



Daffodil
International
University

Online House Servant Management System

BY

Qurratul Aine Tania

ID: 161-35-1464

This Project report has been submitted in fulfillment of the requirements
for the degree of Bachelor of Science in Software Engineering.

Department of Software Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

Fall – 2019



Daffodil International University
Department of Software Engineering, FSIT

SWE-431 Project / Thesis

Project Documentation
Information sharing System

Supervised by:

Dr. Md. Asraf Ali
Associated professor
Department of Software Engineering
Daffodil International University

Submitted by:

Qurratul Aine Tania
161-35-1464
Department of Software Engineering

Daffodil International University Approval

DECLARATION

It hereby declares that this project has been completed by me under the supervision of Dr. Md. Asraf Ali, Associated professor, Department of Software Engineering, Daffodil International University. It is also declared that neither this work nor any part of this has been submitted elsewhere for award of any degree by me.

tanial

Qurratul Aine Tania

Student ID: 161-35-1464

Batch: 19th

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Certified by:



Dr. Md. Asraf Ali

Associated professor

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

ACKNOWLEDGEMENT

First of all I would like to express my cordial gratefulness to Almighty ALLAH for HIS Kindness, for which thing I successfully completed my project.

While working on this project I have received many invaluable help from a large number of people. I would like to take this opportunity to express my deepest gratitude to everyone who helped us.

I felt grateful to express my boundless honor and respect to my supervisor, DR.MD.ASRAF ALI, Associated professor, Department of Software Engineering, Daffodil International University for her deep knowledge and keen interest in the field of software development that influenced me to carry out of this project. Her endless patient helps, friendly support, which have guided me throughout our work and showed the path of achievement.

ABSTRACT

This project is a web based system for booking servant online. Now-a-days people face problem of finding servant. This system help them to get about the details of servants. People can search servant details in this website. People who are busy with their works and they do not get time of searching servant, so for them this system is very useful. One can easily get servant information from this system. Admin can insert, delete, update servant information and manage information.

Contents

Daffodil International University	i
Chapter 1: Introduction	1
1.1 Project overview	2
1.2 Project purpose.....	2
1.2.1 Background of the project	2
1.2.2 Benefits & Beneficiaries	3
1.3 User module:	3
1.4 Stakeholders	4
1.5 Proposed system model.....	4
1.6 Schedule.....	5
1.6.1 Gantt chart	6
1.6.2 Release Plan/Milestone	7
1.7 Software Deliverable	7
1.8 Project Scope	8
Chapter 2.....	9
Software Requirement Specification	9
2.1 Functional Requirement.....	10
2.1.1 Registration	10
2.1.2 Login.....	10
2.1.3 Search Servant	11
2.1.4 Servant Information	11
2.1.5 Manage servant Information	12
2.2 Non Functional Requirement	12
2.2.1 Security	12
2.2.2 Availability.....	12
2.2.3 Usability.....	12
2.3 Performance Requirement.....	13
2.3.1 Speed and latency requirements	13

2.3.2 Precision and accuracy requirements	13
2.3.3 Capacity Requirements	14
2.4 Dependability requirements	14
2.4.1 Reliability and Availability	15
2.4.2 Robust and fault tolerance requirements.....	15
2.4.3 Safety critical requirements.....	16
2.5 Security requirements	16
2.5.1 Access requirements.....	16
2.5.2 Integrity requirements.....	17
2.5.3 Privacy requirements	17
2.6 Look and Feel	17
2.6.1 Appearance Requirements	18
2.6.2 Style Requirements	18
2.7 Operational and Environmental Requirements	19
2.7.1 Expected Physical Requirements	19
2.7.2 Requirement for Interfacing with Adjacent System	19
There is no specific interfacing with adjacent system requirements.	19
2.7.3 Release Requirements	19
Chapter 3.....	20
System Analysis.....	20
3.1 Introduction.....	21
3.2 Use Case Diagram.....	21
3.3 Use case Description.....	22
3.3.1 Registration.....	22
3.3.2 Login.....	23
3.3.3 Search servant.....	24
3.3.4 Servant Information	25
3.3.5 Update information	26
3.3.6 Manage Servant Information.....	27
3.4 Activity Diagram	28
3.5 System Sequence Diagram.....	29

3.5.1 For Use Case Registration	29
3.5.2 For Use Case login	30
3.5.3 For Use Case search servant	31
3.5.4 Servant information sequence diagram	32
3.5.5 For Use Case Update information.....	33
3.5.6 For Use Case Manage servant Information	34
3.6 Data Flow Diagram (DFD)	35
3.6.1 Context Diagram (Level-0 DFD).....	35
3.6.2 Level-1	36
Chapter 4.....	37
System Design Specification	37
4.1 Class Responsibilities Collaboration	38
4.1.1 Registration class	38
4.1.2 User Class	38
4.1.3 Admin Class	39
4.1.4 Servant Information Class	39
4.1.5 Servant Class	39
4.1.6 Location Class.....	40
4.2 Class Diagram.....	40
4.3 Entity Relationship Diagram.....	41
4.4 Development Tools and Technology	42
4.4.1 User Interface and Technology	42
4.4.2 Implementation Tools and Platform.....	42
Chapter 5.....	43
System Testing.....	43
5.1 Importance of Testing	44
5.2 Testing Feature.....	44
5.3 Testing Strategies.....	44
5.4 Testing Environment (hardware/software requirements).....	45
5.5 Test Cases	45
5.5.1 Test Case Module-1 Registration	45

5.5.2 Test Case Module-2 Login	47
5.5.3 Test Case Module-3 Search servant.....	49
Chapter 6.....	51
User Manual.....	51
6.1 Login Page	52
6.2 Registration Page	53
6.3 Home Page	54
6.4 Admin Login Page	55
6.5 Servant Info page	56
6.6 Admin Dashboard	57
6.7 User Dashboard.....	58
6.8 Booking Type.....	59
Chapter 7.....	60
Conclusion	60
7.1 Critical Evolution.....	61
7.2 Limitation.....	61
7.3 Future Scope	61
7.4 Final Synopsis.....	61

Chapter 1: Introduction

Introduction:

Online House Servant Business is a small management business in Dhaka city. The purpose may to be highlight booking a servant for house purpose. Anyone can easily find a servant and pay with the website. The website have some data of servant which are collected from the servant management.

1.1 Project overview

Thousands of web system released today to make peoples work easier. **“Online house servant Management System” is a web platform where we can booking a servant easily.”**

In Dhaka city about 10k people lived and they facing problem about house servant every moments. So that people don't know which process they can solve their problems. My project can be help for them.

1.2 Project purpose

1.2.1 Background of the project

The main objective of this document is to illustrate the requirement of the project . The document gives the detailed description of the both functional and non-functional requirements proposed by the client. The purpose of this project is to provide a friendly environment to maintain the details of servants. The main purpose of this project is to maintain search servant and book servant.

1.2.2 Benefits & Beneficiaries

- Save time
- Easy to use
- Friendly user interface
- Updated servant info

1.3 User module:

- Registration
- Log in
- Search servant
- Show servant information
- Book servant

Admin module:

- Manage servant info
- Edit and update function

1.4 Stakeholders

There are many members associate with this project. They have helped to develop the system directly or indirectly.

Internal stakeholders

- User
- Admin

1.5 Proposed system model

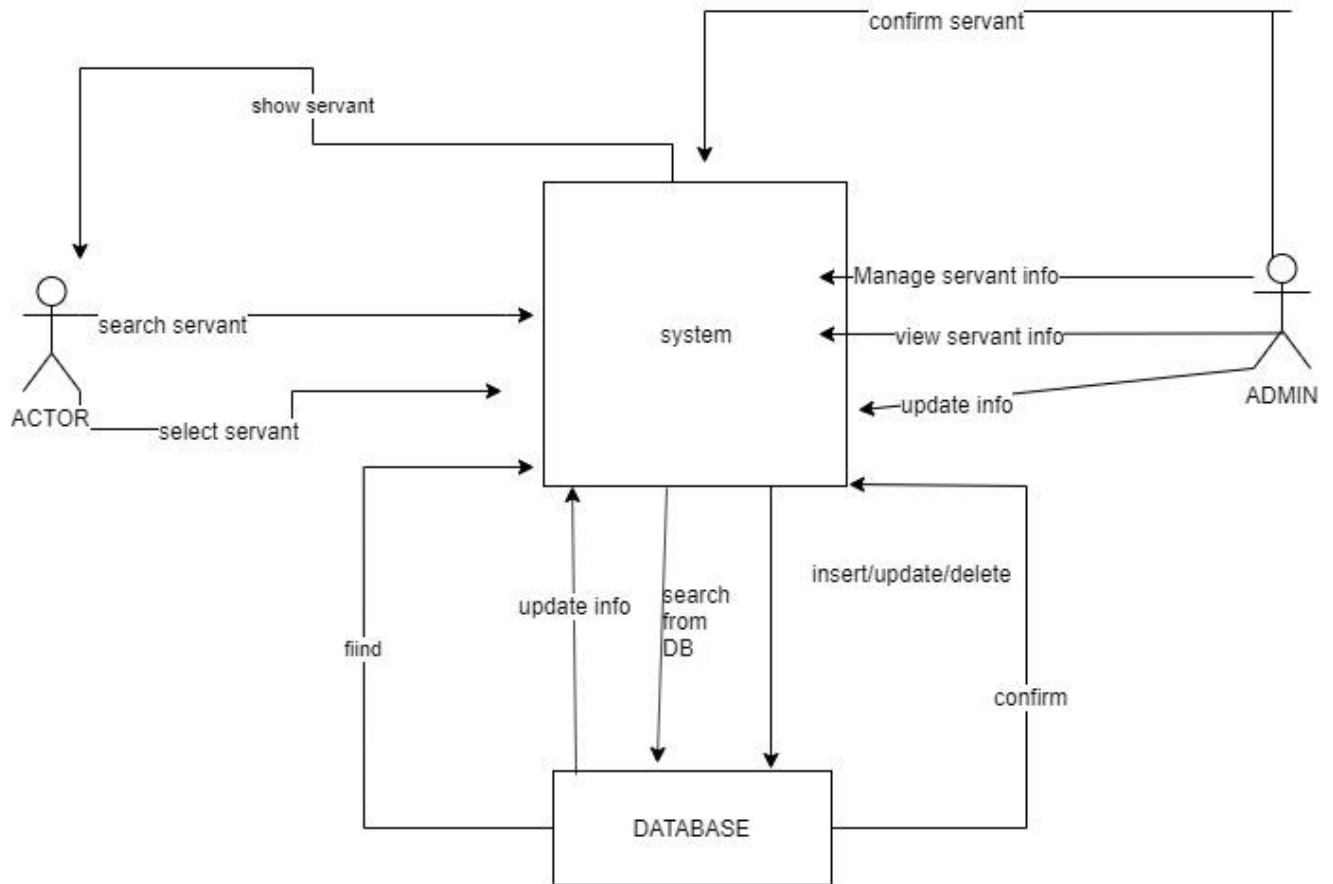


Diagram 1: Propose System

1.6 Schedule

For developing project or something else, schedule helps for proper planning. I also make a schedule for developing and executing my project properly.

1.6.1 Gantt chart

Activities	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16
Planning	█	█	█													
Requirements		█	█	█												
QA-1				█												
System Design				█	█	█	█									
Implementation-1				█	█	█	█	█	█							
QA-2								█	█	█						
Implementation-2										█	█	█	█	█		
Testing													█	█	█	
Delivery															█	█

1.6.2 Release Plan/Milestone

Milestones, a timeline of a project, will clarify the task. This project milestones are as follows:

Task No	Task Name	Time
01	Requirements gathering and analysis	2 week
02	Project Proposal	1 week
03	Database Design	2 Week
04	UI Design	2 Week
05	Implementation	3 Week
06	Testing	2 week
07	Evaluation	1 Week

1.7 Software Deliverable

The main deliverables of this project is the complete system and the system document that will include

- Software requirements and specification
- Software design document

1.8 Project Scope

Online house servant management is an application that is basically updating the search system into an internet-based application so that the user can know the details of the servant information like name,id,phone number,gender,& etc.

The project is specifically designed for the use of the users. The project will work as a complete user interface for search servant and listing the information details for ordinary users.

The project can be easily implemented under various situations. We can add new features as and when we require, making reusability possible as there is flexibility in all the modules. The language used for developing the project is Laravel as it is quite advantageous than other language.

Chapter 2

Software Requirement Specification

2.1 Functional Requirement

The whole project has to develop on the basis of the following requirements

2.1.1 Registration

FR-01	Registration
Description	This system allows to user to registration the system using name, email, address, phone number, password and become a valid user.
Stakeholders	User

2.1.2 Login

FR-02	Login
Description	System allows to user to login the system to access the relevant functions according to the user role. To login to the system user can search vehicle and show area update.
Stakeholders	User

2.1.3 Search Servant

FR-03	Search Servant
Description	System allows to a valid user to search servant by route and show the details of servant and fare for the specific route.
Stakeholders	User, Admin

2.1.4 Servant Information

FR-04	Servant Information
Description	System allows the user to show the details of servant information when user log into the system and request to show servant information.
Stakeholders	User

2.1.5 Manage servant Information

FR-05	Manage servant Information
Description	System allows admin to insert update delete the servant information from the system and database.
Stakeholders	Admin

2.2 Non Functional Requirement

2.2.1 Security

Personal details and admin, login id & password, must be remain confidential.

2.2.2 Availability

The website is available for 24 hours a day. It is always available for stakeholder.

2.2.3 Usability

It has good graphical user interface. It is user friendly. So user feels easy to use.

2.3 Performance Requirement

It's very important to maintain the performance of the project. To ensure a good performance, this project have to meet some requirements which will ensure a good performance.

2.3.1 Speed and latency requirements

While inserting or viewing the system in the mobile application, system need a minimum amount of speed to perform the task.

SLR-01	The system will be faster
Description	While the user and admin run the application then the application will be up. It also depends on internet connection.
Stakeholders	User

2.3.2 Precision and accuracy requirements

System have to ensure the precision and accuracy of the data

SLR-02	Data accuracy
Description	Data should have to accurate at the time of

	inputting. If the data will not accurate system will not allow to save the data. Like registration, login when user run the application first time they must register with valid data input. Admin insert vehicle data is correctly otherwise system will not allow to save.
Stakeholders	User, Admin

2.3.3 Capacity Requirements

System is able to manage all the inserting data of audit person

SLR-03	The system will manage all the inserting data in database.
Description	User details, vehicle details, area information will be stored in the database.
Stakeholders	Admin

2.4 Dependability requirements

By the terms of dependability, it does not mean that this project is totally rely on something.

Here, dependability means the running time of this project.

2.4.1 Reliability and Availability

In order to support global and smooth operations the system must be available around the clock.

On the other hand most services in this system are not mission-critical.

RAR-01	The system must be available 24x7
Description	<ul style="list-style-type: none"><input type="checkbox"/> The system must be available 24 hours in a day<input type="checkbox"/> The system must be updated regularly<input type="checkbox"/> The system must generate report and other things un time

2.4.2 Robust and fault tolerance requirements

In every system, there will have some person for destroying something. System will have to handle this type of person easily.

RFT-01	The system handles over access and system errors
Description	Sometimes multiple user can over access to this system. The system can handle multiple user acces

2.4.3 Safety critical requirements

There are no specific safety critical requirements.

2.5 Security requirements

There are no access requirements beside those that have been outlined in the below: ss

SR-1. Log in as a user

SR-2. Log out as a user

To get access to this system or a specific module the system must provide a central authentication mechanism. In order to prevent anyone to exploit stolen participants all participants password must be encrypted in hash process.

2.5.1 Access requirements

To get access to the system, the system provides authorization/authentication way. This system uses various modules.

AR-1	The system provides security strategies.
Description	The system is designed in way that allows all modules to access a mechanism that provides security services.
Stakeholders	User

2.5.2 Integrity requirements

To protect credentials of user from being stolen, all passwords are stored in encrypted form. The Requirements significantly reduces the value of stolen user credentials, it's not easy to decrypt the password.

2.5.3 Privacy requirements

The system provides a protection of the database in the server. However, the system will have to increment this level of protection because of the personal data made available on the system & the larger share of people that will be having access to it through the system's registration. The user's privacy will be granted by the limited access that the log in process is going to give to the database.

PR-01	All data will be protected
Description	The main requirement in the context is the generation of participant's data for analysis.
Stakeholders	Admin

2.6 Look and Feel

There should not exist any unnecessary things on this project.

2.6.1 Appearance Requirements

It should be clear to admin which fields need to be filled and which can be left blank in this system.

AR-01	Labels of mandatory fields must be bold
Description	Labels of mandatory fields must be bold to identify them as being of mandatory.
Stakeholders	User

2.6.2 Style Requirements

User interface will be mobile application. For styling the interface and making lucrative, I need to use Layout framework, v4 compact library, v4 fragment library, v7 cardview, v7 recyclerview, vector drawable library.

SR-01	The look and feel must be controllable using style sheet.
Description	The styling of the elements of the android application user interface will be defined using layout framework, v4 compact library, v4 fragment library, v7 cardview v7 recyclerview vector drawable library.
Stakeholders	User , Admin

2.7 Operational and Environmental Requirements

Operational and environmental requirements is very important because this project may not work in every environment and its operation may not accurate in every time.

2.7.1 Expected Physical Requirements

There is no specific expected physical requirements.

2.7.2 Requirement for Interfacing with Adjacent System

There is no specific interfacing with adjacent system requirements.

2.7.3 Release Requirements

There are no specific release requirements but in the project schedule section it was described briefly.

Chapter 3

System Analysis

3.1 Introduction

Design is the first step in the development phase for any techniques. Software system design is important thing for an application. If design is not proper then application could be problematic. System should have good design. Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.

3.2 Use Case Diagram

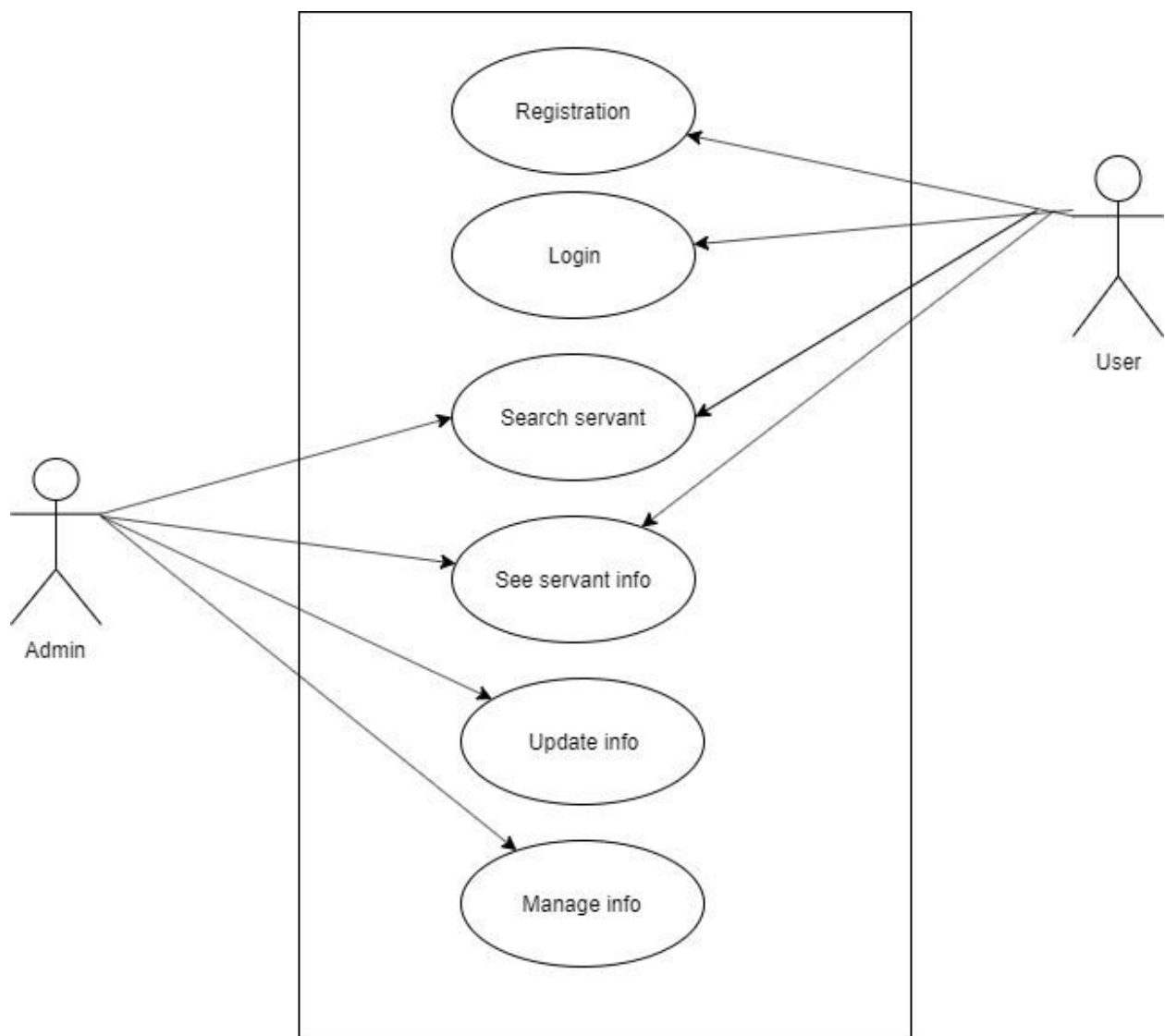


Diagram 2: Use Case

3.3 Use case Description

3.3.1 Registration

Use Case ID	UCID1
Use Case Name	Registration
Created By	Qurratul Aine Tania
Date	2 Dec 2019
Description	This use case allows to user to registration the system using name, email, address, phone number, password and become a valid user.
Primary Actor	User
Secondary Actor	None
Main Flow	Registration is successful
Pre-condition	User provide valid information
Post-condition	The system display the relevant homepage

3.3.2 Login

Use Case ID	UCID2
Use Case Name	Login
Created By	Qurratul Aine Tania
Date	2 Dec 2019
Description	This use case allows to user to login the system to access the relevant functions according to the user role. To login to the system user can search vehicle and show area update.
Primary Actor	User
Secondary Actor	None
Main Flow	Login is successful
Pre-condition	User has to have a valid account
Post-condition	The system display the relevant homepage

3.3.3 Search servant

Use Case ID	UCID3
Use Case Name	Search Servant
Created By	Qurratul Aine Tania
Date	2 Dec 2019
Description	This use case allows to a valid user to search servant by user and show the details of servant and fare for the specific route.
Primary Actor	User
Secondary Actor	None
Main Flow	Show the servant list
Pre-condition	<ol style="list-style-type: none">1. User login to the system2. Search by servant
Post-condition	The system display the specific servant

3.3.4 Servant Information

Use Case ID	UCID4
Use Case Name	Servant Information
Created By	Qurratul Aine Tania
Date	2 Dec 2019
Description	The use case allows the user to show the details of servant information when user log into the system and request to show servant information.
Primary Actor	User
Secondary Actor	Admin
Main Flow	Show the details servant list with fare and route
Pre-condition	<ol style="list-style-type: none">1. User login to the system2. Search by servant
Post-condition	The system display the details of servant

	list .
--	--------

3.3.5 Update information

Use Case ID	UCID5
Use Case Name	Update information
Created By	Qurratul Aine Tania
Date	2 Dec 2019
Description	This use case allows to show the details of servant information when user log into the system and request to see servant information.
Primary Actor	User
Secondary Actor	Admin
Main Flow	Show the update information
Pre-condition	<ol style="list-style-type: none"> 1. User login to the system 2. Show servant information 3. Admin access the system
Post-condition	The system display the relevant area

	update page
--	-------------

3.3.6 Manage Servant Information

Use Case ID	UCID6
Use Case Name	Manage servant Information
Created By	Qurratul Aine Tania
Date	2 Dec 2019
Description	This use case allows admin to insert update delete the servant information from the system and database. This is crudbooster operation.
Primary Actor	Admin
Secondary Actor	None
Main Flow	Crudbooster
Pre-condition	Admin access the system
Post-condition	This system display the dashboard page.

3.4 Activity Diagram

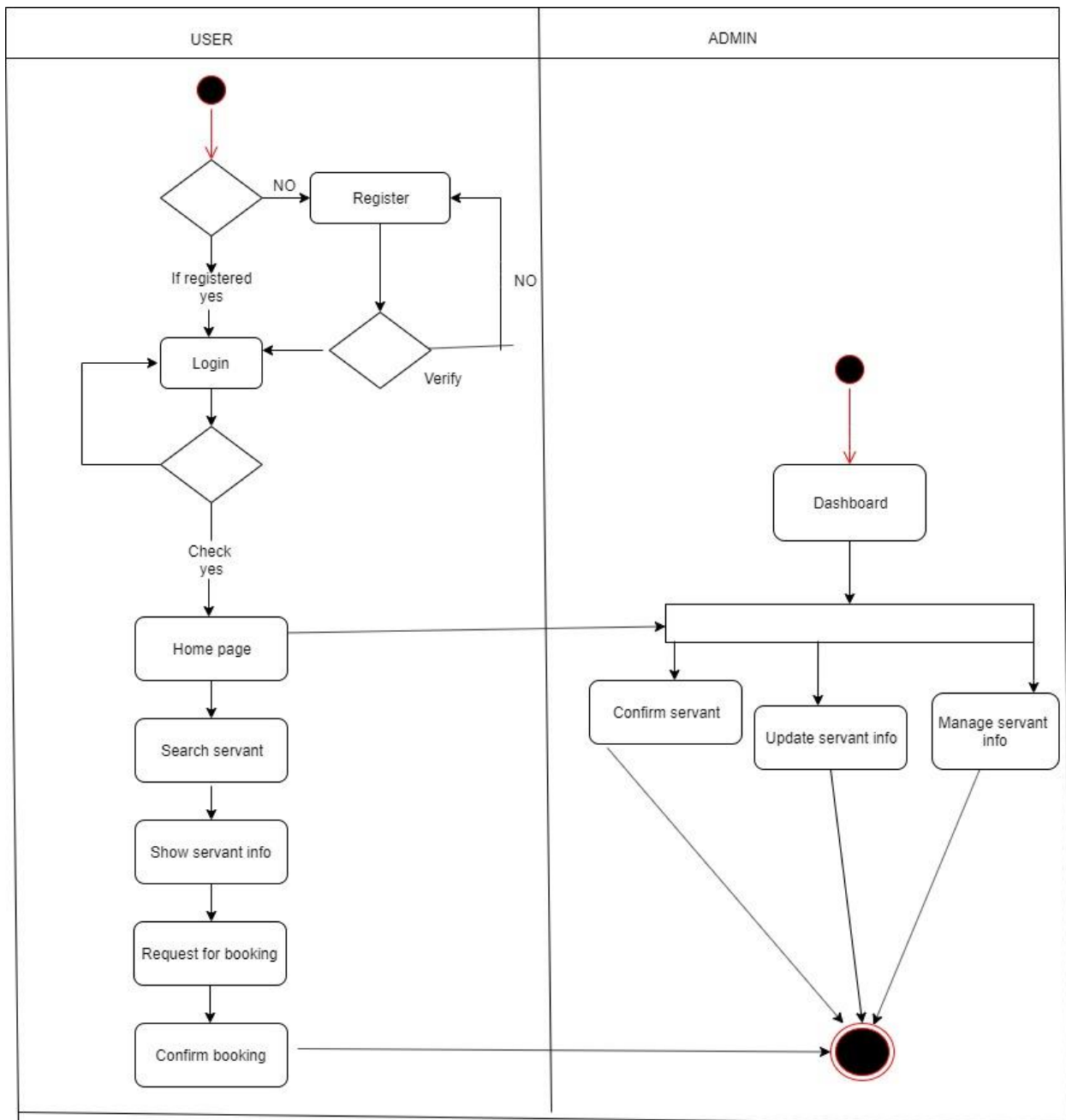


Diagram 3: Activity

3.5 System Sequence Diagram

3.5.1 For Use Case Registration

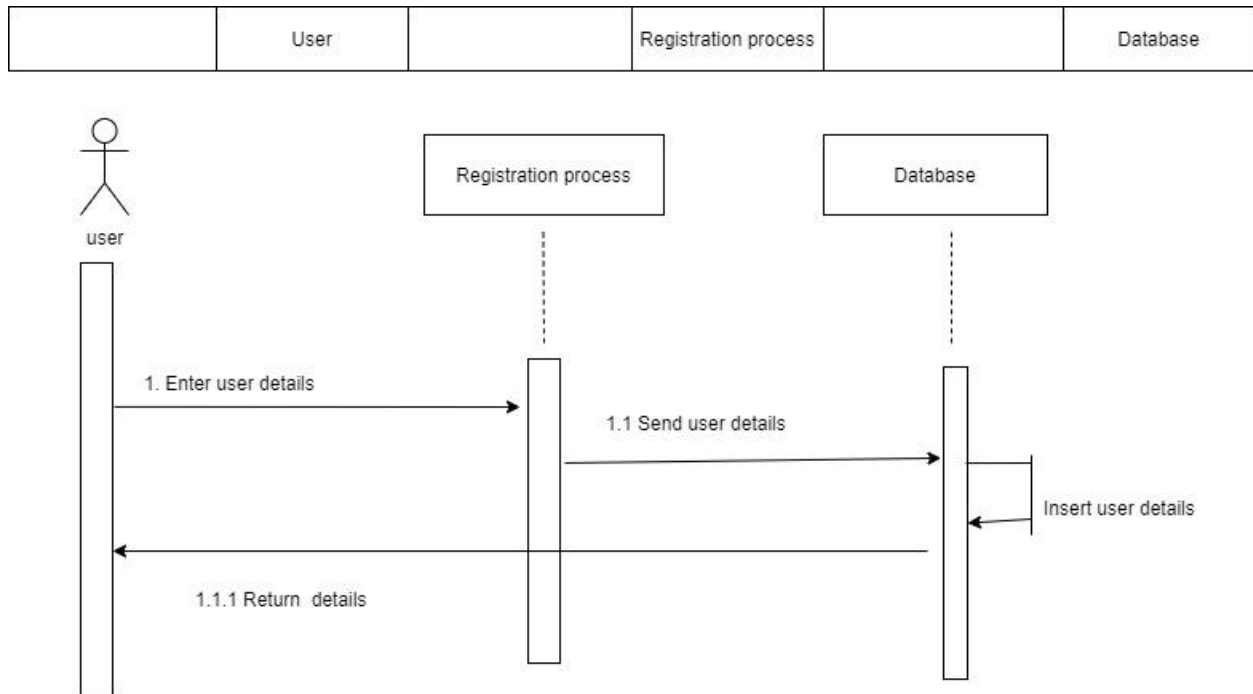


Diagram 4: Registration Sequence Diagram

3.5.2 For Use Case login

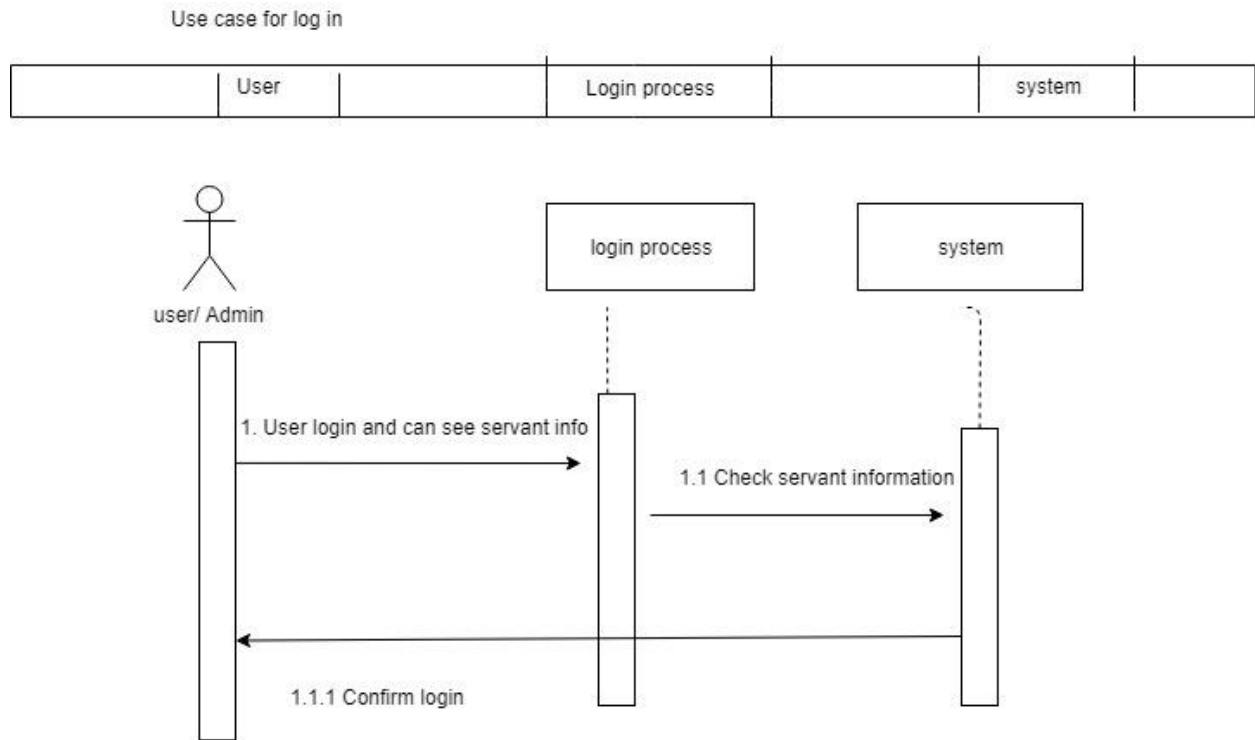


Diagram 5: Login Sequence Diagram

3.5.3 For Use Case search servant

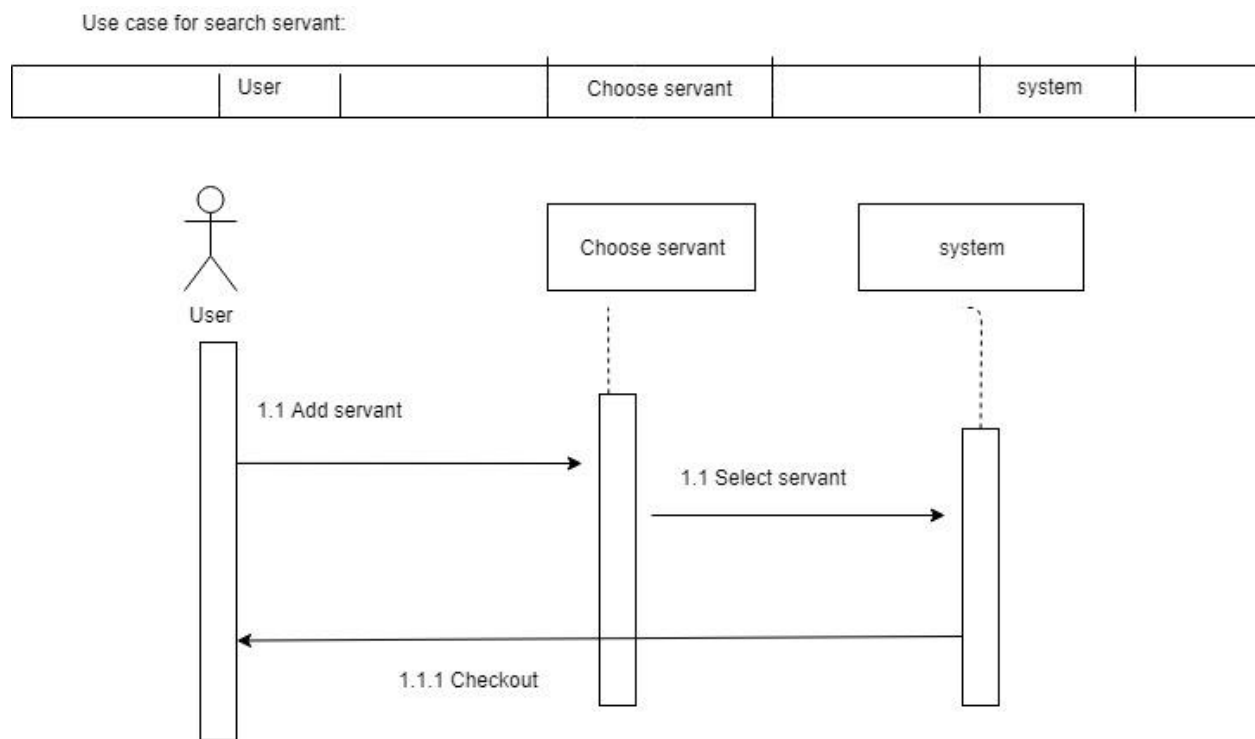


Diagram 6: Search servant Sequence Diagram

3.5.4 Servant information sequence diagram

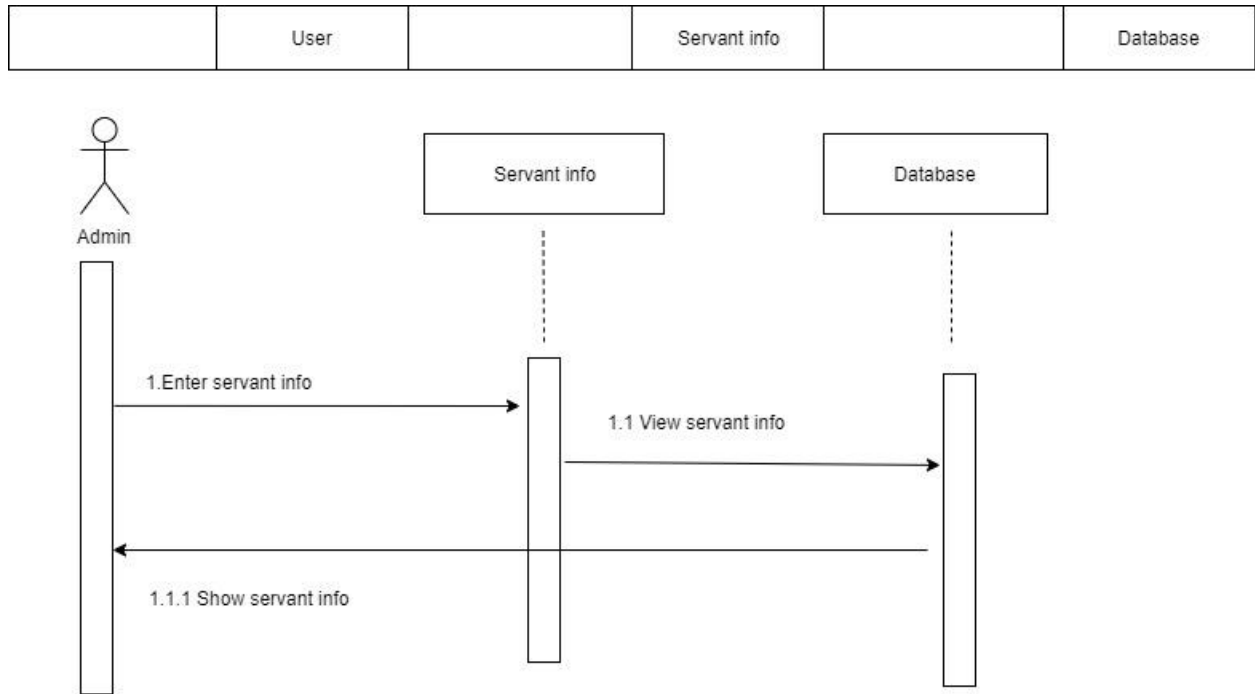


Diagram 7: Servant Information Sequence Diagram

3.5.5 For Use Case Update information

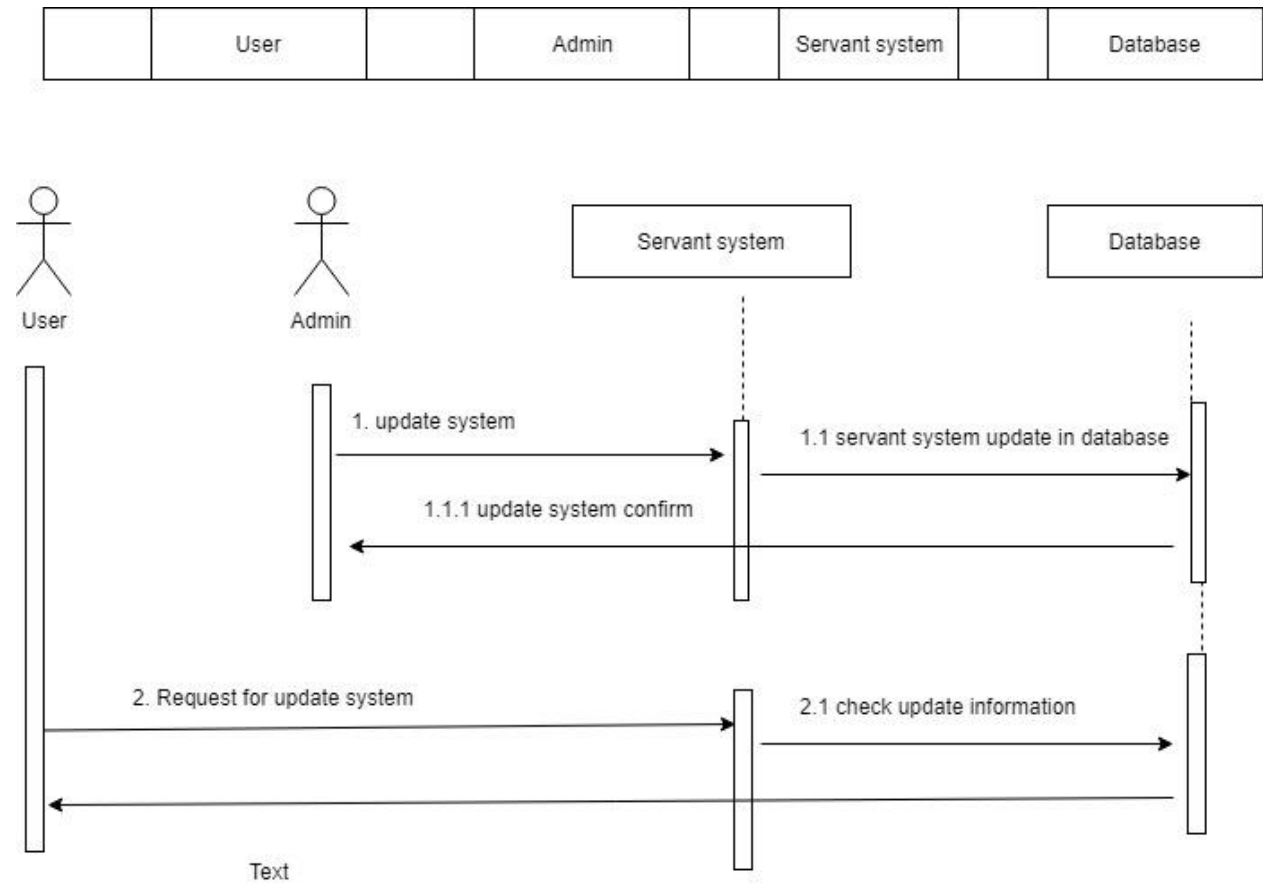


Diagram 8: Update servant information

3.5.6 For Use Case Manage servant Information

User case for manage servant information:

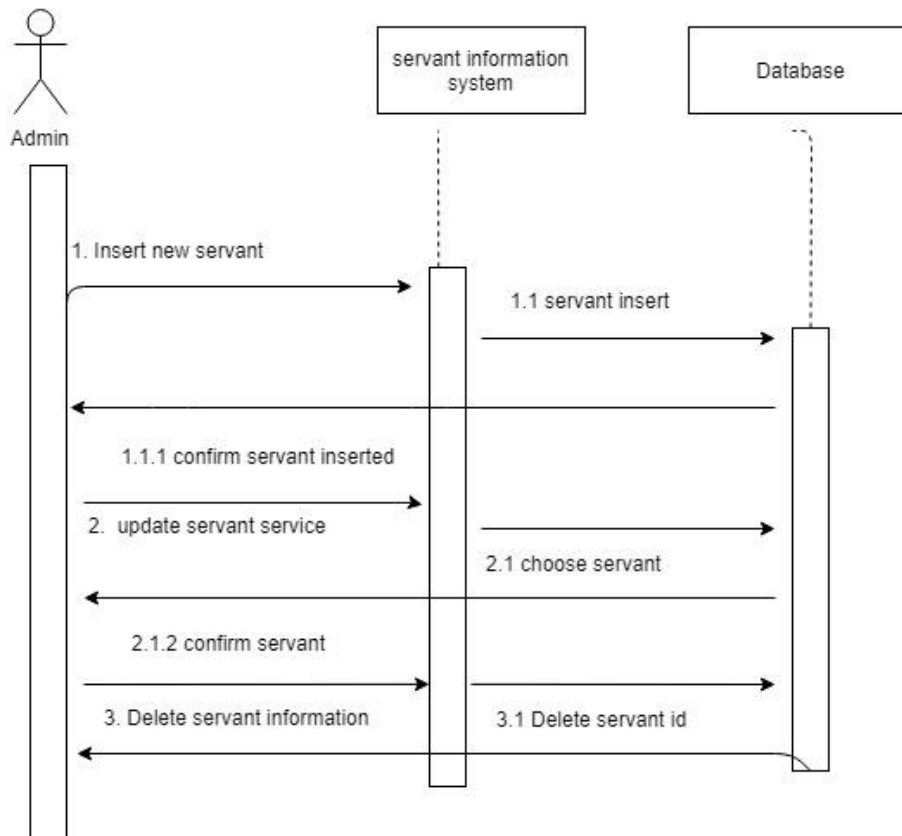
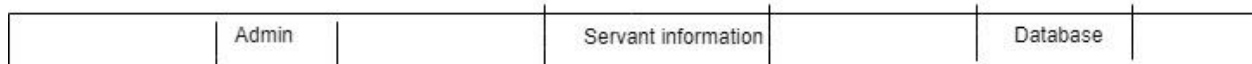


Diagram 9: Manage servant Information sequence Diagram

3.6 Data Flow Diagram (DFD)

3.6.1 Context Diagram (Level-0 DFD)

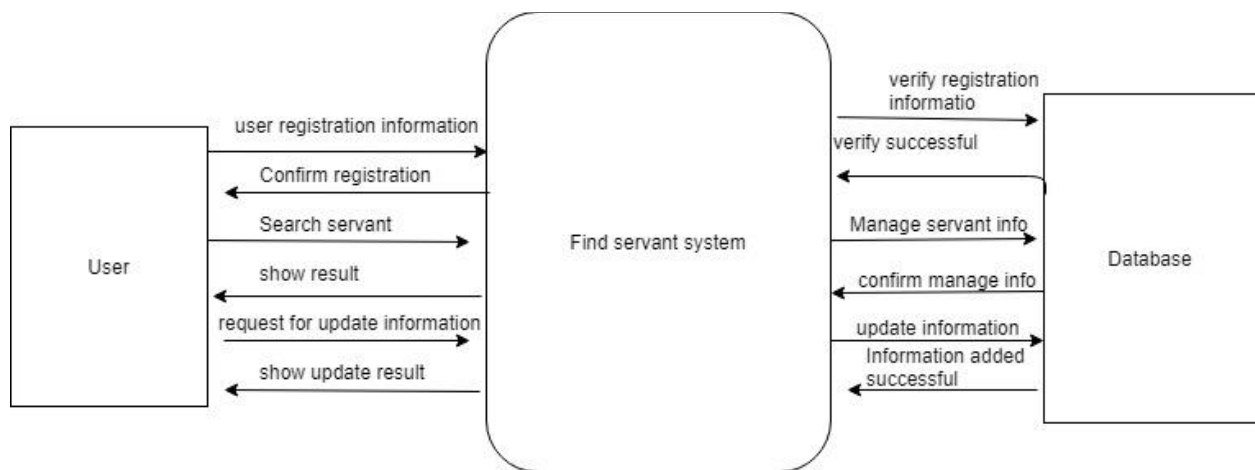


Diagram 10: DFD Level-0

3.6.2 Level-1

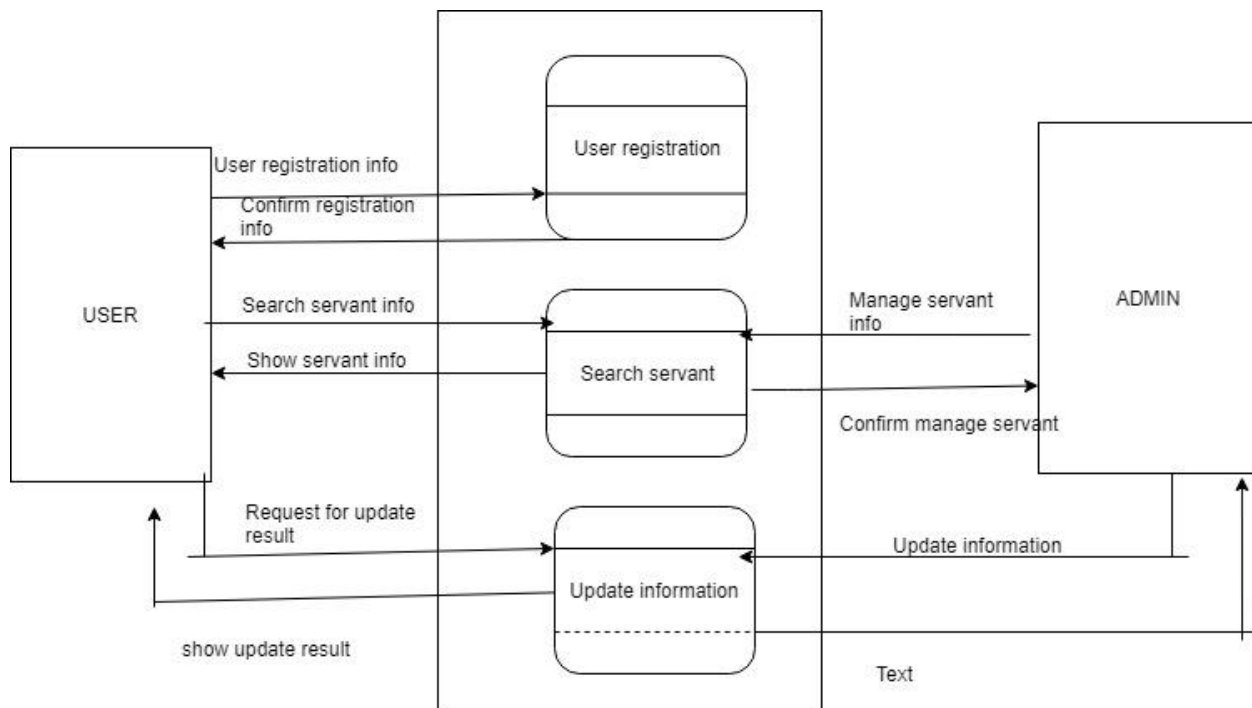


Diagram 11: DFD Level-1

Chapter 4

System Design Specification

4.1 Class Responsibilities Collaboration

4.1.1 Registration class

Registration	
Full Name	USER
User id	
Phone_no	
Email	
Address	
Password	

4.1.2 User Class

Servant class	
Servant name	User
Servant-id	Admin
Gender	

4.1.3 Admin Class

Admin	
Admin_id Admin_name	Servant information Confirmation Route

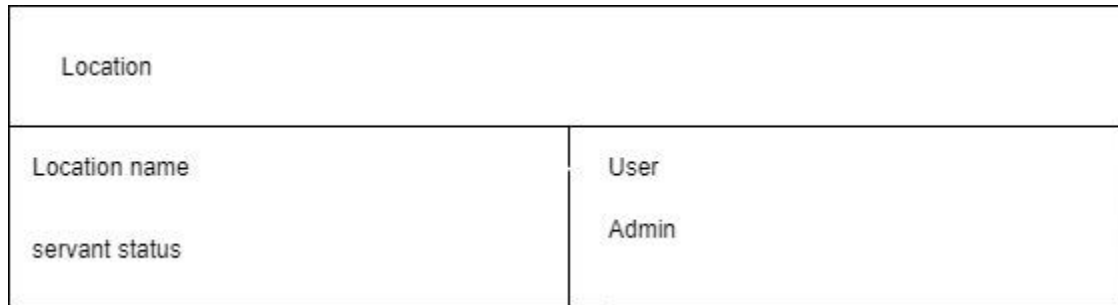
4.1.4 Servant Information Class

Servant information	
Servant_name Servant_categorie Servant_gender	User Admin Confirmation

4.1.5 Servant Class

Servant class	
Servant name Servant-id Gender	User Admin

4.1.6 Location Class



4.2 Class Diagram

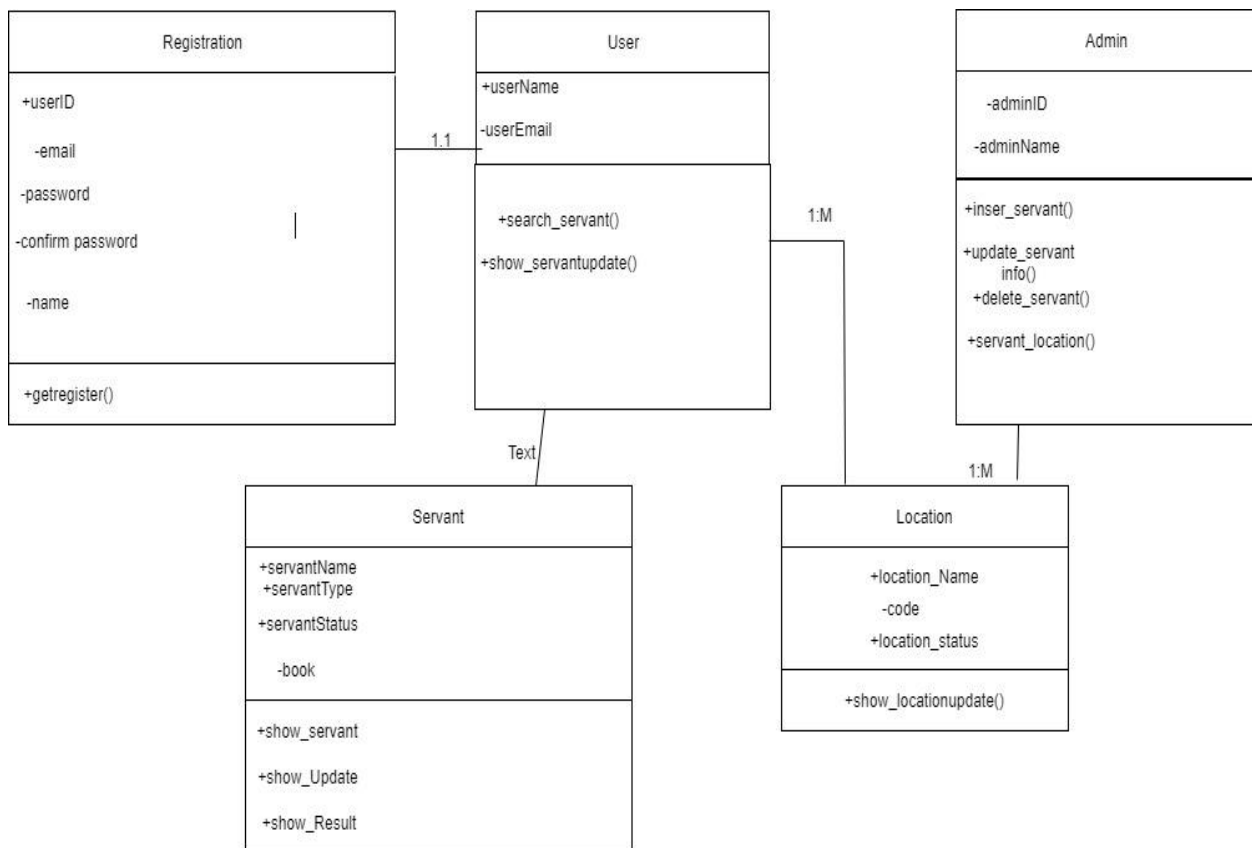


Diagram 8: Class Diagram

4.3 Entity Relationship Diagram

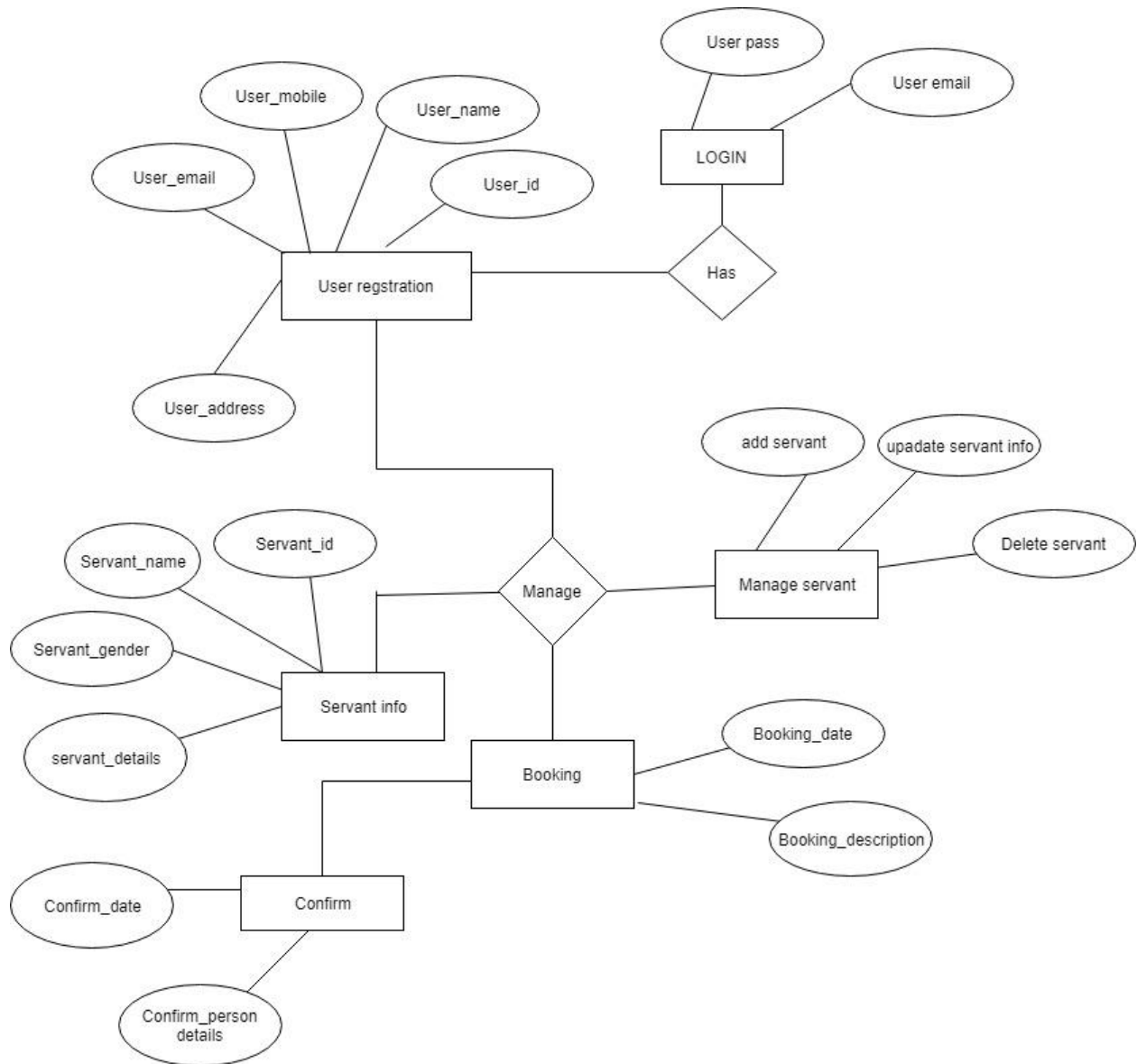


Diagram 11: ER Diagram

4.4 Development Tools and Technology

4.4.1 User Interface and Technology

- Web based design library

4.4.2 Implementation Tools and Platform

Framework: Laravel Framework

Server: apache

Language: PHP

Web Design:HTML,CSS

Database: MySQL

Chapter 5

System Testing

5.1 Importance of Testing

- Testing is find errors on this application program.
- It is used for both validation and verification to develop a product.
- It make sure customer reliability and satisfaction.
- Make sure quality of this application.
- Make sure great performance of this application.
- Required to stay in software business.

5.2 Testing Feature

Feature testing is the process of making changes in software system to add one or more new features or to make modifications in the already existing features. Each of these feature is said to have a characteristics that is designed to be useful, intuitive, and effective.

5.3 Testing Strategies

A testing strategy is a general approach to the testing process rather than a method of devising particular system or component tests. Different testing strategies may be adopted depending on the type of system to be tested and the development process used.

5.4 Testing Environment (hardware/software requirements)

- Test data
- Web Application
- Database: my SQL
- Language: laravel framework & server apache

5.5 Test Cases

5.5.1 Test Case Module-1 Registration

Test Case ID: ATC1	Test Design by: Qurratul Aine Tania
Test Priority: High	Test Design Date: 28/11/2019
Module Name: Registration	Test Execute By: : Qurratul Aine Tania
Description: This section covers the functionality of registering new users.	Test Execute Date: 28/11/2019

Precondition: User has no precondition

Dependencies: Enter input required information

Step	Test Case	Test Data	Expected Result	P/F	Actual Result
01	Enter empty value for any required Field	Display error message	Pass		Display error Message
02	Enter Non-sdjfh		Display error	Pass	Display error

	numeric value in Phone No field		Message “Enter valid Number”		Message “Enter valid Number”
03	Enter different email format instead of xxx@xxx.xx	Gmail.com -ptz.dm	Display error Message “Invalid email address”	Pass	Display error Message “Invalid email address”
04	Enter password less than 6 character long	12345	Password length must be 6 character	Pass	Display error message “Password length must be 6 character”
05	New user registration	NA	Display message “submitted successfully”	Pass	Display message “submitted successfully”

5.5.2 Test Case Module-2 Login

Test Case ID: ATC1	Test Design by: Qurratul Aine Tania
Test Priority: High	Test Design Date:28/11/2019
Module Name: Login	Test Execute By: Qurratul Aine Tania
Description: This section covers the functionality of login register users.	Test Execute Date: 28/11/2019

Precondition: User has no precondition

Dependencies: Enter input required information

Step	Test Case	Test Data	Expected Result	Actual Result	P/F
01	Enter empty value for username and password	NA	Please enter username and password	Please enter username and password	Pass
02	Enter correct value for username and password	User name and password	Accept	Accept"	Pass

03	Enter wrong value for username and password	User name and password	Invalid login Attempt	Invalid login Attempt	Pass
04	After giving input click submit button	NA	Direct on Home Page	Direct on Home Page	Pass

5.5.3 Test Case Module-3 Search servant

Test Case ID: ATC1	Test Design by: Qurratul Aine Tania
Test Priority: High	Test Design Date: 28/11/2019
Module Name: Search servant	Test Execute By: : Qurratul Aine Tania
Description: This section covers the functionality of searching.	Test Execute Date: 28/11/2019

Precondition: User must be registered already.

Dependencies: Input correct user name & password.

Step	Test Case	Test Data	Expected Result	Actual Result	P/F
01	Search by servant	Uttara to Dhanmondi	Servant name,phone, gender and status.	Servant name, fare, type and status.	Pass
02	Enter Empty value	NA	Display message "Please enter input"	Display message "Please enter input"	Pass
03	Enter wrong input	Uwtyw kdjhvsk	Display Message	Display Message	Pass

			“No vehicle found”	“No vehicle found”	
04	Enter out of Dhaka city route	Uttara to Mymanshing	Display Message “Out of Dhaka city”	Display Message “Out of Dhaka city”	Pass

Chapter 6

User Manual

6.1 Login Page



Login

E-Mail Address

Password

Remember Me

[Forgot Your Password?](#)

Figure 1: Login Page

6.2 Registration Page



Register

Name

E-Mail Address

Password

Confirm Password

Figure 2: Registration Page

6.3 Home Page

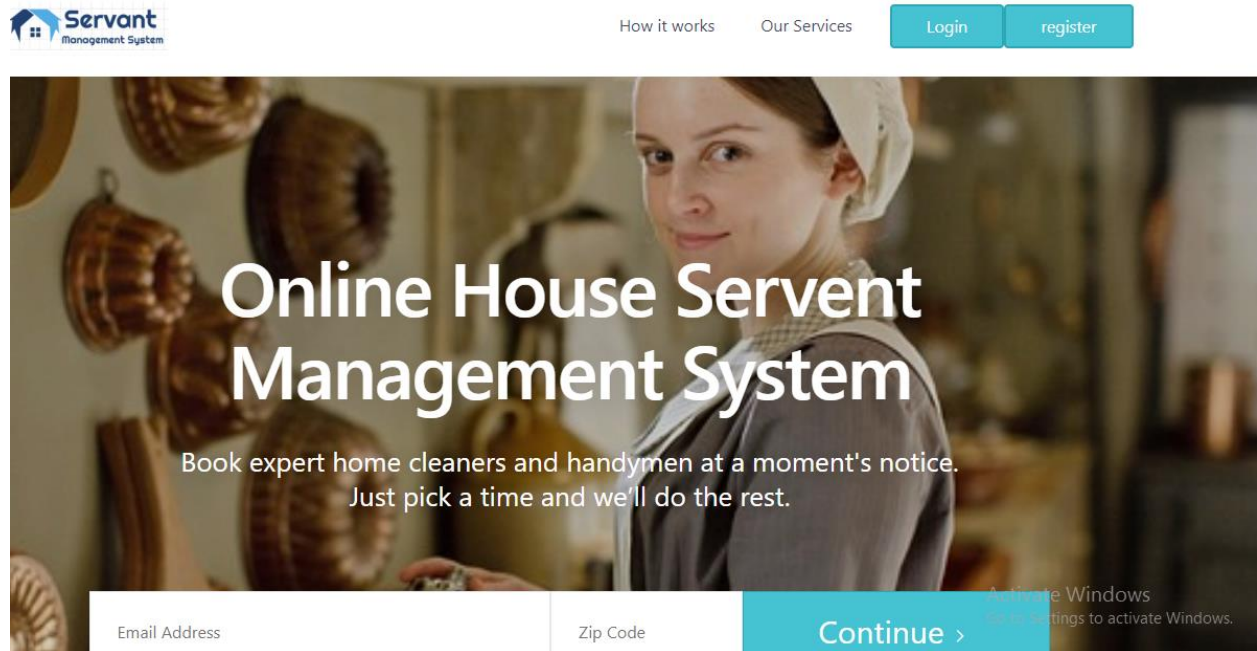
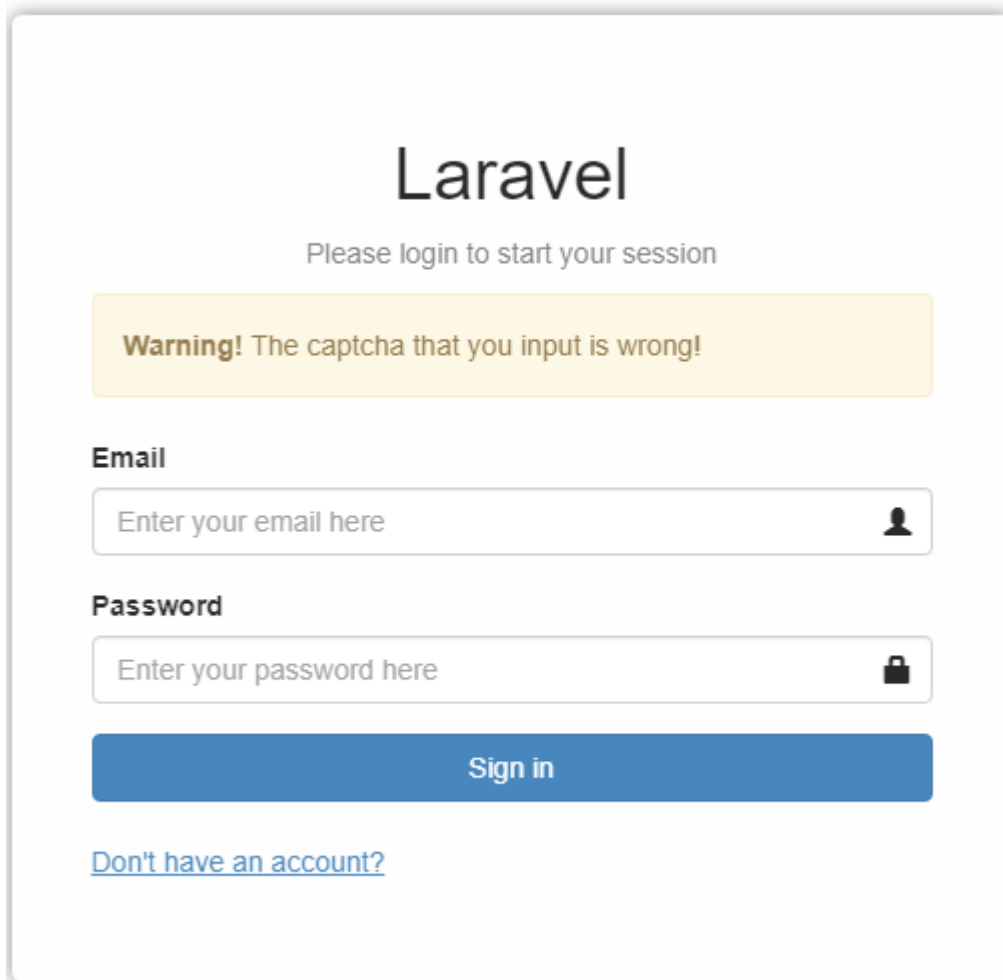


Figure 3: Home Page

6.4 Admin Login Page



The image shows a login form for Laravel. At the top, the word "Laravel" is displayed in a large, bold font. Below it, the text "Please login to start your session" is centered. A yellow warning box contains the message "Warning! The captcha that you input is wrong!". The form includes two input fields: "Email" with a placeholder "Enter your email here" and a user icon, and "Password" with a placeholder "Enter your password here" and a lock icon. A blue "Sign in" button is positioned below the password field. At the bottom, there is a blue link that says "Don't have an account?".

Figure 4: Admin Login

6.5 Servant Info page

The screenshot displays a web application interface for managing servant information. At the top, there is a green header with the text 'Laravel' and a user profile for 'tania'. Below the header, a dark sidebar contains navigation links: 'Menu Navigation', 'Dashboard', 'Sarvent', 'Clean', 'Rate', and 'Booking'. The main content area features a 'Sarvent' header with an 'Add Data' button and a breadcrumb trail 'Home > Sarvent'. A green success message reads 'Success! The data has been added'. Below this is a 'Browse Data' section with a search bar and a dropdown menu set to '10'. The data is presented in a table with the following columns: Name, Phone, Photo, Address, Gender, Experience, Jobtype, and Action. The table contains four rows of servant information. At the bottom right, a watermark 'Activate Windows' is visible along with the text 'Show 1 to 4 total 4'.



































Name	Phone	Photo	Address	Gender	Experience	Jobtype	Action
Mafia jannat	0165436788		uttora,Dhaka	female	2 years	house cleaning	  
Humayun kobir	01675437899		East rajabajar,Dhanmondi	male	7years	cooking	  
janntul islam	0183678993		west rajabajar,dhanmondi	female	none	house cleaning	  
bipasa bosu	017593788		sukrabad,dhanmondi	female	2 years	cooking	  

Figure 5: Servant info

6.6 Admin Dashboard

The screenshot displays an admin dashboard for a cleaning service. The dashboard features a sidebar with navigation options: Dashboard, Sarvent, Clean, Rate, and Booking. The main content area is titled "Browse Data" and contains a table of cleaning jobs. The table has the following columns: Created At, Updated At, Type, Time, Bedroom, Bathroom, Hour, Date, Start Time, User, Approved, Amount, and Action. The table contains 6 rows of data. The first row shows a job created on 2019-12-01, updated on 2019-12-01 at 18:47:31, with a type of STANDARD, time of ONE TIME, 2 bedrooms, 1 bathroom, and 5 hours. The job is approved and has an amount of 2500. The other rows show jobs with various types (DEEP, STANDARD), frequencies (BI-WEEKLY, WEEKLY), and dates (12/20/2019, 12/26/2019, 12/19/2019, 12/11/2019). The table also includes a search bar and a pagination control showing 10 items per page.



















Created At	Updated At	Type	Time	Bedroom	Bathroom	Hour	Date	Start Time	User	Approved	Amount	Action
	2019-12-01 18:47:31	STANDARD	ONE TIME	2	1	5	2020202020202020-MarMonMon	9AM	3	Approved	2500	  
		DEEP	BI-WEEKLY	3	2	01	12/20/2019	5PM	2	Not Approved		  
		DEEP	BI-WEEKLY	1	1	2	12/26/2019	9AM	1	Not Approved		  
		STANDARD	WEEKLY	2	2	2	12/20/2019	7AM	1	Not Approved		  
		STANDARD	WEEKLY	5	3	12	12/19/2019	5PM	1	Not Approved		  
		DEEP	WEEKLY				12/11/2019	7AM	1	Not Approved		  

Show 1 to 6 total 6

Figure 6: Admin Dashboard

6.7 User Dashboard

The screenshot displays a user dashboard for a Laravel application. The top navigation bar is green and contains the 'Laravel' logo, a hamburger menu icon, and a user profile icon labeled 'user'. A dark sidebar on the left provides navigation options: 'Dashboard', 'Clean', and 'Booking'. The main content area is titled 'Clean' and features a 'Browse Data' section with a search bar and a dropdown menu set to '10'. Below this is a table with the following data:

Created At	Updated At	Type	Time	Bedroom	Bathroom	Hour	Date	Start Time	User	Approved	Amount	Action
	2019-12-01 18:47:31	STANDARD	ONE TIME	2	1	5	2020202020202020-MarMar-MonMon	9AM	3	Approved	2500	  
		DEEP	BI-WEEKLY	3	2	01	12/20/2019	5PM	2	Not Approved		  
		DEEP	BI-WEEKLY	1	1	2	12/26/2019	9AM	1	Not Approved		  
		STANDARD	WEEKLY	2	2	2	12/20/2019	7AM	1	Not Approved		  
		STANDARD	WEEKLY	5	3	12	12/19/2019	5PM	1	Not Approved		  
		DEEP	WEEKLY				12/11/2019	7AM	1	Not Approved		  

At the bottom right of the table, it says 'Show 1 to 6 total 6'.

Figure 7: User Dashboard

6.8 Booking Type

Cleanly

Book your cleaning

Its time to book our cleaning service for your home or apartment.

Cleaning Preferences

What type of cleaning?

STANDARD DEEP MOVE IN/OUT

How often would you like cleaning?

ONE TIME WEEKLY BI-WEEKLY MONTHLY

Figure 8: Booking

Chapter 7

Conclusion

7.1 Critical Evolution

If user is not comfortable with website he/she should have problem with this application. User must know English because whole application is in English. He/she should have good knowledge of internet source. User should be registered and then he/she can login the system.

7.2 Limitation

It is very hard to develop something without any limitations. This project has some limitations.

Limitation are as follows

- Not highly secure
- No map facility

7.3 Future Scope

If this website use properly and user can understand its usability It can use for other things like add google map, search nearby places, show route and area in map and others. It can use in daily life . It can improve for other things if it is useful for user and they got help from this.

7.4 Final Synopsis

It has been a great pleasure to work on this project. This project help to know new technology. This project thing is some kind of weird thing but this is helping people. Helping people with developing an application make a better world.

