PROJECT REPORT

ON

DIGITAL BUSINESS CARD

FOR

CODEBROTHER INDIA

BY

MISS PRITI SANTOSH MAURYA
(MC21954)

POST GRADUATE DEPARTMENT OF COMPUTER SCIENCE,
S.N.D.T. WOMEN'S UNIVERSITY

MASTER OF COMPUTER APPLICATION
SEM VI - (2020 – 2021)

CERTIFICATE

This is to certify that MISS.PRITI SANTOSH MAURYA has completed the project on DIGITAL BUSINESS CARD satisfactorily as a partial fulfillment of the Post Graduate Degree of Master of Computer Application (MCA)

Internal Examiner :	**External Examiner :**
Signature :	Signature :
Name : PROF.SHAESTA KHAN	Name :
Date :24/08/2021	Date :

Head of Department :

Signature :

Name :DR.GANESH MAGAR

Date :24/08/2021

CODEBROTHER INDIA

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Miss Priti Maurya** has done her online internship in developing basic web applications using PHP and Bootstrap at Codebrother India, Kanpur, from 28th Jan 2021 to 28th Jun 2021.

She has worked on the projects titled “Digital Business Card”& “Online Voting Portal”. As part of the projects, she has designed layouts, developed web apps.

During her internship, she has demonstrated her skills with self-motivation to learn new skills. Her performance exceeded our expectations and she was able to complete the project on time.

We wish her all the best for her upcoming career.

Akash Maurya

Project Manager

+91 6394818365
office @codebrother.co.in
141, Malviya Vihar, Karrahi
Kanpur - 208027
www.codebrother.co.in

Acknowledgement

I would like to express my sincere gratitude to my mentor Prof.Shaesta Khan for her valuable guidance and support in completing my project.

I would also like to express my gratitude towards her for allowing me to do a project on Digital Business Card. This project helped me learn many new things. Without the support and suggestions from my mentor, this project would not have been completed.

I take this opportunity to thank SNDT Woman`s University for giving me chance to do this project.

Also, I would like to express my deepest gratitude and special thanks to Mr. Gaurav Maurya, Managing Director, who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my project at their esteemed organization and extending during the training.

I express my deepest thanks to Mr. Akash Maurya , CEO for taking part in useful decision & giving necessary advices and guidance and arranged all facilities to make life easier. I choose this moment to acknowledge his contribution gratefully.

I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future,

Name and signature:
Priti Santosh Maurya

Abstract

The **Digital Business Card** is a software that handles the sales and purchase of various Cards. It includes everything from initial contact with the customer to final settlement of their bill. The system stores the details of a variety of Visiting Card and Business Card. And also provides billing and data processing are handled with accuracy. The project **'Digital Business card '** is divided into four modules to make the development and maintenance easy. Each of the modules undertakes distinguishable tasks.

The Main modules are Users (User/Admin) Module, My order Module, Order module, Payment/Report module. The Users module stores the details of Business card, and customers who want to purchase those Card. This module will increase the dynamism of the administration system. The change that occurs in the general administration reflects in the overall functionality of the project. The administrator is responsible to insert Designing regarding new trends that are emerging in the Business Card . It is also the duty of the administrator to add, remove and update the various information's of the application.

The Order module user can order variety of Cards. The user can select any number of Card Designing at any time. And if he is satisfied with that he can order the Card. Otherwise he can go for a better search of Visiting Card and can order the Cards as he wishes. The user must have to pay the amount by using cash on delivery or Online Payment.

The Search module includes the comprehensive searching for Cards. The user can search for Card of particular Cards. They are by Price, by Card name.The report module is handled by the administrator. This module generates all reports of different types of payments made by the user and generates the reports of all the services.

The project has been developed in **PHP** as front end and My**SQL** as back end which develop to help powerful software.

List of Figures

Figure	Page No.
Figure 1: Agile Model	28
Figure 2: Use Case Diagram	36
Figure 3.1: Activity Diagram	41
Figure 3.2 : Activity Diagram	42
Figure 4: Sequence Diagram	45
Figure 5: Class Diagram	51
Figure 6: Object Diagram	53
Figure 7: Entity Relationship Diagram	57
Figure 8: Web Site Map Diagram	59

TABLE OF CONTENTS

Contents	Page No.
Chapter 1: INTRODUCTION 1.1 Company Profile 1.2 Existing System and Need for System 1.3 Scope of work 1.4 Operating Environment – Hardware and Software 1.5 Detail Description of Technology Used	8 8 13 14 15 16
CHAPTER 2: PROPOSED SYSTEM 2.1 Proposed System 2.2 Objectives of System 2.3 User Requirements 2.4 Software Development Model (Lifecycle Model)	28 28 28 29 30
CHAPTER 3: ANALYSIS & DESIGN 3.1 Use Case Diagrams 3.2 Activity Diagram 3.3 Sequence Diagram 3.4 Class Diagram 3.5 Object Diagram 3.6 Entity Relationship Diagram 3.7 Web Site Map Diagram 3.8 Schema Diagram 3.9 User Interface Design 3.10 Test Plan and Test Cases	35 38 44 47 53 55 58 59 61 62 64
Drawbacks and Limitations	77
Proposed Enhancements	78
Conclusions	79
Bibliography	80

CHAPTER 1: INTRODUCTION

1.1Company Profile

BACKGROUND

Internships are formal programs designed to provide practical experience in real world environment to students who are new to field. Internships help to build competent resume by giving students visible work experience Although colleges and universities assists students in finding the right Internship programs, it is the responsibility of the would-be interns to carefully examine internship programs, and see if those programs actually offer the training they need.

There is no standardized duration for how long an Internship program lasts but typically an Internship is the period of six months. During this period interns have the opportunity to explore their field of interest, find out what future they are diving and whether or not their skill sets match their path of career.

Internship program provides a distinct platform to interns to build a solid foundation to their career, understand the workplace culture, gain vital work related experience, develop relevant skills, and also presents the opportunity to find permanent employment in the host company.

Some of the major benefits of Internship program to students include:

- Students can learn about the workplace culture and gain much needed work experience.
- Student can finally experience practical implementation of things they have been learning in the classroom
- Students can build a strong foundation that will bolster their career
- Good performance in the internship program can secure permanent employment with the host company
- Some internship programs are paid programs meaning students can earn money while they get necessary exposure.

PURPOSE OF INTERNSHIP

Internship program provides a new perspective to the students. It provides learning opportunities outside the boundaries of the course curriculum and classroom activities. These opportunities enhance student's capability to apply theoretical understanding into real-world scenarios, thereby enhancing the students' academic and career goals .

From the student perspective, internship assists with career development in different diverse fields such as coding, programming, networking, marketing, etc. by providing vital work experience that allows students to explore their area of interests and develop relevant skills and competencies.

From the organization perspective, internship provides a unique opportunity to train fresh talents, enhance their professional development as well as aid the professional growth of the intern mentors/supervisor. Organization can also find potential employees within the interns.

OBJECTIVE OF INTERNSHIP

The major objectives of internships are:

- To expose students to a particular job and a profession or industry
- To provide students with opportunity to develop skills in the field of interest
- To assist students in gaining vital work related experience and building strong resume for bright career
- To help students in developing business contacts i.e. creating network contacts
- To help students potentially land permanent or contractual jobs from host company

ANALYSIS OF ACTIVITY DONE

ORGANIZATION SELECTION

It is very important to select organization that fulfills our objectives. As it is for our internship, it is necessary to select organization where there is learning environment because as an intern our first motive is to learn how the works are carried out in the real field.

The Master computer Application of allows us to attain knowledge on various aspects of Information Technology. At the same time the internship is the one of the major highlight of the program to expose the students to the professional world. Among the various criteria and sectors provided to us in internship prerequisite statement, software Development Company was chosen. Various organizations were shortlisted and approached out of which the organization with the best lucrative offer and environment was selected.

Codebrother India. which is located Kanpur an IT company that covers software development, wired & wireless solutions, network design and implementation, e-Business applications solutions and others internet related systems and author got selected there. The company helped me gain wide experience by getting me involved in their projects.

DURATION

Start Date: JAN 28, 2021

Duration: 6 month

Position: Associate

Supervisors: MR.AKASH MAURYA

Office Hour: 10:00 am – 5:30 pm

End Date: Jun 27, 2021

ROLES AND RESPONSIBILITY

During the internship period knowledge of various platforms and programming languages are gained. The target was to deliver a fully-fledged web based system by using different tools and frameworks such as PHP (Laravel) and JS.

Hence, to meet that objective, this internship required the extensive preliminary studies about the core PHP before actually analysing the actual requirement of the system. The study was required not only to understand the subject under study but also to realize the solutions to the existing problems and implementing the findings from the study was another bigger challenge. Besides study of, Core PHP other major activities carried out during internship was extensive study of current online platform, presentations of study analysis and practical implementations, and most importantly the team discussions to analyse the customer change request. The regular meetings with the Supervisor and discussion with mentors helped me to wide my horizon of knowledge of the existing system and problem background. Software development is one of the major services of the Soft web Developers Company. Soft web Developers provides a flexible and scalable solutions of software (web based or desktop based) to meet our needs for small or large firm.

Software development is the collective processes involved in creating software programs, embodying all the stages throughout the systems development life cycle (SDLC). During my internship, Author worked on Visiting card project which was based on PHP (Laravel) and JavaScript frame work.

TEAM

Author has been assigned with my senior programmer Akash for the project. He gave me introductions so that author could understand the project in short time and also assigned me task and for testing part the problem to be fixed. Without his helpful mentoring, author could not involve with the project successfully.

1.2 Existing System and Need for System

The old manual system was suffering from a series of drawbacks. Since whole of system was to be maintained with hands the process of keeping, maintaining and retrieving the information was very tedious and lengthy. The records were never used to be in a systematic order. A lot of difficulties were faced when associating a transaction with a particular context. To find information a lot of registers had to be checked .There would always be unnecessary consumption of time while entering and retrieving records .One more problem was to find error while entering record. Once a record was entered it was difficult to update these records.

For these reasons the need for an online visiting card system raised. Most of the processes are now partially automated. And it saves a lot of time while making the process easier.

In the existing system everything is to be done manually but in the proposed system we have to computerize all the information using this application

- Lack of security of data
- More man power
- Time consuming
- Needs manual calculations.
- More manual work

1.3 Scope of Work

The scope of the project is to create a Business card website. The reason behind to create a web based application easily customizable, accessible anywhere, accessible for a range devices, increased security etc.

The aim is to automate its existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data information can be stored for a longer period with easy accessing and manipulation of the same.

It may help collecting perfect management in details.in a very short time the collection will be simple and sensible it will help a person to know management of passed year vividly. It also helps in current works relative to online visiting card it will reduce the cost of collecting management and collection procedure will go on smoothly.

The user has to fill various forms and the system generates types of information that can be used for various purposes

1.4 Operating Environment – Hardware and Software

HARDWARE REQUIREMENTS

Intel core i3 7^{Th} generation is used as a processor because it is fast than other processors an provide reliable and stable and we can run our pc for long time. By using this processor we can keep on developing our project without any worries.

Ram 1gb is used as it will provide fast reading and writing capabilities and will in turn support in processing

SOFTWARE REQUIREMENTS

Operating system- Windows 10 is used as the operating system as it is stable and supports more features and is more user friendly

Database MYSQL-MYSQL is used as database as it easy to maintain and retrieve records by simple queries which are in English language which are easy to understand and easy to write.

Development tools and Programming language- HTML is used to write the whole code and develop webpages with css, java script for styling work and php for sever side scripting.

1.5 Detail Description of Technology Used

PHP

Php is a server side scripting language. That is used to develop static websites or dynamic websites or web applications. Php stands for hypertext pre-processor, that earlier stood for personal home pages.

Php scripts can only be interpreted on a server that has php installed.

The client computers accessing the php scripts require a web browser only.

A php file contains php tags and ends with the extension ".php".

A PHP file can also contain tags such as HTML and client side scripts such as JavaScript.

- **HTML is an added advantage** when learning PHP Language. You can even learn PHP without knowing HTML but it's recommended you at least know the basics of HTML.
- **Database management systems** DBMS for database powered applications.
- For more advanced topics such as interactive applications and web services, you will need **JavaScript and XML**.

- PHP is **open source and free.**
- Short learning curve compared to other languages such as JSP, ASP etc.
- Large community document
- Most web hosting servers support PHP by default unlike other languages such as ASP that need IIS. This makes PHP a cost effective choice.
- PHP is regular updated to keep abreast with the latest technology trends.
- Other benefit that you get with PHP is that it's a **server side scripting language**; this means you only need to install it on the server and client computers requesting for resources from the server do not need to have PHP installed; only a web browser would be enough.
- PHP has **in built support for working hand in hand with MySQL**; this doesn't mean you can't use PHP with other database management systems. You can still use PHP with
    - Postgres
    - Oracle

    - MS SQL Server
    - ODBC etc.
- PHP is **cross platform;** this means you can deploy your application on a number of different operating systems such as windows, Linux, Mac OS etc.

File extension and Tags In order for the **server** to **identify** our **PHP files** and **scripts**, we must **save** the **file** with the **".php" extension**. Older PHP file extensions include

- .phtml
- .php3
- .php4
- .php5
- .phps

PHP was designed to work with HTML, and as such, it can be embedded into the HTML code

PHP statements end with a semi colon (;). If you only have one statement, you can omit the semi colon. If you have more than one statement, then you must end each line with a semi colon. For the sake of consistency, it is recommended that you always end your statement(s) with a semi colon. PHP scripts are executed on the server. The output is returned in form of HTML

Laravel php

Laravel is an open-source PHP framework, which is robust and easy to understand. It follows a model-view-controller design pattern. Laravel reuses the existing components of different frameworks which helps in creating a web application. The web application thus designed is more structured and pragmatic.

Laravel offers a rich set of functionalities which incorporates the basic features of PHP frameworks like CodeIgniter, Yii and other programming languages like Ruby on Rails. Laravel has a very rich set of features which will boost the speed of web development.

If you are familiar with Core PHP and Advanced PHP, Laravel will make your task easier. It saves a lot time if you are planning to develop a website from scratch. Moreover, a website built in Laravel is secure and prevents several web attacks.

XAMPP Server

XAMPP is an open source cross platform web server, MySQL database engine, and PHP and Perl package. It is compiled and maintained by apache. The acronym XAMPP stands for;

- X – [cross platform operating systems] meaning it can run on any OS Mac OX , Windows , Linux etc.
- A – Apache - this is the web server software.
- M – MySQL - Database.
- P – PHP
- P – Perl – scripting language

In order to use PHP, you will need to install PHP, Apache and may be even MySQL. It's not easy to install Apache and configure it. If you install Apache on its own, you will still have to set it up and integrate it with PHP and Perl among other things. XAMPP deals with all the complexity in setting up and integrating with PHP and Perl. Unlike Java that runs with the Java SDK only, PHP requires a web server to work.

XAMPP provides an easy to use control panel to manage Apache, MySQL and other programs such as Tomcat, FileZilla etc. You don't have to memorize commands for starting apache, MySQL etc.

JavaScript

JavaScript is an open source & most popular client side scripting language supported by all browsers. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development. **JavaScript and Java are very much unrelated. Java is a very complex programming language whereas JavaScript is only a scripting language**.

The syntax of JavaScript is mostly influenced by the programming language C.

Being a scripting language, **JavaScript cannot run on its own. In fact, the browser is responsible for running JavaScript code**. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it is up to the browser to execute it. The main advantage of JavaScript is that **all modern web browsers support** JavaScript. So, you do not have to worry about whether your site visitor uses Internet Explorer, Google Chrome,

Firefox or any other browser. JavaScript will be supported. Also, JavaScript **runs on any operating system** including Windows, Linux or Mac. Thus, JavaScript overcomes the main disadvantages of VBScript (Now deprecated) which is limited to just IE and Windows's

To start with, you need a text editor to write your code and a browser to display the web pages you develop. You can use a text editor of your choice including Notepad++, Visual Studio Code, Sublime Text, Atom or any other text editor you are comfortable with. You can use any web browser including Google Chrome, Firefox, Microsoft Edge, Internet Explorer etc.

JavaScript Engines are complicated. But it works on some simple basics:

- The engine reads ("parses:) the script.
- Then it converts or compiles the script to the machine language.
- After that machine code runs.

Here, JavaScript engine applies optimizations at each step of the process. It reads a compiled script and analyzes the data that passes in JavaScript engine. After that, it applies optimizations to the machine code from that acquired knowledge. When this process is completed, scripts run quite fast.

JavaScript's functionality depends on the environment it's running in. For example, Node.js supports functions which allows JavaScript to read and write arbitrary files, perform network requests, object-oriented, etc. The roles that JavaScript plays in both client-side (front end) and server-side (back end) development of applications can vary wildly.
In-browser JavaScript also allows you to perform webpage manipulation, interaction with the user and with the web server.

ECMAScript is a specification governed by ECMA international aimed at standardizing JavaScript. The latest version is ECMA9 also called JavaScript 9. It is supported by all major browsers like Chrome, Firefox, Internet Explorer, etc. Though each browser has an array of unique commands that are not part of the standards.

SQL

MySQL is the most popular open-source database.

A database is a systematic collection of data. They support electronic storage and manipulation of data. Databases make data management easy.

SQL is the standard language for dealing with Relational Databases. SQL can be used to insert, search, update, and delete database records. SQL can do lots of other operations, including optimizing and maintenance of databases.

SQL stands for Structured Query language, pronounced as "S-Q-L" or sometimes as "See-Quel"... Relational databases like MySQL Database, Oracle, MS SQL Server, Sybase, etc. use ANSI SQL.

- It helps users to access data in the RDBMS system.
- It helps you to describe the data.
- It allows you to define the data in a database and manipulate that specific data.
- With the help of SQL, you can create and drop databases and tables.
- SQL offers you to use the function in a database, create a view, and stored procedure.
- You can set permissions on tables, procedures, and views.

Five types of widely used SQL queries.

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)
- Transaction Control Language (TCL)
- Data Query Language (DQL)

A list of some of the most commonly used **SQL commands**:

- **CREATE** - defines the database structure schema
- **INSERT** - inserts data into the row of a table
- **UPDATE** - updates data in a database
- **DELETE** - removes one or more rows from a table
- **SELECT** - selects the attribute based on the condition described by the WHERE clause
- **DROP** - removes tables and databases

important elements of SQL language:

- **Keywords:** Each SQL statement contains single or multiple keywords.
- **Identifiers:** Identifiers are names of objects in the database, like user IDs, tables, and columns.
- **Strings:** Strings can be either literal strings or expressions with VARCHAR or CHAR data types.
- **Expressions:** Expressions are formed from several elements, like constants, SQL operators, column names, and subqueries.
- **Search Conditions:** Conditions are used to select a subset of the rows from a table or used to control statements like an IF statement to determine control of flow.
- **Special Values:** Special values should be used in expressions and as column defaults when building tables.
- **Variables:** Sybase IQ supports local variables, global variables, and connection-level variables.
- **Comments:** Comment is another SQL element which is used to attach explanatory text to SQL statements or blocks of statements. The database server does not execute any comment.
- **NULL Value:** Use NULL, which helps you to specify a value that is unknown, missing, or not applicable.

MySQL is an open source relational database.
MySQL is cross platform which means it runs on a number of different platforms such as Windows, Linux, and Mac OS etc.

- MySQL supports multiple storage engines each with its own specifications while other systems like SQL server only support a single storage engine
- MySQL has high performance compared to other relation database systems. This is due to its simplicity in design and support for multiple-storage engines.
- Cost effective, it's relatively cheaper in terms of cost when compared to other relational databases. In fact, the community edition is free. The commercial edition has a licensing fee which is also cost effective compared to licensing fees for products such as Microsoft SQL Server.

- Cross platform - MySQL works on many platforms which means it can be deployed on most machines. Other systems such as MS SQL Server only run on the windows platform.

In order to interact with MySQL, you will need a **server access tool** that can communicate with MySQL server. MySQL supports multiple user connections.

HTML

First developed by Tim Berners-Lee in 1990, HTML is short for Hypertext Markup Language. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.

HTML code ensures the proper formatting of text and images for your Internet browser. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance).

Most HTML tags have an opening tag containing the tag name, tag attributes, a closing tag containing a forward slash, and the tag name being closed. For tags that do not have a closing tag like <img>, it is best practice to end the tag with a forward slash.

Most tags are contained in a less than and greater than angle brackets, and everything between the open and close tag is displayed or affected by the tag.

1. The DOCTYPE line describes what version of HTML the page was written in so that an Internet browser can interpret the text that follows.
2. The <html> opening tag lets the browser know that it is reading HTML code.
3. The <head> section contains information about the page, such as its title, meta tags, and where to locate the CSS file.
4. The <body> section contains everything that's viewable on the browser. For example, all the text seen here is contained in the body tags.
5. The <h1> tag is the visible heading of the page.
6. The <p> tag is a paragraph of text. Most web pages (like this one) have several paragraph tags.

7. Contained in the paragraph is the <b> tag that bolds the word example in the paragraph.
8. Finally, the closing tags wrap each of the above tags.

Because HTML is a markup language, it can be created and viewed in any text editor if saved with a *.htm* or *.html* file extension. However, most find it easier to design and create web pages in HTML using an HTML editor.
Once the HTML file is created, it can be viewed locally or uploaded to a web server to be viewed online using a browser.

HTML files use either the .htm or .html file extension. Older versions of Windows (Windows 3.x) only allow three-letter file extensions, so they used .htm instead of .html. However, both file extensions have the same meaning, and either may be used today. That being said, we recommend sticking to one naming convention as certain web servers may prefer one extension over the other.

CSS

Short for **cascading style sheets**, **CSS** is a language used to describe reusable styles for presenting documents written in a markup language. Its concept was originated by Håkon Wium Lie in 1994. In December 1996, CSS was made a specification by the W3C and today allows web developers to alter the layout and appearance of their web pages. For example, CSS may be used to change the font used in certain HTML element, and its size and colour. A single CSS file may be linked to multiple pages, which allows a developer to change the appearance of all the pages at the same time.

The following box contains a basic example of using CSS code to define fonts, the colour of hyperlinks, and the colour of a link when the mouse cursor hovers over. In this specific example, we are only changing the HTML tags <a> and <body>, rather than creating any new class or id selectors.

If you want to use the CSS code on multiple pages, we suggest storing the code in a separate CSS file and then loading it on every page. For example, the CSS code shown in the first box on this page can be copied and pasted into a file with the .css file extension.

After the file is saved, it must linked to in the head of the HTML code using the <link> tag. The following box shows an example of this element in use.

<link rel="stylesheet" Type="text/css" href="*URL or path to css file here*">

If you named the CSS file example.css, and it's in the same directory as the HTML file it's being loaded from, the following line would link the CSS file.

<link rel="stylesheet" Type="text/css" href="example.css">

CSS is a **style sheet language** that gives appearance changes to a markup language. For example, HTML is used to create the basic layout of a web page such as this paragraph of text. CSS is used to define the font, font size, font weight, its position, and other visual settings.

Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Bootstrap is a HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container. While the latter always fills the width of the web page, the former uses one of the four predefined fixed widths, depending on the size of the screen showing the page:

Smaller than 576 pixels

576–768 pixels

768–992 pixels

992–1200 pixels

Larger than 1200 pixels

Once a container is in place, other Bootstrap layout components implement a CSS Flexbox layout through defining rows and columns.

A precompiled version of Bootstrap is available in the form of one CSS file and three JavaScript files that can be readily added to any project. The raw form of Bootstrap, however, enables developers to implement further customization and size optimizations. This raw form is modular, meaning that the developer can remove unneeded components, apply a theme and modify the uncompiled Sass files.

CHAPTER 2: PROPOSED SYSTEM

2.1 Proposed System

The aim of the proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work.

- Security of data
- Ensure data accuracy
- Minimize manual data entry
- Minimize time needed for the various processing
- Greater efficiency
- Better service
- User friendly and interactive
- Time saving

2.2 Objectives of System

The main objective of the project is to manage the details of Card, Orders, Order Update and the Customer.

It manages all the information related to the Business Card in differ aspects. The project has an administrative module which has responsibility of handling the customers.

The purpose of the project is to build an application program to reduce the redundant and manual work for managing the Business Card.

It helps in keeping the information secure and trackable as well.

2.3 User Requirements

• Users should be able to use the application from any Web browser supporting HTML 3.2 (or later) and cookies.

• Visitors new to the site should be able to register by themselves. Users will be differentiated by unique user identifiers.

• Site visitors should be able to purchase Card or services via the Visiting card.

• Users should be able to view a complete list of specified items available through the site.

• Users should be able to search for items by related attributes.

• Site visitors should be able to search the database using relevant keywords to identify items of interest.

• Users should be able to view the status of items they have Procced.

•Large numbers of users should be able to use the application simultaneously. The performance of the application should not degrade with an increase in the number of goods or services offered.

2.4 Agile Model (Lifecycle Model)

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

Here is a graphical illustration of the Agile Model −

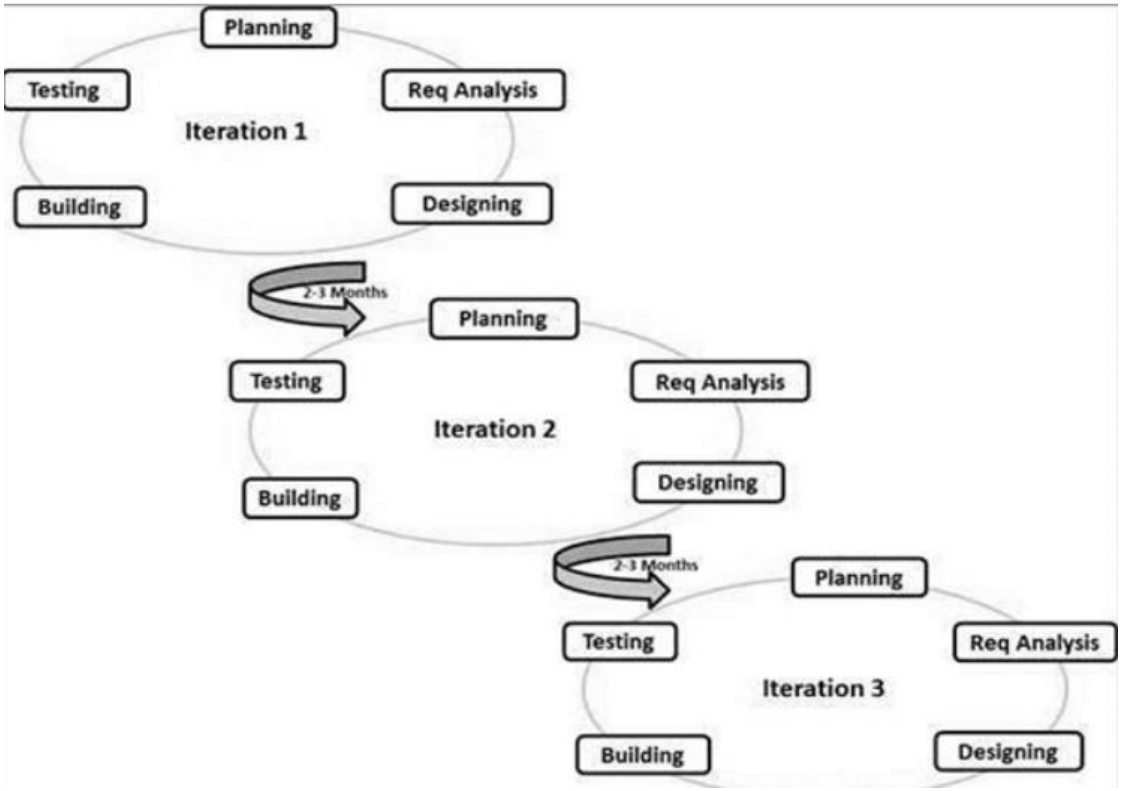


Figure 2.4.1 Agile Model

The Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

Following are the Agile Manifesto principles −

- **Individuals and interactions** − In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.

- **Working software** − Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
- **Customer collaboration** − As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
- **Responding to change** − Agile Development is focused on quick responses to change and continuous development.

Agile over another Traditional SDLC Models

Agile is based on the **adaptive software development methods**, whereas the traditional SDLC models like the waterfall model is based on a predictive approach. Predictive teams in the traditional SDLC models usually work with detailed planning and have a complete forecast of the exact tasks and features to be delivered in the next few months or during the product life cycle.

Predictive methods entirely depend on the **requirement analysis and planning** done in the beginning of cycle. Any changes to be incorporated go through a strict change control management and prioritization.

Agile uses an **adaptive approach** where there is no detailed planning and there is clarity on future tasks only in respect of what features need to be developed. There is feature driven development and the team adapts to the changing product requirements dynamically. The product is tested very frequently, through the release iterations, minimizing the risk of any major failures in future.

Customer Interaction is the backbone of this Agile methodology, and open communication with minimum documentation are the typical features of Agile development environment. The agile teams work in close collaboration with each other and are most often located in the same geographical location.

Agile Model - Pros and Cons

Agile methods are being widely accepted in the software world recently. However, this method may not always be suitable for all products. Here are some pros and cons of the Agile model.

The advantages of the Agile Model are as follows −

- Is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required.
- Easy to manage.
- Gives flexibility to developers.

The disadvantages of the Agile Model are as follows −

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- There is a very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.

CHAPTER 3: ANALYSIS & DESIGN

Why we use UML diagrams

A Unified Modeling Language (UML) diagram provides a visual representation of an aspect of a system.

UML diagrams illustrate the quantifiable aspects of a system that can be described visually, such as relationships, behavior, structure, and functionality. For example, a class diagram describes the structure of the system or the details of an implementation, while a sequence diagram shows the interaction between objects over time.

In a UML diagram, the diagram elements visually represent the classifiers in a system or application. These classifiers are the diagrammatic representation of a source element. UML diagrams provide views of source elements; however, diagram elements do not have semantic value.

UML diagrams can help system architects and developers understand, collaborate on, and develop an application. High-level architects and managers can use UML diagrams to visualize an entire system or project and separate applications into smaller components for development.

System developers can use UML diagrams to specify, visualize, and document applications, which can increase efficiency and improve their application design. UML diagrams can also help identify patterns of behavior, which can provide opportunities for reuse and streamlined applications.

The visual representation of a system that UML diagrams provide can offer both low-level and high-level insight into the concept and design of an application.

You can use a wide variety of diagram types to model a system or application, based on the system, audience, and detail of the model you create. Depending on the diagram choice, you can select the detail and level of abstraction that the diagrams display.

3.1 Use Case Diagrams

What is the Use Case Diagram?

Use Case Diagram captures the system's functionality and requirements by using actors and use cases. Use Cases model the services, tasks, function that a system needs to perform. Use cases represent high-level functionalities and how a user will handle the system. Use-cases are the core concepts of Unified Modelling language modeling.

Why Use-Case diagram?

A Use Case consists of use cases, persons, or various things that are invoking the features called as actors and the elements that are responsible for implementing the use cases. Use case diagrams capture the dynamic behavior of a live system. It models how an external entity interacts with the system to make it work. Use case diagrams are responsible for visualizing the external things that interact with the part of the system.

Use-case diagram notations

Following are the common notations used in a use case diagram:

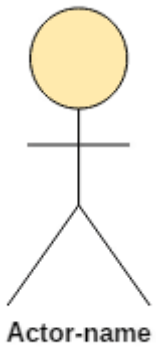
Use-case:

Use cases are used to represent high-level functionalities and how the user will handle the system. A use case represents a distinct functionality of a system, a component, a package, or a class. It is denoted by an oval shape with the name of a use case written inside the oval shape. The notation of a use case in UML is given below:

UML UseCase Notation

Actor:

It is used inside use case diagrams. The actor is an entity that interacts with the system. A user is the best example of an actor. An actor is an entity that initiates the use case from outside the scope of a use case. It can be any element that can trigger an interaction with the use case. One actor can be associated with multiple use cases in the system. The actor notation in UML is given below.



UML Actor Notation

How to draw a use-case diagram?

To draw a use case diagram in UML first one need to analyse the entire system carefully. You have to find out every single function that is provided by the system. After all the functionalities of a system are found out, then these functionalities are converted into various use cases which will be used in the use case diagram.

A use case is nothing but a core functionality of any working system. After organizing the use cases, we have to enlist the various actors or things that are going to interact with the system. These actors are responsible for invoking the functionality of a system. Actors can be a person or a thing. It can also be a private entity of a system. These actors must be relevant to the functionality or a system they are interacting with.

After the actors and use cases are enlisted, then you have to explore the relationship of a particular actor with the use case or a system. One must identify the total number of ways an actor could interact with the system. A single actor can interact with multiple use cases at the same time, or it can interact with numerous use cases simultaneously.

Following rules must be followed while drawing use-case for any system:

The name of an actor or a use case must be meaningful and relevant to the system.

Interaction of an actor with the use case must be defined clearly and in an understandable way.

Annotations must be used wherever they are required.

If a use case or an actor has multiple relationships, then only significant interactions must be displayed.

Tips for drawing a use-case diagram

1. A use case diagram should be as simple as possible.
2. A use case diagram should be complete.
3. A use case diagram should represent all interactions with the use case.
4. If there are too many use cases or actors, then only the essential use cases should be represented.
5. A use case diagram should describe at least a single module of a system.
6. If the use case diagram is large, then it should be generalized.

When to use a use-case diagram?

A use case is a unique functionality of a system which is accomplished by a user. A purpose of use case diagram is to capture core functionalities of a system and visualize the interactions of various things called as actors with the use case. This is the general use of a use case diagram.

The use case diagrams represent the core parts of a system and the workflow between them. In use case, implementation details are hidden from the external use only the event flow is represented.

With the help of use case diagrams, we can find out pre and post conditions after the interaction with the actor. These conditions can be determined using various test cases.

In general use case diagrams are used for:

1. Analyzing the requirements of a system
2. High-level visual software designing
3. Capturing the functionalities of a system
4. Modeling the basic idea behind the system
5. Forward and reverse engineering of a system using various test cases.

Use cases are intended to convey desired functionality so the exact scope of a use case may vary according to the system and the purpose of creating UML model.

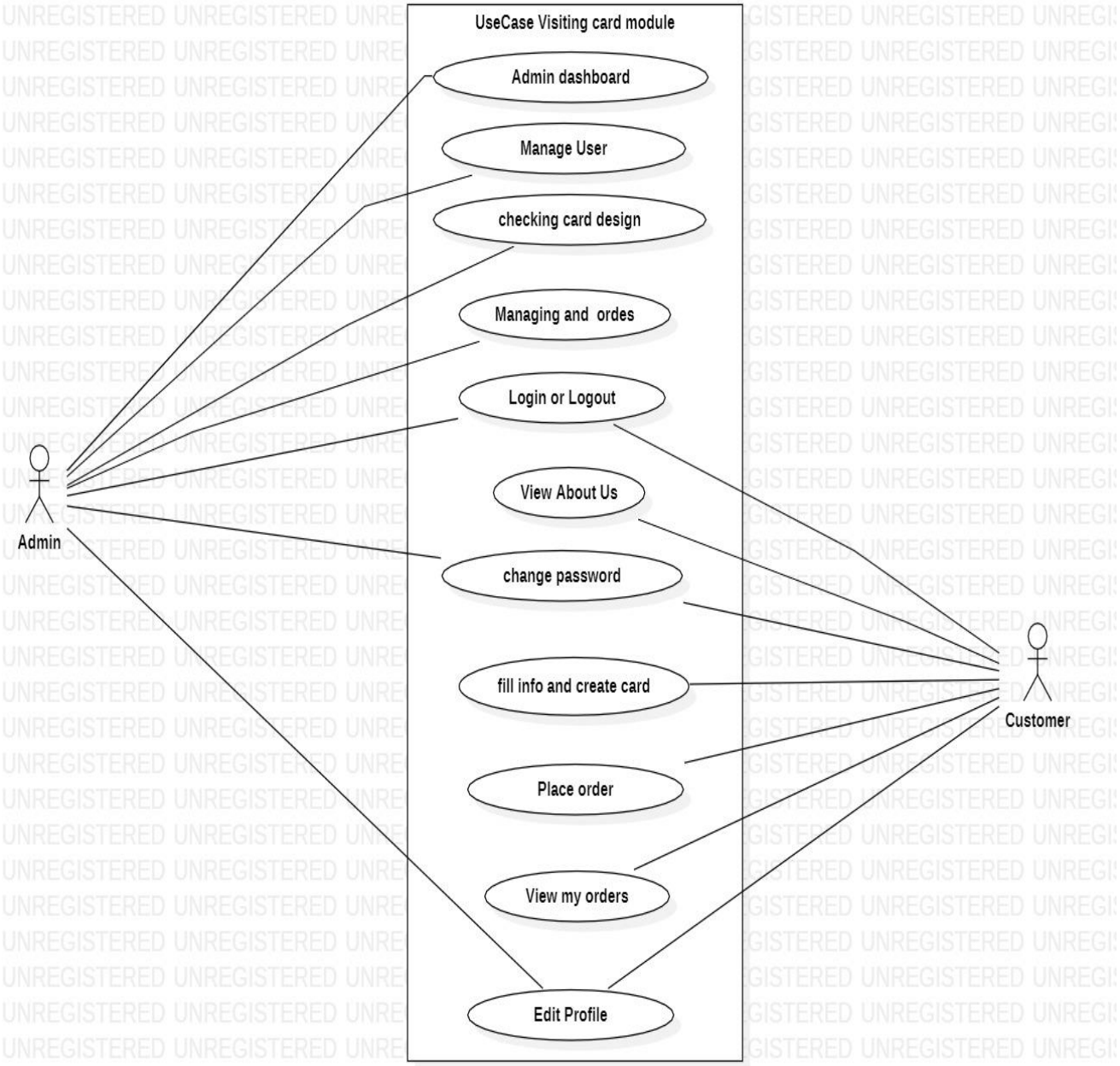


Figure 3.1 UseCase Diagram

3.2 Activity diagrams

What is an Activity Diagram in UML?

ACTIVITY DIAGRAM is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The basic purpose of activity diagrams is to capture the dynamic behavior of the system.. It is also called object-oriented flowchart.

This UML diagram focuses on the execution and flow of the behavior of a system instead of implementation. Activity diagrams consist of activities that are made up of actions that apply to behavioral modeling technology.

Components of Activity Diagram

Activities

It is a behavior that is divided into one or more actions. Activities are a network of nodes connected by edges. There can be action nodes, control nodes, or object nodes. Action nodes represent some action. Control nodes represent the control flow of an activity. Object nodes are used to describe objects used inside an activity. Edges are used to show a path or a flow of execution. Activities start at an initial node and terminate at a final node.

Activity partition/swimlane

An activity partition or a swimlane is a high-level grouping of a set of related actions. A single partition can refer to many things, such as classes, use cases, components, or interfaces.

If a partition cannot be shown clearly, then the name of a partition is written on top of the name of an activity.

Fork and Join nodes

Using a fork and join nodes, concurrent flows within an activity can be generated. A fork node has one incoming edge and numerous outgoing edges. It is similar to one too many decision parameters. When data arrives at an incoming edge, it is duplicated and split across numerous outgoing edges simultaneously. A single incoming flow is divided into multiple parallel flows.

A join node is opposite of a fork node as It has many incoming edges and a single outgoing edge. It performs logical AND operation on all the incoming edges. This helps you to synchronize the input flow across a single output edge.

Pins

An activity diagram that has a lot of flows gets very complicated and messy.

Pins are used to clearing up the things. It provides a way to manage the execution flow of activity by sorting all the flows and cleaning up messy thins. It is an object node that represents one input to or an output from an action.

Both input and output pins have precisely one edge.

Why use Activity Diagrams?

Activity diagram in UML allows you to create an event as an activity which contains a collection of nodes joined by edges. An activity can be attached to any modeling element to model its behavior. Activity diagrams are used to model,

Use cases

Classes

Interfaces

Components

Collaborations

Activity diagrams are used to model processes and workflows. The essence of a useful activity diagram is focused on communicating a specific aspect of a system's dynamic behavior. Activity diagrams capture the dynamic elements of a system.

Activity diagram is similar to a flowchart that visualizes flow from one activity to another activity. Activity diagram is identical to the flowchart, but it is not a flowchart. The flow of activity can be controlled using various control elements in the UML flow diagram. In simple words, an activity diagram is used to activity diagrams that describe the flow of execution between multiple activities.

Activity Diagram Notations

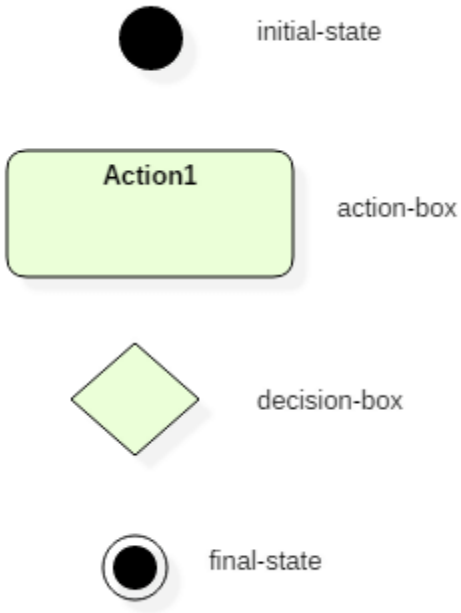
Activity diagrams symbols can be generated by using the following notations:

Initial states: The starting stage before an activity takes place is depicted as the initial state

Final states: The state which the system reaches when a specific process ends is known as a Final State

State or an activity box:

Decision box: It is a diamond shape box which represents a decision with alternate paths. It represents the flow of control.



Activity Diagram Notation and Symbol

How to draw an activity diagram?

Activity diagram is a flowchart of activities. It represents the workflow between various system activities. Activity diagrams are similar to the flowcharts, but they are not flowcharts. Activity diagram is an advancement of a flowchart that contains some unique capabilities.

Activity diagrams include swim lanes, branching, parallel flow, control nodes, expansion nodes, and object nodes. Activity diagram also supports exception handling.

To draw an activity diagram, one must understand and explore the entire system. All the elements and entities that are going to be used inside the diagram must be known by the user. The central concept which is nothing but an activity must be clear to the user. After analyzing all activities, these activities should be explored to find various constraints that are applied to activities. If there is such a constraint, then it should be noted before developing an activity diagram.

All the activities, conditions, and associations must be known. Once all the necessary things are gathered, then an abstract or a prototype is generated, which is later converted into the actual diagram.

Following rules must be followed while developing an activity diagram,

1. All activities in the system should be named.
2. Activity names should be meaningful.
3. Constraints must be identified.
4. Activity associations must be known.

When Use Activity Diagram

Activity diagram is used to model business processes and workflows. These diagrams are used in software modeling as well as business modeling.

1. Most commonly activity diagrams are used to,
2. Model the workflow in a graphical way, which is easily understandable.
3. Model the execution flow between various entities of a system.
4. Model the detailed information about any function or an algorithm which is used inside the system.
5. Model business processes and their workflows.
6. Capture the dynamic behavior of a system.
7. Generate high-level flowcharts to represent the workflow of any application.
8. Model high-level view of an object-oriented or a distributed system.

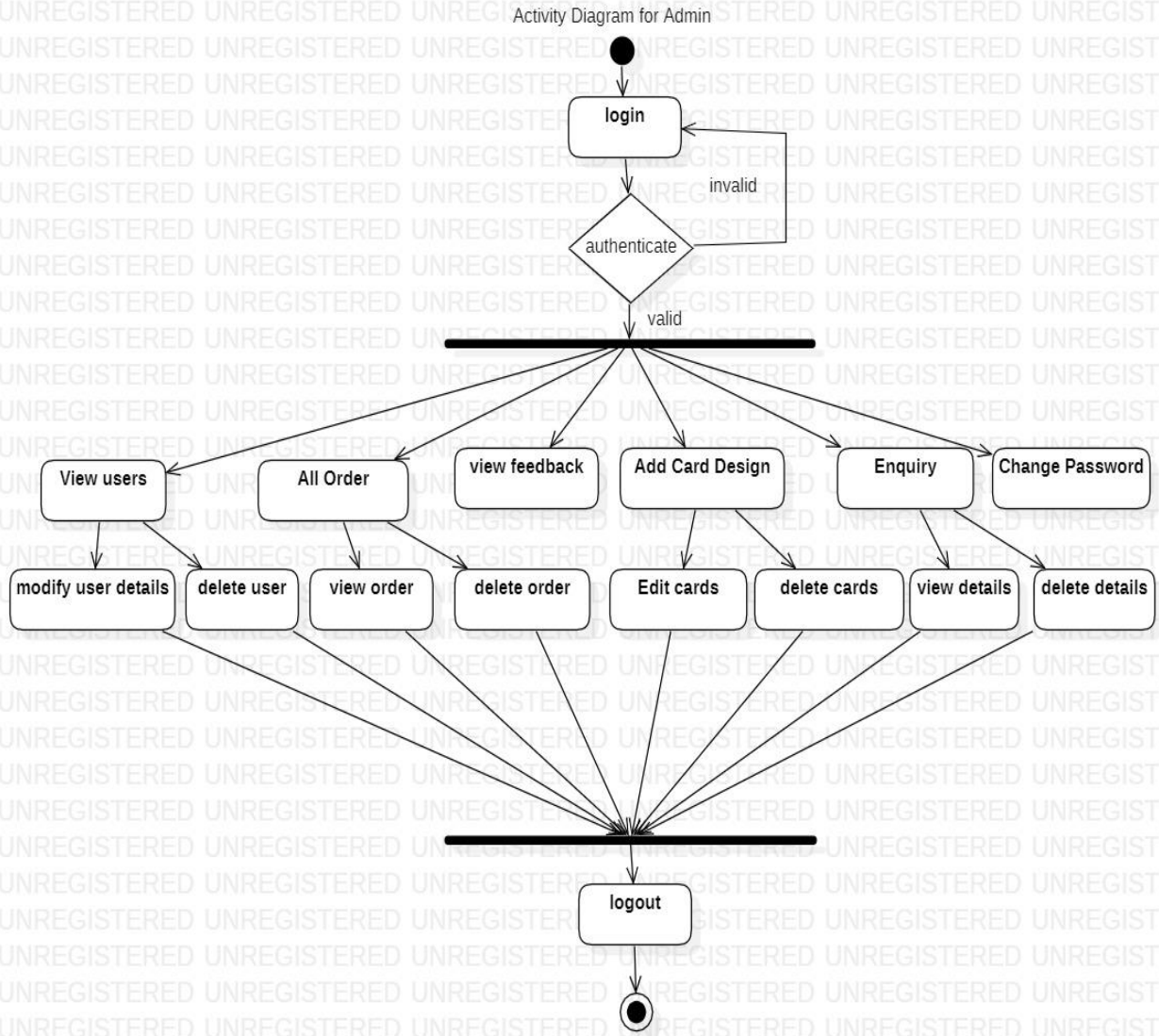


Figure 3.2 Activity Diagram – Admin

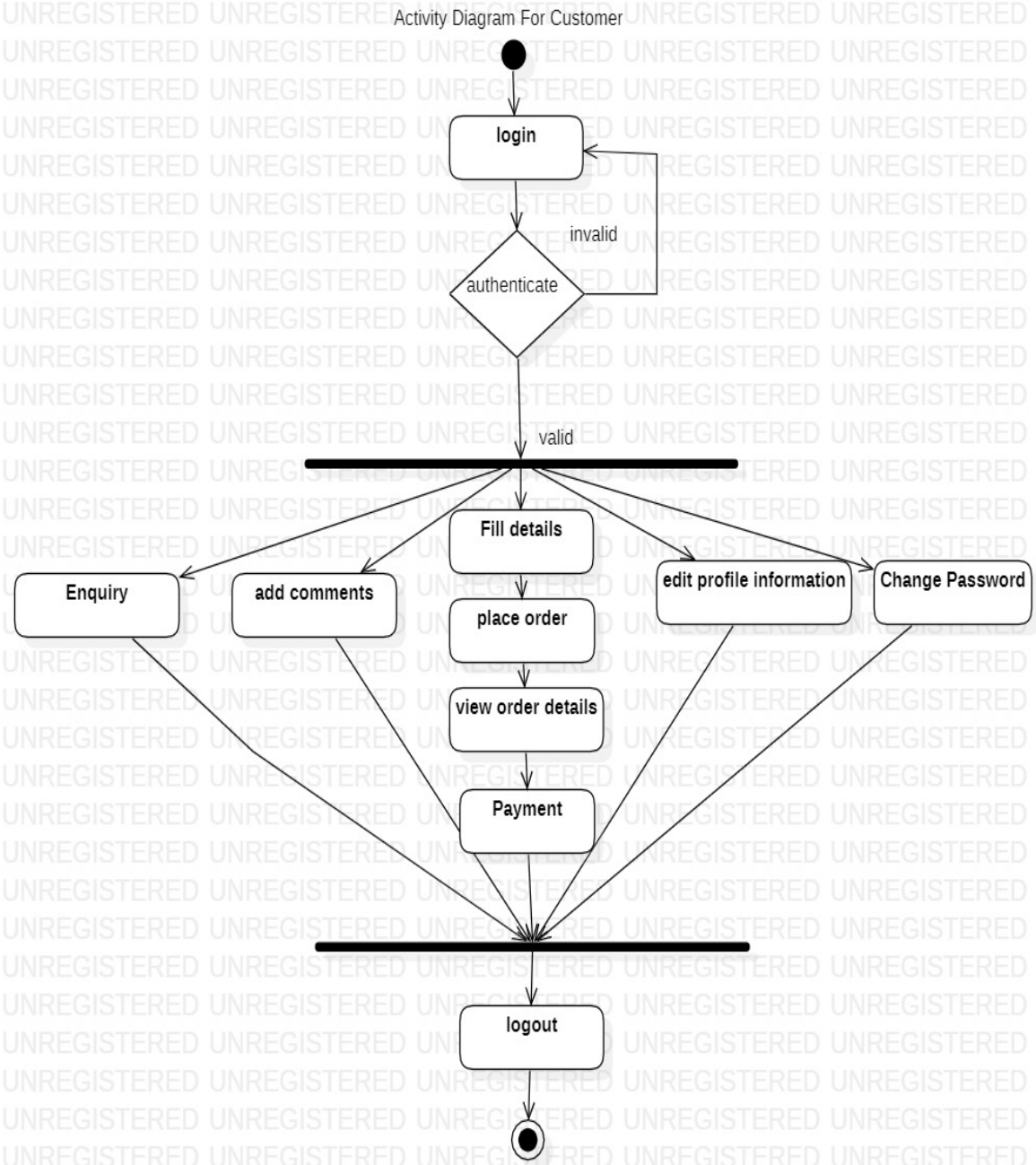


Figure 3.2.2 Activity Diagram –Customer

3.3 Sequence Diagrams

What is a Sequence Diagram?

A Sequence Diagram simply depicts interaction between objects in a sequential order. The purpose of a sequence diagram in UML is to visualize the sequence of a message flow in the system. The sequence diagram shows the interaction between two lifelines as a time-ordered sequence of events.

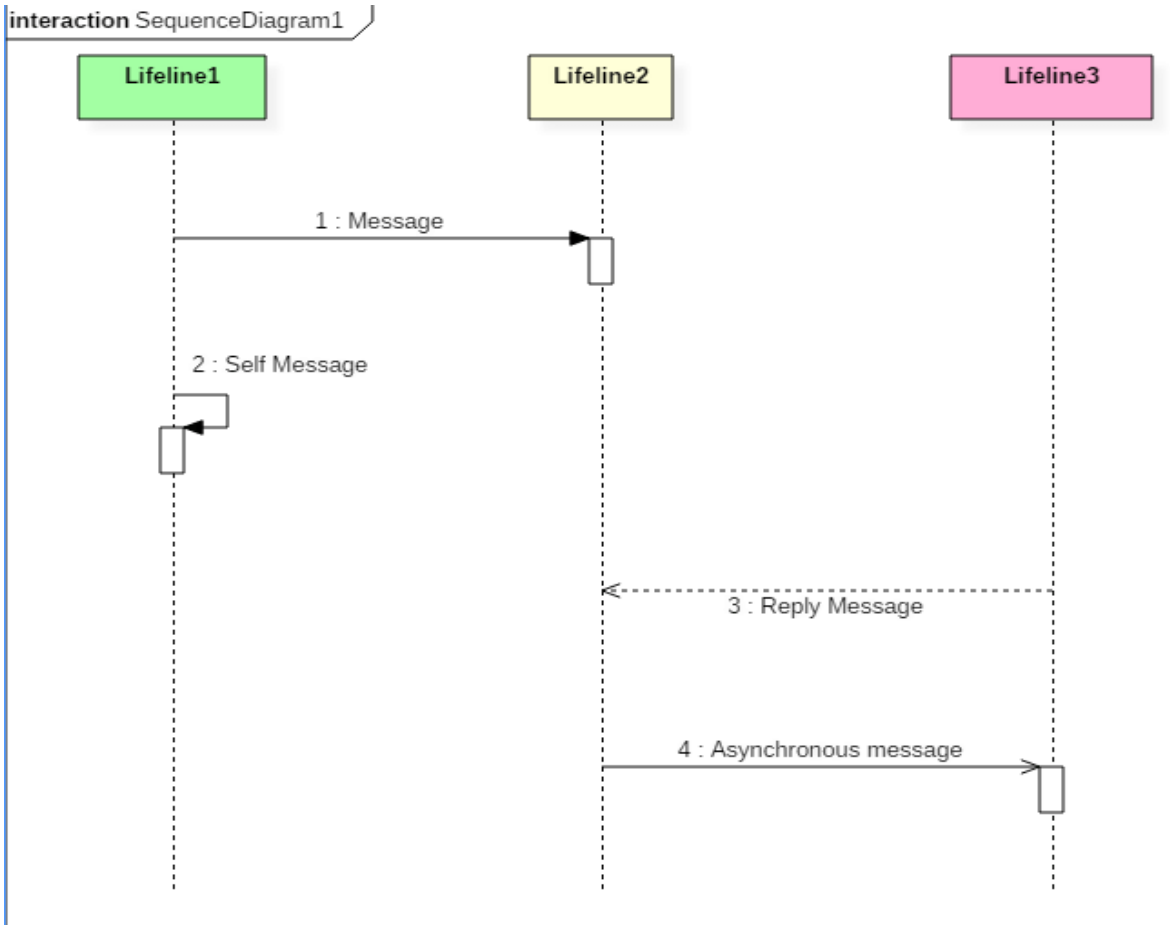
A sequence diagram shows an implementation of a scenario in the system. Lifelines in the system take part during the execution of a system.

In a sequence diagram, a lifeline is represented by a vertical bar.

A message flow between two or more objects is represented using a vertical dotted line which extends across the bottom of the page.

In a sequence diagram, different types of messages and operators are used which are described above.

In a sequence diagram, iteration and branching are also used.



Notations in Sequence Diagram

The above sequence diagram contains lifeline notations and notation of various messages used in a sequence diagram such as a create, reply, asynchronous message, etc.

Benefits of a Sequence Diagram

1. Sequence diagrams are used to explore any real application or a system.
2. Sequence diagrams are used to represent message flow from one object to another object.
3. Sequence diagrams are easier to maintain.
4. Sequence diagrams are easier to generate.
5. Sequence diagrams can be easily updated according to the changes within a system.
6. Sequence diagram allows reverse as well as forward engineering.

Drawbacks of a sequence diagram

1. Sequence diagrams can become complex when too many lifelines are involved in the system.
2. If the order of message sequence is changed, then incorrect results are produced.
3. Each sequence needs to be represented using different message notation, which can be a little complex.
4. The type of message decides the type of sequence inside the diagram.

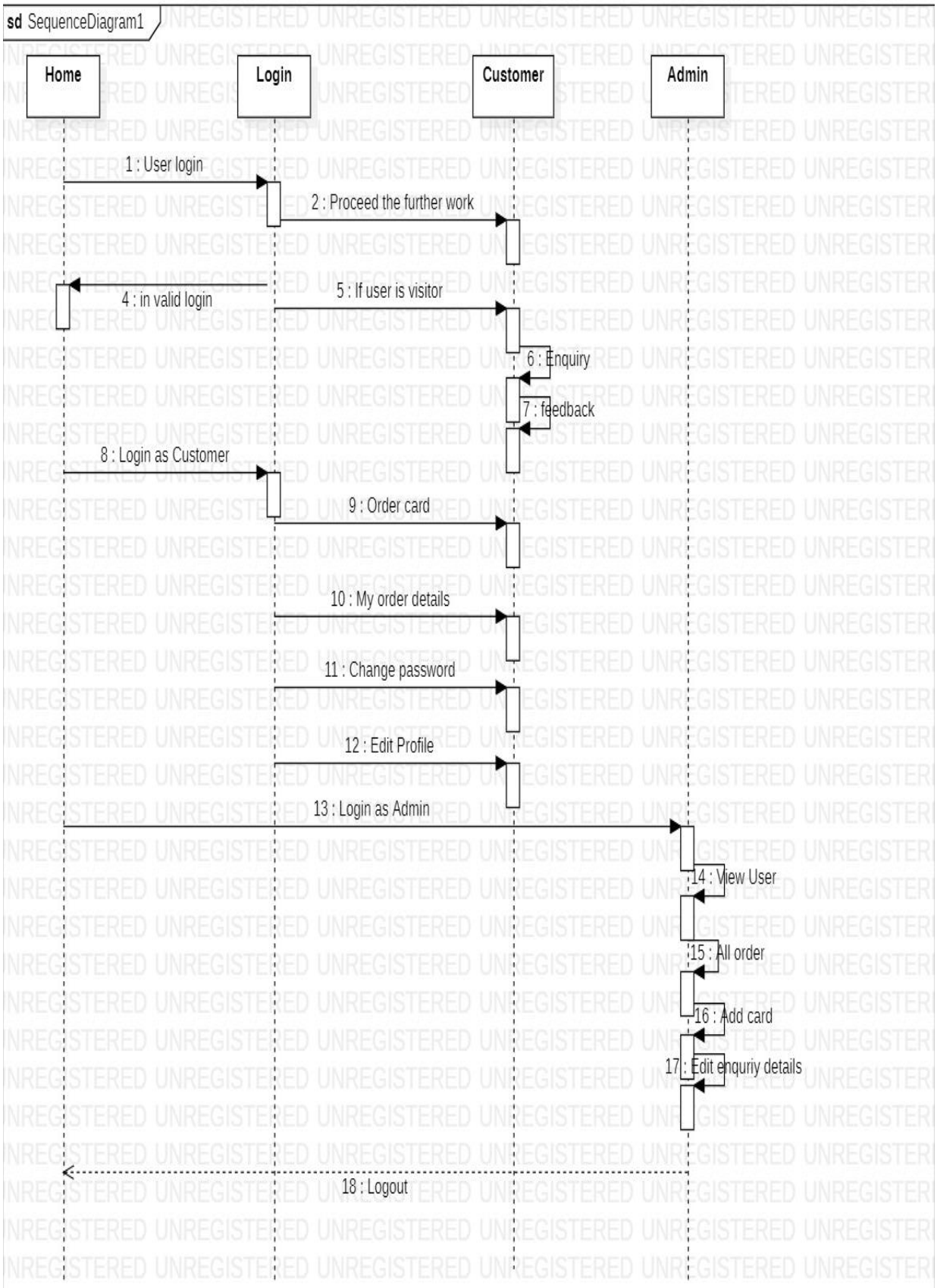


Figure 3.3 Sequence Diagram

3.4 Class diagram

What is Class in UML Diagram?

A Class in UML diagram is a blueprint used to create an object or set of objects. The Class defines what an object can do. It is a template to create various objects and implement their behavior in the system. A Class in UML is represented by a rectangle that includes rows with class names, attributes, and operations.

What is Class Diagram?

A Class Diagram in Software engineering is a static structure that gives an overview of a software system by displaying classes, attributes, operations, and their relationships between each other. This Diagram includes the class name, attributes, and operation in separate designated compartments. Class Diagram helps construct the code for the software application development.

Class Diagram defines the types of objects in the system and the different types of relationships that exist among them. It gives a high-level view of an application. This modeling method can run with almost all Object-Oriented Methods. A class can refer to another class. A class can have its objects or may inherit from other classes.

Benefits of Class Diagram

Class Diagram Illustrates data models for even very complex information systems

It provides an overview of how the application is structured before studying the actual code. This can easily reduce the maintenance time

It helps for better understanding of general schematics of an application.

Allows drawing detailed charts which highlights code required to be programmed

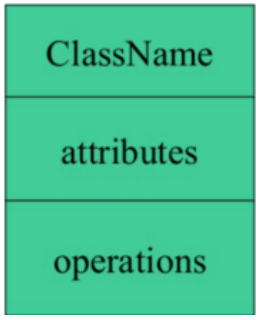
Helpful for developers and other stakeholders.

Essential elements of A UML class diagram

Essential elements of UML class diagram are:

1. Class Name
2. Attributes
3. Operations

Class Name



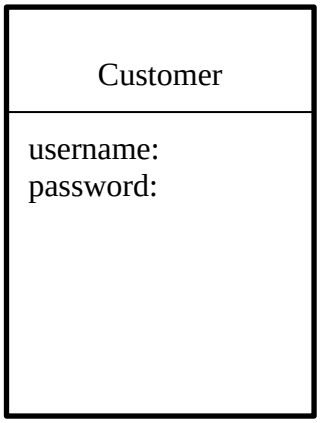
The name of the class is only needed in the graphical representation of the class. It appears in the topmost compartment. A class is the blueprint of an object which can share the same relationships, attributes, operations, & semantics. The class is rendered as a rectangle, including its name, attributes, and operations in sperate compartments.

Following rules must be taken care of while representing a class:

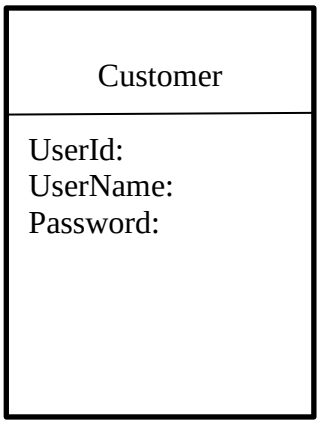
1. A class name should always start with a capital letter.
2. A class name should always be in the center of the first compartment.
3. A class name should always be written in bold format.
4. UML abstract class name should be written in italics format.

Attributes:

An attribute is named property of a class which describes the object being modeled. In the class diagram, this component is placed just below the name-compartment.



A derived attribute is computed from other attributes. For example, an age of the student can be easily computed from his/her birth date.



Attributes characteristics

1. The attributes are generally written along with the visibility factor.
2. Public, private, protected and package are the four visibilities which are denoted by +, -, #, or ~ signs respectively.
3. Visibility describes the accessibility of an attribute of a class.
4. Attributes must have a meaningful name that describes the use of it in a class.

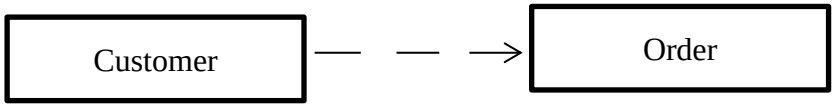
Relationships

There are mainly three kinds of relationships in UML:

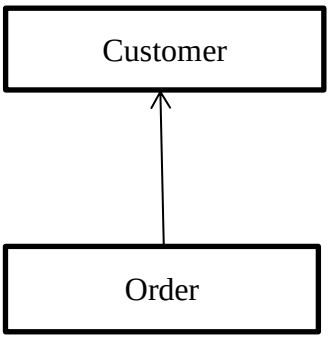
1. Dependencies
2. Generalizations
3. Associations
4. Dependency

A dependency means the relation between two or more classes in which a change in one may force changes in the other. However, it will always create a weaker relationship. Dependency indicates that one class depends on another.

In the following UML class diagram examples, Student has a dependency on College



Generalization:



A generalization helps to connect a subclass to its superclass. A sub-class is inherited from its superclass. Generalization relationship can't be used to model interface implementation. Class diagram allows inheriting from multiple super classes.

Association:

This kind of relationship represents static relationships between classes A and B. For example; an employee works for an organization.

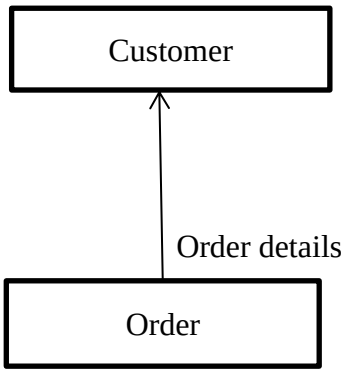
Here are some rules for Association:

Association is mostly verb or a verb phrase or noun or noun phrase.

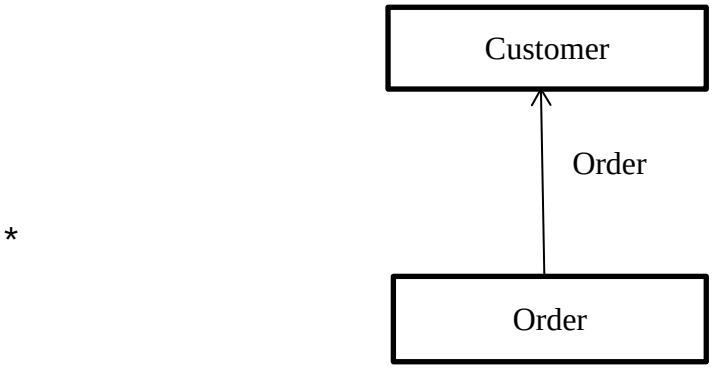
It should be named to indicate the role played by the class attached at the end of the association path.

Mandatory for reflexive associations

In this example, the relationship between Customer and order is shown which is order.



Multiplicity

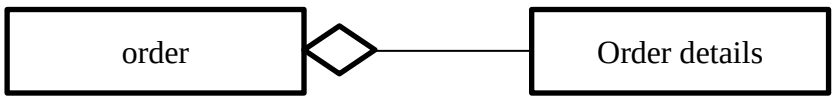


A multiplicity is a factor associated with an attribute. It specifies how many instances of attributes are created when a class is initialized. If a multiplicity is not specified, by default one is considered as a default multiplicity.

Let's say that that there are 100 students in one college. The college can have multiple students.

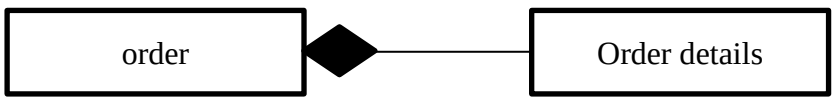
Aggregation

Aggregation is a special type of association that models a whole- part relationship between aggregate and its parts.



For example, the class college is made up of one or more student. In aggregation, the contained classes are never totally dependent on the lifecycle of the container. Here, the college class will remain even if the student is not available.

Composition:



The composition is a special type of aggregation which denotes strong ownership between two classes when one class is a part of another class.

For example, if college is composed of classes student. The college could contain many students, while each student belongs to only one college. So, if college is not functioning all the students also removed.

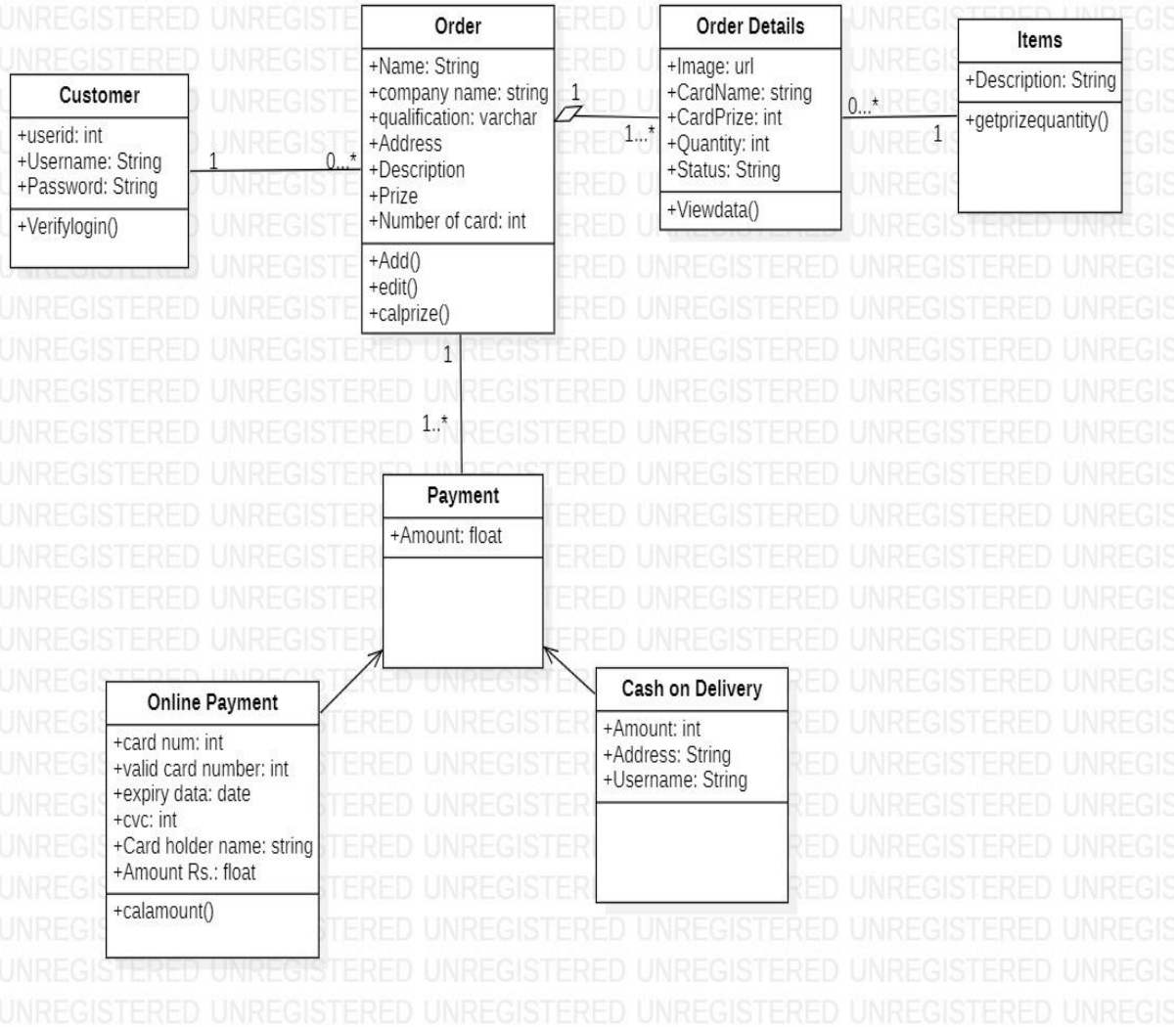


Figure 3.4 Class Diagram

3.5 Object diagrams

What is an Object Diagram?

Objects are the real-world entities whose behavior is defined by the classes. Objects are used to represent the static view of an object-oriented system. We cannot define an object without its class. Object and class diagrams are somewhat similar.

The difference between the class and object diagram is that the class diagram mainly represents the bird's eye view of a system which is also referred to as an abstract view. An object diagram describes the instance of a class. It visualizes the particular functionality of a system.

How to draw an object diagram?

1. Before drawing an object diagram, one should analyze all the objects inside the system.
2. The relations of the object must be known before creating the diagram.
3. Association between various objects must be cleared before.
4. An object should have a meaningful name that describes its functionality.
5. An object must be explored to analyze various functionalities of it.

Purpose of an object diagram:

1. It is used to describe the static aspect of a system.
2. It is used to represent an instance of a class.
3. It can be used to perform forward and reverse engineering on systems.
4. It is used to understand the behavior of an object.
5. It can be used to explore the relations of an object and can be used to analyze other connecting objects.

Applications of Object Diagrams:

1. Object diagrams play an essential role while generating a blueprint of an object-oriented system.
2. Object diagrams provide means of modeling the classes, data and other information as a set or a single unit.
3. It is used for analyzing the online or offline system. The functioning of a system can be visualized using object diagrams.

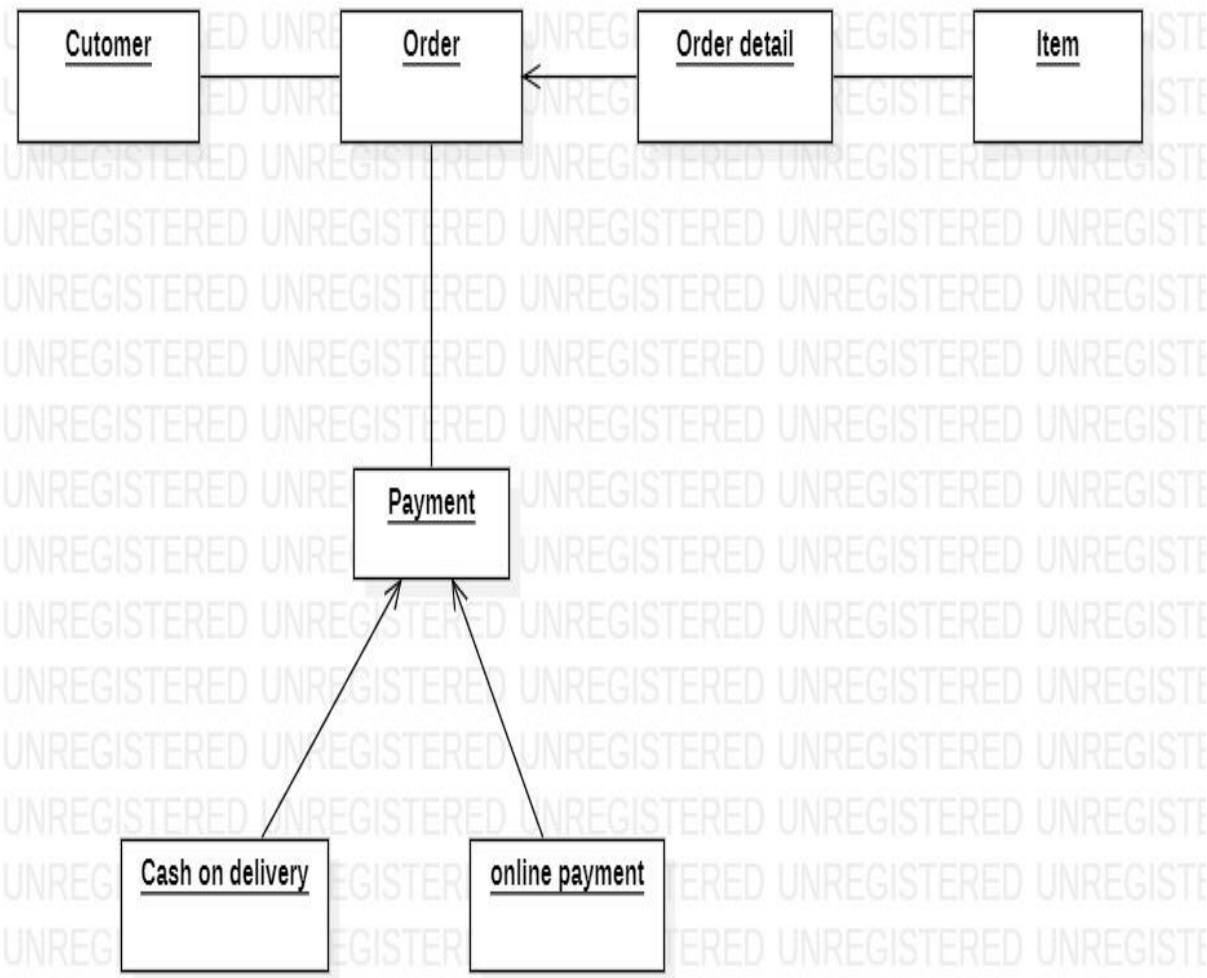


Figure 3.5 Object Diagram

3.6 ERD - Entity Relational Diagram

What is ER Modeling?

Entity Relationship Model (ER Modeling) is a graphical approach to database design. It is a high-level data model that defines data elements and their relationship for a specified software system. An ER model is used to represent real-world objects.

An Entity is a thing or object in real world that is distinguishable from surrounding environment. For example, each employee of an organization is a separate entity. Following are some of major characteristics of entities.

- An entity has a set of properties.
- Entity properties can have values.

Why use ER Model?

Now you may think why use ER modeling when we can simply create the database and all of its objects without ER modeling? One of the challenges faced when designing a database is the fact that designers, developers, and end-users tend to view data and its usage differently. If this situation is left unchecked, we can end up producing a database system that does not meet the requirements of the users.

Communication tools understood by all stakeholders(technical as well as non-technical users) are critical in producing database systems that meet the requirements of the users. ER models are examples of such tools.

ER diagrams also increase user productivity as they can be easily translated into relational tables.

Advantages of ER Model

1. Conceptually it is very simple: ER model is very simple because if we know relationship between entities and attributes, then we can easily draw an ER diagram.

2. Better visual representation: ER model is a diagrammatic representation of any logical structure of database. By seeing ER diagram, we can easily understand relationship among entities and relationship.

3. Effective communication tool: It is an effective communication tool for database designer.

4. Highly integrated with relational model: ER model can be easily converted into relational model by simply converting ER model into tables.

5. Easy conversion to any data model: ER model can be easily converted into another data model like hierarchical data model, network data model and so on.

Disadvantages of ER Model

1. Limited constraints and specification
2. Loss of information content: Some information be lost or hidden in ER model
3. Limited relationship representation: ER model represents limited relationship as compared to another data models like relational model etc.
4. No representation of data manipulation: It is difficult to show data manipulation in ER model.
5. Popular for high level design: ER model is very popular for designing high level design

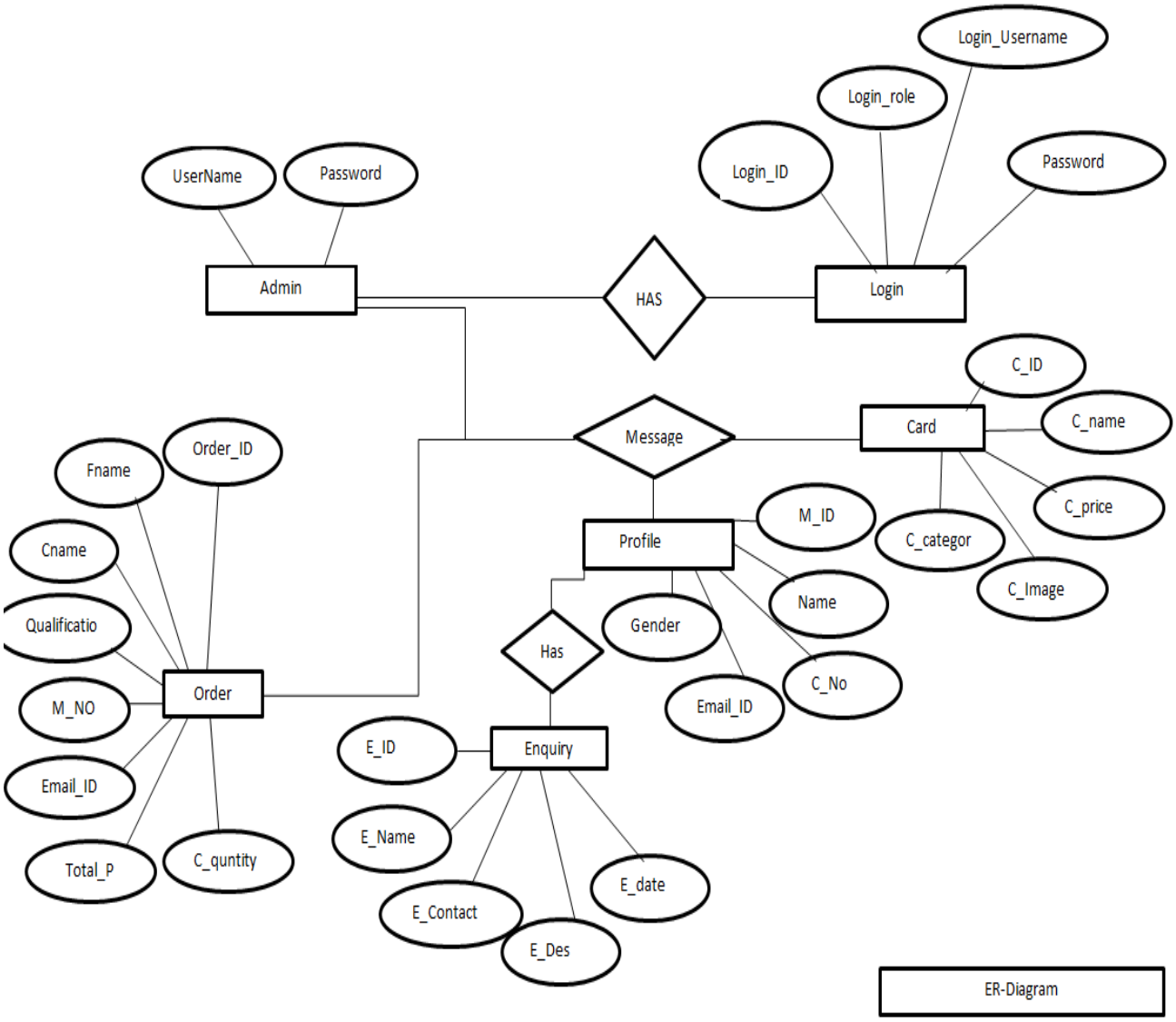


Figure 3.6 ER – Diagra

3.7 Web Site Map Diagram (in case of Web Site)

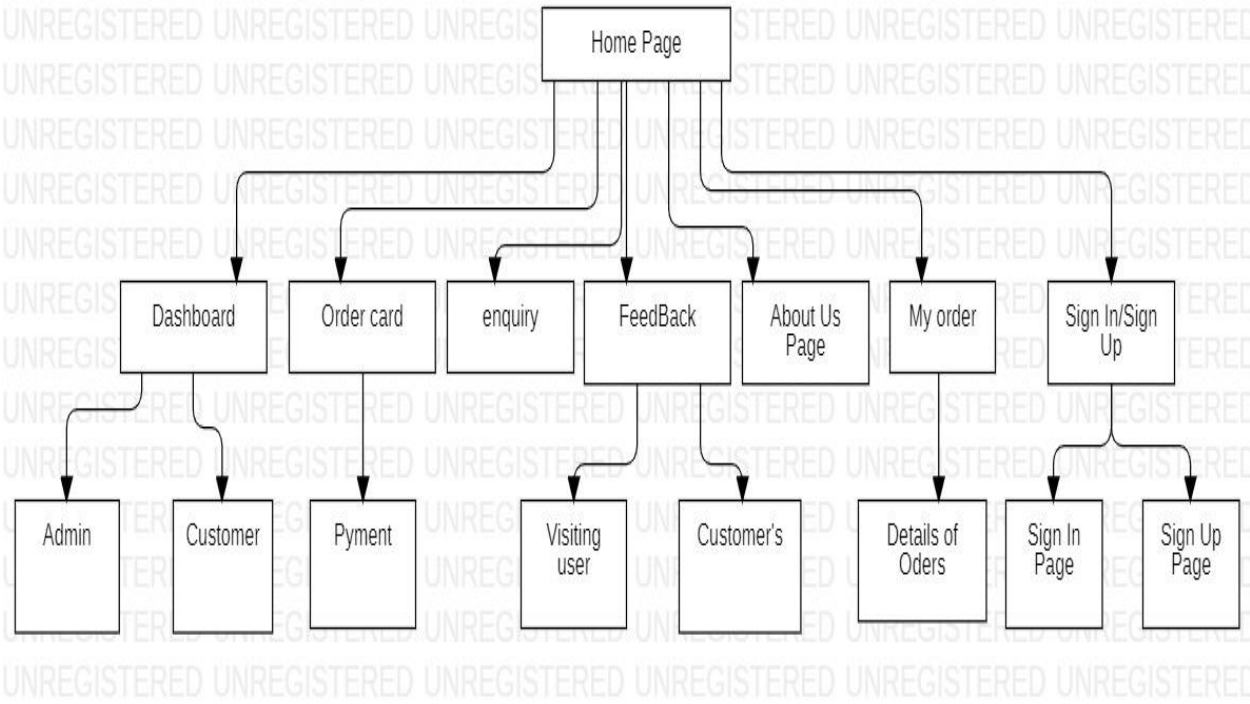


Figure 3.7 Website-map – Diagram

3.8 Schema Diagram :

What is database schema?

Database schema is a visual representation of database. The structure is described by formal language and is supported by database management system (DBMS). It represents logical view of the database.

The database schema has a list of attributes and instructions for database engine showing how data is organized and how the components are associated in a database.

A database schema describes entities and relationship among them.

What is schema?

In its literal meaning, the term schema is called “form” or “shape” of the database which shows that how the data is organized in database.

A schema is a representation of design or idea in the shape of model. A database may contain one or multiple schemas.

A database schema consists of objects like tables, views and stored procedure etc. In other words, we can say that database schema is a container of objects.

A collection of database objects is called schema in Oracle Database. These objects are associated with a database user name who is called schema owner.

Depending upon database privileges, user can have complete control over these objects.

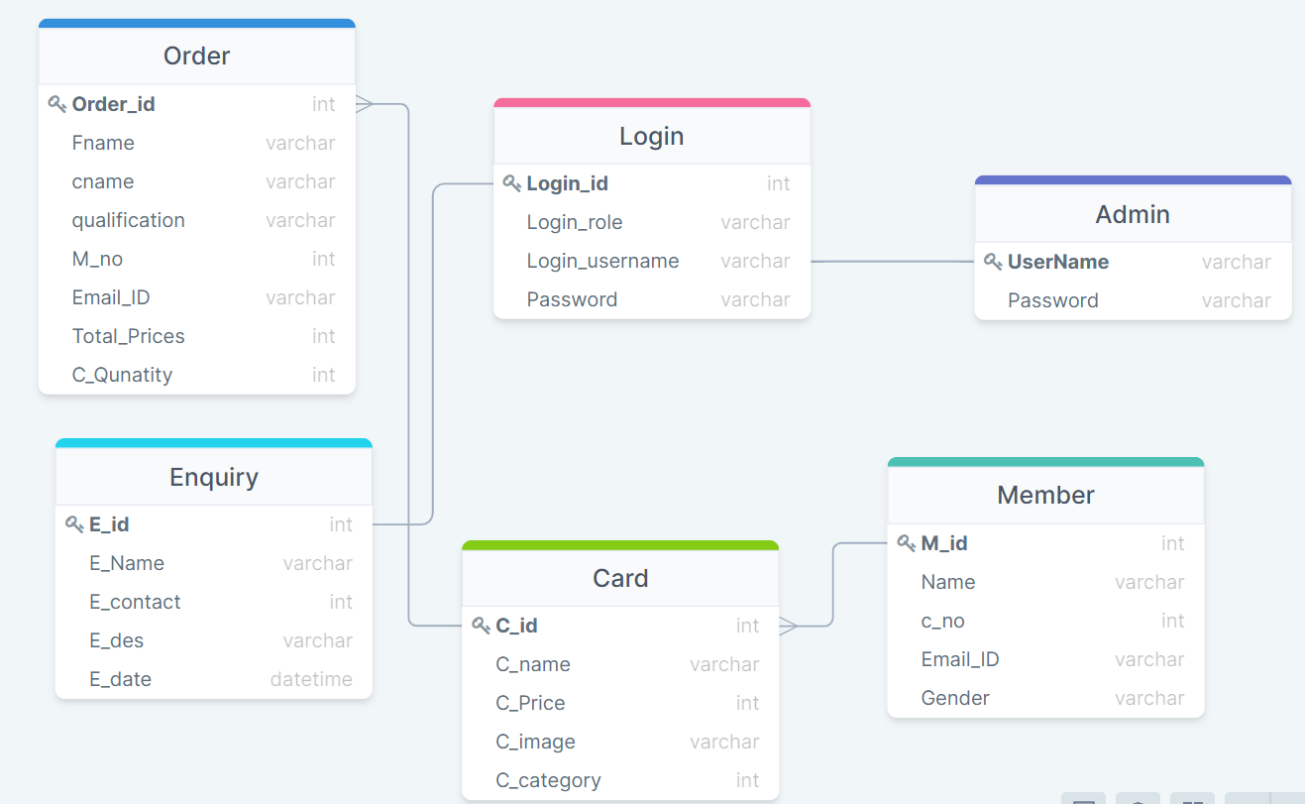
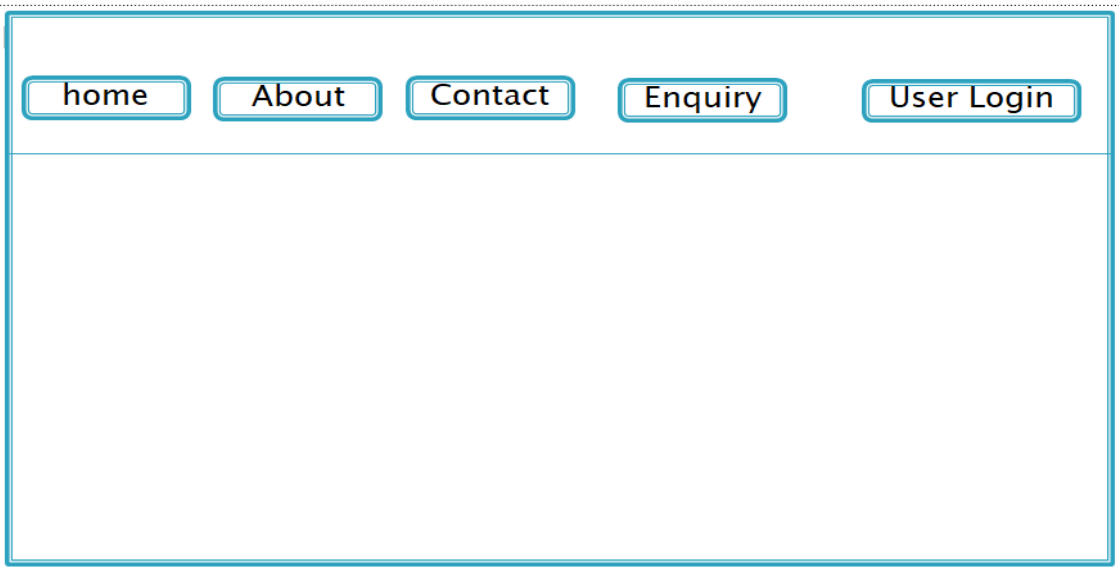


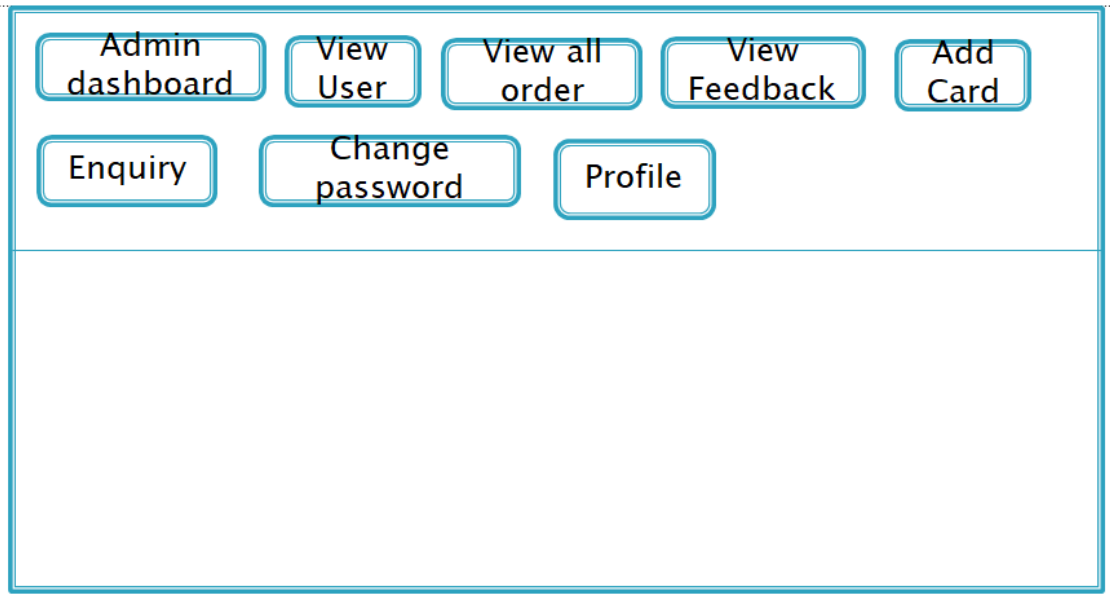
Figure 3.8 Schema Diagram

3.9 User Interface Design

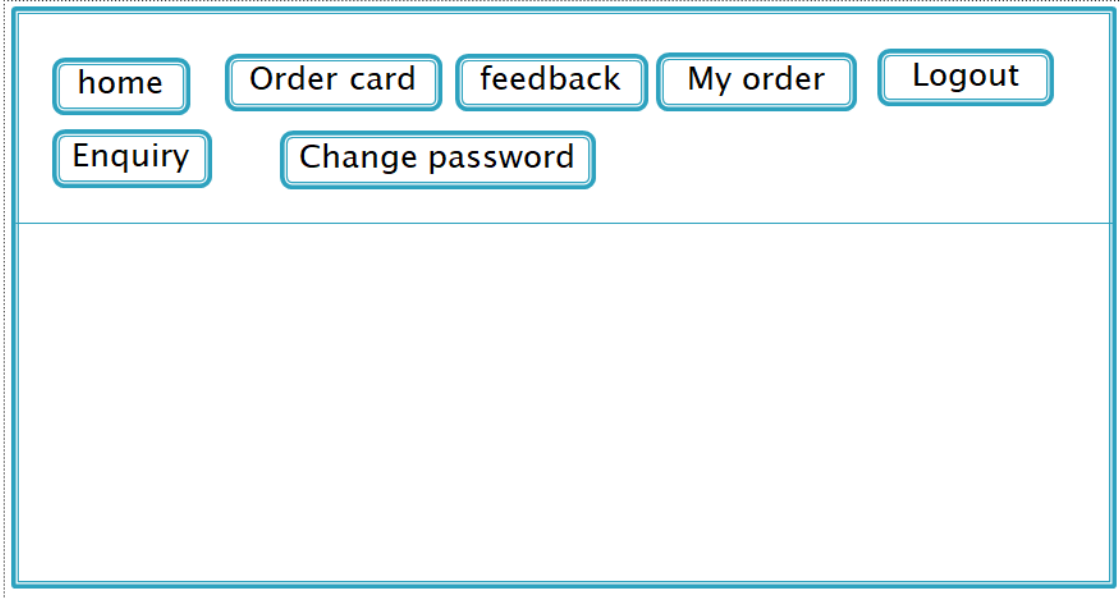
Diagram –HomePage



Admin –homepage



Customer


home
Order card
feedback
My order
Logout
Enquiry
Change password

3.10 Test Plans and Test cases

What is Unit Testing?

Unit testing is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

In SDLC, STLC, V Model, Unit testing is first level of testing done before integration testing. Unit testing is a White-Box testing technique that is usually performed by the developer. Though, in a practical world due to time crunch or reluctance of developers to tests, QA engineers also do unit testing.

Why Unit Testing?

Unit Testing is important because software developers sometimes try saving time doing minimal unit testing and this is myth because inappropriate unit testing leads to high cost Defect fixing during System Testing, Integration Testing and even Beta Testing after application is built. If proper unit testing is done in early development, then it saves time and money in the end.

Here, are the key reasons to perform unit testing in software engineering:

1. Unit tests help to fix bugs early in the development cycle and save costs.
2. It helps the developers to understand the testing code base and enables them to make changes quickly
3. Good unit tests serve as project documentation
4. Unit tests help with code re-use. Migrate both your code and your tests to your new project. Tweak the code until the tests run again.

How to do Unit Testing

In order to do Unit Testing, developers write a section of code to test a specific function in software application. Developers can also isolate this function to test more rigorously which reveals unnecessary dependencies between function being tested and other units so the dependencies can be eliminated. Developers generally use Unit-Test framework to develop automated test cases for unit testing.

Unit Testing is of two types

- Manual
- Automated

Unit testing is commonly automated but may still be performed manually. Software Engineering does not favour one over the other but automation is preferred. A manual approach to unit testing may employ a step-by-step instructional document.

Under the automated approach-

- A developer writes a section of code in the application just to test the function. They would later comment out and finally remove the test code when the application is deployed.
- A developer could also isolate the function to test it more rigorously. This is a more thorough unit testing practice that involves copy and paste of code to its own testing environment than its natural environment. Isolating the code helps in revealing unnecessary dependencies between the code being tested and other units or data spaces in the product. These dependencies can then be eliminated.
- A coder generally uses a UnitTest Framework to develop automated test cases. Using an automation framework, the developer codes criteria into the test to verify the correctness of the code. During execution of the test cases, the framework logs failing test cases. Many frameworks will also automatically flag and report, in summary, these failed test cases. Depending on the severity of a failure, the framework may halt subsequent testing.
- The workflow of Unit Testing is 1) Create Test Cases 2) Review/Rework 3) Baseline 4) Execute Test Cases.

Unit Testing Techniques

The Unit Testing Techniques are mainly categorized into three parts which are Black box testing that involves testing of user interface along with input and output, White box testing that involves testing the functional behaviour of the software application and Gray box testing that is used to execute test suites, test methods, test cases and performing risk analysis. Code coverage techniques used in Unit Testing are listed below:

- Statement Coverage
- Decision Coverage
- Branch Coverage
- Condition Coverage
- Finite State Machine Coverage

Unit Testing Advantage

- Developers looking to learn what functionality is provided by a unit and how to use it can look at the unit tests to gain a basic understanding of the unit API.
- Unit testing allows the programmer to refactor code at a later date, and make sure the module still works correctly (i.e. Regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified and fixed.
- Due to the modular nature of the unit testing, we can test parts of the project without waiting for others to be completed.

Unit Testing Disadvantages

- Unit testing can't be expected to catch every error in a program. It is not possible to evaluate all execution paths even in the most trivial programs
- Unit testing by its very nature focuses on a unit of code. Hence it can't catch integration errors or broad system level errors.

It's recommended unit testing be used in conjunction with other testing activities.

What is Integration Testing?

INTEGRATION TESTING is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated

Integration Testing focuses on checking data communication amongst these modules. Hence it is also termed as **'I & T'** (Integration and Testing), **'String Testing'** and sometimes **'Thread Testing'**.

Integration testing is performed to test individual components to check how they function together. In other words, it is performed to test the modules which are working fine individually and do not show bugs when integrated. It is the most common functional testing type and performed as automated testing.

Generally, developers build different modules of the system/software simultaneously and don't focus on others. They perform extensive black and white box functional verification, commonly known as unit tests, on the individual modules. Integration tests cause data and operational commands to flow between modules which means that they have to act as parts of a whole system rather than individual components. This typically uncovers issues with UI operations, data formats, operation timing, API calls, and database access and user interface operation.

Why do Integration Testing?

Although each software module is unit tested, defects still exist for various reasons like

- A Module, in general, is designed by an individual software developer whose understanding and programming logic may differ from other programmers. Integration Testing becomes necessary to verify the software modules work in unity
- At the time of module development, there are wide chances of change in requirements by the clients. These new requirements may not be unit tested and hence system integration Testing becomes necessary.
- Interfaces of the software modules with the database could be erroneous
- External Hardware interfaces, if any, could be erroneous

Example of Integration Test Case

Integration Test Case differs from other test cases in the sense it focuses mainly on the interfaces & flow of data/information between the modules**.** Here priority is to be given for the integrating links rather than the unit functions which are already tested.

Sample Integration Test Cases for the following scenario: Application has 3 modules say 'Login Page', 'Mailbox' and 'Delete emails' and each of them is integrated logically.

Here do not concentrate much on the Login Page testing as it's already been done in Unit Testing. But check how it's linked to the Mail Box Page.

Similarly Mail Box: Check its integration to the Delete Mails Module.

Test Case ID	Test Case Objective	Test Case Description	Expected Result
1	Check the interface link between the Login and Mailbox module	Enter login credentials and click on the Login button	To be directed to the Mail Box
2	Check the interface link between the Mailbox and Delete Mails Module	From Mailbox select the email and click a delete button	Selected email should appear in the Deleted/Trash folder

How to do Integration Testing?

The Integration test procedure irrespective of the Software testing strategies (discussed above):

1. Prepare the Integration Tests Plan
2. Design the Test Scenarios, Cases, and Scripts.
3. Executing the test Cases followed by reporting the defects.
4. Tracking & re-testing the defects.
5. Steps 3 and 4 are repeated until the completion of Integration is successful.

Test Cases This section includes test cases of all the available functions in web side of Business Card.

a) Unit Testing These are unit testing between developer and Supervisor

b) Test Cases for Web Users These are the test cases for web users.

System Testing

System testing is testing conducted on a complete, integrated system to evaluate its compliance with the specified requirements.
After the completion of the integration testing, the product is passed for system testing. System testing is undertaken by independent testers who haven't played a role in developing the program. This testing is performed in an environment that closely mirrors production. System Testing is very important because it verifies that the application meets the technical, functional, and business requirements that were set by the stakeholder.

In our example, we can perform system testing when all the modules are developed and passed integration successfully. For example, the complete product may include features like leave application, reports, employee details, performance tracker, etc.

User Acceptance Testing

User acceptance testing (UAT) is the last phase of the software testing process. In UAT actual software/app users test the software to make sure it can handle required tasks in real-world scenarios. Generally, it is performed at the time of product delivery to stakeholders as a final checkpoint among all functional testing types.

From starting to deployment, the software/app undergoes through various types of testing by testing team and developers. The end goal of all the efforts is to deliver a working software/app that fulfills users' requirements and the client's expectations. Both the teams become so familiar with the application that they might become a victim of tunnel vision. They are fully aware of workarounds and may skip certain scenarios which might be critical for end users.

The users are naive about how the application works. They are focused on 'How the application should behave?' They use the application with a fresh mind and whether it is intuitive to follow or not. UAT is based on user stories and establishes how well it meets their requirements. Users do not use 'Test to Break' approach while doing user acceptance testing. Rather, UAT is a measure of how good your application performs in normal scenarios.

1 Unit Testing (Supervisor & Developer)

Index	Title	Description	Tester Name	Developer Name
1	Customer Sign Up & Log In	Customer sign up and login I have made by using reference from my pass web programing class assignment. This sign up and login connection I showed to my supervisor on submission of milestone 2, it works as expected.	No problem it works fine.	No problem
2	Forms	This VISITING CARD website contains up to 7 form, for different functions, Form List: 1) Sign Up & Login 2) Order Form 3) Payment Form 4) Card Update Options 5) Transaction History Details Form 6) Admin Approval form 7) Contact Details	Validate the forms, and put every single details in the form to make user understand what they should take note. For example, below phone number column put small note to inform user that it should be remembered to proceed with payment process.	I have done validation for some columns, but the details I have put perfectly on all the forms.
3	Payment Process	This payment process I was done differently on the beginning, the way is customer have to pay first before proceed with order. Then after milestone 2, my supervisor give me some idea and reference webpages that buying goods via online, to improve my payment method. The improvised version is customer can order their goods first then they need to pay the amount of their order then put the	Make changes on the payment method, do some research on how can make people do it easily.	New payment method much more efficient than before.

		transaction number to finalise the order for delivery.		
4	Admin Approval	Don’t make customer think they are not safe, provide details such as in progress to show them the order is on process.	Don’t make customer think they are not safe, provide details such as in progress to show them the order is on process.	The approval details is important because, as a customer who make the payment will need notification on every step we have done.

Profile :

Index	Test Case	Test Data	State	Test Input Values	Expected Result
1.	Contact No should accept only 0-9 values	Characters except 0-9 numbers	Invalid	‘sfr’	Enter numeric values only
		Numbers between 0-9	Valid	7030998907	Input is accepted
2	Username accept only string	Characters except	valid	‘String’	Input values is accepted
		Numbers not Accept	Invalid	79793709	Enter string values
3	Name	Numbers not Accept	Invalid	priti79	Enter String values
		Character except	Valid	‘hello’	Input is accepted
4	Age	Characters except	Invalid	‘hi’	Enter numerical values only
		Numbers accepts	valid	22	Input is accepted
5	Birthday	Characters except	Invalid	‘birthday’	Enter numerical values only
		Numbers accepts	valid	22-05-1998	Input is accepted

6	Email	Character and numbers or special character is accepted	valid	‘hello27@gmail.com’	Input values accepted
7	Zipcode	Characters except	Invalid	‘hello’	Enter numerical values only
		Numbers accepts	valid	400120	Input is accepted

Add card:

Index	Test Case	Test Data	State	Test Input Values	Expected Result
1.	The Textbox Should contain only A-Z / a-z	Number or any other Special characters	Invalid	‘1238as’	This textbox only accept alphabetical characters
		Alphabets between A-Z & a-z	Valid	shreya	Input is accepted
2	The textbox should content only 2 digit	Number	valid	‘1238’	Input is accepted
		Alphabets between A-Z & a-z	Invalid	shreya	This textbox only accept alphabetical characters
3	Img url	Accept only Image	valid	IMG	Only accept IMG

Drawbacks and Limitations

Though the software presents a broad range of options to its users some intricate options could not be covered into it; partly because of logistic and partly due to lack of sophistication. Time was also a major constraint this it was not possible to make the software fool proof and dynamic.

Considerable efforts have made the software easy to operate even for people who are not great with computers but it can be a little overwhelming at the first instance

The user is provided help at each step for his convenience in working with the software

List of drawbacks and limitations:

1) Website Design This website was design using basic html coding and CSS, so that it looks very simple. Most of the websites now using WordPress and also many another sites to develop a website.

2) Picture Upload In my system when customer order their card they can only upload their picture once on the time they order the card, they cannot update their picture on card update details. Nowadays people are more attached to many accounts that they can update the details easily include their profile pictures.

3) Payment Method Payment method in this system was inspired by small business via online, basically what people promote and sell using Facebook. For example people who buying goods online is contact the person and bank in the amount to purchase anything. My system also using same method for example after customer made their order they may transfer the amount to my account or cash deposit to my account. This way of buying process might be not efficient for customer.

4) Card design In my system customer can choose their card designs to be printed. Currently I have provided 40 designs for the customer, it is very limited for customers to choose. Every customer has their own design to make their card, so that they need have privileges to edit their own design.

Proposed Enhancements

In short ,it can be summarised that the future scope of the project and the enhancements to be done maybe some of the following:

- Host the software as a cloud based including more facilities
- Integrate more functionality
- Reduce overload of database queries
- Implement backup mechanism on regular basis

1) Website Design The design of the website will upgraded into new themes and background, which makes customer catchier so that there will be many site visitor and make them buy the product. In page navigation set perfectly, to make customer easily navigate to the subpages easily

2) Picture Upload Change the card update process such as Facebook profile creating, so that customer will be able to easily update any of their information privately including their profile pictures and so on. Customer will able to owned the own profile page to share with their clients using NFC Business Card.

3) Payment Method Nowadays people who are buying goods via online are very busy, and that’s the reason the buy via online, so that in future my website will no longer need for customer cash deposit and online transfer transaction number to make their payment, they can just pay using plug-in of iPay88, it is the system that deduct money directly from customer bank account, example Digi Prepaid card buying via online.

4) Card design People nowadays need to have their own space to design their card, even people who printing paper business card also giving the printing service company many designs, in future in my system customer can design their card using their own picture and backgrounds, and they can also choose material of the card.

The above mentioned points are Enhancements which can be done to increase the applicability and the usage of this project.

Conclusions

Nowadays business people who using business cards are having some problems, whereby they need to find bring a lot of business cards together when they go to meet their clients, but even though they put as much information in the card, they cannot put more than 6 lines of info. Some people meets many of their clients for example marketers they provide many cards and print the card every 6 months, this makes waste of money. The objective of this system is to reduce use of paper business card, and make this Visiting Card reach people who using paper business card. Maintaining customer data in database which used to retrieve using Visiting Card. Reporting of card sales report of customer buying this Visiting Card in the database. Create a website to promote and sell Visiting Card. Write data to the Visiting Card with certain hardware. This report give readers understanding about how the system should work and how it should looks like. This report combine introduction, analysis and system requirements, design, implementation and testing and result.

Bibliography

[1] *javascript*. JavaScript tutorial. (n.d.). https://www.w3schools.com/js/default.asp.

[2]http://dspace.library.daffodilvarsity.edu.bd:8080/bitstream/handle/20.500.11948/3254/131-15- 2188.pdf?sequence=1&isAllowed=y

[3] HTML Tutorial. 2021. *W3schools.com*. https://www.w3schools.com/html/default.asp, August 9, 2021.

[4] Internship Final Report. 2021. *Slideshare.net*. https://www.slideshare.net/NadiaIIT/intern-report-final, August 9, 2021.

[5] Vtech Soft IT Services. 2021. *Vtechsoftitservices.com*. http://vtechsoftitservices.com, August 9, 2021.

[6]phpMyAdmin-Wikipedia.2021. *En.wikipedia.org*. https://en.wikipedia.org/wiki/PhpMyAdmin, August 9, 2021.

[7] JavaScript - Wikipedia. 2021. *En.wikipedia.org*. https://en.wikipedia.org/wiki/JavaScript, August 9, 2021.

[8]CSS-Wikipedia.2021. *En.wikipedia.org*. https://en.wikipedia.org/wiki/Cascading_Style_Sheets, August 9, 2021.

[9] Bootstrap (front-end framework) - Wikipedia. 2021. *En.wikipedia.org*. https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework), August 9, 2021.

[10] HTML5 - Wikipedia. 2021. *En.wikipedia.org*. https://en.wikipedia.org/wiki/HTML5, August 9, 2021.

THANK

YOU