

**Apartment Finding System**

**Software Project 1**

A Software Project Submitted

By

|  |  |
| --- | --- |
| Name | ID |
| **Manik Saha** | 18-38965-3 |
| **Merina Tanjin** | 18-37783-2 |
| **Effat Jahan** | 18-38718-3 |
| **Fyzun naher Mim** | 18-36446-1 |

Under the supervision of

**Abhijit Bhowmik**

Associate Professor

Department of Computer Science

Faculty of Science and Technology

American International University Bangladesh

Summer Semester 2020-2021

**Disclaimer**

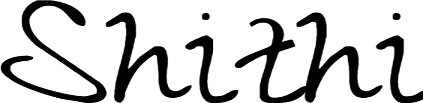
This is to certify that this project is our original work. No part of this has been submitted elsewhere partially or fully for the award of any other degree. Any material reproduced in this project has been properly acknowledged.

****

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Name: MANIK SAHA

ID No: 18-38965-3



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Name: MERINA TANJIN

ID No: 18-37782-2



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Name: EFFAT JAHAN

ID No: 18-38718-3



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Name: MIM,FYZUN NAHER

ID No: 18-36446-1

**Approval**

The Software Project or Project titled “Apartment Finding System” has been submitted to the following respected members of the Board of Examiners of the Faculty of Science and Information Technology in partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering on 31th July 2021 by the following students and has been accepted satisfactory.

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Abhijit Bhowmik  Associate Professor & Supervisor  Department of Computer Science  American International University- Bangladesh  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Professor Dr. Tafazzal Hossain  Dean in Charge  Faculty of Science & Technology  American International University-Bangladesh | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Syeda Amika Tasnim  Lecturer & External  Department of Computer Science  American International University-  Bangladesh  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Dr. Carmen Z. Lamagna  Vice Chancellor  American International University-  Bangladesh |
|  |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Dr Md. Mahbub Chowdhury Mishu**  Head (Undergraduate)  Department of Computer Science  American International University- Bangladesh |  |
|  |  |
|  |  |
|  |  |

**Acknowledgements**

We would like to take the chance to express our gratitude to our honorable teacher & Supervisor Abhijit Bhowmik, for his continuous guidance and support regarding this report. Besides this, we have found so many sincere and productive advices from many people that we would like pay homage to them. We convey our gratitude to our honorable Vice Chancellor, Dr. Carmen Z. Lamagna for encouragement.

Table of Contents

[Chapter 1: Statement of Work 8](#_Toc64269097)

[1.1 Purpose/Objectives 8](#_Toc64269098)

[1.2 Scope 8](#_Toc64269099)

[1.3 Proposed System 8](#_Toc64269100)

[1.4 System Features 9](#_Toc64269101)

[1.5 Environment 9](#_Toc64269102)

[1.5.1 Organizations Involved 9](#_Toc64269103)

[1.5.2 Processing 10](#_Toc64269104)

[1.5.3 Security 10](#_Toc64269105)

[1.6 Assumptions 10](#_Toc64269106)

[1.7 Constraints 10](#_Toc64269107)

[1.8 Proposed System 11](#_Toc64269108)

[1.8.1 Description/Improvements of Proposed System 11](#_Toc64269109)

[1.8.2 Resources 11](#_Toc64269110)

[1.8.3 Hardware 12](#_Toc64269111)

[1.8.4 Software 12](#_Toc64269112)

[1.8.5 Operating Environment 12](#_Toc64269113)

[1.9 Project Time & Cost 13](#_Toc64269114)

[1.9.1 Project Schedule and timeline of Milestones 13](#_Toc64269115)

[1.9.2 Effort Allocation 14](#_Toc64269116)

[1.9.3 Working Days 15](#_Toc64269117)

1.9.4 Scheduling and Budgeting…………………………………………………..............16

1.9.5 Project Scheduling …………………………………………………………………..17

1.9.6 Domain & Hosting Package……………………………………………………..…..19

[1.10 Risk assessment 20](#_Toc64269118)

[Chapter 2: Software Requirement Specification 21](#_Toc64269120)

[2.1 Objectives and Scope 21](#_Toc64269121)

[2.2 Overview of the Present System 22](#_Toc64269122)

[2.3 Data Flow Diagram of the Present System 22](#_Toc64269123)

[2.4 Weakness of the Present System 22](#_Toc64269124)

[2.5 Overview of the Proposed System 22](#_Toc64269125)

[2.6 Benefits of Proposed System 23](#_Toc64269126)

[2.7 System Features 23](#_Toc64269127)

[2.8 Hardware and Software Requirements 25](#_Toc64269128)

[2.8.2 Hardware 25](#_Toc64269129)

[2.8.3 Software 25](#_Toc64269130)

[2.9 Human Resource Requirements 26](#_Toc64269131)

[2.10 Constraints and Limitations 26](#_Toc64269132)

[2.11 Budget 26](#_Toc64269133)

[2.12 Conclusion 27](#_Toc64269134)

[Chapter-3: Diagram 28](#_Toc64269135)

[3.1 Use Case Diagram 28](#_Toc64269136)

[3.1.1. Admin Functionality: 28](#_Toc64269137)

[3.1.2. Apartment Owner Functionality: 29](#_Toc64269138)

[3.1.3 Customer Functionality: 30](#_Toc64269139)

[3.1.4 Manager Functionality : 31](#_Toc64269140)

[3.2 Activity Diagram 32](#_Toc64269141)

[3.3 Prototype 35](#_Toc64269142)

[Chapter-4: Software Project Management Plan 38](#_Toc64269143)

[4.1.1 Document History and Distribution 38](#_Toc64269144)

[4.1.2 Distribution 38](#_Toc64269146)

[4.2 Project Organization 38](#_Toc64269156)

[4.2.1 External Interfaces 38](#_Toc64269157)

[4.2.2 Internal Structure 39](#_Toc64269158)

[4.2.3 Roles and Responsibilities 39](#_Toc64269159)

[4.3 Managerial Process Plans 39](#_Toc64269160)

[4.3.1 Project Start-up Plan 39](#_Toc64269161)

[4.3.2 Estimation Plan 39](#_Toc64269162)

[4.3.3 Staffing Plan 39](#_Toc64269163)

[4.3.4 Resource Acquisition Plan 40](#_Toc64269164)

[4.3.5 Project Staff Training Plan 40](#_Toc64269165)

[4.4 Work Plan 40](#_Toc64269166)

[4.5 Control Plan 40](#_Toc64269169)

[4.5.1 Requirements Control Plan 40](#_Toc64269170)

[4.5.2 Schedule Control Plan 40](#_Toc64269171)

[4.5.3 Budget Control Plan 41](#_Toc64269172)

[4.5.4 Quality Control Plan 41](#_Toc64269173)

[4.5.5 Reporting Plan 41](#_Toc64269174)

[4.5.6 Metrics Collection Plan 41](#_Toc64269175)

[4.6 Risk Management Plan 42](#_Toc64269176)

[4.7 Closeout Plan 42](#_Toc64269177)

[4.8 Technical process plans 42](#_Toc64269178)

[4.9 Process Model 42](#_Toc64269179)

[4.10 Methods, Tools and Techniques 43](#_Toc64269180)

[4.11 Infrastructure Plan 43](#_Toc64269181)

[4.12 Product Acceptance Plan 43](#_Toc64269182)

[4.13 Supporting Process Plans 43](#_Toc64269183)

[4.14 Configuration Management Plan 43](#_Toc64269184)

[4.15 Verification And Validation Plan 43](#_Toc64269185)

[4.16 Documentation Plan 44](#_Toc64269186)

[4.17 Quality Assurance Plan 44](#_Toc64269187)

[4.18 Reviews and Audits Plan 44](#_Toc64269188)

[4.19 Problem Resolution Plan 44](#_Toc64269189)

[4.20 Subcontractor Management Plans 44](#_Toc64269190)

List of Figure and Table

Fig1: Work Structure……………………………………………………………………..13

Fig 2: Use case of Admin (user)……………………...………………………………......28

Fig 3: Use case of Apartment Owner (user)……………………………………………...29

Fig 4: Use case of Customer (user)………………………………………………………30

Fig 5: Use case of manager (user)………………………………………………………...31

Fig 6: Activity diagram of login………………………………………………………….32

Fig 7: Activity diagram of Admin role…………………………………………………..33

Fig 8: Activity diagram of manager role…………………………………………………33

Fig 9: Activity diagram of other users…………………………………………………..34

Fig 10: User interface of login page…………………………………………………… 35

Fig 11: User interface of Home page…………………………………………………... 35

Fig 12: User interface of search result…………………………………………………. 36

Fig 13: User interface of Apartment details……………………………………………..36

Fig 14: User interface of Owner dashboard……………………………………………..37

Fig 15: User interface of posting Apartment details…………………………………….37

Table 1: Working days…………………………………………………………………..15

Table 2: Cost of project application……………………………………………………. 16

Table 3: Project Scheduling (month 1 to month 4)………………………………………17

Table 4: Approximate date of project completion (month 1 to month 4)………..………17

Table 5: Project Scheduling (month 5 to month 8)………………………………………18

Table 6: Approximate date of project completion (month 5 to month 8)………………..18

Table 7: Cost Assumption………………………………………………………………..27

Table 8: Distribution list……………………………………………………………… 38

Table 9: Risk management plan……………………………………………………........42

[REFERENCES 45](#_Toc64269192)

# 

# Chapter 1: Statement of Work

* 1. **Purpose:**

The goal of the study is to develop a system that reduces people's hassle of changing homes. In this modern world, people prefer everything digitized. This study will help people to find their choice able apartment easily from a variety of collections by sitting at home for renting or buying. It will also save lots of time with less physical work.

This study proposes a web-based application that would give all of the necessary facilities and features for users to reduce hassle and make home changing easy. The end result might be a self-contained program that can be accessed by people all around the country.

**1.2 Scope:**

Admin has access to the entire system. Admin tasks include updating or modifying existing systems, ensuring data security and user authentication, and assigning appropriate roles and rights to system users and clients. Admin has the authority to revoke any privileges that have been granted to users or clients. The manager will be appointed based on the area. They will handle the area-based apartment information. The Owner will upload the apartment details and the customer can choose them for renting or buying.

**1.3 Proposed System:**

This software or Web Application is intended for implementing a Smart apartment finding system so that a customer can easily access our service from anywhere, anytime. This system can make a customer’s house shifting much easier.

Benefits /Improvements of Proposed System

●      No hassle for changing home

●      Customers can find choice able apartments by sitting at home

●      save time with less physical work

●      House cleaning options

●      facilities to buy an apartment

●      Recommends for choosing apartment according to requirements

●      Well planned budget for all type of customer (house shifting, cleaning, designing)

●      Give preference to every client according to their limitation of expense

●      Fast & safe booking

●      Instant transaction history

●      Client satisfaction

●      Compliance with time

**1.4 System Features:**

**Customer**:

* They can view the apartment list for free. If a customer wants to view details, he has to create an account.
* If a customer chooses any apartment, he can directly contact the owner without any third person.
* If a customer wants, he can also chat with the owner through our website.
* A customer can choose other services from our website.
* Customer can update his profile.
* Customer can review the services

Such as:

* House shifting
* Apartment Cleaning
* Decorate apartment

**Owner:**

* An owner can create his own account.
* He can upload his apartment details like photos, contact no etc.
* Can update his profile and apartment details.

**Admin**

* Admin will appoint the managers.
* Update the salary list.
* Check monthly revenue

**Manager**

* Manager will be appointed based on the area.
* They will handle the area-based apartment information

**Other service givers**

* They will update their profile mentioning their skill and hourly/service-based charge
* Service givers will have a profile on our website. They will need to give a monthly subscription fee for their profile to be activated.
* They can chat with the customer on their own. And provide the service.
* They can review the customer

**1.5.1 Environment**

**1.5.1 Organizations Involved**

Project Client: ABHIJIT BHOWMIK.

Developer: Intelligent Apartment Finding System team

User: Online Users or Customers

## **1.5.2 Processing**

* This Web Application will have a graphical user interface which can be accessed through any web browser
* The website is browser dependent
* There are four working modules. Customer, Apartment Owner, Manager, Admin
* This website will store the information of all registered user which can be viewed by user themselves and the administrator of this software
* Authenticated & secure login system and secure data transmission for all user
* Detailed log of all previous transaction

## **1.5.3 Security**

System’s security requirement:

## User authentication is required to access the application.

* A client o user must be a registered user to login to use the features
* Without proper authentication no personal information will be given
* Without proper authentication no service transaction will be allowed

**1.6 Assumptions**

Some third-party UI components may be used to build up this project. These are free component libraries like Angular material or React Bootstrap. We will be using google chrome, Mozilla Firefox as a web browser to access the user interface. So our project will not be using anything illegal.

Some open-source libraries and software’s are used to build up this project:

* Figma (Free version for UI design)
* Angular Material component library (for Dynamic Interface)
* Typescript and CSS is used
* Verification done by API call with MD5 hash generator password

**1.7 Constraints**

* **Usage outside regulation:** Data passes from client to server through TCP/IP & we are not using any public key encryption service like SSL certificate. So we have constraints in case of passing user data. It may cause – Confidentiality, Integrity problems. Only registered users are valid & valid users can use the software through Client Application with help of Internet Browser on server side. For any missing password found by an anonymous user, responsibility goes to the valid user.

* **Bandwidth limitations:** It may lose server connection for technical error (Depends on Hardware/Internet connection). We need to run the query again.
* **Databases:**  We are using MySQL Database. User queries more than server’s limitations we need to check databases and refresh table data. In case of lack of DB caching.
* **Parallel operations:**  Parallel use of other Internet applications with this software may hamper in bandwidth, may occur taking time for a query for slow connections.
* **Language requirements:** Language used in this software is TypeScript. Suppose any user wants Oracle Database we need to use a bind variable technique.
* **Communications protocols:** Communication protocols we are using- TCP/IP to interact with the server. Other protocols are not considerable if the user wants.
* **Security considerations:** If a user does not want to buy SSL security, then client applications will not use any public key encryption service like SSL certificate (i.e., 128-bit RSA encryption). So, we have constraints in case of passing user data.

It may cause (In case of internet security) –

* **Authentication problem:** Server may not recognize/confirm actual valid user.
* **Confidentiality problem:** User, intended server fails “understanding” message contents.
* **Integrity problem:** sender, server may fail to ensure message not altered without detection
* **Eavesdrop**: There may be Intercept messaging, actively insert messages into connection
* **Impersonation:** can fake (spoof) source address in packet (or any field in packet)
* **Hijacking:** “take over” ongoing connection by removing sender or server, inserting himself in place

**Denial of service:** prevent service from being used by others (e.g., by overloading resources)

**1.8 Proposed System**

**1.8.1 Description/Improvement Of proposed system**

* Apartment seller and buyer satisfaction.
* Renter will find rent easily.
* Every user from our system can able to update their profile.
* Other facilities like house shifting, apartment cleaning and decorate apartment will be available at the website.
* House owner can upload his apartment details like photos, contact no. etc.
* Efficient, smooth and secure online and offline transaction.
* Finding and changing apartment will be easier.
* Can chat through the website.
* This system will actually save time in today’s busy life.
* Monitor system performance efficiently.
* Renters can review about the apartment.

**1.8.2 Resources**

There are two kinds of resources. One is hardware and the other one is software.All resources needed is provided below.

**1.8.3 Hardware**

* Minimum requirements for server:
* Processor: Xeon based microprocessor (2.0 GHz or faster recommended)
* Ram: minimum 16 GB
* System type: Linux (64 bits)
* Storage: minimum 1TB SSD or higher.
* For Storage Service: Network File System (NFS)
* Minimum requirements for client:
* Processor: Dual-core.
* RAM: 2 GB.
* System: Windows, MAC OS X, Linux.
* Web Browser: Firefox, Google Chrome, Opera

**1.8.4 Software**

* Text editor like Notepad++, Sublime Text, VS code.
* C#, Microsoft SQL server.
* Web Browser: Firefox, Google Chrome, Opera.

**1.8.5 Operating Environment**

The whole system will be operated from the external (your preferred data center) Linux Server. Hosting server has 99% Uptime. This website is platform independent that means user can use it from PC, phone, tab or any kind of internet operate devices. User application is mainly accessible through various kinds of browsers like Opera, Mozilla Firefox, and Google Chrome etc. This website is a web application where client application has user interfaces through browser and main part is hosted on Server. IBM or MAC or Windows any user from any platform can use. Operating System can be used Windows of any version from Windows 98, Windows XP/Vista to Windows 10, MAC OS X 10.5 or above.

**1.9 Project Time & Cost:**

**1.9.1 Project Schedule and timeline of Milestones**

It is the process of predicting the most realistic amount of effort required to develop or maintain software based on incomplete, uncertain and noisy input. Effort estimates may be used as input to project plans, iteration plans, budgets, investment analysis, pricing processes and bidding rounds We need to follow WBS (Work Breakdown Structure) Chart.

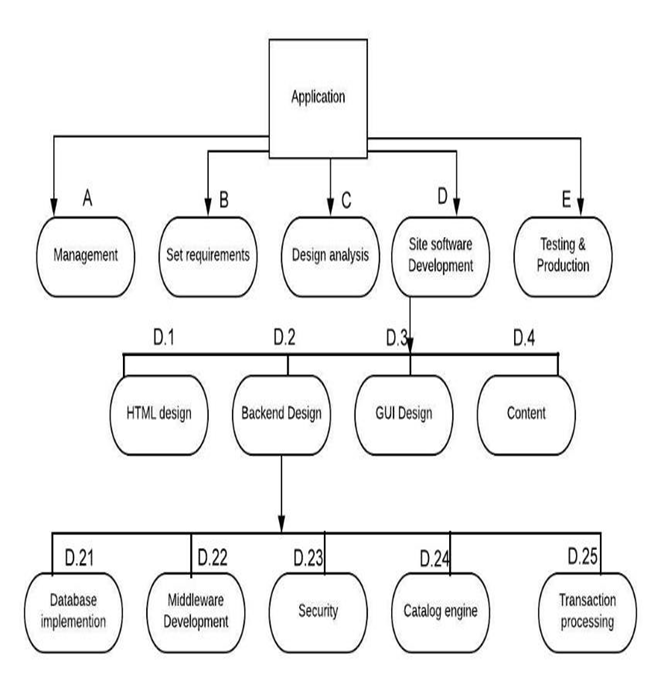


Fig1: Work Structure

## 

## **1.9.2 Effort Allocation:**

## Front-end activities (40-50%)

•  Customer communication

•  Analysis

•  Design

•  Review and modification

**Construction activities (15-20%)**

• Coding or code generation

**Testing and installation (30-40%)**

•  Unit testing, integration

•  White-box, Black box

•  Regression

Once the WBS is ready and the size and effort estimates are known, you are ready for scheduling the tasks.

While scheduling the tasks, certain things should be taken into account −

•  **Precedence**: A task that must occur before another is said to have precedence of the other.

•  **Concurrence**: Concurrent tasks are those that can occur at the same time (in parallel).

•  **Critical Path**: Specific set of sequential tasks upon which the project completion date depends.

o All projects have a critical path.

o Accelerating non-critical tasks do not directly shorten the schedule.

**1.9.3 Working Days**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No. | Items | Total (Working days) | Dependencies |
| 1 | Registration Wizard | 3 |  |
| 2 | User Access  Management - with role management | 9 | 1 |
| 3 | Forum & Discussion module | 6 |  |
| 4 | Task manufacturing | 33 |  |
| 5 | Database management system | 14 |  |
| 6 | Messaging system | 17 | 2,4,5 |
| 7 | Exam or Testing module | 11 | 4 |
| 8 | Advertisement | 29 | 3,4,7 |
| 9 | Reporting | 16 | 2,3,4,7,8 |

Table 1: Working days

Total days for development: **138** working days

Total time for development: **8** months

**1.9.4 Scheduling and Budgeting:**

The following assumptions can be made about the costs for the project application:

|  |  |  |  |
| --- | --- | --- | --- |
| Items | Daily Cost (BDT) | No. of Days | Total Cost (BDT) |
| Front-end developer | 650 | 96 | 62,400 |
| Back-end developer | 750 | 88 | 66,000 |
| Business engineering | 400 | 25 | 10,000 |
| Total development cost |  |  | BDT 138,400 |
| Overhead Cost (20%) |  |  | 27,680 |
| Safety net for spillover (8%) |  |  | 11,072 |
| Cloud Server |  |  | 260,000 |
| Net cost |  |  | BDT 437,152 |

Table 2: Cost of project application

**1.9.5 Project Scheduling:**

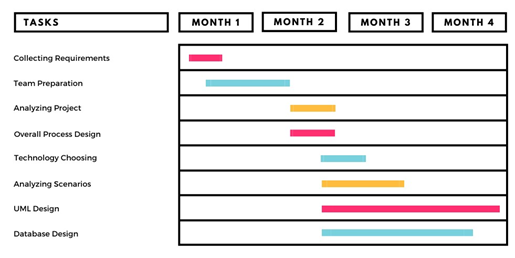


Table 3: Project scheduling (month 1 to month 4)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tasks | MONTH 1 | MONTH 2 | MONTH 3 | MONTH 4 |
| Collecting Requirements | 1/8/2021 to 15/8/2021 |  |  |  |
| Team Preparation | 8/8/2021 to | 12/9/2021 |  |  |
| Analyzing project |  | 12/9/2021 to 30/9/2021 |  |  |
| Overall Process Design |  | 12/9/2021 to 30/9/2021 |  |  |
| Technology Choosing |  | 25/9/2021 to | 10/10/2021 |  |
| Analyzing Scenarios |  | 25/9/2021 to | 25/10/2021 |  |
| UML Designs |  | 25/9/2021 to |  | 28/11/2021 |
| Database Design |  | 25/9/2021 to |  | 15/11/2021 |

Table 4: Approximate date of project completion (month 1 to month 4)

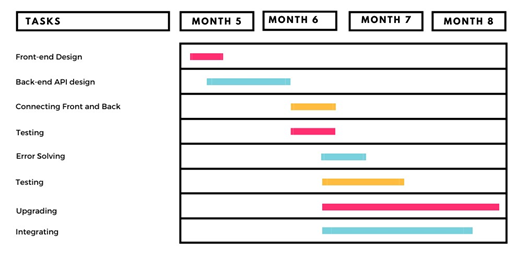


Table 5: Project Scheduling (month 5 to month 8)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tasks | MONTH  5 | MONTH  6 | MONTH 7 | MONTH 8 |
| Front-end design | 1/12/2021 to 20/12/2021 |  |  |  |
| Back-end and API design | 15/12/2021 to | 15/1/2022 |  |  |
| Connecting front and back |  | 10/1/2022 to 30/1/2022 |  |  |
| Testing |  | 10/1/2022 to 30/1/2022 |  |  |
| Error solving |  | 15/1/2022 to | 10/2/2022 |  |
| Testing |  | 15/1/2022 to | 20/2/2022 |  |
| Upgrading |  | 15/1/2022 to |  | 30/3/2022 |
| Integrating |  | 15/1/2022 to |  | 22/3/2022 |

Table 6: Approximate date of project completion (month 5 to month 8)

## **1.9.6 Domain & Hosting Package**

**Domain**

* **.com** 950 tk/yr
* **.net** 950 tk/yr
* **.org** 950 tk/yr
* **.biz** 850 tk/yr
* **.info** 850 tk/yr
* **.xyz** 200 tk/yr

**Hosting Package A:** Great for small websites

* Web Space: 1GB SSD Storage
* Bandwidth: 30GB/monthly
* RAID 10 SSD Server
* Unlimited Sub Domains
* Unlimited Email Accounts
* Unlimited Databases
* Tk. 1500/Year

**Hosting Package B:** Perfect for medium sized websites

* 3GB SSD Storage
* 90 GB Bandwidth Monthly
* RAID 10 SSD Server
* Lite Speed Web Server
* Three Addon Domains
* Unlimited Sub Domains
* Unlimited Email Accounts
* Unlimited Databases
* Tk. 2500/year

**Hosting Package C:** For the demanding sites

* 5 GB SSD Storage
* 150 GB Bandwidth Monthly
* RAID 10 SSD Server
* Lite Speed Web Server
* Five Addon Domains
* Unlimited Sub Domains
* Unlimited Email Accounts
* Unlimited Databases
* Tk. 3500/year

**Hosting Package D:** For the highly demanding sites

* 20 GB SSD Storage
* 500 GB Bandwidth Monthly
* RAID 10 SSD Server
* Lite Speed Web Server
* Nine Addon Domains
* Unlimited Sub Domains
* Unlimited Email Accounts
* Unlimited Databases
* Tk. 7000/year

**1.10 Risk management Strategy**

To create a successful application, the product team must minimize risks inside and outside the scope of the project. Some common risks are:

•       Inaccuracies in the time and resource estimations.

•       Scalability of the core framework architecture.

•       Sudden change of requirements.

•       Government rule changes.

•       More focus on technical complexity, rather than user experience.

If the team is able to address these issues, we should not face any problems in delivering the final application. But, as there are chances of risk, so we should follow a risk management plan. The focus on the risk-management planning is to develop a plan to handle each of the high -priority risks.

• **Risk Mitigation:**

The activity of mitigating and avoiding software risks is based on information gained from the previous activities of identifying, planning, and assessing risks. Risk mitigation handling options include:

•       Setup some built in tutorials which will help our users to learn how to use the application.

•       Develop the software in the proper way & test all the possibilities of bugs.

•       Rather than changing requirements constantly the team should highly focus on what the team is developing.

•       If government rules get a change, it should be accepted by the team & should try to cope with the problem. 

# Chapter 2: Software Requirement Specification

**2.1 Objectives and Scope**

 This software or Web Application is intended for implementing a Smart apartment finding system so that a customer can easily access our service from anywhere, anytime. This system can make a customer’s house shifting much easier.

Admin has access to the entire system. Admin tasks include updating or modifying existing systems, ensuring data security and user authentication, and assigning appropriate roles and rights to system users and clients. Admin has the authority to revoke any privileges that have been granted to users or clients. The manager will be appointed based on the area. They will handle the area-based apartment information. The Owner will upload the apartment details and the customer can choose them for renting or buying.

Smart Apartment System is supposed to have the following features:

For customer

* They can view the apartment list for free. If a customer wants to view details, he has to create an account.
* If a customer chooses any apartment, he can directly contact the owner without any third person.
* If a customer wants, he can also chat with the owner through our website.
* A customer can choose other services from our website.
* Customer can update his profile.
* Customer can review the services

Such as:

* House shifting
* Apartment Cleaning
* Decorate apartment

**For Owner:**

* An owner can create his own account.
* He can upload his apartment details like photos, contact no etc.
* Can update his profile and apartment details.

**For Admin**

* Admin will appoint the managers.
* Update the salary list.
* Check monthly revenue

**For Manager**

* Manager will be appointed based on the area.
* They will handle the area-based apartment information

**Other service givers**

* They will update their profile mentioning their skill and hourly/service-based charge
* Service givers will have a profile on our website. They will need to give a monthly subscription fee for their profile to be activated.
* They can chat with the customer on their own. And provide the service.
* They can review the customer

The features that are described in this document are used in the future phases of the software development cycle. The features described here meet the needs of all the users.

## **2.2 Overview of the Present System**

Currently there are very few systems like B-property implemented in Bangladesh that are based only on finding apartments. In present system Customer cannot contact the apartment owners directly. Customers cannot get the apartment location without contacting the support. When searching for an area-based apartment, Apartment listings from 1-3 years ago come up as search results at the very beginning. There is no option for apartment buying and selling. There are no extra services available for the customer. Also, the apartment owner needs to put the apartment listing post every time it gets vacant. There is no listing booster system

## **2.3 Data Flow Diagram of the Present System**

Not required.

**2.4 Weakness of the Present System**

* Customer cannot contact the apartment owners directly
* Customers cannot get the apartment location without contacting the support.
* When searching for an area-based apartment, Apartment listings from 1-3 years ago come up as search results at the very beginning.
* There is no option for apartment buying and selling.
* There are no extra services available for the customer
* The apartment owner needs to put the apartment listing post every time it gets vacant.
* There is no listing booster system

**2.5 Overview of the Proposed System**

Smart apartment System will have the systems written below-

* They can view the apartment list for free. If a customer wants to view details, he has to create an account.
* If a customer chooses any apartment, he can directly contact the owner without any third person.
* If a customer wants, he can also chat with the owner through our website.
* A customer can choose other services from our website.
* Customer can update his profile.
* Customer can review the services

Such as:

* House shifting: Experience movers will help with this task
* Apartment Cleaning: experienced cleaners will do this task
* Decorate apartment: professional architecture will be available for collaborating.
* An owner can create his own account.
* He can upload his apartment details like photos, contact no etc.
* Can update his profile and apartment details.

**2.6 Benefits of Proposed System**

**Benefits/Improvements of Proposed System**

* Customers can get the contact of the apartment owners and the apartment location directly simply by registering and logging in to our website.
* When searching for an area-based apartment only the recent listing which are marked available will show up
* There will be options for apartment buying and selling.
* There will be extra services available for the customer,

Such as:

* House shifting
* Apartment Cleaning
* Decorate apartment
* The apartment owner does not need to put the apartment listing post every time it gets vacant. He can simply turn on the available button and the apartment listing will get hidden for the time being.
* There will be a listing booster system. Where for a certain payment the owner will be able to boost his/her post to reach more customers.
* For further improvement of this web application, we want to expand our service area

The other service givers will get the features mentioned down below-

* They will update their profile mentioning their skill and hourly/service-based charge
* Service givers will have a profile on our website. They will need to give a monthly subscription fee for their profile to be activated.
* They can chat with the customer on their own. And provide the service.
* They can review the customer
* The future improvements also hold the plan to make the user interface bilingual. English and Bangla.

**2.7 System Features**

* **Admin**
* Admin will appoint the managers.
* Update the salary list.
* Check monthly revenue
* Can view all apartment and owner details
* Can delete any user from system.
* Can update his self-profile.
* Fan contact with the manager and apartment owners.
* **Customer**
* They can view the apartment list for free. If a customer wants to view details, he has to create an account.
* If a customer chooses any apartment, he can directly contact the owner without any third person.
* If a customer wants, he can also chat with the owner through our website.
* A customer can choose other services from our website.
* Customer can update his profile.
* Can review apartment.
* View house shifting details.
* Get service for decorating the apartment from system.
* Can get recommendation for apartment according to requirements.
* Can register in the system for free.
* Customer can review the services

Such as:

* House shifting
* Apartment Cleaning
* Decorate apartment
* **Apartment Owner**
* An owner can create his own account.
* He can upload his apartment details like photos, contact no etc.
* Can update his profile and apartment details.
* Can contact with admin.
* Can contact with a renter.
* Can get apartment cleaning service.
* Can get service for decorating apartment.
* Pay monthly subscription fee.
* Can advertise through system by paying at a certain amount.
* **Manager**
* Can create his own account.
* Update his account details.
* View apartment details like photos, address of owners etc.
* Can access the owner and customer details.
* Will work based on area.
* **Day-to-day plan**
* Keeping the log of all the activities in the system.
* Every individual apartment owner payment.
* Security and authentication, authorization, verification for the system.
* Keeping the log of new apartment owner, user, customer in the system.
* Also keep the track of the fault in the system.
* Reply of all individuals who have problems from system.
* **Individual Apartment module**
* Can upload all apartment details like photos, address, phone number etc.
* Can update or modify all those details of apartment.
* Can be reviewed by the customers.
* Can be complained by the customer.
* Can be suggested to others by their customers.
* **Payment**
* Keeping log of all the transaction history.
* Individual client wise payment.
* Security about payment.
* Authentication, authorization for payment.
* **Security**
* Payment security.
* User data security.
* Unwanted user access blocked.
* **Transaction history**
* Login/Registration/Authentication and Validation process

**2.8 Hardware and Software requirements**

**2.8.2 Hardware**

* Minimum requirements for server:
* Processor: Xeon based microprocessor (2.0 GHz or faster recommended)
* Ram: minimum 16 GB
* System type: Linux (64 bits)
* Storage: minimum 1TB SSD or higher.
* For Storage Service: Network File System (NFS)
* Minimum requirements for client:
* Processor: Dual-core.
* RAM: 2 GB.
* System: Windows, MAC OS X, Linux.
* Web Browser: Firefox, Google Chrome, Opera

**2.8.3 Software**

* Text editor like Notepad++, Sublime Text, VS code.
* C#, Microsoft SQL server.
* Web Browser: Firefox, Google Chrome, Opera.

## **2.9 Human Resource Requirements**

The total human resource needed for implementing and operating the system is mentioned below.

* **Hardware Specialist**: A part time hardware specialist is needed to manage all the computers of the center. So, in case of any hardware failure, he/she may come and solve the problem.
* **Computer operator/Data entry operator**: A computer operator is needed to upload the online question to the system in case a specific user lacks the system's knowledge.

## **2.10 Constraints and Limitations**

**Assumptions and Dependencies**

* The users have sufficient knowledge of computers.
* The user’s computer should have Internet connection and Internet server capabilities.
* The users know the English language, as the user interface will be provided in English.

**Constraints**

* **Bandwidth limitations**: It may lose server connection for technical error (Depends on Hardware/Internet connection). We need to run query again
* **Databases**: We are using MsSQL Database. User queries more than server’s limitations we need to check databases and refresh table data.
* **Parallel operations:** Parallel use of other Internet applications with this software may hamper in bandwidth, may occur taking time for a query for slow connections.
* **Language requirements:** If any user wants to use any language other than what we used for Oracle Database, we need to use bind variable techniques.
* **Communications protocols:** Communication protocols we are using- TCP/IP to interact with the server. Other protocols are not considered, if the user wants.

## **2.11 Budget**

|  |  |
| --- | --- |
| **Description** | **Cost Assumption** |
| Site launch (hosting) | 50,152 BDT |
| Maintenance (1 year) | 870,000 BDT |
| Developers | 3,00,000 BDT |
| **Grand total** | **437,152 BDT** |

Table 7: Cost Assumption

## **2.12 Conclusion**

This Requirement Specification Document has been developed based upon the studying common scenario and web applications of similar kind. In this document we tried to provide solutions for the problems we came across in similar existing websites. There is more room for improvement. If the budget and timeframe can accommodate more features then there will be changes in this document and development process. Thus, it can be said that the requirement specification document is an ongoing modification process. If any unusual circumstances that arise in the process of development may derail the values and time period mentioned in this document.

**Chapter-3: Diagram**

**3.1 Use Case Diagram**

**3.1.1 Admin Functionality:**

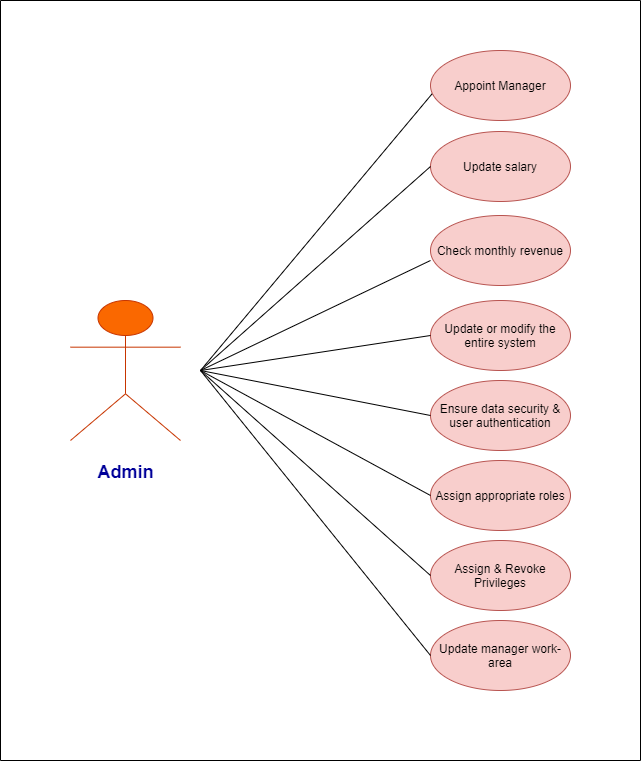


Fig 2: Use case of Admin (user)

**Admin**

* Admin will appoint the managers.
* Update the salary list.
* Check monthly revenue

**3.1.2 Apartment Owner Functionality:**

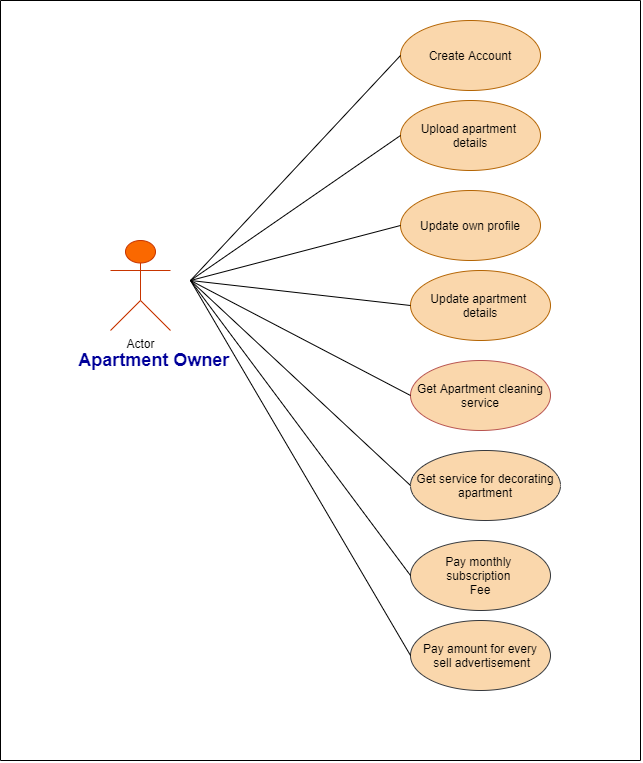


Fig 3: Use case of Apartment Owner (user)

**Owner:**

* An owner can create his own account.
* He can upload his apartment details like photos, contact no etc.
* Can update his profile and apartment details.

**3.1.3 Customer Functionality:**

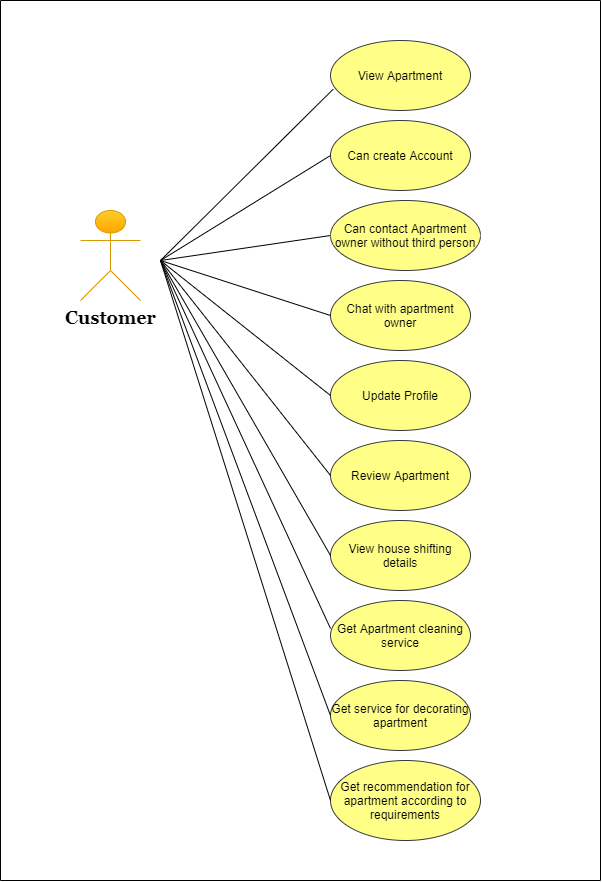


Fig 4: Use case of Customer (user)

**Customer**:

* They can view the apartment list for free. If a customer wants to view details, he has to create an account.
* If a customer chooses any apartment, he can directly contact the owner without any third person.
* If a customer wants, he can also chat with the owner through our website.
* A customer can choose other services from our website.
* Customer can update his profile.
* Customer can review the services

Such as:

* House shifting
* Apartment Cleaning
* Decorate apartment

**3.1.1 Manager Functionality:**

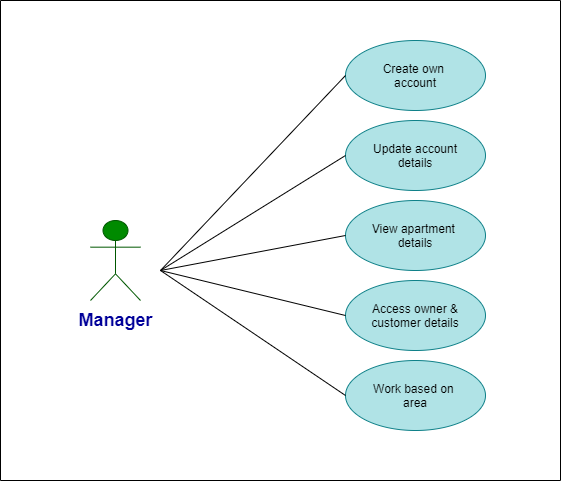


Fig 5: Use case of manager(user)

**Manager**

* Manager will be appointed based on the area.
* They will handle the area-based apartment information

**3.2 Activity Diagram**

**Login activity diagram**

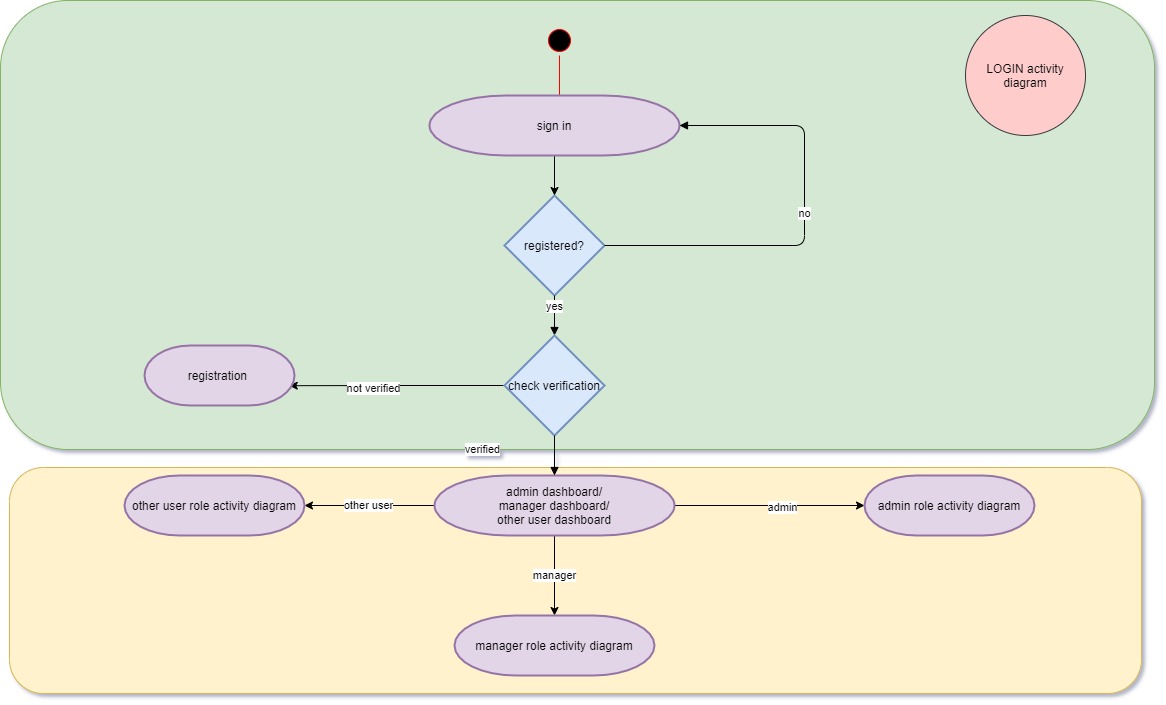


Fig 6: Activity diagram of login

**Admin role activity diagram**



Fig 7: Activity diagram of Admin role

**Manager activity diagram**

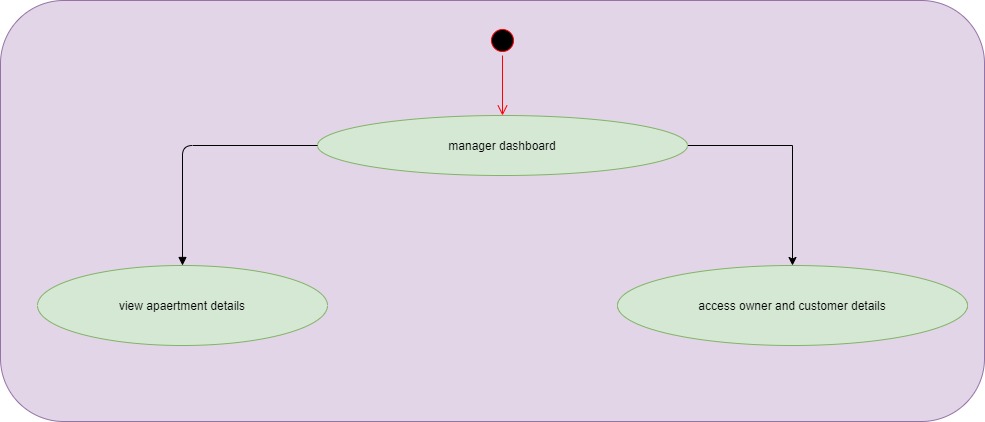


Fig 8: Activity diagram of manager role

**Other users**

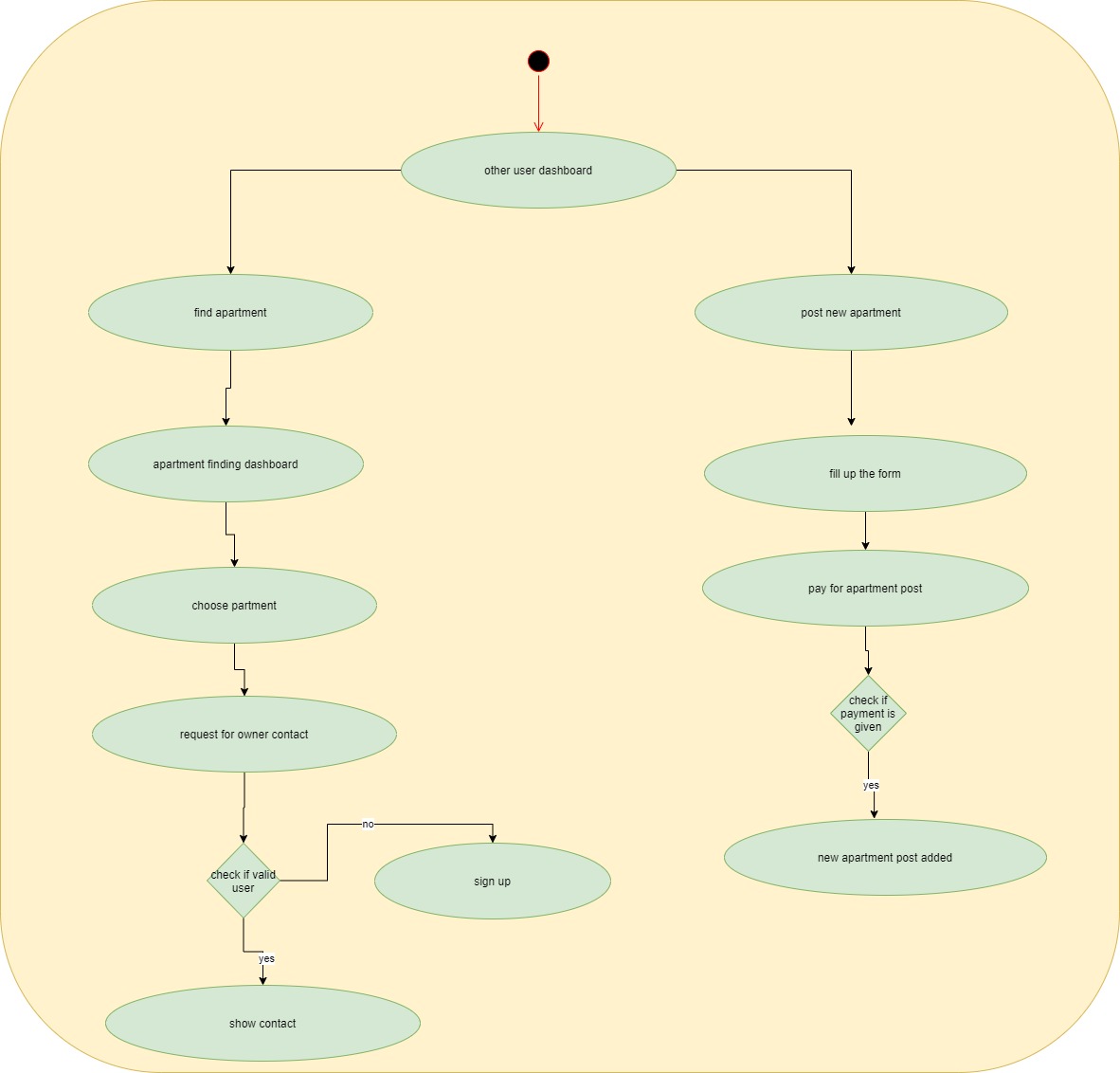


Fig 9: Activity diagram of other users

**3.3 Prototype:**

**Login**:

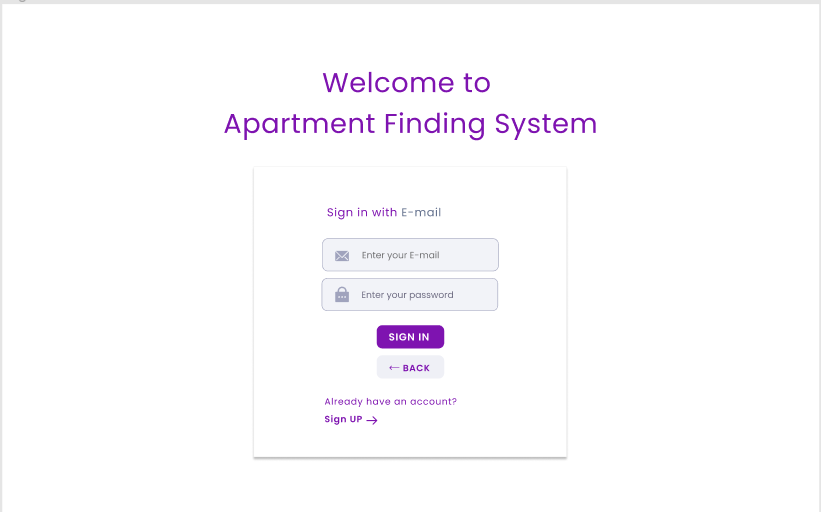


Fig 10: User interface of login page

**Home Page:**

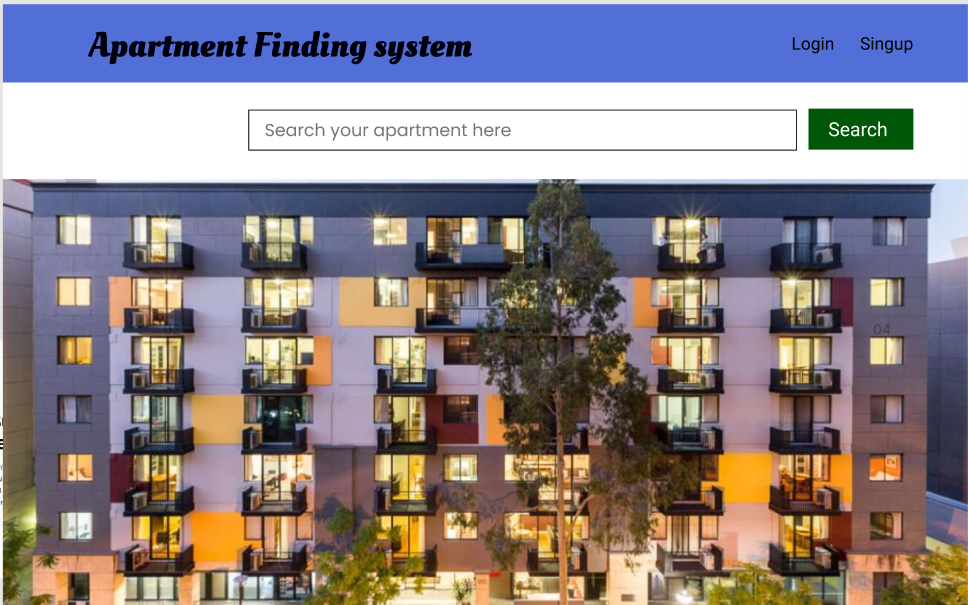


Fig 11: User interface of Home page

**Search results for apartments:**

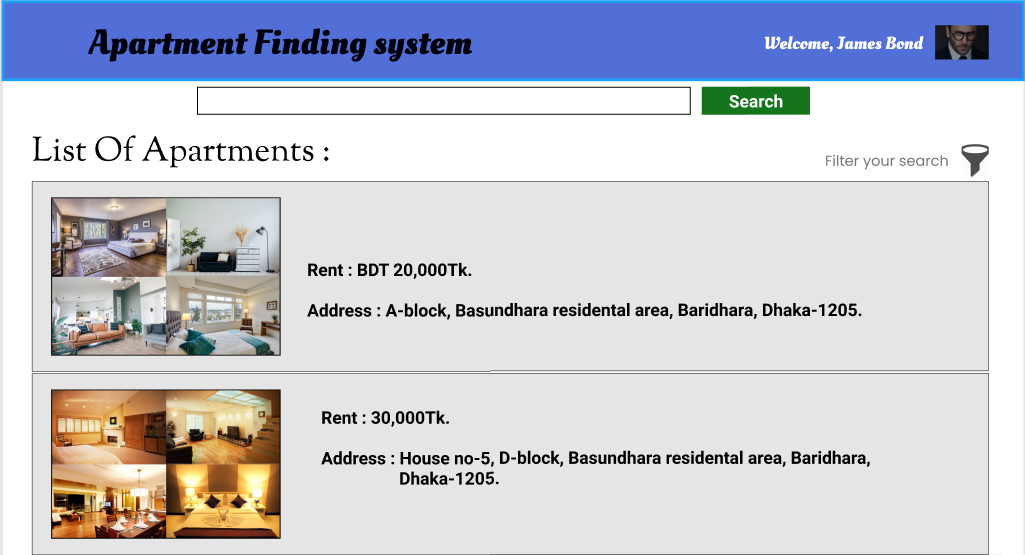


Fig 12: User interface of search result

**Apartments details:**

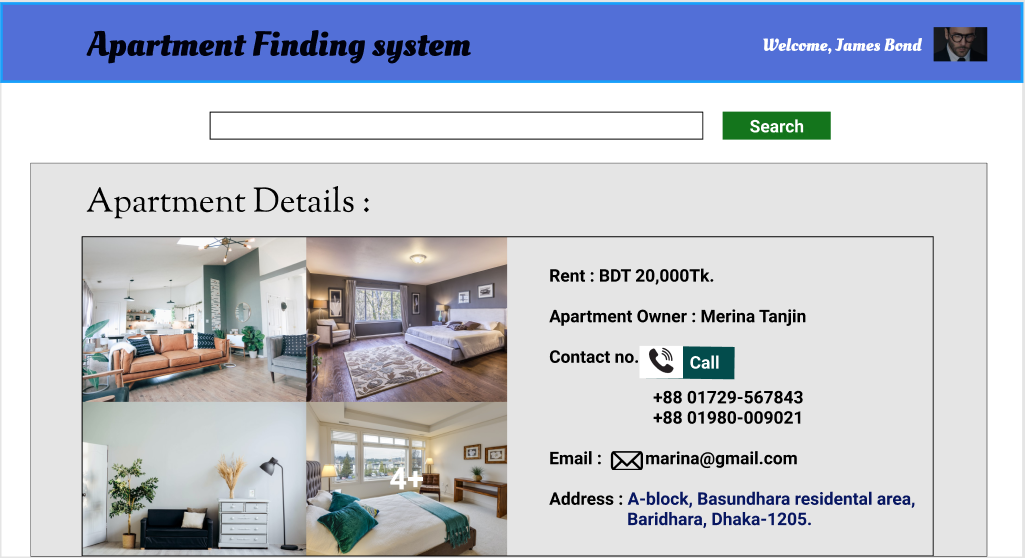


Fig 13: User interface of Apartment details

**Apartment owner dashboard:**

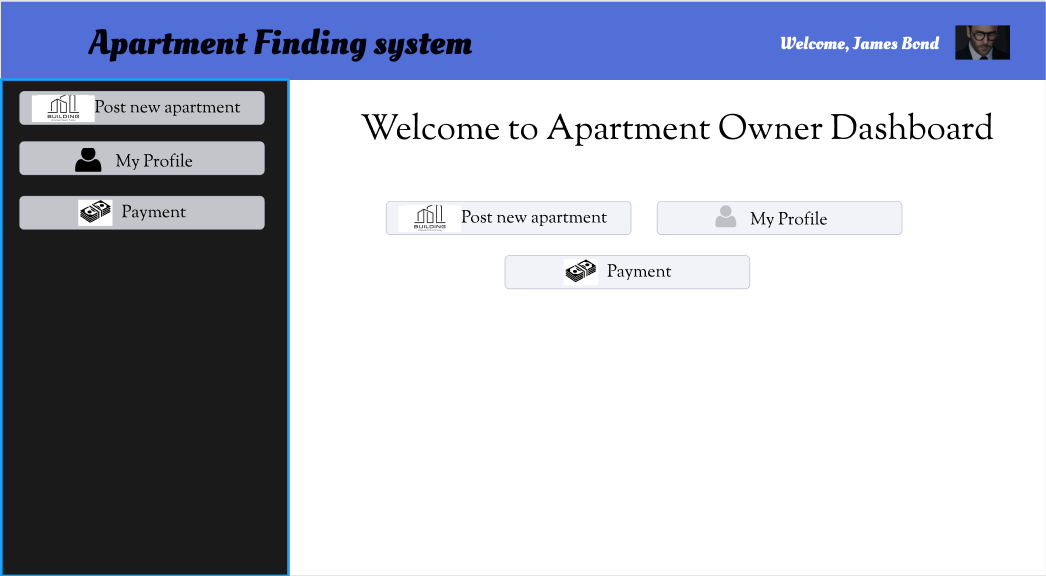


Fig 14: User interface of Owner dashboard

**Post new apartments for apartment owner:**

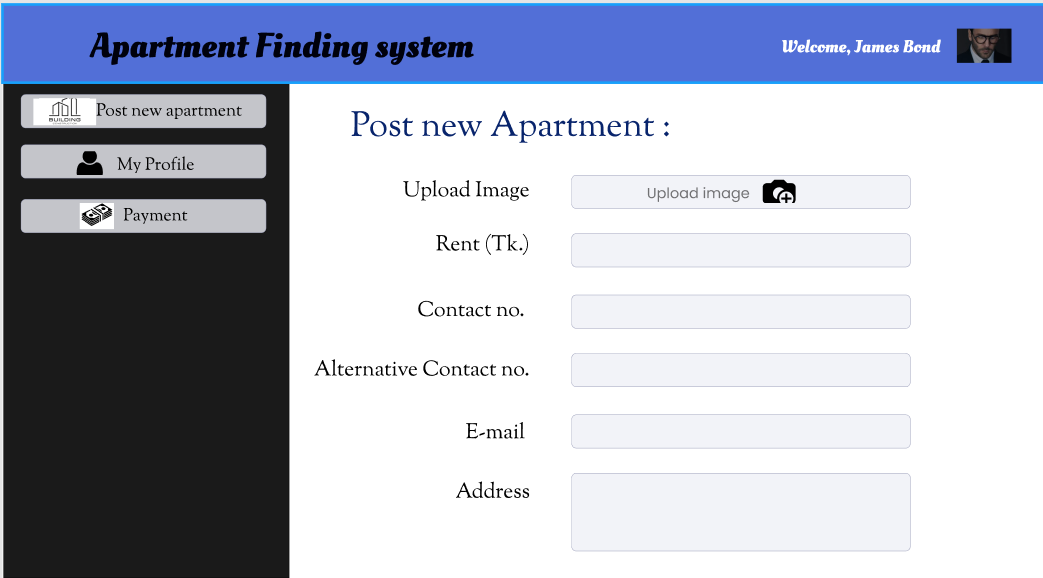


Fig 15: User interface of posting Apartment details

# 

# Chapter-4: Software Project Management Plan

**4.1.1 Document History and Distribution**

This software or Web Application is intended for implementing a Smart apartment finding system so that a customer can easily access our service from anywhere, anytime. This system can make a customer’s house shifting much easier. The development Smart Apartment System is a knowledge-based system where users can get all kinds of information about renting and buying apartments including cleaning and designing it. Also, users can enjoy every service by sitting at home. Users can benefit from this service by saving valuable time.

## **4.1.2 Distribution**

|  |  |  |
| --- | --- | --- |
| **Recipient Name** | **Recipient Organization** | **Distribution Method** |
| Abhijit Bhowmik | AIUB | Soft Copy |
| Abhijit Bhowmik | AIUB | Soft Copy |

Table 8: Distribution list

Soft copy Distributed copy mentioned on the table.

**4.2 Project Organization**

Project organization depends on three major structures

**4.2.1 External Interfaces**

The system user’s relationship will be responsible for formal interaction between the apartment owner and the customer contact. Necessary interaction will be done through the allocated manager on the area, but all discussions with the user will be documented clearly for record. All the administrative changes will be contacted between the manager and admin. All user requests for services or configuration item changes will be in writing and approved by the project’s Configuration Control Board (CCB), which consists of all team members.

**4.2.2 Internal Structure**

There are four developers for this project. All members have specified areas of responsibility and everybody contributes equally to the project. Because there are only four members on the team, each member holds more than one role. Two members will be responsible for the frontend development and the other two will be responsible for the backend development. But if they need any type of help the other members will be available for additional help.

The team members will change roles throughout the life of the project, and each member will continue to have more than one role.

## **4.2.3 Roles and Responsibilities**

The software developers are responsible for all documentation to be developed and also for all work to be done.

A few roles and responsibility of developers are-

1. Project manager
2. Project HR
3. Requirement documentation manager
4. Project head (responsible for daily scrum if we’re following the agile methodology)
5. All four members will be responsible for deciding the sprint.

## **4.3 Managerial Process Plans**

## **4.3.1 Project Start-up Plan**

This section describes the materials and resources required to start the project.  Because most of this information was pre-defined for the team, this section will not describe the rationale for many of these choices.

## **4.3.2 Estimation Plan**

As previously stated, the total development time is estimated to be 138 working days and the total internal cost to be BDT 437,152. These figures were obtained by expert judgment by analogy, that is, by comparison with similar projects.

## **4.3.3 Staffing Plan**

Each team member will be available for 8 hours per day as the project purpose. This time includes the team and supervisor meetings, document preparation and inspection, and tool development. At this moment

## **4.3.4 Resource Acquisition Plan**

* All resources for the project will be available at the start of the project and will not change substantially over time.  Below are the planned changes: The technical writer will change after completing a documentation
* The team member’s roles will change according to project needs

## **4.3.5 Project Staff Training Plan**

No additional staff training is needed for this project**.**

## **4.4 Work Plan**

Work Activities and Schedule Allocation

**4.5 Control Plan**

**4.5.1 Requirements Control Plan**

After the first release of the software requirement specification document, the requirement will be strictly followed. Then if any change is made in the SRS document, then it will be forward to the developers and project manager for checking the requirements. Changes will be made only when the approval come from the supervisor, project manager, and higher officials team member of the project. Without any approval, change in changing the SRS document or any requirements is strictly prohibited. Once a decision has been made, then the requirements will be changed in the software requirement specification document and that changed and updated version will be released to developers for the next development. If any change request is canceled then nothing will be changed in the SRS document.

**4.5.2 Schedule Control Plan**

The project schedule should be maintained correctly. If any projects schedule is left behind then the developers should spend some extra time on project work. On the other hand, some new developers can be added to the project for the delay schedule. Thus, the project schedule can be maintained in time and

the delivery of the final project in the time possible. There is also another way that we can compromise the quality and features. But it is not a good idea to compromise the quality and features, it will affect the whole project.

**4.5.3 Budget Control plan**

The project budget must be controlled very carefully. Because without this, the project will not be completed within budget. The budget should be reviewed monthly. Average monthly income can be discovered by the end of the year. **Project savings = total budget – current spending**. To increase the project savings is the main goal. Those expenses are reviewed or monitored by the project manager. Various types of report generation is also important, like weekly report, monthly report etc. If the monthly expense is increasing, then the project budget should be reviewed once again. If the plan goes correctly, that means the project costs a good profit.

**4.5.4 Quality Control Plan**

In this section, the quality of the project is controlled. Mainly, the project progress, budget and time are things by which quality is determined. Any changes of the SRS document can affect the quality of the project. Basically, maintaining the proper quality, the project must be finished within the budget and time. Sometimes it is seen that because of less budget and time, the quality is compromised. If there is less budget or time, it’s a good idea to reduce the number of features in stead of the project quality. It will increase the success rate of the project. There are daily meetings within the team member about the progress and problems. In those meetings the quality of the project is also maintained.

**4.5.5 Reporting Plan**

Reporting is also important for the successful completion of the project. If there is any change request in the SRS document then it must be reported first to the project manager and the advisor. Then there will be discussion about the change request and the final decision will be taken. If the developers face any problems, then they will also report to the project manager. And a meeting will be arranged to solve. The budget control team face any budget issue, if the progress rate is less to complete the project in time, then everything will be reported to PM to solve. Actually, there is weekly meetings, monthly meetings to discuss any reported issue and to report the project progress, about the budget and also the time management.

**4.5.6 Metrics Collection Plan**

Various metrics is used for software measurement and quality. Some metrics at various important for software quality, like as agile metrics, production metrics, security response matrix, dependence age, size-oriented measurements, function-oriented methods, pull request, customer satisfaction etc. for

Development of software progress velocity agile matrix is very useful. Production metrics keeps the record about active days failure and repair time productivity. Production metrics keeps the record about active days, failure and repair time and productivity. Size oriented measurements measures KLOC like how many lines of codes have been written? Determine bugs errors etc. Quality assurance metrics is also very important part for

development. It actually measures the software quality. And testing is a very important here. Customer satisfaction also keeps the record about the customer satisfaction from the project

**4.6 Risk Management Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risks | Probability | Impact | Rating | RMMM |
| Project Manager Availability | 50% | 3 | Medium | R-1 |
| Schedule slips | 70% | 1 | High | R-2 |
| System goes hour | 60% | 3 | Medium | R-3 |
| Project canceled | 30% | 4 | Low | R-4 |
| False feature rich | 40% | 2 | Low | R-5 |
| Programmers doesn’t have good experience | 50% | 3 | Medium | R-6 |
| Late delivery | 50% | 3 | Medium | R-7 |
| Customer Participation in Beta Testing | 30% | 4 | Low | R-8 |

Table 9: Risk management plan

**4.7 Closeout Plan**

Closeout plan actually occurs when the project is end or about to end the project. Then some following actions will be occurred like.

* The project teams will make a report about all documents source code plans, progress etc.
* The developer’s team will also copy all of the material in electronic format on a CD ROM.
* Then they will all report those to the project manager.

**4.8 Technical process plans**

Technical process plans define how much technical effort is required to develop the project. The quantitative inputs, lifecycle cost estimates in this technical planning. The planning accomplishes the technical activities and reduce the technical risks based on technical knowledge.

**4.9 process model.**

The XP (extreme programming) agile process model will be followed during the project implementation.

## 

## **4.10** **Methods, Tools and Techniques**

The project, E-Commerce, adapts the system on Personal Computer using HTML, PHP, Visual Studio 2012 and MySQL for database management system. Some other tools can be used like Adobe Dreamweaver, Adobe Photoshop, etc.

## **4.11** **Infrastructure Plan**

The project manager will formally accept each project milestone by signing proper acceptance documentation. The project manager will conduct an acceptance test at the end of each phase. This could lead to more demands for changes and enhancements. The project manager will conduct acceptance testing on the final product/application.

## **4.12** **Product Acceptance Plan**

The project manager will formally accept each project milestone by signing proper acceptance documentation. The project manager will conduct an acceptance test at the end of each phase. This could lead to more demands for changes and enhancements. The project manager will conduct acceptance testing on the final product/application.

## **4.13** **Supporting Process Plans**

The strategies for the supporting procedures that are part of the software project will be included in the Software Project Management Plan. Configuration management, verification and validation, software documentation, quality assurance, reviews and audits, problem resolution, and subcontractor management are just a few of the plans available.

## **4.14** **Configuration Management Plan**

Every project deliverable should be viewed as a configuration item. The configuration item and associated file will be titled after the document, such as SOW or SRS, and will include the version

number. All preliminary versions submitted to the project manager for approval, for example, would be given the acronym followed by 0.1, 0.2. This baseline document will be version 1.0 and issued to project members

after the project manager accepts the basic SPMP. The project manager's informal updates will be numbered 1.1, 1.2, etc., and the next full distribution to the committee will be version 2.0, etc.

## **4.15** **Verification and Validation Plan**

This project's Software Project Management Plan must include the software project's verification and validation plan, as well as tools, techniques, and responsibilities for the verification and validation work activities. The verification and validation plan will be contained in its own document and will be updated as needed.

## **4.16 D****ocumentation Plan**

For all documentation needs, IEEE standards would be followed. Before their baseline versions are released and sent to the members of the committee on the required dates, all of the papers will be discussed and evaluated with the project manager.

## **4.17** **Quality Assurance Plan**

The project manager will ensure that the quality of our project is maintained and verified. He will ensure that the project's quality is maintained.

## **4.18** **Reviews and Audits Plan**

The Software Quality Assurance and Verification & Validation Plan, which would be designed in accordance with suggested departmental standards, would include review and audits.

## **4.19** **Problem Resolution Plan**

All issues would be settled informally by the project manager and the developer. That is to say, there is no clear plan in place. However, if the necessity for such a plan emerges, the Software Project Management Plan will be revised accordingly.

## **4.20** **Subcontractor Management Plans**

There is no structure in place for managing subcontractors who may provide work products for the software project.

## 

## **4.21** **Process Improvement Plan**

The project manager will examine the project on a regular basis once it has been completed, and if any improvements are required, he will advise the developers

## **REFERENCES**

1. M. Nosworthy, G. Blanchard, R. Tripad, “bproperty,” 2015, <https://www.bproperty.com/>
2. C. Capital, F. Partners, “perforce.com”, 2014, <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>
3. J. graph, “appdiagram.com”, 2005, <https://app.diagrams.net/>
4. D. Field, “UI design”, 2016, <https://www.figma.com/files/recent?fuid=1001531349736242039>
5. Wikipedia, “software development", last edited 30 July, 2021, <https://en.wikipedia.org/wiki/Software_development>
6. M. Mohtashim, “tutorialspoint”, 2006, <https://www.tutorialspoint.com/software_engineering/software_project_management.htm>
7. M. Mohtashim, “tutorialspoint”, 2006, <https://www.tutorialspoint.com/software_engineering/software_requirements.htm>
8. B. Smith, “landing,” 2019, <https://www.hellolanding.com/blog/apartment-hunting-tips/>
9. A. Stringer, C. Stringer, “Molly maid”, 1979, <https://www.mollymaid.com/our-services/apartment-cleaning/>
10. A. I. Halim, “sheba.xyz”, 2015, <https://www.sheba.xyz/house-shifting-service>
11. S. Jain, “geeksforgeeks”, 2008, <https://www.geeksforgeeks.org/software-engineering-software-quality-assurance/>
12. Wrangler Topco. LLC, “time management”, 2006, <https://www.wrike.com/project-management-guide/faq/what-is-time-management-in-project-management/>