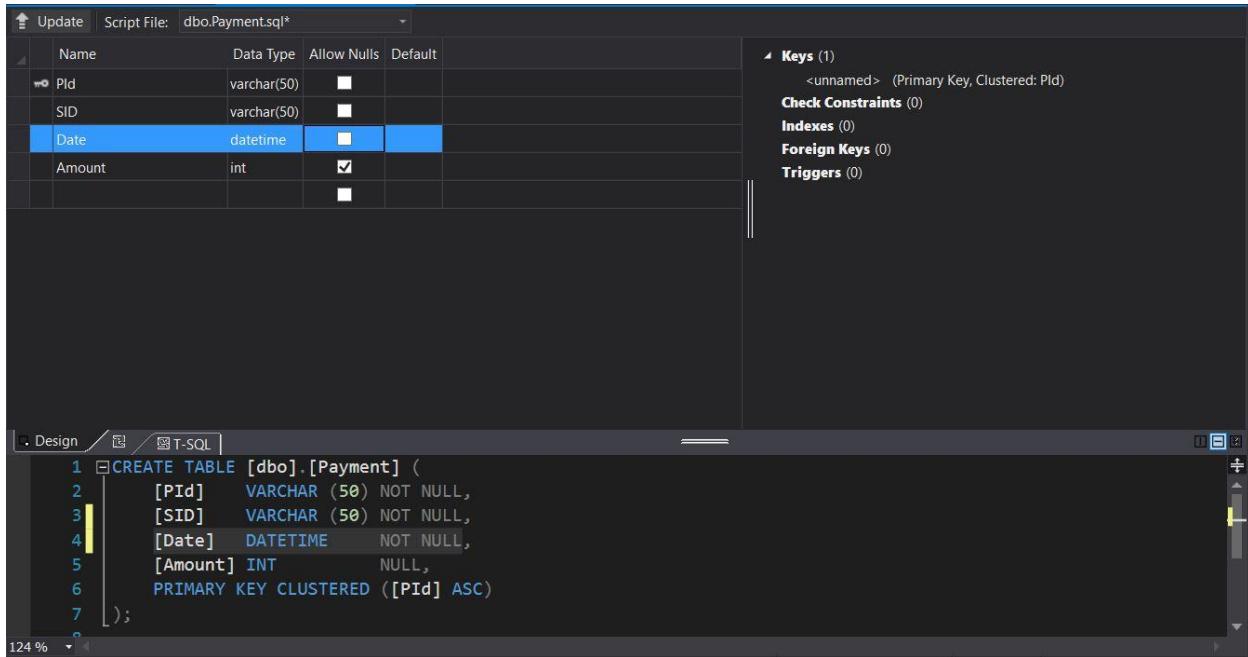
Name	Data Type	Allow Nulls	Default
Id	varchar(50)	<input checked="" type="checkbox"/>	
Date	date	<input checked="" type="checkbox"/>	
Sname	varchar(50)	<input checked="" type="checkbox"/>	
Pname	varchar(50)	<input checked="" type="checkbox"/>	
Quantity	int	<input checked="" type="checkbox"/>	
Rate	int	<input checked="" type="checkbox"/>	
Amount	int	<input checked="" type="checkbox"/>	

Keys (0)
 Check Constraints (0)
 Indexes (0)
 Foreign Keys (0)
 Triggers (0)

Design T-SQL

```
1 CREATE TABLE [dbo].[Purchase] (
2     [Id]        VARCHAR (50) NOT NULL,
3     [Date]      DATE        NOT NULL,
4     [Sname]     VARCHAR (50) NOT NULL,
5     [Pname]     VARCHAR (50) NOT NULL,
6     [Quantity]  INT         NOT NULL,
7     [Rate]      INT         NOT NULL,
```

Payment:



3.5.5. The coding part:

Coding is a very important part of a system's physical development. In the coding process, different codes are written to create functionality in the system. The data and information can be retrieved, validated and stored in the database through coding process.

Login:

```

{
    con.Open();
    string checkpass = "select Password from Login where Username = '" + txtUname.Text + "'";
    SqlCommand comd = new SqlCommand(checkpass, con);
    string pass = comd.ExecuteScalar().ToString().Replace(" ", "");
    con.Close();
    if (pass == txtPass.Text)
    {
        Session["new"] = txtUname.Text;
        Response.Write("Password is correct");
        Response.Redirect("Homepage.aspx");
    }
    else
    {
        lblwarning.Visible = true;
        lblwarning.Text = "Password is incorrect";
    }
}

```

In this coding part, the registered password is being retrieved and compared to the valued entered by the user in txtPass textbox. A new session is also being created in case of successful login.

Registration:

```

string insertuser =
    "INSERT into [Login] (Id,Username,Password,Firstname,Lastname,Email,Phone,userlevel,Recovery) values (@Id,@Username,@Password,
SqlCommand cmd = new SqlCommand(insertuser, con);
cmd.Parameters.AddWithValue("@Id", Id.ToString());
cmd.Parameters.AddWithValue("@Username", txt_username.Text);
cmd.Parameters.AddWithValue("@Password", txt_password.Text);
cmd.Parameters.AddWithValue("@Firstname", txt_fname.Text);
cmd.Parameters.AddWithValue("@Lastname", txt_lname.Text);
cmd.Parameters.AddWithValue("@Email", txt_email.Text);
cmd.Parameters.AddWithValue("@Phone", txt_phone.Text);
cmd.Parameters.AddWithValue("@userlevel", ddlist.Text);
cmd.Parameters.AddWithValue("@Recovery", txt_recovery.Text);
cmd.ExecuteNonQuery();
labelwarning.Text = "The User has been registered successfully";
labelwarning.ForeColor = System.Drawing.Color.White;
labelwarning.BackColor = System.Drawing.Color.Green;

```

In this coding part, the values entered by the user are being stored in the login table through the registration page. The attributes are defined with proper parameters and their respective values are also defined and the insert query is executed.

Customers:

```

if (FileUpload1.HasFile)
{
    Guid id = Guid.NewGuid();
    string str = FileUpload1.FileName;
    FileUpload1.PostedFile.SaveAs(Server.MapPath("~/Images/" + str));
    string Name = txt_name.Text;
    string age = txt_age.Text;
    string contact = txt_contact.Text;
    string address = txt_address.Text;
    string email = txt_email.Text;
    string balance = txt_balance.Text;
    string Image = "~/Images/" + str.ToString();
    SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["MobileStoreCONNECTIONSTRING"].ConnectionString);
    SqlCommand cmd = new SqlCommand("INSERT INTO customers VALUES(@Id,@Name,@Age,@Contact,@Address,@Email,@Balance,@Image);", con);
    cmd.Parameters.AddWithValue("@Id", id.ToString());
    cmd.Parameters.AddWithValue("@Name", Name);
    cmd.Parameters.AddWithValue("@Age", age);
    cmd.Parameters.AddWithValue("@Contact", contact);
    cmd.Parameters.AddWithValue("@Address", address);
    cmd.Parameters.AddWithValue("@Email", email);
    cmd.Parameters.AddWithValue("@Balance", balance);
    cmd.Parameters.AddWithValue("@Image", Image);
    con.Open();
    cmd.ExecuteNonQuery();
    GridView1.DataBind();
    con.Close();
    clear();
    lblSucces.Visible = true;
    lblSucces.ForeColor = System.Drawing.Color.ForestGreen;
}

```

In this coding part of customers form, the file upload is given condition of having file or else. If it is having a file then the insert command will be initiated in the customer table with the given values and parameters.

Supplier:

```

protected void btnyes_Click(object sender, EventArgs e)
{
    Session["new"] = "";
    Response.Redirect("Form_Login.aspx");
}
SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["MobileStoreCONNECTIONSTRING"].ConnectionString);
protected void Button1_Click(object sender, EventArgs e)
{
    if (txt_name.Text == "" || txt_address.Text == "" || txt_contact.Text == "" || txt_email.Text == "" || txt_balance.Text == "")
    {
        lblWarning.Visible = true;
        lblWarning.Text = "Fields Cannot be Left Empty";
        lblWarning.ForeColor = System.Drawing.Color.Red;
    }
    else
    {
        try
        {

            con.Open();
            Guid Id = Guid.NewGuid();
            string insertsupp =
                "INSERT into [Supplier] (Id,Suppliername,Address,Contact,Email,Balance) values (@Id,@Suppliername,@Address,@Co
            SqlCommand cmd = new SqlCommand(insertsupp, con);
            cmd.Parameters.AddWithValue("@Id", Id.ToString());

```

In this coding section of supplier page, the respective fields of the pages are checked if they are empty or not. Display of error message is set in the code in case of empty fields in the form and initiation of the insert command into the supplier table is done if the fields are not empty.

Homepage:

```

if (Session["new"] != null)
{
    string username = Session["new"].ToString();
    con.Open();
    SqlDataAdapter da = new SqlDataAdapter("Select userlevel from [Login] where Username='" + username + "'", con);
    DataTable dt = new DataTable();
    con.Close();
    da.Fill(dt);
    int userlevel = Convert.ToInt32(dt.Rows[0][0].ToString());
    if (userlevel == 1)
    {

        Btnpr.Visible = true;
        Btnpr0.Visible = true;
        btnregister.Visible = true;

    }
    else
    {

        Btnpr.Visible = false;
        Btnpr0.Visible = false;
        btnregister.Visible = false;
    }
}
else
{
    Response.Redirect("Form_Login.aspx");
}

```

In this coding section of homepage form, session is verified at the first. IF the session is empty, the form is redirected to login form and if the session is not empty, the user level is retrieved from login table in data table. Further the user level is compared with values and different form elements are controlled in accordance of user level.

Sales:

```

if (qty <= Convert.ToInt32(txtAvailableQuantity.Text))
{
    lblkhatra.Visible = true;
    int remainingstock = Convert.ToInt32(txtAvailableQuantity.Text) - qty;

    lblkhatra.ForeColor = System.Drawing.Color.Red;
    try
    {
        con.Open();
        int stockedquantity = Convert.ToInt32(txtAvailableQuantity.Text);
        int newstock = stockedquantity - qty;
        SqlCommand updatecommand = new SqlCommand("Update [Totalproduct] set Quantity =@NESTOCK WHERE PId =@PID",
        updatecommand.Parameters.AddWithValue("@NESTOCK", newstock.ToString());
        updatecommand.Parameters.AddWithValue("@PID", productid);
        updatecommand.ExecuteNonQuery();
        con.Close();

        try
        {
            con.Open();
            SqlCommand insertcommand = new SqlCommand("Insert Into [Sales] values (@Id,@Name,@CID,@Item,@Date,@Pur
            insertcommand.Parameters.AddWithValue("@Id", sid);
            insertcommand.Parameters.AddWithValue("@Name", customerid );
            insertcommand.Parameters.AddWithValue("@CID", txtcname.Text);
            insertcommand.Parameters.AddWithValue("@Item", itemnaam.Text);
            insertcommand.Parameters.AddWithValue("@DATE", txt_date.Text);
        }
        catch (Exception ex)
        {
            lblkhatra.Visible = true;
            lblkhatra.Text = "Inner Exception" + ex.ToString();
            lblkhatra.ForeColor = System.Drawing.Color.Red;
        }
    }
    catch (Exception ex)
    {
        Response.Write(ex.ToString());
    }
}
else
{
    lblkhatra.Visible = true;
    lblkhatra.Text = "Sales Qantity shouldnot be greater than the stocked good";
}

```

In the above coding parts, the quantity entered by the user is being compared with the stock quantity of the system. If the quantity is sufficient, then the purchase process is initiated. The quantity in the total product table is updated with new quantity and the insert command is initiated in the sales table. In case of selling quantity being higher than stocked quantity, the system is set to display an error message. The exception of the query is caught and displayed through a label.

Purchase Returns:

```

protected void btninfo(object sender, EventArgs e)
{
    string purchaseid = ddlpurchaseid.SelectedValue;

    try
    {
        con.Open();
        SqlDataAdapter da = new SqlDataAdapter("Select [Pname],[Quantity],[Rate],[Sname] FROM [Purchase] where [Id] = '" + pur
        DataTable dt = new DataTable();
        da.Fill(dt);
        lblpid.Text = dt.Rows[0][0].ToString();
        lblquantity.Text = dt.Rows[0][1].ToString();
        lblrate.Text = dt.Rows[0][2].ToString();
        lblsname.Text = dt.Rows[0][3].ToString();
        dt.Clear();
        con.Close();
    }
    catch (Exception ex)
    {
        lblkhatra.Text = ex.ToString();
    }
}

try
{
    if (totalqty < returnqty)
    {
        lblkhatra.Text = "Not enough product quantity in the stock!";
        lblkhatra.Visible = true;
        lblkhatra.ForeColor = System.Drawing.Color.Red;
    }
    else if (returnqty <= Convert.ToInt32(lblquantity.Text))

    {

        con.Open();

        int newstock = totalqty - returnqty;
        SqlCommand updatecommand = new SqlCommand("Update [Totalproduct] set Quantity =@NEWSTOCK WHERE Name =@PID", co
        updatecommand.Parameters.AddWithValue("@NEWSTOCK", newstock.ToString());
        updatecommand.Parameters.AddWithValue("@PID", pname);
        updatecommand.ExecuteNonQuery();
        con.Close();
        con.Open();
        SqlCommand updatecommandd = new SqlCommand("Update [Supplier] set Balance =@NEWBALANCE WHERE Suppliername =@Sn
        updatecommandd.Parameters.AddWithValue("@NEWBALANCE", remainingbalance.ToString());
        updatecommandd.Parameters.AddWithValue("@Sname", sname);
        updatecommandd.ExecuteNonQuery();
        con.Close();
        try
        {

```

In this coding part of Purchase returns form, the product name, quantity, rate, and supplier name are being retrieved in a data table from purchase table in accordance to the purchase id. The retrieved data are being transferred to labels and textboxes in accordance to their respective index numbers in data table. Further the return quantity is being compared with the stocked quantity and error message is set to display in case of insufficient stocked quantity and quantity of the total product is being updated to the new calculated stock. The balance of the supplier table is also being updated with the remaining balance.

Chapter 4: System Testing & Implementation:

4.1. System Testing:

Testing is one of the most important parts of software life cycle. Proper testing helps to provide better quality software to the final users. This process is done to make sure that the end users won't have to face some software issues. Functionality of software is validated through the testing process. Testing process provides the following information:

- It finds bugs and errors of the system that can be fixed before the finalization of the system.
- It helps to create bug free software for the final delivery.
- Assurance of reliability and stability of software in different test conditions.

Every developed system must be tested with different methods. Without proper testing, a system with many bugs and errors would be delivered to the final user. Delivery of such faulty application will decrease credibility, and the customers will be very unsatisfied with the system. The testing process is generally conducted by quality assurance and development teams. The functionality of the system is validated through this type of testing. Different types of testing methods are performed in order to validate a specific module of the system. Database of the system is the main module, therefore it possesses main role in the functionality of the system.

4.1.1. Testing Types:

There are mainly two types of Software testing methodologies. They are as follows:

1. White Box testing and
2. Black Box testing.

4.1.1.1 White box testing:

Knowledge of internal implementation of the system is required for white box testing method. It is also called “open-box testing” because the implementation of the system is open to the tester. In white box testing, the tester requires specific knowledge of programming codes in order to examine the output. The accuracy of the test is dependent upon the knowledge of tester upon the exact functionality of the system. Functional tests can be considered for the process of database testing to validate working functionality.

4.1.1.3. Black box testing:

The black box testing is that kind of method of software testing in which the testing of software is performed without the actual knowledge of internal structure of the program or code. The knowledge of programming is not required to carry out this type of testing. This type of testing is applicable with high level of testing which includes System, Acceptance testing. Functional testing, close box testing, behavior testing etc. are carried out under black box testing.

What kind of testing is important?

There are various methods of testing that we need to apply for the application test. In general, there are 2 different test methods that are needed for database testing. They are Unit Test and Functional Test.

Unit tests: It is one of the types of black box testing. It is used to verify the contracts exposed by the interfaces. To perform a unit test, certain inputs are required, and then the procedures of the system should return accurate outputs from the system. For this process, the applicable procedure is not important but the pass fail criteria must be defined. A unit test is considered as passed if the output generated by the application is valid. In other words, the unit test, tests the ability and capability of interfaces to communicate with the users as specified by their contracts.

Functional Tests:

The actual functionality of the system is verified through the use of functional testing. It is a type of white box testing. It is the software testing process in which, the system is tested in order to ensure that it meets all of its requirements. Functional test conveys many kinds of tests of applications. In this type of test, procedures as well as the output is used to determine the pass or fail criteria. Procedures are needed to be considered for the validation of the application.

4.2. Test Cases:

No.	Type	Cases	Result
1.	Database connection validation	Tesing if the user can order a new item.	Test is Successful!
2.	Database connection validation	Testing if the order process can be carried out with a different connection string.	Test is Successful!
3.	Login Page Validation	Testing if the system redirects to the Homepage in case of successful login.	Test is Successful!

4.	Login Page Validation	Testing if the required validator of the login page works or not.	Test is Successful!
5.	Login Page Validation	Testing if the system displays “unregistered user” after entering a random username.	Test is Successful!
6.	Orders page validation (ADD/Delete)	Customers can add an order or delete an order in the orders page.	Test is Successful!
7.	Homepage Functionality validation	Automatic calculation of total sales and sales today	Test is Successful!
8.	Homepage Functionality validation	Automatic calculation of total purchase and purchases today	Test is Successful!
9.	Homepage Functionality validation	Automatic calculation of total orders from customers, orders made at current day, total orders made to suppliers, and orders made to suppliers at current day.	Test is Successful!
10.	Database connection validation (Repair Records Page)	Testing if the users can successfully register a repair record.	Test is Successful!

11.	Page validation	Testing if the field required validator works or not.	Test is Successful!
12.	Page validation (Delete)	To check if the repair records are deleted or not	Test is Successful!
13.	Database connection validaion	Testing if the user can enter a new product into the system from Products form.	Test is Successful!
14.	Page validation	Testing if the Required validator of the Products page works or not.	Test is Successful!
15.	Page validation	Testing if the imagefile of products can be uploaded other than (jepg,jpg,bmg,png)	Test is Successful!
16.	Database connection validation	Testing if the user can register a new customer into the system from Customers form.	Test is Successful!
17.	Page validation	Testing if the recorded customers can be edited/deleted or not.	Test is Successful!
18.	Page validation	Testing if the required validator in the Customers page works or not.	Test is Successful!

19.	Database connection validation	To check if the user can register a new supplier in the system using Supplier form.	Test is Successful!
20.	Page validation	Testing if the suppliers can be deleted or not and the required and other validations of the Supplier page works or not.	Test is Successful!
21.	Database connection validation	Testing if the user can purchase a product from the purchase page.	Test is Successful!
22.	Database connection validation	Testing if the quantity of total product is updated after successful purchase.	Test is Successful!
23.	Page validation	Testing if the required validator of the form works or not	Test is Successful!
24.	Database connection validation	Testing if the record of sales is recorded in the sales table or not.	Test is Successful!
25.	Database connection validation	Testing if the balance and quantity of customers and total product are updated or not.	Test is Successful!
26.	Database connection validation	Testing if a user can enter a record in the purchase returns table through Purchase Returns form.	Test is Successful!

27.	Validation Testing	Making sure if user cannot return more amount of products than purchased quantity.	Test is Successful!
28.	Validation Testing	Testing if the user cannot return more amount of product than the stock product.	Test is Successful!
29.	Database connection validation	Testing if the user can enter a sales return entry into the sales return table or not.	Test is Successful!
30.	Page validation	Testing if the user can return more amount of product than sold products.	Test is Successful!
31.	Database connection Validation	Testing if the admin can register a new user into a system or not through Registration page	Test is Successful!
32.	Form validation	Testing required and Compare validator of the registration page.	Test is Successful!
33.	Database connection validation	Testing if the user can enter records in Payment table through Payment form.	Test is Successful!
34.	Form validation	Testing if the user can enter an amount to pay that is greater than stored amount to be paid of supplier	Test is Successful!

35	Database connection validation.	Testing if the user can entry payment records of customers from Payment form	Test is Successful!
36.	Page validation	Testing if the user can enter an amount being receive that is greater than stored amount to be paid by customer	Test is Successful!
37.	Database connection validation.	Testing if the user can change their password and recovery key or not	Test is Successful!
38	Form validation	Testing if the compare validator of new password works or not.	Test is Successful!
39.	Database connection validation	Testing if the users can successfully reset their password or not	Test is Successful!
40.	Form validation	Testing if the compare validator compares the recovery key and gain access to update the new password.	Test is Successful!

4.2.1. Output Screen of Test cases:

Test Case 1: Tesing if the user can order a new item.

Please select the ordering product

39a2f3a9-8880-4351-b34b-93f4495f49f1	<input type="button" value="Get Order ID"/>
Order Date	5/10/2018
Customer ID	33444203-1dd1-4f2f-8979-aaba
Ordering Product	Hp Envy Laptop
Rate	94000
Quantity	1
<input type="button" value="ADD"/>	
Item Inserted. Confirm down to complete the order process.	

ID	OrderDate	CustomerName	OrderingProduct	PID	Quantity	Rate	Total	CID
39a2f3a9-8880-4351-b34b-93f4495f49f1	5/10/2018 12:00:00 AM	Aabish Shrestha	Hp Envy Laptop	7beb8b0e-f7c8-4a97-ae79-03c14b586301	1	94000	94000	33444203-1dd1-4f2f-8979-aabac745f7c2

Result: Order record is inserted. Test is successful.

Test case 3: Testing if the system redirects to the Homepage in case of successful login.

The screenshot shows the homepage of the Mobile Store Management System. At the top, there is a navigation bar with links for Repair Records, Purchase, Sales, Customers, Supplier, Products, Payment, Receipt, Order Supplier, Register, and a welcome message for 'cracetha221'. Below the navigation bar, there are four main sections displaying product information:

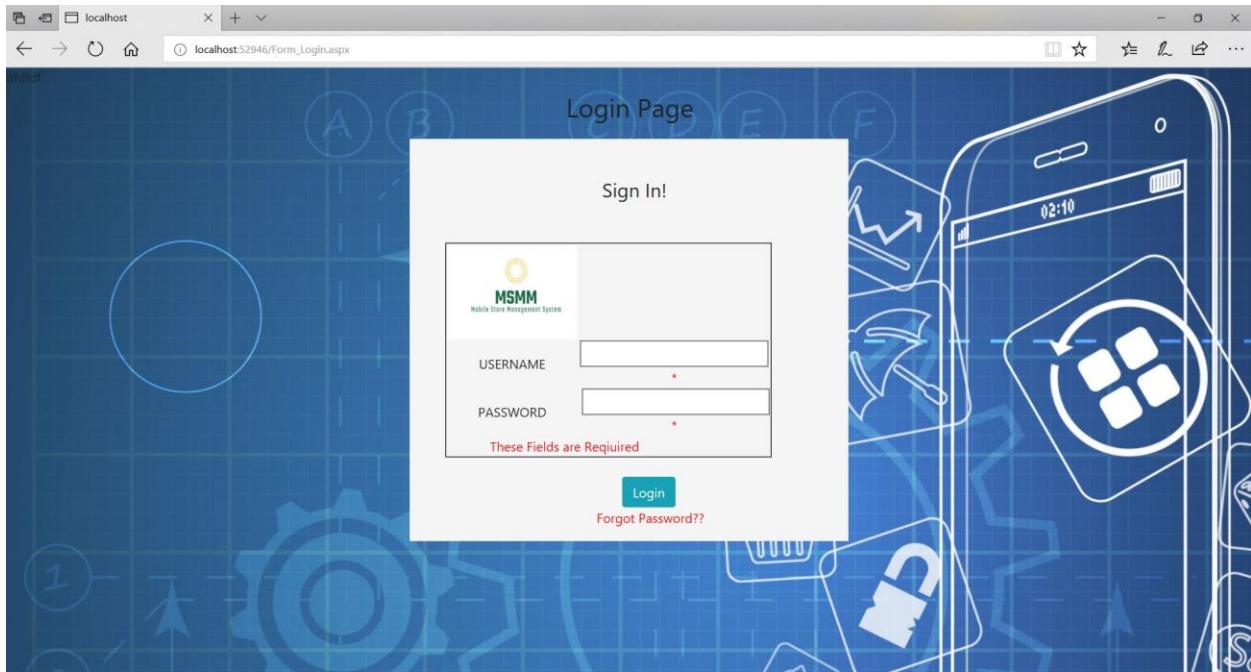
- Low Stock:** Shows three iPhone X phones. Below it, text reads "iPhone X Quantity:9".
- High Stock:** Shows a Samsung G330 mobile phone. Below it, text reads "Samsung G330 Mobile Quantity:1587".
- Expensive Product:** Shows three iPhone X phones. Below it, text reads "iPhone X Rs. 150000".
- Cheapest Product:** Shows two Nokia 2 mobile phones. Below it, text reads "Nokia 2 Mobile Rs. 10600".

Below these sections, there are four cards representing daily metrics:

- Today's Purchase:** Shows a laptop icon.
- Total Purchase:** Shows a money bag icon.
- Sales Today:** Shows a smartphone icon with a graph and coins.
- Total Sales:** Shows a bar chart icon.

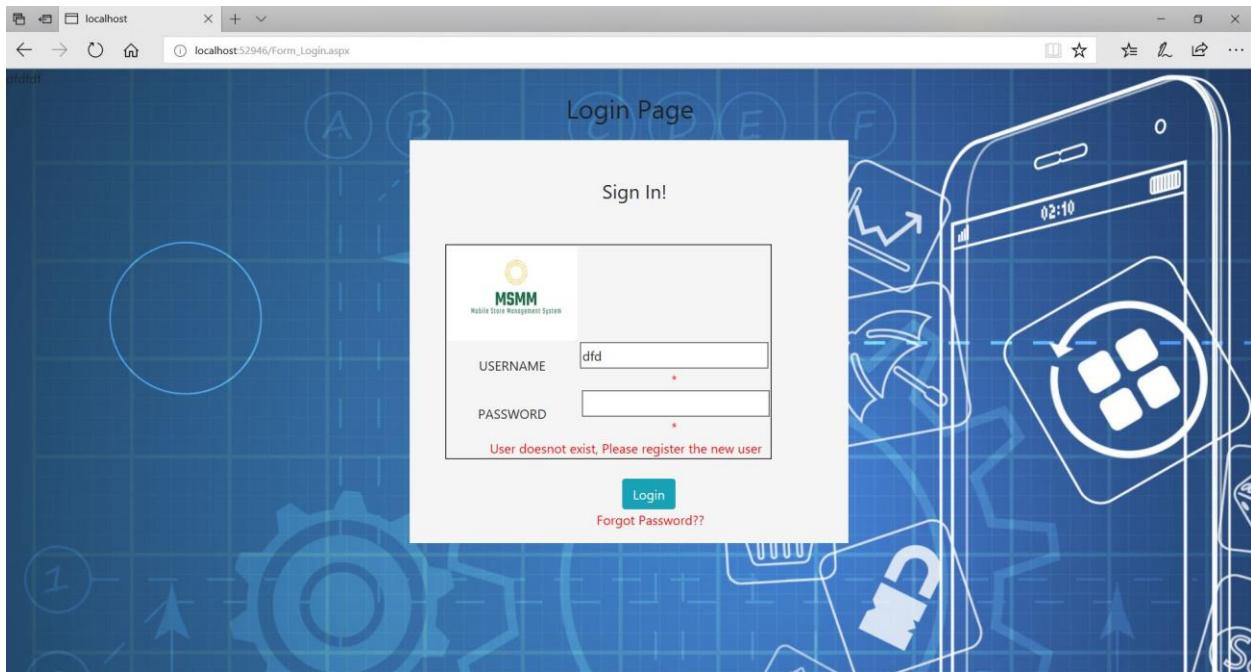
Result: The system is redirected to homepage. Test is successful.

Test case 4: Testing if the required validator of the login page works or not.



Result: Required validator works. Test Successful.

Test case 5: Testing if the system displays “unregistered user” after entering a random username.



Result: The system is displaying the required message. Test successful.

Test case 6: Customers can add an order or delete an order in the orders page.

The screenshot shows a user interface for managing mobile store orders. At the top, there is a form for adding a new item to an order. The fields include 'Rate' (Quantity), '1', and an 'ADD' button. Below this, a message says 'Item Inserted. Confirm down to complete the order process.' A table lists existing order records:

Id	OrderDate	CustomerName	OrderingProduct	PID	Quantity	Rate	Total	CID	Action
c2dfaef7e-e6c5-48c8-b4b1-a533a65d95a9	5/10/2018 12:00:00 AM	Aabish Shrestha	Nokia 3 Mobile	5b9deb73-ad11-4e77-9c56-2c700c395940	1	13500	13500	33444203-1dd1-4f2f-8979-aabac745f7c2	Delete
c2dfaef7e-e6c5-48c8-b4b1-a533a65d95a9	5/10/2018 12:00:00 AM	Suraj Shrestha	Hp Envy Laptop	7beb8b0e-f7c8-4a97-ae79-03c14b586301	1	94000	94000	393c8a5a-5749-443f-9ccb-55f0676920f1	Delete

At the bottom, there is a large image of a keyboard with a 'Confirm Order' button overlaid.

The second part of the screenshot shows the same interface after changes have been made. The table now contains only one record:

Id	OrderDate	CustomerName	OrderingProduct	PID	Quantity	Rate	Total	CID	Action
c67d29d7-24ab-483f-b3c5-8889992c2398	5/10/2018 12:00:00 AM	Aabish Shrestha	Hp Envy Laptop	7beb8b0e-f7c8-4a97-ae79-03c14b586301	1	94000	94000	33444203-1dd1-4f2f-8979-aabac745f7c2	Delete

Again, a large image of a keyboard with a 'Confirm Order' button is overlaid at the bottom.

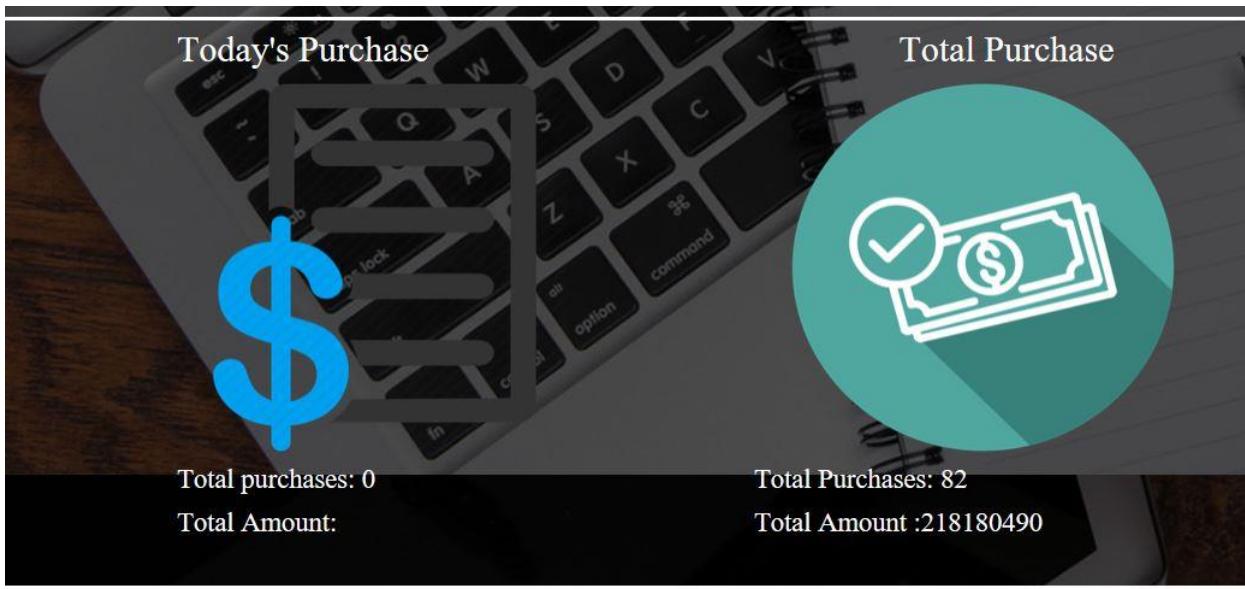
Result: Two records are added and further one record is deleted. Test successful.

Test Case 7: Automatic calculation of total sales and sales today:



Result: Calculation of Sales is automatically performed. Test successful.

Test Case 8: Automatic calculation of total purchase and purchases today:



Result: Sales amount is automatically calculated. Test is successful.

Test case 9: Automatic calculation of total orders from customers, orders made at current day, total orders made to suppliers, and orders made to suppliers at current day.



Result: The required values are automatically calculated. Test is successful.

Test case 10: Testing if the users can successfully register a repair record.

The screenshot shows a web page titled "Repair Records Page" with the URL "localhost:52946/Form_Repair.aspx". The page has a header "Enter Details For Repair Service!" and a form for inputting repair information. The form fields include:

- Customer Name: dropdown menu showing "33444203-1dd1-4f2f-8979-aabac745f7c2".
- Address: [empty input field].
- Phone Model: [empty input field].
- IMEI Number: [empty input field].
- Problem Description: [empty input field].
- Given Date: [empty input field] (mm/dd/yyyy format).
- Select Image: [empty input field] with a "Browse..." button.
- Save: [blue Save button].
- Get Info: [blue button].
- Customer Name: Customer ID: [text input field showing "Customer Name: Customer ID"]

Below the form is a table of repair records:

ID	Customer Name	Customer ID	Address	Phone Model	IMEI Number	Problem	Date	Image
025bcfe5-cb26-4f87-b7e2-5e3ce498f5b5	Imran Ali	71e5e1aa-976d-4a76-b830-01873609418f	Kathmandu	Nokia 6	548651258456849	k vakokhai	5/2/2018 12:00:00 AM	[Image of Imran Ali]

Result: Repair record successfully recorded. Test successful.

Test case 11: Testing if the field required validator works or not.

Enter Details For Repair Service!

Customer Name	<input type="text" value="33444203-1dd1-4f2f-8979-aabac745f7c2"/>	Get Info
Address	<input type="text"/>	
Phone Model	<input type="text"/>	
IMEI Number	<input type="text"/>	
Problem Description	<input type="text"/>	
Given Date	<input type="text" value="mm/dd/yyyy"/>	<input type="button" value="Save"/>
Select Image	<input type="file"/>	Browse...
Please Choose Image File (.jpg, .bmp, .jpeg, .png) & These Fields are Required!		

Result: Required validator is working. Test is successful.

Test case 12: To check if the repair records are deleted or not

Id	Customer Name	Customer ID	Address	Phone Model	IMEI Number	Problem	Date	Image	
025bcfe5-cb26-4f87-b7e2-5e3ce498f5b5	Imran Ali	71e5e1aa-976d-4a76-b830-01873609418f	Kathmandu	Nokia 6	548651258456849	k vakokhai	5/2/2018 12:00:00 AM		Edit Delete
944db9ec-1c2d-4365-9e6b-d26c59e477ab	Default	Phone Model	Kathmandu	Nokia 6	548651258456849	ojoihigfuhih	5/2/2018 12:00:00 AM		Edit Delete
a774d417-36d0-42bf-9fd1-4e92ad7d3546	Bibek Shrestha	1234568794598498494	Pokhara	Nokia 6	548651258456849	Touch problem and battery disorder	4/22/2018 12:00:00 AM		Edit Delete

Id	Customer Name	Customer ID	Address	Phone Model	IMEI Number	Problem	Date	Image
a774d417-36d0-42bf-9fd1-4e92ad7d3546	Bibek Shrestha	1234568794598498494	Pokhara	Nokia 6	548651258456849	Touch problem and battery disorder	4/22/2018 12:00:00 AM	

Result: Two records are deleted from the database. Test is successful.

Test Case 13: Testing if the user can enter a new product into the system from Products form.

Product ID	Name	Quantity	Rate	Sellingrate	Image
398da4f9-45f5-4aa4-8280-ef677d08485a	Samsung Galaxy S9 Mobile	1483	99990	120000	

Result: Product is successfully added. Test is successful.

Test case 14: Testing if the required validator of the Products page works or not.

ADD Product Details

Name

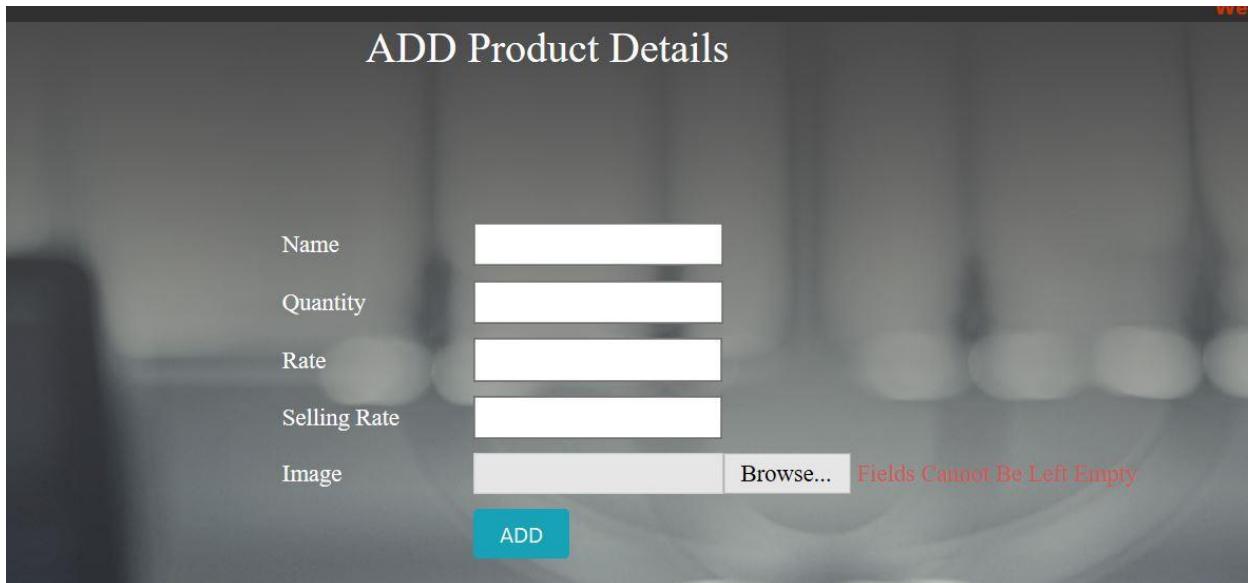
Quantity

Rate

Selling Rate

Image Browse... Fields Cannot Be Left Empty

ADD



Result: Test is successful. The required validator is working.

Test case 15: Testing if the imagefile of products can be uploaded other than (jepg,jpg,bmg,png):

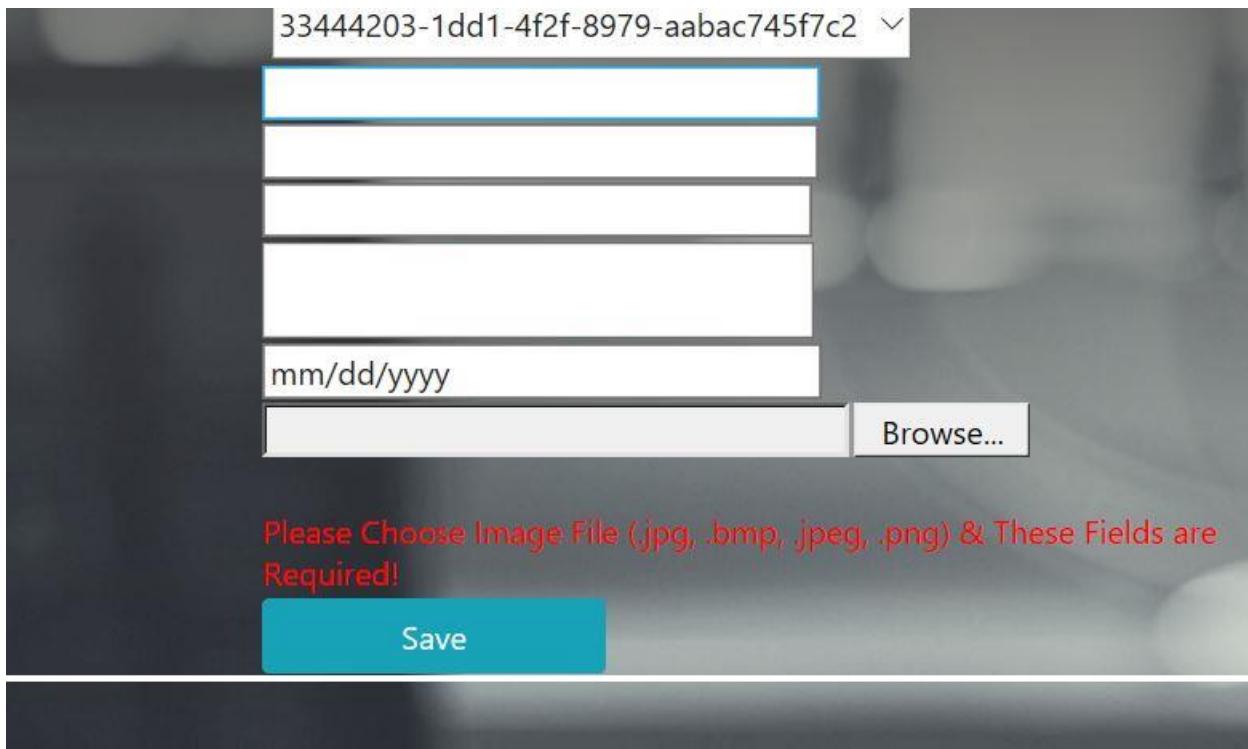
33444203-1dd1-4f2f-8979-aabac745f7c2

mm/dd/yyyy

Browse...

Please Choose Image File (.jpg, .bmp, .jpeg, .png) & These Fields are Required!

Save



Result: Test is successful.

Test case 16: Testing if the user can register a new customer into the system from Customers form.

Id	Name	Age	Contact	Address	Email	Balance	Image	
33444203-1dd1-4f2f-8979-aabac745f7c2	Aabish Shrestha	19	9846072218	Syangja	aabish22@gmail.com	100		<button>Delete</button>
393c8a5a-5749-443f-9ccb-55f0676920f1	Suraj Shrestha	25	9817111661	Syangja	suraj007@gmail.com	0		<button>Delete</button>
49f38bf3-c964-4206-bf8e-db2e23e1d3f4	Sourabh Gurung	29	9802245875	Vedifarm Pokhara	sourabh.motu@gmail.com	0		<button>Delete</button>
								

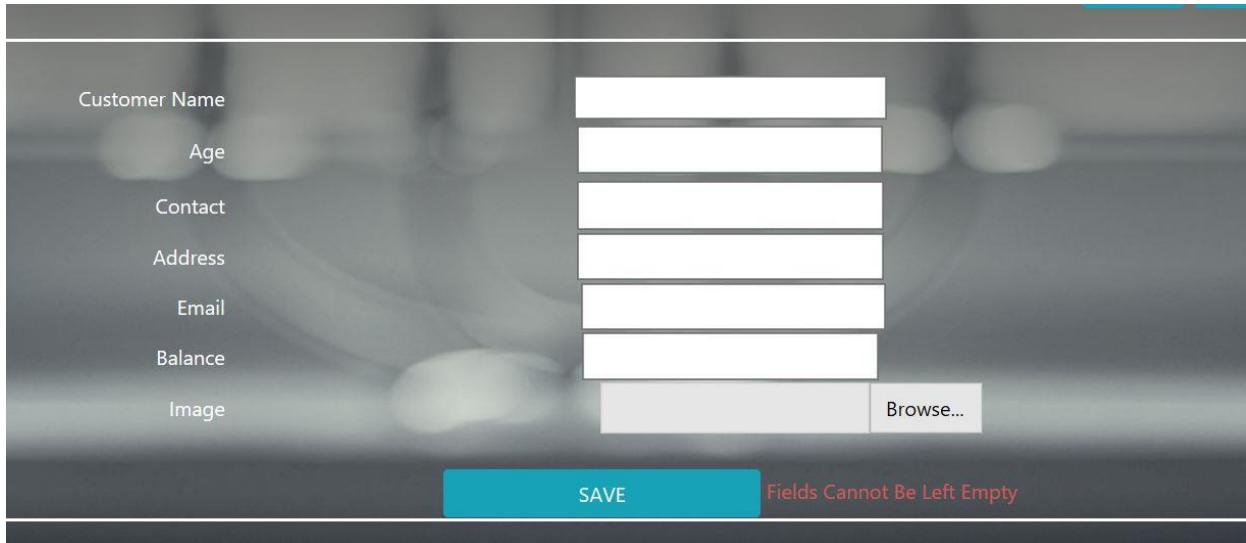
Result: Customers are registered into the system.

Test case 17: Testing if the recorded customers can be edited/deleted or not.

Id	Name	Age	Contact	Address	Email	Balance	Image	
393c8a5a-5749-443f-9ccb-55f0676920f1	Suraj Shrestha	25	9817111661	Syangja	suraj007@gmail.com	0		<button>Delete</button>
49f38bf3-c964-4206-bf8e-db2e23e1d3f4	Sourabh Gurung	29	9802245875	Vedifarm Pokhara	sourabh.motu@gmail.com	0		<button>Delete</button>
78f0e4ce-5897-4cb2-b0ad-480ad4bf1185	Sujata Shrestha	22	9846093965	Walling	sujata123@gmail.com	171250		<button>Delete</button>

Result: The first customer is successfully deleted.

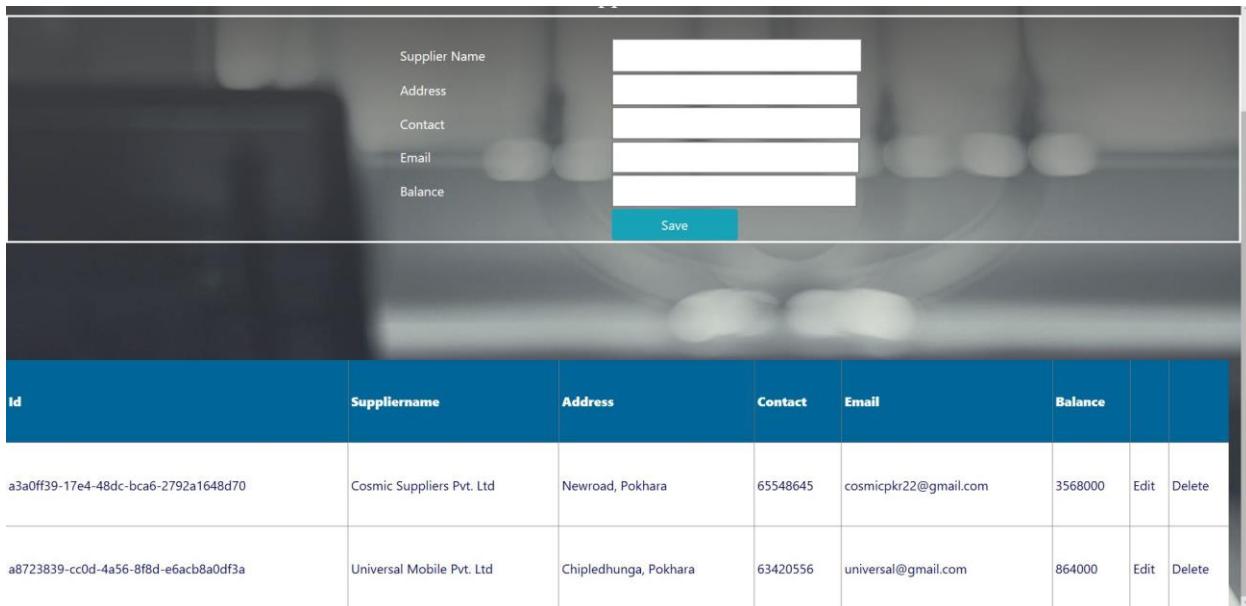
Test case 18: Testing if the required validator in the Customers page works or not.



A screenshot of a mobile application's customer registration form. The form fields include Customer Name, Age, Contact, Address, Email, Balance, and Image. The 'Image' field has a 'Browse...' button. Below the form is a teal 'SAVE' button and a red error message: 'Fields Cannot Be Left Empty'.

Result: Required validator is working. Test is successful.

Test case 19: To check if the user can register a new supplier in the system using Supplier form.



A screenshot of a mobile application's supplier registration form and a list of registered suppliers. The form fields are Supplier Name, Address, Contact, Email, and Balance, with a 'Save' button. Below the form is a table listing suppliers:

Id	Suppliername	Address	Contact	Email	Balance	Edit	Delete
a3a0ff39-17e4-48dc-bca6-2792a1648d70	Cosmic Suppliers Pvt. Ltd	Newroad, Pokhara	65548645	cosmicpkr22@gmail.com	3568000	Edit	Delete
a8723839-cc0d-4a56-8f8d-e6acb8a0df3a	Universal Mobile Pvt. Ltd	Chipledhunga, Pokhara	63420556	universal@gmail.com	864000	Edit	Delete

Result: Suppliers are registered. Test is successful.

Test case 20: Testing if the suppliers can be deleted or not and the required and other validations of the Supplier page works or not.

The screenshot shows a mobile application interface titled "Enter Supplier Details". On the left, there is a vertical list of form fields: "Supplier Name", "Address", "Contact", "Email", and "Balance". To the right of each field is a corresponding input field. A red error message "Fields Cannot be Left Empty" is displayed next to the "Balance" input field. At the bottom center is a blue "Save" button.

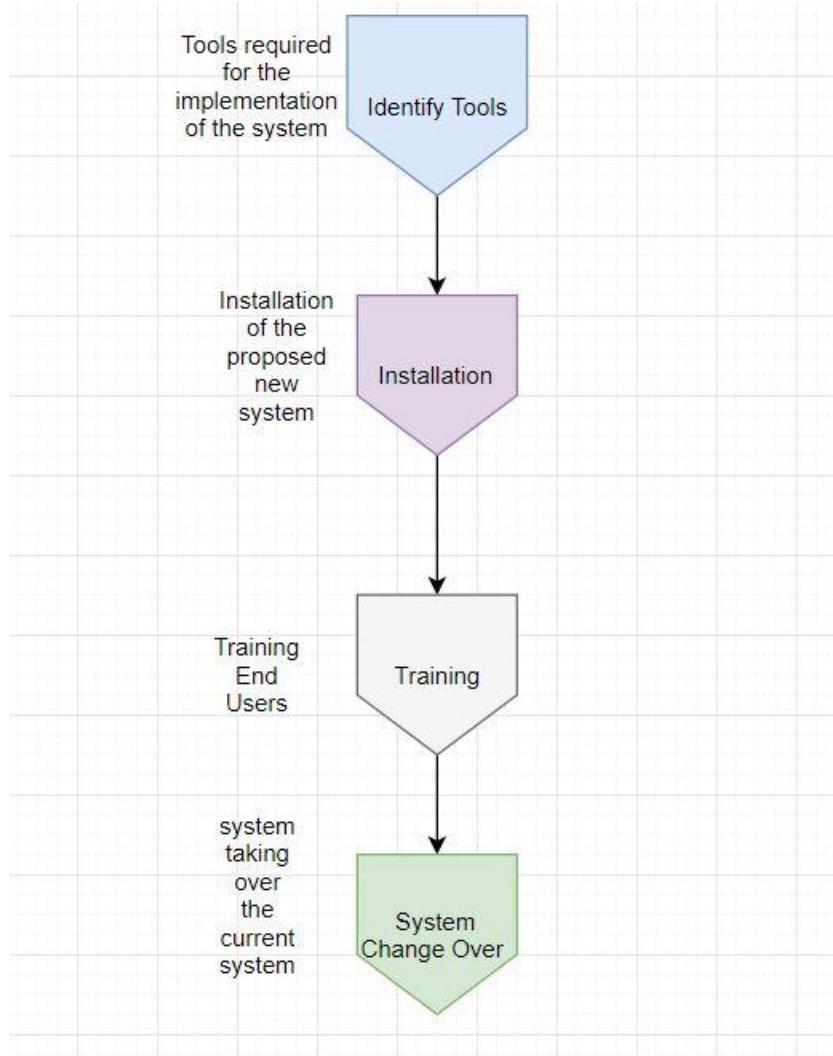
Result: Test is successful.

The remaining test cases are justified in the appendix section of report.

4.3. Implementation Plan:

After the testing process of the proposed new system for all the possible errors and bugs, they have been properly eliminated and removed. Now the system is finally ready to be implemented in the real field. The implementation process is very important as the current system is being operated in the concerned field for so long. This process will create hindrances in daily transactions of the system.

Implementation plan processes:



At the end of the implementation plan, the proposed new system will fully or partially overcome the current system.

4.4. Development Tool Adopted:

Visual Studio 2017:

Microsoft Visual studio is IDE Integrated Development Environment from Microsoft. Visual studio is mostly used to build different computer programs for Microsoft Windows, web sites, web applications, and mobile applications. It utilizes the Microsoft software development platforms like Windows Forms, Windows API, Windows presentation Foundation, Windows Store, Microsoft Silverlight etc. It can also generate both managed and native code.

4.5. Database Adopted:

Microsoft SQL Server:

Microsoft SQL server is a RDBMS relational database management system. It supports a wide range of business intelligence, transaction processing, and analytics applications in corporate IT environments. The Microsoft SQL server is one of the most market-leading database technologies including Oracle Database ad IBM's DB2.

4.6. Hardware Specification:

- Intel Core I3 or higher.
- 1GB Ram or higher.
- 50 GB of permanent storage.
- 1GB Display Memory.

4.7. Software Specifications:

- Windows 7 or higher.
- Microsoft Visual studio 2017.
- Microsoft SQL Server.
- ASP.Net Framework.

4.8. Installation of the System:

The installation of the proposed new system will be based on its different components. As it is a web-based system, the application will be more straightforward. For the installation process, the system should be copied to main directory of the web server.

The installation of the system includes the following steps:

- Installation of MySQL server to the main web server
- Installation of Visual Studio 2017
- Installation of ASP.Net Framework

4.9. Training:

Training is a vital step of the implementation plan. Training process helps the end users/ clients to understand the system in a better way. If the users are properly trained, the system will be widely accepted and used. It is very essential to let the end users know about the operations that are carried out by the proposed system and the differences of those operations to the operations that were carried out by the old system. For this purpose, the following steps can be performed:

- A Detailed Presentation of the system.
- Advertisement of the new system.
- Different seminars and training classes can be performed.
- Online demonstration of the system.
- Different videos and blogs can be made on the system.

4.10. System Changeover:

The system changeover is the process in which the current old system is fully/partially removed or replaced by the proposed new system.

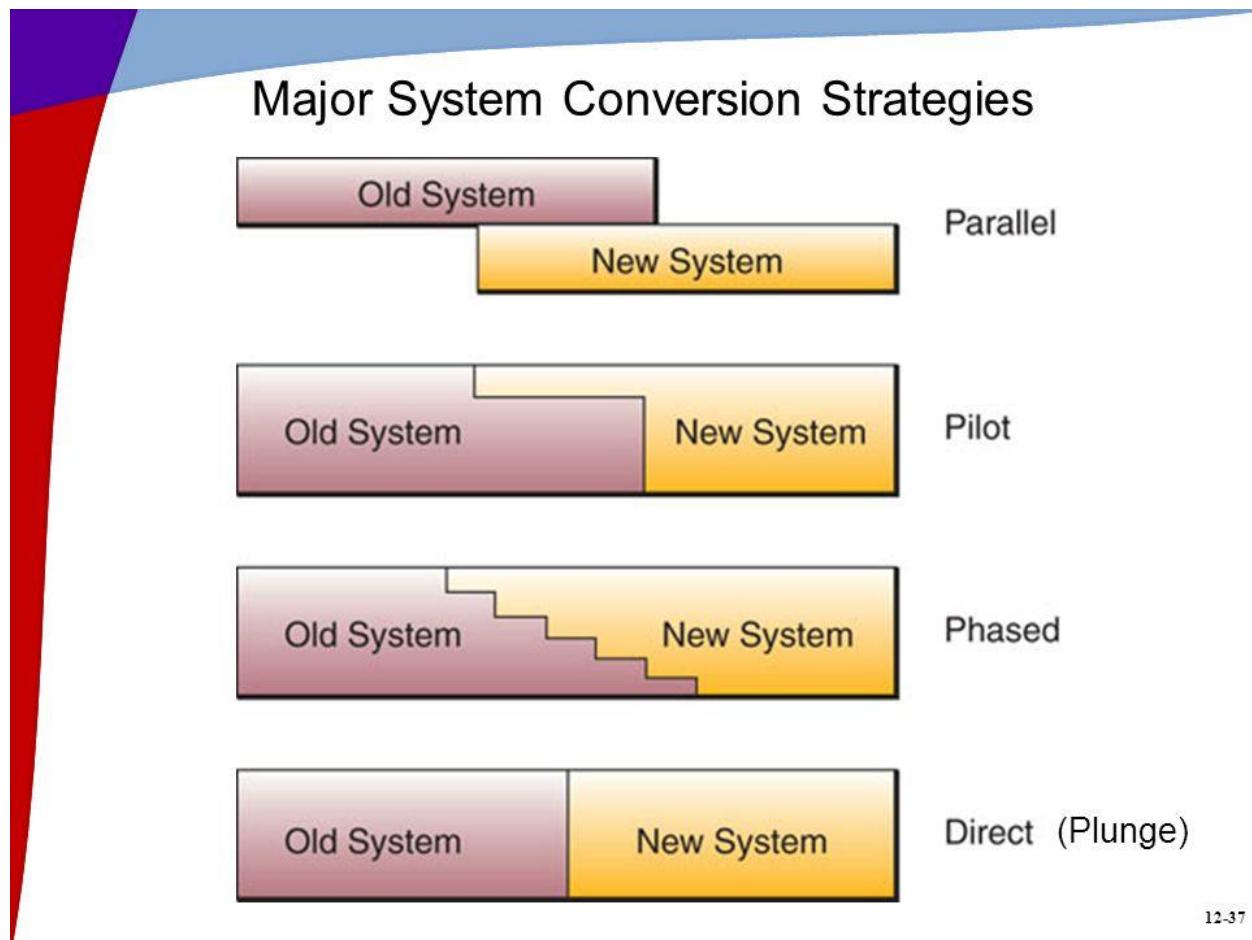


Fig: System Changeover techniques.

Chapter Summary:

The overall testing and implementation plan of the proposed new system are discussed in this chapter. Different Strategies of testing were introduced and were performed. The system was fully evaluated by different test cases and their results. It was further documented. Further the implementation plans and actions were introduced and explained in this chapter.

Chapter 5: Conclusion and Review:

5.1. Conclusion Outline

The conclusion section brings the project into its principle conclusion with respect to its diverse components and improvement strategies. Here, the principle goal of this clarification is to bring an adequate measure of final conclusion of the project with the guide of its elements of the result of the project and the strategy utilized as a part of its advancement methodology. The primary summarization of the accomplishment of the framework are examined in this part to recall the essential astute data in the project improvement and the objective accomplished among the advancement of the considerable number of periods of project advancement from beginning to the last stages.

The understanding process of the current store management system is carried out in order to carry out the initials of the project. . After the collection of this information, we have known about the future service that can be further be added along with the proposed system which will be beneficial for the system.

5.2. Project Achievements:

The major aim of this approach was to overcome the weakness and demerits of the current system by developing a proposed new system. Firstly, the system development process was initialized keeping the focus on the problem domain. The problem domain process was performed through different series of research, system development, testing evaluation, and implementation phase. I was able to understand the scope of the project as a whole and further implement the solutions to the problem while the continuation of the development process. There were lots of problems and challenges on bringing the new features like purchase returns and proving the product information to the user. The final system was tested with its flexibility, acceptability and maintainability. Hence, I was able to achieve lots of achievements in the process along with their respective importance which was been able through performing detailed study on the properties of the system focusing on the ideas.

5.3. Future Enhancement:

The proposed system has been carried out for “Suraj Mobile Zone” store in order to replace the current system with new system. The current system had no or limited provision of storage of data and information in digital basis. With the implementation of the current system, there can be many changes in the management process of the store. The changes include storage of repair records of the customer’s phones, storage of customers and supplier’s information. Information about the quantity of the stocked goods, information about the total sales, purchases, orders and orders made to supplier are some of the major changes in the system. The idea of the new system is very applicable and efficient.

As we all know that no system is perfect enough to fulfill all the requirements and needs. In the current Mobile store management system, there is no division of products available. The functionality in some of the forms like Sales and Purchase returns is very difficult due to presence of many buttons and drop downs. Many enhancements can be done to the system and a means of contacting the supplier can also be added in the system. The User Interface of the system can also be made more attractive and the purchase and sales process of the system can be simplified. Furthermore, the products presented in the system can be classified according to type and Adding all the product of the store manually can be a huge challenge for the user. Provision of auto backup generation and cloud storage can also be added in the system.

5.4. Evaluation Introduction:

Evaluation of the system segments is described by this chapter. The central idea of this chapter is for carrying out an overview of the evaluation process which was very necessary during the development of the proposed system which will define the importance of applied solutions which surrounds the two types of strategies for the project.

The discussion on the evaluation of the project, its outcome and the practices carried out has been carefully elaborated in this chapter.

5.5. Evaluation of the project Deliverables:

The final goal for the implementation process of this proposed new system is to provide accurate and reliable inventory and store management features to the store. The system tends to overcome all the limitations of the currently used system.

The digital records maintained by the system, information about the products, order process contact information and the management ability of the new system should overcome all the incompleteness and problems of the currently used system.

5.6. System Evaluation:

The system evaluation is done on the premise of all the working capacities inside the system. During the time spent on system evaluation, the functional evaluation of the project is done which will ascertain the functional requirements of the system and its usage exactness among the advancement which consequently ought to satisfy the whole set of requirements of the system.

References:

- Techopedia.com. 2017. *What is a Web-Based Application? - Definition from Techopedia.* [ONLINE] Available at: <https://www.techopedia.com/definition/26002/web-based-application>. [Accessed 25 November 2017].
- WhatIs.com. 2017. *What is Web based project management software? - Definition from WhatIs.com.* [ONLINE] Available at: <http://whatis.techtarget.com/definition/Web-based-project-management-software>. [Accessed 1 December 2017].
- Inventory System Software Blog. 2017. *A Brief History of Inventory Management / Inventory System Software Blog.* [ONLINE] Available at: <https://inventorystatusoftware.wordpress.com/2012/02/06/brief-history-inventory-management/>. [Accessed 2 December 2017].
- A brief history of point-of-sale (POS) | Vend POS System. 2017. *A brief history of point-of-sale (POS) | Vend POS System.* [ONLINE] Available at: <https://www.vendhq.com/pos>. [Accessed 22 November 2017].
- Understanding Detailed Architecture of ASP.NET 4.5. 2017. *Understanding Detailed Architecture of ASP.NET 4.5.* [ONLINE] Available at: <http://www.dotnettricks.com/learn/aspnet/understanding-detailed-architecture-of-aspnet-45>. [Accessed 26 November 2017].
- Throwaway Prototyping Model | IT Training and Consulting – Exforsys. 2017. *Throwaway Prototyping Model | IT Training and Consulting – Exforsys.* [ONLINE] Available at: <http://www.exforsys.com/career-center/project-management-life-cycle/throwaway-prototyping-model.html>. [Accessed 11 December 2017].
- <https://techterms.com/definition/aspnet>. [Accessed 22 December 2017].
- Teach-ICT A2 Level ICT OCR exam board - The System Life Cycle, prototyping, rapid application development, RAD. 2017. *Teach-ICT A2 Level ICT OCR exam board - The System Life Cycle, prototyping, rapid application development, RAD.* [ONLINE] Available at: http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/331_systems_cycle/prototyping_RAD/miniweb/pg_3.htm. [Accessed 11 December 2017].
- SearchWinDevelopment. 2017. *What is ASP.NET (ASP+)? - Definition from WhatIs.com.* [ONLINE] Available at: <http://searchwindevelopment.techtarget.com/definition/ASPNET>. [Accessed 21 December 2017].
- Latest Asp.net & C# Projects Topics & Ideas | NevonProjects. 2017. *Latest Asp.net & C# Projects Topics & Ideas | NevonProjects.* [ONLINE] Available at: <http://nevонprojects.com/dot-net-based-projects/>. [Accessed 22 December 2017].

SearchWinDevelopment. 2017. *What is Visual Studio .NET? - Definition from WhatIs.com.* [ONLINE] Available at: <http://searchwindevelopment.techtarget.com/definition/Visual-Studio-NET>. [Accessed 21 December 2017].

WhatIs.com. 2017. *What is PHP (Hypertext Preprocessor) ? - Definition from WhatIs.com.* [ONLINE] Available at: <http://whatis.techtarget.com/definition/PHP-Hypertext-Preprocessor>. [Accessed 22 December 2017].

ASP.NET vs PHP. 2017. *ASP.NET vs PHP.* [ONLINE] Available at: <https://www.orientsoftware.net/technologies/microsoft-net/aspnet-vs-php/>. [Accessed 22 December 2017].

Difference between ASP.Net, C#.Net and VB.Net? - Stack Overflow. 2017. *Difference between ASP.Net, C#.Net and VB.Net? - Stack Overflow.* [ONLINE] Available at: <https://stackoverflow.com/questions/4051507/difference-between-asp-net-c-net-and-vb-net>. [Accessed 22 December 2017].

What are advantages of ASP.net over PHP? How is it better and why? - Updated. 2017. *What are advantages of ASP.net over PHP? How is it better and why? - Updated.* [ONLINE] Available at: <https://www.quora.com/What-are-advantages-of-ASP-net-over-PHP-How-is-it-better-and-why>. [Accessed 22 December 2017].

The Official Forums for Microsoft ASP.NET. 2017. *Why C# Instead of VB? | The ASP.NET Forums.* [ONLINE] Available at: <https://forums.asp.net/t/202828.aspx?Why+C+Instead+of+VB+>. [Accessed 22 December 2017].

Techopedia.com. 2017. *What is ASP.NET? - Definition from Techopedia.* [ONLINE] Available at: <https://www.techopedia.com/definition/3213/asp-net>. [Accessed 22 December 2017].

Scribd. 2017. *Mobile Store Management System | C Sharp (Programming Language) | Pointer (Computer Programming).* [ONLINE] Available at: <https://www.scribd.com/doc/97508346/Mobile-Store-Management-System>. [Accessed 26 November 2017].

PHP (Hypertext Preprocessor) Definition. 2017. *PHP (Hypertext Preprocessor) Definition.* [ONLINE] Available at: <https://techterms.com/definition/php>. [Accessed 22 December 2017].

Appendix:

User Manual

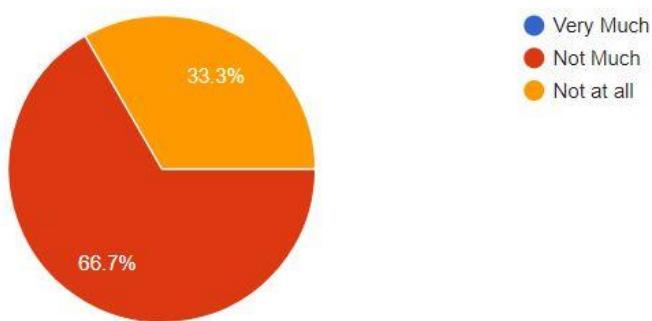
- Firstly, run the program by opening Mobile Store Management System.sln file in Visual Studio.
- The program will open in visual studio.
- Execute the program in a browser.
- After running of program without any errors, the default page will appear.
- On the default page, information about the available products in the store is displayed.
- Users can search the desired product on the search bar entering the name of product and clicking on the search button.
- Users can refresh the search process by clicking on the search button.
- To order product from the system, click on the orders label at the top of the navigation bar.
- The system will be redirected to the orders page.
- On the orders page, Get id button has to be clicked first in order to get an orders id, and then the customer has to select the date of order. Further, the id of customer has to be selected and the product and desirable quantity has to be entered. Then the order needs to be confirmed to complete the order process.
- To get the location of the store, a user has to click on the location link available on the navigation bar of the default page.
- To login into the system, users have to click on the login link on the navigation bar.
- The system will be redirect to the Login page.
- On the login page, the user has to enter the registered username and password.
- After successful login, the system will be redirected to homepage.
- To enter repair records, a user has to click on the repair link of the navigation bar.
- The system will be redirected to the Repair form.
- To enter a repair record form, the user has to select the customer id first at the dropdown list. Then the user has to enter the necessary information and select an image file of the phone and click on the save button to save the record.
- To add a customer in the system, click on the customer logo in the navigation bar.
- The system will be redirected on the customers form. Enter the name age contact and other information in the form and further select an image file in the system. Click on the save button to save the record into the database.
- To search a customer, enter the name of the customer on the search bar and click on the search button. Click on the refresh button to refresh the search process.
- To add a product on the system, go to the Products page, enter the product's Name quantity, rates and select an image file for the respective product. Click on the save button in order to store the details of new products into the database
- To purchase products into the system, click on the purchase label of the navigation bar. The system will be redirect to the products page.

- Firstly, click on the Get Purchase ID button then select date, supplier and product on the form and enter the purchasing quantity. Click on the save button to store the information into purchase database.
- To sell a product, Go to the sales form, click on the Get Sales Id button, select a customer and click on the get info button to retrieve necessary information of the customer. Select the item from the dropdown list to be sold and enter the discount value in %. Further, click on the calculate button to generate the discounted price. Click on add item to add the product into grid view. Then, specify the payment method on the radio buttons. It could be either on cash basis or in credit basis. The balance is not recorded in cash sales. Then click on the Makes Sales button in order to confirm the sales process.
- To return a purchased product, Go to the Purchase returns page, get a purchase returns id by clicking on the Get PRID button then select the respective purchased id from the drop down. The details of the purchase will appear after clicking on the Get Info button. Further select the date and enter the returning quantity. Click on the Done button in order to save the purchase returns entry.
- To return a sold product, Go to the Sales Return page; Get a sales return id by clicking on the Get ID button. Select the respective sales id at the dropdown menu and click on the get info button to retrieve the sales details. Enter the returning quantity and click on the done button in order to save the sales return entry.
- To enter the due amount from customer, Go to the Receipt form, select the id of customer and click on the Get Info button in order to get the balance information. Select a date and enter the received amount and click on the Save button in order to save the record.
- To enter the payment to supplier due entry, Go to the Payment form. Select a date, supplier and click on the Details button to retrieve the balance details. Enter the paid amount to supplier and click on the save button in order to save the payment entry.
- To change the password click on the Welcome, @Username link on the navigation bar. The system will be redirected to the change password form. The current password and recovery key will be available there. Enter the new password or recovery key to update them.
- To recover your password, Click on the Forgot password link on the login page, the system will be redirected to the Password Recovery page. Select your username and enter the recovery key. Click on the check button in order to check the recovery key. If the recovery key is correct, the fields to enter new passwords will be visible. Enter and confirm new password to update them.

Results of Survey:**Results of recently taken survey:**

How satisfied are you with your current system??

3 responses



As the chart illustrates, the current system is not satisfactory for the end users.

Which Management system is better in you view?

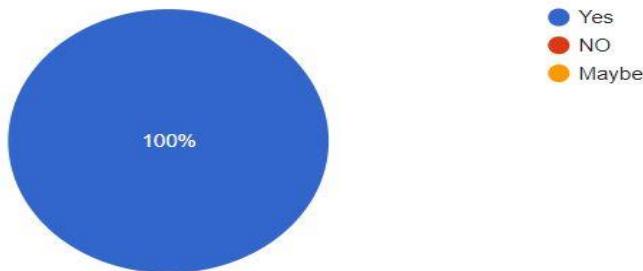
4 responses



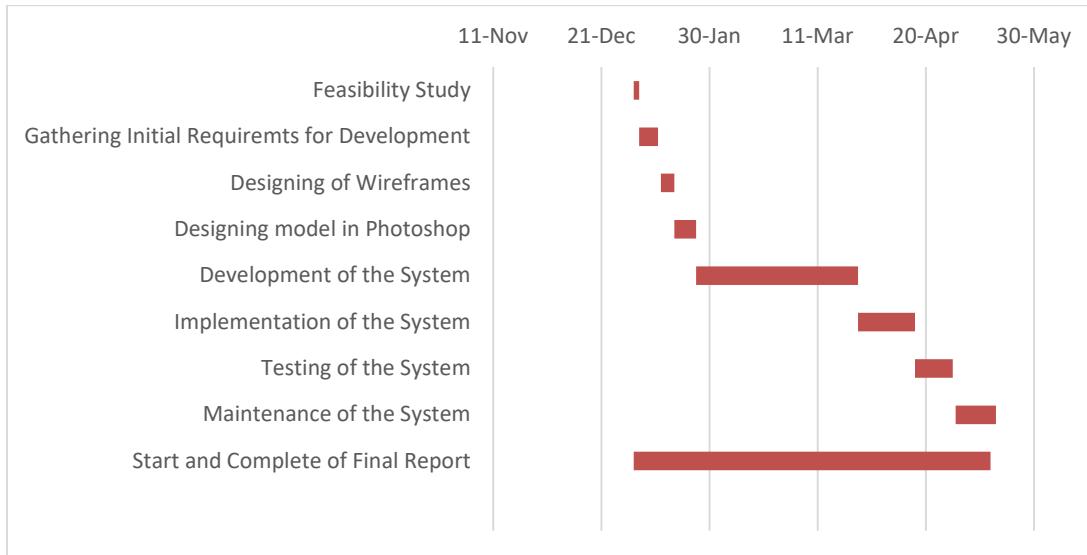
As the chart illustrates, the end users believe that digital system is better than paper based system.

Suppose if you have a store. Would you like a good inventory management system embedded in it?

4 responses



As the chart illustrates, the end users are willing to implement a good inventory management system in their store.

Gantt Chart of Future work*Figure 8: Gantt Chart of Future Work***Analysis of Gantt Chart:**

In accordance to the above Gantt chart, the above activities that will be performed in the future for the completion of this project are clearly stated with its time duration. The Chart illustrates the division of work and the required time for the completion of this project as well as the final report. The activities mentioned in the Gantt chart are according to the selected methodology i.e the Waterfall methodology. Firstly, different requirement of the proposed system are studied gathered and analyzed. Then the development process of the system is carried out. After the development process, the system is implemented, tested and further maintained for the long run.

Software Requirements Specification

for

Mobile Store Management System

approved

Prepared by <Bibek Shrestha>

<Islington College>

<2017-18> |

2.7 Technological Requirements

The technological requirements include the frontend programming tool and the backend database system. The application should be easy to use, and it should be easily managed by any person with little knowledge of computers. The database should be easy to install and configure. At the same time, it should be portable and independent so that we can use the database anywhere and install it on any laptop or PC. This application should be very easy to install on any machine. Finally, this application should not require higher configuration on any machine.

2.8 User Interfaces

The application should be very simple and easy to use by any employee. Below are the basic requirements of general user interface:

- The application shall be easy to use.
- The application shall take few inputs from user.
- The understanding time of the application shall be very small.

2.9 Functional Requirements

The functional requirements of the application are as follows:

- The application shall have all required functionality which is necessary for mobile store inventory and billing system management.
- The application shall have ability to suggest the high stocked and low stocked products.
- The application shall have functionality to be used by different stores from one location.
- The application shall have user name and password protected security system

Alternate programming languages:**1. Visual Basic:**

Visual Basic which is often referred as (VB) is a programming language and development environment designed by Microsoft. It combines the general functions and commands with graphical controls so it can be called an extension of the basic programming language. It provides (GUI) Graphical User Interface which allows a developer to drag and drop the objects of the program and also the manual writing of program code. It was designed to make the development of programs easy, effective and even powerful enough to build advance programs.

2. PHP:

PHP stands for Hypertext Processor. It is a HTML embedded web scripting language that means the codes written in PHP can be inserted in HTML of a web page. The codes of PHP are read by the server when the PHP page is accessed. The functional outputs of the PHP on the web page are returned as HTML code that is easily read by the web browser. As the codes of the PHP are transformed into HTML code before the loading process of the page, users are unable to view the PHP code on the page.

C#, Asp.net over Visual Basic:

C# was developed with .Net in concept. It confirms best to .Net common language specification of any of the languages. Visual Basic had to continuously re-developed to fit in the specific area. People from VB background claim that there is nothing wrong with VB. It works fine and provides easy transition. The C#, ASP.Net is developing much faster and provides much more powerful features which aren't provided by Visual Basic. Pointers and XML documentation are two major advantages of C#, ASP.Net over VB.

ASP.net over PHP:

There are many programming languages for a web developer to choose. All the programming languages are dominated by ASP.Net and PHP. They both have their own advantages and disadvantages. The decision lies in selecting which advantages and disadvantages suits the required development needs. The flexibility and object oriented features are one of the major advantages of ASP.Net over PHP. The library of ASP.Net is task-based as the library is organized into inheritable classes. Parsing of the code is done within its delimiters in PHP. Anything beyond its delimiters is sent directly to the output.

Remaining Test Cases:

Test case21: Testing if the user can purchase a product from the purchase page.

Specify the Items to be purchased

Purchase ID: c03e77f8-3c5e-4af8-951c-a88b245b3b5f	<input type="button" value="Get Purchase ID"/>						
Purchase Date: Purchase Date	<input type="text" value="5/10/2018"/>						
Supplier Name: Supplier	<input type="button" value="Cosmic Suppliers Pvt. Ltd"/>						
Product Name: Product	<input type="button" value="Hp Envy Laptop"/>						
Rate	<input type="text" value="93000"/>						
Quantity	<input type="text" value="1"/>						
Available Quantity :13	Remaining Stock : 14						
<input type="button" value="Add Item"/>							
Item ADDED SUCCESSFULLY!							
Purchase ID	Date	Supplier Name	Product Name	Quantity	Rate	Amount	Operations
c03e77f8-3c5e-4af8-951c-a88b245b3b5f	5/10/2018 12:00:00 AM	Cosmic Suppliers Pvt. Ltd	Hp Envy Laptop	1	93000	93000	<input type="button" value="Delete"/> <input type="button" value="Delete"/>
Payment Method: <input checked="" type="radio"/> Payment on Cash <input type="radio"/> Payment On Credit							

Result: Test is Successful:

Test case 22: Testing if the quantity of total product is updated after successful purchase.

Specify the Items to be purchased

Purchase ID: c03e77f8-3c5e-4af8-951c-a88b245b3b5f	<input type="button" value="Get Purchase ID"/>
Purchase Date: Purchase Date	<input type="text" value="5/10/2018"/>
Supplier Name: Supplier	<input type="button" value="Cosmic Suppliers Pvt. Ltd"/>
Product Name: Product	<input type="button" value="Hp Envy Laptop"/>
Rate	<input type="text" value="93000"/>
Quantity	<input type="text" value=""/>
Available Quantity :13	Remaining Stock : 14

Result: Test is Successful:

Text case 23: Testing if the required validator of the form works or not

Purchase ID: c03e77f8-3c5e-4afd-951c-a88b245b3b5f

Purchase Date: Purchase Date
5/10/2018

Supplier Name: Supplier
Cosmic Suppliers Pvt. Lt^c

Product Name: Product
Hp Envy Laptop

Rate
93000

Quantity

Available Quantity :13

Add Item

Missing Fields

Result: Test is Successful:

Test case 24: Testing if the record of sales is recorded in the sales table or not.

Sales Record Entry																										
Selling Quantity	<input type="text" value="1"/>	Selling Rate	<input type="text" value="95000"/> <th>Discount in %</th> <td><input type="text"/></td> <th>Calculate</th> <th colspan="2">Remaining Stock : 13</th>	Discount in %	<input type="text"/>	Calculate	Remaining Stock : 13																			
Item added																										
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #0070C0; color: white;">Id</th> <th style="background-color: #0070C0; color: white;">Name</th> <th style="background-color: #0070C0; color: white;">CID</th> <th style="background-color: #0070C0; color: white;">Item</th> <th style="background-color: #0070C0; color: white;">Date</th> <th style="background-color: #0070C0; color: white;">Purchaserate</th> <th style="background-color: #0070C0; color: white;">Salesrate</th> <th style="background-color: #0070C0; color: white;">Sellingquantity</th> <th style="background-color: #0070C0; color: white;">Total</th> </tr> </thead> <tbody> <tr> <td>f26c9623-6344-4074-97ce-442b510aa76b</td> <td>Sujata Shrestha</td> <td>78f0e4ce-5897-4cb2-b0ad-480ad4bf1185</td> <td>Hp Envy Laptop</td> <td>5/10/2018 12:00:00 AM</td> <td>93000</td> <td>95000</td> <td>1</td> <td>95000</td> </tr> </tbody> </table>									Id	Name	CID	Item	Date	Purchaserate	Salesrate	Sellingquantity	Total	f26c9623-6344-4074-97ce-442b510aa76b	Sujata Shrestha	78f0e4ce-5897-4cb2-b0ad-480ad4bf1185	Hp Envy Laptop	5/10/2018 12:00:00 AM	93000	95000	1	95000
Id	Name	CID	Item	Date	Purchaserate	Salesrate	Sellingquantity	Total																		
f26c9623-6344-4074-97ce-442b510aa76b	Sujata Shrestha	78f0e4ce-5897-4cb2-b0ad-480ad4bf1185	Hp Envy Laptop	5/10/2018 12:00:00 AM	93000	95000	1	95000																		

Result: Test is Successful:

Test case25: Testing if the balance and quantity of customers and total product are updated or not.

Item	Hp Envy Laptop	Select Item
Item Purchased Rate	93000	
Available Quantity	14	
Selling Quantity	1	
Selling Rate	95000	
Discount in %		Calculate
Remaining Stock : 13		
ADD Item		
Item added		

Result: Test is Successful:

Test case 26: Testing if a user can enter a record in the purchase returns table through Purchase Returns form.

PRId	Date	PID	Pname	Quantity	Rate	Sname	Amount
'1388f10-b13e-4d85-aa4d-317c8c7aa853	4/24/2018 12:00:00 AM	62b26e4b-156b-408f-9567-7151d68ac556	Nokia 3 Mobile	5	12500	Cosmic Suppliers Pvt. Ltd	62500
'26e7e88-aec5-44d7-90c6-da1530f56896	4/24/2018 12:00:00 AM	62b26e4b-156b-408f-9567-7151d68ac556	Nokia 3 Mobile	5	12500	Cosmic Suppliers Pvt. Ltd	62500
'934d209-fd29-402a-a550-901a0e518d8a	4/24/2018 12:00:00 AM	28743477-9107-429e-8649-07d29804e615	Samsung Galaxy S9 Mobile	1	99990	Universal Mobile Pvt. Ltd	99990
'a494475-394f-4451-85e2-685c1ad7281c	4/25/2018 12:00:00 AM	62b26e4b-156b-408f-9567-7151d68ac556	Nokia 3 Mobile	5	12500	Cosmic Suppliers Pvt. Ltd	62500

Result: Test is Successful:

Test case 27: Making sure if user cannot return more amount of products than purchased quantity.

The screenshot shows the 'Purchase Returns' page. The 'Returning Quantity' field contains '11'. A red error message at the bottom states: 'Returning Quantity cannot be greater than Purchased quantity'.

Result: Test is Successful:

Test case 28: Testing if the user cannot return more amount of product than the stock product.

The screenshot shows the 'Purchase Returns' page. The 'Returning Quantity' field contains '1504'. A red error message at the bottom states: 'Not enough product quantity in the stock!'

Result: Test is Successful:

Test case 29: Testing if the user can enter a sales return entry into the sales return table or not.

The screenshot shows the 'Sales Return Page'. The table lists three sales return entries:

Sales Return ID	Date	Sales ID	Product	Quantity	Rate	Customer ID	Amount	Operations
143d073-c891-4dc8-b2eb-3785a62167c	5/3/2018 12:00:00 AM	04ab602b-9e45-4b9b-a7f6-dd36e70c4db5	Samsung G530 Mobile	1	11500	83909ccdd-85c3-4e7b-a037-db4847ca67ec	11500	Edit Delete
6ab5de7-2087-4c5f-a225-24140e2d418	4/27/2018 12:00:00 AM	a32d40a0-cef5-4116-93d3-b28294176751	Nokia 2 Mobile	1	10600	029d4c2e-f73c-4a34-8fa1-e1bc2e8911dc	10600	Edit Delete
6ab8df6-86f1-497e-b2f2-...	5/3/2018 12:00:00 ...	73e550c3-fdf8-44e1-8081-...	Samsung G530 Mobile	1	11500	0d5820af-aa36-4595-a192-...	11500	Edit Delete

Result: Test is Successful:

Test case 30: Testing if the user can return more amount of product than sold products.

The screenshot shows the "Sales Return Page". At the top, there are input fields for "PRID" (d981b71-5931-4119-a623-479d7eb7af02), "Get ID", "Select Sales ID" (04ab602b-9e45-4b9b-a7f6-dd36e70c4db), "Get Info", "Item: Samsung G530 Mobile", "Sold Rate: 11500", "Sold Quantity: 1", "Customer ID: 83909cd-85c3-4e7b-a037-db4847ca67ec", "Balance: Balance", "Date: 5/10/2018", "Returning Quantity: 10", and "Total Quantity: 1587". A button labeled "Get Balance" is visible. A red error message at the bottom right says "Returning Quantity cannot be greater than Purchased quantity".

Result: Test is Successful:

Test case 31: Testing if the admin can register a new user into a system or not through Registration page

The screenshot shows the "Registration Form" with fields for First Name, Last Name, Email, Phone No., Username, Userlevel (set to 1), Password, Confirm Password, and Recovery Password (set to 12345). A "Register" button is at the bottom. Below the form is a table titled "User List" with columns: Id, Username, Password, Fisrtname, Lastname, Email, Phone, Userlevel, Recovery, and Operations. The table contains five rows of user data.

Id	Username	Password	Fisrtname	Lastname	Email	Phone	Userlevel	Recovery	Operations
Baf7ec2c-4cd4-4376-a4bc-289241cad864	basant22	basant123	Basanta	Shrestha	basanta.shrestha22@gmail.com	9846522015	2	12345	Edit Delete
c2948d9c-9b67-48f9-835a-45a7c385784e	temp22	temp	Temp	User	tempuser@gamil.com	9806608816	2	12345	Edit Delete
c7ddf99d-7a22-4f9c-856b-be723e4f81c8	cracetha22	00292	Bibek	Shrestha	duke.ur.amigo@gmail.com	9806608816	1	12345	Edit Delete
f7c37a09-18f9-4cee-ae4e-d5b34f0a2131	prakash123	ok	Prakash	Shrestha	prakash@gmail.com	9846500235	2	12345	Edit Delete

Result: Test is Successful:

Test case 32: Testing required and Compare validator of the registration page.

The screenshot shows a registration form with the following fields and their values:

- First Name: (empty)
- Last Name: (empty)
- Email: (empty)
- Phone N.o: (empty)
- Username: (empty)
- Userlevel: 1
- Password: dfsfdf
- Confirm Password: •••••
- Recovery Password: 12345

A validation message "Password dont match!!" is displayed next to the Confirm Password field.

Result: Test is Successful:

Test case 33: Testing if the user can enter records in Payment table through Payment form.

The screenshot shows a payment form with the following fields and their values:

Date	mm/dd/yyyy
Select Supplier	Cosmic Suppliers Pvt. Ltd
Supplier Id	ID
Previous Balance-	Balance
Amount Received	<input type="text"/>

Below the form is a table of payment records:

PId	SID	Date	Amount	
1b182ed1-722b-4aaa-889c-7155fe35d500	a3a0ff39-17e4-48dc-bca6-2792a1648d70	5/2/2018 12:00:00 AM	1000	Edit Delete
a904cf99-e958-4c3c-8291-0977a938de50	a8723839-cc0d-4a56-8f8d-e6acb8a0df3a	1/1/1900 12:00:00 AM	40000	Edit Delete
cfde98b5-9db4-4876-b073-7d89c7606e9a	a8723839-cc0d-4a56-8f8d-e6acb8a0df3a	5/9/2018 12:00:00 AM	40000	Edit Delete
f91d3e81-6d6d-4443-b827-e13b318d8909	a8723839-cc0d-4a56-8f8d-e6acb8a0df3a	5/3/2018 12:00:00 AM	50000	Edit Delete

Result: Test is Successful:

Test case 34: Testing if the user can enter an amount to pay that is greater than stored amount to be paid of supplier

The screenshot shows the 'Payment To Suppliers' page of the Mobile Store Management System. At the top, there is a navigation bar with links: Repair Records, Purchase, Sales, Customers, Supplier, Products, Payment, Receipt, Order Supplier. A welcome message 'Welcome, cracetha22!' and a 'Signout?' link are also present. The main title 'Payment To Suppliers' is centered above a form. The form fields are: Date (5/10/2018), Select Supplier (Cosmic Suppliers Pvt. Ltd), Supplier Id (a3a0ff39-17e4-48dc-bca6-2792a1648d70), Previous Balance- (3568000), and Amount Received (35680005). Below the form is a button labeled 'Receive' and a message: 'Received amount shouldnot be greater than actual amount to be received'.

Result: Test is Successful:

Test case 35: Testing if the users can entry payment records of customers from Payment form:

The screenshot shows the 'Receive Payment From Customers' page. The form fields are: Date (5/10/2018), Select Customer (Roshani Shrestha), Customer Id (9ef4648a-1664-4c3d-aeca-b4bc1e2bfe36), Previous Balance- (1000), and Amount Received (left empty). Below the form is a button labeled 'Receive'. The page then displays a table of payment records with columns: Receipt ID, Customer ID, Date, Amount, and Operations. The first record in the table is: 1c0e3897-b65d-493e-abb5-5e35301725e8, 0d5820af-aa36-4595-a192-f91e23d5fef2, 5/2/2018 12:00:00 AM, 56, with 'Edit' and 'Delete' buttons.

Result: Test is Successful:

Test case 36: Testing if the user can enter an amount being receive that is greater than stored amount to be paid by customer

The screenshot shows the 'Receive Payment From Customers' page. The form fields are as follows:

- Date: 5/10/2018
- Select Customer: Roshani Shrestha
- Customer Id: 9ef4648a-1664-4c3d-aeca-b4bc1e2bfe36
- Previous Balance: 1000
- Amount Received: 1200457

A red error message at the bottom states: "Received amount shouldnot be greater than actual amount to be received".

Result: Test is Successful:

Test case 37: Testing if the user can change their password and recovery key or not:

The screenshot shows the 'Enter Your New Password or Recoverykey to Continue' page for 'Password Change'. The form fields are:

- Current Password: 00292
- New Password: (empty)
- Confirm Password: (empty)
- Save button

On the right, there is a 'Change Recovery Key' section with fields for Current Recovery Key (00292), New Recovery key, Confirm Recovery Key, and a Save button.

The screenshot shows the same 'Enter Your New Password or Recoverykey to Continue' page, but for 'Change Recovery Key'. The form fields are:

- Current Recovery Key: 00292
- New Recovery key: (empty)
- Confirm Recovery Key: (empty)
- Save button

On the left, there is a 'Password Change' section with fields for Current Password (bibeck123), New Password, Confirm Password, and a Save button.

Result: Test is Successful:

Test case 38: Testing if the compare validator of new password works or not.

The screenshot shows a 'Password Change' form. It has three input fields: 'Current Password' containing 'bibek123', 'New Password' containing three dots ('•••'), and 'Confirm Password' containing five dots ('••••'). A red error message 'Passwords dont Match!' is displayed below the fields. A blue 'Save' button is at the bottom.

Result: Test is Successful:

Test case 39: Testing if the users can successfully reset their password or not:

The screenshot shows a 'Password Recovery Page'. The title is 'Password Recovery Page'. Below it is a sub-instruction: 'Enter your recovery key to reset your password or Contact Admin'. There are three input fields: 'Select Your Username' with 'cracetha22', 'Enter Your Recovery Pin' with '12345', and 'Enter New Password to Continue' with two rows of four dots each. To the right of the first two fields is a 'Check' button. Below the fields are two buttons: 'Save' and 'Correct Pin'. The 'Save' button is highlighted in blue.

Result: Test is Successful:

Test case 40: Testing if the compare validator compares the recovery key and gain access to update the new password.

The screenshot shows a 'Password Recovery Page' with the following fields and status:

- Select Your Username: cracetha22 (dropdown menu)
- Enter Your Recovery Pin: 12345 (button: Check)
- Enter New Password to Continue: (Value: *********)
- Re-enter New Password: (Value: *********) (Status: Password dont match!)
- Buttons: Save (disabled), Correct Pin (disabled)

Result: Test is Successful:

Planned Activities and Remarks:

This section provides the list of the activities done till date:

S.N	Project Activity	Remarks
1	Project Selection and Submission for approval	Done
2	Requirement Engineering	Done
3	Start of Interim Report	Done
4	Research for the Project	Done
5	Research on Methodology	Done
6	Start of Development	Started
7	Completion of Development	On going

SN.	Tasks	In Gantt Chart	In Actual	Remarks
1	Project Selection and Submission for approval	11/16/2017	11/14/2017	On Time
2	Requirement Engineering	11/13/2017	11/12/2017	On Time
3	Start Of Interim Report	11/17/2017	11/15/2017	On Time
4	Research on The Project	11/16/2017	11/15/2017	On Time
5	Research on Similar System	11/16/2017	11/15/2017	On Time
6	Research on Selected Methodology	12/16/2017	12/18/17	Late
7	Start of Development	12/29/16	12/25/2017	On time

Gantt Chart of Progress till date

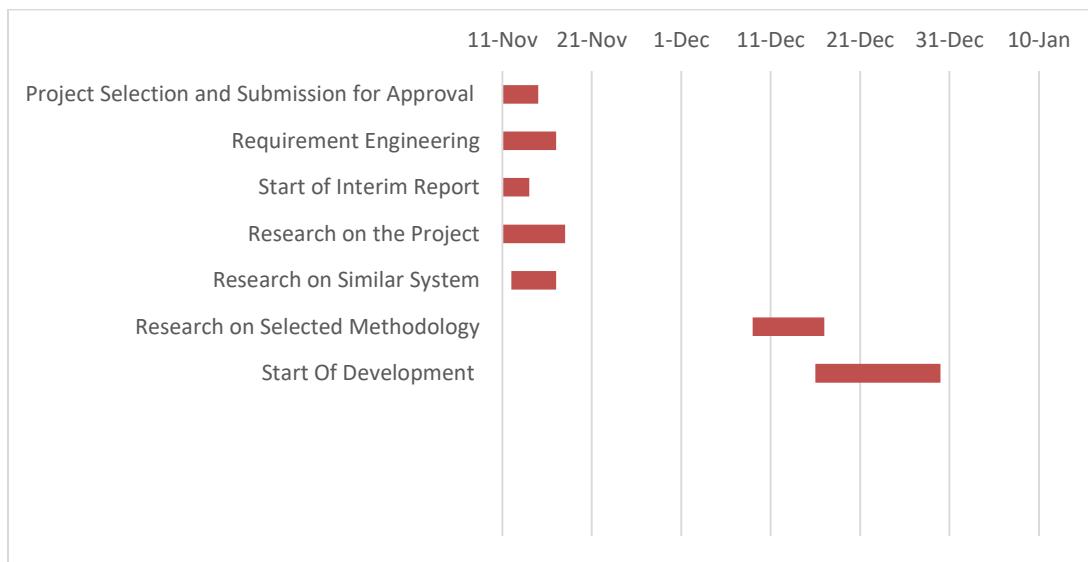


Figure 9: Gantt chart

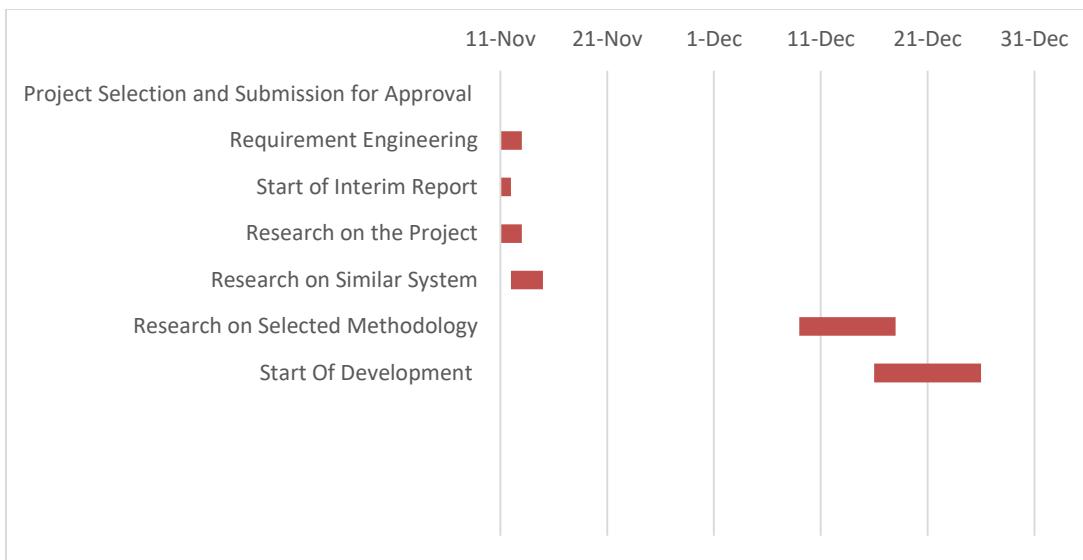


Figure 10: Revised Gantt chart