



Daffodil
International
University

Department of CIS

TITLE OF THE PROJECT

Self Storage

Submitted By

Abul Hasnat Shohag

ID: 181-16-252

Supervised By

Md Sarwar Hossain Mollah

Associate Professor, Head of Department

Department of Computing & Information System (CIS)

Daffodil International University

Course Code: CIS (499)

Section: A

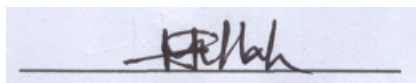
Spring-2022

Submission Date: 23 August 2022

APPROVAL

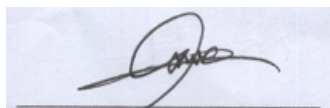
This Project titled **Self Storage**, Submitted by Abul Hsanat Shohag, ID No: 181-16-252 to the Department of Computing & Information Systems, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information Systems and approved as to its style and contents. The presentation has been held on- 23-08-2022.

BOARD OF EXAMINERS



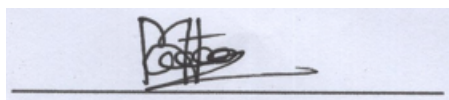
Mr. Md Sarwar Hossain Mollah
Associate Professor and Head
 Department of Computing & Information Systems
 Faculty of Science & Information Technology
 Daffodil International University

Chairman



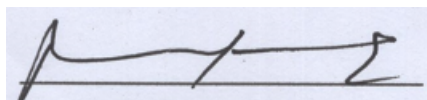
Mr. Abdullah Bin Kasem Bhuiyan
Lecturer
 Department of Computing & Information Systems
 Faculty of Science & Information Technology
 Daffodil International University

Internal Examiner



Mr. Md. Mehedi Hasan
Lecturer
 Department of Computing & Information Systems
 Faculty of Science & Information Technology
 Daffodil International University

Internal Examiner



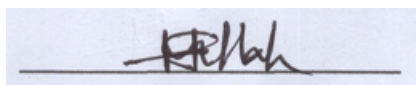
Dr. Saifuddin Md. Tareeq
Professor & Chairman
 Department of Computer Science and Engineering
 University of Dhaka, Dhaka

External Examiner

Declaration

I hereby declare that; this project has been done by me under supervision of **Md Sarwar Hossain Mollah, Associate Professor, Head of Department**, Department of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of there has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

Supervised By



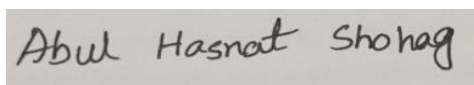
Md Sarwar Hossain Mollah

Associate professor, Head of Department

Department of CIS

Daffodil International University

Submitted By



Name: Abul Hasnat Shohag

ID: 181-16-252

Department of CIS

Daffodil International University

Acknowledgement

I am grateful to the almighty for giving me the strength and capability to complete this project. Also, I want to thank my parents and siblings for their encouragement and support throughout the journey. It would not have been possible without their support. And special thanks to Mr. Md Sarwar Hossain Mollah, Head of CIS department the person who has been a well-wisher and supporter of me from the beginning of my undergrad journey. I know simple thanks won't do justice to the support he provided but I will try to repay the debt throughout the works I will do.

Lastly, I would like to show my gratitude to all the teachers who have taken my classes and taught me lessons to become who I am today. And I really should thank some of my classmates who accepted me (an introverted, socially awkward person) as their friend, without them I would definitely be lost by now.

Dedication

I would like to dedicate this project to my parents and my siblings. I feel like they are the only ones that truly care about what I do and have always supported me no matter what I went through. Without their unconditional support I would not be here.

Executive Summary

Self Storage is an academic project. The project is an ecommerce platform that works with rental of storage units. It allows homeowners to rent out their extra rooms, space to small businesses and persons to use as storage units and warehouses. It also provides warehouse management to the renters, a platform to sell products and rent out the unused tools to others. Here, users can rent rooms to store their belongings and small businesses can store their products. While businesses can take warehouse management services, general users can use the platform to sell/rent their unused belongings to other users. Also, if small businesses want to take services of warehouse management they can take it as well with transportation service all from using this system. It will be beneficial for both general users and small businesses.

Table of Contents

Chapter 1 - Introduction	1
1.1 Introduction.....	1
1.2 Document Contents in Project Document	2
Chapter 2 – Initial Study	4
2.1 Project Proposal	4
2.2 Background of the project	12
2.3 Problem Areas.....	13
2.4 Solutions to the Problems	14
Chapter 3 – Literature Review	14
3.1 Discussion on the Problem Domain	15
3.2 Discussion on the Problem Solution:	15
3.3 Comparison among the leading solutions.....	16
3.4 Recommended Approach	17
Chapter 4 – Methodology	17
4.1 What to Use	17
4.2 Why to Use Methodology	21
4.3 Sections of Methodology.....	22
4.4 Implementation Plans	22
Chapter 5 – Planning	23
5.1 Project Plan.....	23
5.1.1 Work Breakdown Structure	23
5.1.2 Resource Allocation	23
5.1.3 Time Boxing	24
5.1.4 Gantt Chart of Development.....	25
5.2 Test Plan	26
5.2.1 Testing Against the Time Boxes.....	26
5.2.2 Required Test	26
5.2.3 Test Case for Self Storage	27
5.2.4 User Acceptance Test Plan	28
5.3 Risk Management of Self Storage	28

5.3.1 Risk Identification	29
5.3.2 Risk Assessment of Self Storage	31
5.3.3 Risk Precaution / Action Plan	31
5.3.4 Steps Taken for Possible Risks of Self Storage	33
5.4 Change Management	33
5.4.1 Factors that Might Cause Change	33
5.4.2 DSDM Welcome Change of Self Storage	34
5.4.3 Considering Business Priority	34
5.4.4 Change Workshop	34
5.4.5 Changes That are Allowed	35
5.4.6 Key Decision Taker of Change	35
5.5 Quality Management	35
5.5.1 Rules Applied to Maintain Quality	36
5.5.2 DSDM Standard Quality Measures	36
5.5.3 Quality Plan and Measuring Meter	37
Chapter 6 – Feasibility	37
6.1 All Possible Types of Feasibility	37
6.2 Cost Benefit Analysis for Self Storage	40
6.3 Explain DSDM Good or Bad for this Project	41
Chapter 7 – Foundation	42
7.1 The Problem Area Identification	42
7.1.1 Interview	42
7.1.2 Observations	42
7.1.3 Questionnaires	42
7.2 Rich Picture	43
7.3 Specific Problem Area Identification	44
7.4 Possible Solutions	45
7.5 Technology to be Implemented	45
7.6 Recommendation and Justification	47
Chapter 8 – Exploration	47
8.1 Activity Diagram	47

8.2 Full System Use Case.....	48
8.3 Full System Activity Diagram	49
8.4 Catalogue of Requirements	50
Chapter 9 – Engineering	52
9.1 Approach New System Modules:	52
9.2 Use Case Diagram of Self Storage.....	53
9.3 Self Storage's Class Diagram.....	54
9.4 Components of Class Diagram	54
9.5 Self Storage's ERD	55
9.6 Sequence Diagram of Self Storage	56
9.7 Self Storage's Components Diagram.....	57
9.8 Deployment Diagram of Self Storage	58
9.9 System Interface of Self Storage	59
Chapter 10 – Deployment	63
10.1 Core Module Coding Sample:	63
10.2 Possible Problem Breakdown	69
10.3 Prioritization while Developing the Solution	70
Chapter 11 – Testing	71
11.1 Test Plan Acceptance	71
11.2 Test Case.....	72
11.3 Unit Testing	73
11.4 Module Testing	75
11.5 Integration Testing	76
11.6 Acceptance Testing	79
11.7 Security Testing	80
11.8 Accessibility Testing	80
Chapter 12 – Implementation	81
12.1 Training	81
12.2 Implementation Scheme	81
12.3 Scaling	81
12.4 Load Balancing	81

Chapter 13 – Critical Appraisal and Evaluation	82
13.1 Objective Could be Met.....	82
13.2 Objective that totally don't meet of touched	83
Chapter 14 Lessons Learned	84
14.1 Pre-Project-Review-closing	84
14.2 What I have Learnt.....	84
14.3 The problems I have faced.....	84
14.4 What Solution Occurred	84
Chapter 15 – Conclusion.....	85
15.1 Summary of the Project	85
15.2 Goal of the project.....	85
15.3 What I have done in the documentation	85
15.4 Value of the Project.....	85
15.5 My Experience	86
Appendices	87
Works Cited	93
Plagiarism Report	95

List of Figure

Figure 1:Gantt Chart of the Proposed Timebox Estimation	12
Figure 2:DSDM Methodology.....	18
Figure 3:SSADM methodology.....	19
Figure 4:Rapid Application Development methodology	20
Figure 5:Work Breakdown Structure	23
Figure 6: Project Life cycle Gantt Chart	25
Figure 7: Sample of Testing Against the Timebox.....	26
Figure 8: Rich Picture	43
Figure 9:Legends of the rich picture	44
Figure 10: Client Server Application Model.....	46
Figure 11:User Activity Diagram	47
Figure 12:Use Case of the Proposd System	48
Figure 13:Admin Activity Diagram of the System.....	49
Figure 14:General User Activity Diagram System	49
Figure 15:Use Case.....	53
Figure 16: Class Diagram	54
Figure 17:Components of Class Diagram	54
Figure 18:ERD	55
Figure 19:Sequence Diagram.....	56
Figure 20:Component Diagram.....	57
Figure 21:Deployment Diagram	58
Figure 22:Interface design home page.....	59
Figure 23:ads posting.....	59
Figure 24:Warehouse management Service Page interface.....	60
Figure 25:Storage Search Page interface	60
Figure 26:Transport Service Page interface	61
Figure 27:User Storage Page Interface	61
Figure 28:User ads Page Interface.....	62
Figure 29:Ads posting	63
Figure 30:Taking rent	65
Figure 31:Transport.....	66
Figure 32:User Storage	67
Figure 33:Renting Out Personal Goods	68
Figure 34:unit test one test case.....	73
Figure 35:Unit test one result	73
Figure 36:Unit Test two	74
Figure 37:Unit Test three.....	75
Figure 38:Module Test one	76
Figure 39:Integration Test one	77

Figure 40: Integration Test two	78
Figure 41: Acceptance Test.....	79
Figure 42: Security Testing.....	80

List of Table

Table 1: Prioritized Features	9
Table 2: Prioritized Features	10
Table 3:Time box estimation	11
Table 4: Resource allocation list	24
Table 5: List of the Time Boxes	25
Table 6: Sample Test case	28
Table 7: User Acceptance Test Plan	28
Table 8: Risk Identification table	30
Table 9: Risk Assessment table.....	31
Table 10: Risk Precaution.....	32
Table 11: Risk dealing steps.....	33
Table 12: Business Priority	34
Table 13: Web application cost.....	39
Table 14: Mobile application cost.....	39
Table 15:Desktop application cost.....	39
Table 16:Total cost esitimation for the project	40
Table 17: Earning estimation of the project.....	41
Table 18: Estimated revenue on five year scale	41
Table 19:Questionnaire sample	43
Table 20:Questionnaire sample	43
Table 21: Requirement Catalogue for login	50
Table 22: Requirement Catalogue for post ads.....	50
Table 23: Requirement Catalogue for storage renting.	51
Table 24: Requirement Catalogue searching.....	51
Table 25: Module for logging in	52
Table 26: Ads/to let posting Module	52
Table 27: Search Module	52
Table 28:Renting module.....	52
Table 29: Accesibility Testing	80
Table 30: User training	81

Chapter 1 - Introduction

1.1 Introduction

Self Storage is a one step solution for those who are looking for renting storage units and warehouses. It is an online based application that will help the users to find storage units according to their needs and requirements where they can store their unneeded items as well as other belongings that do not necessarily always require but only on special occasions which takes too much space in the house. So, the system will allow users to rent a storage unit for storing their belongings in a controlled environment. Small businesses can use the system for finding and renting a warehouse for storing their business goods with a warehouse management system. From where they can deliver products and manage their inventory as well through the system. The system also provides transportation services for the users and business to get rid of the extra hassles of transporting personal belongings and products from one place to another.

In recent times people are moving around a lot as technology and communication systems have improved and people's lives have gotten a lot easier. As a result most of the people move around from one city to another and change houses frequently. As a result, keeping personal belongings safe is deemed a major concern. And for small businesses that are doing their business through the facebook market often use their home for storing the products and managing inventory which is not ideal for a business. Also, we see that our root level people who produce raw materials and foods have no opportunity to directly supply their products to the customers. The system will help solve all of these problems by providing them an online based platform which will automate the process of business. Farmers and raw materials producers can store their products by renting warehouses and can store for a certain period of time where they will be notified to sell their products at a higher price. It will maximize their profits as well as give them more reach to customers. They can automate the process by taking warehouse management service where they will be able to monitor the prices of their products on the current market and maintain their inventory automatically. When they decide to sell their products through the system they can take the transportation service which will deliver them at their desired location. Here, general users can also rent their unused belongings that are currently stored in their rented storage units if they want to earn extra money and it will also help others who need goods for a short period of time. The system basically will work with the untapped market of storage units and its possibilities.

1.2 Document Contents in Project Document

This document will cover the following chapters for documenting the processes of this project work.

Chapter 1: Introduction

Introduction and overview of the proposed system and project are discussed here.

Chapter 2: Initial Phase

In this chapter the proposed project's initial case studies such as aims and objectives of the project, problem areas the project is going to work with, background study and possible solutions are discussed.

Chapter 3: Literature Review

In this chapter in depth discussion about the problem domain, solutions, evaluation of existing solutions and recommendation is done.

Chapter 4: Methodology

As the name of the chapter suggests, the use of methodology is discussed here. Different types of methodology and what methodology is suited best for the proposed project is evaluated here.

Chapter 5: Planning

Planning is a crucial part as every phase of the development such as design, database development, risk management etc. all requires detailed planning or else the chance of project failure increases. In this chapter all the required types of planning are discussed.

Chapter 6: Feasibility

Feasibility of the proposed system is done here to see if the system is technically and economically feasible.

Chapter 7: Foundation

Based on the problem area analysis requirement list, implementation technology and the justifications behind the decision is done here.

Chapter 8: Exploration

Here different systems are explored that has similarity to the proposed system.

Chapter 9: Engineering

In this chapter the engineering process behind the proposed system's functionalities are discussed.

Chapter 10: Deployment

Here the coding samples of core functionalities are presented along with development priority of the broken down problems.

Chapter 11: Testing

All the required tests for the proposed system and the test plans are documented here.

Chapter 12: Implementation

In this chapter the steps of implementation are discussed such as training module to train the end users.

Chapter 13: Critical Appraisal and Evaluation

Here the system is evaluated to see if the objectives are met which were declared initially in the beginning of the project.

Chapter 14: Lessons Learned

Here the learnings from developing the proposed system is listed along with the problems faced and how I have overcome them.

Chapter 15: Conclusion

A short statement of the whole system's development process is given here along with initially declared goals, success and experience.

Chapter 2 – Initial Study

2.1 Project Proposal

Introduction

Self Storage is a one stop solution for those who are looking for renting storage units and warehouses. It is an online based application that will help the users to find storage units according to their needs and requirements where they can store their unneeded items as well as other belongings that do not necessarily always require but only on special occasions which takes too much space in the house. So, the system will allow users to rent a storage unit for storing their belongings in a controlled environment. Small businesses can use the system for finding and renting a warehouse for storing their business goods with a warehouse management system. From where they can deliver products and manage their inventory as well through the system. The system also provides transportation services for the users and business to get rid of the extra hassles of transporting personal belongings and products from one place to another. A one stop solution for renting space and utilizing it in regard to many related opportunities.

Background Study

There are some renting applications where users can rent houses and warehouses but there is no such system that provides rental service of all kinds within a single platform with systems like warehouse management system, transportation service, and also providing the users to rent or sell the stored products on the same application. The existing systems are basic rental systems where users just post an ad and take rent. Some existing applications are <https://rents.com.bd/>; <https://www.bproperty.com/>; <https://rakhibo.com/>; out of these three applications two applications work with housing rents mainly and one application works with storing personal belongings under their authority.

There is no application that allows users to rent storage by themselves and utilize that storage in the users' favor as they want. The proposed system is not only just a rental application, it gives the users full control of the storage and the belongings that are stored in the rented storage along with other useful automation tools. As this application is providing so much more than the other existing applications it will have high demands among the users and in the market.

Description of the proposed system

Self Storage is an application that will allow users to rent storage, warehouse, post ad of storage and warehouse, take warehouse management service for rented warehouse and transportation service as suited. Users can search and rent storage and warehouse from the ads posted by other users according to their requirements. The warehouse management service comes with lots of perks for the businesses as it will provide a schedule and reminder of when selling products will be most profitable. Users can use the rented space as they want and can list/post ads of stored items for rent as well.

Feasibility Study

❖ Operational feasibility

Operational feasibility is concerned with the ease of access, usability, proper functioning of the required functionalities of the proposed system, the system's proper validation and verification of user inputs, smooth navigation through all the functionalities, location, rent amount, space based search ability and automated business process making the system effortless to use. Users can easily access the ads and post ads of their own storage, warehouse and products with their valid credentials and take service through the system.

❖ Technical Feasibility

The aspects that concern technical feasibility are revealed in technical feasibility tests that users of the Self Storage system can effortlessly rent storage space, warehouse space and perform their business tasks as well from one system instead of managing all these differently on different platforms. I will be using flutter app development with dart, firebase, no sql database, android 12 and microsoft office for documenting the development life cycle of the project. The system is designed and developed with the latest technologies and framework of app development. To ensure the app is compatible with all the devices with low configuration and connectivity. The system will have an access control system to maintain the security of the users, users' functionalities and data.

❖ Economic Feasibility

The proposed system can be developed in various alternatives including desktop application, mobile application, web based application etc.

⌘ Desktop application

Desktop based applications need to be installed in every desktop PC to use. The option is not viable as it will not be an easily accessible and not portable option for the users to use it whenever they need it. It will also require some additional equipment which is more costly.

Equipment	Cost per unit	Cost
Desktop pc (core i7 7th gen, 3.6 GHz processor gtx 1050ti gpu, 32 GB DDR4 RAM, 500GB SSD)	₹90000	₹90000
Web, File and Email servers	₹5000 per month	₹5000
Total		₹95000

⌘ Mobile Application

Mobile applications need to be installed in every smartphone to get the service of the proposed system. As these days smartphones are being used by almost every mobile user, developing mobile application of the proposed system is one of the efficient option for developing the system.

Equipment	Cost per unit	Cost
Desktop pc (core i7 7th gen, 3.6 GHz processor gtx 1050ti gpu, 32 GB DDR4 RAM, 500GB SSD)	₹90000	₹90000
Web, File and Email servers	₹5000 per month	₹5000
Total		₹95000

⌘ Web based application

In web based applications the entire system's data is stored in a web server from where it can be accessed any time with internet connection. For developing the system I will use web application as it can be accessed from all types of devices including PCs, smartphone and any mobile device that has internet connection.

Equipment	Cost per unit	Cost
Extranet network with VPN	₹5000 per month	₹5000
Desktop pc (core i7 7th gen, 3.6 GHz processor gtx 1050ti gpu, 32 GB DDR4 RAM, 500GB SSD)	₹90000	₹90000
Web, File and Email servers	₹5000 per month	₹5000
Total		₹100000

Market Research Analysis Based on the Feasibility Factors

Here the market and user demand is discussed. During the covid-19 outbreak people had to go into lockdown, at that time people had to move back to their permanent homes. And when they had to move back home problems arise. One of the problems is keeping the personal belongings safe with low cost. People had to pay extra rent for a whole flat to just keep their goods which was not affordable for them. To solve these problem storage units could play a vital role as renting a single storage unit is cost friendly. There has been a huge demand for renting storage and warehouses to store goods and belongings. To automate these, Self Storage can capitalize on the market as there is no such system that satisfies all the demands of the people. So the proposed system will provide a one stop solution for satisfying the rising demand.

Foundation

Aims, Objective and Scope

Aims of the proposed system

Aims of the proposed system Self Storage are:

- One stop rental service
- Location, space, cost friendly storage service
- Maintain warehouse inventory of businesses and suggest a more profitable window for selling goods.

Objective of the proposed system:

- Managing the home owners around dhaka to post ads/to-let
- Getting users requirement and purpose of renting storage
- Searching area, space and cost based results for users
- Managing renting market information
- Managing information of stored products for the businesses
- Predicting profitable window for specific goods

Main Goals

- Creating a platform for rental storage units
- Location, size, cost based searching of storage and warehouse
- Ensure providing easy access to storage and related facilities such as transportation

High level features/requirements for achieving the goals and objectives of the proposed system

- Registration/login for the users
- Ads posting ability for the users
- Renting request management for the users
- Ads manipulation system for the admin
- Location, size and storage type authentication
- Warehouse management system for businesses
- Transport service for the users
- Notifications system for the the businesses about market situation
- Contact service for the users
- Preferred storage request option for the users
- Renting option for the stored belongings and goods on storage

Key activities for achieving the goals

Activity	Description
Project Initiation	I have decided to develop the app to capture the market of renting storage units and the business opportunities it presents to the users.
Feasibility Study	Operational, technical and economical analysis have been done to determine the feasibility of the project.
System Analysis	The prioritized requirements and functionalities were selected from the requirement catalog.
System Architecture	Many technical architectures such as DBMS, server and interaction with other systems are measured here.
System Design	Structural and behavioral design is done in this phase
Implementation	The development methodology to implement the project and other required tools such as programming language are used to implement the proposed system
Testing	The development process and finished product is tested.
Other Project Matters	Risk management, configuration etc.
Documentation	The documentation of the project lifecycle is recorded throughout the development process.

Table 1: Key activities for achieving goals

Prioritized Features according to MoSCoW

MoSCoW is a prioritization technique used during the requirement analysis phase of project development. It helps to determine the importance of requirements to the project and develop it accordingly. According to the MoSCoW technique prioritized requirements of the project are:

SL	Requirement	Priority
01	registration & login	must have
02	admin manipulation	should have
03	post ads/to-let of storage/warehouse	must have
04	rent storage/warehouse	must have
05	request storage/warehouse	should have
06	manage storage/warehouse renting	must have
07	provide warehouse management service	should have
08	manage transportation service	must have
09	stored goods and belongings renting	must have
10	manage stored goods and products	must have
11	location, area, cost based search	must have
12	notification system for businesses	should have
13	contact	must have

Table 2: Prioritized Features

Exploration & Engineering

Based on the prioritized requirement list the finalized requirements are:

- User registration/login system
- User management system
- Storage/warehouse renting
- Ads/to-let facility
- Location, are and cost based storage/warehouse search facility
- Transport service
- Warehouse management service for the businesses
- Renting facility of the stored goods/products in storage/warehouse
- Notification alert
- Contact facility

Iterative development – Timeboxing

In iterative development the tasks need to be divided into segments with each segment's estimated time to complete the segments. Actual development in this project takes 50-60%time, feasibility and foundation stage takes 15-20% time and documentation takes the rest 15-20% of total development time.

The proposed project has to be completed within two and half months which is approximately 62 working days. The project is divided into ten time boxes of 4-10 working days.

I have used the “T-Shirt” estimation technique for estimating the time-boxes. The duration for the timeboxes will be estimated by categorizing the timeboxes in t-shirt sizes such as XS, S, M, L, XL. Both time and effort for the bigger size will be more than smaller ones. (Berteig, 2015)

Time Boxing

TIME BOX	Tasks	Start Date	End Date	Duration
TB1	Feasibility and Initial Study	21-05-2022	25-05-2022	4-5 days
TB2	Foundation and Requirement Analysis	25-05-2022	30-05-2022	4-5 days
TB3	Database Design	01-06-2022	03-06-2022	3 days
TB4	Developing User login/registration and Storage Renting System	05-06-2022	15-06-2022	10 days
TB5	Developing Add/to let system	16-06-2022	24-06-2022	8 days
TB6	Developing Warehouse management and Transport service	25-06-2022	05-07-2022	10 days
TB7	Developing user dashboard, user's rental information and payment system	05-07-2022	15-07-2022	10 days
TB8	Implementing and testing the system	17-07-2022	21-07-2022	5 days
TB9	Deployment	23-07-2022	27-07-2022	5 days
TB10	Documentation of the Development Process	21-05-2022	27-07-2022	60-62 days

Table 3:Time box estimation

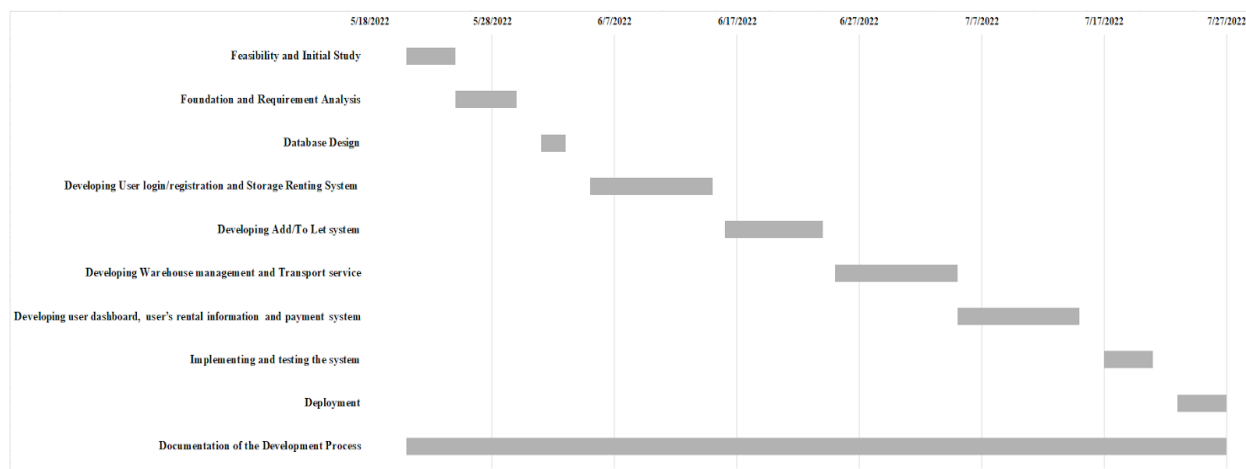


Figure 1:Gantt Chart of the Proposed Timebox Estimation

Deployment

In the deployment phase the recommended solution for the proposed system will be implemented. Here, I will develop the app, database for the proposed system. I will follow an iterative development approach. To do that I will be using the DSDM atern development methodology. I am going to develop the system using flutter app development framework with dart programming language along with Firebase and will use NoSQL for database implementation.

Conclusion

In the discussion above it is established that “Self Storage” will be impactful for the people and it will solve the problems of having to pay unnecessary rent. Location, size, cost based searching helps people find their required storage/warehouse suitable to their needs. And the transportation service of the system will ensure the safe transportation of the belongings and goods of users. Also, the warehouse management service will provide market price for the stored goods. So, it can be said that the project will help general users as well as businesses

2.2 Background of the project

There are few house renting systems but there is no rental system that works with renting storage units and warehouses. The existing system that offers renting housing units just provides the service of introducing the owners and tenants for a small fee while the new proposed system will not just introduce owners and customers but also provide services needed by the customers/users.

There is a void for storage unit and warehouse renting which can turn to a huge market if can be utilized properly. The problem caught eyes when the lockdown happened due to covid-19 outbreak. People had to leave Dhaka and their belongings were left unsupervised and people had to pay for house rent without living there just for storing the belongings. Also, a congested city with little planning like Dhaka does not provide much spacious housing and it is costly as well keeping just the necessary belongings is hard to manage. So, Self Storage can save people from the hassles with a cost free solution. People can move around without having to worry about managing their goods and belongings. It will also provide online businesses that operate from home a place for storing and keeping their products with warehouse management service which will automate their business process and get rid of the hassle of storing business products in living space.

There are few organizations that provide storage service, they can be integrated with the system and can provide their storage for renting through this proposed system. As they only just provide the storage to the customers, through this system they can extend their business a bit more. Because the new system provides a one stop solution for all the storage and warehousing related requirements. So, the demand for such a system can grow huge once people get the idea of utilizing the system.

2.3 Problem Areas

It is a huge issue for the people who have to move around a lot and independent businesses who operate by themselves to manage the storage problem of their personal goods as well as products. Transporting the goods from one location to another is also another hassle for people. One existing system Rakhibo.com works on this issue but they do not give full access to the users. They just take booking online, pick up goods and store it wherever they want. In this case the customers do not have access to their goods and they can not use the storage facility on their own. Also, we see that independent producers often face the problem of collecting their products and selling them as soon as they can without making much profit, but if they could keep their product for a certain time when the products' price will rise they can profit more. These are the problems areas the proposed system is going to work with.

2.4 Solutions to the problems

These problems are complex to solve because they are all connected to each other. Just trying to work with one problem at a time does not provide an efficient solution, it rather creates a mess. For example rakhibo.com provides storage service but they do not offer any other service regarding this and on top of that users do not have access to their goods. And other renting systems like rents.com.bd only allows the user to rent housing space. The possible solution to all these mentioned problems is a system that will provide all the relevant services such as warehouse management, transportation with storage service as well. It will provide location based services preferred by the users. These can be deployed via various technologies but web based application is preferred because it is the best way to reach users and for users the easiest way to access the system from all platform.

Chapter 3 – Literature Review

A literature review is a sort of assessment paper. The term "literature review" refers to a material UI collection on a certain topic, not to the entire world's literary content. The term "literature review" refers to trustworthy information on a certain topic that is published in a respectable publication after thorough research and analysis. A literature review provides the foundation for comprehending a particular topic for the system's development purposes. The identification of problem areas and their remedies is aided by a project literature review. This component facilitates a potential discussion about the research as well as a comparison within the same sector of work, setting the way for a successful project.

3.1 Discussion on the Problem Domain

There are some renting applications where users can rent houses and warehouses but there is no such system that provides rental service of all kinds within a single platform with systems like warehouse management system, transportation service, and also providing the users to rent or sell the stored products on the same application. The existing systems are basic rental systems where users just post an ad and take rent. There are some renting companies that made web based systems for their service but they do not offer various services related to the renting problem. The main problems are:

- There is no application that allows users to rent storage by themselves.
- Not having area, space, location, cost based filtering system
- Not allowing users to post ads
- Not having transportation service
- Do not offer business solutions.
- Do not let users take advantage of their rented services.
- These are the main problems of the current storage and warehouse renting market

3.2 Discussion on the Problem Solution:

The proposed system is a mix of offline and online-based issues, but it can be solved by creating an online based system like the proposed system. Having access to the internet is critical for this kind of solution. Self Storage focuses on ways to solve the problem of renting storage & warehouse units. It works with the home owners and warehouse owners for posting ads of their available units.

- **Safety and security**

As the system will be holding private data, including personal data, data security should be a top focus for the solution. The system should have high-security measures as well as role-based data access restrictions.

- **Trust**

The most serious issue the online based services face is their trustworthiness. Users should be able to see through the entire system & have all the possible access to all information the system holds by also maintaining other users' data privacy.

- **Privacy**

The privacy of the users, their products, and sensitive information about the transaction of that particular system must be protected. To guarantee privacy, the system should use encrypted data transfer and an encrypted VPN connection for client server answers.

- **Server Down or 24/7 Availability**

The application's accessibility from everywhere. The application must be accessible 24/7 to provide continuous service to the users. Sometimes there is a lack of bandwidth, and there is a lot of traffic to manage and these may cause server down problems. So it should have a contingency.

- **Backup**

Because the system deals with a large volume of data, it must guarantee that backups are made on a regular basis. A proxy server and data storage should be kept on hand at all times in case the system has to be backed up.

3.3 Comparison among the leading solutions

As mentioned before there is no such system that provides services like the proposed system discussed. Though there are some websites that provide only renting of housing units they are not satisfying the void regarding the storage and warehouse market. The results of the analysis are divided into two sections, Best Features and Limitations. They are listed below:

Best features of the existing systems:

- User friendly
- Ads/to let section is organized
- Renting system is good
- Notification system

Limitations:

- Only work with housing units
- Provides no other service except renting
- Searching criteria

3.4 Recommended Approach

It is seen from the above discussion that there is no such system that provides the service. And the existing systems do not work with the whole renting market. So, the analysis shows features that the proposed system should have, they are:

- The system's design is user-friendly and appealing.
- The procedure of verifying and approving identity.
- Criteria based search system
- Continuous update regarding stored goods
- Give the users full access to their rented storage/warehouse management.

Chapter 4 – Methodology

For making a project successful and managing the development process of the project during its life cycle a software development methodology is selected by the development team that is best suitable for the project. There are many development methodologies and all of them have their pros and cons. These methodologies' success depends on the type of project they are selected for. Also, the success of a project depends on how a development methodology is being utilized across the project. (alliancesoftware, 2017) in this chapter in depth discussion of the selected methodologies will be done and will discuss the reasons behind selecting one appropriate methodology for the project's development.

4.1 What to Use

Software development methodologies are the defining stages of a software development cycle. Various software development methodologies are used for software development. They are waterfall model, prototype model, agile software development, rapid application development, dynamic systems development model, spiral model etc. From these methodologies the appropriate methodology according to the proposed system's development process is chosen to be followed. Here, three development methodologies will be discussed and one will be selected based on the discussion for the proposed development.

Dynamic System Development Method (DSDM)

DSDM works based on the rapid application development methodology. But what distinguishes this approach from rapid application development is continuous user involvement throughout the whole development process of the project. In this methodology iterative and incremental development approaches are followed strictly. (tatvasoft, 2014) DSDM follows a proper approach for building quality projects with continuous feedback from the users.

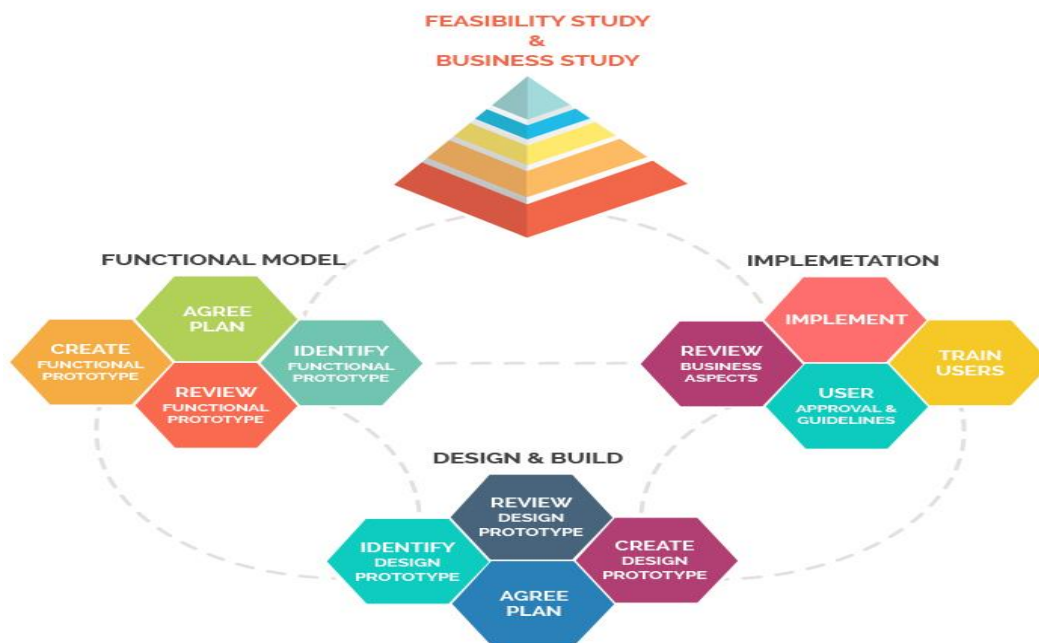


Figure 2:DSDM Methodology

Benefits of DSDM

- High user involvement in this development process so user already get a good idea of the system
- In DSDM main functionalities are developed first based on the prioritized requirements list and the rest of functionalities are developed in incremental basis.

- It delivers the project approach on time and within the budget. (tatvasoft., 2017)
- Allows change and responses quickly to the suggestions and feedback.

Drawbacks of DSDM

- Strictly manages time no matter the cost
- Programmers are not much involved in the decision making process.
- Project can be spoiled if the client is not clear about the project's idea.

Waterfall Model or Structured System Analysis and Design Method (SSADM)

The Waterfall model is a sequential development approach that works like a waterfall, all the process flows through the stages of project life cycle such as analysis, design, implementation and testing; each stage is completed before moving on to the next. It emphasizes the linear flow of the processes from beginning to end. This methodology is appropriate if a project is thoroughly planned and well documented. Success of this methodology relies on consistent and accurate execution of the plan according to the detailed documentation.



SSADM Structure

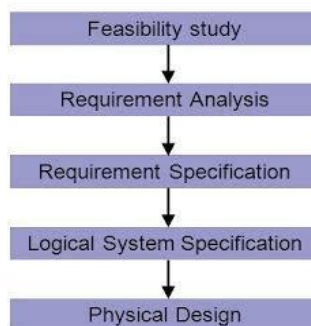


Figure 3: SSADM Structure

Benefits of SSADM

- Easy to understand and simple to implement
- As the stages are very well planned it is easy to estimate accurate cost and time
- Tracking work progress is easier
- Projects with clear requirements are appropriate for this methodology
- Testing phase easy as reference of the previous stages can be used.

Drawbacks of Waterfall

- Do not allow iteration in once development process begins
- Not suitable for a long term project
- If one stage is affected by any crisis whole project gets delayed

Rapid Application Development

Rapid Application Development is one of effective methodologies that allow rapid development and reliable work compared to other existing software development methodologies. It is structured in such a way that it credits the maximum advantages of the software development in an easy way.

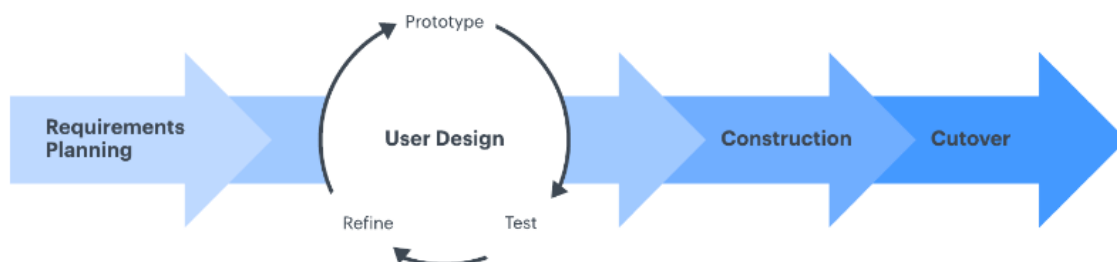


Figure 4:Rapid Application Development methodology

Benefits of RAD

- Decreases the risks to the minimum of the software development.
- Suitable for smaller projects
- Allows feedback in the beginning of the project which encourages customers to provide feedback and it helps to improve the software.
- Allows deliverables to be easily transferred via scripts, high-level abstractions, and intermediate programs.

- Because of code generators and reuse, manual coding has been diminished, and it is very likely that there will be less faults in nature as a result of prototyping.(Martin, 2021)

Drawbacks of RAD

- Needs expert developer because the methodology heavily relies on reusing codes and programs
- Costly approach as heavily experienced developers is required.
- Rapid application development can only be used to design systems that can be modularized throughout the life cycle of the product.(Martin, 2021)

Choosing Methodology

However, the waterfall model has been employed in farm management systems in some circumstances. Because the DSDM technique always provides product and project quality on time and is fairly versatile for application in any company or organization, it will be the finest choice and ideal solution for farm management system project. During the development phase, clients or clients may adjust their requirements if they believe they are necessary for their project. After exploring the above discussed methodologies I have decided to use the DSDM methodology for developing Self Storage. I feel like it is the best available option out of all the other methodologies because it supports iterative development. As this is an academic project and I am a beginner at developing, I might make mistakes that may need to be fixed later. DSDM supports going back and forth between the stages of development but the other discussed methodologies do not have that facility. Also, with DSDM methodology iterative and incremental development support projects can be developed to satisfy the requirements at a full. It offers flexibility and assurance of project's quality within a defined time frame. These are why I have chosen the DSDM methodology for my project.

4.2 Why to Use

To successfully develop a project and satisfy the requirements of an application a development methodology has to be followed, otherwise the project might not be fully successful or may be abandoned midway. So, to make a successful project development, a methodology with

predefined steps needs to be followed. Choosing a methodology for a project is a very complex thing but it will help deliver output and complete the project.

4.3 Sections of Methodology

The DSDM methodology has some fixed predefined steps to be followed by the development team to develop a project. These are:

Pre-project Phase

In this phase the proposed project is initiated by budget and time estimation with defining the primary requirements.

Feasibility Study Phase

Feasibility study is performed analyzed on technical, operational and economic aspect of the proposed system.

Requirement Gathering Phase

Various requirements gathering techniques such as interview, questionnaire, observation etc. are performed in this phase. By analyzing the outcome of these techniques the requirements for the proposed system are defined.

Requirement Analysis and Prioritization Phase

The previous phase's defined requirements are analyzed and prioritized following techniques such as MoSCoW, ranking, grouping etc.

Exploration and Engineering Phase

In this section the main functionality of the DSDM methodology is used which is iterative development of the given requirements and testing the developed features.

Review Phase

In this section the developed deliverables against the provided requirements are tested by the users and if any changes are needed can be sent back to the development process.

4.4 Implementation Plan

This is the final stage of the development work. In this stage the developed product is released for the use of general users. In this phase if there is any problem before release occurs are solved.

The criterias of releasing the product to the users are identified here, and when all criterias are fulfilled the product is finally launched.

Chapter 5 – Planning

5.1 Project Plan

In this step a projection is done for the planning process of completing the project. The project is divided into some defined parts and the project planning will be followed during all the parts of the project. It will ensure that the project is completed in a projected time for each part and the complete project is done according to the plan and on time.

5.1.1 Work Breakdown Structure

In this phase the breaking down of the project into smaller tasks so that it can be done within a specific time frame and in a more efficient and effective manner. This structure will provide us an estimation about how much time will be required to complete each task. Without this structure the completion of this project could be more complex to complete as things might not have a specific order and it will be unorganized. So, the work breakdown structure for the proposed project is given below:

SL	Task	Duration(days)	Start	Finish
1	Introduction	2	21-05-2022	23-05-2022
2	Initial Study	3	24-05-2022	27-05-2022
3	Literature Review	3	28-05-2022	31-05-2022
4	Methodology	3	01-06-2022	03-06-2022
5	Planning	5	04-06-2022	09-06-2022
6	Feasibility	3	10-06-2022	12-06-2022
7	Foundation	5	13-06-2022	18-06-2022
8	Exploration & Engineering	20	20-06-2022	10-07-2022
9	Deployment	5	11-07-2022	15-07-2022
10	Testing	5	16-07-2022	20-07-2022
11	Implementation	3	21-07-2022	23-07-2022
12	Critical Appraisal & Evaluation	2	24-07-2022	25-07-2022
13	Lessons Learned & Conclusion	2	26-07-2022	27-07-2022

Figure 5: Work Breakdown Structure of the System

5.1.2 Resource Allocation

All the necessary assets required to complete this project are managed and allocated in an organized way so there is no complexity during the development of the project. It is a crucial part of development. As this is an academic project I will be portraying the different roles needed to complete the project. For the proposed project the resource allocation is as follows:

SL	Task	Duration(days)	Resources
1	Introduction	2	Analyst
2	Initial Study	3	Analyst
3	Literature Review	3	Analyst
4	Methodology	3	Analyst, Developer
5	Planning	5	Analyst, Developers, Team Leader
6	Feasibility	3	Analyst, Users
7	Foundation	5	Analyst, Designer, Developer, Team Leader
8	Exploration & Engineering	20	Analyst, Designer, Developer, Team Leader
9	Deployment	5	Developer
10	Testing	5	Developer, Tester, Users
11	Implementation	3	Developer, Designer
12	Critical Appraisal & Evaluation	2	Analyst
13	Lessons Learned & Conclusion	2	Analyst, Designer, Developer, Team Leader, Tester
14	Total	61 days	

Table 4: Resource allocation list

5.1.3 Time Boxing

Another very important part of DSDM methodology is timeboxing. In this planning the tasks are divided into time boxes. Each time box contains some tasks that need to be completed within a defined time frame. The tasks are performed iteratively and flexibly within the time box.

TIME BOX	Tasks	Start Date	End Date	Duration
TB1	Feasibility and Initial Study	21-05-2022	25-05-2022	4-5 days
TB2	Foundation and Requirement Analysis	25-05-2022	30-05-2022	4-5 days
TB3	Database Design	01-06-2022	03-06-2022	3 days
TB4	Developing User login/registration and Storage Renting System	05-06-2022	15-06-2022	10 days
TB5	Developing Add/to let system	16-06-2022	24-06-2022	8 days
TB6	Developing Warehouse management and Transport service	25-06-2022	05-07-2022	10 days
TB7	Developing user dashboard, user's rental information and payment system	05-07-2022	15-07-2022	10 days
TB8	Implementing and testing the system	17-07-2022	21-07-2022	5 days
TB9	Deployment	23-07-2022	27-07-2022	5 days
TB10	Documentation of the Development Process	21-05-2022	27-07-2022	60-62 days

Table 5: List of the Time Boxes

5.1.4 Gantt Chart

It is a graphical representation of the project's activity schedule. It represents the progress duration in a bar chart from start to finish of an activity. The gantt chart for the proposed project is:

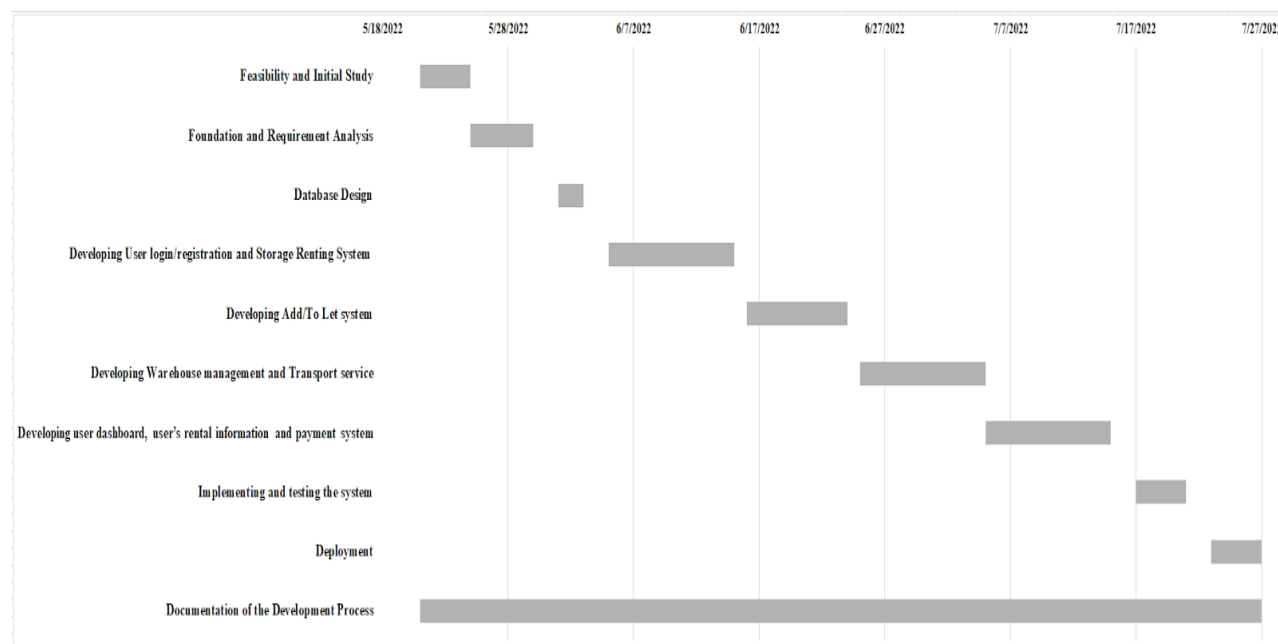


Figure 6: Project life cycle Gantt Chart

5.2 Test Plan

Test plans are developed based on the differences between the given input and expected output from performing a specific task. During the software development following DSDM methodology the testing are done including verification and validation process in every stage of development.(codeProject.com, 2012).

5.2.1 Testing Against the Time Boxes

Time box is done based on the complexity and requirements of the software development process by estimating a minimum and maximum number of working days it may require to complete a segment of the system. Testing Against time boxes:

User Name	Example	Role	Example
Time Box ID			
Time Box Content			

Test Type	Test Steps	Expected Results	Actual Results	Comment
Unit Test				
<u>Integreation Test</u>				
System Test				
<u>Acceptence Test</u>				
Security Test				
Usability Test				
Reliability Test				

Figure 7: Sample of Testing Against the Timebox

5.2.2 Required Test

Functional and Nonfunctional testing are the two major system testing modules among many testing modules:

Functional Testing:

- **Unit Testing:** This testing is done on the smaller parts of the design. In this testing the divided units are tested and checked. Also, these small units are tested

group wise too, to see when integrated how they work. It is done by the developers by using simple input and output.

- **Integration Testing:** After unit tests are done the tested units are integrated into a program infrastructure that implements the design of the system. During this, some groups of components are integrated together to produce expected output based on the given input by the developers. Black box and white box testing are parts of integration testing.(geekforgeeks)
- **System testing:** System testing is done only with black box testing for various OS such as windows, Linux etc. It works by taking required input and demonstrating expected output with internal working (GreekForceGreeks)
- **Acceptance testing:** Acceptance testing is done by involving the users at full extents. After completing this test the functional test is concluded. It observes If it works based on user expectations and satisfies the users.(GeekForGeeks)

Non-Functional Testing:

- **Security testing:** Security testing is done by an external tester. Here, the testing validates unauthorized user or access while also protecting from internal and external attack including SQL injections.
- **Usability testing:** Usability testing is done to see how user friendly the system is, how the users interact with the system and if they can easily get the service from the system. Basically the test is done to validate the usability and interactiveness of the system.
- **Reliability testing:** Reliability testing is done by directly involving the users who are involved directly with this testing. It validates the reliability of the system's operation. It looks out any type of failures of a system.

5.2.3 Test Case

Test case includes test case no, test type, test description and test steps.

Test Case No.			
Test Type			
Test Description			
Test Steps	Expected Results	Actual Results	Comment

Table 6: Sample Test case

5.2.4 User Acceptance Test Plan

It is the last step of the testing stage where it refers and check the system's performance by the involved user.

Test Case No.			
Test Type			
Test Description			
Precondition of Testing			
User's Name			
Act as			
Test Steps	Expected Results	Actual Results	Comment

Table 7: User Acceptance Test Plan

5.3 Risk Management

Risk management is the process of identifying and analyzing any risk that can occur during the project's development cycle. It enables a protection system for the project development life cycle against any risks that may occur. Risk can occur in any stage of the project's lifecycle and it may affect development, budget as well as on performance of development team and quality of the project. There are different types of risk management, these are analyzed below:

- Risk Identification
- Risk Assessment
- Risk Precaution/Action Plan
- Steps Taken for Possible Risks

5.3.1 Risk Identification

Risk identification is the first step towards risk management planning for better handling of risks that may occur. This is the most crucial part of risk management as in this step the risks are identified which may occur and affect the project as well as identify and prepare contingency plans against the risks. For this project some risk identification criteria that can be justified are:

- Possible risk identification and track the risk with documentation
- Identified risk elaboration including actual causes
- Actual causes are considered in forming identified risks.
- Future consequences and effects, as well as the impact of risk

Detailed analysis of possible risks identification and risks tracking of this project is given below:

Type of risk	Causes	Impact and consequence
Scheduling	There are many reasons to occur scheduling risks included- - Unintentionally delay cause of Natural disaster - Specific segments failure included data loses of users	Delayed project submission
Database error and database getting deleted	In this project several types of users were included like users, admin etc. Sometimes an user may not get to take transport service	Users may not find the proper career guideline
Unauthorized access	When a person tries to access this system without permission. This may cause and create a difficulty with risk included May delete/change data of users	Personal information may get hacked and may be used to cause harm.
Technical Problem	- Lack of performance - Lack of required bandwidth - No matching the required configuration	Scheduling failure
Project Backup	Not backing up the project at any other location or system	- Affects cost - May lose whole project
Network Failure	Inability to provide the required connection to get access to some integrated APIs and email management	Email/notification may not be able to sent to the user

Table 8: Risk Identification Table

5.3.2 Risk Assessment

Risk assessment involves with quantitative or qualitative value of risks included threats of a project

Type of risks	Likelihood	Impact	Restore time
Scheduling	5	6	8
Database error and database getting deleted	5	5	5
Unauthorized access	6	6	6
Technical Problem	4	5	5
Project Backup	5	6	4
Network Failure	4	4	8

Table 9: Risk Assessment

5.3.3 Risk Precaution / Action Plan

Risk action plan is taken after completing risk identification and risk assessment. There are various steps that need to be followed to take an action plan against projected risks. They are.

- To avoid the dangers that have already happened and those that may come in the future
- To minimize the hazards that have already happened and those that may arise in the future
- To address the issues that have arisen as well as potential dangers

Type of risk	Action	Action taken by	Required Action
Scheduling	• Make accurate time estimation with tools like gantt chart • Use technology like forecast and hive	developer	During projects initial planning
Database error and database getting deleted	• Make enough normalization • Make perfect relation to the table	developer	During adding data to the database
Unauthorized access	• Perform code by maintaining code smelling • Make multi-authentication system	developer	During setting authentication system
Technical Problem			
Project Backup	• Make Backup of project	developer	Frequent backup
Network Failure	• Make strong communication with network experts to ensure a continued service	developer	During project development

Table 10: Risk Precaution

5.3.4 Steps Taken for Possible Risks

The steps taken for the proposed project by identifying and evaluating the identified possible risks some risks precautions are taken. These are:

Type of risk	Likelihood	Impact	Steps taken
Scheduling	rare	high	The project time boxing and the Gantt chart and work breakdown structure created.
Database error and database getting deleted	likely	high	3rd normal form normalization and entity relationship diagram used and data dictionary developed.
Unauthorized access	unlikely	high	The proper authentication system and access authorization are implemented.
Technical Problem	Frequent	medium	Ensured backup and regular maintenance and monitoring
Project Backup	unlikely	high	Frequently backup to other location
Network Failure	Frequent	low	The proxy server has arranged and high-speed bandwidth allocated.

Table 11: Risk dealing step

5.4 Change Management

5.4.1 Factors that Might Cause Change

Many changes may have to be made for this system which is:

- Scheduling segments has to be adjusted
- Functional requirements need to update
- Various user interface module has to be made
- Security handling needs to be adjusted

5.4.2 DSDM Welcome Change

The project development life cycle may need to be changed time to time based on the project development process and the changes along the progress of the development. For these changes we should follow a methodology that allows changes whenever necessary. This is why DSDM is one of the applicable, trustworthy, appropriate and implementable methodologies. During this system development I will use the DSDM methodology to maintain and make changes during the time of development of the project. How the methodology works on different approaches are:

- Tracking and updating of system module is done using a predefined system
- Continuous user feedback is taken during development in case of needing any changes
- It performs after completion of changes
- Assures the security purpose
- Provides reliability with scalability

5.4.3 Considering Business Priority

Changes are required to fulfill the expected goals, requirements and vision with business purpose. For this changes on functionalities needs to be prioritized with the requirements that will be the causes of benefits for involved parties.

Priorities are as follows:

Area of Topic	Priority Level
Scheduling	5
Functional Requirements	7
User Interface	5
Security Handling	5
Users Feedback	7

Table 12: Business Priority

5.4.4 Change Workshop

Knowing the profile of the development team along with users' communication will be built up strongly. Based on the required changes and additions the developers arrange a workshop with questionnaires which will be used later to make changes and additions to the system.

And knowing users' profiles of diverse users experience discussion can be made based on the implementations done on user experience of the system. These reviews during the discussions will be maintained in the workshop. Change workshop required for:

- To define the change and involved team
- To create strong profile of team members including users

5.4.5 Changes that are allowed

Changes are needed to be accepted based on their priority level. It also needs to be supported by resources such as cost, schedule, quality and risks. During the development of a particular segment of the system priority wise changes should be allowed to be made.

5.4.6 Key Decision Taker of Change

Key decisions of changes will be taken by several experts in their area.

In this project the decision taker to make changes include:

- System Requirements Analyst
- System Developer
- System Tester

If necessary, the analyst can alter the functional requirement, and the developer can alter the applicable code.

5.5 Quality Management

Maintaining the quality that the user agreed to accept during the project's beginning is what quality management is all about. Quality management is based on a number of factors, including:

- For providing the best possible renting option
- Communication within team members and users
- Frequent update of requirements

5.5.1 Rules Applied to Maintain Quality

Rules that are applied vary system to system and users to users, different systems may have different rules for different users. During the quality management process some rules need to be followed including quality control and assurance

Quality Control:

- By authenticating users authenticity quality should be controlled including:
- By taking identification documents from users

Quality Assurance:

- By evaluating the approaches taken if they are right or wrong quality assurance has to be maintained including:
- By handling user's storage information
- Sending appropriate and correct email/notification to the users including quality control and assurance.

5.5.2 DSDM Standard Quality Measures

Solution quality: Solution quality is ensured based on the expected outcome of business need and users' expectations based on the users' requirements. These processes are completed based on two types of prioritization

- Time boxing
- MoScow

These approaches are completed by maintaining the schedule.

Process Quality: Process quality is ensured based on the targeted organization. Two types of approaches are taken to complete this

- CMMI
- DSDM

5.5.3 Quality Plan and Measuring Meter

Many different types of quality control plan have been followed to this stage including:

- Proper scheduling and resource allocation
- Testing in every segments of development
- All the changes are tracked and documented properly for further improvement and execution.

Chapter 6 – Feasibility

6.1 All Possible Types of Feasibility of the Proposed System

Operational Feasibility:

As operational feasibility is concerned with the usability, ease of use and required functionalities of the proposed system, the proposed system provides proper validation for user inputs and verifies them as well, easy to use navigation system, easy storage renting system that helps the user to use the system effortlessly. Users can access the system with valid username and password and take their desired service.

Technical Feasibility:

There are many aspects of technical feasibility. They are:

Technical feasibility study shows that the users of Self Storage can easily find a storage/warehouse according to their interest. The system is designed and developed using the latest web technologies to ensure it runs smoothly on all types of smartphones and compatible with low quality internet connection. The system maintains users' hierarchy to maintain data security. As it is a web based system so it can be accessed from all types of digital devices with browser and internet access, so it is affordable and cost effective to use as the system would not require any extra hardware, so it is technically feasible. Technical aspects of the system are:

Hardware:

- Desktop PC
- Wi-fi router

Software:

- Xampp

- VS code
- MS office
- Google Chrome browser
- Windows 10 OS

Database:

- MySQL

Technology:**Client Side:**

- HTML
- CSS
- Java Script
- Bootstrap
- Firebase
- Flutter
- Android 12
- iOS

Server Side:

- Laravel
- PHP

Economic Feasibility:

There are many other ways to develop the proposed system including web-based application, mobile application, desktop application etc.

Web based application cost:

Web applications do not require installation on a device. The application and related data that requires to make the application functional is stored at remote server. From the remote server the application can be accessed using internet from all over the world at any hours

Equipment	Cost per unit	Cost
Extranet network with VPN	₹2000	₹5000
Desktop pc (core i7 7th gen, 3.6 GHz processor gtx 1050ti gpu, 32 GB DDR4 RAM, 500GB SSD)	₹90000	₹90000
Web, File and Email servers	₹5000 per month	₹5000
Total		₹100000

Table 13: web application cost

Mobile application cost: the app containing the system needs to be installed on every device.

Equipment	Cost per unit	Cost
Desktop pc (core i7 7th gen, 3.6 GHz processor gtx 1050ti gpu, 32 GB DDR4 RAM, 500GB SSD)	₹90000	₹90000
Web, File and Email servers	₹5000 per month	₹5000
Total		₹95000

Table 14: Mobile application cost

Desktop Application cost:

Equipment	Cost per unit	Cost
Desktop pc (core i7 7th gen, 3.6 GHz processor gtx 1050ti gpu, 32 GB DDR4 RAM, 500GB SSD)	₹90000	₹90000
Web, File and Email servers	₹5000 per month	₹5000
Total		₹95000

Table 15: desktop application cost

The system will be made live hosting service and a domain name will be used and a mobile application will also be developed as it tends to provide portable accessibility and using the location service is easier.

Market Research Analysis Based on the Feasibility Factors:

Here the market and user demand is discussed. During the covid-19 outbreak people had to go into lockdown, at that time people had to move back to their permanent homes. And when they had to move back home problems arised. One of the problems is keeping the personal belongings safe with low cost. People had to pay extra rent for a whole flat to just keep their goods which was not affordable for them. To solve this problem storage units could play a vital role as renting a single storage unit is cost friendly. There has been a huge demand for renting storage and warehouses to store goods and belongings. To automate these, Self Storage can capitalize on the market as there is no such system that satisfies all the demands of the people. So the proposed system will provide a one stop solution for satisfying the rising demand.

6.2 Cost Benefit Analysis for Self Storage

Cost benefit analysis's objective is to calculate the estimated earnings against the expenses of developing the project. During the analysis expense and targeted earnings are compared to estimate the revenue the project may generate. Cost benefit analysis for the proposed system are given below for this project:

Total cost:

SL no	Equipm ent	Year 1	Year 2	Year 3	Year 4	Year 5	Total
1	Web based applicati on	₹10000 0					₹10000 0
2	Mobile applicati on	₹95000					₹95000
3	Desktop applicati on	₹95000					₹95000
4	server and hosting	₹5000	₹5000	₹5000	₹5000	₹5000	₹25000
5	Employe e expense	₹50000	₹50000	₹50000	₹50000	₹50000	₹25000 0
6	Other cost	₹20000	₹20000	₹20000	₹20000	₹20000	₹10000 0
	Total cost	₹36500 0	₹75000	₹75000	₹75000	₹75000	₹66500 0

Table 16: Total Cost Estimation for the project

Total Earning:

SL no	Sectors	Year 1	Year 2	Year 3	Year 4	Year 5	Total
1	Renting fees	₹180000	₹200000	₹220000	₹250000	₹280000	₹1130000
2	Warehouse Management Service	₹150000	₹190000	₹230000	₹270000	₹300000	₹1140000
	Total	₹3300000	₹3900000	₹4500000	₹5200000	₹5800000	₹22700000

*Table 17: Earning estimation for the project***Total Revenue:**

SL no	Sectors	Year 1	Year 2	Year 3	Year 4	Year 5	Total
1	Total Earning	₹3300000	₹3900000	₹4500000	₹5200000	₹5800000	₹22700000
2	Total Equipment	₹3650000	₹750000	₹750000	₹750000	₹750000	₹6650000
	Total revenue	-₹350000	₹3150000	₹3750000	₹4450000	₹5050000	₹16050000

Table 18: Estimated Revenue on a Five-year scale

From the cost benefit analysis it is clear that the system can generate a huge amount of profits if implemented correctly while also solving a problem for people.

6.3 Explain DSDM Good or Bad for this Project

Because this is an academic project the budget and time was an issue for this project. As it required the system to be complete with limited resources both technically and economically. Also, for an amateur project development like this requires a lot of going back and forth. Since DSDM supports iterative development the methodology is good for this project as it gives room for accepting and making changes. Along with its iterative development it ensures all the requirements are met for the system within the time frame and budget. So, it can be said that the DSDM is good for the project.

Chapter 7 – Foundation

7.1 The Problem Area Identification

The problem identification is the primary step towards the starting of any project. It can be done by the users as users are the ones that have to suffer because of the lack of such systems and tools. Problems can be identified through various techniques. My problem identification process is described below:

7.1.1 Interview

Interview is one of the most effective ways of gathering information regarding any problem. This allows us to discuss possible solutions to the problems as well. During the interview some of the requirements that have been identified are:

- Do not have storage related solution
- Existing renting solutions do not offer service regarding storage units and warehouses.
- No other system provide business solution

7.1.2 Observations

Observation is one of the most used problem identification techniques. I have seen this by observing the fact that when people needs to travel for a comparatively long time they have to pay extra house rent even though they are not living there. Some main points of observation are:

- No system that provides storage based service
- The idea of renting a storage unit to store personal belongings is relatively new
- People who need this kind of service often settle for service providers like Rakhibo.com where they do not have the complete control of their belongings.

7.1.3 Questionnaires

Questionnaire is considered the best technique to enquire about any system as it allows to ask very specific questions with a projected answer to the questions which is easy to analyze. Some example questionnaire are given below:

Name		Occupation	
Gender		address	
Do you have unused goods that you use occasionally?			
Would you like the idea of storing your personal belongings while traveling in a secured storage unit?			

Table 19: Questionnaire sample

Name		Business name	
Gender		address	
Do you use your own house as a warehouse for your business?			
Would you prefer a warehouse solution?			

Table 20: Questionnaire sample

7.2 Rich Picture

Rich picture is the bird's eye view of the activities of users of the proposed system. It represents the conflicting issues, communication and business process as well. The rich picture for Self Storage is given below:

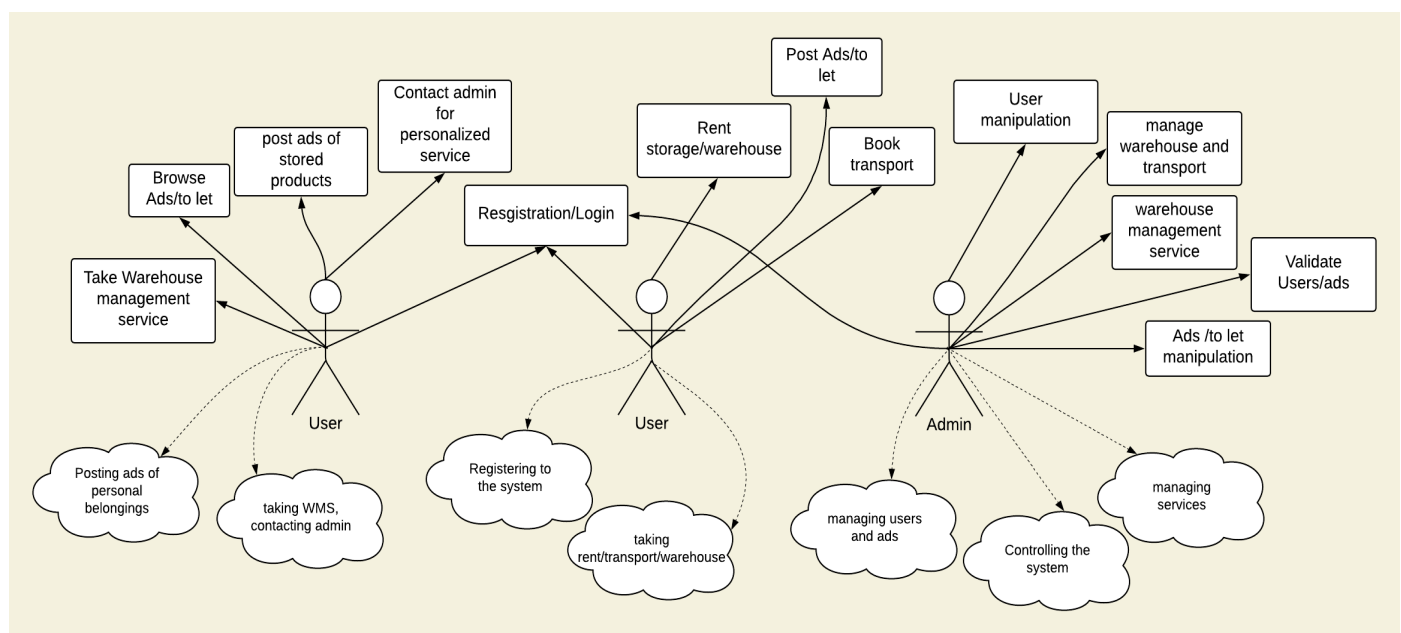


Figure 8: Rich Picture of the system

The legends of rich picture

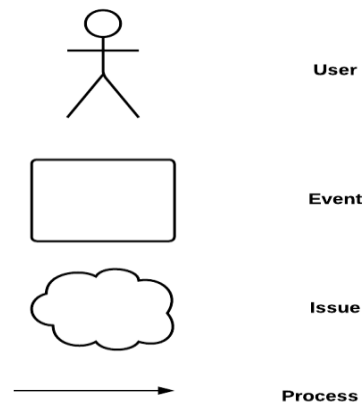


Figure 9: Legends of the rich picture

Key actors

- Users
- Admin

Short Description of Rich Picture:

The picture shows how a user can perform all the tasks from posting an ad to rent storage/warehouse and the admin can manage all the services of the system.

7.3 Specific Problem Area Identification

Problems are identified through using various information gathering techniques in the problem identification section. After analyzing these the identified specific problems are:

- There is no one stop service for renting systems.
- No criteria based search option
- No filtering option
- Lack of warehouse management service
- No transport service.

7.4 Possible Solutions

The finalized overall requirement list is:

Functional Requirements:

- Registration/login for the users
- Ads posting ability for the users
- Renting request management for the users
- Ads manipulation system for the admin
- Location, size and storage type authentication
- Warehouse management system for businesses
- Transport service for the users
- Notifications system for the the businesses about market situation
- Contact service for the users
- Preferred storage request option for the users
- Renting option for the stored belongings and goods on storage

Nonfunctional Requirements:

- Usability of the system
- Interactive system
- Responsiveness of the system
- Maintainability and scalability of the system
- Security of the system
- Backup server

7.5 Technology to be implemented

There are many feasible technologies for developing and implementing the system. Choosing the correct technology that is appropriate for the proposed system is very crucial for the sustainability of the project. The available technologies are:

Client-Server Application Technology

Client-server is a desktop-based technology. In this technology an application is required to be installed in users' desktop to use the system through which the user can access the system stored on server over the internet. (toastytech.com, 2019)

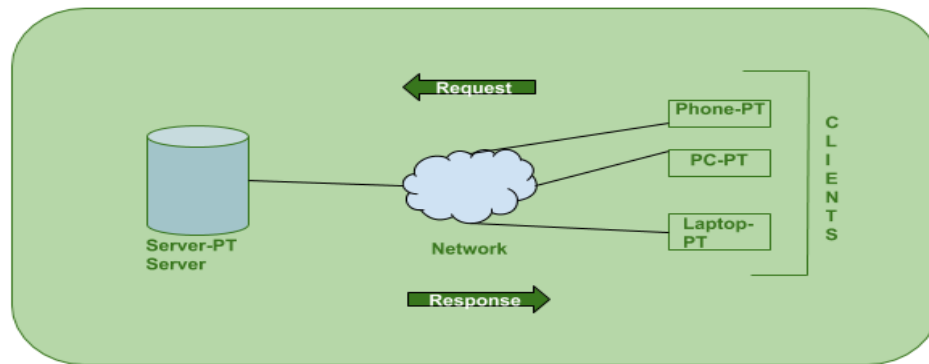


Figure 10: Client Server Application Model

Basic features of client-server application are listed as follows-

- Users require subscription to the client-server application
- The application needs frequent maintenance
- Not portable
- Follows a common communication protocol
- Respond to users following system priority

Web Application

Web applications are hosted in a remote server so it does not require the user to install anything. It can be accessed through the internet on any device with a browser. As it can be accessed from any device and location it is the best possible platform to develop the system. The features of web applications are as follows:

- Can be accessed from any device with internet and browser at any time
- No installation required
- No subscription or buying fees

Mobile Application:

Mobile applications need to be installed in every smartphone to get the service of the proposed system. As these days' smartphones are being used by almost every mobile user, developing mobile application of the proposed system is one of the efficient options for developing the system. The features of mobile applications are as follow:

- Portable
- Can be Accessed with internet with even low configuration phone and bandwidth

- Low cost.
- Portable and accessible at any time

7.6 Recommendation and Justification

The proposed system required location service and has to be portable so users can access it from anywhere without having to require extra hardware. As almost everyone has a smartphone people can access it from anywhere with full functionality and take advantage of location based services. So, mobile application is the recommended approach for the proposed system.

Chapter 8 – Exploration

8.1 Activity Diagram

User Activity Diagram

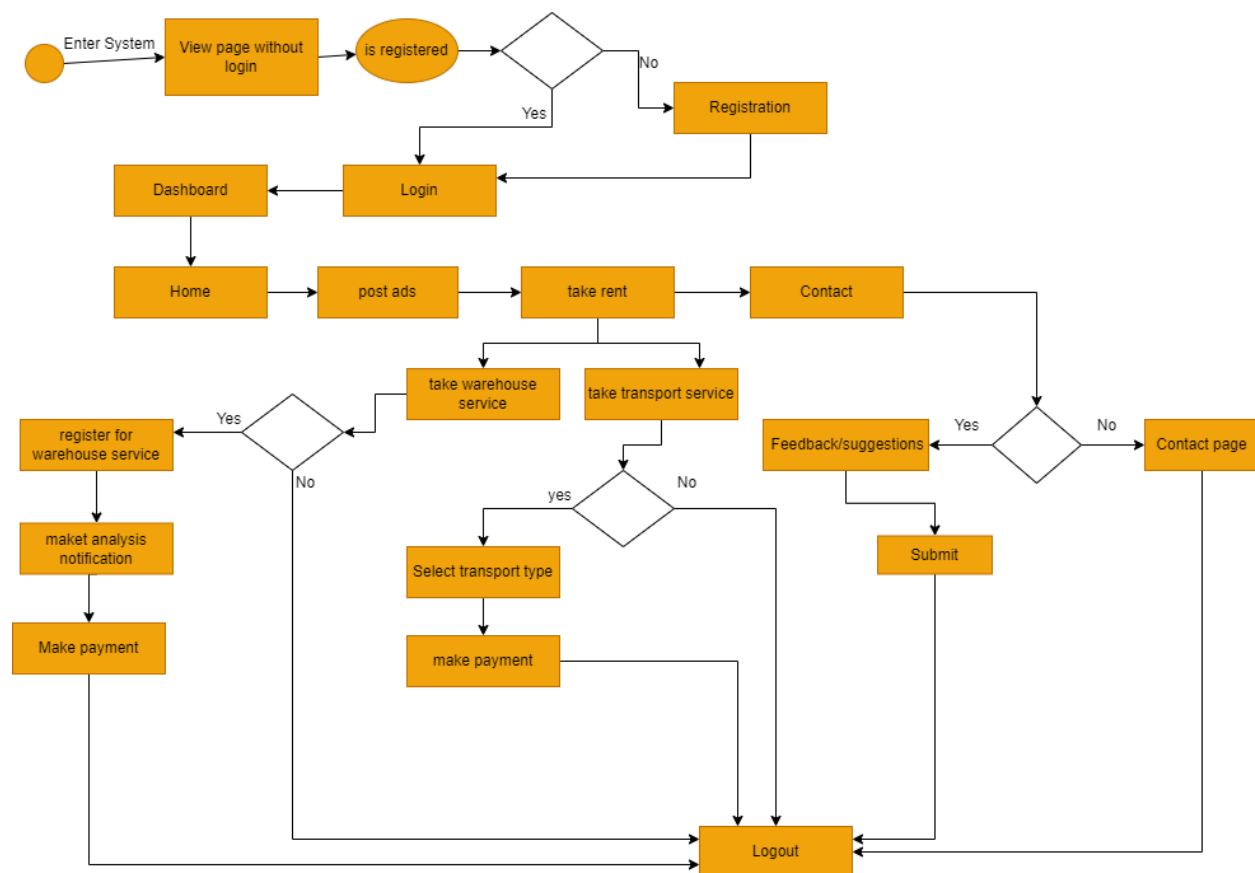


Figure 11: User Activity Diagram

8.2 Full System Use Case

The following is a use case diagram for the proposed system:

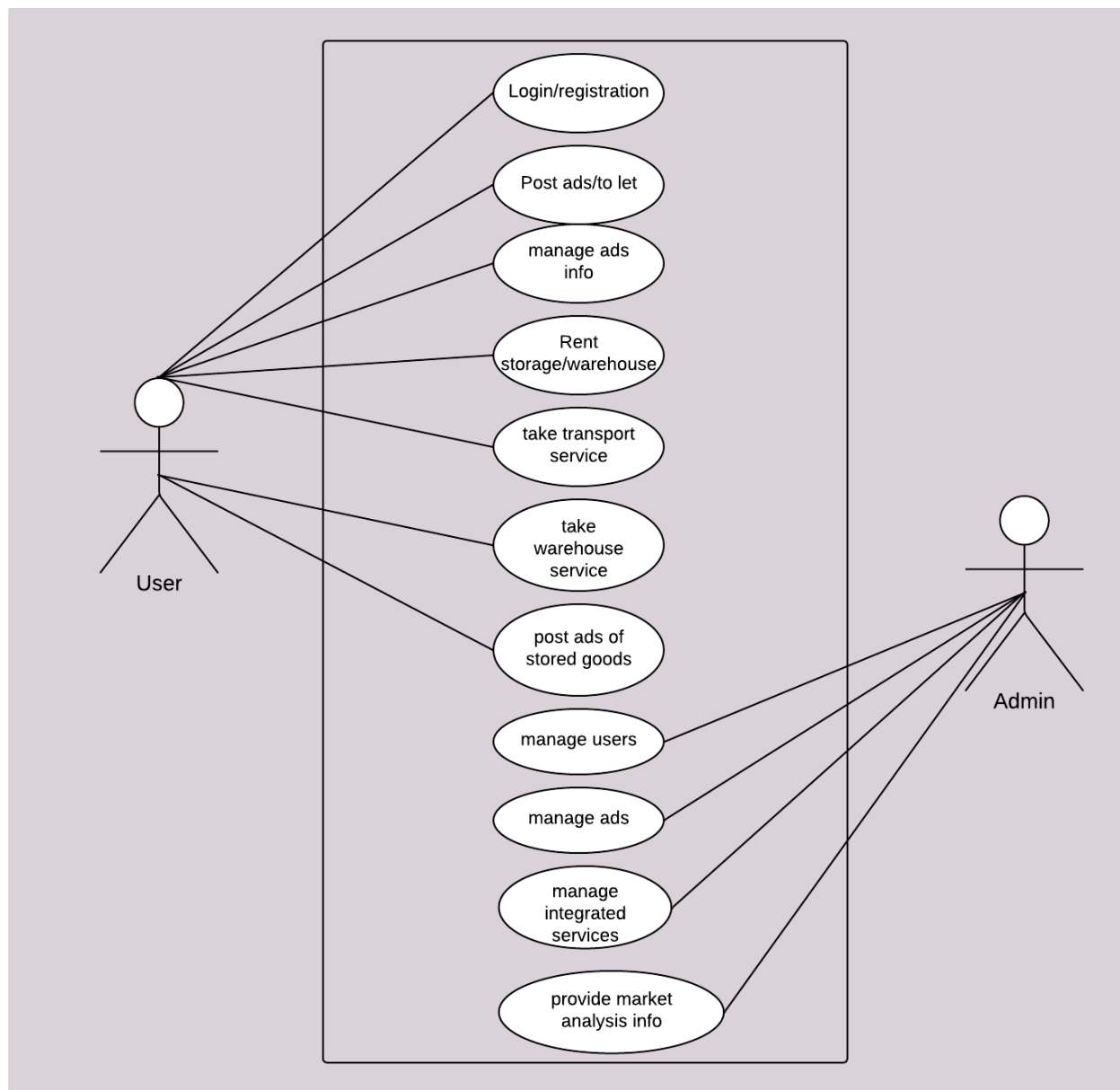


Figure 12: Use Case of the Proposed System

8.3 Full System Activity Diagram

There are two types of users in the planned system, each with their own workflow. The activity diagrams for the users workflow are given below.

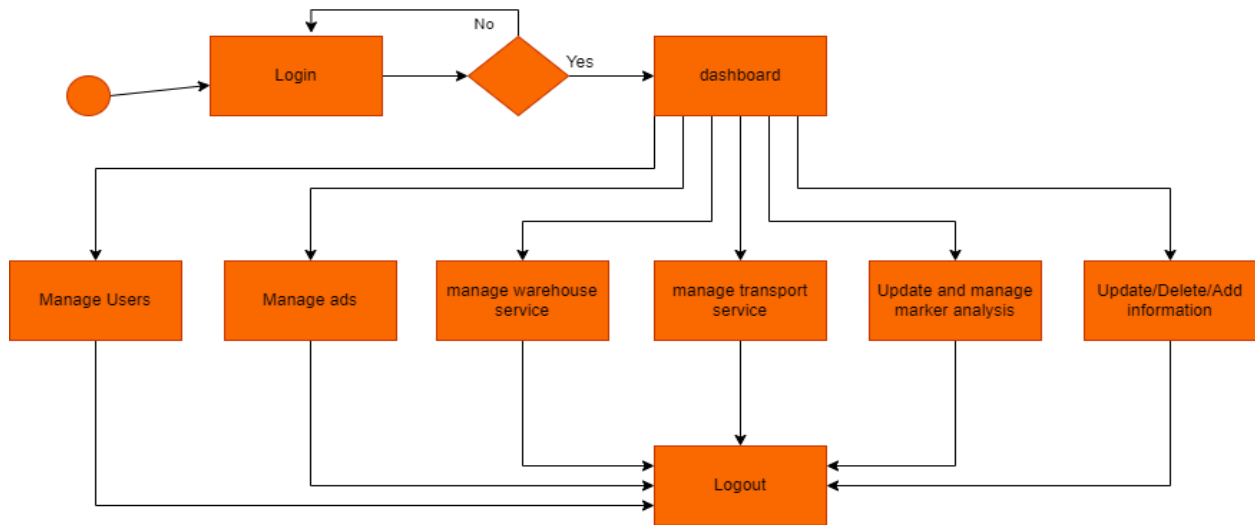


Figure 13: Admin Activity Diagram

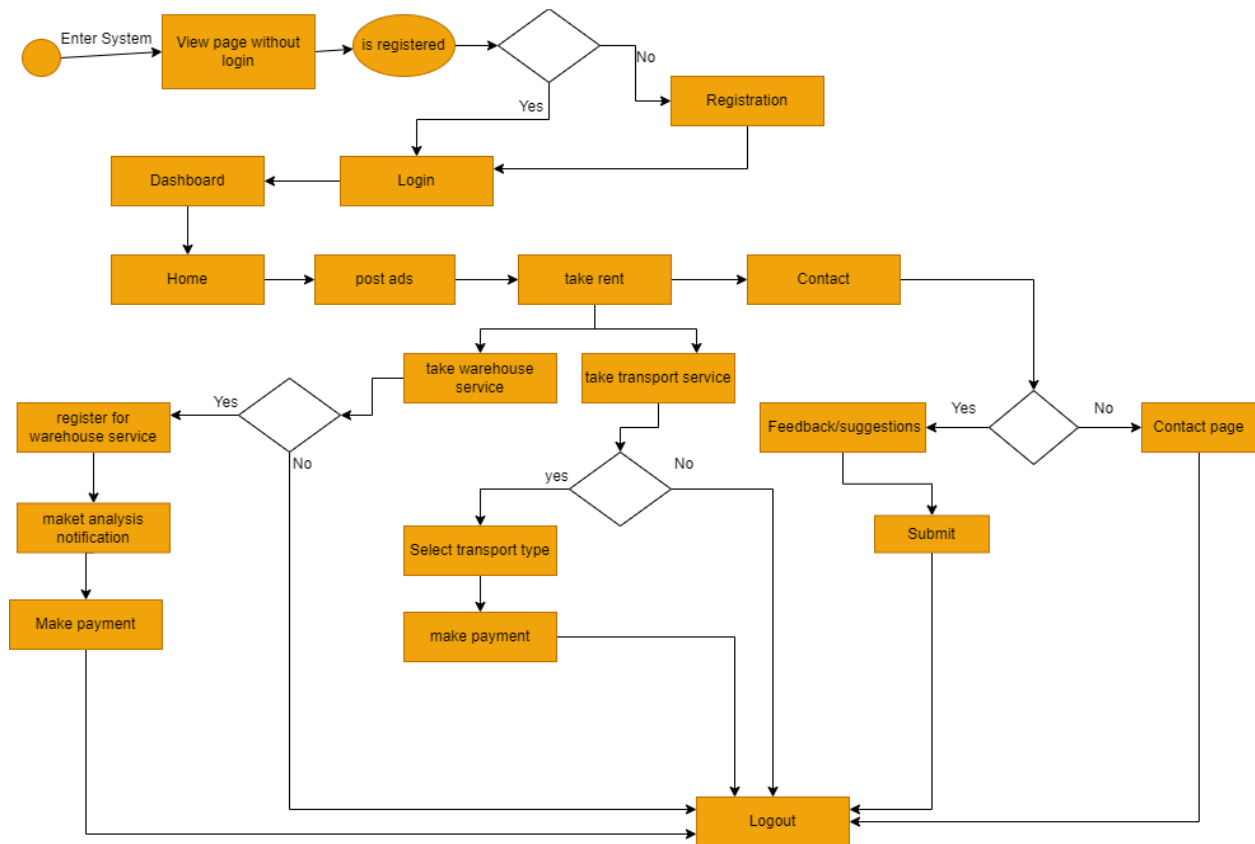


Figure 14: General User Activity Diagram

8.4 Catalogue of Requirements

Requirement catalog is a list containing all the identified requirements of a project. Here I am going to add the identified requirements to the requirement catalog following standard format.

Registration and login requirement catalog

Source:	Sign off:	Priority	Requirement ID:
Admin	All User	Must	M-001
Functional requirement			
Registration and login for users All types of the system users need to get registered and then login into the system. No one can perform an action without proper authentication.			
Non-functional Requirement			
Description	Target Value	Acceptance Value	Comment
Login/registration per day	10000	9000	

Table 21: Requirement Catalogue for login

User ads post catalog

Source:	Sign off:	Priority	Requirement ID:
users	user	Must	M-002
Functional requirement			
post ads users have to be able to post ads/to let			
Non-functional Requirement			
Description	Target Value	Acceptance Value	Comment
appointment booking	10000	9000	

Table 22: Requirement Catalogue for post ads

User storage renting catalog

Source:	Sign off:	Priority	Requirement ID:
users	user	Must	M-003
Functional requirement			
Rent storage users have to be able to rent storage from ads			
Non-functional Requirement			
Description	Target Value	Acceptance Value	Comment
appointment booking	10000	9000	

Table 23: Requirement Catalogue for storage renting

Search Requirement Catalog

Source:	Sign off:	Priority	Requirement ID:
users	user	Must	M-004
Functional requirement			
search users have to be able search desired storage according to their preferred location, size, cost etc.			
Non-functional Requirement			
Description	Target Value	Acceptance Value	Comment
appointment booking	10000	9000	

Table 24: Requirement Catalogue searching

Chapter 9 – Engineering

9.1 Approach New System Modules:

The proposed Self Storage contains several modules. Some of the core modules and their work procedures are presented below.

Module for logging in

SL	User Action	SL	System Action
1	User click on the login link	.1	login page will be shown
2	provide login info	2	credentials will be validated
3	click login	3	system validates the credentials and redirects to home page/user dashboard

Table 25: Module for logging in

Ads/to let posting Module.

SL	User Action	SL	System Action
1	User click on post ads	.1	a form is shown
2	provide ads info	2	the information are stored
3	click post	3	the ads will be posted with provided information

Table 26: Ads/to let posting Module

Search module:

SL	User Action	SL	System Action
1	User clicks search button	.1	storage/warehouse search page will show
2	User will search based on criteria and location	2	system will provide best matched option according to the search criteria
3	User will click on the available options	3	system will show storage/warehouse information
4	User will click take rent button	4	rent page will open where the system will show further steps to take rent

Table 27: Search Module

Renting Module:

SL	User Action	SL	System Action
1	User clicks on take rent	.1	storage/warehouse info are shown along with specifications
2	Users agrees to the terms	2	system will confirm the user's decision
3	User clicks payment	3	system opens payment gateway
4	User confirms payment	4	systems shows payment received message

Table 28: Renting Module.

9.2 Use Case Diagram of the System

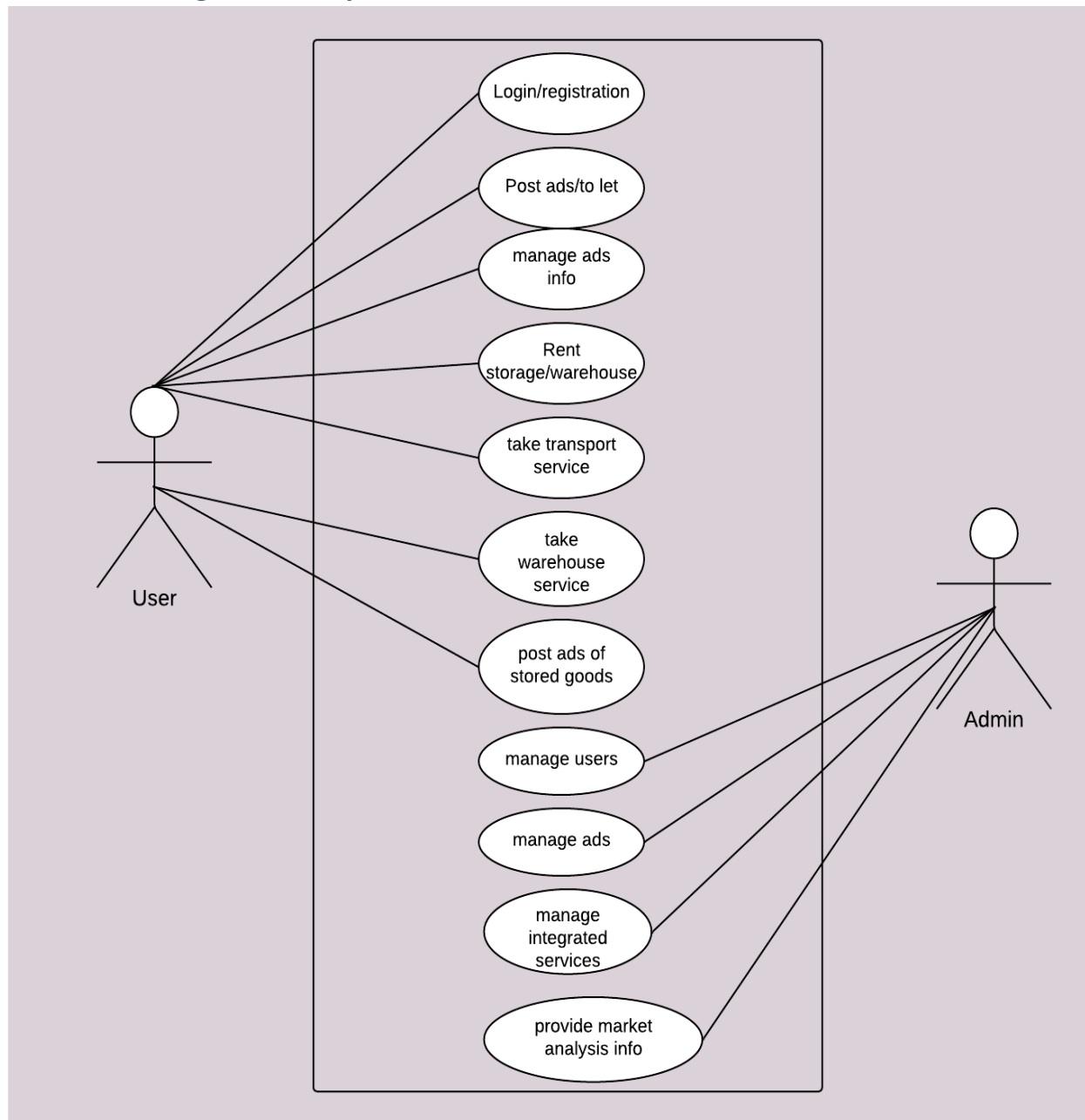


Figure 15:Use Case Diagram

9.3 Class Diagram of Self Storage

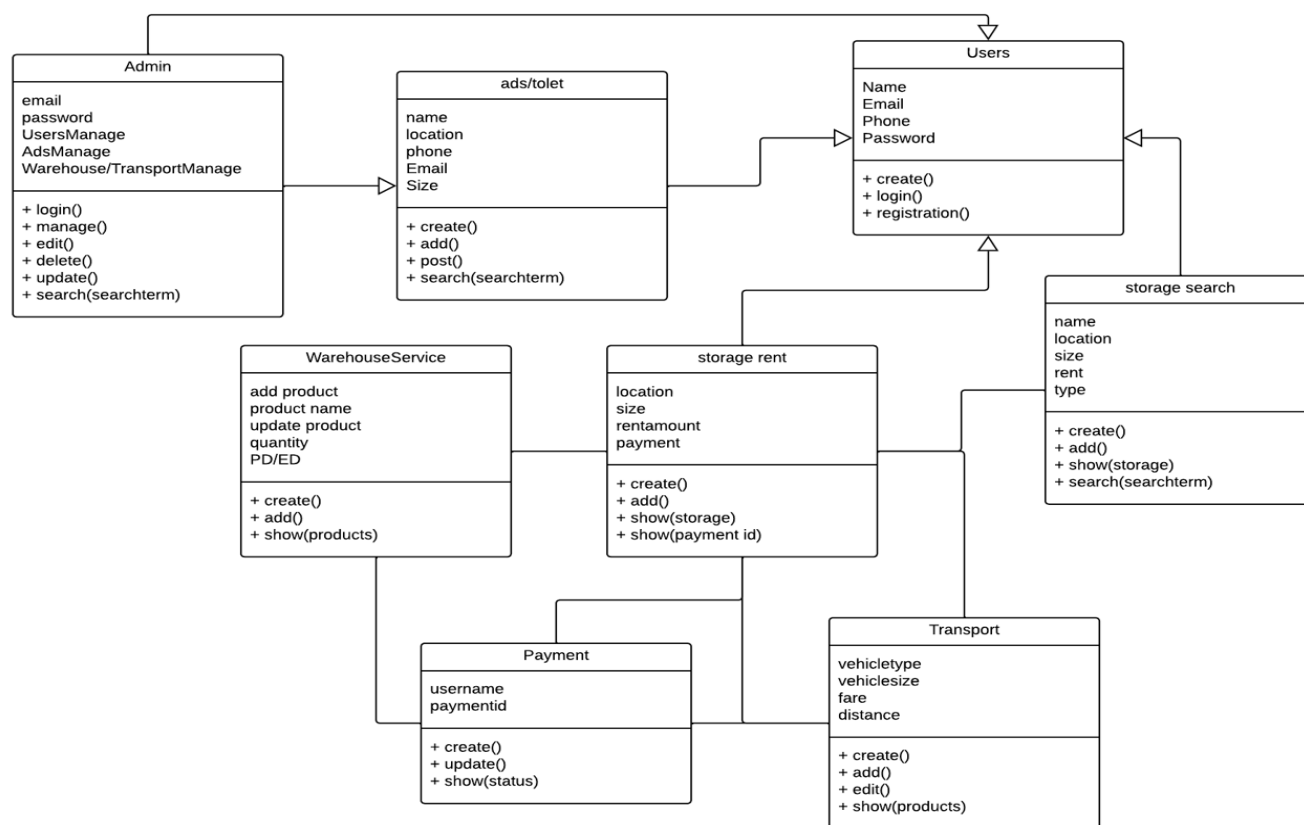


Figure 16:Class Diagram

9.4 Components of Class Diagram

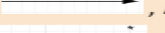
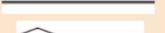
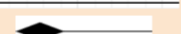
Attributes Characteristics	+, -, #, ~ these signs denotes the characteristics of attributes to public, private, protected and package respectively
Relationship	Each class has multiple relationships with multiple classe. These relationships are classified as Dependency :  , Generalization :  , Association :  , Multiplicity :  , Aggregation :  and Composition :  . Each relationship is represented by respective signs.
Class Name	Attributes
Users	Name; Email; Phone; Password;
Admin	email; password; UsersManage; AdsManage; Warehouse/TransportManage;
Ads/to let	name; location; phone; Email; Size;
Storage rent	location; size; rentamount; payment;
WarehouseService	add product; product name; update product; quantity; PD/ED;
Storage search	name; location; size; rent type;
Payment	username; paymentid;
Transport	<u>vehicletype</u> ; <u>vehiclesize</u> ; fare; distance;

Figure 17:Components of Class Diagram

9.5 Entity Relationship Diagram



Figure 18:ERD

9.6 Self Storage Sequence diagram

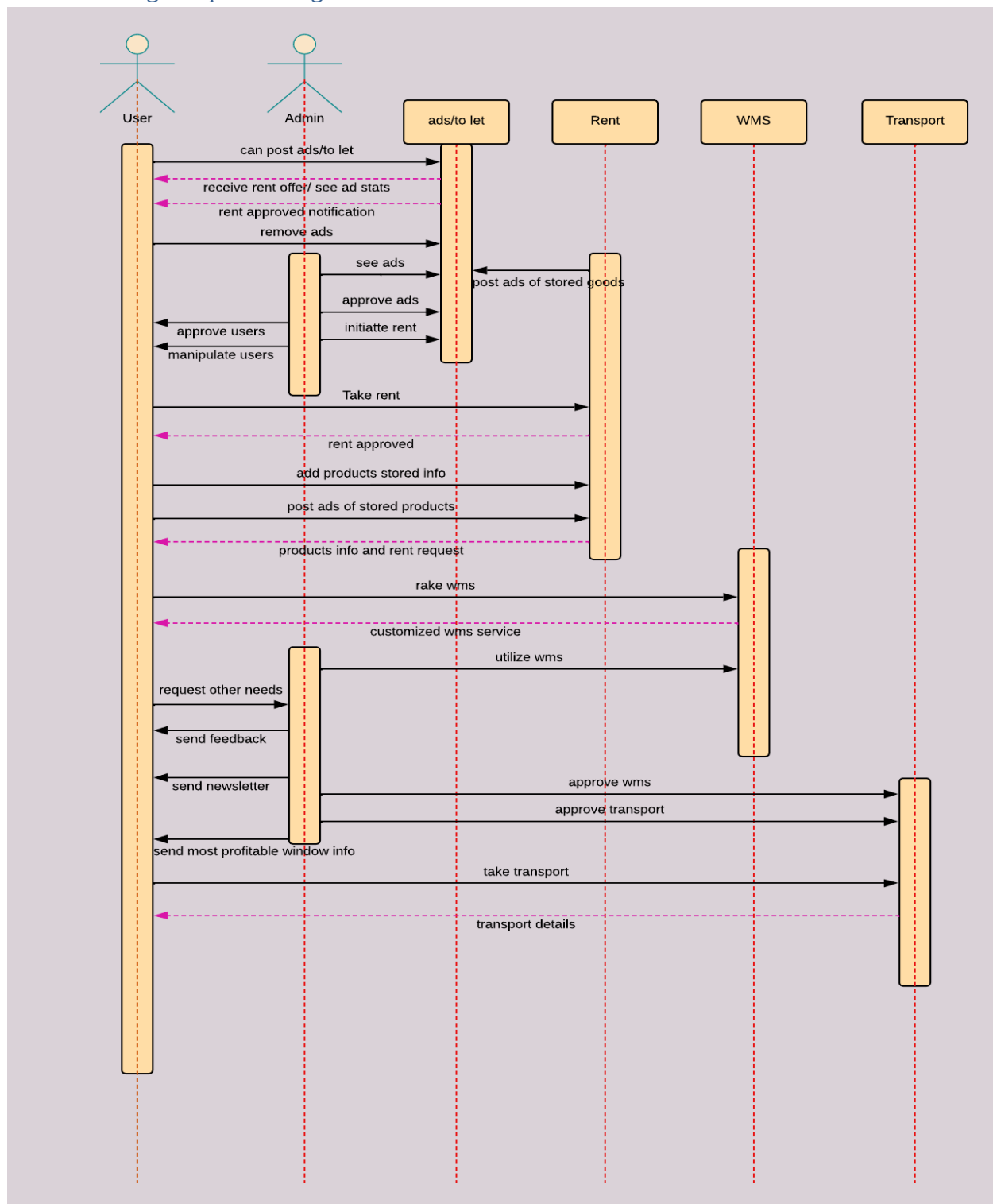


Figure 19: Sequence Diagram of Self Storage

9.7 Self Storage component diagram

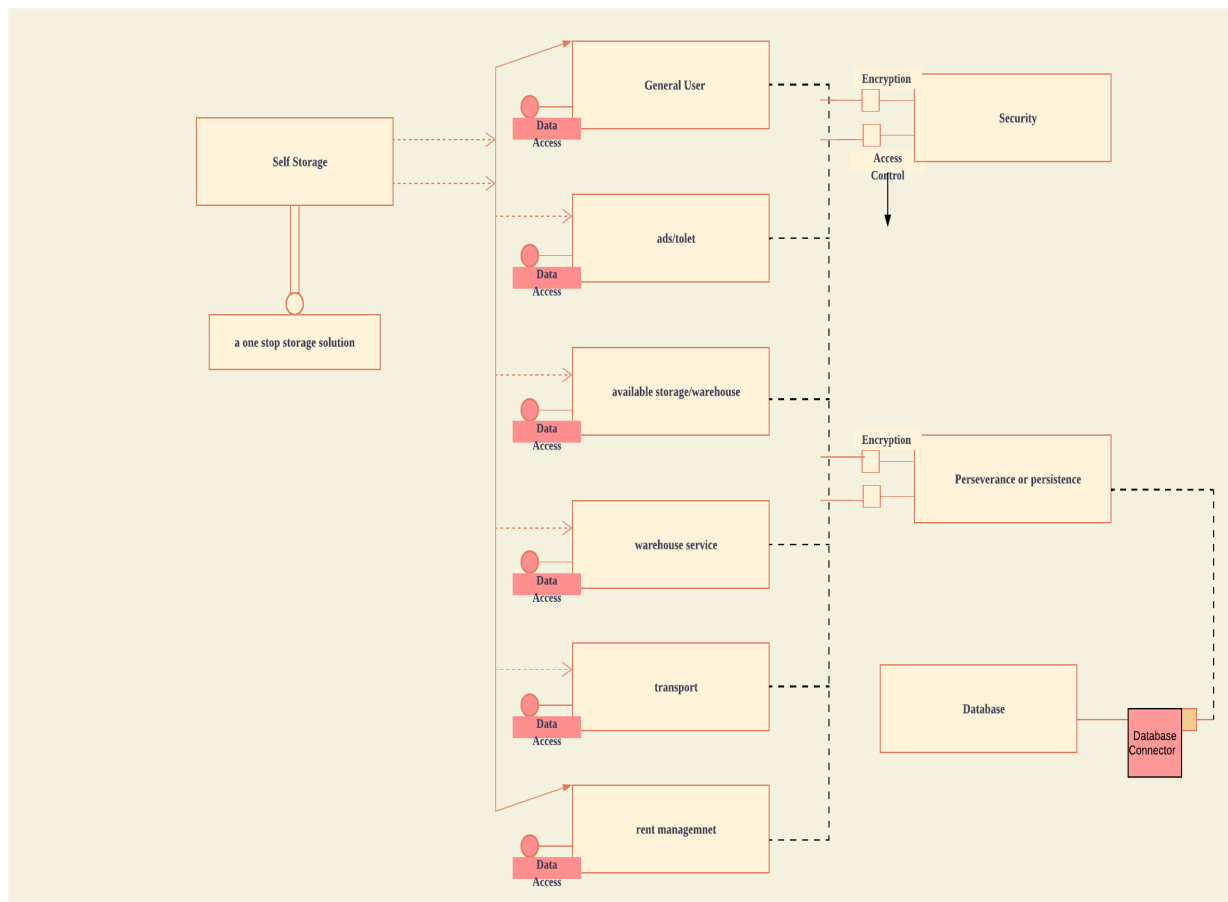


Figure 20: Component Diagram of Self Storage

9.8 Deployment Diagram of Self Storage

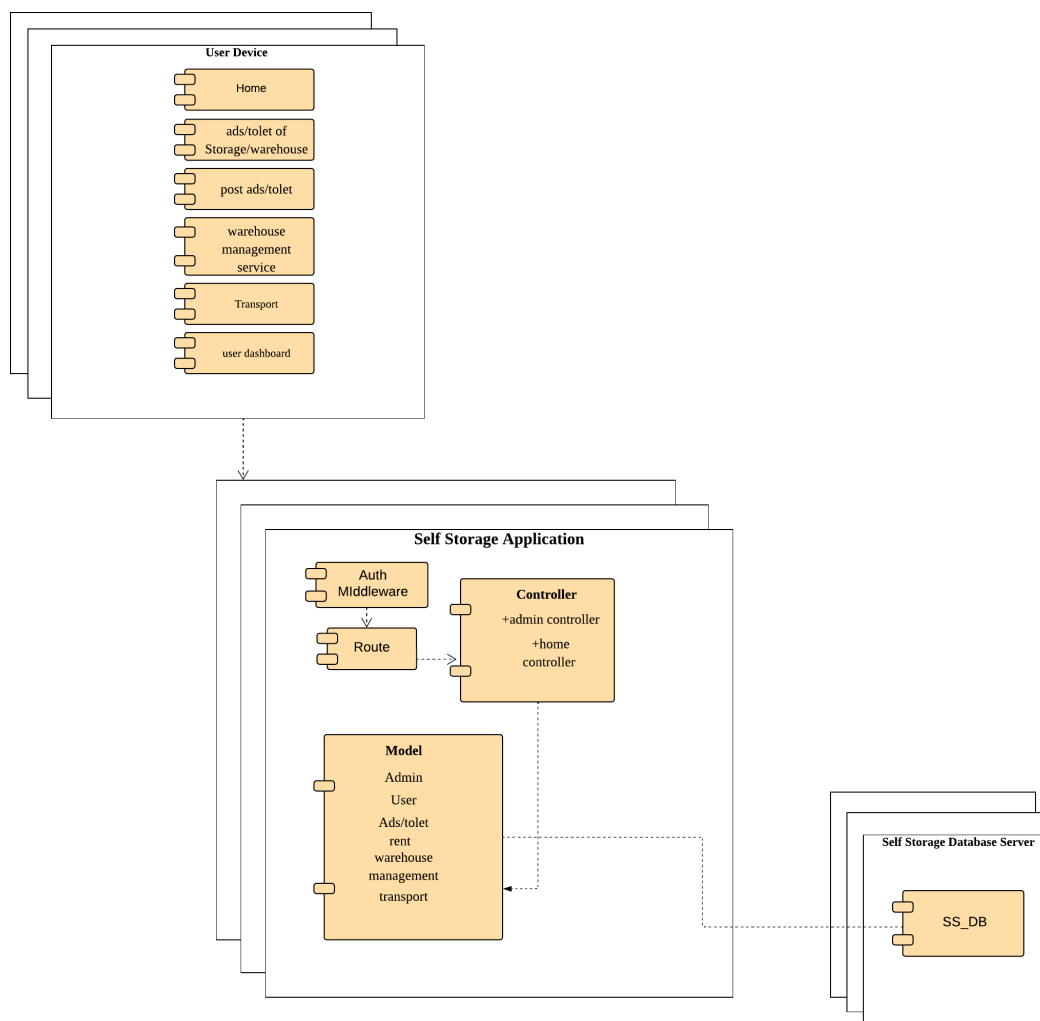


Figure 21:Deployment Diagram of Self Storage

9.9 System Interface Design

Home page Interface Design

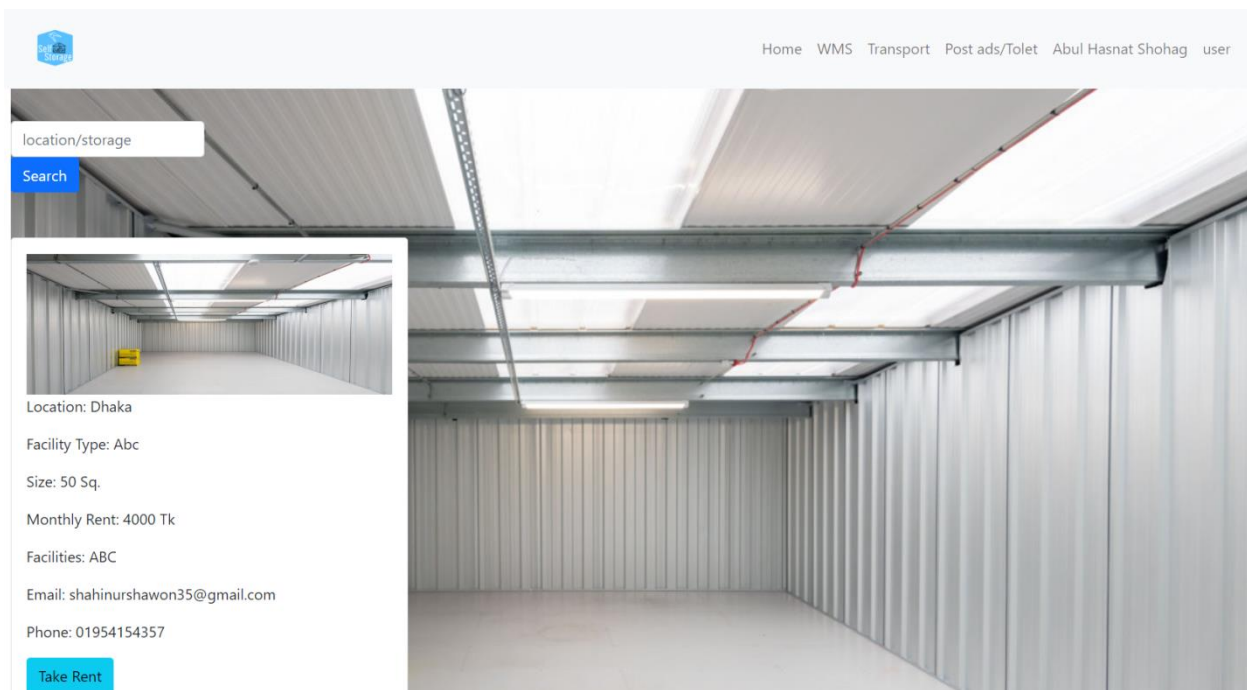


Figure 22:Interface design home page

Ads posting

Figure 23: ads posting

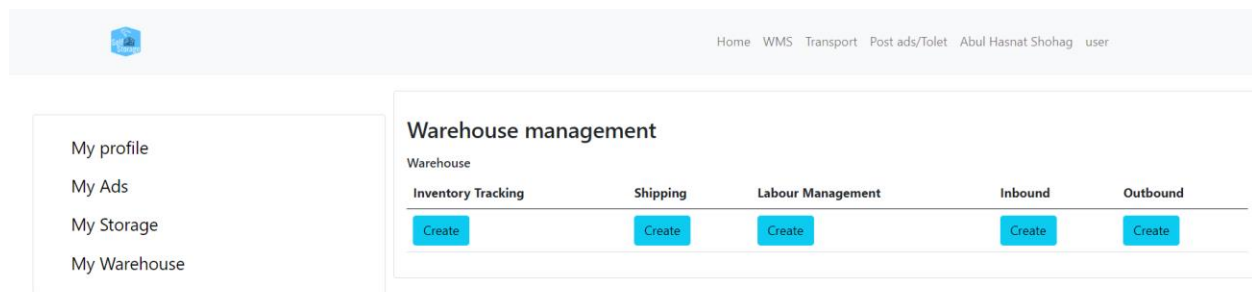


Figure 24:warehouse management service

Search storage

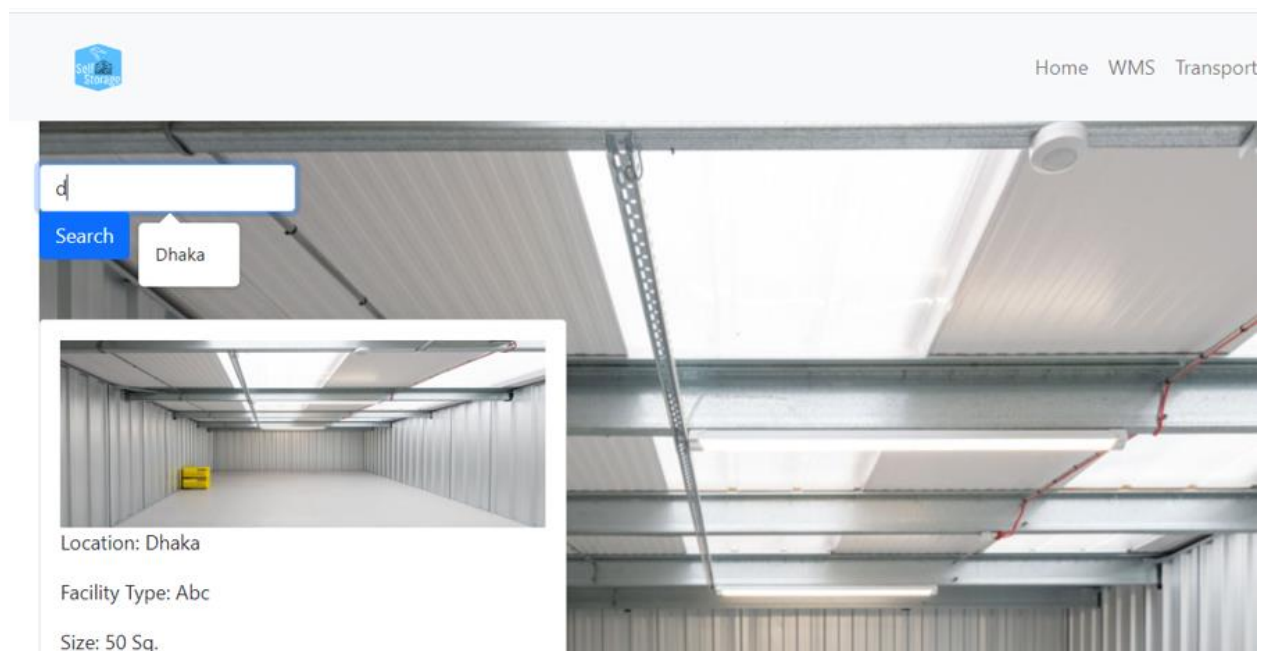
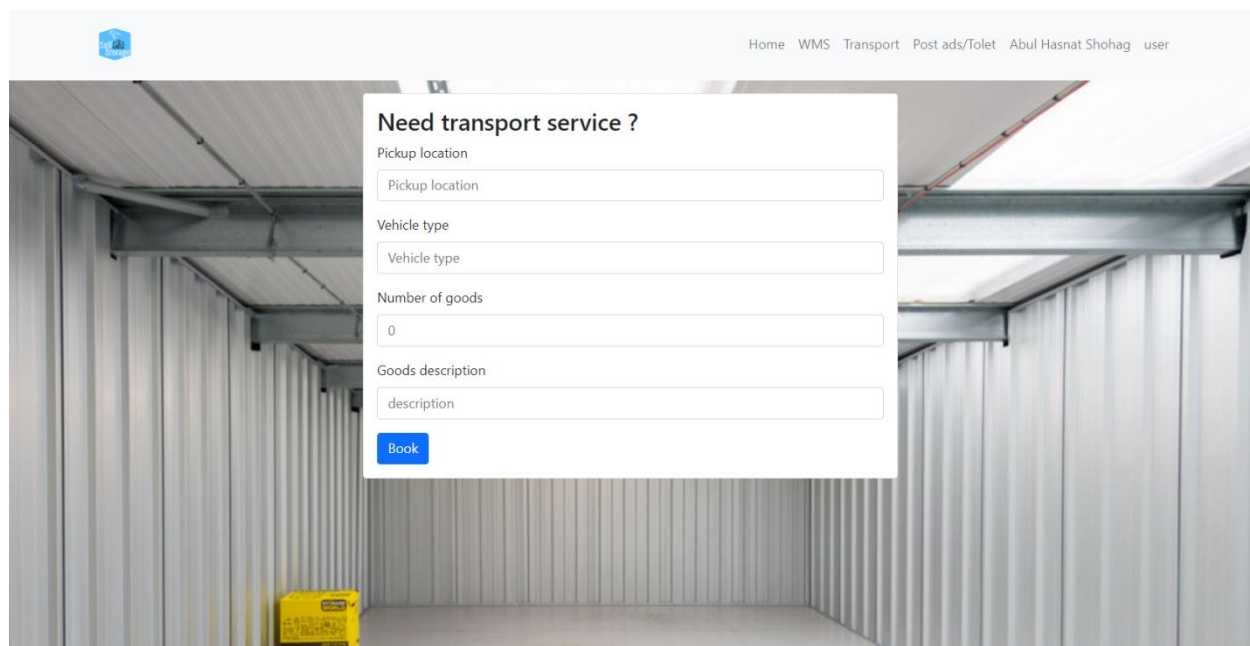


Figure 25:storage search interface

Transport service Page interface



Home WMS Transport Post ads/Toilet Abul Hasnat Shohag user

Need transport service ?

Pickup location

Vehicle type

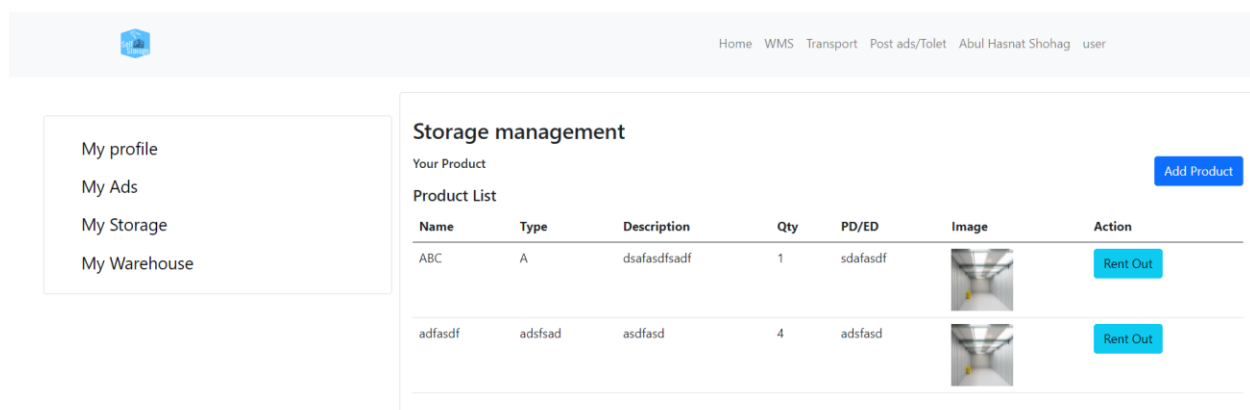
Number of goods

Goods description

[Book](#)

Figure 26:Transport service Page interface

User storage Interface



Home WMS Transport Post ads/Toilet Abul Hasnat Shohag user

My profile

My Ads

My Storage

My Warehouse

Storage management

Your Product [Add Product](#)

Product List

Name	Type	Description	Qty	PD/ED	Image	Action
ABC	A	dsafasdfsadf	1	sdafasdf		Rent Out
adfasdf	adsfsad	asdfsad	4	adsfasd		Rent Out

Figure 27:User storage Interface

User ads Interface



Home WMS Transport Post ads/Toilet Abul Hasnat Shohag user

[My profile](#)
[My Ads](#)
[My Storage](#)
[My Warehouse](#)

Ads management

Location	Facility Type	Size	Monthly Rent	Image	Facilities	Email	Phone
Dhaka	Abc	50 Sq.	4000 TK		ABC	shahinurshawon35@gmail.com	01954154357

Figure 28: User's ads Interface

Chapter 10 – Deployment

10.1 Core Module Coding Sample:

For the front-end development of the Self Storage, I have used HTML5, CSS, bootstrap and JavaScript. And for back-end development, I have used MySQL as a database management system and PHP Laravel. Here are some of the most crucial portion code examples.

```
resources > views > frontend > rent.blade.php
1  @extends('frontend.website_master')
2  @section('website')
3      <div class="bg-image" style="background-image: url('{{ asset('frontend/background.jpeg')}}')">
4          <div class="container">
5              <div class="row signup">
6                  <div class="col-md-6">
7                      @if(session('success'))
8                          <div class="alert alert-success alert-dismissible fade show" role="alert">
9                              <strong>{{ session('success') }}</strong>
10                             <button type="button" class="close" data-dismiss="alert" aria-label="Close">
11                                 <span aria-hidden="true">&times;</span>
12                             </button>
13                         </div>
14                     @endif
15                     <div class="card card-body">
16                         <h3></h3>
17                         <form action="{{ route('warehouse.registration') }}" class="signup-form" method="post">
18                             @csrf
19                             <div class="mb-3">
20                                 <label for="goods" class="form-label">Transaction ID</label>
21                                 <input
22                                     type="text"
23                                     class="form-control"
24                                     name="types_of_goods"
25                                     id="goods"
26                                     placeholder="Transaction ID"
27                                 />
28                             </div>
29
30                             <div class="mb-3">
31                                 <label for="description" class="form-label">
32                                     >Payment Amount</label>
33                                 <input
34                                     type="text"
35                                     class="form-control"
36                                     name="goods_description"
37                                     id="description"
38                                     placeholder="Payment Amount"
39                                 />
40                             </div>
41
42                             <div class="mb-3">
43                                 <label for="Purpose" class="form-label">Payment Method</label>
44                                 <input
45                                     type="text"
46                                     class="form-control"
47                                     name="purpose"
48                                     id="Purpose"
49                                     placeholder="Payment Method"
50                                 />
51                             </div>
52
53                             <button type="submit" class="btn btn-primary">Submit</button>
54                         </form>
55                     </div>
56                 </div>
57             </div>
58         </div>
59     </div>
60 @endsection
61
62
63
```

Figure 29:ads posting

```

resources > views > frontend > ads.blade.php
1 @extends('frontend.website_master')
2 @section('website')
3     <div class="bg-image" style="background-image: url('{{ asset('frontend/background.jpeg') }}')">
4     <div class="container">
5         <div class="row signup">
6             <div class="col-md-6">
7                 @if(session('success'))
8                     <div class="alert alert-success alert-dismissible fade show" role="alert">
9                         <strong>{{ session('success') }}</strong>
10                        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
11                            <span aria-hidden="true">&times;</span>
12                        </button>
13                    </div>
14                @endif
15                <div class="card card-body">
16                    <h3>Went to rent out your storage/warehouse ?</h3>
17                    <form action="{{ route('ads.store') }}" method="post" class="signup-form" enctype="multipart/form-data">
18                        @csrf
19                        <div class="mb-3">
20                            <label for="location" class="form-label">location</label>
21                            <input
22                                type="text"
23                                class="form-control"
24                                name="location"
25                                id="location"
26                                placeholder="location"
27                            />
28                        </div>
29
30                        <div class="mb-3">
31                            <label for="facility" class="form-label">
32                                >Type of facility</label>
33                            <input
34                                type="text"
35                                class="form-control"
36                                name="facility_type"
37                                id="facilityt"

```

```

resources > views > frontend > ads.blade.php
39                                placeholder="facility"
40                            />
41                        </div>
42                        <div class="mb-3">
43                            <label for="Size" class="form-label">Size</label>
44                            <input
45                                type="text"
46                                class="form-control"
47                                name="size"
48                                id="Size"
49                                placeholder=""
50                            />
51                        </div>
52
53                        <div class="mb-3">
54                            <label for="rent" class="form-label">Monthly rent</label>
55                            <input
56                                type="text"
57                                class="form-control"
58                                name="monthly_rent"
59                                id="rent"
60                                placeholder="rent"
61                            />
62                        </div>
63
64                        <div class="mb-3">
65                            <label for="pic" class="form-label">Picture</label>
66                            <input type="file" class="form-control" name="image" id="pic"/>
67                        </div>
68
69                        <div class="mb-3">
70                            <label for="Facilities" class="form-label">Facilities</label>
71                            <input
72                                type="text"
73                                class="form-control"
74                                name="facilities"
75                                id="Facilities"

```

```
resources > views > frontend > ads.blade.php
68
69
70 <div class="mb-3">
71   <label for="Facilities" class="form-label">Facilities</label>
72   <input
73     type="text"
74     class="form-control"
75     name="facilities"
76     id="Facilities"
77     placeholder="Facilities"
78   />
79 </div>
80
81 <div class="mb-3">
82   <label for="Email" class="form-label">Email</label>
83   <input
84     type="email"
85     class="form-control"
86     name="email"
87     id="Email"
88     placeholder="Email"
89   />
90 </div>
91
92 <div class="mb-3">
93   <label for="Phone" class="form-label">Phone</label>
94   <input
95     type="text"
96     class="form-control"
97     name="phone"
98     id="Phone"
99     placeholder="phone number"
100   />
101 </div>
102 <button type="submit" class="btn btn-primary">Save</button>
103
```

Figure 30: taking rent

```

resources > views > frontend > transport.blade.php
1  @extends('frontend.website_master')
2  @section('website')
3      <div class="bg-image" style="background-image: url('{{ asset('frontend/background.jpeg') }}')">
4          <div class="container">
5              <div class="row signup">
6                  <div class="col-md-6">
7                      @if(session('success'))
8                          <div class="alert alert-success alert-dismissible fade show" role="alert">
9                              <strong>{{ session('success') }}</strong>
10                             <button type="button" class="close" data-dismiss="alert" aria-label="Close">
11                                 <span aria-hidden="true">&times;</span>
12                             </button>
13                         </div>
14                     @endif
15                     <div class="card card-body">
16                         <h3>Need transport service ?</h3>
17                         <form action="{{ route('transport.registration') }}" class="signup-form" method="post">
18                             @csrf
19                             <div class="mb-3">
20                                 <label for="Pickup" class="form-label">Pickup location</label>
21                                 <input type="text"
22                                     class="form-control" name="location" id="Pickup" placeholder="Pickup location">
23                             </div>
24                             <div class="mb-3">
25                                 <label for="type" class="form-label">Vehicle type</label>
26                                 <input type="text"
27                                     class="form-control" name="vehicle_type" id="type" placeholder="Vehicle type">
28                             </div>
29                             <div class="mb-3">
30                                 <label for="goods" class="form-label">Number of goods</label>
31                                 <input type="text"
32                                     class="form-control" name="number_of_goods" id="goods" placeholder="0">
33                             </div>
34                             <div class="mb-3">
35                                 <label for="description" class="form-label">Goods description</label>
36                                 <input type="text"
37                                     class="form-control" name="goods_description" id="description" placeholder="description">
38                             </div>
39                             <button type="submit" class="btn btn-primary">Book</button>
40                         </form>
41                     </div>
42                 </div>
43             </div>
44         </div>
45     </div>
46 @endsection

```

Figure 31:transport

```

resources > views > frontend > user_warehouse.blade.php
1  @extends('frontend.website_master')
2  @section('website')
3      <div class="user">
4          <div class="col-md-4">
5              <div id="mySidenav" class="sidenav">
6                  <div class="card card-body">
7                      <a href="{{ route('web.user.profile') }}">My profile</a>
8                  </div>
9                  <a href="{{ route('web.user.ads') }}">My Ads</a>
10                 <a href="{{ route('web.user.product') }}">My Storage</a>
11                 <a href="{{ route('web.user.warehouse') }}">My Warehouse</a>
12             </div>
13         </div>
14         <div class="col-md-8" id="main">
15             <div class="card card-body">
16                 <h3 class="my-3">Warehouse management</h3>
17                 <div class="d-flex justify-content-between">
18                     @if(session('success'))
19                         <div class="alert alert-success alert-dismissible fade show" role="alert">
20                             <strong>{{ session('success') }}</strong>
21                             <button type="button" class="close" data-dismiss="alert" aria-label="Close">
22                                 <span aria-hidden="true">&times;</span>
23                             </button>
24                         </div>
25                     @endif
26                     <h6>Warehouse</h6>
27                 </div>
28                 <table class="table">
29                     <thead>
30                         <tr>
31                             <th scope="col">Inventory Tracking</th>
32                             <th scope="col">Shipping</th>
33                             <th scope="col">Labour Management</th>
34                             <th scope="col">Inbound</th>
35                             <th scope="col">Outbound</th>
36                         </tr>
37                     </thead>

```

Figure 32:user storage

```

resources > views > frontend > user_ads.blade.php
1  @extends('frontend.website_master')
2  @section('website')
3      <div class="user">
4          <div class="col-md-4">
5              <div id="mySidenav" class="sidenav">
6                  <div class="card card-body">
7                      <a href="{{ route('web.user.profile') }}">My profile</a>
8                  </div>
9                      <a href="{{ route('web.user.ads') }}">My Ads</a>
10                     <a href="{{ route('web.user.product') }}">My Storage</a>
11                     <a href="{{ route('web.user.warehouse') }}">My Warehouse</a>
12                 </div>
13             </div>
14         </div>
15         <div class="col-md-8" id="main">
16             <div class="card card-body">
17                 <h3 class="my-3">Ads management</h3>
18                 <table class="table">
19                     <thead>
20                         <tr>
21                             <th scope="col">Location</th>
22                             <th scope="col">Facility Type</th>
23                             <th scope="col">Size</th>
24                             <th scope="col">Monthly Rent</th>
25                             <th scope="col">Image</th>
26                             <th scope="col">Facilities</th>
27                             <th scope="col">Email</th>
28                             <th scope="col">Phone</th>
29                         </tr>
30                     </thead>
31                     <tbody>
32                         @foreach($ads as $row)
33                             <tr>
34                                 <td class="">{{ $row->location }}</td>
35                                 <td class="">{{ $row->facility_type }}</td>
36                                 <td class="">{{ $row->size }} Sq.</td>
37                                 <td class="">{{ $row->monthly_rent }} TK</td>
38                                 <td class="">
39                                     <a href="{{ asset($row->image) }}">
40                                         
42                                     </a>
43                                 </td>
44                                 <td class="">{{ $row->facilities }}</td>
45                                 <td class="">{{ $row->email }}</td>
46                                 <td class="">{{ $row->phone }}</td>
47                             </tr>
48                         @endforeach
49                     </tbody>
50                 </table>
51             </div>
52         </div>
53     </div>
54 @endsection
55
resources > views > frontend > user_ads.blade.php
34 @foreach($ads as $row)
35     <tr>
36         <td class="">{{ $row->location }}</td>
37         <td class="">{{ $row->facility_type }}</td>
38         <td class="">{{ $row->size }} Sq.</td>
39         <td class="">{{ $row->monthly_rent }} TK</td>
40         <td class="">
41             <a href="{{ asset($row->image) }}">
42                 
44             </a>
45         </td>
46         <td class="">{{ $row->facilities }}</td>
47         <td class="">{{ $row->email }}</td>
48         <td class="">{{ $row->phone }}</td>
49     </tr>
50 @endforeach
51 </tbody>
52 </table>
53 </div>
54 </div>
55 @endsection
56

```

Figure 33: renting out personal goods

10.2 Problem Breakdown

The project is divided into smaller tasks to make the development process easy and efficient. The proposed project can be broken down into the following tasks:

- Database design and analysis
- Ads posting feature
- Renting service
- User management
- Warehouse management & Transport service

Database design and analysis

- Identify the requirements.
- Identify the required information for the system
- Gather and normalize the required information
- Develop ERD and data dictionary
- Create database schema based on the data dictionary

Ads posting feature

- Create a form for ads specification
- Front-end design.
- Develop a response system when users show interest to the ads

Renting service

- Front-end design, Registration/login system
- Search system development based on user criteria such as location, size, fare etc.
- Payment system

User management

- Front-end design, Registration/login system
- User dashboard development
- Ads posting and rent management system development

- Warehouse management system

Warehouse management & Transport service

- Front-end and authentication system
- Personalized warehouse management service
- Transport booking system

10.3 Prioritization while Developing the Solution

Requirements are already prioritized for identifying the core functionalities of the system. The smaller divided tasks need prioritizing before going into the development process. Because sometimes once the development starts some functionality may slip out. The prioritized development tasks list is given below:

- Database design and analysis
- Database development
- Registration/login system development
- User management implementation
- Ads posting implementation
- Renting system implementation
- Search and filtering system implementation
- Warehouse management service implementation
- Transport service implementation

Chapter 11 – Testing

11.1 Test Plan Acceptance

Testing is important for the development of a successful software product. With all the other development process a testing plan should also be developed during the initial study. The testing plan should be agreed upon both the developers and users. The test plan of the project will define the testing requirements and acceptance criteria of the user. The two primary types of testing are:

Functional Testing

Functional testing in this project will include three types of testing criteria. They are:

Unit Testing

- Validation of input fields on the form.
- Ads posting ability of user.
- Storage filtering.
- Warehouse and transport service management.

Module Testing

- Registration/login attempt without entering data.
- Registering with invalid data.
- Providing proper data.

Integration Testing

- Login to the system with valid credentials.
- User posting an ad/to let.
- Successful storage rent.

Non-functional Testing

Non-functional testing in this project will include four types of testing:

Acceptance Testing

- Renting request will be shown.
- Updated ads/to let.

Security Testing

- Separate dashboard for individual users.
- Login attempt with invalid credentials.

Accessibility Testing

- Testing for user friendliness.
- Color and contrast testing by physically impaired person.

Usability Testing

- Testing admin panel.
- Testing with general users.

11.2 Test Case

Following test plan, test cases needs to be developed. The test cases for the proposed system are given below:

Unit test – test case:

Test Case Name	Unit Test		
Test Class			
Test Description			
Data Source	Test Steps	Expected Results	Actual Results

Module Test – test case:

Test Case Name	Module test		
Test Class			
Test Description			
Data Source	Test Steps	Expected Results	Actual Results

Integration Testing – test case:

Test Case Name	Integration Test		
Test Class			
Test Description			
Data Source	Test Steps	Expected Results	Actual Results

11.3 Unit Testing

Unit Test one

Test Case

Name of the test case	Unit testing		
Test Class	User Registration Controller		
Description of the test	Name validation for user registration		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1.complete all the information except user name 2, fill up and submit the form	user's name should not be empty warning will be shown	a message appears saying user's name is required to register

Figure 34: unit test one test case

User name validation

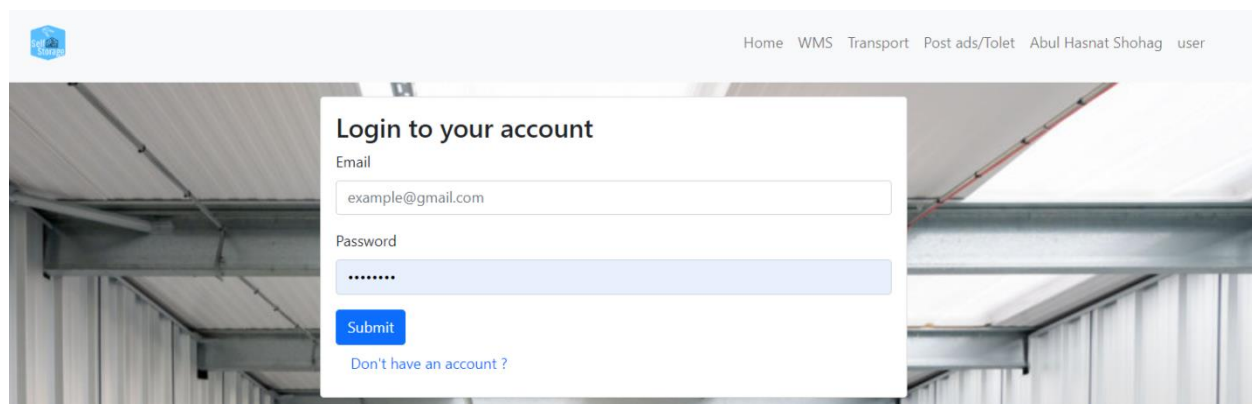


Figure 35:Unit test one result

Unit Test two

Test Case

Name of the test case	Unit testing		
Test Class	admin Controller		
Description of the test	all feature validation		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1.Login as admin	all function should be <u>displayed</u>	showing all the functions
	2, check all the available admin functionality		

List of Features control by admin

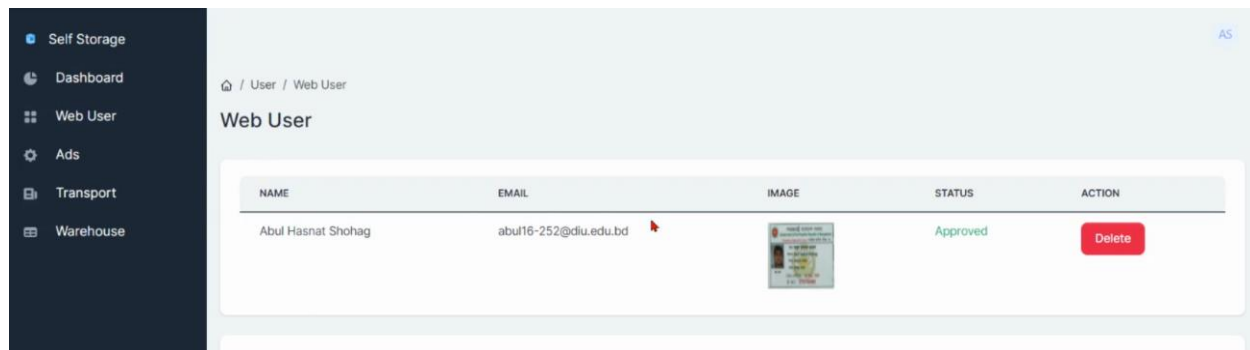


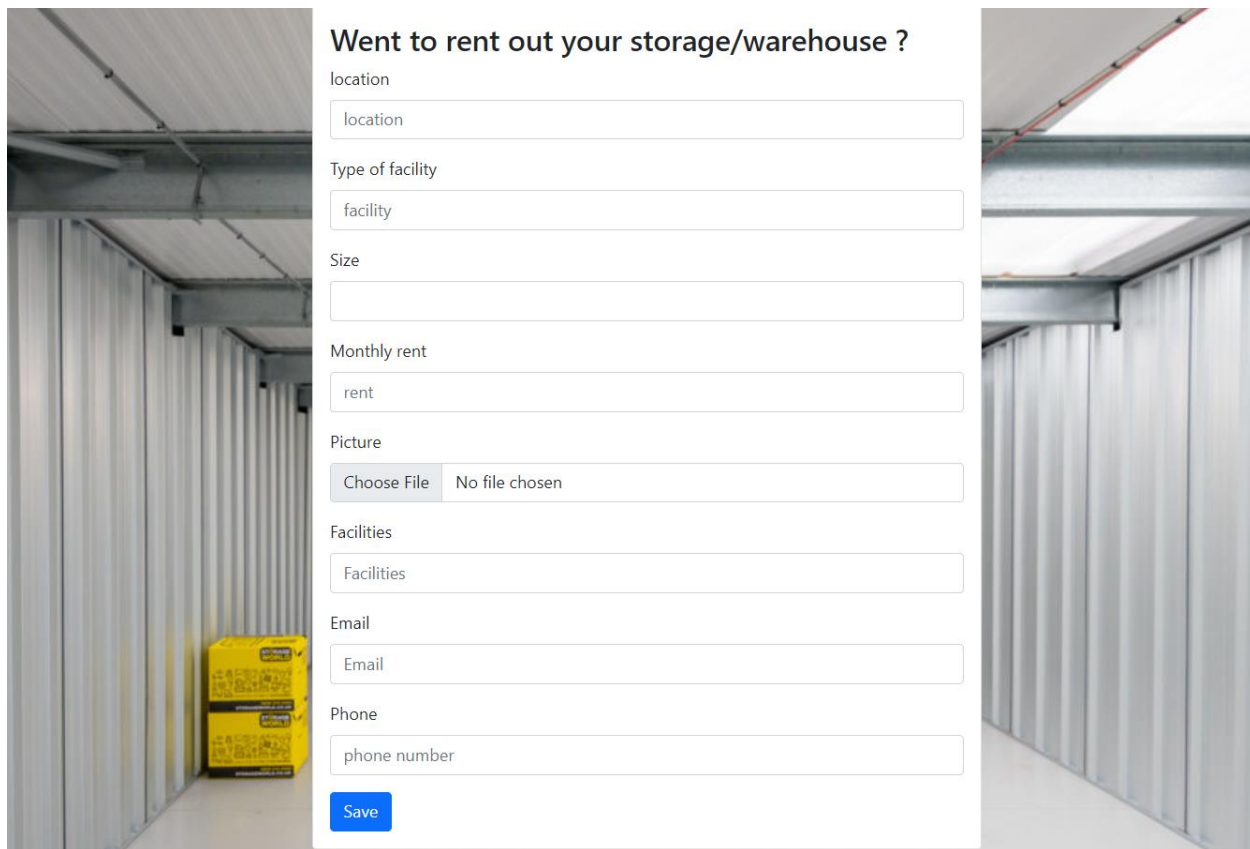
Figure 36: Unit Test two

Unit Test three

Test Case

Name of the test case	Unit testing		
Test Class	ads posting		
Description of the test	validation of ads posting		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1.go to ads posting	ads should be seen in home	showing ads in home page
	2, fill up required info and submit		

Ads posting



Went to rent out your storage/warehouse ?

location

Type of facility

Size

Monthly rent

Picture
 No file chosen

Facilities

Email

Phone

Figure 37:Unit Test three

11.4 Module Testing

Module Test one

Test Case

Name of the test case	Module Test		
Test Class	admin controller		
Description of the test	admin can manipulate users, ads		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1.login as system	admin should see options to update information	admin can see options to update and delete ads and users
	2, see dashboard		

admin dashboard

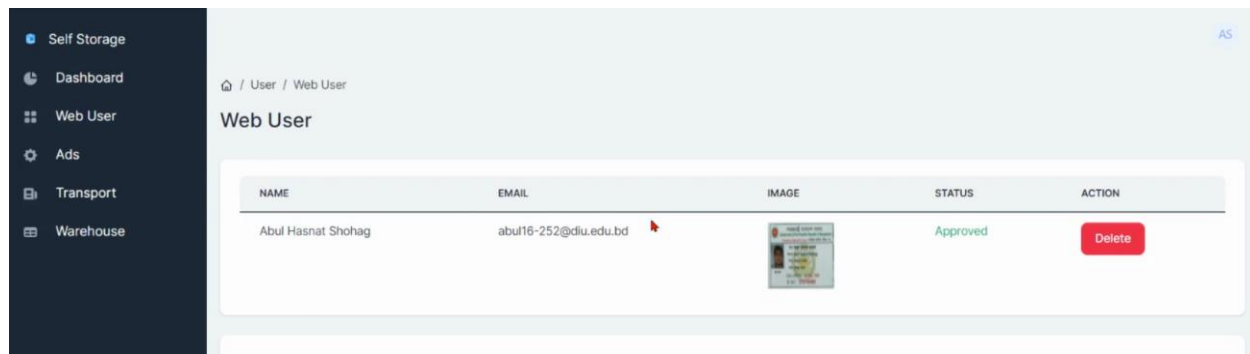


Figure 38:Module Test one

11.5 Integration Testing

Integration Test one

Test Case

Name of the test case	Integration test		
Test Class	login controller		
Description of the test	login successful and redirected to home		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1.go to login page	user should be routed to home page	user is redirected to home page
	2, provide credentials and click login		

Login system integration test

The image displays two screenshots of a web application interface. The top screenshot shows a login form titled "Login to your account" overlaid on a background image of a storage unit. The form includes fields for "Email" (containing "abul16-252@diu.edu.bd") and "Password" (masked with dots), a "Submit" button, and a link "Don't have an account?". The bottom screenshot shows a search results page. At the top, there is a search bar with the text "location/storage" and a "Search" button. Below the search bar, a thumbnail image of a storage unit is shown. To the right of the thumbnail, the following details are listed: "Location: Dhaka", "Facility Type: Abc", "Size: 50 Sq.", "Monthly Rent: 4000 Tk", "Facilities: ABC", "Email: shahinurshawon35@gmail.com", and "Phone: 01954154357". A "Take Rent" button is located at the bottom of this section. The background of the bottom screenshot is a large image of a storage unit interior.

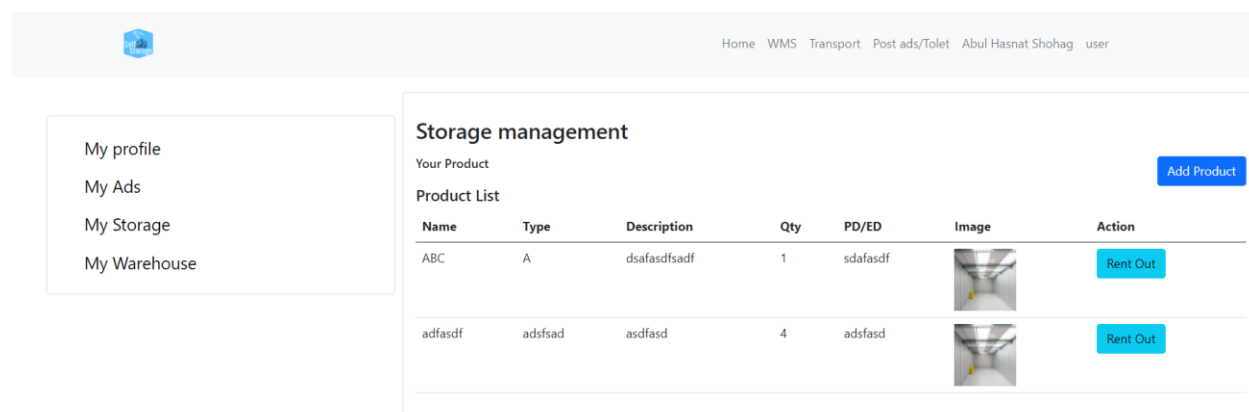
Figure 39: Integration Test one

Integration Test two

Test Case

Name of the test case	Integration test		
Test Class	ads posting		
Description of the test	users can post ads for their storage		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1.visit ads/to let	ads should be posted	the system will show the ad at home
	2, fill up info and submit		

User can rent out goods from their storage



Home WMS Transport Post ads/Tolet Abul Hasnat Shohag user

My profile
My Ads
My Storage
My Warehouse

Storage management

Your Product [Add Product](#)

Product List

Name	Type	Description	Qty	PD/ED	Image	Action
ABC	A	dsafasdfsadf	1	sdafasdf		Rent Out
adfasdf	adsfsad	asdfsad	4	adfsad		Rent Out

Figure 40: Integration Test two

11.6 Acceptance Testing

Acceptance Test one

Test Case

Name of the test case	Acceptance test		
Test Class	user request		
Description of the test	admin will approve users		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1.go to dashboard	user approved	user approved 
	2, approve user		

Answer found acceptance testing

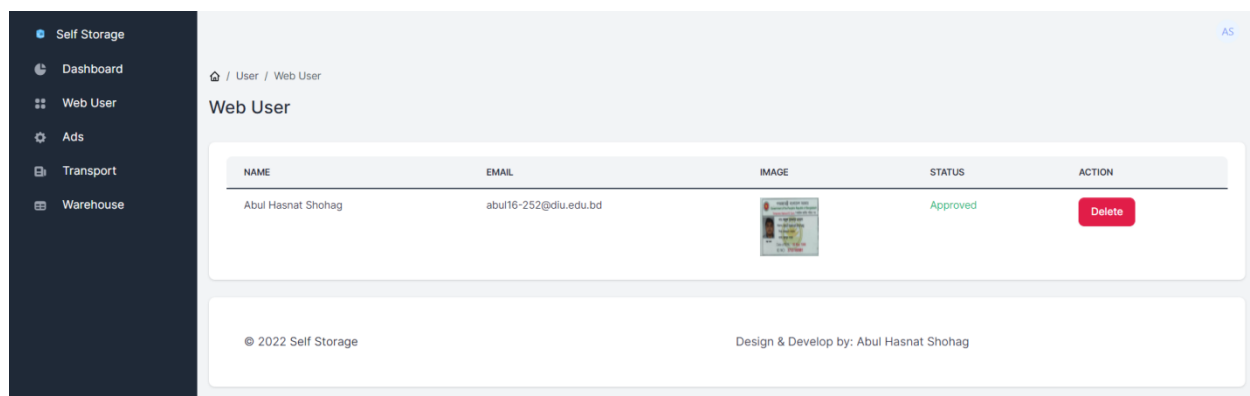


Figure 41:Acceptance Test

11.7 Security Testing

Security Test

Test Case

Name of the test case	Security Testing		
Test Class	auth/login controller		
Description of the test	Security testing of invalid login attempt		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1.go to login page	login error	as expected result
	2, enter wrong credentials and login		

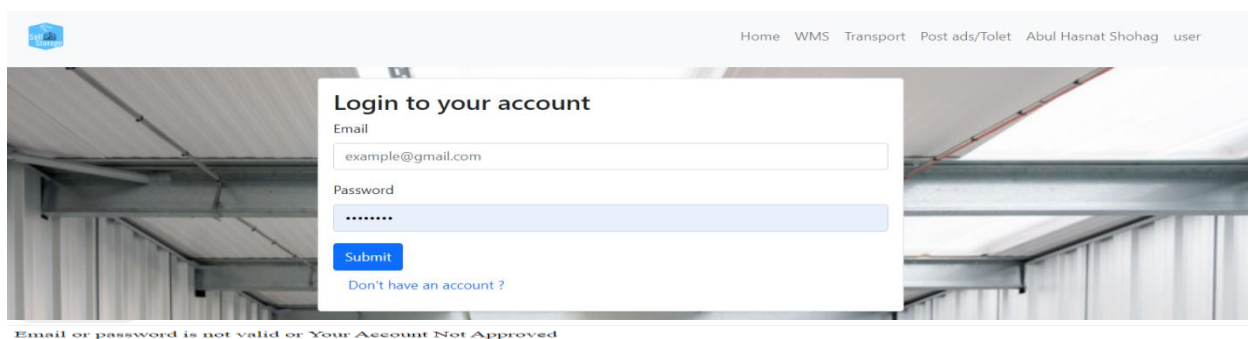


Figure 42: Security Testing

11.8 Accessibility Testing

Accessibility Test

Test Case

Name of the test case	Accessibility Testing		
Test Class	admin controller		
Description of the test	testing user friendliness		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	allowing an admin to utilize the system	the system is simple to use	as expected result

Table 29: Accessibility Testing

Chapter 12 – Implementation

12.1 Training

Users need training on the new system to make them familiar with the system. The development team should ensure that the users can perform the functionalities of the new system efficiently once the project development is finished. A plan for training the users is given here:

SL	User	Training Scope)	Duration	Remarks
1	Users	Registering/ login to the system, posting ads, taking rent, using warehouse management system, taking transport service, searching storage according to location, size, fare etc.	2 hours	users are getting familiar with the idea of the system
2	Admin	managing users, manipulate ads, providing warehouse management service, manage transports	1 hours	admins are able to perform their operation easily

Table 30: User training

12.2 Implementation Scheme

Big Bang

Big Bang works by turning off the old system and replaced by the new one right away. It is time friendly compared to other implementation scheme as it initializes the new system as soon as the test is completed. Transferring the old data from the previous system to the new system might get damaged and unstable as data loss may happen and the new system might crash. Considering the involved human resources it is incorporated in a single site.

12.3 Scaling

Scalability of the application was not considered and no plan was made for the project as it is an academic project.

12.4 Load Balancing

Load balancing is the system's optimization against the users' actions. User hit shows the number of users using the system at the same time and how well the system has sustained. The process is called load balance and load equalizer. It distributes the load among the available servers to keep the system running smoothly. Based on the user hit per day a load balancing strategy should be developed.

Chapter 13 – Critical Appraisal and Evaluation

13.1 Objective could be Met

The objectives which are initially declared are listed below.

- Registration system for users
- Managing the home owners around dhaka to post ads/to-let
- Getting users requirement and purpose of renting storage
- Searching area, space and cost based results for users
- Managing information of stored products for the businesses
- Predicting profitable window for specific goods.

Objective-1

Achievement rate and others

The registration and login system has been developed effectively for the users. All users can perform the registration functionality and login to the system after successful registration to the system. When the users perform login successfully they will be taken to the home page, from here based on the user role they can access the feature that are enabled only for the specific role.

Objective-2

Achievement rate and others:

Managing the home owners around dhaka to post ads/to-let has been fully implemented and successfully working. user can search for the storage/warehouse and rent them from the system.

Objective-3

Achievement rate and others:

Searching area, space and cost based results for users is implemented. Users can search for storage & warehouse based on area they are looking for, how much space they need according to their budget.

Objective-4

Achievement rate and others:

Managing information of stored products for the businesses is implemented. In the system user can see storage & warehouse information as well as see the stored goods.

13.2 Objective that totally don't meet of touched and Why

An objective that could not be touched totally: Predicting profitable windows for specific goods. I could not implement it yet because it needs market research for prices which will require an AI engine. As soon as I develop an AI engine for price forecasting according to market research I will implement it.

What could have been done?

To contact the highlights it needs to reschedule the time box which needs more time and additional opportunities to finish the component. Likewise AI generated machine learning implementation must be learned to truly make the project dynamic.

Chapter 14 – Lessons Learned

14.1 Pre-Project-Review-closing

To develop the Self Storage system I needed to follow a predefined structure. The predefined structure concluded project proposal submitting, project title defense, development and documentation. Self Storage is a one stop storage solution that works with renting, managing the storage.

14.2 What I have learnt

This was my first full application development where I have worked with both web application and mobile application. So, I needed to adapt to smaller functionalities integrating to a full application. I have implemented new technologies that I have never worked with. During the development arranging and managing all these technologies; including backend, frontend, database, system architecture etc. was very exhausting. It helped me improve my arrangement, time management skills and testing abilities as well. And it improved my programming skills and implementation knowledge as this is the first time I have developed a whole project from start to finish. I have learnt app development on flutter. The undertaking kind of forced me to gain a lot of life lessons regarding time management and organized way of working which is really needed to improve life.

14.3 The problems I have faced

I have stumbled many times during the development of the project. Every time I got stuck I have learnt something. As previously stated I have followed DSDM methodology for developing the proposed system. So every time I have made mistakes, face any problems I could solve the problem following the criteria of DSDM methodology. The problems I have faced are missing time box deadline, not knowing implementation of certain technology, not meeting the development milestone etc. As I am not an expert yet I had to make adjustments to the timeboxes few times and working against the time boxes was very hard. While working with firebase I had face problems as it was completely new for me. Rendering the app and running the application without crashing was the most challenging part.

14.4 What Solution Occurred

Whenever I have faced an issue, I had to figure out the solution for it. To solve the problems I had to learn lots of things by myself from various online sources as well as my superiors. The most annoying was the problems that only occurred to me and I had to fix the problem all by

myself. It was frustrating and exciting at the same time to face problems and find solutions for them.

Chapter 15 – Conclusion

15.1 Summary of the Project

Here in Bangladesh most of the students do not have a career goal and even when they do it's too late to emphasize on. So, the CPISS project will help students who are interested to set goals early, it will ensure that they get proper resource and counseling to set goals according to their hobbies and dreams. The project has features that will help students to reach their goal. To develop the project Laravel framework along with the assistance of HTML, CSS, JavaScript, jQuery are used. All the diagrams and charts that are used during the development of the project are submitted within the document.

15.2 Goal of the project

The target of the project was to provide a complete solution regarding storage rental system where all the possible solutions related to renting and storing goods would be satisfied . The main goals are:

- Creating a platform for rental storage units
- Location, size, cost based searching of storage and warehouse
- Ensure providing easy access to storage and related facilities such as transportation.

15.3 What I have done in the documentation

I have documented all the tasks needed to develop the project. This documentation contains all the development process from the very beginning of initial study to deployment of the project. All the processes' walkthrough have been documented in this documentation so the project can be completed such as timeboxing, gantt chart, risk management planning, ERD, class diagram etc.

15.4 Value of the Project

The project is unique to the market but with a smaller target audience. It will be easier to capture the market as projects like this do not exist in the market but a bit harder to expand the target customers. The value of this project in market cannot be estimated accurately as the demand for

storage will very based on the other factors. But it is assured that it can contain an untapped market of storage units which is not a very well established concept in our country.

15.5 My Experience

I have gained lots of knowledge and experience which were completely new to me because this was the first time I have developed a smartphone app. Because I was totally new to this sector. I have faced many issues which allowed me to learn even more as well as gain experience which will be vital for my career moving forward. Especially the experience of managing a whole project development process and developing the project.

Appendices

Other Test Scripts

Name of the test case	Unit Testing		
Test Class	User registration/login controller		
Description of the test	NID validation for user registration		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1. Complete all the information except NID 2. Submit the form	NID is required warning will be shown	NID is required warning is shown

Name of the test case	Integration test		
Test Class	User Dashboard		
Description of the test	User can see own ads/rented storage info		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1. login and go to profile 2. can visit own ads/see own storage 3. ad stored product's info	should be able to see own ads and stored products info and edit the products details	user can see own ads and stored goods and also can add product

Name of the test case	Integration test		
Test Class	Warehouse Management Services		
Description of the test	User can take warehouse management service		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1. login and go to profile 2. can visit own ads/see own storage 3. take warehouse management service 4. can manage product inventory, shipping etc. activities	user should be able to take warehouse management service and utilize the service according to needs	as expected user can take warehouse management service and utilize the service according to needs

Name of the test case	Accessibility Testing		
Test Class	admin controller		
Description of the test	testing user approval		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1. admin login 2. go to new users 3. check the submitted information 4. click approve	user will be registered to the system	user is registered

Name of the test case	Accessibility Testing		
Test Class	admin controller		
Description of the test	testing user manipulation		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1. admin login 2. go to users/ads 3. check if the users and ads can be edited, deleted and updated	user and ads should be able to manipulate	users and ads can be manipulated

Name of the test case	Accessibility Testing		
Test Class	admin controller		
Description of the test	warehouse and transport management		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1. admin login 2. warehouse/transport request 3. approve and initiate required service	warehouse and transport service can be managed	admin can manage warehouse and transport service

Name of the test case	Security testing		
Test Class	stored goods in rented storage		
Description of the test	trying to see what other user stored in storage		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1. try to login using another user's credentials 2. see stored products	user can not access other user's account and can not see other user's storage	wrong info warning shown

Name of the test case	Security testing		
Test Class	warehouse management service		
Description of the test	Security testing of user accessing warehouse management service		
Source of Data	Steps of Testing	Expected Outcome	Actual Outcome
	1. login and go to profile 2. try to access warehouse management service without taking it	warehouse management service should not be available	can not see warehouse management service

User Guide:

1. Create Users account
Brief Description: user will fill out the form and submit with necessary documents
Actors: user
Precondition: submit all the required information
Flow of Events: ❖ go to registration ❖ fill out form ❖ click registration ❖ registration complete
2. User Login
Brief Description: after completing registration login to the system
Actors: user
Precondition: register to the system
Flow of Events: ❖ go to login ❖ fill out email and password ❖ click login ❖ successful login takes to homepage/dashboard
3. admin login
Brief Description: not required registration
Actors: admin
Precondition: create admin entity during development
Flow of Events: ❖ use admin email and password in login page ❖ successful login after validation ❖ go to admin dashboard

4. post ads
Brief Description: user can post ads of their storage/warehouse
Actors: user
Precondition: user must be logged in
Flow of Events: ❖ login ❖ go to ads posting ❖ fill out form with all the required information ❖ post ❖ admin approval ❖ ads shown in the system

5. take rent
Brief Description: user can take rent from the posted ads
Actors: user
Precondition: login to the system
Flow of Events: ❖ login ❖ browse ads ❖ select preferred storage/warehouse ❖ click take rent ❖ fill out the required information ❖ confirm ❖ successful

6. take WMS
Brief Description: user can book WMS for their warehouse/storage
Actors: user
Precondition: user must take rent storage/warehouse
Flow of Events: ❖ login ❖ take storage/warehouse rent ❖ select available wms suitable for own preference ❖ fill out form ❖ wms approved

7. post ads of personal goods stored in rented storage
Brief Description: user can put out personal goods for other users to take rent
Actors: user
Precondition: user must login, take rent storage and add stored goods details in my storage section
Flow of Events: ❖ login ❖ go to my profile ❖ go to my storage ❖ add stored product info ❖ click rent out ❖ ad will be posted without having to input further info

8. contact admin for service
Brief Description: user can contact admin if any service is not in the system
Actors: user
Precondition: user should be logged in
Flow of Events: ❖ login ❖ go to contact ❖ fill out form ❖ submit

Works Cited

Waterfall methodology. [Online]

Available at: <https://www.workfront.com/project-management/methodologies/waterfall>

[Accessed 02 07 2022].

GeeksforGeeks, 2018. *Software engineering*. [Online]

Available at: <https://www.geeksforgeeks.org/software-engineering-integration-testing/>

[Accessed 04 07 2022].

Anon., 2021. *What is Rapid Application Development? Definition of Rapid Application Development, Rapid Application Development meaning*. [Online]

Available at: <https://economictimes.indiatimes.com/definition/rapid-application-development>

[Accessed 02 07 2022].

ApacheBooster, 2020. *Client server technology*. [Online]

Available at: <https://apachebooster.com/blog/client-server-technology/>

[Accessed 02 07 2022].

GeeksforGeeks, 2019. *Acceptance testing*. [Online]

Available at: <https://www.geeksforgeeks.org/acceptance-testing-software-testing/>

[Accessed 04 07 2022].

GeeksforGeeks, 2019. *System testing*. [Online]

Available at: <https://www.geeksforgeeks.org/system-testing/>

[Accessed 04 07 2022].

GeeksforGeeks, 2019. *Usability testing*. [Online]

Available at: <https://www.geeksforgeeks.org/usability-testing/>

[Accessed 06 07 2022].

knowledge, P. m., 2016. *Project management knowledge*. [Online]

Available at: <https://project-management-knowledge.com/definitions/q/questionnaires-and-surveys/>

[Accessed 12 07 2022].

Ltd}, { . A., 2021. *Dynamic Systems Development Methodology*. [Online]

Available at: <https://www.ukessays.com/essays/information-systems/dynamic-systems-development-methodology.php>

[Accessed 13 07 2022].

Martin, M., 2021. *What is RAD Model? Phases, Advantages and Disadvantages*. [Online]

Available at: <https://www.guru99.com/what-is-rad-rapid-software-development-model->

[advantages-disadvantages.html](#)

[Accessed 15 07 2022].

Study.com, 2020. *Agile DSDM: Methodology & Project Framework..* [Online]

Available at: <https://study.com/academy/lesson/agile-dsdm-methodology-project-framework.html>

[Accessed 17 07 2022].

Wikimedia.org, 2019. *Wikimedia.org.* [Online]

Available at: <https://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/Client-server-model.svg/1200px-Client-server-model.svg.png>

Plagiarism Report

181-16-252(final)

ORIGINALITY REPORT

22% SIMILARITY INDEX	8% INTERNET SOURCES	1% PUBLICATIONS	20% STUDENT PAPERS
--------------------------------	-------------------------------	---------------------------	------------------------------

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	15%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	3%
3	Submitted to University of Greenwich Student Paper	2%
4	Submitted to Kingston University Student Paper	<1%
5	Submitted to NCC Education Student Paper	<1%
6	Submitted to Postgraduate Institute of Medicine Student Paper	<1%
7	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
8	Submitted to Sunway College Student Paper	<1%
9	Submitted to Middle East College of Information Technology Student Paper	<1%
10	Submitted to University of Wollongong Student Paper	<1%
11	WWW.coursehero.com Internet Source	<1%
12	www.workfront.com Internet Source	<1%
13	Mohammed Abdul, Dr. Shahanawaj. "Managing Knowledge in Development of Agile Software", International Journal of Advanced Computer Science and Applications, 2011 Publication	<1%
14	Submitted to University of Mauritius Student Paper	<1%
15	docshare.tips Internet Source	<1%
16	ris.utwente.nl Internet Source	<1%
17	www.sec.gov Internet Source	<1%