

LAUNDRY MANAGEMENT SYSTEM

Project Report Submitted By

TEBIN JOSEPH

Reg. No.: AJC20MCA-2079

In Partial fulfillment for the Award of the Degree Of

**MASTER OF COMPUTER APPLICATIONS
(MCA)**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



**AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2020-2022

DEPARTMENT OF COMPUTER APPLICATIONS
AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, “**LAUNDRY MANAGEMENT SYSTEM**” is the bonafide work of **TEBIN JOSEPH (Reg.No:AJC20MCA-2079)** in partial fulfillment of the requirements for the award of the Post Graduation Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2020-22.

Ms. Anit James
Internal Guide

Mrs.Grace Joseph
Coordinator

Rev.Fr.Dr.Rubin Thottupurathu Jose
Head of the Department

DECLARATION

I hereby declare that the project report “**LAUNDRY MANAGEMENT SYSTEM**” is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Post Graduation Degree of Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2020-2022.

Date: 23/02/2022

KANJIRAPPALLY

TEBIN JOSEPH

Reg. No: AJC20MCA-2079

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Mrs. Grace Joseph** for their valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide, **Ms. Anit James** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

TEBIN JOSEPH

ABSTRACT

I present the design and implementation of a laundry management system (LMS) used in a laundry establishment. Laundry firms are usually faced with difficulties in keeping detailed records of customers clothing; this little problem as seen to most laundry firms is highly discouraging as customers are filled with disappointments, arising from issues such as customer clothes mix-ups and untimely retrieval of clothes. The aim of this application is to determine the clothes collected, in relation to their owners, as this also helps the users fix a date for the collection of their clothes.

CONTENT

Sl. No	Topic	Page No
1	INTRODUCTION	2
1.1	PROJECT OVERVIEW	3
1.2	PROJECT SPECIFICATION	3
2	SYSTEM STUDY	4
2.1	INTRODUCTION	4
2.2	EXISTING SYSTEM	5
2.3	DRAWBACKS OF EXISTING SYSTEM	6
2.4	PROPOSED SYSTEM	6
2.5	ADVANTAGES OF PROPOSED SYSTEM	6
3	REQUIREMENT ANALYSIS	8
3.1	FEASIBILITY STUDY	9
3.1.1	ECONOMICAL FEASIBILITY	9
3.1.2	TECHNICAL FEASIBILITY	10
3.1.3	BEHAVIORAL FEASIBILITY	10
3.2	SYSTEM SPECIFICATION	11
3.2.1	HARDWARE SPECIFICATION	11
3.2.2	SOFTWARE SPECIFICATION	11
3.3	SOFTWARE DESCRIPTION	11
3.3.1	PHP	11
3.3.2	MYSQL	12
4	SYSTEM DESIGN	14
4.1	INTRODUCTION	15
4.2	UML DIAGRAM	15
4.2.1	USE CASE DIAGRAM	16
4.2.2	SEQUENCE DIAGRAM	19
4.5	USER INTERFACE DESIGN	21
4.6	DATA BASE DESIGN	24
5	SYSTEM TESTING	30
5.1	INTRODUCTION	31
5.2	TEST PLAN	32

5.2.1	UNIT TESTING	32
5.2.2	INTEGRATION TESTING	32
5.2.3	VALIDATION TESTING	32
5.2.4	USER ACCEPTANCE TASTING	33
6	IMPLEMENTATION	35
6.1	INTRODUCTION	36
6.2	IMPLEMENTATION PROCEDURE	36
6.2.1	USER TRAINING	37
6.2.2	TRAINING ON APPLICATION SOFTWARE	37
6.2.3	SYSTEM MAINTENANCE	37
7	CONCLUSION & FUTURE SCOPE	38
7.1	CONCLUSION	39
8	BIBLIOGRAPHY	40
9	APPENDIX	42
9.1	SAMPLE CODE	43
9.2	SCREEN SHOTS	53

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

I present the design and implementation of a laundry management system (LMS) used in a laundry establishment. Laundry firms are usually faced with difficulties in keeping detailed records of customers clothing; this little problem as seen to most laundry firms is highly discouraging as customers are filled with disappointments, arising from issues such as customer clothes mix-ups and untimely retrieval of clothes. The aim of this application is to determine the clothes collected, in relation to their owners, as this also helps the users fix a date for the collection of their clothes.

1.2 PROJECT SPECIFICATION

The proposed system is a website in which user can book online for laundry service. Also that the customers how come to shop also have access to the web were he/she can view all the laundry service.

The system includes 2 modules. They are:

1. Admin Module

- ❖ Admin login.
- ❖ Manage users.
- ❖ Manage dashboard.
- ❖ Manage laundry.
- ❖ View feedback from users

2. Customer Module

- ❖ User login.
- ❖ Edit user profile.
- ❖ Book slots for laundry washing.
- ❖ View laundry status.
- ❖ Make payments.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

Laundry firms currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the laundry management infrastructure. Often information (on forms) is incomplete or does not follow management standards. Records are often lost in transit during computation requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the laundry firm data and may lead to inconsistencies in data in various data stores. A significant part of the operation of any laundry firm involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves; customer personal information and clothing records history, user information, price of delivery and retrieval period, users scheduling as regards customers details and dealings in service rendered, also our products package waiting list. All of this information must be managed in an efficient and cost wise fashion so that the organization resources may be effectively utilized.

The goal of laundry management system is to automate the management of the laundry firm making it more efficient and error free. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies, through the use of highly computerized process that is stress free, reliable and quick through the use of asp.net computer programming language and SQL database application to both the users and the staff in charge of the registration and laundry management processes. HTML would be at the front-end and provide the graphical user interface that relates with the user, while the SQL database will be at the back-end to handle the data storage process.

2.2 EXISTING SYSTEM

Laundry firm currently uses a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the Laundry firm management infrastructure. Often information (on forms) is incomplete, or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. This has lead to inconsistencies in various data due to large volume of contrasting customer details leading to mix-up of clothes in the laundry firm

which thus leads to delay in collecting the clothes back.

2.3 DRAWBACKS OF EXISTING SYSTEM

- ❖ No proper online management of system
- ❖ Human effort is needed.
- ❖ It is difficult to maintain important information in books.
- ❖ More manual hours need to generate required reports.

2.4 PROPOSED SYSTEM

The Laundry Management System is designed for any Laundry firm to replace their existing manual, paper based system. The services are to be provided in an efficient, cost effective manner, with the goal of reducing the delay and resources currently required for such tasks as clothes details are bounded to a particular customer . Since the existing system makes use of tedious administrative tasks, lots paper work and time, in which full information cannot be gotten from busy customers.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources, and the system will work in almost all configurations. It has got following features:

❖ **Better security: -**

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

❖ **Ensure data accuracy: -**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

❖ Better service: -

The product will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- ❖ The costs conduct a full system investigation.
- ❖ The cost of the hardware and software.
- ❖ The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, Daily Fresh was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- ❖ Does the existing technology sufficient for the suggested one?
- ❖ Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- ❖ Is there sufficient support for the users?
- ❖ Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	- Intel core i3
RAM	- 4 G
Hard disk	- 1 TB

3.2.2 Software Specification

Front End	- HTML, CSS
Backend	- MYSQL
Client on PC	- Windows 7 and above.
Technologies	- JS, HTML5, BOOTSTARP, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP:HypertextPreprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

❖ **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

❖ **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

❖ **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

❖ **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

❖ **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After

some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- ❖ Class diagram
- ❖ Object diagram
- ❖ Use case diagram
- ❖ Sequence diagram
- ❖ Collaboration diagram
- ❖ Activity diagram
- ❖ Statechart diagram
- ❖ Deployment diagram
- ❖ Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

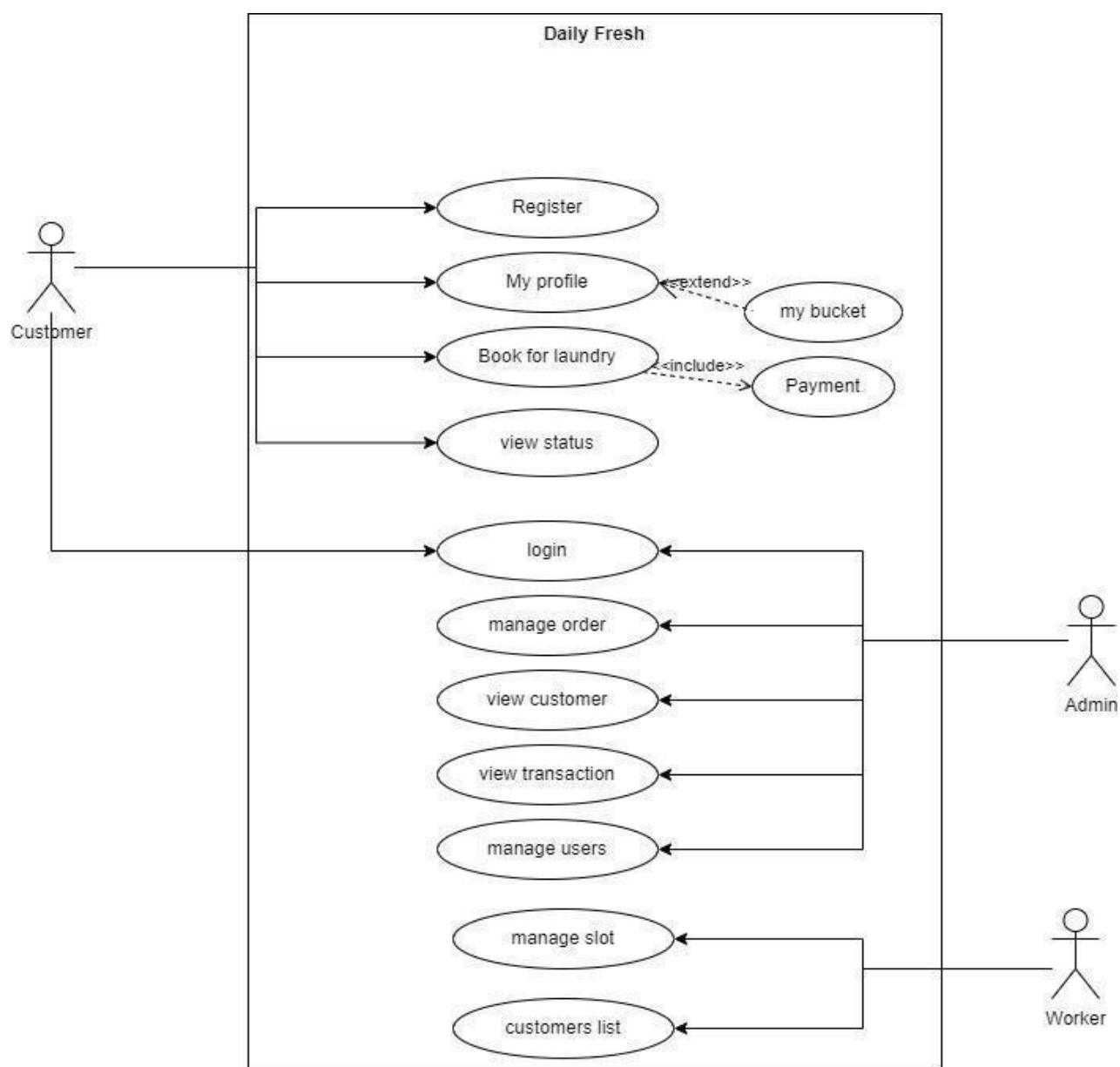
- ❖ The boundary, which defines the system of interest in relation to the world around it.
- ❖ The actors, usually individuals involved with the system defined according to their

roles.

- ❖ The use cases, which are the specific roles are played by the actors within and around the system.
- ❖ The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- ❖ The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- ❖ Give a suitable name for actors.
- ❖ Show relationships and dependencies clearly in the diagram.
- ❖ Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- ❖ Use notes whenever required to clarify some important points.



4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

- i. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.
- ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram
- iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

❖ Synchronous messages

❖ Asynchronous Messages

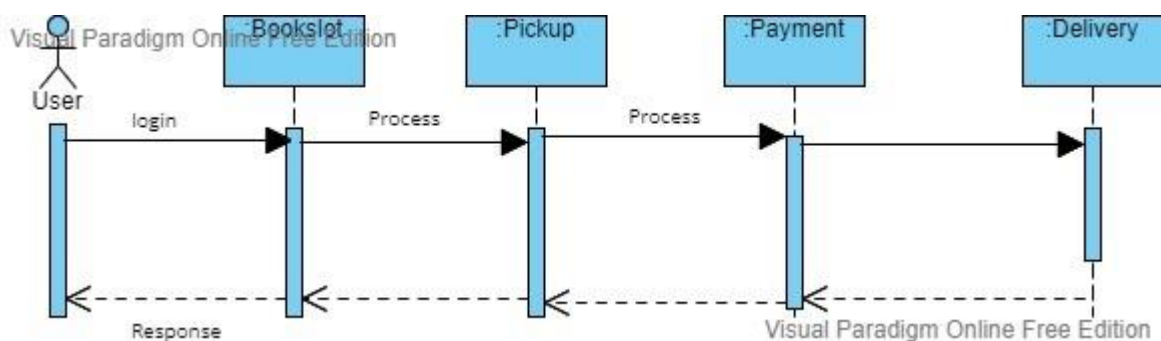
- ❖ Create message
- ❖ Delete Message
- ❖ Self-Message
- ❖ Reply Message
- ❖ Found Message
- ❖ Lost Message

iv. Guards –

To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- ❖ Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- ❖ They are also used to show details of UML use case diagrams.
- ❖ Used to understand the detailed functionality of current or future systems.
- ❖ Visualize how messages and tasks move between objects or components in a system.



4.5 USER INTERFACE DESIGN

4.5.2 INPUT DESIGN

Login



Registration



The registration form is titled "Daily Fresh" in a stylized font. It features a "Register Here" heading. The form includes input fields for Name, Email, Password, Confirm Password, Phn No, and Address. A "Submit" button is located at the bottom of the form. The background of the form is a light blue gradient, and the entire form is set against a backdrop of a white washing machine and a potted plant on a light blue wall.

Daily Fresh

Register Here

Name

Email

Password

Confirm Password

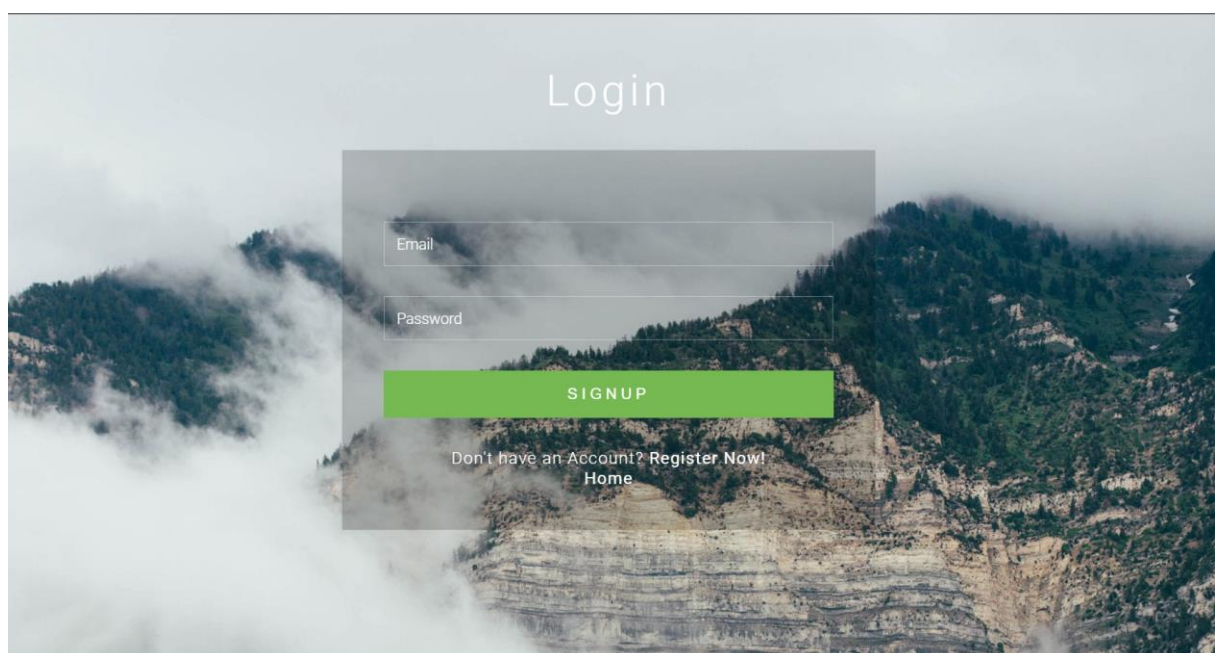
Phn No

Address

Submit

4.5.2 OUTPUT DESIGN

User Login



The login form is titled "Login" and is set against a background of a mountain landscape with clouds. It includes input fields for Email and Password. A green "SIGNUP" button is located below the password field. Below the button, there is a link that says "Don't have an Account? Register Now!" and a "Home" link.

Login

Email

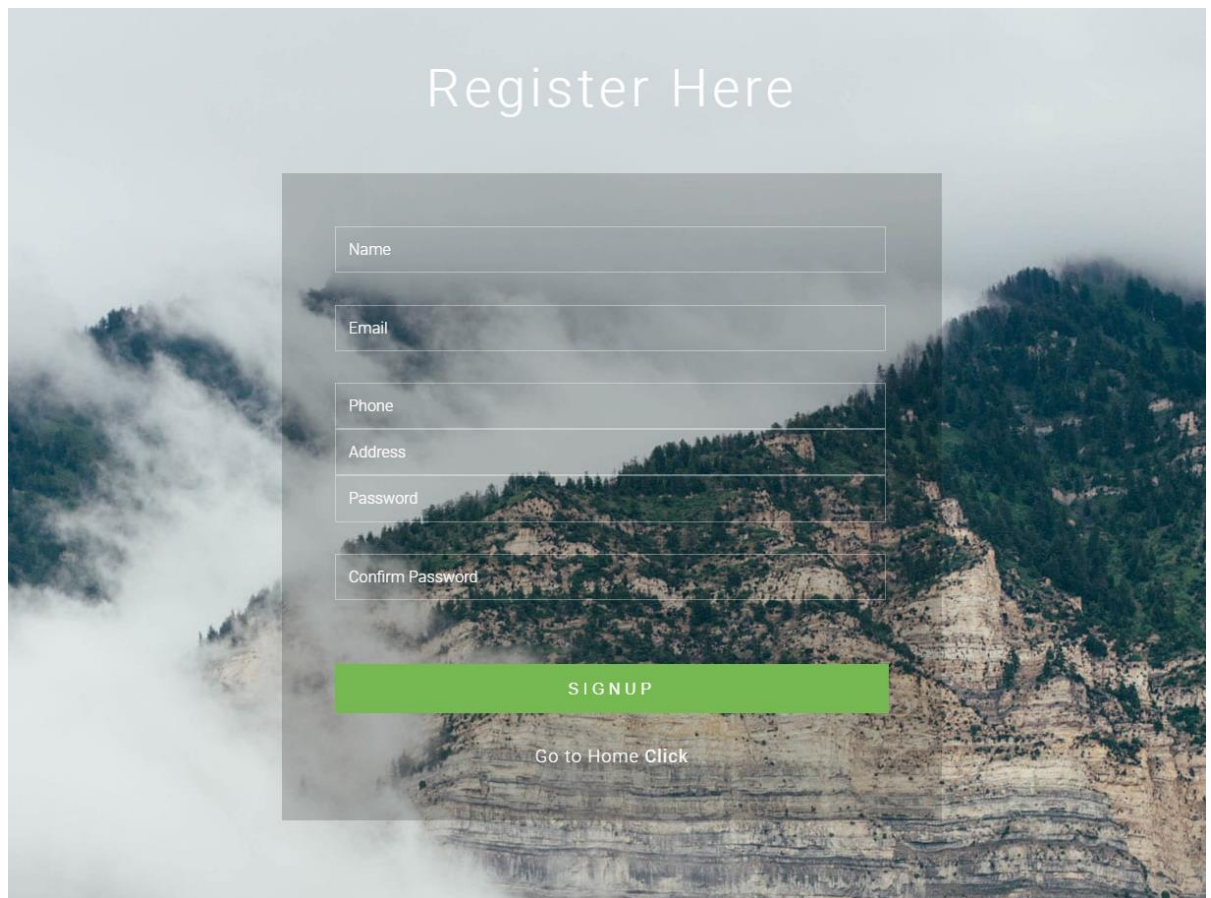
Password

SIGNUP

[Don't have an Account? Register Now!](#)

[Home](#)

User Registration



The image shows a user registration form overlaid on a scenic background of a mountain range with clouds. The form is titled "Register Here" in a large, white, sans-serif font. Below the title, there are six input fields for user information: Name, Email, Phone, Address, Password, and Confirm Password. Each field is a simple white rectangle with a thin border. Below these fields is a prominent green button with the text "SIGNUP" in white, uppercase letters. At the bottom of the form, there is a link that says "Go to Home Click".

Register Here

Name

Email

Phone

Address

Password

Confirm Password

SIGNUP

[Go to Home Click](#)

4.6. DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- ❖ Data Integrity
- ❖ Data independence

4.6.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a table represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

- ❖ Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- ❖ Entity Integrity enforces that no Primary Key can have null values.
- ❖ Referential Integrity enforces that no Primary Key can have null values.
- ❖ Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.6.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ❖ Normalize the data.
- ❖ Choose proper names for the tables and columns.
- ❖ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

TABLE DESIGN

Name:-tbl_Registration

Primary key:-rid

Field	Data type	Description
rid	Int(11)	Primary key
Name	Varchar(20)	Name of user
Email	Varchar(20)	email
Phone no	Int(22)	number
address	Varchar(20)	address

Name:-tbl_login

Primary key:-login_id

Foreign key:-rid refernces tbl_Registration(rid)

Field	Data type	Description
login_id	Int(11)	Primary key
rid	Int(11)	Foreign key
username	Varchar(20)	username
Password	Varchar(20)	Password
Status	Varchar(20)	Status of user

Password	Varchar(20)	Password
Status	Varchar(20)	Status of user

Name:-tbl_clothcatlog

Primary key:-cl_id

Field	Data type	Description
Cl_id	Int(11)	Primary key
Cloth type	Varchar(20)	Type of cloth
price	Varchar(20)	Price of specified type

Name:-tbl_bookslot

Primary key:-bl_id

Foreign key:-Cl_id references tbl_clothcatlog(Cl_id),login_id references tbl_login(login_id)

Field	Data type	Description
bl_id	Int(11)	Primary key
Cl_id	Int(11)	Foreign key
login_id	Int(11)	Foreign key
Date	date	Date of booking
Quantity	Float(50)	Quantity of cloth
Status	Varchar(20)	Booking Status

Name:tbl_pickup

Primary key:pi_id

Foreign key:bl_id references tbl_bookslot(bl_id)

Field	Data type	Description
Pi_id	Int(11)	Primary key
bl_id	Int(11)	Foreign key
Status	Varchar(20)	Pickup Status
Total amount	Float(50)	amount
Date	Date	Date

Name:-tbl_bank

Primary key:-bnk_id

Field	Data type	Description
bnk_id	Int(11)	Primary key
Pay_id	Int(11)	Foreign key
Name	Varchar(20)	Name
Card number	Int(12)	Card number
Cvv	Int(3)	Cvv number
Expiry date	Date	Date
Balance	Float(100)	Balance in account
Status	Varchar(20)	Status from bank

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- ❖ A good test case is one that has high possibility of finding an undiscovered error.
- ❖ A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- ❖ For correctness
- ❖ For implementation efficiency
- ❖ For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- ❖ Input Screen Designs,
- ❖ Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- ❖ Careful planning.
- ❖ Investigation of system and constraints.
- ❖ Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to

ensure that the resistance does not build up, as one has to make sure that:

- ❖ The active user must be aware of the benefits of using the new system.
- ❖ Their confidence in the software is built up.
- ❖ Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

It was a wonderful learning experience for me while working on this project. This project took me through the various phases of project development and gave me real insight into the world of software engineering. The joy of working and the thrill involved while tackling the various problems and challenges gave me a feel of the developers' industry.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- PankajJalote, “*Software engineering: a precise approach*”, 2006.
- Ian Sommerville. [*Software Engineering \(Seventh Edition\)*](#). Addison-Wesley, 2004.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- <https://colorlib.com>

CHAPTER 9

APPENDIX

9.1 Code

Login.html

```
<!DOCTYPE html>
<html>
<head>
<title>Login</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script type="application/x-javascript"> addEventListener("load", function() {
setTimeout(hideURLbar, 0); }, false); function hideURLbar(){ window.scrollTo(0,1); } </script>
<!-- Custom Theme files -->
<link href="css/style.css" rel="stylesheet" type="text/css" media="all" />
<!-- //Custom Theme files -->
<!-- web font -->
<link href="//fonts.googleapis.com/css?family=Roboto:300,300i,400,400i,700,700i" rel="stylesheet">
<!-- //web font -->
</head>
<body>
  <!-- main -->
  <div class="main-w3layouts wrapper">
    <h1>Login</h1>
    <div class="main-agileinfo">
      <div class="agileits-top">
        <form action="codes/login_action.php" method="post">
          <input class="text email" type="email" name="email"
placeholder="Email" required="">
          <input class="text" type="password" name="password"
placeholder="Password" required>
          <div class="wthree-text">
            <div class="clear"></div>
          </div>
          <input type="submit" value="SIGNUP"><!-- <a href="ui-
elements.html">Home</a> -->
        </form>
        <p>Don't have an Account? <a href="register.html">Register
Now!</a></p>
        <p><a href="index.html">Home</a></p>
      </div>
    </div>
  </div>
</body>
</html>
```

Register.html

```

<!DOCTYPE html>
<html>

<head>
  <title>Register</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <script
    type="application/x-javascript"> addEventListener("load", function() { setTimeout(hideURLbar,
0); }, false); function hideURLbar(){ window.scrollTo(0,1); } </script>
  <!-- Custom Theme files -->
  <link href="css/style.css" rel="stylesheet" type="text/css" media="all" />
  <!-- //Custom Theme files -->
  <!-- web font -->
  <link href="//fonts.googleapis.com/css?family=Roboto:300,300i,400,400i,700,700i"
rel="stylesheet">
  <!-- //web font -->
  <script>
    function registration() {

      var name = document.getElementById("name").value;
      var email = document.getElementById("email").value;
      var number = document.getElementById("phone").value;
      var address = document.getElementById("address").value;
      var password = document.getElementById("password").value;
      var cpassword = document.getElementById("cpassword").value;

      //email id expression code
      var pwd_expression = /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*-.])/;
      var letters = /^[A-Za-z]+$;/
      var filter = /^[a-zA-Z0-9_\. -]+\@(((a-zA-Z0-9\ -)+\.)+([a-zA-Z0-9]{2,4})+)$/;
      var phoneno = /^\d{10}$/;

      if (name == "") {
        alert('Please enter your name');
      }
      else if (!letters.test(name)) {
        alert('Name field required only alphabet characters');
      }

      else if (email == "") {
        alert('Please enter your user email id');
      }
      else if (!filter.test(email)) {
        alert('Enter the e-mail format correctly');
      }
      else if (number == "") {
        alert('Please enter your number');
      }
      else if (!phoneno.test(number)) {
        alert('Enter number correctly');
      }
    }
  </script>

```

```

        else if (password == "") {
            alert('Please enter Password');
        }
        else if (address == "") {
            alert('Please enter Address');
        }
        else if (cpassword == "") {
            alert('Enter Confirm Password');
        }
        else if (!pwd_expression.test(password)) {
            alert('Upper case, Lower case, Special character and Numeric letter are required in Password
filed');
        }
        else if (password != cpassword) {
            alert('Password not Matched');
        }
        else if (document.getElementById("password").value.length < 6) {
            alert('Password minimum length is 6');
        }
        else if (document.getElementById("cpassword").value.length > 12) {
            alert('Password max length is 12');
        }
        else {
            alert('Successfully Registered');
        }
    }
}
</script>
</head>

<body>
    <!-- main -->
    <div class="main-w3layouts wrapper">
        <h1>Register Here</h1>
        <div class="main-agileinfo">
            <div class="agileits-top">
                <form action="users/hm/registration.php" method="POST">
                    <input class="text" type="text" id="name" name="name" placeholder="Name"
required="">
                    <input class="text email" type="email" id="email" name="email" placeholder="Email"
required="">
                    <input class="text" type="text" name="phone" id="phone" placeholder="Phone"
required="">
                    <input class="text" type="text" name="address" id="address" placeholder="Address"
required="">
                    <input class="text" type="password" name="password" id="password"
placeholder="Password"
required="">
                    <input class="text w3lpass" type="password" name="cpassword" id="cpassword"
placeholder="Confirm Password" required="">
                    <div class="wthree-text">

                        <div class="clear"> </div>
                    </div>
                </form>
            </div>
        </div>
    </div>

```

```

        <input type="submit" onclick="registration()" name="submit" value="SIGNUP">
    </form>
    <p>Go to Home <a href="index.html">Click</a></p>
    <p>Login <a href="login.html">Click</a></p>

</div>
</div>

</div>
<!-- //main -->
</body>

</html>

```

Index.html

```

<?php
include "config.php";
if(isset($_SESSION['login_id']))
{
?>
<!DOCTYPE html>
<html lang="en">

<head>

    <!-- Basic Page Needs
    ===== -->
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <title>Daily Fresh</title>

    <meta name="description" content="">
    <meta name="author" content="">
    <meta name="keywords" content="">

    <!-- Mobile Specific Metas
    ===== -->
    <meta name="viewport" content="width=device-width, minimum-scale=1.0, maximum-scale=1.0">
    <meta name="apple-mobile-web-app-capable" content="yes">

    <!-- Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Montserrat:300,300i,400,600,700" rel="stylesheet">
    <link rel="stylesheet" href="http://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.4.0/css/font-
awesome.min.css">

    <!-- Favicon
    ===== -->
    <link rel="apple-touch-icon" sizes="180x180" href="hm/img/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="16x16" href="hm/img/favicon.png">

```

```

<!-- Stylesheets
===== -->
<!-- Bootstrap core CSS -->
<link href="hm/css/bootstrap.min.css" rel="stylesheet">

<!-- Custom styles for this template -->
<link href="hm/css/style.css" rel="stylesheet">
<link href="hm/css/responsive.css" rel="stylesheet">

</head>

<body>

<header id="masthead" class="site-header">
  <nav id="primary-navigation" class="site-navigation">
    <div class="container">

      <div class="navbar-header">

        <a class="site-title" style="font-size: 30px"><span>Daily</span>Fresh</a>

      </div><!-- /.navbar-header -->

      <div class="collapse navbar-collapse" id="agency-navbar-collapse">

        <ul class="nav navbar-nav navbar-right">

          <li class="active"><a href="index.html" data-toggle="dropdown">Home</a></li>
          <li class="dropdown"><a href="#" class="dropdown-toggle" data-
toggle="dropdown">Services<i
          class="fa fa-caret-down hidden-xs" aria-hidden="true"></i></a>

          <ul class="dropdown-menu" role="menu" aria-labelledby="menu1">
            <li><a href="hm/portfolio.html">Laundry Washing</a></li>
            <li><a href="hm/portfolio.html">Ironing</a></li>
            <li><a href="hm/portfolio.html">Dry Cleaning</a></li>
            <li><a href="hm/portfolio.html">Coloring</a></li>
          </ul>

          </li>
          <li><a href="hm/portfolio.html">Images</a></li>
          <li><a href="hm/contact.html">Contact</a></li>
          <li><a href="login.html">Login</a></li>
          <li><a href="register.html">Register Here</a></li>

        </ul>

      </div>

    </div>

  </nav><!-- /.site-navigation -->
</header><!-- #masthead -->

<div id="hero" class="hero overlay">

```

```

<div class="hero-content">
  <div class="hero-text">
    <h1>Make Your Day Fresh.</h1>
    <p>Make Youreself Feel Comfortable.</p>
    <!-- <button> <a href="contact.html" class="btn btn-border">Learn more</a></button>-->
    <button class="btn btn-border" type="submit"><a href="contact.html"
      style="text-decoration-color: white">Learn More</a> </button>
  </div><!-- /.hero-text -->
</div><!-- /.hero-content -->
</div><!-- /.hero -->

<main id="main" class="site-main">

  <section class="site-section section-features">
    <div class="container">
      <div class="row">
        <div class="col-sm-5">
          <h2>Laundry Service</h2>
          <p>We use state-of-the-art front loading washing machines and premium detergents for wash
fold
          and laundry services to give your clothes that special care it needs.</p>
        </div>
        <div class="col-sm-7 hidden-xs">
          
        </div>
      </div>
    </div>
  </section><!-- /.section-features -->

  <section class="site-section section-services gray-bg text-center">
    <div class="container">
      <h2 class="heading-separator">How It Works: in 4 Easy Steps</h2>
      <p class="subheading-text">See our service</p>
      <div class="row">
        <div class="col-md-3 col-xs-6">
          <div class="service">
            
            <h3 class="service-title">Step 1</h3>
            <p class="service-info">Bag up all your dirty clothes.</p>
          </div><!-- /.service -->
        </div>
        <div class="col-md-3 col-xs-6">
          <div class="service">
            
            <h3 class="service-title">Step 2</h3>
            <p class="service-info">We pick up your clothes. </p>
          </div><!-- /.service -->
        </div>
        <div class="col-md-3 col-xs-6">
          <div class="service">
            
            <h3 class="service-title">Step 3</h3>
            <p class="service-info">We clean your clothes.</p>
          </div><!-- /.service -->
        </div>
      </div>
    </div>
  </section>

```

```

    <div class="col-md-3 col-xs-6">
      <div class="service">
        
        <h3 class="service-title">Step 4</h3>
        <p class="service-info">We deliver clean, folded clothes.</p>
      </div><!-- /.service -->
    </div>

  </div>
</div>
</section><!-- /.section-services -->

<section class="site-section section-map-feature text-center">

  <div class="container">
    <h2>We Make yourself feel confident</h2>
    <p>Freshness Throughout The Day</p>
    <a href="#" class="btn btn-fill">Learn More</a>
    <div class="row">
      <div class="col-sm-3 col-xs-6">
        <div class="counter-item">
          <p class="counter" data-to="377" data-speed="2000">0</p>
          <h3>Review</h3>
        </div><!-- /.counter-item -->
      </div>
      <div class="col-sm-3 col-xs-6">
        <div class="counter-item">
          <p class="counter" data-to="1204" data-speed="2000">0</p>
          <h3>Works completed</h3>
        </div> <!-- /.counter-item -->
      </div>
      <div class="col-sm-3 col-xs-6">
        <div class="counter-item">
          <p class="counter" data-to="459" data-speed="1000">0</p>
          <h3>Happy Clients</h3>
        </div><!-- /.counter-item -->
      </div>
      <div class="col-sm-3 col-xs-6">
        <div class="counter-item">
          <p class="counter" data-to="675" data-speed="1000">0</p>
          <h3>Average Deal</h3>
        </div><!-- /.counter-item -->
      </div>
    </div>
  </div>
</div>

</section><!-- /.section-map-feature -->

<section class="site-section section-portfolio">
  <div class="container">
    <div class="text-center">
      <h2 class="heading-separator">Our Work</h2>
      <p class="subheading-text">some of our work</p>
    </div>
    <div class="row">

```



```

<div class="col-lg-3 col-md-4 col-xs-6">
  <div class="portfolio-item">
    
    <!--<h4 class="portfolio-item-title">Portfolio item</h4-->
    <a><i class="fa fa-arrow-right" aria-hidden="true"></i></a>
  </div><!-- /.portfolio-item -->
</div>
<div class="col-lg-3 col-md-4 col-xs-6">
  <div class="portfolio-item">
    
    <!--<h4 class="portfolio-item-title">Portfolio item</h4-->
    <a><i class="fa fa-arrow-right" aria-hidden="true"></i></a>
  </div><!-- /.portfolio-item -->
</div>
<div class="col-lg-3 col-md-4 col-xs-6">
  <div class="portfolio-item">
    
    <!--<h4 class="portfolio-item-title">Portfolio item</h4-->
    <a><i class="fa fa-arrow-right" aria-hidden="true"></i></a>
  </div><!-- /.portfolio-item -->
</div>
<div class="col-lg-3 col-md-4 col-xs-6">
  <div class="portfolio-item">
    
    <!--<h4 class="portfolio-item-title">Portfolio item</h4-->
    <a><i class="fa fa-arrow-right" aria-hidden="true"></i></a>
  </div><!-- /.portfolio-item -->
</div>
<div class="col-lg-3 col-md-4 col-xs-6">
  <div class="portfolio-item">
    
    <!--<h4 class="portfolio-item-title">Portfolio item</h4-->
    <a><i class="fa fa-arrow-right" aria-hidden="true"></i></a>
  </div><!-- /.portfolio-item -->
</div>
<div class="col-lg-3 col-md-4 col-xs-6">
  <div class="portfolio-item">
    
    <!--<h4 class="portfolio-item-title">Portfolio item</h4-->
    <a><i class="fa fa-arrow-right" aria-hidden="true"></i></a>
  </div><!-- /.portfolio-item -->
</div>
<div class="col-lg-3 col-md-4 col-xs-6">
  <div class="portfolio-item">
    
    <!--<h4 class="portfolio-item-title">Portfolio item</h4-->
    <a><i class="fa fa-arrow-right" aria-hidden="true"></i></a>
  </div><!-- /.portfolio-item -->
</div>
<div class="col-lg-3 col-md-4 col-xs-6">
  <div class="portfolio-item">
    
    <!--<h4 class="portfolio-item-title">Portfolio item</h4-->
    <a><i class="fa fa-arrow-right" aria-hidden="true"></i></a>
  </div><!-- /.portfolio-item -->

```

```

        </div>
    </div>
</div>
</section><!-- /.section-portfolio -->

<section class="site-section section-newsletter text-center">
    <div class="container">
        <h2>Subscribe for Enquires</h2>
        <form class="form-group newsletter-group">
            <input type="email" class="form-control" placeholder="Your e-mail" required>
            <button class="btn btn-green" type="button">Subscribe</button>
        </form><!-- /.newsletter-group -->
    </div>
</section><!-- /.section-newsletter -->

</main><!-- /#main -->

<footer id="colophon" class="site-footer">
    <div class="container">
        <div class="row">
            <div class="col-md-3 col-sm-4 col-xs-6">
                <a class="site-title"><span>WHY WE ARE </span>THE BEST ?</a>
                <p>We are striving continuously to provide you with laundry and dry-cleaning services for
garments,
                shoes and bags that are at unique and one of its own kind</p>
            </div>
            <div class="col-md-3 col-sm-4 col-xs-6">
                <a class="site-title"><span>Free </span>Pickup Drop</a>
                <p>Free pick-up and delivery at your doorsteps so that you can enjoy hassle-free dry cleaning
and
                laundry services.</p>
                <br>
                <a class="site-title"><span>Fast </span>Processing</a>
                <p>We ensure that you get your articles processed within committed time. Most of our
processes have
                been automated to ensure same experience every time</p>
            </div>
            <div class="col-md-3 col-sm-4 col-xs-6">
                <a class="site-title"><span>CUSTOM PACKAGING </span>PACKAGING</a>
                <p>Each article is returned to you individually packaged to protect against dust, light and
mildew.</p>
                <br>
                <br>
                <a class="site-title"><span>AFFORDABLE </span></a>
                <p>Daily Fresh is a quick, efficient and cost effective way for your dry cleaning needs. Online
laundry space was never that affordable.</p>
            </div>
        </div>
    </div>
</div>
<div class="copyright">
    <div class="container">
        <div class="row">

```

```
<div class="col-xs-8">

</div>
<div class="col-xs-4">

</div>
</div>
</div><!-- /.copyright -->

</footer><!-- /#footer -->

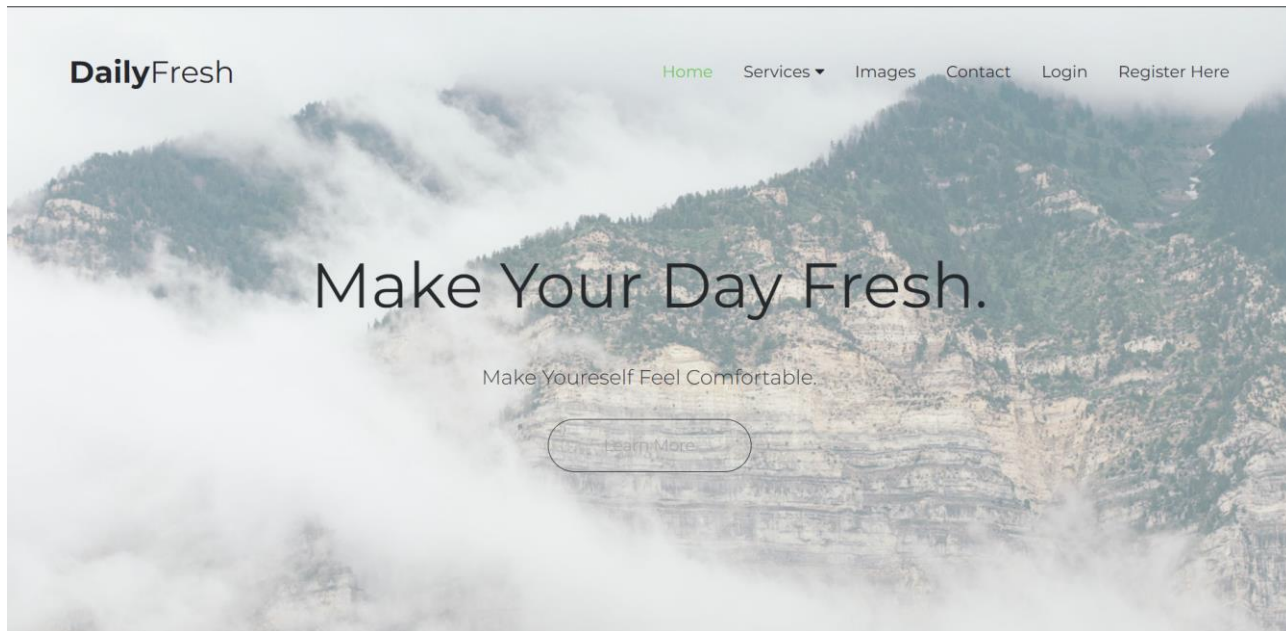
<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script src="hm/js/bootstrap.min.js"></script>
<script src="hm/js/bootstrap-select.min.js"></script>
<script src="hm/js/jquery.slicknav.min.js"></script>
<script src="hm/js/jquery.countTo.min.js"></script>
<script src="hm/js/jquery.shuffle.min.js"></script>
<script src="hm/js/script.js"></script>

</body>

</html>
<?php
}
?>
```

9.2 Screen Shots

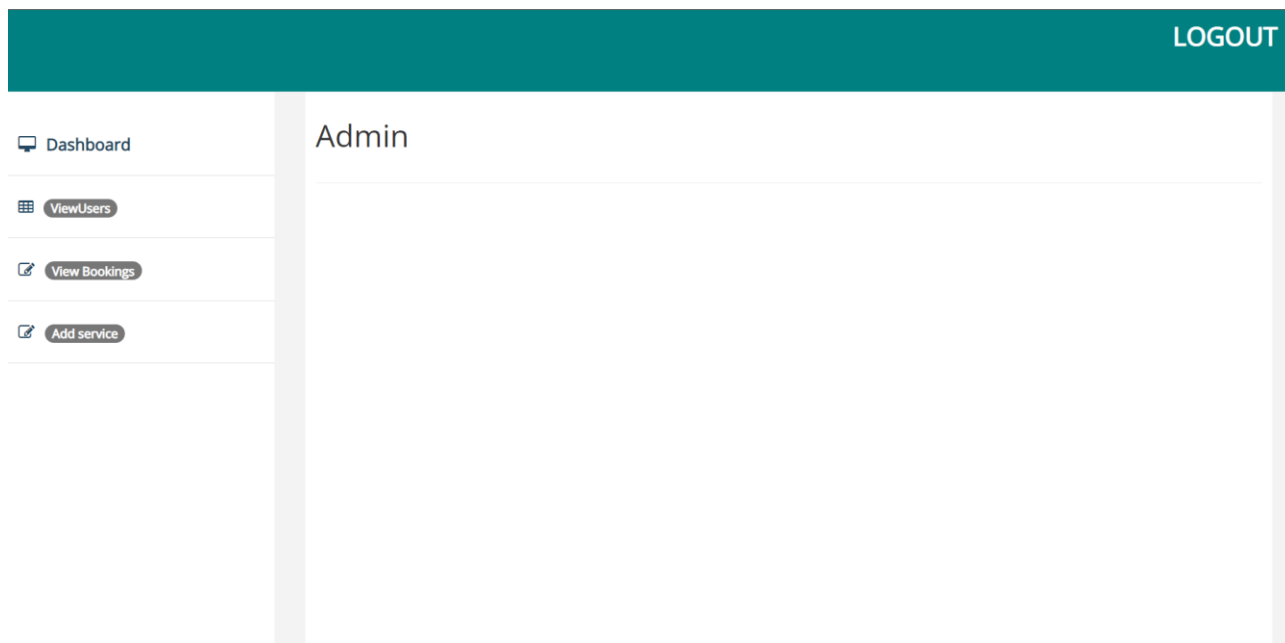
Home page



Laundry Booking page

The screenshot displays a web application interface for laundry booking. At the top, a dark teal header bar contains a 'LOGOUT' link on the right. On the left side, there is a vertical sidebar with three menu items: 'Dashboard' (with a monitor icon), 'Bookslot' (with a calendar icon and a dark grey background), and 'Status' (with a checkmark icon). The main content area on the right begins with a welcome message 'Welcome tebin'. Below this, the section is titled 'Book Your Laundry'. It contains a form with two input fields: 'LaundryType' with a dropdown menu currently showing 'Select type', and 'Book Date' with a text input field showing the placeholder 'dd-mm-yyyy' and a calendar icon on the right. A blue 'Add' button is located below these fields. The entire interface is set against a light grey background.

Admin page



Registered users page

