**Logo

Description automatically generated**

**Term Project**

**Name:**

**ID:**

**Course ID:** CSE307

**Course Name:** Systems Analysis and Design

**Section:** 2

**Group:** 30

**Instructor:** Sabrina Alam

**SECTION1:**

1. Introduction

“Residential Management and Monitoring System” - An automation for the easier residence works.

Bringing the pain of manual residential works automated, not only for the casual household works but also for the landlord.

1. History leading to project request:

Have been seeing this for a long-time by not only my family but also neighbors and so many others that I always pondered a solution of, people are generally lazy and sometimes deliberately miss deadlines and make excuses also sometimes miss them due to some unavoidable circumstances, workers lying to their owner taking advantage of not having proper evidence, with all the difficulties so many household family face seeing the complains in the social I always wanted to bring the end of this by creating a platform for automating and keeping records.

1. Identify Problem, opportunity:

• Hassle of doing things physically,

• A lot of physical activity that can be automated.

• If one person is not present then the whole chain is broken this is very much problematic when the work consists of multiple people to be present at the same time.

• Getting payments late due to not being able to pay physically or forgetting the payment date.

• Ask for assistance without any hassle.

• Workers tend to lie to steal money or not do the work deliberately.

• Not being able to trace if any crime took place by only CCTV footage.

1. Project goal and objectives

✓ Create a platform for all the committee members to chat and hold virtual meeting.

✓ A Committee member can propose a decision and others submit their opinion

✓ A renter can state rules and check if they are maintained by the rentee, guards and caretakers also warn them against any rules broken

✓ Receive rent from the rentee automatically even if he is not available physically

✓ Manager can be informed of any need through the app and receive feedback.

✓ A Manager can receive complain from rentee check if the complaint is authentic and act accordingly

✓ Flat owners & Renters can see the time of arrival and departure of servants.

✓ Payment can be done through the app for Servant / any other service.

✓ Guards keep record of visitors entering & exiting the building their NID number and time, increasing security.

**SECTION2:**

1. Product Description
2. Product Summary

Break disadvantage and consequences formed due the physical inability of being present or forgetting a work that had to be done stated in the building rule, keeping track of everyday visitors and servants work time to holding virtual meetings between committee of renters this is a go to app for every Apartment renter and rentee!

1. Product Stakeholders
2. Renter
3. Flat Owner (Self Living)
4. Rentee
5. Manager
6. Servant
7. Guard
8. Technician
9. Payment Agent
10. System Context diagram

Diagram

Description automatically generated

1. Hardware and Software detail

The hardware components that would be required are given below:

Smart phone or tablet

Laptop or desktop computer

Active Internet connection

The software components that would be required are given below:

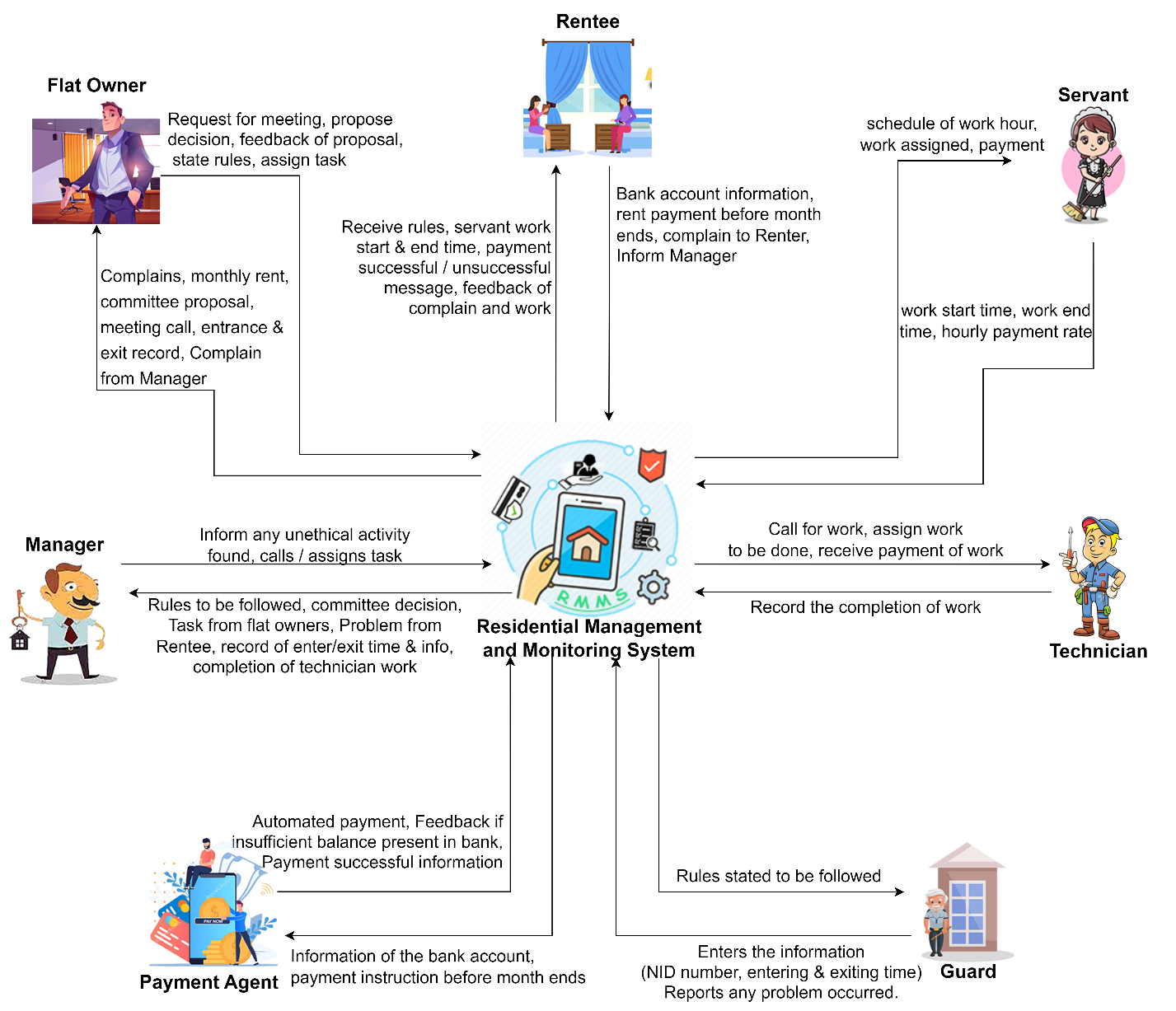
Figma

HTML, CSS (Bootstrap & Flexbox), JS

PHP

MySQL

Rich Picture:



1. Key Technical Features of Software

The technical features of the software are mentioned below:

* The process of sending and receiving rents will be much easier
* Secure transaction
* Data integrity and security is ensured
* The process of sending complains and assigning task will be much simpler
* Records will be kept of every action, reducing fraud activity
* Communication process is simplified

**SECTION3:**

1. Information Gathering methods (At least three methods)

Gathering accurate information is a very important aspect of developing the software that fulfils its goals. It is practically impossible to meet the needs of the users unless we consider their needs, ideas, and viewpoints. For my system, I've picked three relevant information collection methods: stories, interviews, and questionnaires

* 1. **Stories**

Surprisingly, my closest persons are the people who were victims of not having an automated system that would manage some basic household things on the move or at absence, my father being a businessman have to travel a lot and my mother being a school teacher have to be at school at times the rent is due and the owners charge extra as late fine, the housemaids tends to miss works and it becomes very difficult at times to catch, one day my mother was casually talking to me and saying “Things are advancing a lot and why isn’t there anything on the internet that keeps track of house works and pays automatically” from there I started wondering that an automation system is very much needed.

1.2. **Interviewing**

I have decided to meet with the most important stakeholders to have a better understanding about their needs and how they want it to be fulfilled with my application, because this application is hierarchy biased.

It is very important to know who my important users are and whether they are interested in using this app and will benefit them or not. Having an in-depth conversation with them will help to know what they exactly need and want.

Selected interviewee: Renter and Rentee

Questions:

1. In your opinion how much will you be benefited from this application?

2. What challenges and consequences did you face while not being able to be physically present for some works that have been proposed to be automated here?

3. Have this kind of idea ever pondered in your mind that having an automated system would have saved you from the consequences that you have faced?

4. Does this system is the one that you have been desiring for and conveys all the functions that crossed your mind?

5. If the answer to the previous question is No, then If you were in charge of this system, what features would you change / add?

1. Major functionalities offered by the system

1) Users will be able to use the app on a variety of devices.

2) Users will have the option of logging in as a Flat owner as well as renter if he is both.

3) The user interfaces for different will be distinct, but same themed.

4) The user will be able to both input and select the data that is required.

5) All saved data will be safe and secure and will not be tampered with.

6) For each person in a residence, a user profile will be created.

7) User will be allowed to edit profile anytime.

8) The information that the user has altered will be automatically updated and saved.

9) Automation of bill payment must be notified.

10) All user data will be encrypted for data security.

11) The user will be able to reset their password by submitting a request.

12) Any offensive words will be blocked.

1. Use Case Diagram

Diagram

Description automatically generated

1. Normal Scenarios (At least 2)

**Scenario 1:**

|  |  |  |
| --- | --- | --- |
| Use Case Name: Rent Payment | | Unique ID: SA-00001 |
| Actor(s): Rentee. | | |
| Stakeholders: Rentee, Payment Agent, Renter. | | |
| Description: Rentee pays the monthly rent from the app. | | |
| Triggering Event: User log in the app and tap on Payment, then monthly rent. | | |
| Trigger type: External | | |
| Steps Performed: | Information Required for Steps: | |
| 1. The registration interface loads. |  | |
| 2. Selects user type. | user having an account. | |
| 3. User makes the following Steps:  Press “Make Payment” button, select “Renter” contact, Select “Monthly Rent” in reason | Logging in with correct credentials | |
| 4. Taps the Send Payment button. | All required information must be filled up. | |
| 5. If all information is correct then the Monthly Bill will be paid. | Sufficient balance in account. | |
| 1. A success window will be shown to the user interface. | Successful message | |
| Pre-condition: User needs to be a rentee of the residence. | | |
| Post-condition: User must carefully input all required information and have sufficient balance. | | |
| Assumption: User wants to pay the monthly rent. | | |

**Scenario 2:**

|  |  |  |
| --- | --- | --- |
| Use Case Name: State Rule | | |
| Actor(s): Renter. | |
| Stakeholders: Rentee, Renter. | |
| Description: Renter states rule the app. | |
| Triggering Event: User log in the app and tap on State Rules | |
| Trigger type: External | |
| Steps Performed: | Information Required for Steps: |
| 1. The login interface loads. |  |
| 2. Selects user occupation. | User being a renter. |
| 3. User makes the following Steps:  Press “State Rules” button, Type the Rules in the box. | Logging in with correct credentials |
| 4. Taps the “State Rule” button. | All required information must be filled up. |
| 5. If all information is given then the rule will be stated for the renter. | All information is given. |
| 1. A success window will be shown to the user interface. | Successful message |
| Pre-condition: User needs to be a renter of the residence. | |
| Post-condition: User must carefully input all required information. | |
| Assumption: User wants to state a new rule for the rentee. | |

1. Alternate Scenarios (At least 2)

**Scenario 1:**

|  |  |  |
| --- | --- | --- |
| Use Case Name: Rent Payment | | Unique ID: SA-00001 |
| Actor(s): Rentee. | | |
| Stakeholders: Rentee, Payment Agent, Renter. | | |
| Description: Rentee pays the monthly rent from the app. | | |
| Triggering Event: User log in the app and tap on Payment, then monthly rent. | | |
| Trigger type: External | | |
| Steps Performed: | Information Required for Steps: | |
| 1. The registration interface loads. |  | |
| 2. Selects user type. | user having an account. | |
| 3. User makes the following Steps:  Press “Make Payment” button, select “Renter” contact, Select “Monthly Rent” in reason | Logging in with correct credentials | |
| 4. Taps the Send Payment button. | All required information must be filled up. | |
| 5. If all information is correct then the Monthly Bill will be paid. | Error Message insufficient balance in account. | |
| 1. Error Message | Unsuccessful message | |
| Pre-condition: User needs to be a rentee of the residence. | | |
| Post-condition: User must carefully input all required information and have sufficient balance. | | |
| Assumption: User wants to pay the monthly rent. | | |

**Scenario 2:**

|  |  |  |
| --- | --- | --- |
| Use Case Name: State Rule | | |
| Actor(s): Renter. | |
| Stakeholders: Rentee, Renter. | |
| Description: Renter states rule the app. | |
| Triggering Event: User log in the app and tap on State Rules | |
| Trigger type: External | |
| Steps Performed: | Information Required for Steps: |
| 1. The login interface loads. |  |
| 2. Selects user occupation. | User being a renter. |
| 3. User makes the following Steps:  Press “State Rules” button, Type the Rules in the box. | Logging in with correct credentials |
| 4. Taps the “State Rule” button. | All required information must be filled up. |
| 5. If all information is given then the rule will be stated for the renter. | All information is not given. |
| 1. An unsuccess window will be shown to the user interface. | unsuccessful message |
| Pre-condition: User needs to be a renter of the residence. | |
| Post-condition: User must carefully input all required information. | |
| Assumption: User wants to state a new rule for the rentee. | |

1. Functional Requirements

1) Users will be able to use the app on a variety of devices.

2) Users will have the option of logging in as a Flat owner as well as renter if he is both.

3) The user interfaces for different will be distinct, but same themed.

4) The user will be able to both input and select the data that is required.

5) All saved data will be safe and secure and will not be tampered with.

6) For each person in a residence, a user profile will be created.

7) User will be allowed to edit profile anytime.

8) The information that the user has altered will be automatically updated and saved.

9) Automation of bill payment must be notified.

10) All user data will be encrypted for data security.

11) The user will be able to reset their password by submitting a request.

12) Any offensive words will be blocked.

1. Non-Functional Requirements (At least 5)

Reliability:

• The system must be always available and should not experience any downtime.

• All tasks must be completed without any software-related errors or with minimal errors.

Performance:

• It should show payment made in previous payment window in 2/3 seconds.

• The user interface must be visible in less than five seconds.

Maintainability:

• All this system's data must be maintained safe no matter what happens to it; hence the system will require data backup planning.

• If an error happens during any of the system's processes, the system will be able to trace each fault and correct it as quickly as possible before continuing the operation.

Efficiency:

The system must be able to accommodate at least 10,000 tasks at the same time while also consuming minimal resources such as memory, CPU, and disk space.

Security:

• To utilize this system, users must first register, after which only registered users can access the system's features by logging in with their ID and password.

• Ascertain that the system, or any of its data, is never subjected to malware attacks or illegal access.

• Unless it is required, a user's personal information or other data should not be available to other users.

Scalability:

This system must be scalable because it will be used by anyone living in Bangladesh to order sweets 24 hours a day from sweet businesses all throughout the country. That is, regardless of how much memory, servers, or disk space is required, the system should be able to accommodate an increase in the number of users and processes without affecting performance.

**SECTION4:**

1. Entity Relationship Diagram

**Diagram, engineering drawing

Description automatically generated**

1. Logical Data Flow diagram

**Diagram

Description automatically generated**

1. Physical Data Flow diagram

**Diagram

Description automatically generated**

1. Activity diagrams
2. Activity diagram for Renter
3. **Propose Decision**

**Diagram, schematic

Description automatically generated**

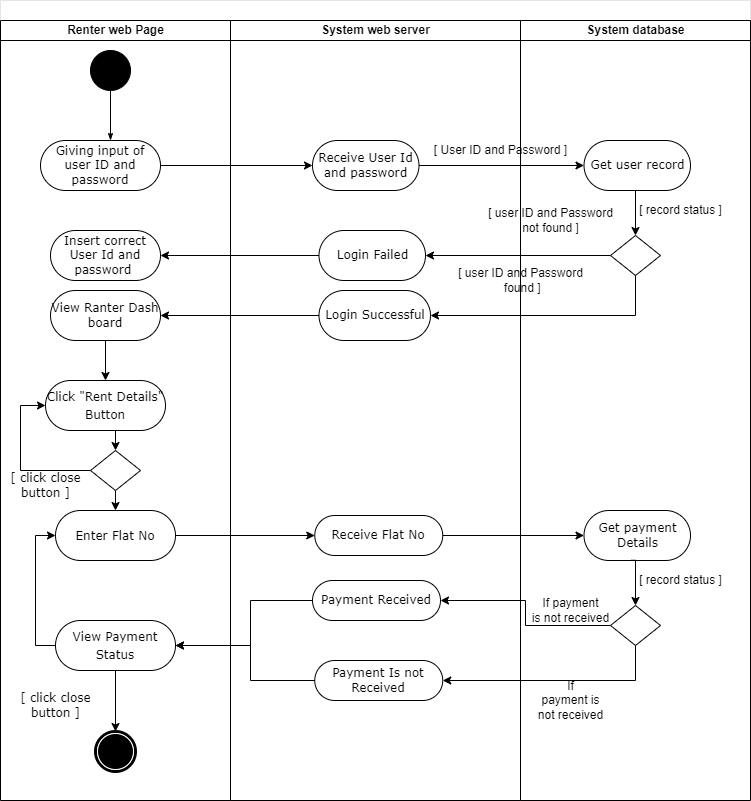
1. **Diagram, schematic

   Description automatically generatedAssign Task**

**Diagram

Description automatically generated3. State Rules**

**4. Receive Rent**

****

**5. Request for Meeting**

Diagram

Description automatically generated

**6.View Complain**

Diagram, schematic

Description automatically generated

1. **Activity diagram for Rentee**

**1. View Rules**

**Diagram, schematic

Description automatically generated**

**2. Make Payment**

**Diagram, schematic

Description automatically generated**

**3. View servant schedule**

**Diagram, schematic

Description automatically generated**

1. **View complain feedback**

**Diagram

Description automatically generated**

1. **Activity diagram for Servant**

**1. View assign work**

**Diagram

Description automatically generated**

**2. Receive payment**

**Diagram, schematic

Description automatically generated**

1. **Activity diagram for Technician**

**1. View work**

**Diagram

Description automatically generated**

**2. Submit completion of workDiagram

Description automatically generated**

1. **Activity diagram for Guard**

**1. View rules to be followed**

**Diagram, schematic

Description automatically generated**

1. **Activity diagram for Manager**

**1. View Rules**

**Diagram, schematic

Description automatically generated**

**2 . Assign task**

**Diagram, schematic

Description automatically generated**

**3. Inform Unethical Activity**

**Diagram, schematic

Description automatically generated**

1. Sequence diagrams

**View Complain**

**Diagram

Description automatically generated**

**State Rules**

**Diagram

Description automatically generated**

**Propose Decision**

**Diagram

Description automatically generated**

**Assign Task**

**Diagram

Description automatically generated**

**Make Payment**

**Diagram

Description automatically generated**

**Do complain**

**Diagram

Description automatically generated**

1. Communication diagrams

**Diagram

Description automatically generated**

1. Class diagrams

Diagram

Description automatically generated

1. State-chart diagrams
2. CRUD matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Renter** | **Rentee** | **Guard** | **Servent** | **Employee** | **Manager** | **Building** | **Technician** | **Payment** | **Rules** | **Task** | **Complain** |
| Assign rules | **R** |  |  |  |  |  |  |  |  | **C** |  |  |
| Assign task | **R** |  |  |  |  |  |  |  |  |  | **C** |  |
| View rules |  |  |  |  |  |  |  |  |  | **R** |  |  |
| Add Renter | **C** |  |  |  |  |  |  |  |  |  |  |  |
| Add Guard |  |  | **C** |  |  |  |  |  |  |  |  |  |
| Add Rentee |  | **C** |  |  |  |  |  |  |  |  |  |  |
| Add employee |  |  |  |  | **C** |  |  |  |  |  |  |  |
| Delete rules |  |  |  |  |  |  |  |  |  | **D** |  |  |
| View task |  |  |  |  |  |  |  |  |  |  | **R** |  |
| Submit work |  |  |  |  |  |  |  |  |  |  | **U** |  |
| Do complain |  |  |  |  |  |  |  |  |  |  |  | **C** |
| View Complain |  |  |  |  |  |  |  |  |  |  |  | **R** |
| Update building info |  |  |  |  |  |  | **U** |  |  |  |  |  |
| Add manager |  |  |  |  |  | **C** |  |  |  |  |  |  |
| Print payment bill |  | **R** |  | **R** | **R** | **R** |  |  | **R** |  |  |  |
| Add servent |  |  |  | **C** |  |  |  |  |  |  |  |  |
| Delete servent |  |  |  | **D** |  |  |  |  |  |  |  |  |
| Update rules | **R** |  |  |  |  |  |  |  |  | **U** |  |  |
| View rent status |  | **R** |  |  |  |  | **R** |  |  |  |  |  |
| Delete Guard |  |  | **D** |  |  |  |  |  |  |  |  |  |
| Add new building info |  |  |  |  |  |  | **C** |  |  |  |  |  |
| Add Technician |  |  |  |  |  |  |  | **C** |  |  |  |  |
| Delete building info |  |  |  |  |  |  | **D** |  |  |  |  |  |

**SECTION5:**

1. Structure English pseudo code for the system
2. **Sign In**

**Browse the interface**

Select Sign In

Enter Name

Enter Email

Enter Password

Retype Password

Confirm sign in

an account created notification will be displayed.

1. **Login**

**Browse the interface**

Select Sign in

Enter Email

Enter Password

Confirm

If (credential matches)

User will view his/her interface.

Else

Incorrect password notification will be displayed.

1. **Do Complain**

User will select the complaint button to do complain.

User will select the user type to complain

Enter user id

In the complaint box the user will write down the complaint details.

Enter submit button

If (submitted successfully)

Users will view a successful notification.

Else

Not saved notification will be displayed.

1. **View Complain Reviews**

User will select view complain button

If (Previous complain is available)

User will see the list of complain

Select complain from the list

User will show the review details.

Else

No complain is displayed

1. **Set Rules**

User will select the set rules button to set the rules.

User will select the user type to whom the rules are stated

In the box the user will write down the rules in details.

Enter submit button

If (submitted successfully)

Users will view a successful notification.

Else

Not saved notification will be displayed.

1. **Do Payment**

Enter the payment button.

Select month

Select payment details

Select payment method

Enter amount

Enter submit

If (submitted successfully)

Users will view a successful notification.

Else

Not saved notification will be displayed.

1. Prototype the user interface Graphical user interface, application

   Description automatically generated

