Project Report

RENTAL ACCOMMODATION MANAGEMENT SYSTEM

MODULE TITLE: SOFTWARE ENGINEERING

MODULE CODE: B9IS119

GROUP MEMBERS:

SINGUPURAM NAVIN KUMAR | 10612366

DIGVIJAY JATKAR | 10508676

RASHMEE SAXENA | 10632656

MODULE LEADER: YUSUF AYTAS

TABLE OF CONTENTS

TABLE OF FIGURES

1. OVERVIEW

When all we want is merely stuff that works, technology is all we have. There is a pressing need to embrace and value technology given the present paradigm change in the technological industry. The housing sector continues to be alert to the difficulties of change by implementing a new approach that makes managing rental properties simple. Therefore, it is necessary to create a system for managing rental properties that may make life easier for rental managers and increase the productivity of all of their work.

Landlords struggled to find tenants throughout the year and vice versa. There is no appropriately allocated housing, and it is difficult to set up the system per user or client interests. Additionally, the house management system is virtually entirely manual. It will take a lot of time, money, and human resources for students and people from other cities to find the ideal preferences and accommodations they had anticipated.

Understanding the challenges people encounter in finding housing, we developed an idea that aims to create a system that will be advantageous to both tenants and property owners and speed up the process of looking for homes and renting a virtual system where property owners can quickly discover tenants and vice versa with a pre-determined agreement for their renting purposes. The system will be used to assist a variety of customers, including those who are looking for commercial properties, student housing, paying guest apartments, and other types of real estate, while also taking into account their preferences for things like parking spaces, garages, and other specific requirements.

We have supported several services with the system's design and implementation while keeping the software principles in mind. The login service was originally incorporated into the design to give authentication and authorization services to user roles like customers, owners, etc. Users can register for the system using their Google base authentication or a variety of additional methods, depending on how it is implemented. They can log in with their credentials and make advertisement listings or view them with categories once they have registered in the system and agreed to the privacy policies. Additionally, they will be permitted access to all system capabilities. After viewing the advertisement, if a user is still interested, they can get in touch with the landlord directly using the push notification messaging feature.

An essential component of our application is real-time communications considerations. The majority of other websites either offer the option to phone or email the owner, which would be laborious and time-consuming to wait upon the needs; these other websites do not feature such laborious capabilities. However, we added the built-in chat feature after considering the difficulties clients confront in the actual world. Therefore, a client can send an owner a direct message, and the owner will receive push alerts. We also assist the client in providing updates if the owner or advertisement lister is unresponsive. In the advertisement section, we have added filtering and sorting options along with the price or date filtering options due to the budget capacity or choice and also the latest listing to understand more choices of their preferences. These options can be used while browsing or viewing any of the listed advertisements on the website.

Out of all the system's capabilities and services, we concentrated on user experience design to boost business efficiency. We have an idea to add a lot more features and alternatives for the convenience of the clients and the owners, etc., based on the contact with the system. Support the inclusion of the feature of a helpline to further maintain the system.

2. BUSINESS REQUIREMENTS

To determine the requirements for system functions, the business requirements requirement analysis defines customer demands and objectives in the context of intended customer use, surroundings, and specified system characteristics. It required user participation, declarations of facts and presumptions that outline the system's expectations in terms of the mission's goals, the surrounding environment, its restrictions, and metrics for measuring performance and suitability. In essence, the users:

- A method that increases the effectiveness of information retrieval and storage.
- A simple to understand and use the system.
- A system that processes transactions quickly.
- A flexible, secure, and practical system.

We would like to deep dive into the functionality and features implemented:

- An authentication system for the application has been designed and developed so that the users can use only the features such as posting advertisements or messaging to the lister. So, to use the above features the users need to be authenticated by creating a new account of their own by filling out a few terms and conditions. Once signed up, they can use the features with ease.

- We have added the ability to buy, sell, and rent properties in consideration of the preferences and needs of various clients. Additionally, with this implementation, a variety of audiences will be able to use the system.
- The users' ability to provide reviews and feedback has been included as a new feature. According to the survey and reliable sources of information, this will be one of the key components of the system. For instance, if one of the owners legitimately accessed the website with a certain advertisement but ultimately discovered that the owner was conniving or if any improbable events occurred during the chat, they can always express their opinions.
- We introduced a reporting function where the owner himself can report that they have come across a bad user or the user can report vice-versa that they came to know about a poor owner to boost the efficiency and productivity of the system without wasting any time of the owners or the users. Along with this, they can, of course, also report on specific advertisements and communication that occurred during the event.
- As well with adhere to make the search easier for the users as well to the owners, the feature like sorting, searching, and filtering options to view the advertisements. With this, the users can be able to find the particular budgeted properties or accommodations with the required specifications.
- Push alerts notifications and instant messaging are the system's most noticeable features. believing that it is essential to incorporate into the design to facilitate quick and straightforward communication between clients and property owners/landlords. The landlords can be reached by clients over the phone or by email. With the advent of this option, the work is simpler, more time-saving, and more effective for those who do not want to reveal their emails or phone numbers to avoid scammers.
- A feature for customer support is also included. With that said, if consumers encounter any issues, we will assist them with live assistance. If a user has difficulty with the service or a feature or any type of problem and without any customer assistance included, we think you'll lose that user as a client. To increase the number of clients we have also included that functionality because we understand the user experience and evolved with it.

We have now described the contents of our application. Let's clarify what got excluded from the scope of the system:

- We won't be adding payments to our website. Our goal is to assist clients in finding only what they require. For the time being, we will not be adding payments to our website. Since we do not accept payments on our website, our application will not have any third-party payment authentication in the system.
- Our website is similarly devoid of any legal documents. If users reach an agreement, they are responsible for resolving these issues on their own.
- As a first step, we won't be adding features to our website like a budget calculator. But depending on the feedback we would be introducing it later.

This system properly explains the issue's in and out scopes. Initially, we added the in-scope features. As the website gains in popularity, we may broaden its scope to incorporate new functions. We decided to include these capabilities solely for the time being though, as our resources and time are now constrained.

3. AGILE METHODOLOGY

In agile software development, epics are used to summarize a product feature. User stories are frequently a component of the epic and outline particular user behaviors that must take place to achieve their main objective. Epics and user stories are the most widely used methods for expressing features in the software development process, even though there are numerous other agile methodologies for producing new software.

3.1 EPIC AGILE METHODOLOGY

An epic need to be divided into stories and tasks because it is most likely too huge to fit into a sprint. Typically, epics are outlined in the initial product roadmap or backlog and broken down into stories when more information about the product list becomes available. In story mapping, epics are written in a user story structure. A shared goal and a clear conclusion, a significant user requirement, or a step in the journey or process involved in utilizing the product are all present in an epic's storylines. An epic is a big document that lists the characteristics of your product under the agile process.

3.2 USER STORIES AGILE METHODOLOGY

A user story is a short, written description of how the software should work from the perspective of the end user. A good user story will include an action or event, an actor (the person or thing that makes the request, or the user), and a result. The result can be either in direct response to the action or event of the user story, or it can be something that follows the action.

For example:

- I want to know about forthcoming events so I can plan my visit as a consumer.
- I want to be able to view future events on my phone as a visitor.

Accordingly, let us see what we have functionalities covered in our project as chunks of user stories and epics:

Epic - Login Authentication and Authorization

- Design a signup form and store information in the database
- Login validations and authentication for the users
- Reset password functionality or generate a new password

The process starts with gathering user data and saving it so that users can enter details and the functionality is implemented to save the data gets stored in databases so that they can subsequently be utilized to show information on the website. This data can initially be used to display users' contact information during login.

Epic – User (i.e. Owner) choice to create an advertisement for the property

- The owner can publish advertisements as per his property choice
- The deal can be closed once the bargain and meetings are over

A rental accommodation system's key feature is the ability for users to publish rental property advertisements so that potential tenants or buyers can use that information and go further. This tool allows you to upload details about the property along with a few images and text-based or supplemental information. There are numerous alternatives to renting, such as housing for students, common areas, parking, storage, and commercial structures.

Epic - User capacity to transmit information for further details and request viewing

Users have the option to submit their personal information and indicate their level of interest on the website to schedule a viewing session.

Epic - Give the owner or landlord the power to rent out or advertise to sell.

- The ability to post a property-selling advertisement
- The capacity to end a campaign once the house has sold

With the use of this function, users can market their properties for sale or accommodation for rental purposes. This function functions similarly to renting a house.

Epic - Rent a property and reserve a slot from the system

- Access an advertisement's desired information
- The capacity to express interest in a property

With the help of this feature, users can view the details that the seller has posted about the properties as well as the various options that are open to them. This feature doesn't always prioritize a better user experience.

Epic - Build better user experience through interface and features

- Allow the user to sort the search results.
- Allow users to add filters to the search results.
- Allow instant messaging with customer service for the user.
- Allow users to report fraudulent ads.

Users won't have to scroll through the entire result set because this feature applies to all properties, which is a bonus. The user can narrow down the result set and set some preferences to satisfy their request.

This also includes reviews and ratings of the customer experience; we can do this by capturing both the buyer's and the seller's perspectives. The user may benefit from additional features like budget calculation to aid in decision-making.

4. CONTEXT MODEL

Contextual Design is a method for gathering information about users in the field, interpreting and compiling that information in a structured way, using that information to develop and prototype ideas for products and services, and iteratively testing and refining those ideas with users. All models of contextual design are graphical representations of the structure, conduct, and experiences of human activities. An integral component of a developer's world is diagrams. A context model outlines the organization and upkeep of context data. It is crucial in assisting effective context management. It is utilized to represent the components' reusable context data. The objective of this study is to create a formal or semi-formal description of the context data that is available in a software design. The context model also illustrates how IT applications function about their users and the companies they service. They are occasionally referred to as conceptual models, high-level design models, and enterprise architecture models. Models of the context show how the main object communicates with its surroundings, whether through the exchange of information, things in the real world, or monetary.

Below is the figure that demonstrates the context model for Rental Accommodation management system:

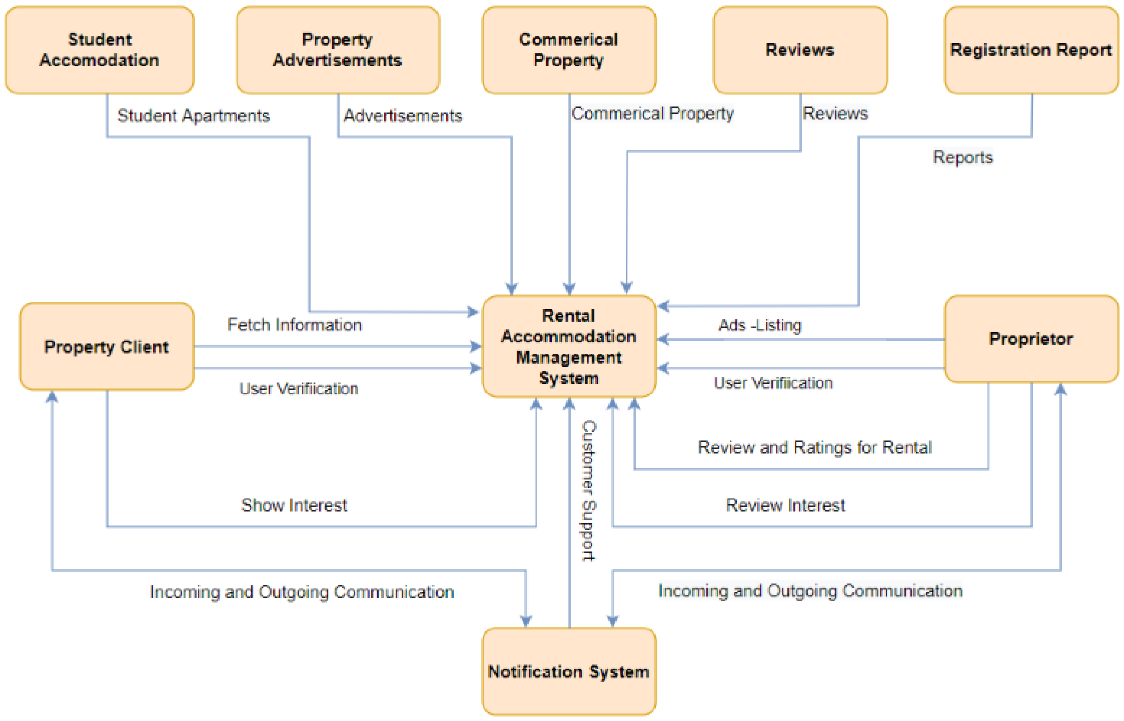


Figure 1: Context Model

The above figure states the context model which is used to confirm the project scope and also identify the potential impacts of the changes required to complete an overall project.

- External entities with which the main entity interacts are shown in the boxes (organizations, departments, systems, or processes)
- The connectors: use to interact between different entities
- The interactions' flows are labeled to show how information, physical objects, or financial resources move between the entities.

The system has two different types of users i.e. the proprietor and the tenant or the client. I would like to start by clarifying that the owner can publish many kinds of adverts once they log in, including those for rental apartments, student housing, commercial property, parking, and storage. Next, we have a user who is referred to in the system as a client or tenant. The customer can peruse the ads listed by owners or proprietors after logging into the system. The user has the choice to communicate with the owner directly using push notifications and instant messaging and then continue. Additionally, the user has the option to provide reviews and feedback, which may be helpful to other users.

Whatever your line of work, the context model is a simple and effective tool from a business standpoint. Before the project gets off track, it can be used to have a quick discussion, clear up any misunderstandings, and improve communication within the company.

5. ENTITY RELATIONSHIP DIAGRAM

We are aware that the entity-relationship model diagram, often known as an *ER diagram*, is a graphical depiction of the relationships between individuals, things, ideas, and events within an information system. It is one of the crucial pillars of database design because it also offers the graphics needed to build a database. It aids designers in comprehending the many inputs required by the information system. The objects that make up the system's entities are nothing more than those things. The attributes of the entity, which are its qualities, are further differentiated as primary keys or foreign keys to denote the significance. The connection or communication between these things is shown by the relationship. One-to-one, zero-to-many, and many-to-many are just a few examples of the different kinds of relationships.

The below ER diagram shows the entities present in the rental accommodation management system. The system can be used by the clients and the proprietor or owners therefore we maintain the user's entity to maintain the user information of the authenticated customers. The client id has been added as the primary key in the client table because we are maintaining the client and proprietor's information separately and the proprietor_id as the primary key in the proprietor table. Password_Encrypt and the security question are stored as attributes that are used when someone logins into the system. The owner and another client who can access the advertisements to obtain the desired requirements are both logged in to the system as separate users. In order to obtain and keep the information, the proprietor or owner has attributes like proprietor_id of the owner, also there are other fields like proprietor_name,proprietor_address, proprietor_contact number, proprietor_email, proprietor_address, and proprietor_review_id.

The proprietor can post one or many advertisements, so we can see the property_advertisement is connected with the proprietor with one to many optional relationships. The primary key for the property advertisement entity's storage or retrieval of associated data is advertisement_id. Other characteristics of the property advertisement include the proprietor_id, which provides information about who posted the advertisement. The attribute that identifies the different types of properties is advertisement type. As we can see, there are various kinds of properties i.e. student accommodations, commercial property, parking, etc. hence, we keep track of the information so that adding or removing the type of property and retrieving each advertisement type separately will be simple and the information of the property type can also be avoided redundancy with good productivity in mind with future design changes.

The advertisement type entity has attributes like advertisement_type_id, advertisement_type_description, and advertisement_type_title which is having one to one relationship with the property advertisements. Property_advertisements entity has other attributes such as advertisement title, description, price, and availability information that the owner can post to provide additional inspiration for the advertisements. The customer, who has access to the information the owner posts on the website, is another significant entity. A distinct identifier called Client_Id will be used to access the system's user data and the other attributes which are client name, mobile number, email, address, and review id. Upon logging in, the client can view one or more advertisements, so the relationship between the client table and the listings advertisement property, there are one to many optional relationships. A reviews entity has also been added to the system to track user satisfaction. Any user may submit a report, a complaint, or a rating, and this table may be used to keep such information.

As it's an optional field, there may be 0 or more reviews we preserved it as zero to many relationships. For the reviews entity, there are attributes such as review_id, client_id, proprietor_id, rating, and review description which describe more detail about the entity. Thus, we have provided an explanation of the entity relationship diagram for the rental accommodation management system.

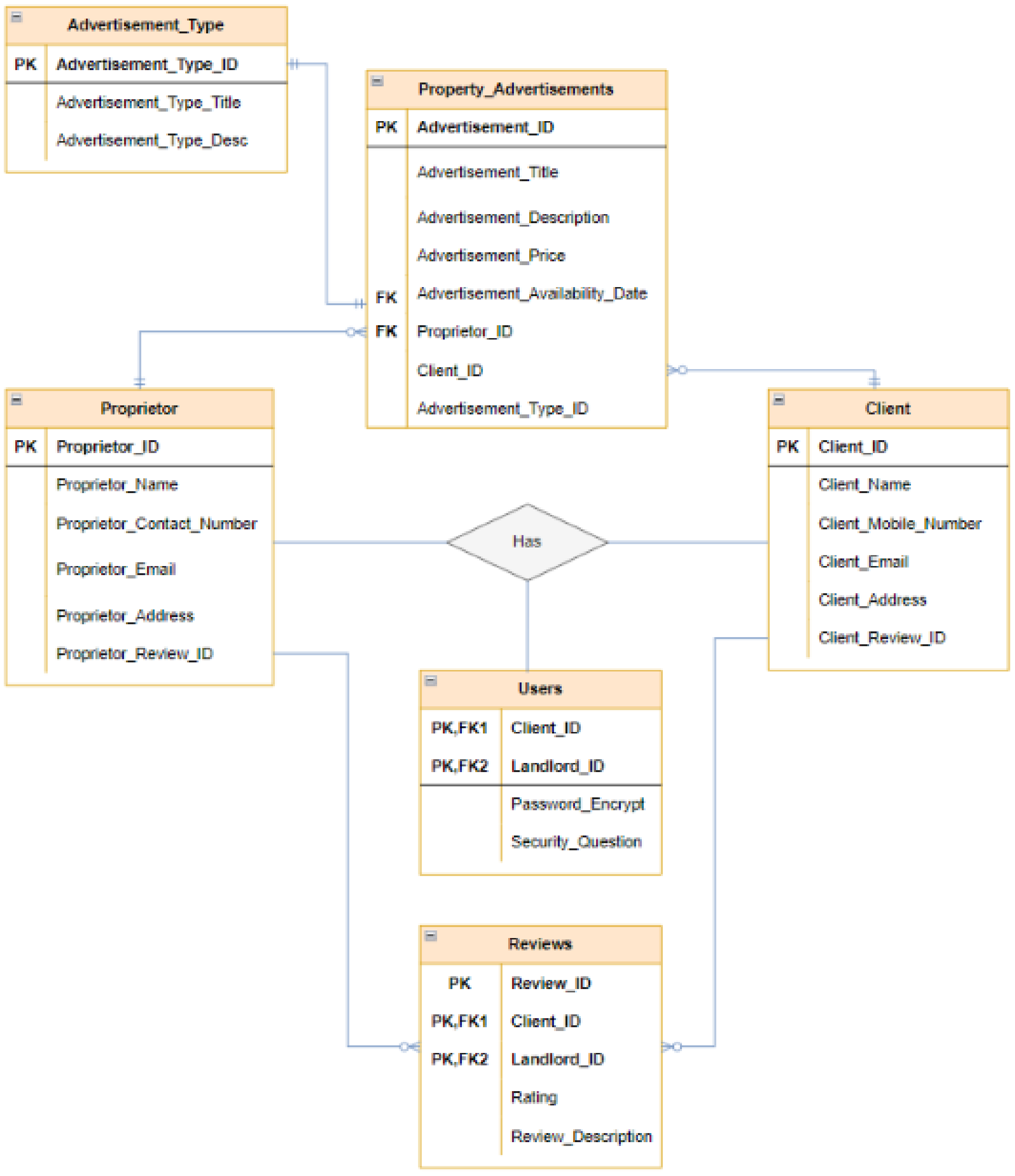


Figure 2: Entity Relationship Diagram

6. CLASS DIAGRAM

The primary component of object-oriented modeling is the Class Diagram. It is used for detailed modeling, which converts the models into computer code, as well as for general conceptual modeling of the application's structure. Data modeling can also employ class diagrams. The advantages of class diagrams are numerous for any company. Class diagrams help us better understand the overall layout of an application's schematics, regardless of how basic or complex the data models for information systems may be. An implementation-independent description of the types used in a system is provided, together with descriptive charts that emphasize any specific code that must be coded and implemented to the stated structure. These charts are then passed between the system's components. Additionally, it communicates any specific needs of a system visually and does so across the entire company.

One of the diagrams used in software development, the class diagram, displays the system's object classes as well as the relationships between them. Class Diagrams only show how object-oriented systems function. A static sort of structure diagram called a class diagram displays the classes in a system along with the functions, connections, and characteristics of each class. It is employed to describe the structure of a system. A rectangular box with three sections serves as the visual representation of the class diagram. The class name is specified in the top compartment, the attributes are specified in the second or middle compartment, and the methods or operations are displayed in the third or bottom compartment.

As shown in the below figure for Rental Accommodation Management System:

The Rental Accommodation System, Proprietor, Accommodation Business Registration, Personal Identity, Property Advertisement, Tenant, Preferences, Review, Accommodation Property, Apartment accommodation, Student Accommodation, Commercial Property, Accommodation Parking, and accommodation Storage. The rental accommodation system consists of four attributes Name, Phone Number, Email Address, and Rental Status. The rental accommodation system contacts the proprietor class which has four attributes including proprietor_name, proprietor_contact_number, proprietor_email, and proprietor_address. They are both connected through contacts and payments that have a multiplicity value of 1. As a result, there is exactly one instance on each side.

The rental accommodation system assigns a review class which has attributes review_id, review_description, and review_date. To show that a rental accommodation system could have multiple agents connected to it, a rental accommodation system. The review must be submitted with a multiplicity value of 1 and a multiplicity value of 1..* at one end of the line. A Preferences class with seven qualities is included in the review: location, cost, size, number of bedrooms and bathrooms, garage, and parking. At one end of the line, from * on the preferences to * on the review, there is a multiplicity value of 1.

Review oversees with property advertisement class with characteristics involving ad_id, ad_types, and ad_description. With a multiplicity value of 1..* on the review to 1 on the real estate advertisement, at one end of the line. The proprietor class then accepts a multiplicity value of 1 from both the proprietor class and the property advertisement class. Then, a property advertisement with the following attributes are created and delivered to the tenant class: tenant name, tenant login ID, tenant contact number, tenant email address, and tenant vehicle registration.

A property advertisement with a multiplicity value of 1 to 1 is located at the end of the line on the tenant side. It has a multiplicity value of 1..* on both sides and is tied to the Preference class. Person Class with Personal Legal Documents and Person Information, Accommodation Business Registration Class with characteristics business info, and Legal Documentation business all interact with 1..* on both sides and are connected to the proprietor class.

Proprietor Class and Accommodation Property Class are connected. With a multiplicity valve of 1 on both sides, the landlord class interacts with and rents to properties with the attributes Property Types and Property IDs.

They, therefore, have precisely one instance on each side and five classes connected, including student housing, commercial property, parking for accommodations, and storage for accommodations, as well as characteristics like size, rooms, facilities, and building standards for apartments and student housing. Size, quality, and business parking are among the characteristics of commercial property. Size, yard space, and garage are aspects of the parking class for accommodations. Size and yard space-related parameters for the accommodation storage class. The diagrams are perceived as descriptions of various objects. The domain's concepts are fundamental ideas that will logically relate to the classes that carry out their implementation.

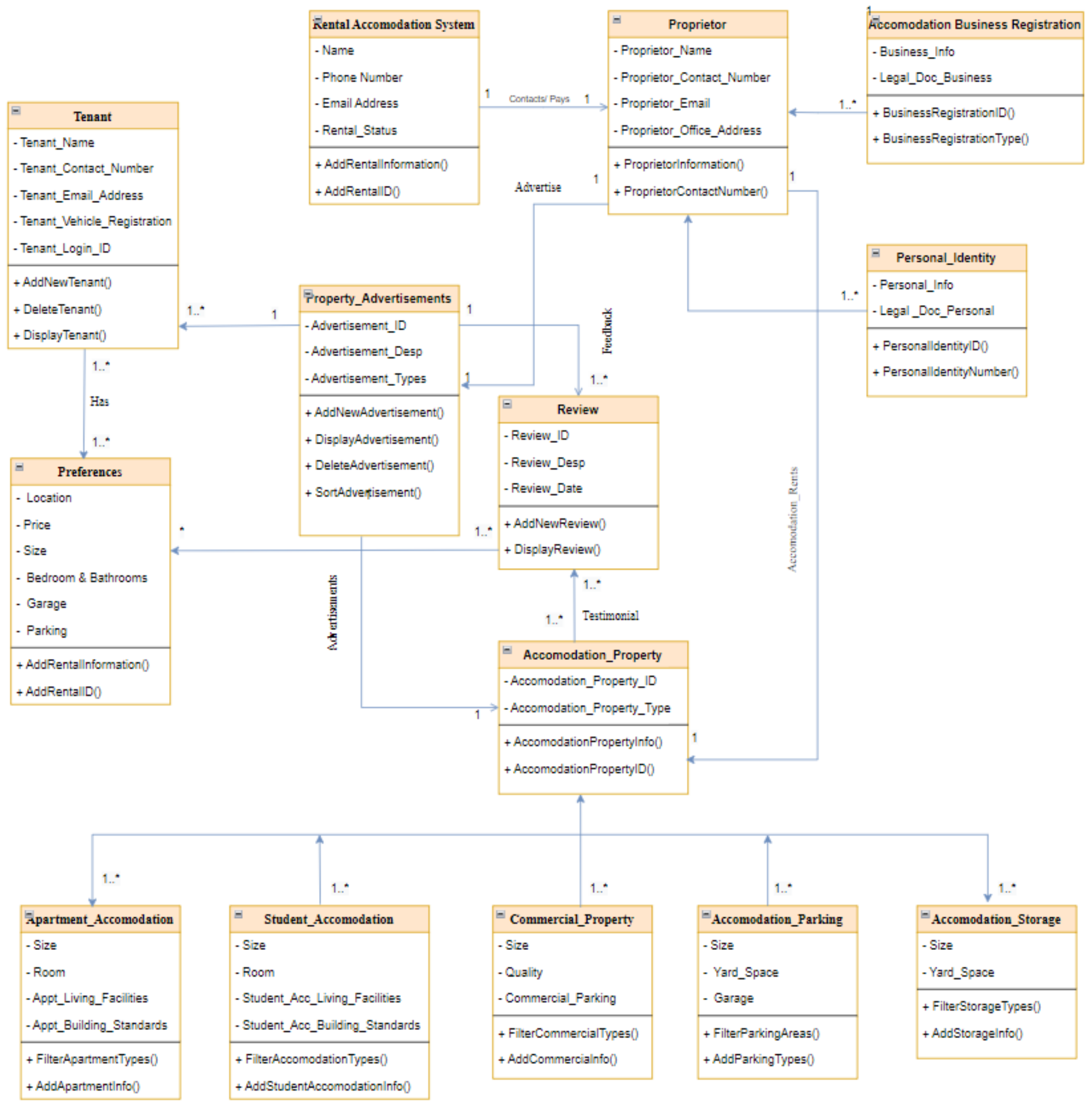


Figure 3: Class Diagram

7. SEQUENCE DIAGRAM

In the world of software engineering, a sequence diagram illustrates object exchanges organized in temporal sequence. The objects involved in the scenario are shown, as well as the series of messages that must be exchanged for the scenario's functionality to be carried out. Sequence diagrams and use case realizations are frequently connected in the system's logical view. Event graphs and event procedures are other names for sequence diagrams.

A sequence diagram shows a chronology that begins at the top and gradually lowers to mark the succession of interactions. The messages passed back and forth between each object's column are represented by arrows. Sequence Diagrams are interaction charts that outline the steps that must be taken during an operation. They capture the interaction between items in the context of a partnership. The vertical axis of the diagram is used to represent time in sequence diagrams, which show the order of the interaction visually by designating when messages are sent.

Sequence Diagrams consist of the:

- Interactions inside a partnership that carries out either an operation or a use case
- High-level discussions between the design and the system's user, the design and alternative approaches, or between subsystems

The sequence diagram's purpose is:

- An example of high-level interactions between active procedure objects
- - Track how different object instances communicate with one another within a partnership that implements a use case.

Either model specific instances of interaction or replicate broad transactions (illustrating all likely paths through the interaction).

Sequence Diagrams at a Glimpse:

These lifeline notations, which are numerous, should be arranged horizontally across the top of a sequence diagram. Lifeline notations shouldn't cross each other. They stand in for the various components or things that interact with one another along the sequence of the system. A lifeline notation with an actor element symbol is used when the particular sequence diagram is owned by a use case. The box attached to the lifeline is the activation bar. It is employed to indicate that an object is active when two objects are interacting. The rectangle's length specifies how long the objects will remain active.

In a sequence diagram, an interaction between two items happens when one object delivers a message to another. The use of the activation bar on the lifelines of the Message Caller (the object that sends the message) and the Message Receiver (the object that receives the message) indicates that both are active/ are instantiated during the exchange of the message. In a flowchart, a message is specified by an arrow going from the message caller to the message receiver. Any direction—from left to right, right o left, or even back to the message caller—can be followed by a message. While you can describe the message being moved from one object to another on the arrow, you can also specify the type of message being sent or received by using different arrowheads.

Sequence Diagrams are arranged according to object (horizontally) and time to depict elements as they interact across time (vertically):

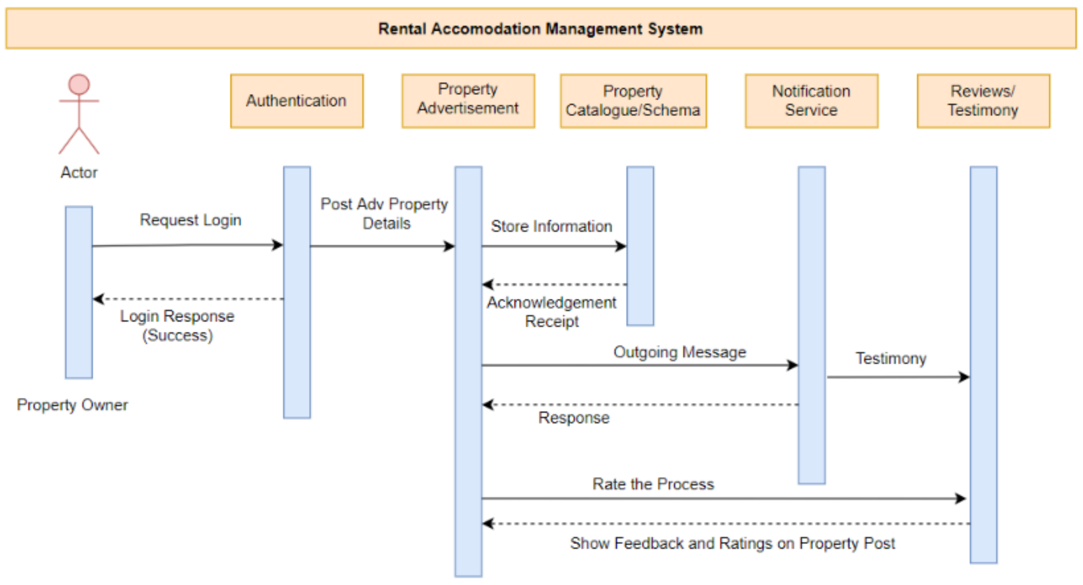
Object Size

- The components that are involved in the interaction are shown on the horizontal axis.
- Depending on when they appear in the new series, the parties taking part in the process are normally listed from left to right. However, the sequence of the information on the horizontal axis is not fixed.

Period Size

- Time progresses (or advances) down the courier as indicated by the vertical axis.

An illustration of the Rental Accommodation Management System's flow is provided below. The sequence diagram's object initiator is described in two steps:



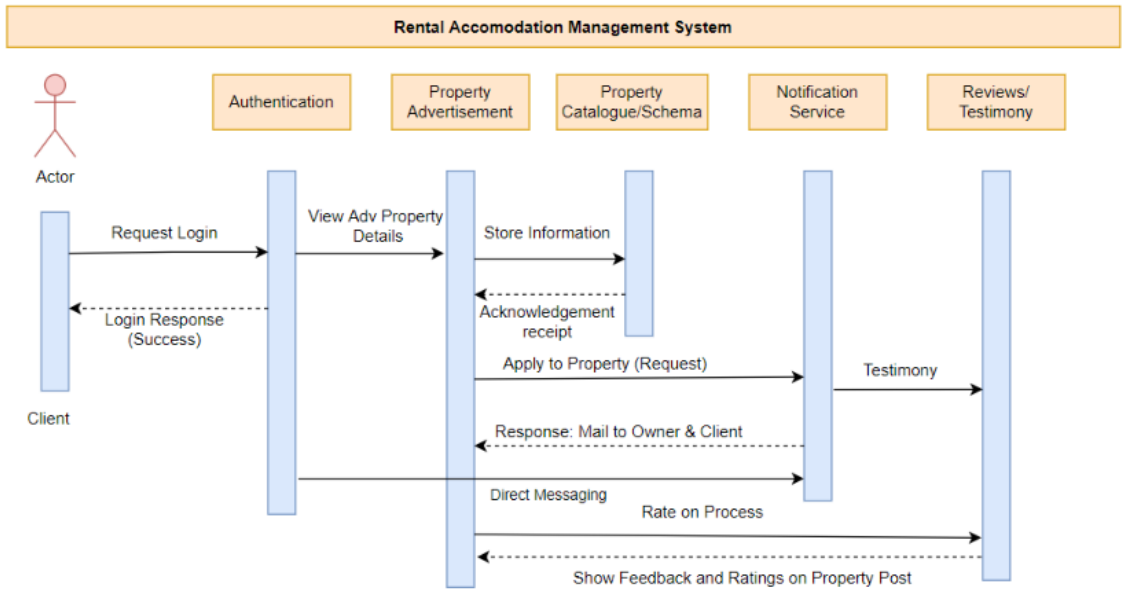


Figure 4: Context Diagram

Entities that interact with the system or are external to it are represented by an actor symbol. A dashed line with a lined arrowhead is the symbol for an asynchronous return message. When the message caller wants to send further messages to other components of the system before waiting for the message recipient to process and respond to the news, they employ an asynchronous message.

A dashed line with a lined arrowhead is the reply message symbol, which represents messages that are answers to calls. Sending a return message indicates that a message receiver has finished processing it and is returning it to the message caller. Return message compositions are optional because return messages are always recommended by an activation bar that is activated by an asynchronous message.

The reply message sign denotes messages that reply to calls and is represented by a dashed line with a lined arrowhead. Sending a return message indicates that a message receiver has finished processing it and is returning it to the message caller. Return message compositions are optional because return messages are always recommended by an activation bar that is activated by an asynchronous message.

In this context, stage one is described as the Property Owner logging into the system for the first time to become authenticated and publish adverts. Following that, the posts are kept in the databases and made visible on the website. On the website, an instant messaging feature allows customers and property owners to communicate immediately and discuss the property, pricing, and other details. The system refreshes the procedures on the website for a new consumer to read after the property owner requests evaluations and feedback from the clients. The user logs into the website to examine adverts for various properties in stage two, which is discussed concerning the consumer. If they are interested in a property, they can communicate with the owners via instant message. Customers can rate the building themselves after reading the reviews. The diagrams are thought to define several concepts. The data kinds will be logically related to those types if they use the concepts in the domain.

8. SOFTWARE TESTING

Software testing is a method for figuring out whether the real piece of software meets requirements and is error-free. Thoroughly tested software guarantees dependability, security, and excellent performance, as well as time savings, cost-effectiveness, and customer pleasure.

Software testing in the software development life cycle has the following advantages:

- It saves money because the fix is simple to implement if the issue is discovered early on or during the pre-production stages, which reduces losses. Fixing a flaw in a production environment might result in significant financial and professional losses.
- A tested product is less susceptible to damaging attacks, enhancing system security.
- Customer satisfaction is one of the key goals of software testing, and it may be attained by using a well-tested product.

There are numerous sorts of testing techniques, including user acceptance tests, smoke tests, integration tests, and unit tests. Let's concentrate on website unit testing.

Let's write unit tests for the class Advertisement and Feedback using the above class diagram as a guide.

__Classes:__

```
Class TestProperty_Advertisement {
    Public AddAdvertisement_CreatesNewAdvertisement_ReturnSuccess () {
    # Once the add is formed, the add new add method receives input data from the user and returns the success flag.
    # This test can provide users with the confidence that when the owner's input is requested for the construction of an advertisement, the ad will be made and published on the website as a result.
     assertTrue(AddNewAdvertisement(informationList[])
     }
```

<u>Reason to select the above test case:</u>

In the system, the most important test case is the advertisement test case which acts as the main class or table to communicate between two different users and is a key to understanding the idea behind the design and development.

```
    Public DisplayAdvertisement_FetchesAdAdvertisementInfo (){

    #It returns the information string for the advertisement when given an advertisement id.

    # This test enables the user to confirm that the specific additional information is presented on the screen whenever the user clicks on any advertisement.

    assertNotNull( displayad(advertisement_id))
```

```
        }
        **Public deleteAdvertisement_DeletesAdInformation (){**
        # When given the advertising id, it will look for the advertisement's existence in
        the database and delete the record, returning a success message once the record
        has been successfully deleted.
        # This test will make sure that if the user decides to remove a property's record,
        others won't be able to access it.
        assertArrayEquals(True, deleteAd (advertisement_id))
        }
        Public sortAd_AcceptsAdInfo (){
        # Sort the advertising data according to time.
        # This technique makes sure that the user sees the most recent adverts when
        they click on sorting the results for advertisements.
            assertArrayEqls( expectedArra[], sortAd())
        }
}
Class TestsReviewProperty {
        Public addReview_acceptsReview_returnsSucess(){
        #This test will make sure that when a user attempts to rate or review an
        advertisement, the attempt is saved and a success message is returned, allowing
        the user to be certain that the review was entered into the database.
        assertEquals( True, addNewReview())
        }
        Public displayReview_displaysReviewAd_returnsReviewObjectArray {
        # These methods make sure that when a specific advertising id is submitted to a
        method, a review array related to the advertisement will be returned in the
        response.
        assertArrayEquals (expectedArray, displayReview( ad_id))
        }
        Public displayRatingsAd_returnsRatingObjectArray {
        # These techniques make sure that when a certain advertisement is placed and
        the user is able to provide accurate ratings, those ratings can be displayed on the
        screen, and this approach ensures that the appropriate rating array will be
        returned in the response for the advertisement.
        assertArrayEquals (expectedArray, displayRating( ad_id))
        }
}
```

9. CONCLUSION

With our concept of design and execution of this system, we tried to address the housing accommodation demand while keeping in mind all the software principles. One of the concerns that many people nowadays confront regularly is the difficulty that the client and entrepreneur face. We decide to follow that idea in light of our individual experiences. We believe that technology can be used to make things better since we have dealt with similar problems and had similar experiences. We have made an effort to simplify things for users and the user interaction experience to make it easier for individuals to complete this procedure in the future. All of the in-scope and out-of-scope things are clearly described in this text. Because time and resources would be limited in the beginning, we attempted to keep things simple. However, we want to add more functionality once our minimum viable product has been released. We carefully created the product with the bare minimum requirements of the clients, owners, or the marketing they perform to prevent any future software engineering troubles. This document contains a variety of tools to assist you in avoiding these problems, including user stories, a context model, entity relations, class diagrams, and sequence diagrams. There has been incorporated all the data necessary for the implementation step. As soon as is practical, the remaining stages should be put into action. To demonstrate to our funders and even customers how well our program functions, we adhere to the software design principles before moving forward. Following that, if we can have a sizable enough impact, we can start adding more team members to the project to advance our application. Our earlier experiences have shown us that Ireland is a great place to start our idea because the present approach is challenging. We'll make the most of it and get things done as soon as we can.

10. BIBLIOGRAPHY

Kirti Rathore*1, Aleena Syed*2, Rahul Patel*3, 2021. RENTAL HOUSE MANAGEMENT SYSTEM Volume 03 (e-ISSN: 2582-5208) – IRJMET

Henry Peter Gommans *, George Mwenda Njiru **, Arphaxad Nguka Owange **,2014. Rental House Management System Volume 04 (ISSN 2250-3153) - IJSRP

Yingying Chen ; Taizhi Lv, 2021. Design and Implementation of House Rental System (2021): 35 - 39

Wen-juan, ZHU Yuan-yuan SHAO, and H. U. A. N. G. Zhi-yuan. "Design and Implementation of House Rental Information Search System Based on Scrapy." Computer & Telecommunication 1.6 (2019): 14-1

Software-engi.blogspot.com. 2021. Context Diagram in Software Engineering. [online] Available at: <https://software-engi.blogspot.com/2012/12/context-diagram-in-software-engineering.html> [Accessed 10 December 2021].