



Web Based Online Service Rental System

For

Smart Design & Construction (Pvt) Ltd

N.T Thamara Sugandhi

Index Number: 1002368

BIT Registration Number: R100236

Supervisor(s) :Mr. Bhagya Madhawa Ranasinghe

2017



This dissertation is submitted in partial fulfilment of the requirement of the
Degree of Bachelor of Information Technology (external) of the
University of Colombo School of Computing

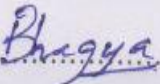
Declaration

“I Certify that this dissertation does not incorporate, without acknowledgment , any material previously submitted for degree or diploma in any university and to the best of my knowledge and belief, It does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans and for the title and abstract to be made available to outside organization”

Signature of Candidate:  Date: 28-09-2017

Name of Candidate: N.T.T Sugandhi

Countersigned by:

Signature of Supervisor:  Date: 28-09-2017

Name of Supervisor: Mr. Bhagya Madhawa Ranasinghe

Abstract

Construction industry is not yet fully digitalized In Sri Lanka and still practicing with conventional manual system throughout last few decades. Similarly, client of this project, who is Smart Design and Construction (Pvt) Ltd is also continuing their business with conventional manual system, which is driven by number of paper works on different format stored in several file and folders to manage their current activities. This process has been handled with heavy paper work currently and which is inefficient and time wasting.

While addressing existing issues in current system, this project was focused to introduce Web based online system for renting construction tools and equipment.

The developed Web based online rental system provides number of competitive advantages to the company to improve efficiency, productivity and which is ultimately caused to enhance profitability and reputation of the company. On the other hand, this web based online system was provided for more efficient, effective and user-friendly system to the customers to take necessary services provided by the system with new experience,

Comprehensive set of services were provided by this system including with customer registration, administration registration, supplier registration, stock management, products management, products delivery, generate reports to the right time, handling daily bills, handling daily rent products and also keep records of late payment charges etc. Critical advantage of this system is, that the system has a web shopping cart that enables the day to day sales functions.

Web based online service rental System developed based on the Rational Unified Process (RUP) framework. PHP which is an object oriented language which has been selected as the programming language to implement this system and NetBeans has been selected as the IDE (integrated development environment). PHP which is a server-side scripting language which is used with the MVC architecture and MySQL is used to handle databases. The system used Object oriented Techniques and Unified Modeling Language (UML) for Analysis and Design.

Acknowledgement

First of all I wish to thanks to the BIT Coordinator and Academic staff of University of Colombo school of Computing for giving me Valuable Project to gain knowledge through the BIT program.

Special thank g to my supervisor Mr. Bhagya Madhawa Ranasinge, Software Engineer of Softknowdge Business Solution (pvt) ltd. He gave me a technical guidance and always assist to how handle correct way Client despite his busy schedule.

My Sincere appreciation goes to the Staff of Smart design & Construction (pvt) ltd. They always provide me their company day to day details and Co-operated in the course of development of the Project.

I would also like to thank ESOFTE Computer Studies Institute and its respectable teaching panel for providing valuable knowledge and assistance throughout the BIT degree program.

I am also grateful to my family and friends especially to my parents, for supporting and encouraging me to complete this project successfully.

Table of Contents

Declaration	ii
Abstract	iii
Acknowledgement	iv
Table of Contents	v
List of Figures	viii
List of Tables	xi
List of Acronyms	xii
CHAPTER 01 – INTRODUCTION	1
1.1 Introduction.....	1
1.2 Motivation of the Project	2
1.3 Scope of the Proposed Project	2
1.4 Objectives of the System	3
1.5 Structure of the Dissertations.....	4
CHAPTER 02 – ANALYSIS.....	6
2.1 Introduction.....	6
2.2 Fact gathering.....	6
2.3 Analyzing Current Manual System.....	7
2.3.1 Customer Registration	8
2.3.2 Supplier Registration	8
2.3.3 Order Management	8
2.3.4 Product Management.....	8
2.3.5 Payments.....	8
2.3.6 Reports.....	8
2.4 Functional Requirements	9
2.4.1 Customer management	9
2.4.2 Supplier management	9
2.4.3 Product management	9
2.4.4 Order management	9
2.4.5 Payment management.....	9
2.4.6 Stock management.....	9
2.4.7 Report management	10
2.4.8 User management	10
2.4.9 Feedback.....	10
2.4.10 Delivery	10
2.5 Non Functional Requirements	10

2.5.1 Security.....	10
2.5.2 Usability.....	10
2.5.3 Accuracy.....	10
2.6 Existing Similar System.....	11
CHAPTER 03 – DESIGN.....	12
3.1 Introduction.....	12
3.2 System Development Life Cycle.....	12
3.3 Methodology for the Proposed System.....	12
3.4 Object Oriented Designing.....	13
3.4.1 High Level Use case Diagram for the Online Service Rental System.....	14
3.4.2 Class Diagram for the Online Service Rental System.....	15
3.5 Database Design.....	15
3.5.1 Database Normalization.....	16
3.6 Admin Panel Interface Design.....	16
3.6.1 User Login Interface.....	20
3.6.2 Dashboard Interface.....	20
3.6.3 Product management module Interface and Product Views.....	20
3.6.4 Admin management module Interface.....	21
3.6.5 Customer management module Interface.....	22
3.6.6 Supplier management module Interface.....	22
3.6.7 Stock management module Interface.....	22
3.6.8 Order management module Interface.....	23
3.6.9 Payment management module Interface.....	23
3.6.10 Feedback management module Interface.....	24
3.6.11 Delivery management module Interface.....	24
3.7 Website Interface Design.....	25
3.7.1 Login management module interface.....	25
3.7.2 Product management module interface.....	26
3.7.3 Shopping Cart page.....	27
3.7.4 Change web page content from Admin panel.....	28
3.7.5 Web pages.....	28
CHAPTER 04–IMPLEMENTATION.....	29
4.1 Introduction.....	29
4.2 Hardware and Software Requirements.....	29
4.2.1 Hardware Requirements.....	29
4.2.2 Software Requirements.....	29
4.3 Development Tools.....	30
4.3.1 Net Beans IDE 8.0.2.....	30

4.3.2 Codeigniter framework.....	30
4.3.4 Mysql.....	30
4.4 Major Code Segment	31
4.4.1 Model.....	32
4.4.2 Controller.....	32
4.4.3 View	33
4.4.4 Data validation.....	34
4.4.5 Preventing SQL Injection.....	34
4.4.6 Hiding PHP errors	35
4.5 Reused Existing Codes	35
CHAPTER 05 – EVALUATION	36
5.1 Introduction.....	36
5.2 Software Testing	36
5.3 Techniques of Software Testing	36
5.4 Types of Testing	37
5.4.1 Unit Testing.....	37
5.4.2 Integration Testing.....	37
5.4.3 System Testing	37
5.4.4 Acceptance Testing	37
5.4.5 Regression Testing	37
5.4.6 Beta Testing.....	37
5.5 Test case.....	38
5.6 Test plan.....	38
5.7 User Evaluation.....	41
CHAPTER 06–CONCLUSION.....	42
6.1 Introduction.....	42
6.2 Future Enhancement	42
6.3 Lesson Learnt.....	42
References.....	43
Appendix A- SYSTEM DOCUMENTATION	44
Appendix B –DESIGN DOCUMENTATION	47
Appendix C- USER DOCUMENTATION	61
Appendix D-MANAGEMENT REPORTS.....	71
Appendix E –TEST RESULTS	74
Appendix F-CODE LISTING	81
Appendix G – CLIENT CERTIFICATE.....	85
Glossary	85
Index	87

List of Figures

Figure 2. 1 Use case diagram for current manual system.....	7
Figure 3. 1 High level use case diagra.....	14
Figure 3. 2 Class diagram.....	15
Figure 3. 3 Database design	15
Figure 3. 4 Tab index.....	17
Figure 3. 5 Image link to pages	17
Figure 3. 6 Main url in admin panel.....	17
Figure 3. 7 Save successfully message.....	18
Figure 3. 8 Please try again message.....	18
Figure 3. 9 Validation message.....	18
Figure 3. 10 Select date from calendar.....	18
Figure 3. 11 Combo box.....	19
Figure 3. 12 Customized textarea.....	19
Figure 3. 13 About us content change form	19
Figure 3. 14 Login interface	20
Figure 3. 15 Dashboard interface	20
Figure 3. 16 Add new product interface	21
Figure 3. 17 View products interface	21
Figure 3. 18 Add new administrator interface.....	21
Figure 3. 19 Add new customer interface.....	22
Figure 3. 20 Add new supplier interface	22
Figure 3. 21 Add new stock interface.....	23
Figure 3. 22 View order details	23
Figure 3. 23 View income report.....	24
Figure 3. 24 View feedback details	24
Figure 3. 25 Add delivery details	24
Figure 3. 26 Website registration interface.....	25
Figure 3. 27 Website login interface.....	25
Figure 3. 28 Website product page.....	26
Figure 3. 29 Website single product page.....	26
Figure 3. 30 Add to shopping cart view	27
Figure 3. 31 Checkout process page.....	27
Figure 3. 32 Payment confirmation status.....	27

Figure 3. 33 About us admin panel form.....	28
Figure 3. 34 About us website page	28
Figure 3. 35 Website home page	28
Figure 3. 36 Website contact us page	28
Figure 4. 1 Show MVC Architecture.....	31
Figure 4. 2 Layers of MVC architecture	31
Figure 5. 1 User evaluation form.....	41
Figure A. 1 XAMPP setup.....	44
Figure A. 2 Select components.....	44
Figure A. 3 Choose folder path on disk C	45
Figure A. 4 Run XAMPP panel.....	45
Figure A. 5 Tables list in smart_db database	46
Figure B. 1 Use case diagram of the product management module	47
Figure B. 2 Use case diagram of the report management module.....	48
Figure B. 3 Use case diagram of the customer management module.....	49
Figure B. 4 Use case diagram of the Supplier management module.....	50
Figure B. 5 Use case diagram of the Order management module.....	50
Figure B. 6 Use case diagram of the Stock management module	51
Figure B. 7 Use case diagram of the User management module.....	53
Figure B. 8 Use case diagram of the Login management module.....	53
Figure B. 9 Use case diagram of the feedback management module.....	54
Figure B. 10 Use case diagram of the Delivery management module	55
Figure B. 11 Use case diagram of the Payment management module	55
Figure B. 12 Activity diagram of the Customer management module.....	57
Figure B. 13 Activity diagram of the Stock management module	58
Figure B. 14 Activity diagram of the Product management module	59
Figure B. 15 Sequence diagram of the Customer management module	60
Figure B. 16 Sequence diagram of the Payment management module	60
Figure C. 1 Login Form.....	61
Figure C. 2 Dashboard Form.....	61
Figure C. 3 Main menu	62
Figure C. 4 Sub menu items	62
Figure C. 5 User management form	62
Figure C. 6 Customer management form	63
Figure C. 7 View Customer management table.....	63
Figure C. 8 Product management form	64
Figure C. 9 View Product management table.....	64
Figure C.10 Stock management form.....	65

Figure C.11 View Income report	65
Figure C.12 View Feedback management table	66
Figure C.13 Delivery management form.....	66
Figure C.14 Website register form	67
Figure C.15 Website login form	67
Figure C.16 Website products page.....	68
Figure C.17 Display single products page.....	68
Figure C.18 Add to shopping cart view.....	69
Figure C.19 Checkout process.....	69
Figure C.20 About us website page.....	70
Figure C.21 Website home page.....	70
Figure C.22 Website contact us page.....	70
Figure D. 1 Product report.....	71
Figure D. 2 Category report	71
Figure D. 3 Stock report.....	72
Figure D. 4 Income report	72
Figure D. 5 Customer report.....	73

List of Tables

Table 5.1 High level Test plan.....	41
Table B.1 Use case description for Product management module	47
Table B.2 Use case description for Show product.....	48
Table B.3 Use case description for Generate Reports	48
Table B.4 Use case description for Customer module.....	49
Table B.5 Use case description for Customer order products	49
Table B.6 Use case description for Supplier module.....	50
Table B.7 Use case description for Order module.....	51
Table B.8 Use case description for Stock module	52
Table B.9 Use case description for Damaged stock module	52
Table B.10 Use case description for User module	53
Table B.11 Use case description for User module.....	54
Table B.12 Use case description for Feedback module.....	54
Table B.13 Use case description for Delivery module	55
Table B.14 Use case description for payment module-Customer	56
Table B.15 Use case description for payment module-Administrator.....	56
Table E.1 Test cases of Product module	75
Table E.2 Test cases of User module.....	75
Table E.3 Test cases of Customer module	76
Table E.4 Test cases of Stock module	78
Table E.5 Test cases of Feedback module	78
Table E.6 Test cases of Report module.....	79
Table E.7 Test cases of Payment module	79
Table E.8 Test cases of Order module	80
Table E.9 Test cases of Delivery module	80

List of Acronyms

AJAX	- Asynchronous JavaScript and XML
CD ROM	- Compact Disc Read Only Memory
CSS	- Cascading Style Sheet
1NF	- First Normal Form
2NF	- Second Normal Form
3NF	- Third Normal Form
DBMS	- Database Management Systems
ER	- Entity Relationship
GUI	- Graphical User Interface
HTML	- Hypertext Mark-up Language
MVC	- Model-View-Controller
OO	- Object Orientation
PDF	- Portable Document Format
PHP	- Hypertext Pre-processor
RAD	- Rapid Application Development
RAM	- Random Access Memory
RUP	- Rational Unified Process
SQL	- Structured Query Language
UML	- Unified Modelling Language
URL	- Uniform Resource Locator
WYSIWYG	- What You See Is What You Get

CHAPTER 01 – INTRODUCTION

1.1 Introduction

Web Based Online Service Rental System Project is a process of transferring thousands of paper work into a reality through effective management of time, budget, technology, infrastructure and environmental conditions according to the regulations of local authorities without damaging the sustainability of the green environment. The ideas on two-dimensional sketches and texts on thousands of papers will be converted into actual three-dimensional models in the space with adequate strength, appearance, durability and as well as functionality.

This software project is referred to online service rental system and the client of this project is Smart Design and Construction (Pvt) Ltd. For running the business forward in this highly competitive industry, there should be some accurate and faster tools for pricing, profitability analyzing and project management. This software project is mainly focus on the financial aspect of online service rental process. Pricing of a project is the riskiest part of the project. Although most challengeable part of a project was management. There are number of popular tools for project management. However, the pricing strategies will change by company to company. Therefore, it can be recommended to have specific customized software for each construction organization for pricing.

There are three main parties like client, owner and supplier who are basically contributed to take a successful output of a online service rental process. The process is initiated with a requirement of client. Owner is the one who owns all the services project. Suppliers will be selected by client to hire a service.

The main three parties in online service rental process (client, owner and supplier) are working together through out several years form difference distance location to realize the concept. So when these three parties get together correctly a successful online service rental is completed.

1.2 Motivation of the Project

Smart Design & Construction (Pvt) Ltd is a newly formed company .So the company is engaging with construction related activities such as building construction, building modification & Supply their Construction equipment as a rental service. Many customers use their rental service to get what they want for a limited time.

Still Smart design & construction (pvt) ltd is using a manual system to manage their customer details, Prices, Delivery areas etc. According to the Business which is related to this company they can't spend more time for customers and their stocks. So my suggestion for this company is to develop an Online rental system. So they can record all the necessary details without using their manual system. After considering my proposal Smart design & Construction (pvt) ltd came to a conclusion that they needed a Online web based service rental system.

The significant increase in usage of Internet and Web based technologies by industries, led me to implement a web based system based on MVC architecture. By implementing this project, I got the opportunity to apply the theoretical and practical knowledge, technologies and experiences gathered throughout the BIT (Bachelor of Information Technology) Degree program to a real world situation. The Web Based Online Service Rental System will facilitate in solving certain problems in the company.

1.3 Scope of the Proposed Project

Scope of the project can be listed in point form.

The web based Online service rental system can help customers to buy a service quickly rather than coming to Smart design & Construction (pvt) ltd .This System help to minimize wasting time ,effort and resource and also make the system secure.

The System helps Smart design & Construction (pvt) ltd to keep track of their services without using manually created document. This System helps to keep stock levels up to date, handle customer login details as well as customer payment details, to maintain company website regularly, handle supplier details, keep customer and supplier area records, handle online payment gateway when buying products, generate various kind of reports, file handling, sending Notifications to various parties.

It also keep track of the rented goods so that the company can have more details about their stocks.

The main feature of the system is buying the services/products using the Online shopping cart. So it will help the customer to buy services/products without break a sweat.

Modules:

- Customer management
- Supplier management
- Product management
- Order management
- Payment management
- Stock management
- Report management
- User management
- Feedback management
- Delivery management

1.4 Objectives of the System

Main objective of the project can be listed in point form

- Allow different system users to access own privilege levels.
- To provide informative data.
- Update and Maintain product details, Supplier details, Customer details, Payment details, Stock details and User account privilege details etc.
- Maintain Customer and Supplier relationship.
- Maintain day to day product stock levels.
- Enable customer and product manager to view stock details.
- Provide Secure Online payment to customer.
- Customer can add orders to shopping cart.
- Order notifications send to Customers, suppliers and company staff.
- Provide delivery Availability to different areas customers.

- Provide Reports and Invoice to PDF, Table and Graph formats and also get a Print reports.
- Maintain file handling.
- Maintain Stock balance and always update the Stock details.
- Provide user-friendly interface to external and internal users.
- Provide quality service to users.

1.5 Structure of the Dissertations

This Dissertation provides the overall knowledge about the Online Service Rental System. Dissertation structure as follows.

CHAPTER 02 – ANALYSIS

This chapter is provide the detailed and specific requirements of the system plus explanation of design decisions and made are stated. Current situation of the system with reference to requirement analysis, project goals and limitation of goals with respect to the system environment will also be described and Existing similar systems are discussed here. Description of the prerequisites that must apply for the system to be used (called success factors), specification of the number of users, the frequency of use, and the jobs of the users, and example will be described. Functional requirements covering system functionality expected by the users and non-functional requirements covering reliability, portability, and response and processing times will be addressed with detailed justification.

CHAPTER 03 – DESIGN

The structure of the overall system will be clear discussed in this chapter including methodical approach to the design of the system, alternate solutions and the one selected. And there will be explanation and justification with respect to the selection. Coherent and logical arguments are encouraged. The correct use of appropriate tools and techniques will be demonstrated. User interface design will also be discussed in this chapter.

CHAPTER 04 – IMPLEMENTAION

This chapter will be described the implementation of the system with the identification and explanation of all major code and module structures. Also, the implementation environment (hardware and software), any existing code that was reused, development tools used, and platform dependence will be discussed.

CHAPTER 05 – EVALUATION

This chapter will be provided detail about testing of system was done including with comprehensive test plan that was used to verify and validate the system. This will be included detail of wide range of test data. Evidence will be provided to show that all aspects of the system have been tested and specification has been met. Descriptions of the effects of various kinds of errors and the required system behavior upon occurrence of an error have included. The evaluation of the system with potential users will also be provided here.

CHAPTER 06 – CONCLUSION

This chapter will conclude the dissertation with a critical evaluation of the system and suggestions for future work. The chapter will discuss extend of the project objectives were satisfied and the reasons for them. Experience gain during the course of the project will also be expanded on this chapter.

APPENDIXES

These are provided further details about the content of the dissertation chapters which were not included in the chapters.

CHAPTER 02 – ANALYSIS

2.1 Introduction

This chapter provides the detailed and specific requirements of the system plus explanation of design decisions and made are stated. Initial situation of the system with reference to requirement analysis, project goals and limitation of goals with respect to the system environment will also be described and Existing similar systems are discussed here. Description of the prerequisites that must apply for the system to be used (called success factors), specification of the number of users, the frequency of use, and the jobs of the users, and example will be described. Functional requirements covering system functionality expected by the users and non-functional requirements covering reliability, portability, and response and processing times will be addressed with detailed justification.

2.2 Fact gathering

Fact gathering technique is most critical part of the project. It using analysis phase. There are different way to gather information of the system. In this project facts were gather in proper way which use suitable fact gathering techniques. These are following,

- Interviews
- Observation
- Questionnaires
- Company Documents.

Interviews are most used technique in requirement gathering. Requirements gather from company staff and Administrator from face to face. And also company manager support to it. We carried it structural way and unstructured way. Sometime it was hard to collect requirements .Because most company staff were new to office.

Most Important thing is staff feedback and questions were gathered through in this fact gathering techniques.

Also used observation technique to requirements gather. But it was difficult to gathered requirements. Because existing system was very complex and hard to understand. It was not used in proper way.

Questionnaires also done by interviews. We can gather clear requirements from that also.

Most advantage face gathering technique was company documents. We can gather clear and understandable facts from company documents. Also we can get clear understand about company current situation, when we confuse through observation of existing system. We collect some samples like company bills, Invoices, company repots etc.

2.3 Analyzing Current Manual System

Figure 2.1 Use case Diagram of Current Manual System

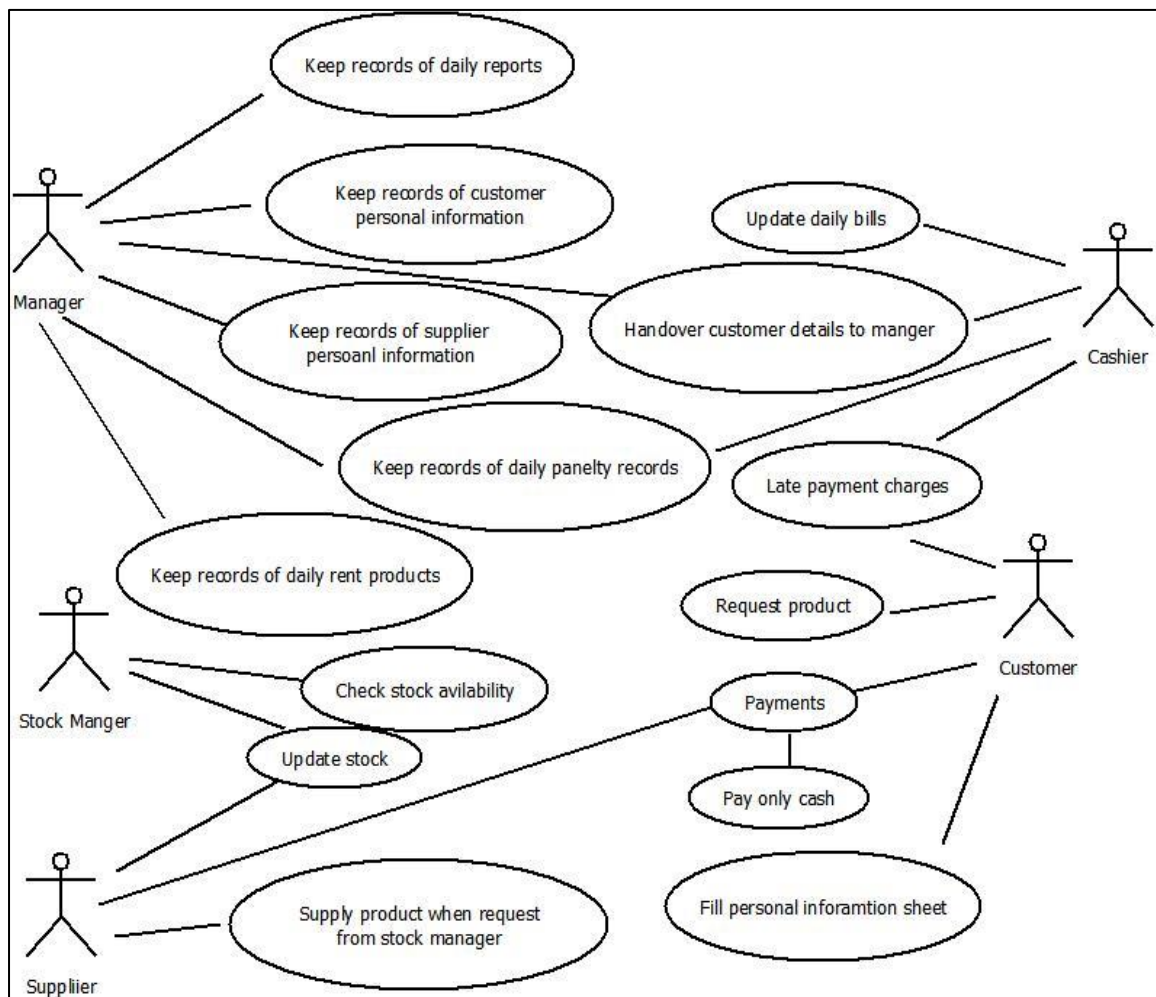


Figure 2. 1 Use case diagram for current manual system

2.3.1 Customer Registration

There are few customers to handle this manual system. Most of them are close friends, neighbours, and related companies officers. Therefore, there are no systematic way to handle these customer base. They fill their personal details before rent products. The manual system store customer name, customer address, customer telephone numbers and amount of products.

2.3.2 Supplier Registration

Mainly they store supplier personal details in current manual system. They store Supplier name, Supplier Address, Supplier telephone no and Product details.

2.3.3 Order Management

Order details are saved in to an excel sheet manually when an Order is completed. At the end of the day company manager look at those excel reports to make decisions.

2.3.4 Product Management

Every product details are saved into the excel sheet and update manually. Stock details are also updated manually. Less details are kept due to manual updating.

2.3.5 Payments

Payments are done by cash only. Penalties are calculated manually.

2.3.6 Reports

At the end of the month all the excel sheets are analysed by the manager to make company decisions.

2.4 Functional Requirements

What the system is supposed to achieve, is known as functional requirements. Functional requirement can be a calculation, technical detail, etc. Following are the functional requirements that have been identified through the system analysis. WIKI [1]

2.4.1 Customer management

Customer can sign in to dashboard. And have to require fill personal details. Such as customer name, customer address, customer telephone number, customer email address etc. Customer can be able to update, edit, delete, add and view product details.

2.4.2 Supplier management

Administrator can store all the supplier details using dashboard. By doing so all the details of suppliers will be at one place and will be able to all the necessary details about them quickly. Supplier name, address, phone number, email address etc information will be stored in the database.

2.4.3 Product management

All products need to be saved in the database. Products are saved based on category and sub category. So categories and sub categories need to be saved separately. Individual product prices also need to be saved with delay penalty.

2.4.4 Order management

All the order details need to be saved with all the customer details and prices. Eventually those details need to be shown in reports.

2.4.5 Payment management

Payments are calculated after an order is done. All invoice details are captured so that at the end payments can be done. All the payments are done using pay pal or cash payments.

2.4.6 Stock management

Stock needs to be maintained properly as the product quantity always changes due to renting process. When a supplier brings the products, they are added to the system by the administrator. If any damages were found stock need to be updated also.

2.4.7 Report management

Reports are vital when making a decision in a company. All the invoice reports, order reports, stock reports need to be generated so that company management can take a decision.

2.4.8 User management

In this user management module we manage 3 kind of users. Suppliers, Customers and system admin. System admin can create users and suppliers. Users can buy stuff buy using the web site.

2.4.9 Feedback

Feedback module can be used to capture customer reviews so that more people will be able to buy the correct product.

2.4.10 Delivery

Delivery module is to capture whether the product has been successfully delivered to the customer or not.

2.5 Non Functional Requirements

Non Functional requirements are also important facts which we need to consider when developing the system. Nonfunctional requirements describe the system properties and constraints of the system. Those requirements are applied to the whole system, not only for the individual parts of the system. If we do not consider about nonfunctional requirements system can be useless. WIKI [2]

2.5.1 Security

System should be having more security. More sensitive data like customer details, supplier details, and payment details need to be stored safely.

2.5.2 Usability

Usability requirement has been gained by using various methods such as mobile friendly menu navigation and responsive interfaces and use of matching colors.

2.5.3 Accuracy

Validation techniques are used so that the correct data type is stored on the database. Otherwise there will be errors caused in the system.

2.6 Existing Similar System

When considering the Online service rental systems for a product rent services, there is no open source or propriety software in the web. Most of the systems don't include any data handling or processing systems. They are only including the detail viewing for the outside viewers. WIKI [3]

Online shopping cart systems are also available of different e-commerce websites. There were different options to match with their client requirements. There were bulks of unnecessary functions which are not related to the proposed systems, so that the best option is to make bespoke software to match with the client's requirements.

CHAPTER 03 – DESIGN

3.1 Introduction

In software designing, System should have good design before implementing. System design provides the detail and description of the system including components, interface, modules, architecture and data. WIKI [4]

System designers may use various system design approaches. Such as modern structured design, prototyping, JAD, RAD and object-oriented design.

Prototype model is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements.

Joint application design (JAD) is a process used in the life cycle area of the dynamic systems development method (DSDM) to collect business requirements while developing new information systems for a company.

Rapid Application Development (RAD) is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype.

Object-oriented designing is the widely used approach in modern software designing. The work has done in a project can be easily reuse than in the other approaches. Also when doing the coding will be easier to write, easier to understand and will contain fewer errors.

3.2 System Development Life Cycle

SDLC, Software Development Life Cycle is a process used by software industry to design, develop and test high quality systems. The SDLC aims to produce a high quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

3.3 Methodology for the Proposed System

RUP (Rational Unified process) has been selected for the proposed system. Reasons of selecting RUP are it is an iterative software development process framework and it supports object oriented development. Initially client does not have a clear idea about system requirements, so requirements may likely to be changed. Further system has been divided into modules such as Reservation module, Master Module and vehicle allocation module etc. Each module has developed incrementally and iteratively. Therefore RUP is the most appropriate SDLC (Software Development Life Cycle) methodology. Inception, Elaboration, Construction and Transition are the main phases of the RUP lifecycle. Brief description is given below. WIKI [5]

Inception

This phase is oriented to achieve agreement between customer and team on requirements, the expectation for the functionality, and the procedural approach that is to be taken to implement the project.

Elaboration

This phase is to construct an architectural baseline that implements a working application with limited functionality, and to formulate an agreement with customer to further pursue the project.

Construction

This phase is oriented to supporting documents, software product, test suite and user manual. Objective here is to understand all requirements and finish development based on baseline architecture.

Transition

Transition starts after initial product testing has been performed and the system has been finally deployed. Its objective is to assure software is available to end user.

3.4 Object Oriented Designing

Object-oriented design is concerned with developing an object oriented model of a software system to implement the identified requirements. The main goal of this design approach is defining objects define objects and establishing relationship between classes by sending and receiving messages. [Whitten, 2007]

Object-oriented design is the process of planning a system of interacting objects for the purpose of solving a software problem. It is one approach to software design.

Figure 3.1 shows High Level Use case Diagram for the Online Service Rental System

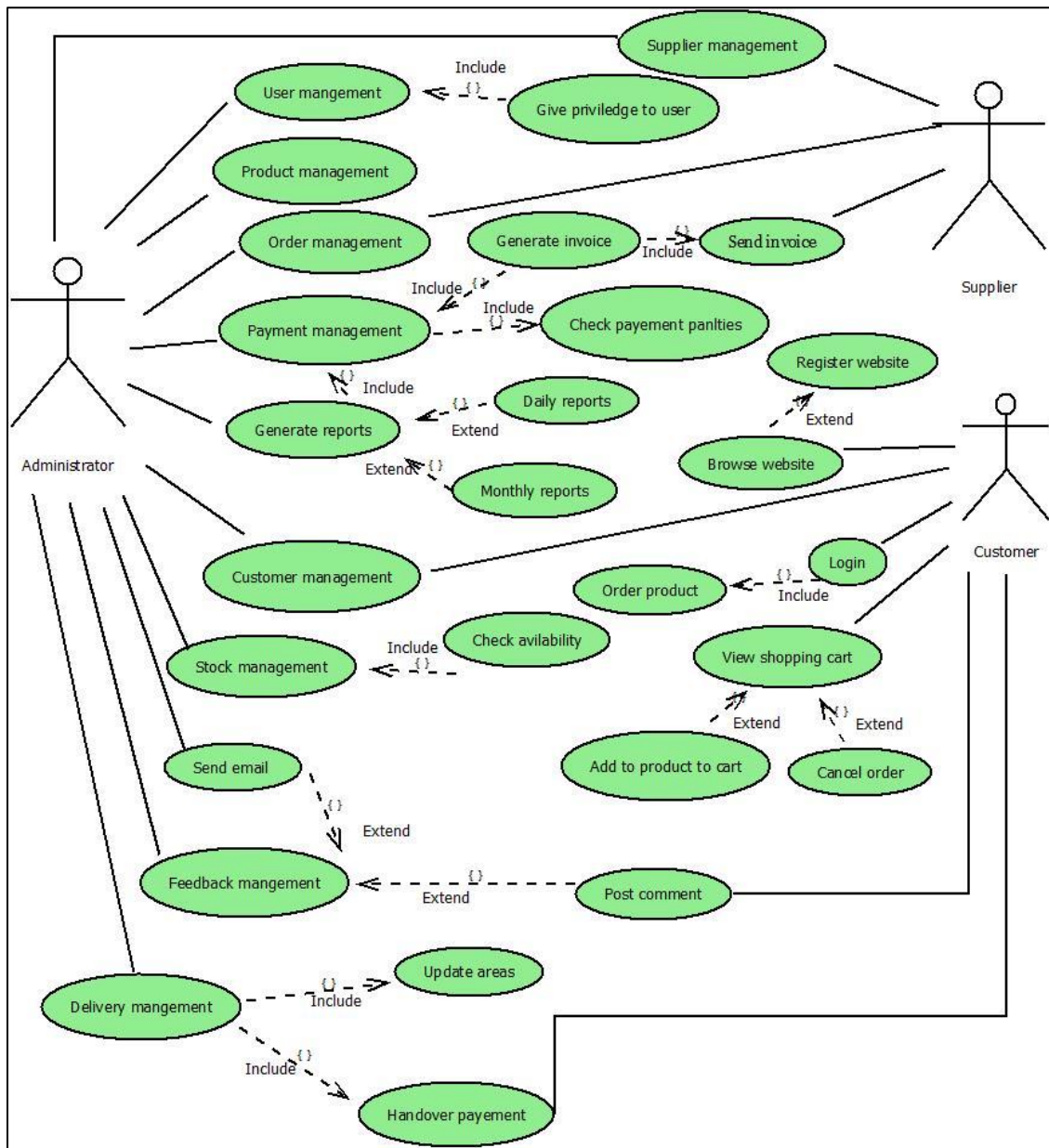


Figure 3.2 shows Class Diagram for the Online Service Rental System

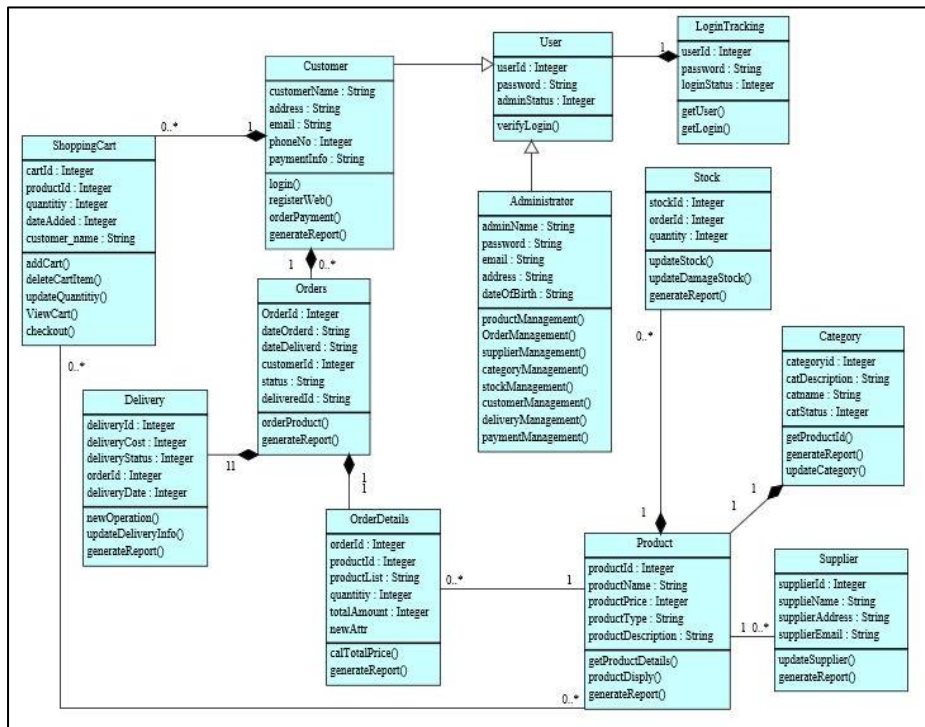
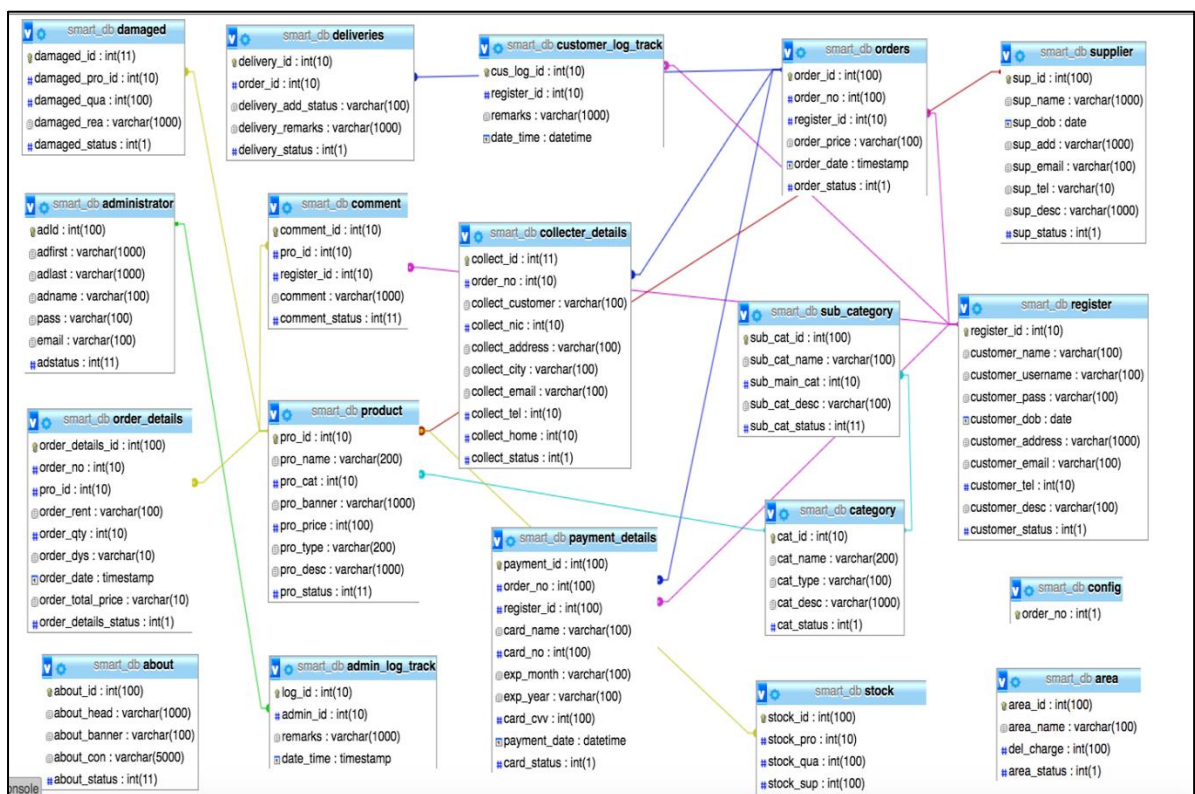


Figure 3.3 shows Database Design for the Online Service Rental System



Database design is the process of producing a detailed data model of database . Conceptual design, logical design and physical design are the 3 main sections in database design. WIKI [6]

3.5.1 Database Normalization

Database normalization, or simply and organizing, is the process of organizing the columns (attributes) and tables (relations) of a relational database to reduce data redundancy and improve data integrity. WIKI [7]

1st Normal Form

There are 3 rules in 1NF.

1. Define the data items required, because they become the columns in a table. Place related data items in a table.
2. Ensure that there are no repeating groups of data.
3. Ensure that there is a primary key.

2nd Normal Form

Second normal form states that it should meet all the rules for 1NF and there must be no partial dependences of any of the columns on the primary key.

3rd Normal Form

A table is in third normal form when it is in second normal form and non primary fields are dependent on the primary key.

3.6 Admin Panel Interface Design

User interfaces are design by integrating all necessary information closely so that all the user can easily browsed and acquire necessary information efficiently. The numbers of forms are limited as much as possible in the entire module. All detail is integrated into minimum numbers of forms. So user can scroll down can reach to required information. Since this is a professional web site, visualization and animation is limited so as to get the user attraction directly into information.

Following are the user interface design principles [Sommerville, 2007]

- User Familiarity

The interface should use terms and concepts drawn from the experience of the people who use the system.

- Consistency

The interfaces should be consistent in that wherever possible, comparable operations should be activated in the same way.

- Minimal surprise

Users should never be surprised by the behavior of the system.

- Recoverability

The interfaces should include mechanisms to allow users to recover from errors.

- User guidance

The interface should provide meaningful feedback when errors occur and provide help facility.

- User Diversity

The interface should provide appropriate interaction facilities for different type of users. [Sommerville, 2007]

This system uses good features and user friendly interfaces throughout whole system, They are,

- Figure 3.4 show tab indexes for navigation.



Figure 3.4 Tab index

- Used light green color for whole system. Light green is Eye friendly color to any user.
- Tab button and Enter button are used for enter user details and save details without click the mouse. It is easy for users when they are busy.
- Figure 3.5 shows image url and figure 3.6 shows menu url were used for easy access. It is increase effectiveness of system.

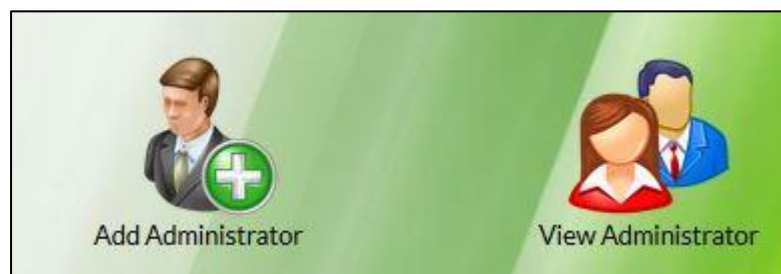


Figure 3.5 Image link to pages

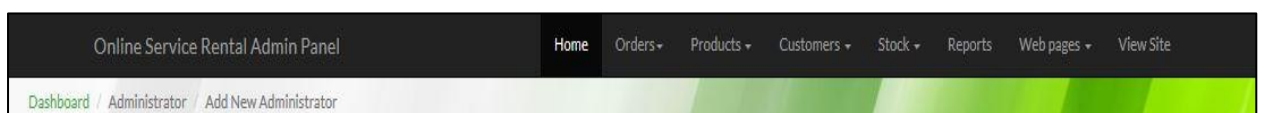


Figure 3.6 Main url in admin panel

- After saving forms display successful messages in every time.



Figure 3.7 Save successfully message

- System show error messages when we enter invalid data to the system. And also if we skip necessary fields in form system show messages to users. Then we can identify empty fields in forms. Figure 3.8 and figure 3.9 show error message and system validation forms.



Figure 3.8 Please try again message



Figure 3.9 Validation message

- System used calendar to display dates and stored dates with form. Calendar is very important to show reports. Administrator can select date range and he can view selected date range activates in reports.

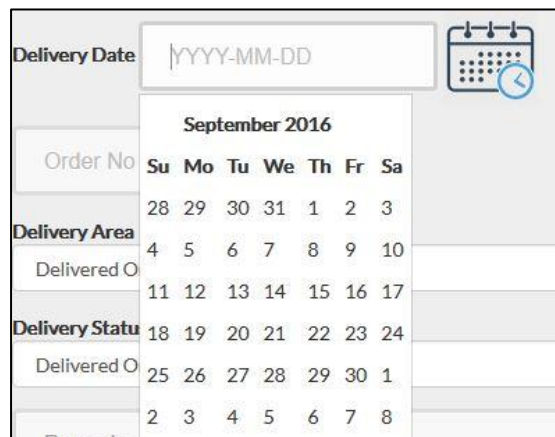
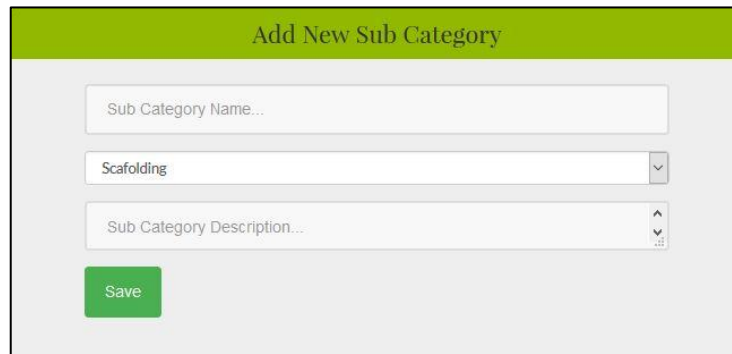


Figure 3.10 Select date from calendar

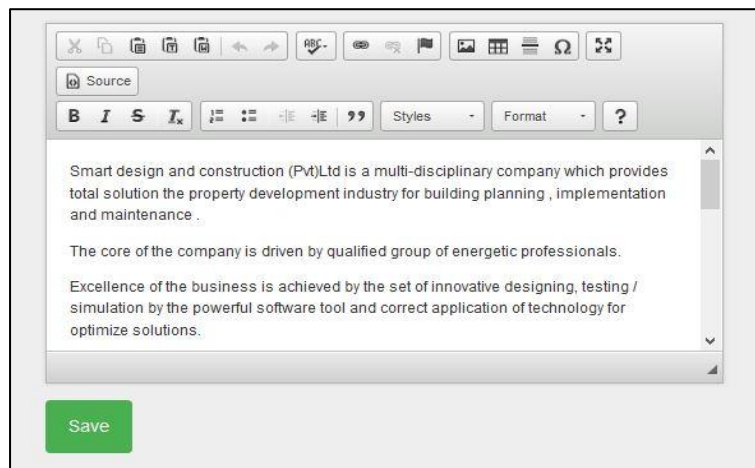
- Figure 3.11 shows combo box to retrieve data from another form or database.



The screenshot shows a web form titled "Add New Sub Category" with a green header. It contains three input fields: "Sub Category Name..." (text), "Scaffolding" (dropdown menu), and "Sub Category Description..." (text). A green "Save" button is located at the bottom left.

Figure 3.11 Combo box

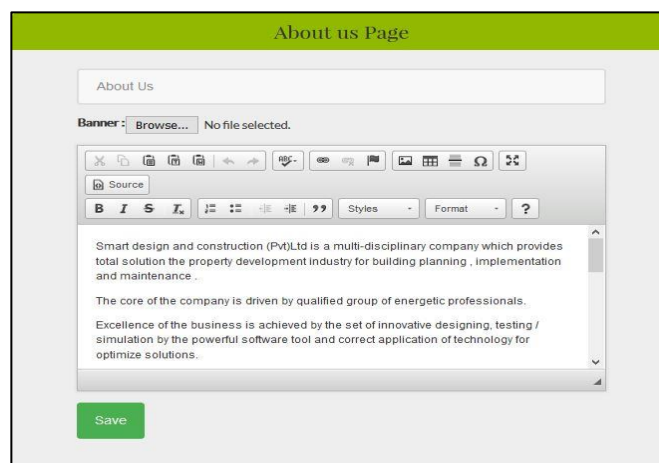
- Figure 3.12 show customize text area for increase the user-friendliness.



The screenshot displays a rich text editor interface. It features a toolbar with icons for cut, copy, paste, undo, redo, bold, italic, strikethrough, text color, background color, bulleted list, numbered list, link, unlink, indent, outdent, quote, styles, and format. Below the toolbar is a text area containing three paragraphs of text. A green "Save" button is positioned at the bottom left.

Figure 3.12 Customized textarea

- And also administrator can change contents of website anytime without knowing the programming language, He can do it from admin panel. Therefore he didn't want to contact system designer for change the contents of website. Figure 3.13 show website about page in admin panel.



The screenshot shows an admin form titled "About us Page" with a green header. It includes a text input field for "About Us", a "Banner:" label with a "Browse..." button and "No file selected." text, and a rich text editor with the same toolbar as in Figure 3.12. The text area contains the same three paragraphs of text. A green "Save" button is at the bottom left.

Figure 3.13 About us content change form

3.6.1 User Login Interface

Only administrators can log into admin panel. If users details are wrong it will not allow to log in. Figure 3.14 shows login interface.

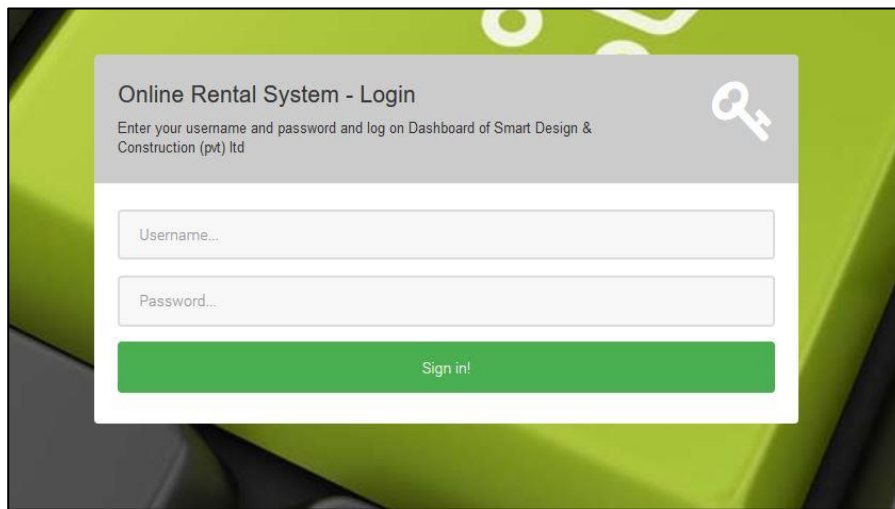


Figure 3.14 Login interface

3.6.2 Dashboard Interface

Figure 3.15 shows Dashboard. It has all modules in system. This can facilitate direct access to modules forms.



Figure 3.15 Dashboard interface

3.6.3 Product management module Interface and Product Views

Product form used to add product, edit product, update product and Delete product. It load all categories from category table to the combo box. This form facilitated to store image of the products. Further After save the product details user can view the all products view product image. If user want to edit product, system redirect edit product page. And updating product.

And most important thing is when user update product details on product form , it automatically update on website. Therefore website will be update automatically. So that user

can easily manipulate and view data easily. Figure 3.16 and figure 3.17 shows add product form and view all products form.

Figure 3.16 Add new product interface

ID	Product Category	Product Sub Category	Product Name	Product price	Product Description	Edit	Delete
1	Scaffolding	Single scaffolding	Mason scaffolding	Rs.500.00	Saferstack 5 ft * 5 ft * 7ft		
2	Scaffolding	Double scaffolding	Meta-tech scaffolding	Rs.450.00	10ft * 7ft * 5ft scaffold double lift		
3	Scaffolding	Double scaffolding	Double lift scaffolding	Rs.400.00	10ft * 7ft * 5ft Aluminum double lift scaffolding		
4	Hand tools	Pliers	6 pliers set	Rs.250.00	8 inch diagonal cutting pliers		
5	Hand tools	Hammer	Sledge hammer	Rs.170.00	Power strike sledge with fiberglass hammer		
6	Hand tools	Hammer	Framing hammer	Rs.350.00	14 inch fiber glass handle		

Figure 3.17 View products interface

3.6.4 Admin management module Interface

Figure 3.18 represent the administrator login page which can access admin panel. Administrator must have to fill the form. When type the passwords system automatically check the both passwords .If one password is not match with previous, Administrator have to recheck it. And also this form show all administrator names in view administrator image icon

Figure 3.18 Add new administrator interface

3.6.5 Customer management module Interface

Figure 3.19 represent customer management form which can update customer base on business. When customer want to rent a product through online he have to fill his personal details from here. And system administrator can view all the customer details from this customer form. Well maintained customer base is very useful for send notification , send emails , and promote product deals.

System calculate discounts for existing customers. Therefore customer form is main part of this system. Customer use in various phases in this system.


The screenshot shows a web interface for adding a new customer. At the top, there are two buttons: 'Add Customer' with a person icon and a plus sign, and 'View Customer' with a person icon and a magnifying glass. Below these is a green header bar with the text 'Add New Customer Details'. The main form area contains several input fields: 'Customer full Name' with a text box and a small person icon; 'Date of Birth' with a date picker showing 'mm/dd/yyyy'; 'Telephone No' with a text box and a telephone icon; 'Email' with a text box and an envelope icon; 'Address' with a text box and a house icon; and 'Description' with a text box and a dropdown arrow. A green 'Save' button is located at the bottom left of the form.

Figure 3.19 Add new customer interface

3.6.6 Supplier management module Interface

Figure 3.20 represent Supplier management form which can update supplier base on business. System must have to maintain supplier database. Because it will be help to inform about their products. Supplier_id connect with product table, stock table, damage stock table, order table and also administrator table. Supplier character is main role of the system.

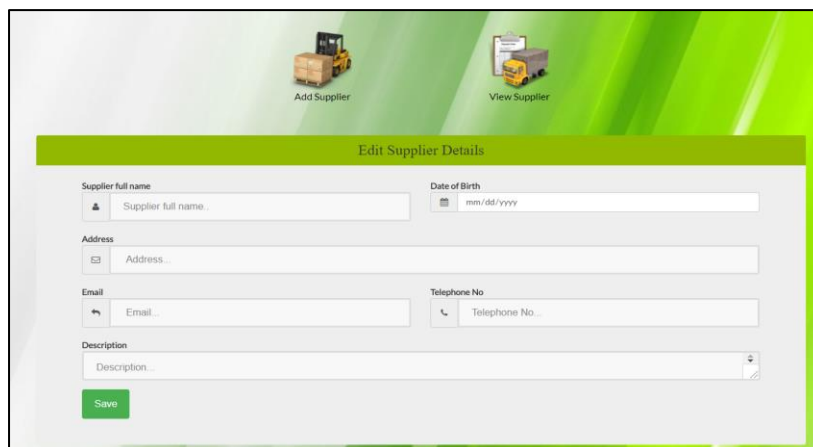
The screenshot shows a web interface for editing a supplier. At the top, there are two buttons: 'Add Supplier' with a forklift icon and 'View Supplier' with a truck icon. Below these is a green header bar with the text 'Edit Supplier Details'. The main form area contains several input fields: 'Supplier full name' with a text box and a small person icon; 'Date of Birth' with a date picker showing 'mm/dd/yyyy'; 'Address' with a text box and a house icon; 'Email' with a text box and an envelope icon; 'Telephone No' with a text box and a telephone icon; and 'Description' with a text box and a dropdown arrow. A green 'Save' button is located at the bottom left of the form.

Figure 3.20 Add new supplier interface

3.6.7 Stock management module Interface

Figure 3.21 represent Stock management form which can update product stock, update return stock and update damaged stock. Administrator can add stocks with quantity and supplier name. And if he want to see return stock, he has to enter order no ,then display return stock.And also there is damage stock form. If there is damage product ,stock automatically update and display update product quantity in shopping cart page.

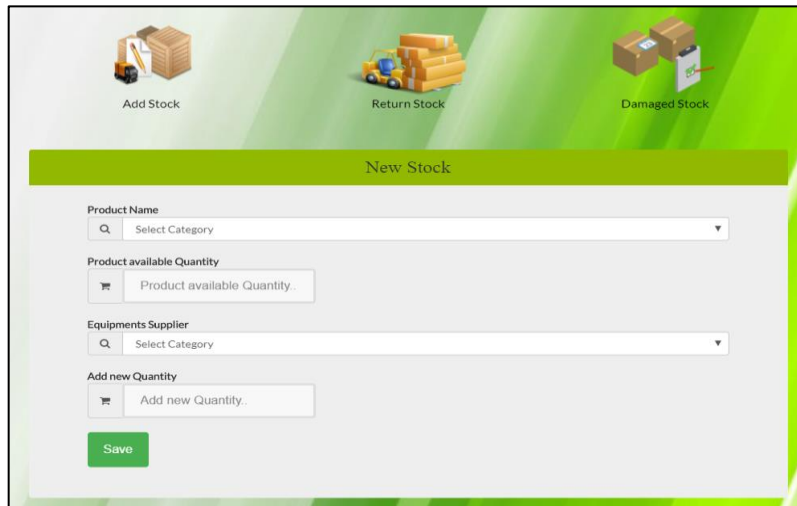


Figure 3.21 stock interface

3.6.8 Order management module Interface

Figure 3.22 (Admin Order screen) represent the order module form which can be viewed by the admin. Admin can view orders status with this module. Admin is able to view all the order status. And he can cancel order if customer didn't pay.

Order No	Customer Name	Order List	Order Date Time	Order Quantity	Total Order Price	Delete
ODR11	Sanka Pathama	Mason scaffolding	2017-11-27 19:32:58	2	Rs.2,000.00	
ODR12	Ruwan Wickramasinghe	Yellow hats	2017-11-29 10:52:01	10	Rs.20,000.00	
ODR13	Udesh Malinga	Red stop sign	2017-11-29 10:58:48	32	Rs.96,000.00	
ODR14	Nuwan Ranasinghe	Chop saw	2017-11-29 13:53:06	1	Rs.1,050.00	
ODR18	Thamara Sugandhi	Mason scaffolding	2017-12-01 00:10:30	1	Rs.500.00	
ODR18	Thamara Sugandhi	Black helmets	2017-12-01 00:10:30	1	Rs.200.00	
ODR21	Idunil Bandara	Black helmets	2017-12-01 15:11:11	5	Rs.2,000.00	
ODR21	Idunil Bandara	Chop saw	2017-12-01 15:11:11	1	Rs.700.00	
ODR22	Kapila Athanayaka	Sledge hammer	2017-12-01 17:41:50	12	Rs.6,120.00	
ODR22	Kapila Athanayaka	Red stop sign	2017-12-01 17:41:50	5	Rs.1,000.00	
ODR23	Thamara Sugandhi	Meta-tech scaffolding	2017-12-01 20:37:24	1	Rs.450.00	
ODR24	Gayan perera	Compact cutter	2017-12-02 07:15:48	6	Rs.5,400.00	
ODR25	Thamara Sugandhi	Meta-tech scaffolding	2017-12-02 13:49:35	1	Rs.450.00	

Figure 3.22 View order details

3.6.9 Payment management module Interface

Figure 3.23 (Admin payment screen) represent the payment module form which can be viewed by the admin. Admin can check income report with this module. Admin is able to view all the order details. With this module company management can take a clear track of payments made by customers.

Income Report			
Reports from 2017-11-06 To 2017-12-06			
ID	Order Date	Order No	Orders Price
1	2017-11-27 19:32:58	ODR11	Rs.2,000
2	2017-11-29 10:52:01	ODR12	Rs.20,000
3	2017-11-29 10:58:48	ODR13	Rs.96,000
4	2017-11-29 13:53:06	ODR14	Rs.1,050
5	2017-12-01 00:10:30	ODR18	Rs.500
6	2017-12-01 00:10:30	ODR18	Rs.200
7	2017-12-01 15:11:11	ODR21	Rs.2,000
8	2017-12-01 15:11:11	ODR21	Rs.700
9	2017-12-01 17:41:50	ODR22	Rs.6,120
10	2017-12-01 17:41:50	ODR22	Rs.1,000
11	2017-12-01 20:37:24	ODR23	Rs.450
12	2017-12-02 07:15:48	ODR24	Rs.5,400
13	2017-12-02 13:49:35	ODR25	Rs.450

From Date	To Date	Total Order price
2017-11-06	2017-12-06	Rs.135,870

Figure 3.23 View income report

3.6.10 Feedback management module Interface

Figure 3.24 represent Feedback management view table. Administrator can view all comments of customer and also he can delete if he want. Customer post a comment in single product page.

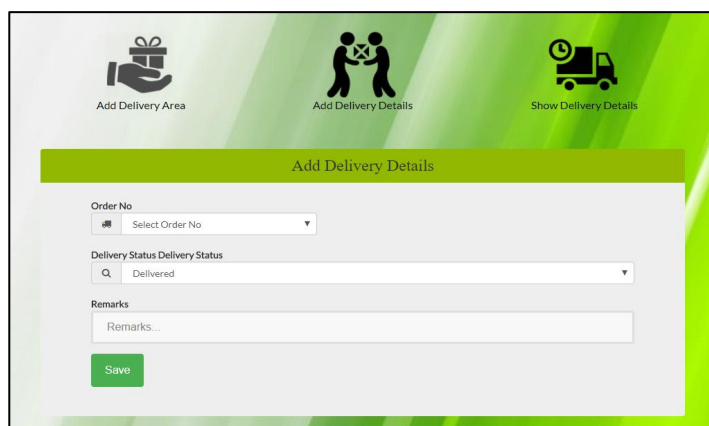





ID	Product Name	Customer Name	Comments	Delete
1	Mason scaffolding	Sanka Pathama	Good Product. I rented many times this products from here	
2	Chop saw	Nuwan Ranasinghe	Some of saw blades not work properly.	
3	Compact cutter	Gayan perera	Good Product	

Figure 3.24 View feedback details

3.6.11 Delivery management module Interface

Figure 3.25 represent Delivery management form which administrator can show delivery status. Before show the delivery status, he has to select Order no, then automatically system enter delivery status. After that administrator can enter remarks if he wants.



 Add Delivery Area
  Add Delivery Details
  Show Delivery Details

Add Delivery Details

Order No

Delivery Status Delivery Status

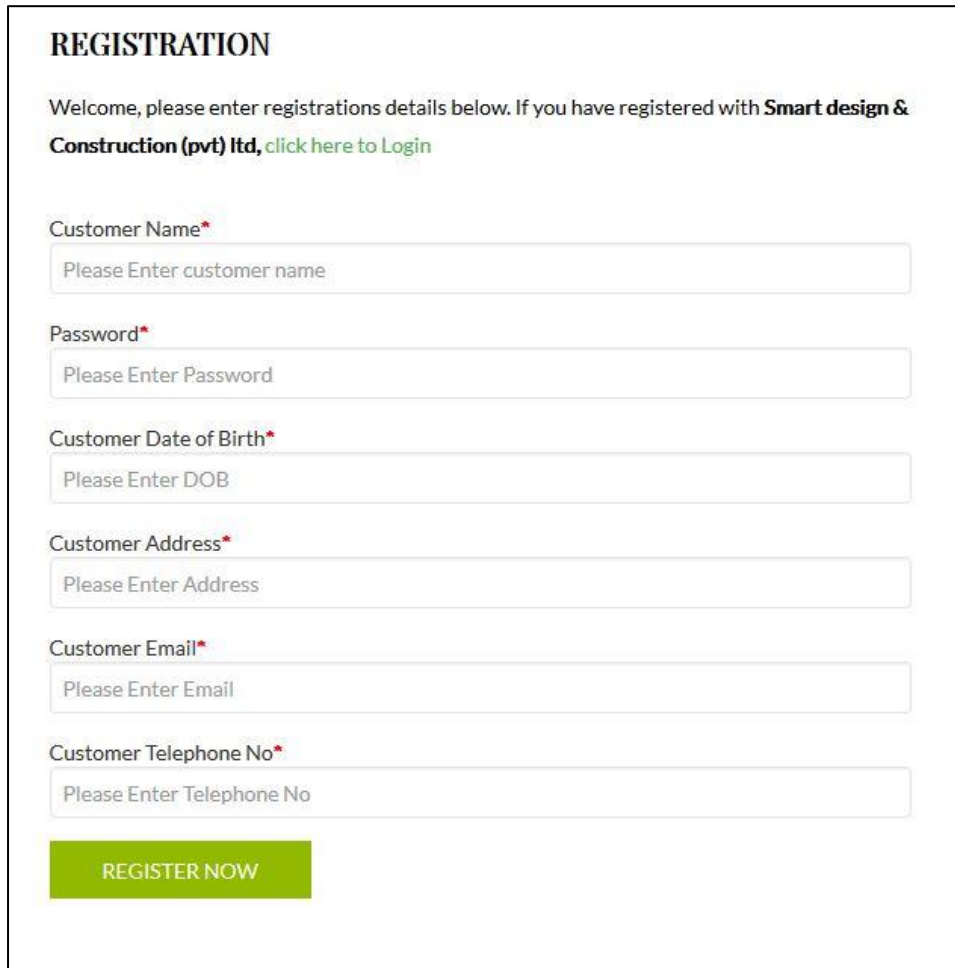
Remarks

Figure 3.25 Add delivery details

3.7 Website Interface Design

3.7.1 Login management module interface

Customer can rent products through system website. Before order the products must have to logged in. He was not logged in before he have to register on website. Figure 3.26 register form and figure 3.27 Login form on website.



The registration form is titled "REGISTRATION" and includes a welcome message: "Welcome, please enter registrations details below. If you have registered with **Smart design & Construction (pvt) ltd**, [click here to Login](#)". The form contains several input fields, each with a red asterisk indicating a required field: "Customer Name*", "Password*", "Customer Date of Birth*", "Customer Address*", "Customer Email*", and "Customer Telephone No*". Each field has a placeholder text: "Please Enter customer name", "Please Enter Password", "Please Enter DOB", "Please Enter Address", "Please Enter Email", and "Please Enter Telephone No". A green "REGISTER NOW" button is located at the bottom of the form.

Figure 3.26 Website registration interface



The login form is titled "REGISTERED CUSTOMERS" and includes the instruction: "If you have an account with us, please log in.". The form contains two input fields, each with a red asterisk indicating a required field: "Email Address*" and "Password*". Each field has a placeholder text: "Please Enter Email" and "Please Enter Password". A green "LOGIN" button is located at the bottom of the form.

Figure 3.27 Website login interface

3.7.2 Product management module interface

After logged in to website, figure 3.28 shows rent products. Customer can browse products through website menu. And after display all products images, products daily rent, and also products description. As well as left hand side there is product categories.

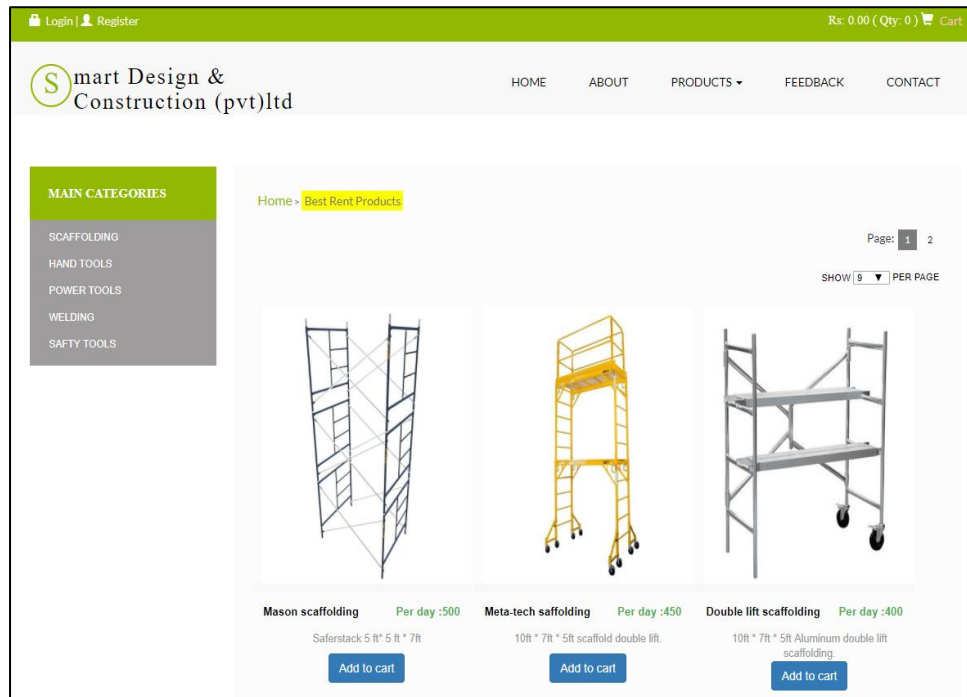


Figure 3.28 Website product page

Customer click on the product image, browser change product page to single product display page. And then customer can enter quantity, days to rent of product and click on the Add to Cart button to order product. And also there is comment area. If customer want to type comment about this display product he can post a comment on there.

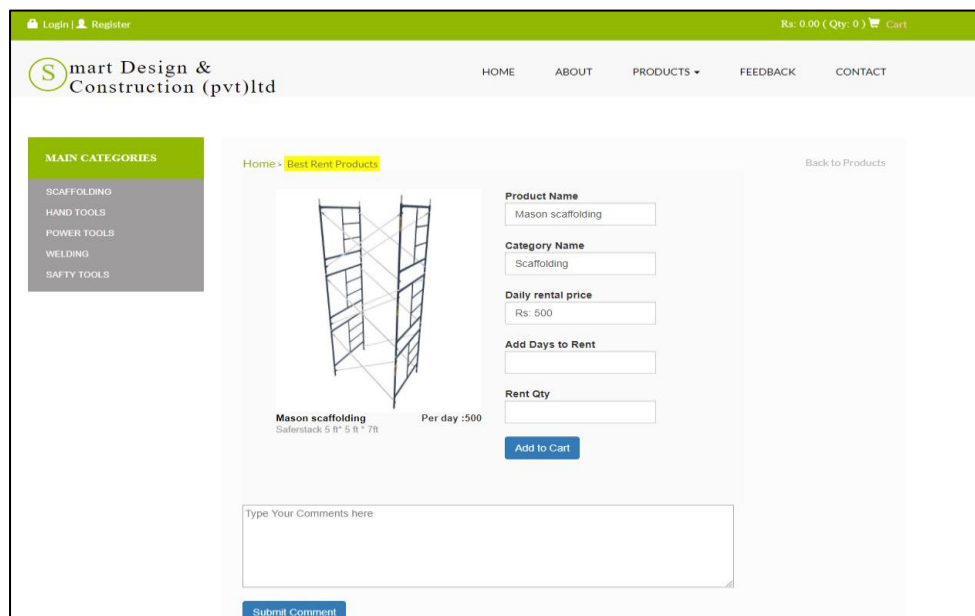


Figure 3.29 Website single product page

3.7.3 Shopping Cart page

Figure 3.30 and Figure 3.31 represents Shopping cart screen and payment checkout screen. Those are payment module from customer side. After checking out customer can enter payment details and pay the necessary amount. Once it is done customer will receive an email notification. Figure 3.32 shows payment confirmation status.

All the details entered here will be saved in the database. If the customer wants to cancel the order, he/she can call the company and cancel it by giving the order no to the company.

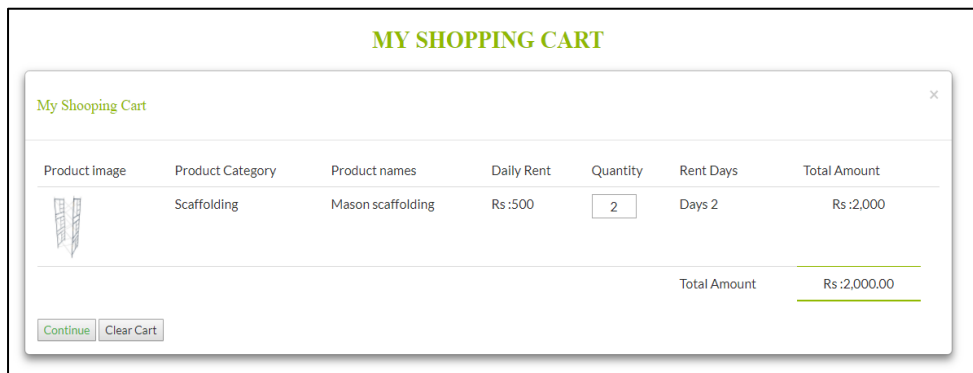


Figure 3.30 Add to shopping cart view

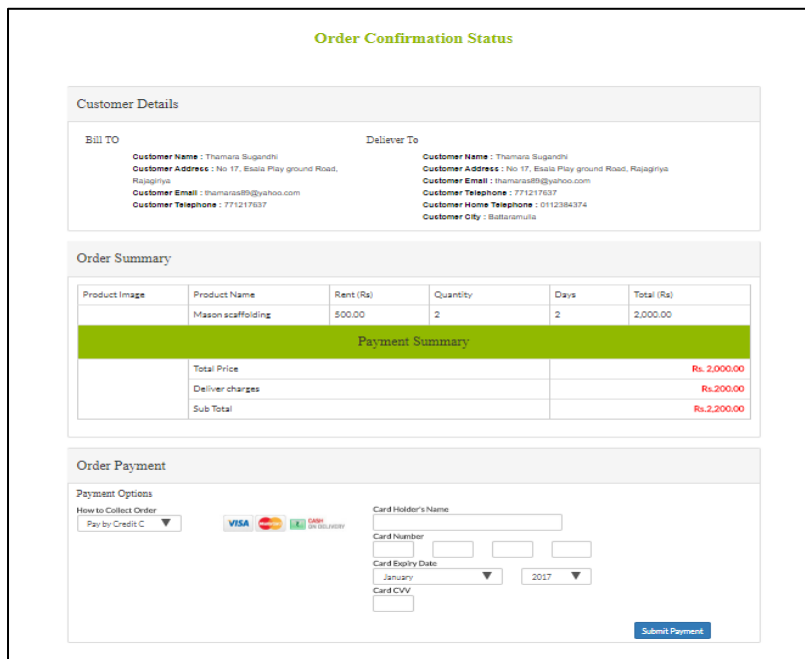


Figure 3.31 Checkout process page



Figure 3.32 Payment confirmation status

3.7.4 Change web page content from Admin panel

If administrator want to change website content without coding ,he can do it easily from admin panel. He can change content and images. Figure 3.33 show about us page in admin panel and 3.34 show about us website page.

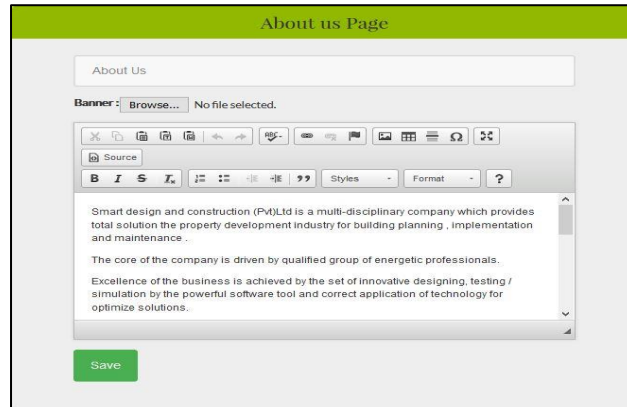


Figure 3.33 About us admin panel form

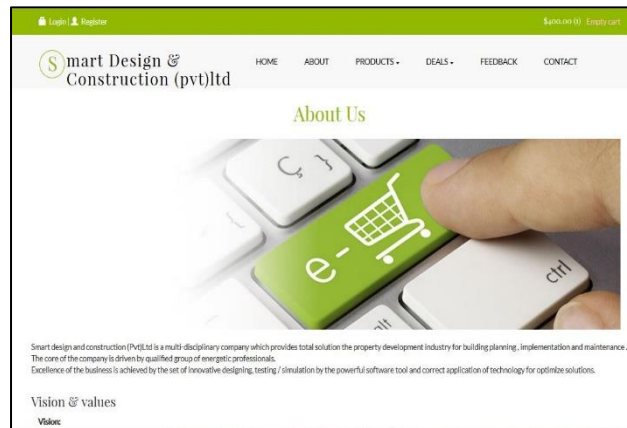


Figure 3.34 About us website page

3.7.5 Web pages



Figure 3.35 Website home page

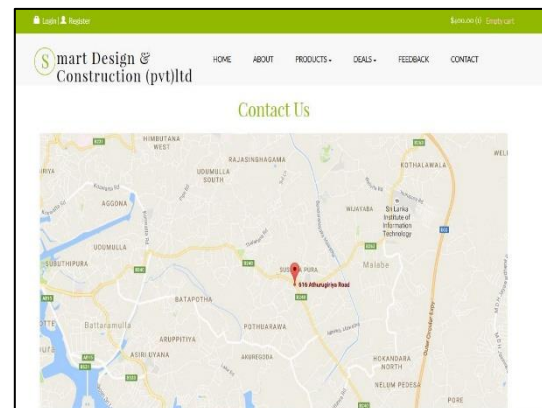


Figure 3.36 Website contact us page

CHAPTER 04–IMPLEMENTATION

4.1 Introduction

This chapter is a description of the activities that were carried out during the development of the Online Service Rental System and their results. And Software is developed according to the details design based on client requirements. Implementation phase is time consuming phase in the software development life cycle. There are some important things to do. When we code the system, using comments is a very important thing. Because if we want to change design again we can do it easily. And another is thing is validation part of coding. Every programmer must consider about validation and code comments while programming. To develop the system I used Codeigniter which is an object oriented framework based on php.

4.2 Hardware and Software Requirements

4.2.1 Hardware Requirements

Server :

- Operating System : Windows 10
- Processor : Intel(R) Pentium® CPU N3540 2.16GHz
- RAM : 4.00GB
- System Type : 64 Operating System

Client :

- Operating System : Windows 7 Professional
- Processor : Intel® Core™ i3
- RAM : 4.00GB
- System Type : 32 Operating System

4.2.2 Software Requirements

Server:

- Operating System : Windows 10
- Web Scripting : Object Oriented PHP
- Frame work : Codeignite 3.0
- Web Server : Apache
- DBMS : Mysql
- IDE : NetBeans 8.0.2
- Software Bundle : XAMPP
- Drawing Tools : SmartDraw, Ms Project, Dia, Rational Rose
- Technologie : HTML,HTML5,CSS,CSS3,Javascript,Bootstrap, AJAX,Jquery,Google map, Dom pdf

Client:

- Operating System : Windows 7
- Web Browser : IE ,Firefox, Chrome

4.3 Development Tools

4.3.1 Net Beans IDE 8.0.2

NetBeans is an open-source integrated development environment (IDE) for developing with Java, PHP, C++, and other programming languages. NetBeans is also referred to as a platform of modular components used for developing Java desktop applications. WIKI [8]

Main reasons to use Netbeans as main IDE

- Fast
- Simple
- SCM integration
- PHP unit testing

4.3.2 Codeigniter framework

CodeIgniter is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications. CodeIgniter was created by EllisLab, and is now a project of the British Columbia Institute of Technology.

4.3.4 Mysql

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments. WIKI [9]

4.4 Major Code Segment

CodeIgniter is based on the Model-View-Controller (MVC) development pattern. MVC is a software approach that separates application logic from presentation. In practice, it permits your web pages to contain minimal scripting since the presentation is separate from the PHP scripting.

The **Model** represents your data structures. Typically, your model classes will contain functions that help you retrieve, insert and update information in your database.

The **View** is information that is being presented to a user. A View will normally be a web page, but in CodeIgniter, a view can also be a page fragment like a header or footer. It can also be an RSS page, or any other type of “page”.

The **Controller** serves as an intermediary between the Model, the View, and any other resources needed to process the HTTP request and generate a web page.

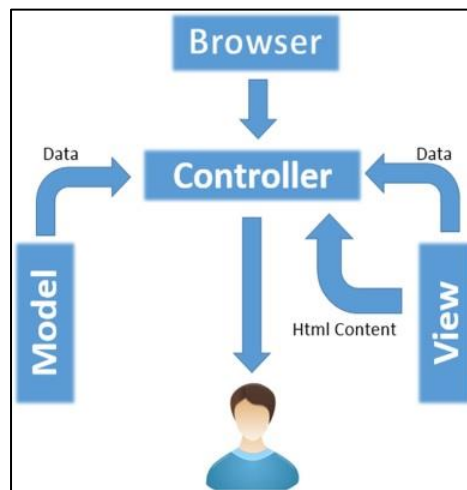


Figure 4.1 Show MVC Architecture

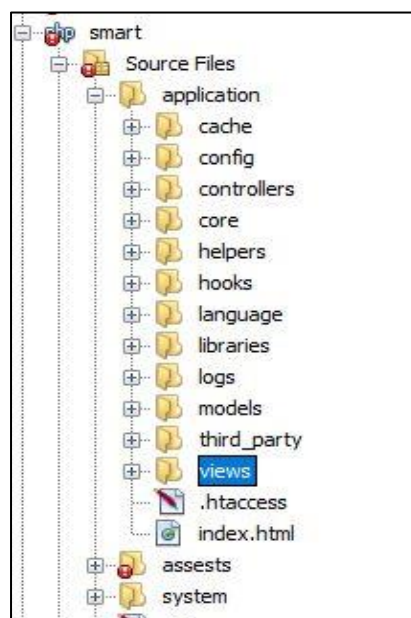


Figure 4.2 Layers of MVC architecture

4.4.1 Model

The model is where the system carries out the operations in the database tables. The following code fragment shows the code for insertion of a record.

```

Class Category_model extends CI_Model{

function insert_category(){
    $cat_name = $this->input->post('form-catname');
    $cat_type = $this->input->post('form-cattype');
    $cat_desc = $this->input->post('form-catdis');

    $insert_data = array(
        'cat_name' => $cat_name,
        'cat_type' => $cat_type,
        'cat_desc' => $cat_desc
    );

    $this->db->insert('category', $insert_data);
    return 1;
}

```

4.4.2 Controller

Controller acts as a messenger between the models and views. It gets the requests and responses from each model, helper classes and updates the view files according to the results. Following code shows how the insert, update and delete is been done in Controller. Due to limited space only small portion is added.

Insert, Update and Delete data

```

public function index()
{
    $this->form_validation->set_rules('form-catname', 'Category Name', 'required');
    $this->form_validation->set_rules('form-cattype', 'Category Type', 'required');
    $this->form_validation->set_rules('form-catdis', 'Category Description', 'required');

    if($this->form_validation->run() == FALSE)
    {
        $this->load->view('admin/common/header');
        $this->load->view('admin/products/categories');
        $this->load->view('admin/common/footer');
    }
    else
    {
        $res = $this->category_model->insert_category();
        //var_dump($res);
        if($res == 1)
        {
            $message_success = '<div class="alert alert-success">Saved Successfully!</div>';
            $this->session->set_flashdata('message', $message_success);
            redirect('admin/products/categories');
        }
        else{
            $message_error = '<div class="alert alert-danger">Please try again!</div>';
            $this->session->set_flashdata('message', $message_error);
            redirect('admin/products/categories');
        }
        //redirect('admin/products/categories/view');
    }
}

```

```

public function delete_category($cat_id){
    $res = $this->category_model->delete_category($cat_id);
    //var_dump($res);
    if($res == 1)
    {
        $message_success = '<div class="alert alert-success">Deleted Successfully!</div>';
        $this->session->set_flashdata('message', $message_success);
    }
    else{
        $message_error = '<div class="alert alert-danger">Please try again!</div>';
        $this->session->set_flashdata('message', $message_error);
    }
    redirect('admin/products/categories/view_category');
}

```

```

public function edit_category($cat_id){

    $res['cat_data'] = $this->category_model->get_category_data($cat_id);

    $this->load->view('admin/common/header');
    $this->load->view('admin/products/categories',$res);
    $this->load->view('admin/common/footer');

}

```

4.4.3 View

View is where all the interfaces and presentation display area. More effort was taken in the separation of logics in the views, the session variables play a reasonable role in the views because same forms are used for data insertion as well as the data updates. We also can load more than one view into an interface.

```

<div class="form-bottomnew">
    <?php
    $attributes = array('class' => 'cat-form', 'id' => 'catform', 'name'=>'catform');
    echo form_open('admin/products/categories', $attributes);
    ?>

    <div class="form-group">
        <input type="text" name="cat_id" style="display:none;" id="cat_id" value="<?php if(←
isset($cat_data)){ echo $cat_data[0]->cat_id; } ?>" />
        <label class="sr-only" for="form-catname">Category Name</label>
        <input type="text" name="form-catname" placeholder="Category Name..." class="form-←
control" id="form-catname" value="<?php if(isset($cat_data)){ echo $cat_data[0]->cat_name; }else{ echo set_value('form-←
-catname'); } ?>">

        <?php if(form_error('form-catname')){ ?>
        <div class="err_danger"><?php echo form_error('form-catname'); ?></div>
        <?php
        }
        ?>
    </div>

```

4.4.4 Data validation

CodeIgniter provides a comprehensive form validation and data prepping class that helped to minimize the amount of code I wrote.

Following is the data validation process of codeigniter.

- A form is displayed.
- You fill it in and submit it.
- If you submitted something invalid, or perhaps missed a required item, the form is redisplayed containing your data along with an error message describing the problem.
- This process continues until you have submitted a valid form.

On the receiving end, the script must:

- Check for required data.
- Verify that the data is of the correct type, and meets the correct criteria. For example, if a username is submitted it must be validated to contain only permitted characters. It must be of a minimum length, and not exceed a maximum length. The username can't be someone else's existing username, or perhaps even a reserved word. Etc.
- Sanitize the data for security.
- Pre-format the data if needed (Does the data need to be trimmed? HTML encoded? Etc.)
- Prep the data for insertion in the database.

Although there is nothing terribly complex about the above process, it usually requires a significant amount of code, and to display error messages, various control structures are usually placed within the form HTML. Form validation, while simple to create, is generally very messy and tedious to implement.

Following is a sample code to show how the validation is done.

```
$this->form_validation->set_rules('form-catname', 'Category Name', 'required');  
$this->form_validation->set_rules('form-cattype', 'Category Type', 'required');  
$this->form_validation->set_rules('form-catdis', 'Category Description', 'required');
```

4.4.5 Preventing SQL Injection

SQL injection is an attack made on database query. In PHP, we use `mysql_real_escape_string()` function to prevent this along with other techniques but CodeIgniter provides inbuilt functions and libraries to prevent this.

Sql injection is prevented in 2 different ways in the system.

Escaping queries

`$this->db->escape()` function automatically adds single quotes around the data and determines the data type so that it can escape only string data.

Active Record Class

Using active records, query syntax is generated by each database adapter. It also allows after queries, since the values escape automatically.

Example :

```
public function log_track_logout(){
    $ad_id = $this->session->userdata('admin_id');
    $insert_data = array(
        'admin_id' => $ad_id,
        'remarks' => 'Successfully logged out'
    );
}
```

4.4.6 Hiding PHP errors

In production environment, we often do not want to display any error message to the users. It is good if it is enabled in the development environment for debugging purposes. These error messages may contain some information, which we should not show to the site users for security reasons.

We can hide php errors using **error_reporting(0)** in the index.php file.

To turn off database errors set the **db_debug** to **false** in database.php file.

4.5 Reused Existing Codes

Some codes were found by referring internet and used in this project. Extracted codes were customized and well tested. Following codes are extracted from the websites.

- Bootstrap css and js files
- Bootstrap calendar
- Ckeditor

CHAPTER 05 – EVALUATION

5.1 Introduction

Testing is the process of evaluating a system or its component with the intent to find whether it satisfies the specified requirements or not. Further testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements. WIKI [10]

Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

5.2 Software Testing

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of A software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process. .WIKI [11]

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

5.3 Techniques of Software Testing

There are two basics of software testing: blackbox testing and whitebox testing.

1. Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing. WIKI [12]

2. White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

5.4 Types of Testing

There are many types of testing like.

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

5.4.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input. WIKI [13]

5.4.2 Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

5.4.3 System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

5.4.4 Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

5.4.5 Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

5.4.6 Beta Testing

Beta testing is the testing which is done by end users, a team outside development, or publicly releasing full pre-version of the product which is known as beta version. The aim of beta testing is to cover unexpected errors. It falls under the class of black box testing.

5.5 Test case

A test case is typically defined as a document that lays out the following:

- Test data
- Procedures/inputs
- Scenarios
- Descriptions
- Testing environment
- Expected results
- Actual results

It's used to for a singular test scenario. As a rule, there are usually both a positive and negative test case for each scenario.

The purpose of a test case is two-fold: It's designed to find any errors or bugs within the software application, and it's also designed to show how the application should be executed if it performs correctly. It also demonstrates real-world use of the product and whether it fits the customer needs. [WIKI-11]

5.6 Test plan

While a test case is only designed to test a particular scenario, a test plan is a comprehensive document that lays out all major activities associated with a particular testing project. A test plan includes: [WIKI-12]

- Scope of the project
- Objectives
- Target market
- Assumptions
- Testing cycle start/end dates
- Major roles and responsibilities/overall resources
- Testing environment
- Deliverables
- Major risks and how to handle these risks
- Defect reporting and mitigation
- Testing end date

Following Figure Table 5.1 shows test plan. Please refer Appendix E for other test cases.

Module Name	Function Name	Test priority
User Management	Insert User Details	High
	Delete User Details	High
	Edit User Details	Medium
	Update User Details	Medium
	View Users	High
	Privilege access to website	High
Login Management	Insert Login Details	High
	Update Login Details	High
	Edit Login Details	High
	Search by customer name	Law
	Search by admin name	Law
	Confirm deletion of Login files	Medium
Customer Management	Insert Customer Details	High
	Delete Customer Details	High
	Update Customer Details	High
	Edit Customer Details	High
	Show Customers	High
	Search by Customer Names	Medium
	Show Customer reports	High
	Search Customer history	High
Supplier Management	Insert Supplier Details	High
	Update Supplier Details	High
	Edit Supplier Details	High
	Delete Supplier Details	High
	Show Suppliers	High
	Show Supplier history	Medium
	Update Suppliers' Products	Medium
	Search most active Supplier	Law
Product Management	Insert Products	High
	Edit Products	High
	Delete Products	High
	Update Products	High
	Show Products	High
	Show most rent Products	Law
	Search products details	Medium
	Load all products on website	High
	Update Products Images	High
	Delete product Images	High
	Insert Product Description	High
	Update Product Description	High
	Edit Product Description	High
	Delete Product Description	High

Stock Management	Insert Stock	High
	Update Stock	High
	Edit Stock	High
	Delete Stock	High
	Show Stock Details	High
	Insert Damaged Stock	High
	Update Damaged Stock	High
	Edit Damaged Stock	High
	Delete Damaged Stock	High
	Insert Stock images	Medium
	Insert Damaged Stock images	Medium
	Insert Stock Quantity	High
	Update Stock Quantity	High
	Insert stock description	High
	Update stock description	High
	Edit stock description	High
	Delete stock description	High
Order Management	Edit Orders	High
	Update Orders	High
	Insert Orders	High
	Delete Orders	High
	Show Orders by name	Medium
	Show Orders by Customers name	Medium
	Show Order dates and Time	Medium
	Cancel Orders	High
	Insert Order Quantity	High
	Update Order Quantity	High
Feedback Management	Delete customer comments	High
	View customer comments	High
	Send email notifications to Customer	High
	View notifications	Medium
Delivery Management	Insert order no	High
	Select delivery date	High
	Select delivery status	High
	Insert order remarks	High
	View delivery reports	High
Report Management	Generate Customer Reports	High
	Generate Supplier Reports	High
	Generate Products Reports	High
	Generate Order Reports	High
	Generate Invoices	High
	Generate daily income reports	High
	Generate monthly income reports	High
Generate monthly maintenance reports	High	

	Generate history reports	Medium
Payment Management	Customer order product	High
	Customer add product to shopping cart	High
	Customer cancel product	High
	Customer payment for orders	High
	Administrator view payment status	High
	Administrator cancel order	High
	Administrator view order history	Medium

Table 5.1 High level Test plan

5.7 User Evaluation

User Evaluation is done by selecting users of the System. In this Online Service Rental System has different user levels. Administrator and other users has different privileges. In this system, user evaluation questionnaire was sent to system administrator through email. And he send his reply to us using same word document. Following Figure 5.1 shows user evaluation questionnaire.

<u>User Evaluation Questionnaire</u>					
Name of User : K.P <u>Kariyawasam</u>			Role of User : Administrator		
Evaluating Item	Very Good	Good	Average	Poor	Very Poor
Overall Reaction	√				
Character Readability		√			
Color Scheme	√				
System Navigation	√				
Ease of Usage	√				
Functionalities	√				
Interfaces	√				
Ease of Learning		√			
Response Time	√				
<u>Comments</u> : Friendly interface and this system is very useful to speed our services .					

Figure 5.1 User evaluation form

CHAPTER 06–CONCLUSION

6.1 Introduction

Smart Design & Construction (pvt) ltd is a popular construction rent service company in Malabe. They had many problems with their business. They didn't have enough new technology knowledge and well maintained staff to run their business. Therefore they need to enhance their service with modern technology.

In this online service rental system run their business effectively. They have to spend more times to learn this process. Currently they run their business activates perfectly and effectively with new system. And their business process results automatically come through this online service rental system.

This System generated reports in daily, monthly and yearly .So then Manager can get analyze their business status and he can get solutions if there will be problems in the reports. And important thing is he can discuss about business activates and business decisions with company staff.

During the analysis phase functional and non-functional requirements identify in each modules in system. And after getting the idea from that analysis phase, each and every module were successfully completed in implement phase. End of the system processing cycle, client request additional functionalities and features to add the system. After this process client was satisfied about my project.

6.2 Future Enhancement

There are some future enhancement of this proposed system.

- Improving delivery by using GPS technology.
- Introduces online rent through using mobile application.
- Improving reports and showing graphs by using analytical tool.
- Introduce online rent auction to the website.
- Improving payment options to the customers..
- Improve high security methods to the system.
- Introduce shopping cart whislist the system.

6.3 Lesson Learnt

This project gave me a valuable programming knowledge in my life. In addition I understood how I used programming theories in practically. And I gained complete knowledge about Software Development Life cycle. Most important thing is I develop my project within time manageable project schedule.

This project give me a chance to learn new programming language, that is Php codeigniter. Php codeigniter is object oriented language. I learned many things about php codeigniter. Besides, special efforts were taken to learn the MVC design pattern. And also find out and learn upcoming technologies (AJAX, JQuery, PHP, Mysql, CSS3, HTML5 , Bootstrap) .

Furthermore working on this project helped me to improve my intellectual skill and technical skill. As well as help me to guide how to collaborating with many individuals from collective fields.

References

- WIKI [1] Functional requirement, http://en.wikipedia.org/wiki/Functional_requirement
- WIKI [2] Non Functional requirements, http://en.wikipedia.org/wiki/Nonfunctional_requirement
- WIKI [3] Fine Vision Group, <https://finevision.lk>
Equipment Rental in Sri Lanka - Colombo | KFTV
- WIKI [4] System Design, https://en.wikipedia.org/wiki/Systems_design
- WIKI [5] Rational Unified process, http://en.wikibooks.org/wiki/RUP_IBM_Rational_Unified_Process/Phases
- [Whitten,2007] Jeffrey L whitten & Lonnie D Bentley ,Systems Analysis and Design methods 7th edition,2007
- WIKI [6] Database Design, http://en.wikipedia.org/wiki/Database_design.
- WIKI [7] Database Normalization, https://en.wikipedia.org/wiki/Database_normalization
- [Sommerville, 2007] Ian Sommerville, Software Engineering, 8th edition, Addison-Wesley, 2007
- WIKI [8] Net beans IDE 8.0.3, <https://www.techopedia.com/definition/24735/netbeans>
- WIKI [9] My Sql, <https://www.tutorialspoint.com/mysql/mysql-introduction.htm>
- WIKI [10] Evaluation, Software Testing http://www.tutorialspoint.com/software_testing
- WIKI [11] Software Testing , <http://www.codeproject.com/Tips/351122/What-is-software-testing-What-are-the-different-ty>.
- WIKI [12] Test Case ,<http://www.ibeta.com/test-plan-vs-test-cases-whats-the-difference/>
- WIKI [13] Test Plan,<http://www.ibeta.com/test-plan-vs-test-cases-whats-the-difference/>

Appendix A- SYSTEM DOCUMENTATION

System Manual

Step 1 - Install xampp

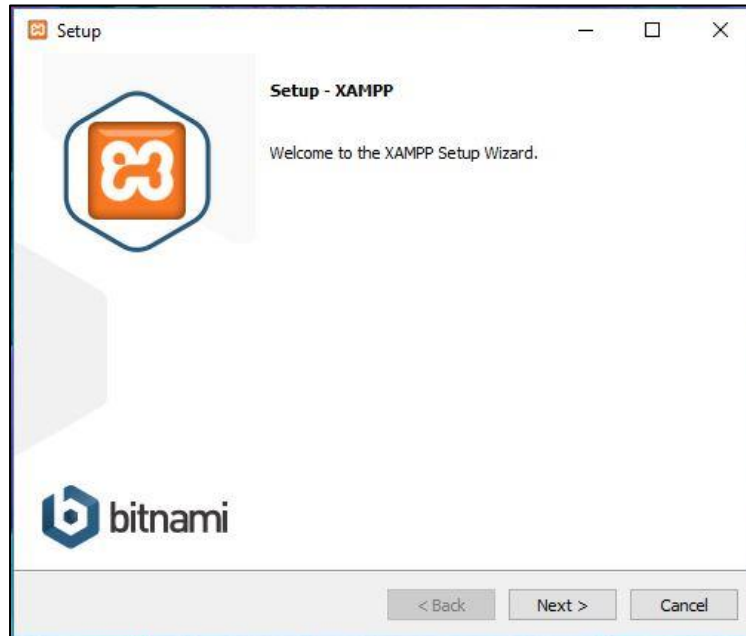


Figure A.1 XAMPP setup

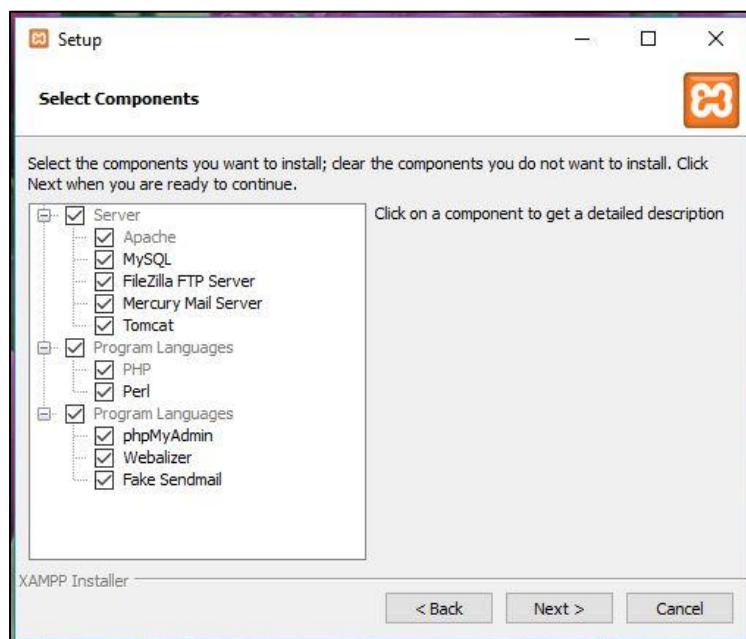


Figure A.2 Select components

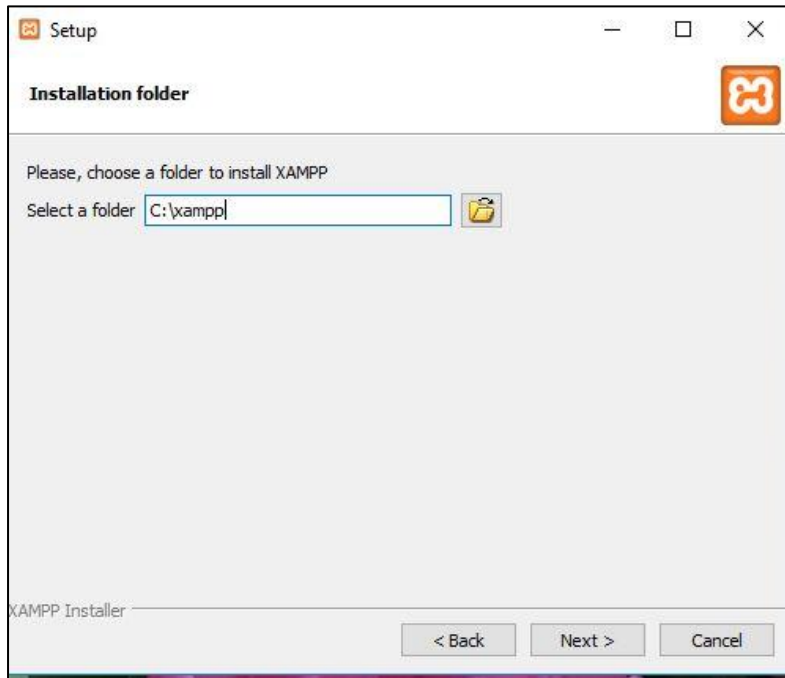


Figure A.3 Choose folder path on disk C

Step 2 - After install xampp , Run xampp control panel. And Start Apache and Mysql.

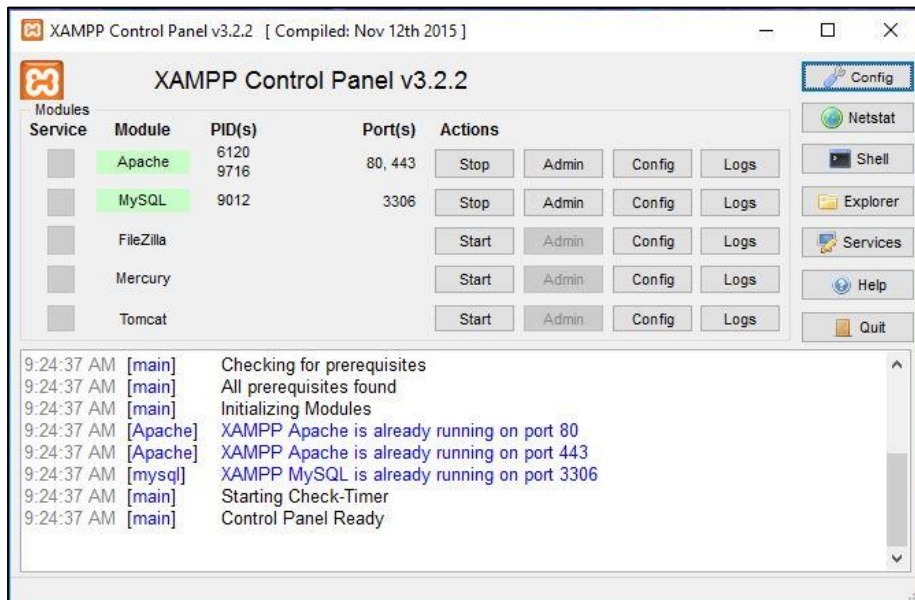


Figure A.4 Run XAMPP panel

Step 3 – Copy the smart folder given in the CD and paste it inside the htdocs folder in the below path.

Windows Environment with XAMPP installed - the path would be - C:\xampp\htdocs .

Database setup

Step 1 – Open phpMyAdmin by typing the <http://localhost/phpmyadmin> URL in browser's address bar

Step 2 – Create Smart_db database .

Step 3 - Click the Import tab and browse through the CD's database folder .And import smart_db.sql file

Step 4 – Click go button to import the folder into created smart_db database.

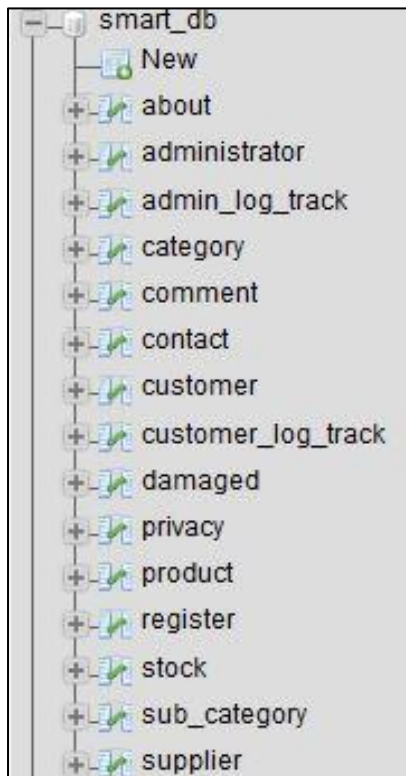


Figure A.5 Tables list in smart_db database

Appendix B –DESIGN DOCUMENTATION

Use case diagrams and description

The following diagrams and its description might give a clear understand about the system design.

Use case diagram for Product Module

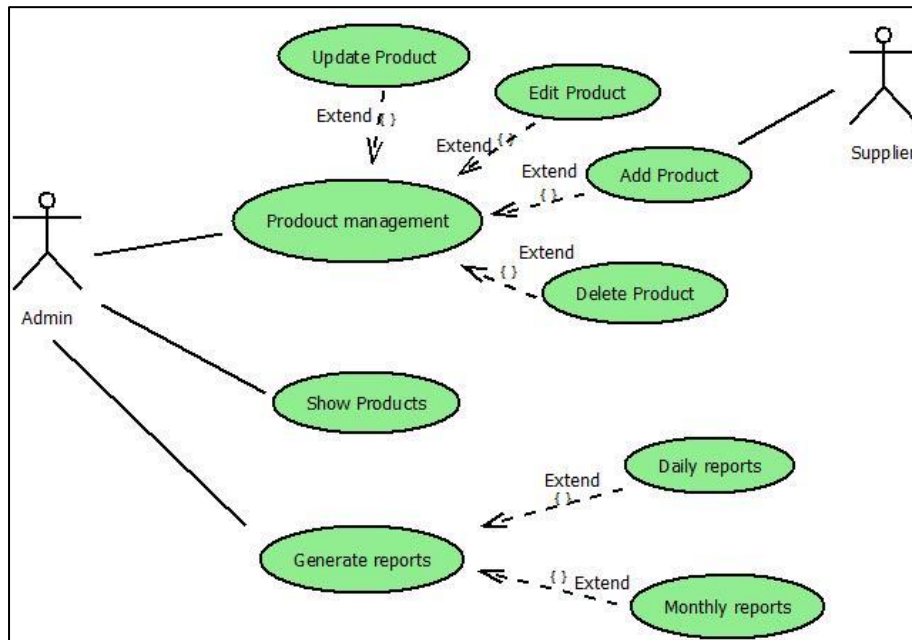


Figure B.1 Use case diagram of the product management module

Use Case Name	Product Management
Actors	Administrator , Supplier
Description	Product Details are added to the System
Pre-conditions	Administrator should be logged into the system
Flow of events	<ol style="list-style-type: none"> 1) Administrator should be insert products details. 2) Select category type of the products. 3) Insert Products images. 4) Enter product price and Description 5) Click save button 6) Display successfully message 7) Show all insert products 8) Automatically Generate products reports.
Post -condition	All data are save in the database

Table B.1 Use case description for Product management module

Use Case Name	Show Products
Actors	Administrator
Description	Show product details in the system
Pre-conditions	Administrator should be logged into the system
Flow of events	<ol style="list-style-type: none"> 1) Go to view products. 2) Products will be displayed automatically. 3) Click edit/delete to edit/delete products 4) Display message
Post -condition	All data are save in the database

Table B.2 Use case description for Show product

Use case diagram for Report Module

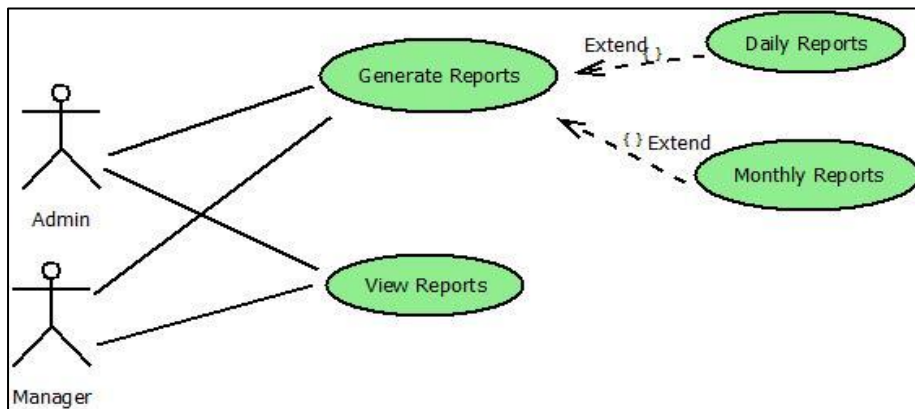


Figure B.2 Use case diagram of the report management module

Use Case Name	Generate Reports
Actors	Administrator
Description	Generates reports about products in the system
Pre-conditions	Administrator should be logged into the system
	<ol style="list-style-type: none"> 1) Go to reports. 2) Go to products tab. 3) Click reports button 4) Report will be automatically generated
Post -condition	All data are save in the database

Table B.3 Use case description for Generate Reports

Use case diagram for Customer Module

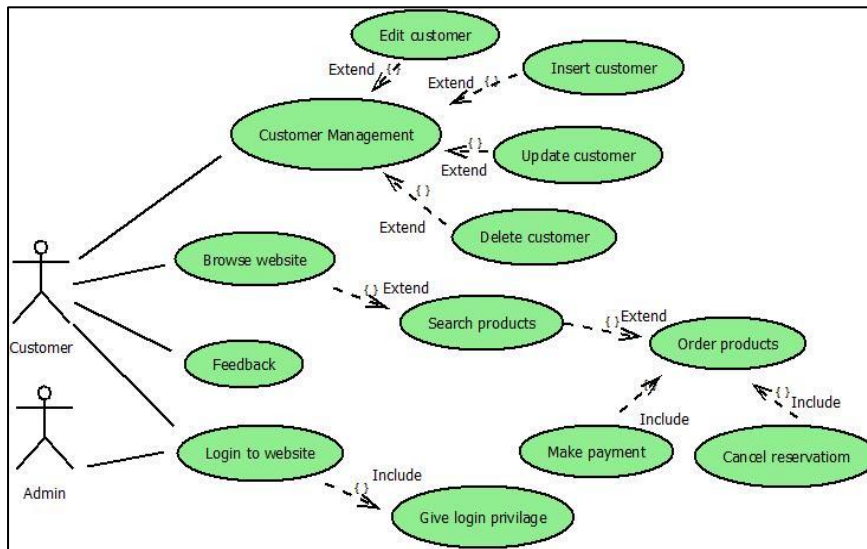


Figure B.3 Use case diagram of the customer management module

Use Case Name	Customer Management
Actors	Administrator, Customer
Description	Customer details are added to the system
Pre-conditions	Administrator should be logged into the system
Flow of events	<ol style="list-style-type: none"> 1) Administrator can insert customer details from the admin panel. 2) Admin also can delete the customers 3) Customers can register themselves by going through the website. 4) They can create their own account. 5) Email will be checked when they try to create a account.

Table B.4 Use case description for Customer module

Use Case Name	Customer Order Products
Actors	Customer
Description	Order details
Pre-conditions	Customer must logged into the website
Flow of events	<ol style="list-style-type: none"> 1) Once customer logged in he can go to product page. 2) Add product to cart 3) View cart 4) Edit cart if needed 5) Checkout 6) Make payment
Post Conditions	Order details are saved to generate reports reports

Table B.5 Use case description for Customer order products

Use case diagram for Supplier Module

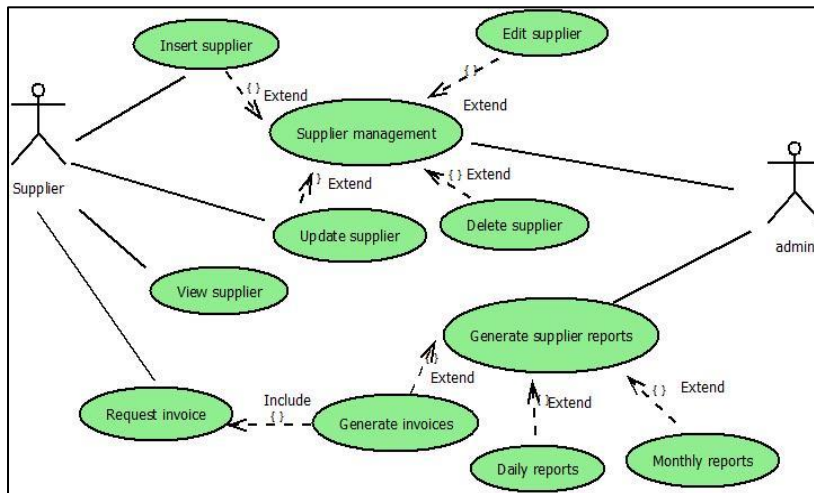


Figure B.4 Use case diagram of the Supplier management module

Use Case Name	Supplier Management
Actors	Administrator, Supplier
Description	Supplier details are added to the system
Pre-conditions	Administrator should be signed in.
Flow of events	<ol style="list-style-type: none"> 1) Administrator enter details of the supplier 2) Click save button 3) Save supplier 4) Display message 5) Click edit to customer 6) Click delete to customer 7) Generate report automatically
Post -condition	All data are save in the database

Table B.6 Use case description for Supplier module

Use case diagram for Order Module

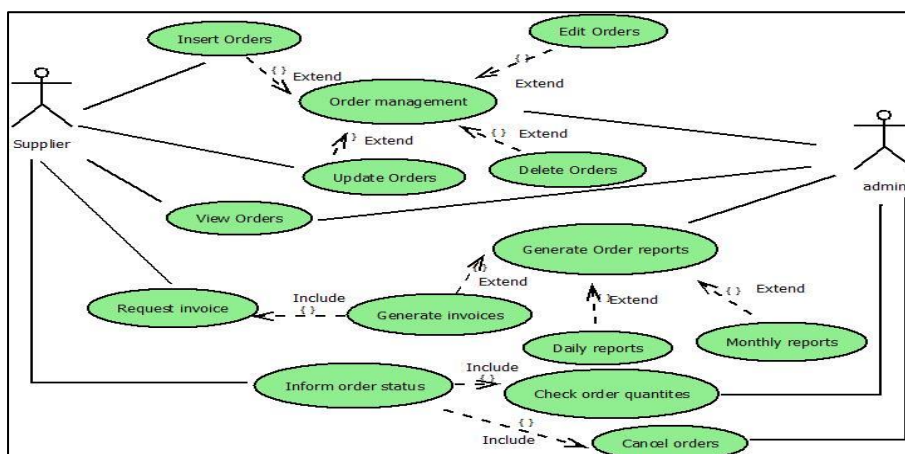


Figure B.5 Use case diagram of the Order management module

Use Case Name	Order Management
Actors	Administrator, Supplier
Description	Check Order details.
Pre-conditions	Administrator should be signed in.
Flow of events	<ol style="list-style-type: none"> 1) Administrator check order status. 2) Check order quantities and update quantities 3) Check customers who orders products. 4) If customer not paid for order , administrator can cancel orders. 5) If customer already paid for order, send a mail to customer 6) Generate report automatically
Post -condition	All data are save in the database

Table B.7 Use case description for Order module

Use case diagram for Stock Module

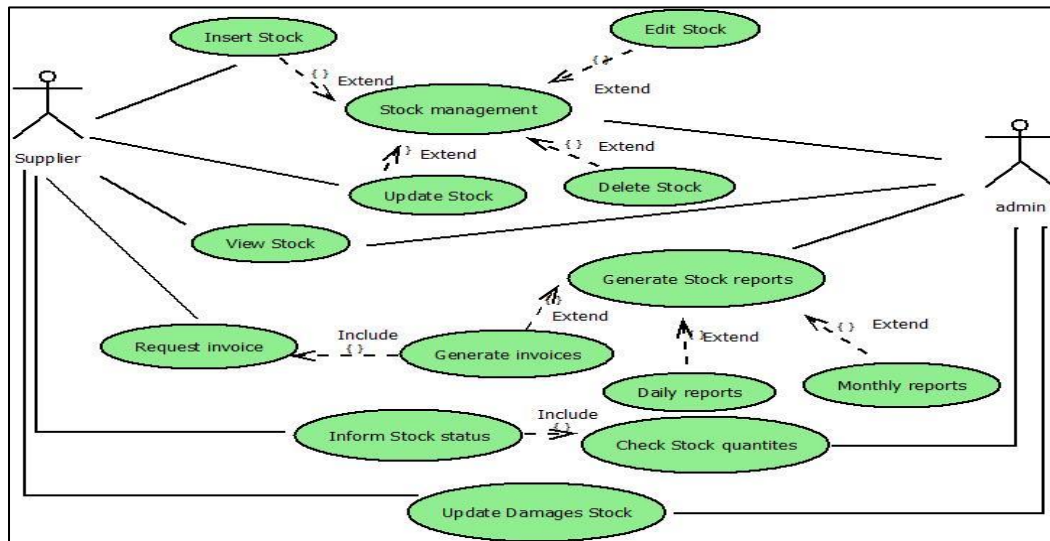


Figure B.6 Use case diagram of the Stock management module

Use Case Name	Stock Management
Actors	Administrator, Supplier
Description	Stock details are added to the system
Pre-conditions	Administrator must be logged in.
Flow of events	<ol style="list-style-type: none"> 1) Administrator enter details of the stock 2) Check stock quantities and update quantities 3) Click save button 4) Save stock 5) Display message 6) Click edit to stock 7) Click delete to stock 8) Generate report automatically
Post -condition	All data are save in the database

Table B.8 Use case description for Stock module

Use Case Name	Stock Management
Actors	Administrator, Supplier
Description	Damage Stock details are added to the system
Pre-conditions	Administrator must be logged in.
Flow of events	<ol style="list-style-type: none"> 1) Administrator enter damage details of the stock 2) Check stock quantities and update quantities 3) Click save button 4) Save damage stock 5) Display message 6) Click edit to damage stock 7) Click delete to damage stock 8) Generate report automatically
Post -condition	All data are save in the database

Table B.9 Use case description for damaged stock module

Use case diagram for User Module

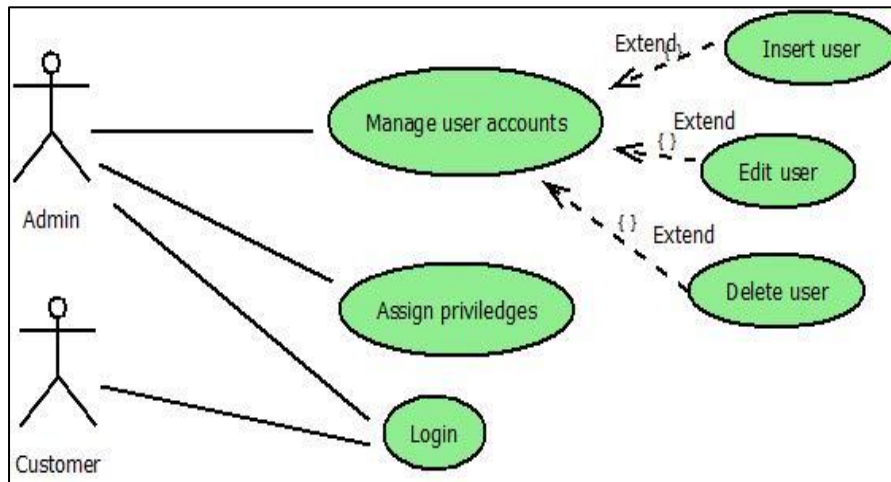


Figure B.7 Use case diagram of the User management module

Use Case Name	User Management
Actors	Administrator, Customer
Description	User details are added to the system
Pre-conditions	Administrator must be logged in.
Flow of events	<ol style="list-style-type: none"> 1) Administrator enter user details 2) Click save button 3) Click edit to user details 4) Click delete to user details 5) Assign user privileges
Post -condition	All data are save in the database

Table B.10 Use case description for User module

Use case diagram for Login track Module

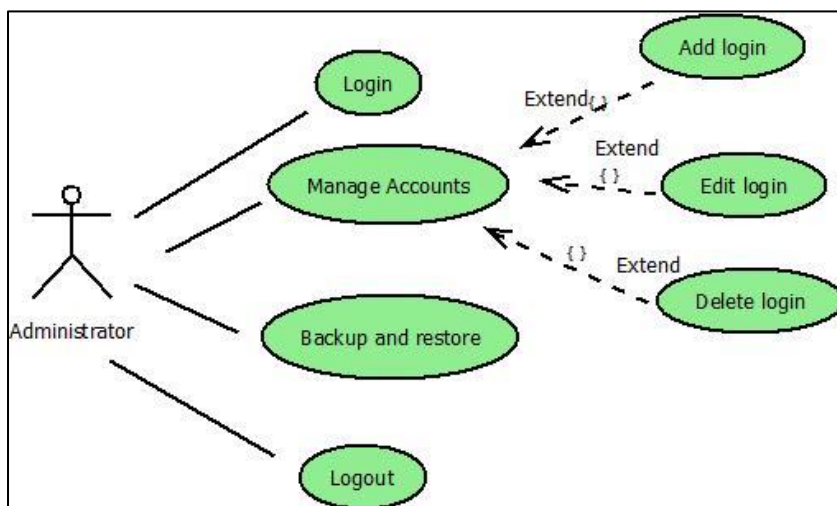


Figure B.8 Use case diagram of the Login management module

Use Case Name	Login Management
Actors	Administrator
Description	Login details are added to the system
Pre-conditions	Administrator must be logged in.
Flow of events	<ol style="list-style-type: none"> 1) Administrator check the system for about status of the login 2) View admin login details 3) View website login details
Post -condition	All data are save in the database

Table B.11 Use case description for User module

Use case diagram for Feedback Module

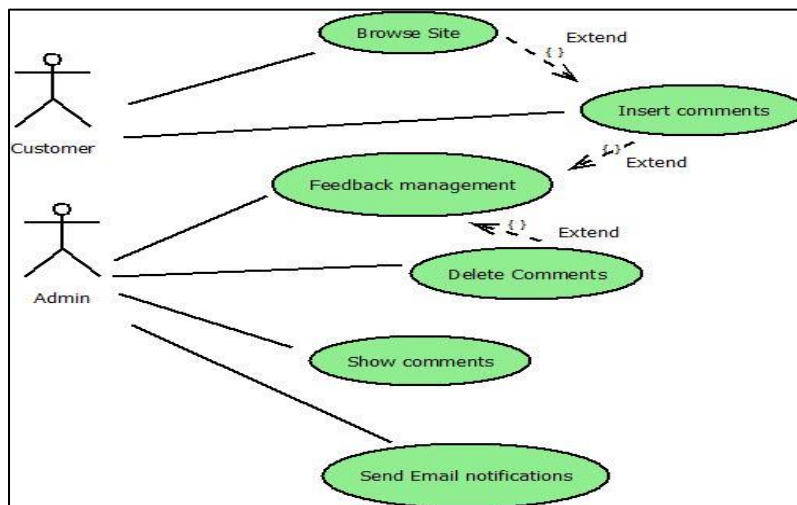


Figure B.9 Use case diagram of the feedback management module

Use Case Name	Feedback Management
Actors	Administrator , Customer
Description	Managing feedback
Pre-conditions	Administrator must be logged in.
Flow of events	<ol style="list-style-type: none"> 1) Customer logged in 2) Customer search products 3) Customer post a feedback in comment area 4) Email notification sent 5) Admin logged in to admin panel 6) View / delete feedback

Table B.12 Use case description for Feedback module

Use case diagram for Delivery Module

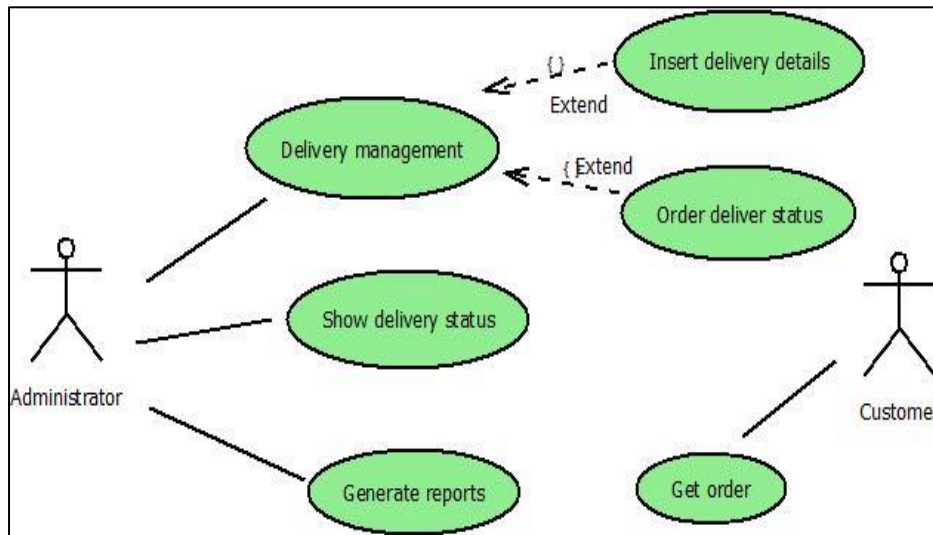


Figure B.10 Use case diagram of the Delivery management module

Use Case Name	Delivery Management
Actors	Administrator , Customer
Description	Managing deliveries
Pre-conditions	Administrator must be logged in.
Flow of events	<ol style="list-style-type: none"> 1) Administrator logged in 2) Insert Order no 3) Select delivery date 4) Insert Order status 5) View delivery report

Table B.13 Use case description for Delivery module

Use case diagram for Payment Module

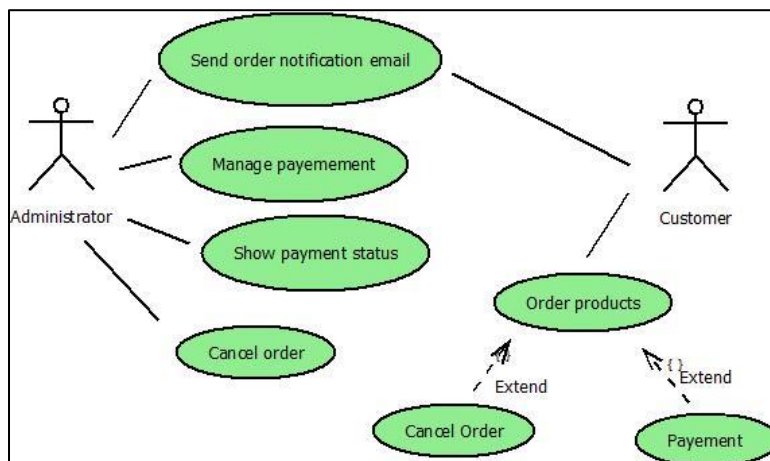


Figure B.11 Use case diagram of the Payment management module

Use Case Name	Payment Management
Actors	Administrator , Customer
Description	Order checkout process
Pre-conditions	Customer must be logged in.
Flow of events	<ol style="list-style-type: none"> 1) Customer logged in to website 2) Select orders 3) Select order quantities 4) Add products to shopping cart 5) Cancel order or process payment checkout

Table B.14 Use case description for payment module-Customer

Use Case Name	Payment Management
Actors	Administrator , Customer
Description	Managing payments
Pre-conditions	Administrator must be logged in.
Flow of events	<ol style="list-style-type: none"> 1) Administrator must logged in to admin panel 2) Check payment status 3) Check customer delivery status.

Table B.15 Use case description for payment module-Administrator

Activity Diagrams

Activity diagram for customer module

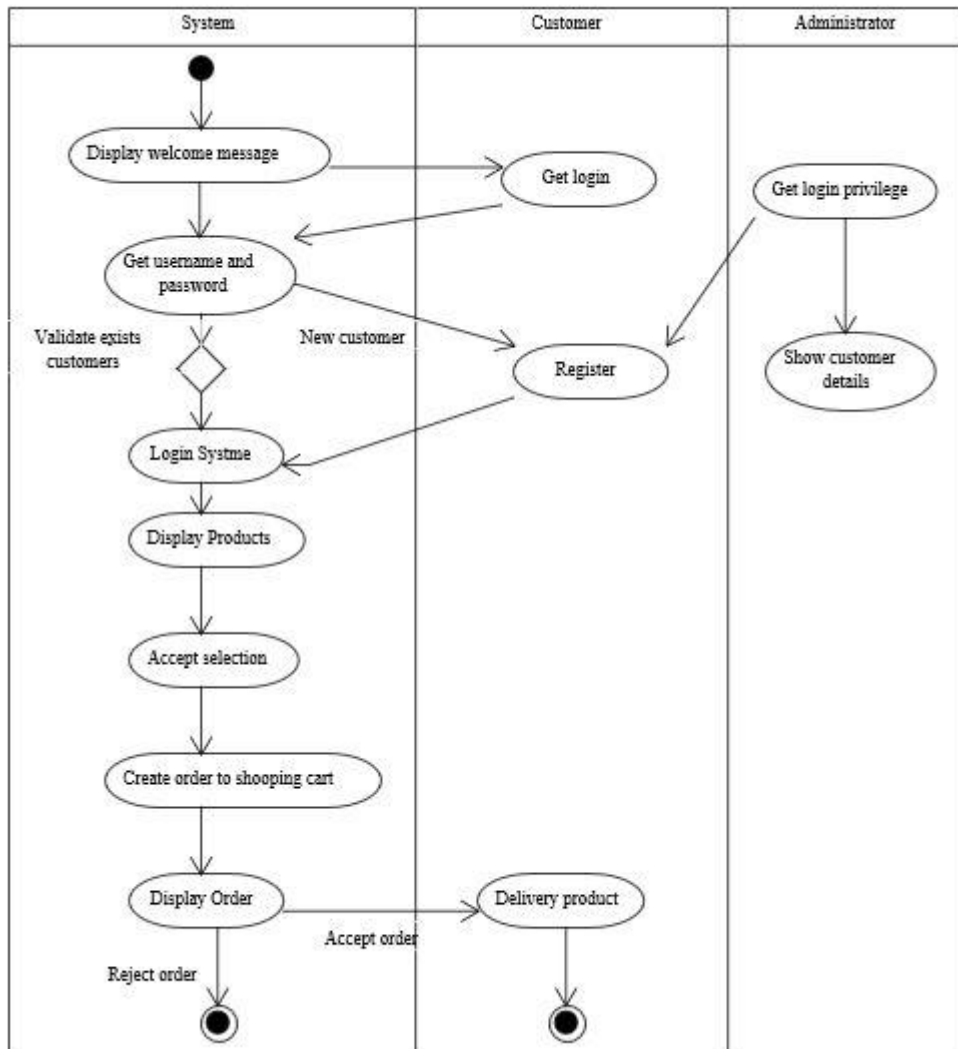


Figure B.12 Activity diagram of the Customer management module

Activity diagram for Stock module

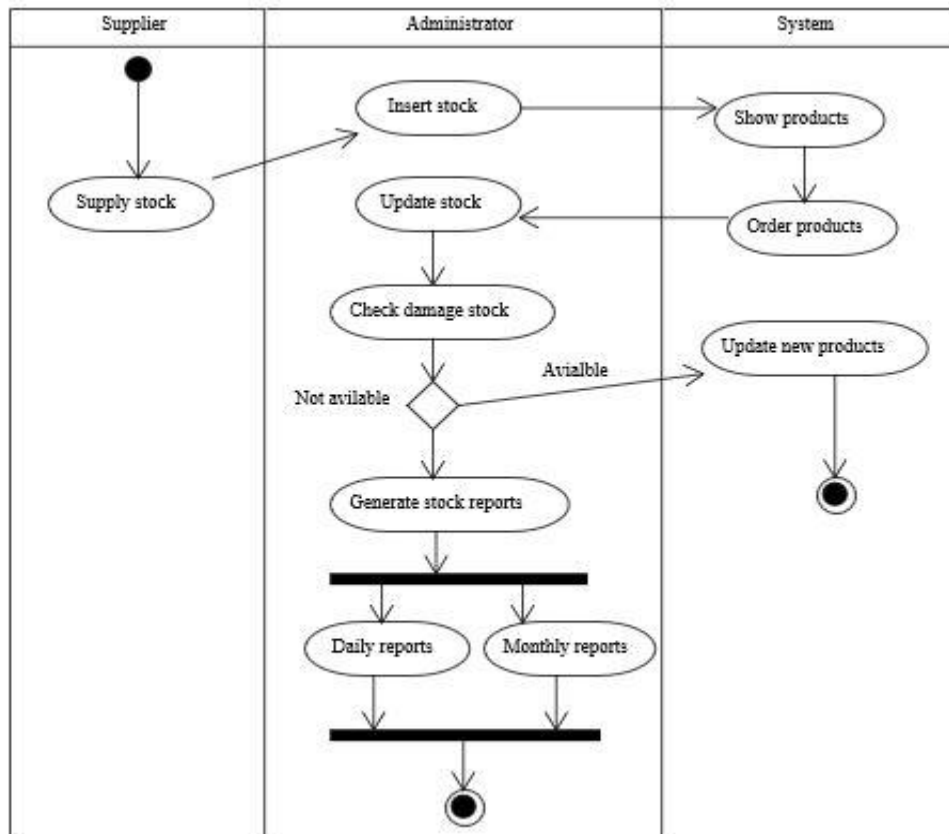


Figure B.13 Activity diagram of the Stock management module

Activity diagram for Product module

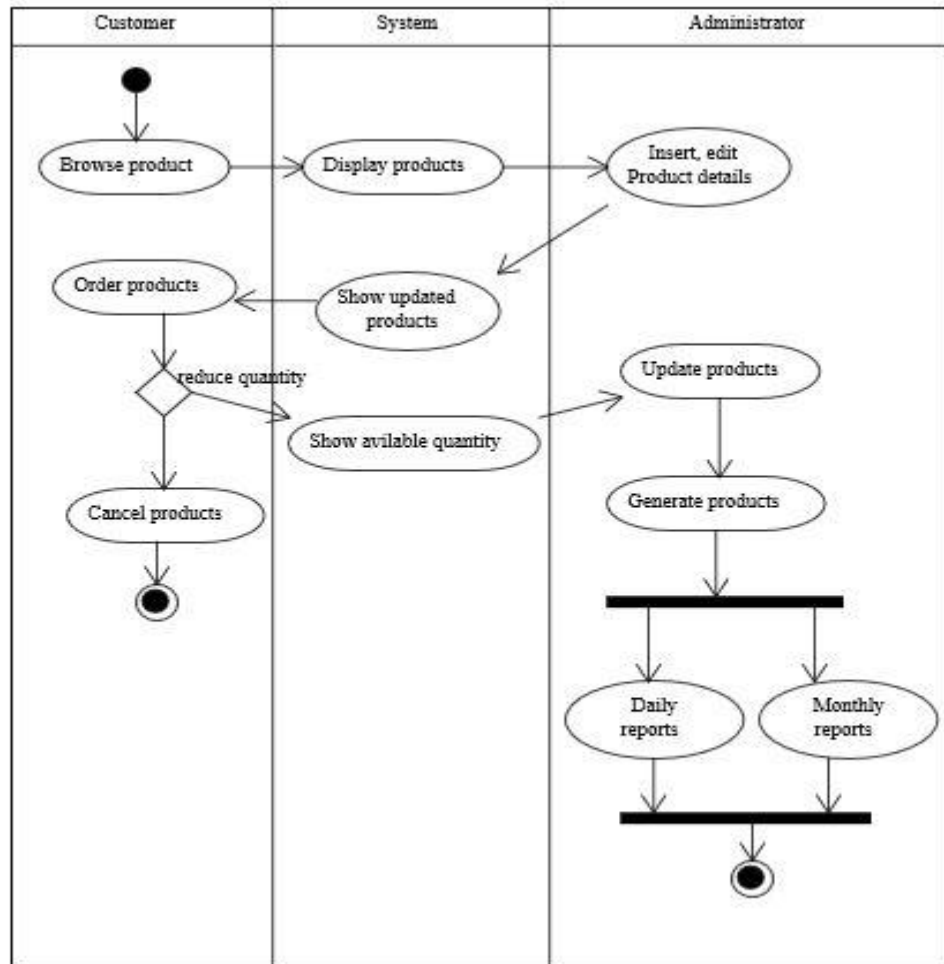


Figure B.14 Activity diagram of the Product management module

Sequence diagram for Customer module

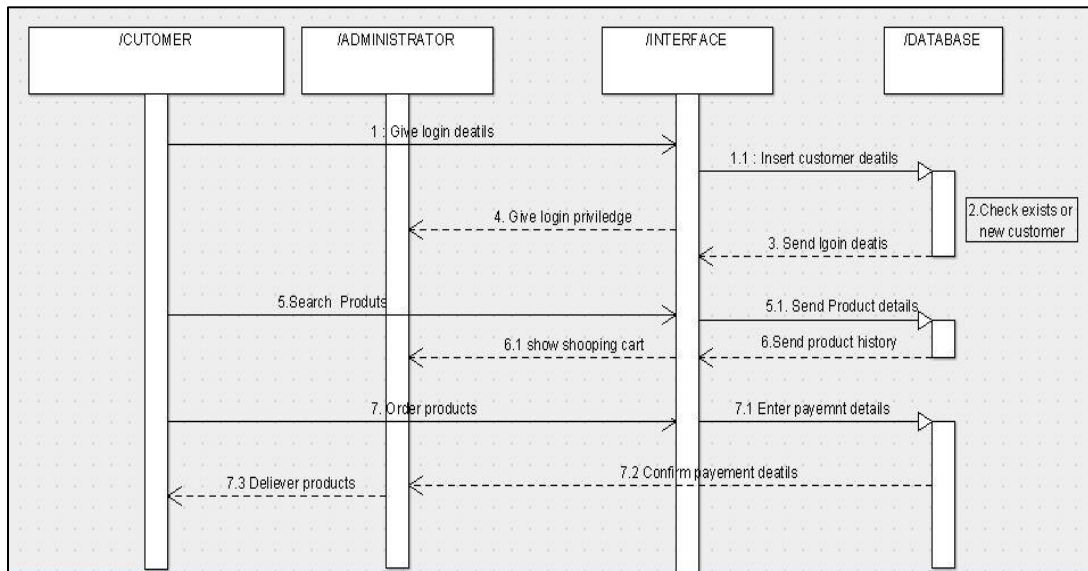


Figure B.15 Sequence diagram of the Customer management module

Sequence diagram for Payment module

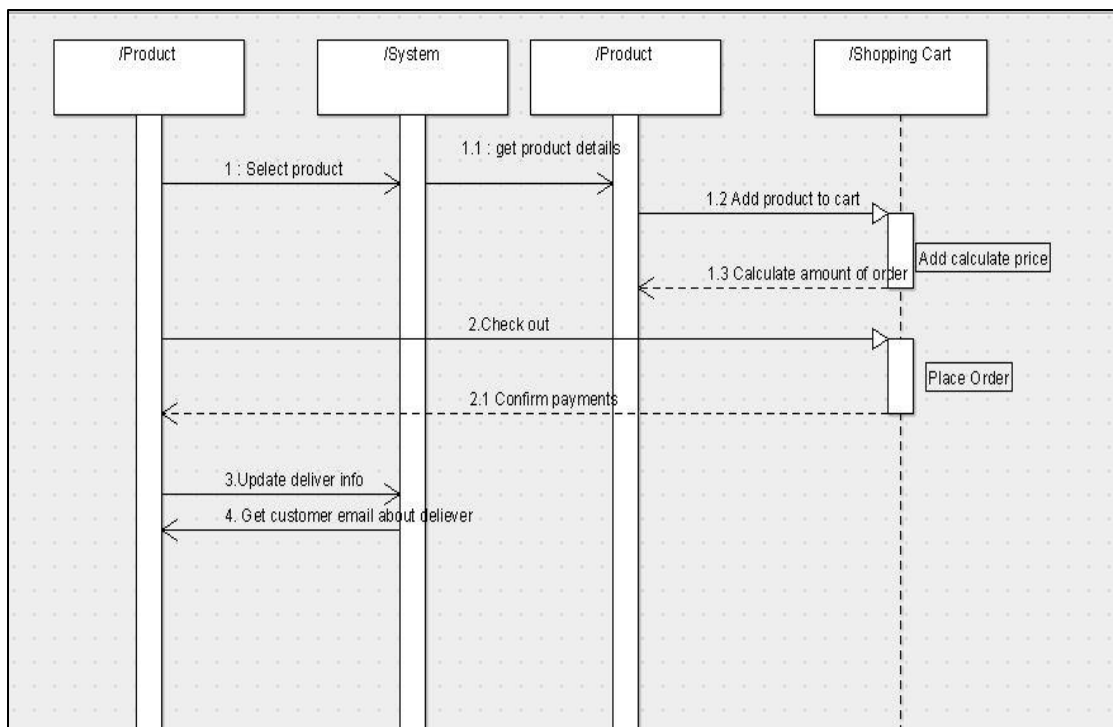


Figure B.16 Sequence diagram of the Payment management module

Appendix C- USER DOCUMENTATION

Admin Panel

- Login Form

Figure C.1 Display user login form , its allow users to log into admin panel. Administrator can log into the system in one form. When administrator try to log in the system check whether this user is valid login privilege o or not.

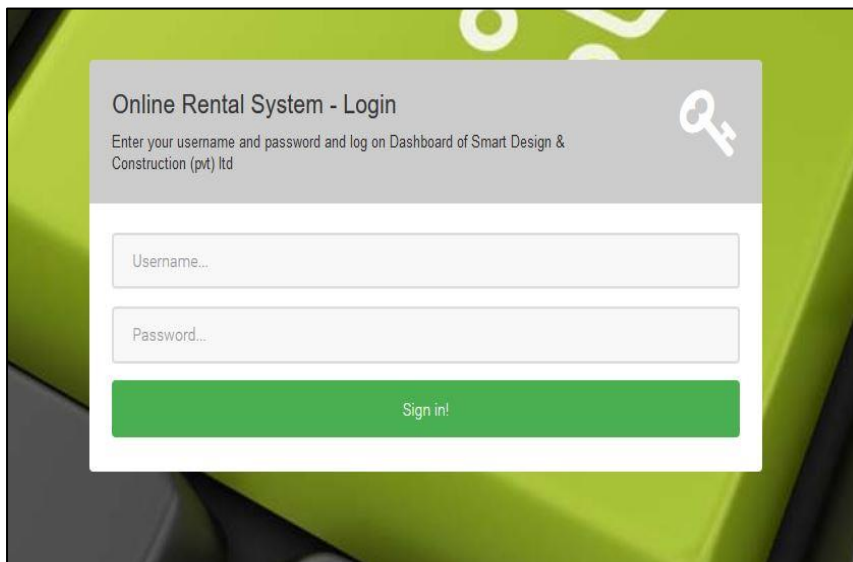


Figure C.1 Login Form

- Dashboard Form

Every module can be accessible by login users. And also they are is a main menu to easy accessible. Figure C.2 show how dashboard looks like.



Figure C.2 Dashboard Form

- Main menu

Figure C.3 show main menu in admin panel.



Figure C.3 Main menu

- Sub menu

Figure C.4 show sub menu items in admin panel.



Figure C.4 Sub menu items

- User management form

All the users' details can be stored via this form. User can do insert, delete , update and clear functions. And all insert data are displayed in view table. And also user can edit user data and can do update data. If user enter invalid data in form fields ,then display error messages. And also user try to save form with empty fields and also display error messages.

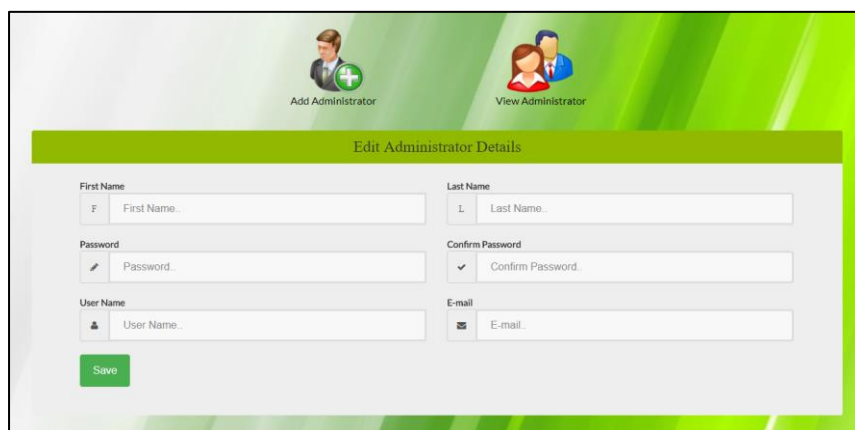


Figure C.5 User management form

- Customer management form

All the customer details can be stored via this forms such as customer name, customer address, customer email etc. Administrator can do insert, update, delete and clear functions via this form. Administrator must fill all the details of customer who was register with system website. If administrator try to fill invalid data, system display error messages. And he skip necessary fields without enter values, also display error messages. And also administrator can show all customer details in one table.

Add New Customer Details

Customer full Name:

Date of Birth:

Telephone No:

Email:

Address:

Description:

Figure C.6 Customer management form

Add Customer View Customer

ID	Customer Name	Date of Birth	Address	Email	Telephone No	Description	Edit	Delete
1	Sunil	0000-00-00	Rajagiriya	sunit@gmail.com	771232586			
2	Bhagya	1989-02-08	asdasd	bhagamadhawa@yahoo.com	12345678			
3	Amalh	1989-02-03	Boralla	amal@yahoo.com	122	bad		
4	amal	0000-00-00	Rajagiriya	amal@yahoo.com	122			

Figure C.7 View Customer management table

- **Product management form**

Figure C.8 show how product form looks like.

1. Type product Name.
2. Select category from combo box.
3. Insert relevant product image.
4. Type daily rental price
5. Insert product type.
6. Type product description
7. Click save button and data saved in product table in database
8. Click product view image and display all entered products data .C.9 show all products data

Figure C.8 Product management form

ID	Product Category	Product Sub Category	Product Name	Product price	Product Description	Edit	Delete
1	Scaffolding	Single scaffolding	Mason scaffolding	Rs.500.00	Saferstack 5 ft* 5 ft * 7ft		
2	Scaffolding	Double scaffolding	Meta-tech scaffolding	Rs.450.00	10ft * 7ft * 5ft scaffold double lift.		
3	Scaffolding	Double scaffolding	Double lift scaffolding	Rs.400.00	10ft * 7ft * 5ft Aluminum double lift scaffolding.		
4	Hand tools	Pliers	6 pliers set	Rs.250.00	8 inch diagonal cutting pliers.		
5	Hand tools	Hammer	Sledge hammer	Rs.170.00	Power strike sledge with fiberglass hammer.		
6	Hand tools	Hammer	Framing hammer	Rs.350.00	14 inch fiber glass handler.		

Figure C.9 View Product management table

- **Supplier management form**

All the Supplier details can be stored via this forms such as Supplier name, Supplier address, Supplier email etc. Administrator can do insert, update, delete and clear functions via this form. If administrator try to fill invalid data, system display error messages. And he skip necessary fields without enter values, also display error messages. And also administrator can show all supplier details in one table.

- **Stock management form**

All the stock details can be stored via this form. Administrator must have to update this data. He can do insert , edit , update and delete data. According to this form he has to select product name and Product’s supplier name. Product quantity automatically load form product table. Figure C.10 shows stock form .

And also administrator must have to update damaged stock form. He has to insert damaged stock quantity for future use. After he entered data of this form ,the website Order quantities will be automatically update.

Figure C.10 Stock management form

- Order management form

Administrator can check order status from admin order management form. He can show order no, Customer name, his ordered list, Order quantity, relevant total price. And customer already not paid for order administrator can cancel order from this table. And customer get email notification about cancel order.

- Payment management form

Administrator can view income report from payment management form. When he select relevant date range, he can see the income report from here.

Income Report			
Reports from 2017-11-06 To 2017-12-06			
ID	Order Date	Order No	Orders Price
1	2017-11-27 19:32:58	ODR11	Rs.2,000
2	2017-11-29 10:52:01	ODR12	Rs.20,000
3	2017-11-29 10:58:48	ODR13	Rs.96,000
4	2017-11-29 13:53:06	ODR14	Rs.1,050
5	2017-12-01 00:10:30	ODR18	Rs.500
6	2017-12-01 00:10:30	ODR18	Rs.200
7	2017-12-01 15:11:11	ODR21	Rs.2,000
8	2017-12-01 15:11:11	ODR21	Rs.700
9	2017-12-01 17:41:50	ODR22	Rs.6,120
10	2017-12-01 17:41:50	ODR22	Rs.1,000
11	2017-12-01 20:37:24	ODR23	Rs.450
12	2017-12-02 07:15:48	ODR24	Rs.5,400
13	2017-12-02 13:49:35	ODR25	Rs.450

From Date	To Date	Total Order price
2017-11-06	2017-12-06	Rs.135,870

Figure C.11 View Income report

- Feedback management form

1. Customer logged into system.
2. Give comments to display products in website.
3. Administrator can show all comments in one table.
4. If he want to delete comments, he can delete it.



ID	Product Name	Customer Name	Comments	Delete
1	Mason scaffolding	Sanka Pathama	Good Product. I rented many times this products from here	
2	Chop saw	Nuwan Ranasinghe	Some of saw blades not work properly.	
3	Compact cutter	Gayana perera	Good Product	

Figure C.12 View Feedback management table

- Login track management form

After login to admin panel and website , Administrator can view the login status as a one table.

- Delivery management form

1. Administrator logged into system.
2. Select delivery date.
3. Insert Order no.
4. Select delivery area.
5. Insert delivery status and insert remarks for deliveries.
6. After he can view deliveries details.

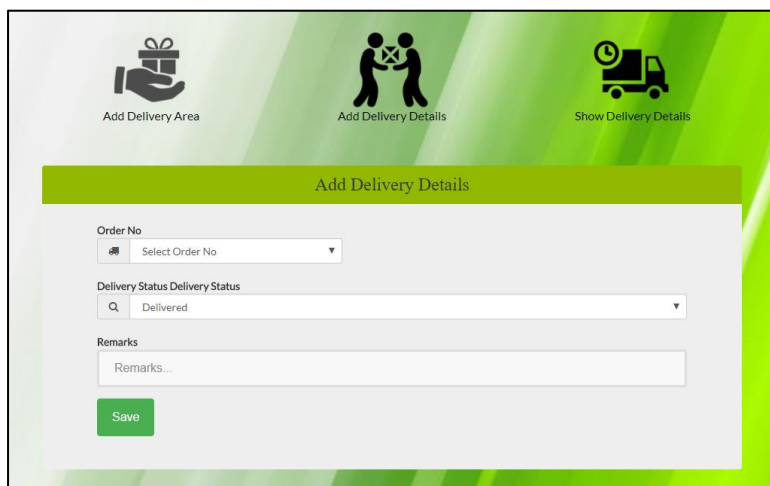


Figure C.13 Delivery management form

Website

Customer can rent products through system website. Before order the products must have to logged in. He was not logged in before he have to register on website. Figure C.14 register form and C.15 Login form on website.

REGISTRATION

Welcome, please enter registrations details below. If you have registered with **Smart design & Construction (pvt) ltd**, [click here to Login](#)

Customer Name*

Password*

Customer Date of Birth*

Customer Address*

Customer Email*

Customer Telephone No*

REGISTER NOW

Figure C.14 Website register form

REGISTERED CUSTOMERS

If you have an account with us, please log in.

Email Address*

Password*

LOGIN

Figure C.15 Website login form

After logged in to website he can browse rent products. Customer can browse products through website menu. And after display all products images, products daily rent, and also products description. As well as left hand side there is product categories.

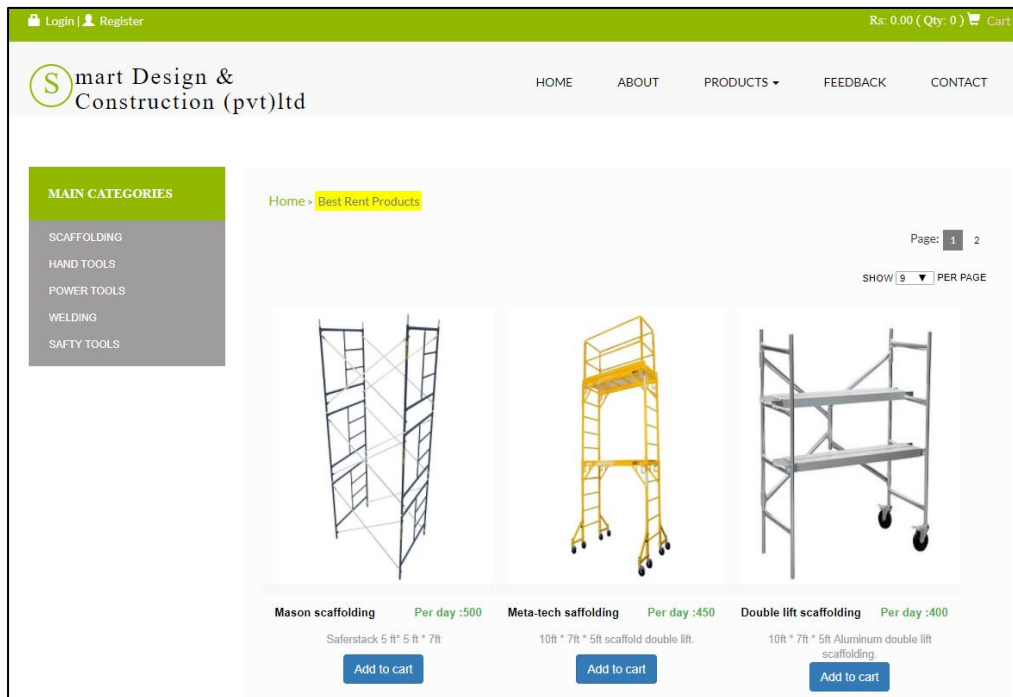


Figure C.16 Website products page

Customer click on the product image, browser change product page to single product display page. And then customer can enter quantity of product and click on the Add to Cart button to order product. And also there is comment area. If customer want to type comment about display product he can post a comment on there.

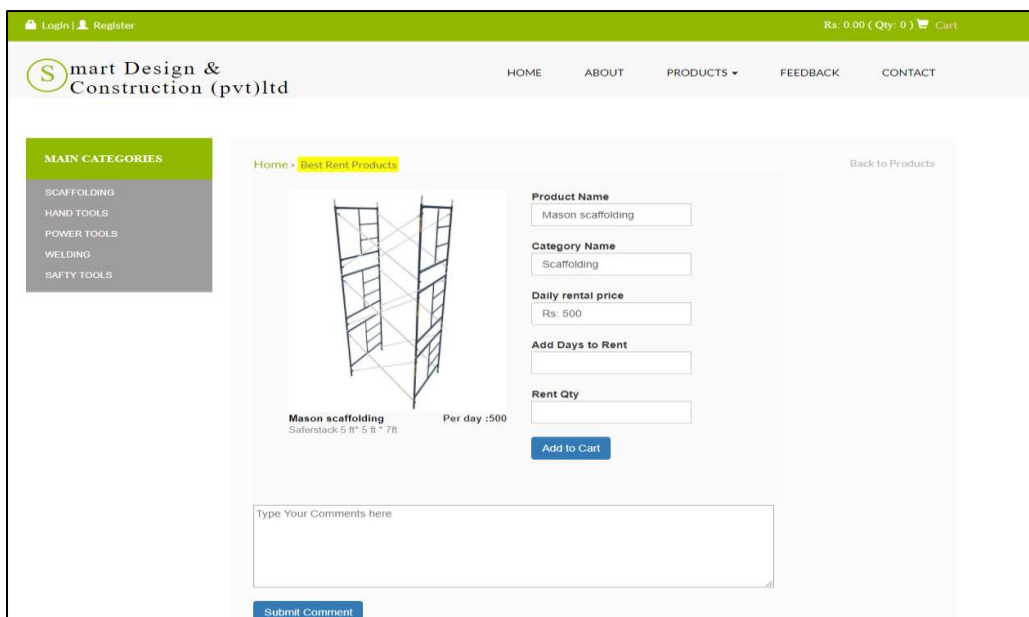


Figure C.17 Display single products page

And then Products add to shopping cart

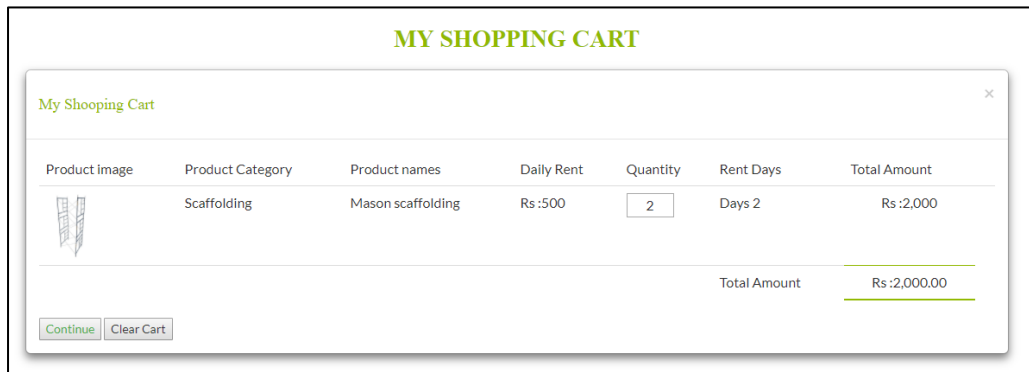


Figure C.18 Add to shopping cart view

After he add orders to shopping cart customer has to insert customer details and credit/debit card details. That is checkout process in this website. After processing customer can get his order.

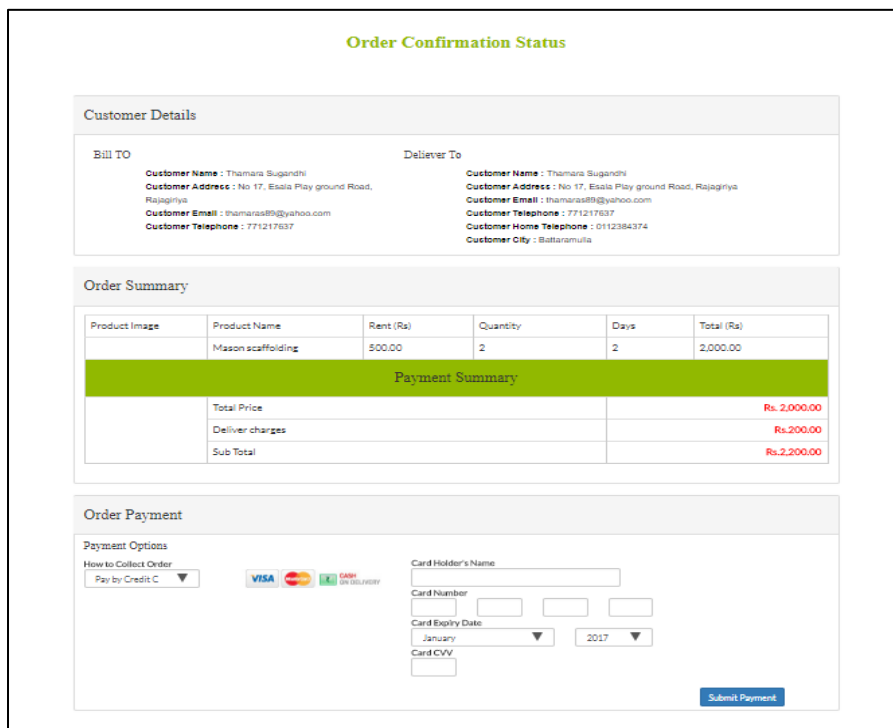


Figure C.19 Checkout process

If administrator want to change website content without coding, he can do it easily from admin panel. He can change website contents and images. Figure C.20, C.21, C.22 shows about us page ,home page and Contact us page in website.

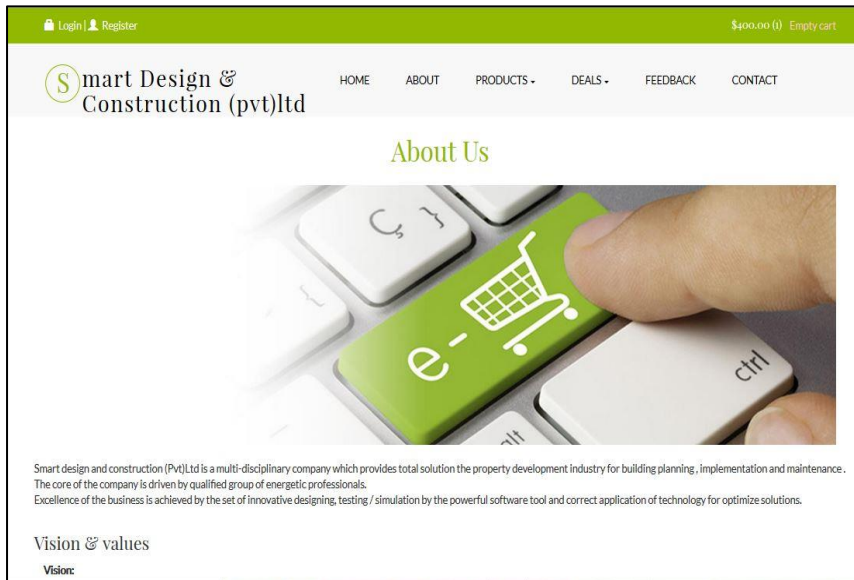


Figure C.20 About us website page



Figure C.21 Website home page

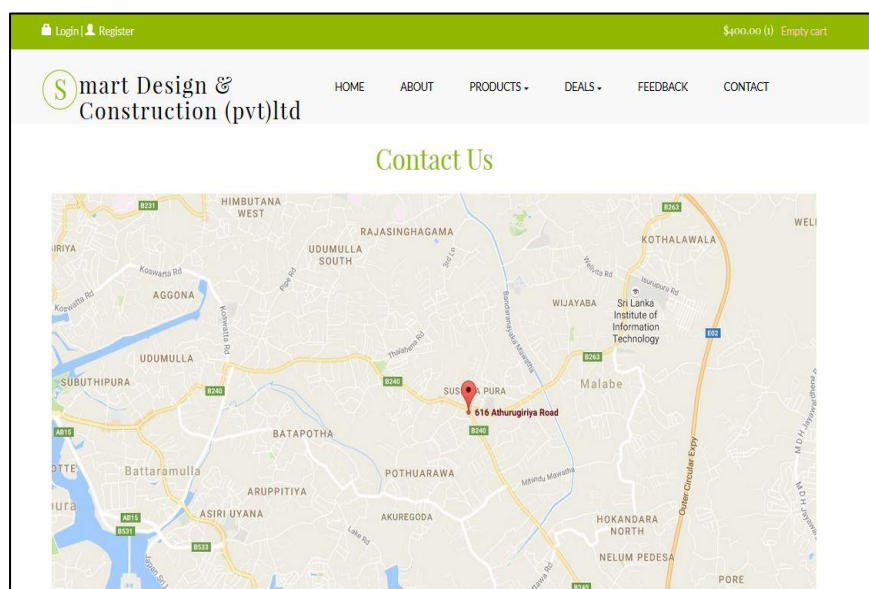


Figure C.22 Website contact us page

Appendix D-MANAGEMENT REPORTS

Product Report

Figure D.1 show Product report in Online Service Rental system.

Smart Design & Construction (pvt)ltd				
Product Report				
ID	Product Name	Product Category	Product Rent price	Product Description
1	Mason scaffolding	Scaffolding	Rs.500	Saferstack 5 ft* 5 ft * 7ft
2	Meta-tech scaffolding	Scaffolding	Rs.450	10ft * 7ft * 5ft scaffold double lift.
3	Double lift scaffolding	Scaffolding	Rs.400	10ft * 7ft * 5ft Aluminum double lift scaffolding.
4	6 pliers set	Hand tools	Rs.250	8 inch diagonal cutting pliers.
5	Sledge hammer	Hand tools	Rs.170	Power strike sledge with fiberglass hammer.
6	Framing hammer	Hand tools	Rs.350	14 inch fiber glass handler.
7	Bolt cutter	Hand tools	Rs.200	14 inch bolt cutter.
8	Compact cutter	Hand tools	Rs.300	9-1/2 inch compact cutter.
9	3 set sledge hammer	Hand tools	Rs.250	4 inch fiber glass handler.
10	RYOBI drill	Power tools	Rs.400	5.5 - amp 3/8 speed drill.
11	Codded drill	Power tools	Rs.300	18 volt ONE+lithium-ion codded drill.
12	Chop saw	Power tools	Rs.350	15 Amp 14 inch cut off saw.
13	Black helmets	Welding	Rs.200	Welding helmets with No. 10
14	Flip up welding helmets	Welding	Rs.300	Flip-Up Welding Helmet with Large No. 11 Lens

Hot Line: (+94)70 2510610 | (+94)70 3510610

Figure D.1 Product report

Category Report

Figure D.2 show Category report in Online Service Rental system.

Category Report		
ID	Category Name	Description
1	Scaffolding	Scaffolding, also called scaffold or staging, is a temporary structure used to support a work crew and materials to aid in the construction, maintenance and repair of buildings, bridges and all other man made structures
2	Hand tools	Hand tool is any tool that is powered by hand rather than a motor. Categories of hand tools include wrenches, pliers, cutters, striking tools, struck or hammered tools, screwdrivers, vises, clamps, snips, saws, drills and knives
3	Power tools	A power tool is a tool that is actuated by an additional power source and mechanism other than the solely manual labour used with hand tools.
4	Welding	Welding Equipments are Mig Welders, Arc Welders, Soldering Irons and Plasma Cutters. As well as Mig welders and welding equipment etc.
5	Safty tools	Its include protective gloves. Hearing protection. Full face shields when cutting, grinding, or chipping. Chemical splash goggles.

Figure D.2 Category report

Stock Report

Figure D.3 show Stock report in Online Service Rental system.

Stock Report					
ID	Product Name	Stock Qty	Ordered Qty	Damaged Qty	Available Qty
1	Yellow hats	50	10	3	37
2	Red stop sign	42	37	2	3
3	Black helmets	25	6	2	17
4	Flip up welding helmets	33	0	3	30
5	6 pliers set	22	0	5	17
6	Sledge hammer	18	12	0	6
7	Framing hammer	37	0	0	37
8	Bold cutter	23	0	0	23
9	Compact cutter	12	6	0	6
10	RYOBI drill	20	0	4	16
11	Codded drill	33	0	0	33
12	Chop saw	25	2	3	20
13	Mason scaffolding	17	3	4	10
14	Meta-tech scaffolding	16	2	0	14
15	Double lift scaffolding	21	0	2	19

Hot Line: (+9

Figure D.3 Stock report

Income Report

Figure D.4 show Income report in Online Service Rental system.

Income Report			
Reports from 2017-11-06 To 2017-12-06			
ID	Order Date	Order No	Orders Price
1	2017-11-27 19:32:58	ODR11	Rs.2,000
2	2017-11-29 10:52:01	ODR12	Rs.20,000
3	2017-11-29 10:58:48	ODR13	Rs.96,000
4	2017-11-29 13:53:06	ODR14	Rs.1,050
5	2017-12-01 00:10:30	ODR18	Rs.500
6	2017-12-01 00:10:30	ODR18	Rs.200
7	2017-12-01 15:11:11	ODR21	Rs.2,000
8	2017-12-01 15:11:11	ODR21	Rs.700
9	2017-12-01 17:41:50	ODR22	Rs.6,120
10	2017-12-01 17:41:50	ODR22	Rs.1,000
11	2017-12-01 20:37:24	ODR23	Rs.450
12	2017-12-02 07:15:48	ODR24	Rs.5,400
13	2017-12-02 13:49:35	ODR25	Rs.450

From Date	To Date	Total Order price
2017-11-06	2017-12-06	Rs.135,870

Figure D.4 Income report

Customer Report








Figure D.5 show Customer report in Online Service Rental system.

Customer Report					
ID	Customer Name	Date of Birth	Address	Email	Telephone No
1	Charith Madhusanka	1993-12-04	21-B, Mirihana road, Mirihana	charith993@yahoo.com	712770738
2	Wimal Abewikrama	1993-12-04	No 55-B, Rajamalwatha road, Battaramulla	wimalAbe@yahoo.com	785632536
3	Sunil Nanayakra	1988-07-07	No 4/C1, Rajagiriya road, Rajagiriya	sunil23232@yahoo.com	782323232
4	Ayesh Dusmanth	1987-02-09	No 15, Kottawa road, Pannipitiya	ayeshdd@yahoo.com	724673699
5	Chathura Maduranga	1990-12-08	No 123B, Mahragama road, Pannipitya	chathura@gmail.com	753235625
6	Kamal de Soysa	1977-12-04	kamlade@yahoo.com	kaml652@yahoo.com	782362363
7	Udesh Chathuranga	1997-01-06	No 67, Piris Mawatha, Nugegoda	udesh67@yahoo.com	764589676
8	Thamara Sugandhi	1989-02-04	No 17, Esala Play ground Road, Rajagiriya	thamaras89@yahoo.com	771217637
9	Sanka Pathama	1991-12-12	No 1/A, Rajagiriya Road, Rajagiriya	sanka@yahoo.com	712372323
10	Ruwan Wickramasinghe	1987-12-03	No 2/A, Wickramasinghe pura, Battaramulla.	ruwan@yahoo.com	723437467
11	Udesh Malinga	1979-04-08	No 123-23, Mahwatta road, Maharagma	udesh@yahoo.com	785365265
12	Nuwan Ranasinghe	1992-03-03	No 456-B,	nuwan@yahoo.com	718723878
13	Lakamal Perera	1984-02-06	No 12-A, Kottawa Road, Kottawa	lakmal@gmail.com	753452432
14	Gayan Maduranga	1977-01-12	No 12/12B, Piris Mawatha, Boralla	gayan@yahoo.com	776153625

Figure D.5 Customer report

Appendix E –TEST RESULTS

• Test cases for Product management Module

No	Test Case	Expected Output	Actual Output	Status
1	When required fields are not entered	Prompts user to enter required fields	Prompts user to enter required fields 	Pass
2	When all required fields are filled and then click save button.	Finally display “Saved Successful” message.		Pass
3	When Administrator filled wrong details about products which not match with product database.	Display “Please try again” error message.		Pass
4	When search all products	Display all products in one table.	Display all products in one table.	Pass
5	When administrator want to edit products details click edit button.	Redirect to Edit page	Redirect to Edit page. 	Pass
6	When administrator want to delete products details click delete button.	Click delete button display “Delete Successfully” messages.		Pass
7	When click product category combo box, all categories loaded.	Display all product categories.		Pass
8	Insert product image to load image to products web page.	Click image file to insert image.		Pass

9	Search details by Product name, Product type.	Relevant rows should be filtered	Relevant rows should be filtered	Pass
10	Search details by Product daily rent price.	Relevant rows should be filtered	Relevant rows should be filtered	Pass

Table E.1 Test cases of Product module

• **Test cases for User management Module**






No	Test Case	Expected Output	Actual Output	Status
1	When required fields are not entered	Prompts user to enter required fields	Prompts user to enter required fields 	Pass
2	When all required fields are filled and then click save button.	Finally display “Saved Successful” message.		Pass
3	When Administrator filled wrong details about administrator which not match with administrator database.	Display “Please try again” error message.		Pass
4	When search all administrators	Display all administrators in one table.	Display all administrators in one table.	Pass
5	When administrator want to edit user details click edit button.	Redirect to Edit page	Redirect to Edit page. 	Pass
6	When administrator want to delete user details click delete button.	Click delete button display “Delete Successfully” messages.		Pass

Table E.2 Test cases of User module

• Test cases for Customer management Module






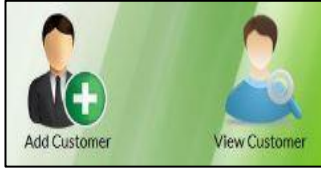
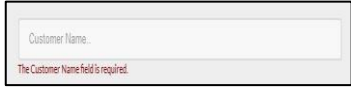



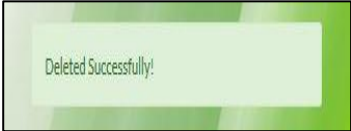


No	Test Case	Expected Output	Actual Output	Status
1	When required fields are not entered	Prompts user to enter required fields	Prompts user to enter required fields 	Pass
2	When all required fields are filled and then click save button.	Finally display “Saved Successful” message.		Pass
3	When Administrator filled wrong details about customer which not match with customer database.	Display “Please try again” error message.		Pass
4	When search all customers.	Display all customers in one table.	Display all customers in one table.	Pass
5	When administrator want to edit customers details click edit button.	Redirect to Edit page	Redirect to Edit page. 	Pass
6	When administrator want to delete customers details, click delete button.	Click delete button display “Delete Successfully” messages.		Pass
7	When administrator can easy navigate to add new and view button.	Click Add new button and View button.		Pass
9	Search details by Customer name, Customer email, Customer address.	Relevant rows should be filtered	Relevant rows should be filtered	Pass
10	Search details of Customer telephone no	Relevant rows should be filtered	Relevant rows should be filtered	Pass

Table E.3 Test cases of Customer module

• Test cases for Stock management Module

No	Test Case	Expected Output	Actual Output	Status
1	When required fields are not entered	Prompts user to enter required fields	Prompts user to enter required fields 	Pass
2	When all required fields are filled and then click save button.	Finally display “Saved Successful” message.		Pass
3	When Administrator filled wrong details about stock which not match with customer database.	Display “Please try again” error message.		Pass
4	When search all stocks.	Display all stocks in one table.	Display all stocks in one table.	Pass
5	When administrator want to edit stocks details click edit button.	Redirect to Edit page	Redirect to Edit page. 	Pass
6	When administrator want to delete stocks details, click delete button.	Click delete button display “Delete Successfully” messages.		Pass
7	When administrator can easy navigate to add new and view button.	Click Add new button and View button.		Pass
9	Search details by product name, Supplier name.	Relevant rows should be filtered	Relevant rows should be filtered	Pass
10	When change stocks quantity. Administrator Want to insert quantity.	Relevant rows should be filtered		Pass



11	When change Supplier name. Administrator Want to insert supplier name in combo box.	Select name from combo box.		Pass
12	When change Product name. Administrator Want to insert Product name in combo box.	Select Product name from combo box.		Pass

Table E.4 Test cases of Stock module

• **Test cases for Feedback management Module**



No	Test Case	Expected Output	Actual Output	Status
1	When click the “send mail” button	Display “Sent Successful” message.	Display “Sent Successful” message.	Pass
2	When type post comment	Display comments in product page		Pass
3	When Administrator want to display comments.	Display all comments in one table	Display all comments in one table	Pass
4	When Administrator want to delete comments.	Click delete button display “Delete Successfully” messages.		Pass

Table E.5 Test cases of Feedback module

• **Test cases for Report management Module**

No	Test Case	Expected Output	Actual Output	Status
1	When select the date and click the daily button	Daily income report should be generated	Daily income report should be generated	Pass
2	When select the date and click the Monthly button	Monthly income report should be generated	Monthly income report should be generated	Pass


3	When select the date range and click button	Date range income should be generated		Pass
4	When click product report button	Display all product details as a report	Display all product details as a report	Pass
5	When click Customer report button	Display all Customer details as a report	Display all Customer details as a report	Pass
6	When click Stock report button	Display all Stock details as a report	Display all Stock details as a report	Pass
7	When click Order report button	Display all Order details as a report	Display all Order details as a report	Pass

Table E.6 Test cases of Report module

• **Test cases for Payment management Module**

No	Test Case	Expected Output	Actual Output	Status
1	When administrator want show customer name who ordered product	Display customer name	Display customer name in payment status table	Pass
2	When administrator want show ordered list	Display ordered list	Display ordered list in payment status table	Pass
3	When administrator want show ordered list total amount	Display total amount	Display total amount in payment status table	Pass
4	When administrator want show delivered status	Display delivered status	Display delivered status in payment status table	Pass

Table E.7 Test cases of Payment module

• **Test cases for Order management Module**

No	Test Case	Expected Output	Actual Output	Status
1	When administrator want show customer name who ordered product	Display customer name	Display customer name in order status table	Pass
2	When administrator want show ordered list	Display ordered list	Display ordered list in order status table	Pass

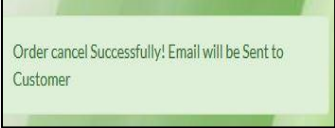
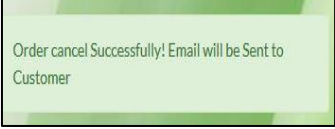
3	When administrator want cancel order and click cancel button	Display “ cancel order successfully “ message		Pass
4	When administrator want cancel order and send email to customer	Display “Email will be sent to customer “ message		Pass

Table E.8 Test cases of Order module

• **Test cases for Deliverer management Module**







No	Test Case	Expected Output	Actual Output	Status
1	When required fields are not entered	Prompts user to enter required fields	Prompts user to enter required fields 	Pass
2	When all required fields are filled and then click save button.	Finally display “Saved Successful” message.		Pass
3	When Administrator filled wrong details about products which not match with product database.	Display “Please try again” error message.		Pass
4	When show all deliveries	Display all deliveries in one table.	Display all deliveries in one table.	Pass
5	When select delivery date	Display calendar		Pass
6	When select deliver area.	Select deliver area from combo box.		Pass
7	When select deliver status	Select deliver status from combo box.		Pass

Table E.9 Test cases of Delivery module

Appendix F-CODE LISTING

Online service rental system has adapted MVC design pattern. Codes related to each function, comprises of separate Model View and Controller. Some important code segments are explained below

- **Code for User management module**

Insert user- Model

Administrator_model.php -Model

```

Class Administrator_model extends CI_Model{

    function insert_user()
    {
        $adfirst = $this->input->post('form-adfirst');
        $adlast  = $this->input->post('form-adlast');
        $adname  = $this->input->post('form-adname');
        $pass    = $this->input->post('form-adpass');
        $email   = $this->input->post('form-ademail');

        $insert_data = array(
            'adfirst' => $adfirst,
            'adlast'  => $adlast,
            'adname'  => $adname,
            'pass'    => md5($pass),
            'email'   => $email
        );

        $this->db->insert('administrator', $insert_data );
        return 1;
    }
}

```

Insert user- Controller

Administrator.php - Controller

```

public function index()
{
    $this->form_validation->set_rules('form-adfirst', 'First Name', 'required');
    $this->form_validation->set_rules('form-adlast', 'Last Name', 'required');
    $this->form_validation->set_rules('form-adname', 'User Name', 'required');
    $this->form_validation->set_rules('form-adpass', 'Password', 'required');
    $this->form_validation->set_rules('form-adcompass', 'Confirm Password', 'required');
    $this->form_validation->set_rules('form-ademail', 'Email', 'required');

    if($this->form_validation->run() == FALSE)
    {
        $this->load->view('admin/common/header');
        $this->load->view('admin/settings/administrator');
        $this->load->view('admin/common/footer');
    }

    else
    {
        $res = $this->administrator_model->insert_user();
        if($res == 1)
        {
            $message_success = '<div class="alert alert-success">Saved Successfully!</div>';
            $this->session->set_flashdata('message', $message_success);
            redirect('admin/settings/administrator');
        }
        else
        {

```

Given that the model interacts with the database, it should be loaded within the model for it to use. This can be achieved by placing a call to database in the model's constructor as shown in the following code snippet.

```
function __construct() {
    parent::__construct();
    if (!$this->session->userdata('admin_logged_in')) {
        redirect('admin/login', 'refresh');
    }
    $this->load->model('administrator_model');
}
```

Insert user- View

This code show first name field in form. Other last name ,email address, telephone no, password, address ,date of birth and description from tags are looks like this.

Administrator.php - View

```
<div class="form-bottomnew">
    <?php
    $attributes = array('class' => 'ad-form', 'id' => 'adform', 'name'=>'adform');
    echo form_open('admin/settings/administrator', $attributes);
    ?>

    <div class="form-group">
        <input type="text" name="adId" id="adId" style="display:none;" value="<?php if(isset($administrator_data)) { echo $administrator_data[0]->adId;} ?>">

        <label class="sr-only" for="form-adfirst">First Name</label>
        <input type="text" name="form-adfirst" placeholder="First Name.." class="form-control" id="form-adfirst" value="<?php if(isset($administrator_data)) { echo $administrator_data[0]->adfirst;} else {echo set_value('form-adfirst');} ?>">

        <?php if(form_error('form-adfirst')){ ?>
        <div class="err_danger"><?php echo form_error('form-adfirst');?></div>
        <?php
        }
    ?>

    </div>
```

Update user

This code shows how the data is taken from database referring to an ID passed by the controller \$_POST variable. \$res['administrator_data'] passed from controller.

```
<input type="text" name="form-adfirst" placeholder="First Name.." class="form-control" id="form-adfirst" value="<?php if(isset($administrator_data)) { echo $administrator_data[0]->adfirst;} else {echo set_value('form-adfirst');} ?>">
```

This code shows how the data taken from database with sql query. This is the model update code.

```
function get_administrator_data($adId){
    $data = array(
        'adId' => $adId
    );

    $this->db->select('*');
    $this->db->where($data);
    $this->db->from('administrator');

    $query = $this->db->get();
    $res_array = array();

    foreach ($query->result() as $row){
        $res_array[]=$row;
    }
    return $res_array;
}
```

Once the administrator clicks the edit button in the form, the values are again validated through the controller and validates again in PHP and pass it to the model through a statement like given below.

```
$res['administrator_data'] = $this->administrator_model->get_administrator_data($adId);
```

Then administrator can edit user form. And updated database again.

Delete user

The Administrator clicks the delete button in the table in a given data view. The controller gets the ID through \$this->administrator_model->delete_administrator(\$adId) method and sends the value to the model. This is controller code.

```
$res = $this->administrator_model->delete_administrator($adId);
//var_dump($res);
```

This code shows how the data taken from database with sql query. This is the model delete function code.

```
function delete_administrator($adId){
    $data=array (
        'adstatus' => 0
    );

    $this->db->set($data);
    $this->db->where('adId', $adId);
    $this->db->update(administrator);

    return 1;
}
```

The model returns a message to controller and display the deletion was successful or not.

```
public function delete_administrator($adId){
    $res = $this->administrator_model->delete_administrator($adId);
    //var_dump($res);

    if($res == 1)
    {
        $message_success = '<div class="alert alert-success">Deleted Successfully!</div>';
        $this->session->set_flashdata('message', $message_success);
    }
    else{
        $message_error = '<div class="alert alert-danger">Please try again!</div>';
        $this->session->set_flashdata('message', $message_error);
    }
    redirect('admin/settings/administrator/view_admin');
}
```

View user

The Administrator clicks the view image icon ,Display all administrator details. The controller gets the ID through \$this->administrator_model->get_data(); method and sends the value to the model. This is controller code.

```
$data['admin_data'] = $this->administrator_model->get_data();
```

This code shows how the data taken from database with sql query. This is the model delete function code.

```
function get_data(){
    $this->db->select('*');
    $this->db->from('administrator');
    $this->db->where('adstatus', '1');

    $query = $this->db->get();

    $res_array = array();
    //var_dump($this->db->last_query());
    foreach ($query->result() as $row) {
        $res_array[] = $row;
    }

    return $res_array;
}
```

The model returns a message to controller and display administrator view page

```
public function view_admin(){
    $data['admin_data'] = $this->administrator_model->get_data();

    $this->load->view('admin/common/header');
    $this->load->view('admin/settings/administrator_view', $data);
    $this->load->view('admin/common/footer');
}
```


Appendix G – CLIENT CERTIFICATE



Smart Design and Construction (Pvt) Ltd
No. 616,
Athurugiriya Road,
Malabe.
TEL: 011 5683663/0702510610 web: www.smarts.lk
Email: smartsconstruction@gmail.com

26 October 2017

Coordinator,
BIT Degree,
University of Colombo,
School of Computing

Dear Sir,

Letter of Certification

This is to certify that Miss. N. T Thamara Sugandhi has successfully designed and developed a Web based online service rental system for Smart Design and Construction (pvt) ltd in Malabe. The project was undertaken by her as a partial fulfillment of a requirement for the Bachelor of Information Technology Degree program.

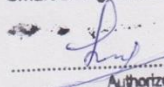
I am pleased to certify that the system developed by Miss. N. T Thamara Sugandhi fulfill the requirements of our Rental business and could be used as our Online Service rental System.

This certification is issued on the request of Miss. N. T Thamara Sugandhi.

Thank You,

Your faithfully,

Smart Design And Construction (Pvt) Ltd.


.....
Authorized Signatory

D.M.P.S Senarathne

Managing Director,

Smart Design and Construction (pvt) Ltd

Glossary

- Use Case Diagram - A use case diagram displays the relationship among actors and use cases.
- Activity Diagram – A diagram used to show the flow of a business process.
- Class diagram – shows the object classes of the system and the relationships between them.
- Database-Collection of data.
- GUI-Graphical user interface which user can interact with the system.

Index

A

activities 14, 47, 59
 analysis 16, 18, 21, 23, 66
 application..... 25, 49, 59, 66, 72

B

backups 72
 Beta..... vii, 58
 business xii, 23, 35, 36, 65, 66, 115

C

click ... 30, 37, 41, 99, 103, 104, 105, 106, 107, 108, 109
 client iii, xii, 25, 47, 66
 component..... 56, 58
 components 24, 48, 58, 69
 Customer... v, vi, 15, 16, 20, 21, 35, 39, 41, 60, 63, 75, 76, 79, 81, 83, 84, 86, 89, 92, 96, 98, 99, 105, 106, 108

D

damage 36, 37, 80
 data... 15, 17, 22, 23, 24, 28, 30, 31, 34, 38, 49, 51, 52, 53, 54, 55, 59, 74, 75, 77, 78, 79, 80, 81, 82, 91, 92, 93, 94, 110, 111, 112, 113, 114, 115
 database.. 21, 22, 28, 31, 36, 49, 51, 54, 55, 56, 71, 72, 74, 75, 77, 78, 79, 80, 81, 82, 93, 103, 104, 105, 106, 109, 111, 112, 113, 114
 design..iv, ix, 14, 16, 18, 24, 25, 26, 28, 29, 47, 67, 68, 73, 109
 document..... 14, 59, 64

E

efficiently iii, 29
 error 17, 30, 52, 53, 54, 56, 91, 92, 94, 103, 104, 105, 106, 109, 111, 113
 evaluation..... 17, 57, 64, 65

F

feedback 18, 29, 83

I

images 41, 42, 62, 74, 99, 100
 interfaces 22, 29, 53
 interviews 19

L

language iii, 32, 49, 67
 login..... 14, 32, 34, 40, 72, 82, 90, 96, 98, 111

M

manipulate 34
 message 30, 52, 54, 56, 74, 77, 78, 79, 80, 103, 104, 105, 106, 107, 109, 111, 113, 114
 messages.. 26, 30, 54, 56, 91, 92, 94, 103, 105, 106, 107, 108
 modelsxii, 51
 MySQL.....iii, 48, 49

N

nonfunctional 22
 notification 16, 35, 83

O

observation..... 19
 operations 29, 51

P

panel ... iv, vi, 32, 34, 42, 43, 70, 76, 83, 90, 91, 96, 100
 payment 14, 15, 22, 66, 76, 102
 policy..... 42, 45, 100
 Privacy..... 42, 45, 100
 programmer 47, 58

Q

Quantity 37, 62, 63
 questionnaire 64, 65

R

Rational xi, 25, 48, 68
 register.....39, 40, 76, 92, 98
 requirements . iii, 16, 18, 19, 21, 22, 25, 26, 47, 56, 57, 58, 66, 67

S

Scenarios 59
 Software iv, vi, vii, 25, 47, 48, 56, 57, 66, 68
Stock....v, vi, 15, 16, 20, 21, 37, 62, 79, 80, 86, 94, 95, 102, 106, 108
 structures 17, 49, 54

system iii, xii, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 25, 26, 29, 30, 32, 33, 34, 35, 36, 39,
47, 49, 51, 55, 56, 57, 58, 64, 65, 66, 72, 73,
74, 75, 76, 77, 78, 79, 80, 81, 82, 90, 92, 94,
96, 97, 98, 109, 114, 115

U

User. v, vi, vii, xi, 15, 16, 22, 29, 32, 60, 64, 80,
81, 82, 91, 104, 110

V

verification..... 56, 57
views..... 51, 53

W

website... 14, 32, 34, 39, 41, 42, 43, 44, 45, 46,
60, 61, 66, 76, 82, 92, 94, 96, 98, 99, 100