Group Project Log

Group Name:	Group 12

Group Members:	Dhruv Oza, Kushang Mistry, Rishabh Domadiya, Sanjuna Konda, Vishvesh Naik

Deliverable:	Housify Project Technical Report & Code

Group Member Name	**Work Done (%)**
Dhruv Oza	20%
Kushang Arunbhai Mistry	20%
Rishabh Domadiya	20%
Sanjuna Konda	20%
Vishvesh Naik	20%
Total:	**100%**

TECHNICAL REPORT

PROJECT GROUP 12

‘Housify’

A one stop rental solution

Members and Contributors

Dhruv Oza
B00856235
dhruv.oza@dal.ca

Kushang Arunbhai Mistry
B00870521
kushang.mistry@dal.ca

Rishabh Domadiya
B00902679
rs582144@dal.ca

Sanjuna Konda
B00894913
sn493898@dal.ca

Vishvesh Naik
B00885077
vishvesh@dal.ca

Faculty of Computer Science
Dalhousie University

11 April, 2022

ABSTRACT

One of the basic human needs is a house. Especially for students and new immigrants, it is a challenging task. We are trying to continuously mitigate this challenge and make it easy for everyone to settle in a new city or even a new country. Getting housing is not a new requirement, so some websites have come up with solutions like Kijiji and Facebook Marketplace, but they are not primarily focused on housing. Our goal is to create an appropriate housing website covering all the aspects of housing/renting that satisfy the users.

KEYWORDS

- Housify
- ReactJS
- NodeJS
- MongoDB
- Rental Application
- House Search
- Integrated Payments
- Extensive Search
- Rental Solution
- Property Advertisement

CONTENTS

1. INTRODUCTION

Halifax is one of the major and fastest growing cities, not only from economic point of view but from population side also. A rise in population directly proportional to rise in demand of housing properties. Considering Halifax as a major study hub, rental houses in Halifax are always in demand and being hot favorite between international students' communities as well as local communities for rental purposes. Given that increasing popularity of the place and continuous inflow of international immigrants (majority of them are international students), Halifax housing crisis is going up in increasing manner. Students are not only facing difficulties finding rental houses but also have lack of preferences, most of the time they need to adjust to the situation without having any preferences. What the actual need is, students want to search for affordable housing, where at one stop they can search and analyses the information about all rental properties at one stop, moreover they can do extensive search for their preferences, for example they can filter the property based on options such as other housemates are weather vegetarian or not (this is example of one of the preferences). The same way, if we consider property owners, they advertise the property to platforms such as Facebook Marketplace, KIJIJI, but these are decentralized platforms where owners do not have choice of any specific platform dedicated to housing management only (paying/accepting rents, request/provide services). Moreover, the two listed platforms are only for advertisement purpose, they do not have additional functionality which are the need of the system [1].

As identifies reason is, there is no such central facility where both tenant as well as property owners can communicate with each other easily, owner can list the properties, tenant can view the properties, both can schedule a meeting and in future both can be in contact about the property, rent and maintenance purposes. The key purpose of 'Housify' is to bring property owners and (future) tenants such as students, to one place and making smooth the whole renting process, with additional functionalities and extensive search [1].

Considering all aspects of needs and the development, 'Housify' aimed to solve problems of landlords/property owners as well as a community of tenants who are in search of the property for rental purposes. With 'Housify', it is aimed to minimize the gap between two stakeholders which are landlords/property owners and tenants respectively, moreover, it will be a one stop solution and efficient platform where tenants can perform extensive search and find properties based on their preferences. Unlike other available solutions, 'Housify' will also include solution for all the housing/rental property management needs such as, rent management, regular payment of the property, management of rental property maintenance services as well as contract management [1].

Features Completed:

1. **Profile Management:**
   Profile management feature primarily consists of some of the basic functionalities and tasks such as user registration to the application or platform as well as a login page which will open doors to the application and user can access the platform without any limitation. Not only that user management also contains user profile

management, which inherently contains user preferences such as preferred location as well as saved items / properties in which user is interested [1].

2. **Payment Module:**
   Payment module is the key element when user wants to pay their deposit via online payment gateway. In addition to that, when user wants to pay their monthly rent, i.e., a rent on subscription basis there the payment module comes into picture, with proper validations and checking about the inputs by user, this feature can be implemented, however, third-party payment gateway can be challenging part, instead we can implement a dummy feature which can depict the same process [1].

3. **Search for houses using filter option:**
   Once after successful registration and login, user will be directed to the home page. In this home page, there will be an option for the user to search for a house depending upon several factors like filtering houses or flats based upon the location, cost, type of house (1-bedroom, 2-bedroom etc.) and number of persons willing to live in that house or flat [1].

4. **Application Dashboard:**
   The user will be able to apply to rent a house or a room as needed, as well as see the progress of their application. Aside from that, the user will be able to see all of the applications for which a request has been made [1].

5. **Discussion Forum:**
   The users will be able to create, view, and comment on threads. This feature is like a forum where users interact with each other on the same topic. They can create multiple topics to share and gain knowledge. They can also view various threads to know more about the subject [1].

6. **Manage Houses:**
   With this feature, the user has the ability to list out the house for rent out the house for other users. The user will create new post/ad for renting out the house which would be viewed by all other tenants. Furthermore, the house listed by the renter can be updated for editing the details of the house with the help of this feature. The renter is also provided with the option of deleting the post, if the house is rented or in any other situation. All-inclusive, this feature is more inclined towards the renters [1].

7. **Housing Analysis:**

   This feature enables the tenants to provide review for the house they are currently living in or have previously lived in. The tenants can add the rating and feedback for the houses, and also, they can edit the same. The analysis section provides the statistics about the house to the owner/renter which includes visual representation such as number of reviews of the house, number of tenants, and many more [1].

1.1 Live Project URL

Live demonstration of the application: https://housify-group12.herokuapp.com/

GitLab Front-End project repository: https://git.cs.dal.ca/oza/csci-5709-group-12

GitLab Back-End project repository: https://git.cs.dal.ca/kmistry/csci-5709-group-12-backend

2. BACKGROUND

2.1 Competitive Landscape

'Housify', the one stop rental solution aiming to provide a platform to mitigate communication barrier between tenant and landlord/property owners. Below are listed competitors of the 'Housify' and differentiators that makes platform unique. Existing competitors of the platform are, Facebook Marketplace, KIJIJI, CapRent and RentSeeker [1].

Extensive Search
'Housify' exclusively offers extensive search options and wide range of filters to search for properties in designated area, which tenant can use it to reach to the exact search/properties. Existing applications are having search filters but are generalized and not extensive for people such as international students or other tenants who are looking to rent property based on extra filters such as property amenities, preferred room-partner (gender, food preferences) [1].

Integrated Payments
This is one of the key differentiators which differentiates 'Housify' from existing competitors. Handling payments from and withing the platform is the convenience we are providing users; users will not only pay monthly rents through platform but also users can access payment history and can keep track of the payments [1].

In-app Maintenance Service Management
The question to consider after renting the house is, what after renting approval? One thing is timely payments from tenant to landlords/property owners, but when there might be some problem in the property or provided amenities, the tenant informs to the respective property owners about inconvenience they are facing, and both can connect and work together towards the solution. In-short, 'Housify' provides maintenance service management and tracking, a after renting service to the tenants [1].

Visual Analytics of Property Trends
Existing platforms provides, price in plain format, we are representing the data such as property price data in a visually appealing manner so that tenant can analyse and can conclude their decision about the property. For example, they can compare the price of the property and can view rental property price trends [1].

2.2 Problem and Approach

Halifax had been named as the fastest growing city in terms of population according to a Canadian resource. The increase in population number is proportionate to increasing demands of livelihood and properties to reside. As observed last year, a housing crisis occurred in the city as the residents could not find a property to live in. and although, if there would have been, there are many chances such that people would have not been able to reach them. As Halifax is also considered as popular study venue, the student number has also increased to a greater extent including the international students. The immigrants i.e., mainly the students not only face difficulties finding property to share with other students but also lack preferences. The problem faced by the students, international as well as the localites include**:**

- Facing difficulties finding rental houses [4].
- Lack of preferences [4].
- Search for affordable housing, performing extensive search with preferences [4].
- Decentralized platforms for posting house rental [4].

The approach that we followed while designing our application started with researching about the problem of housing crisis. The team went through the existing competitor's applications to bring out the what our application needed. Our application aimed at resolving the problems faced by the tenants as well property owners. A proper interaction between the two would resolve the issue to large proportion. Users can search according to their own needs and preferences with 'Housify'. The tenants can also indulge in a forum discussing about the any topic, and they can add reviews to the houses they have previously lived in. By adding the reviews feature, it makes the selection for the tenants easier.

3. APPLICATION DETAILS

3.1 Target User Insights

Tenant (who are searching for the rental property): This is the very first focused userbase from among both, who will be using the app to fulfill the purpose of searching the rental properties, viewing information/details about the property and then schedule the meetings with landlord for further discussions. This userbase usually wants to perform detailed search, wants to manage their payments to the landlord in addition to managing the maintenance of the property. The trust factor of the tenants will be based on availability of the properties in designated area as well as advertised property on the platform [1].

Landlords/property owners (who have property to rent): As mentioned above, tenants and landlords are two key and targeted users, and both are interconnected with each other. If a user who is landlord having properties and wants to rent the property, with using 'Housify' landlord can advertise their property on the platform with all detailed specifications of the property, including images. This way, advertisement from the landlord can be viewed by tenants and they can interact further. Also, not only connections but after renting the property, landlords can receive the payment from tenants and vice versa, tenants can pay rent with the help of feature on app 'Housify' [1].

3.2 User-Centered Design Approach

3.2.1 Information Architecture

3.2.1.1 Proposed Sitemap

The sitemap given below is expected pages or the key pages of our application, this can be modified based on future requirements and change it as per the need [1].

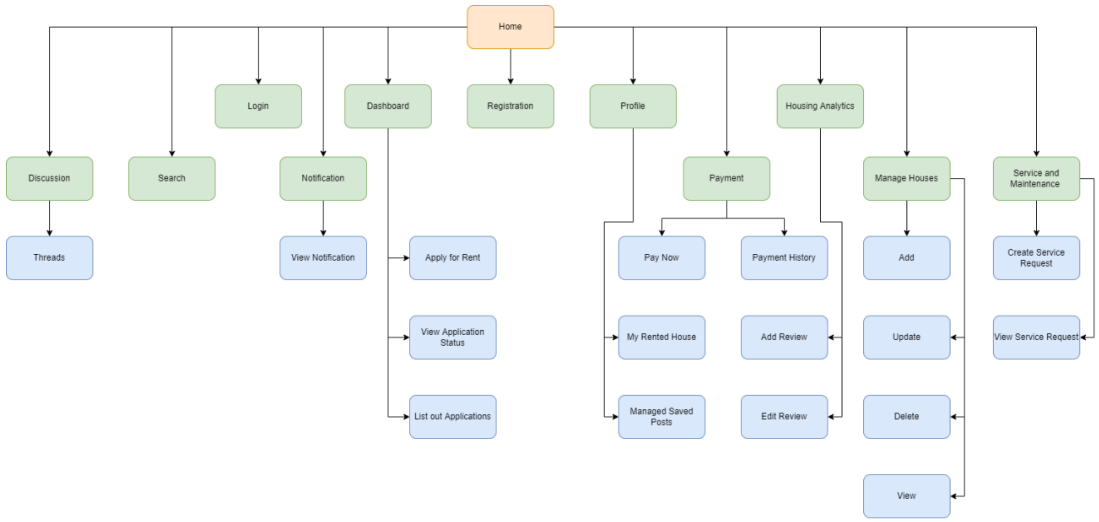

Figure 1: Sitemap for Housify [1]

3.2.1.2 Proposed Wireframes

Feature 1: Profile Management

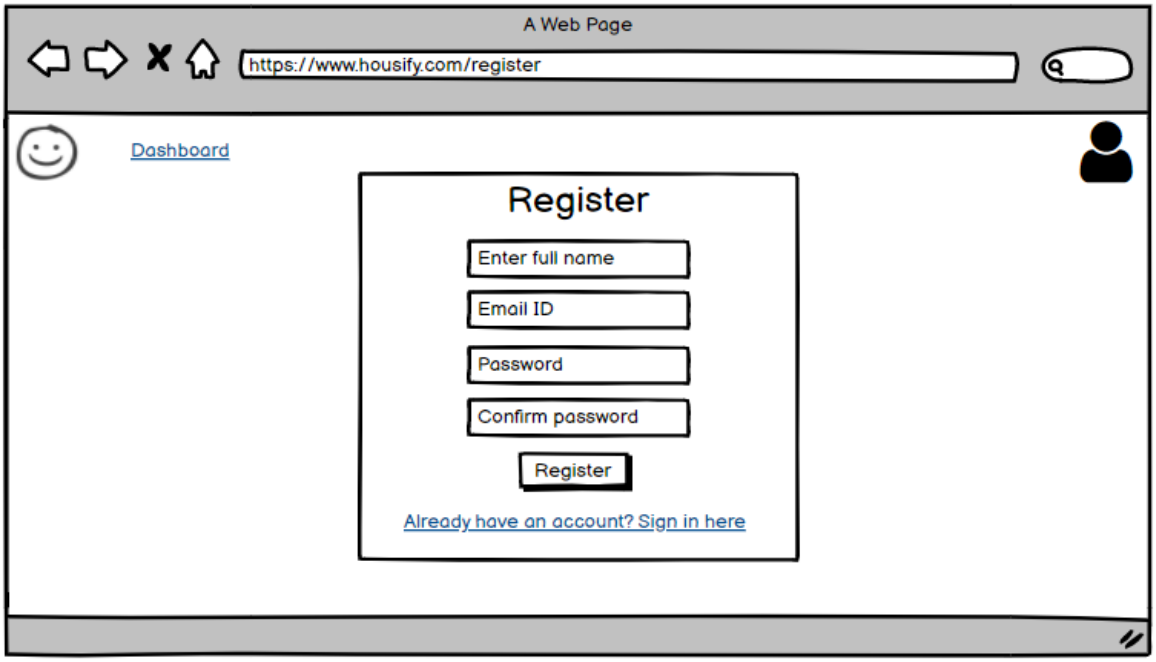

Figure 2 : Wireframe – Registration page [1]

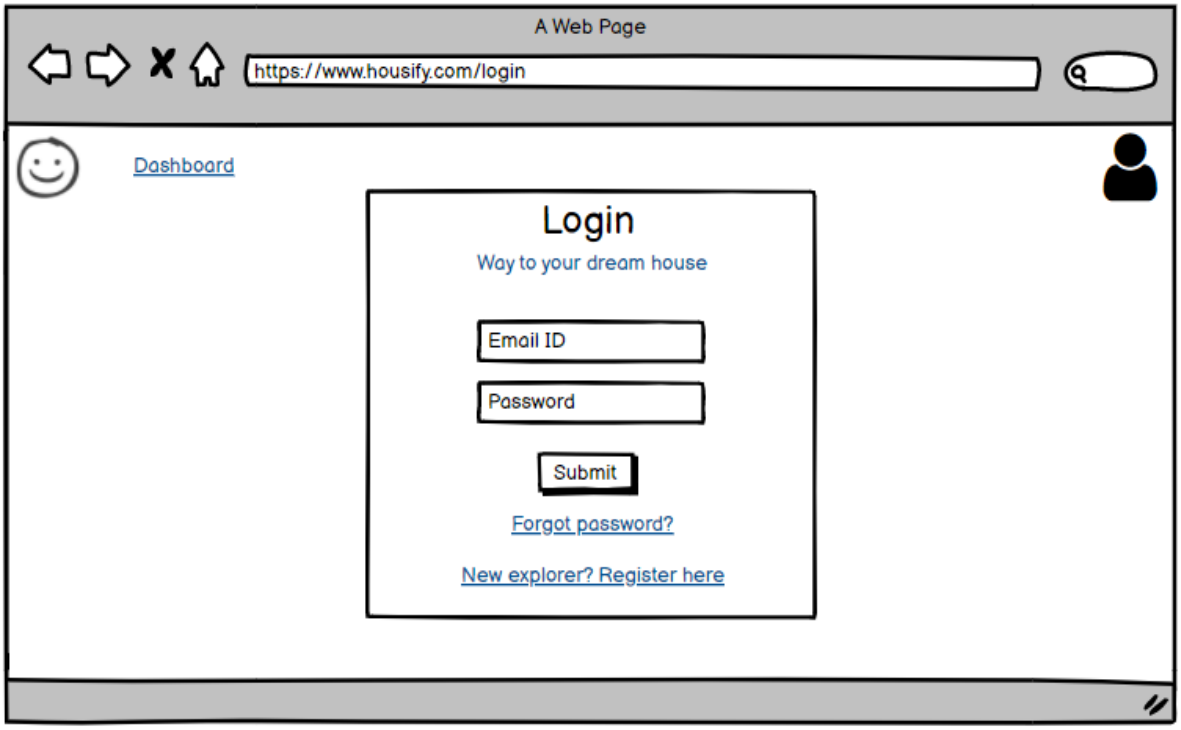

Figure 3 : Wireframe – Login page [1]

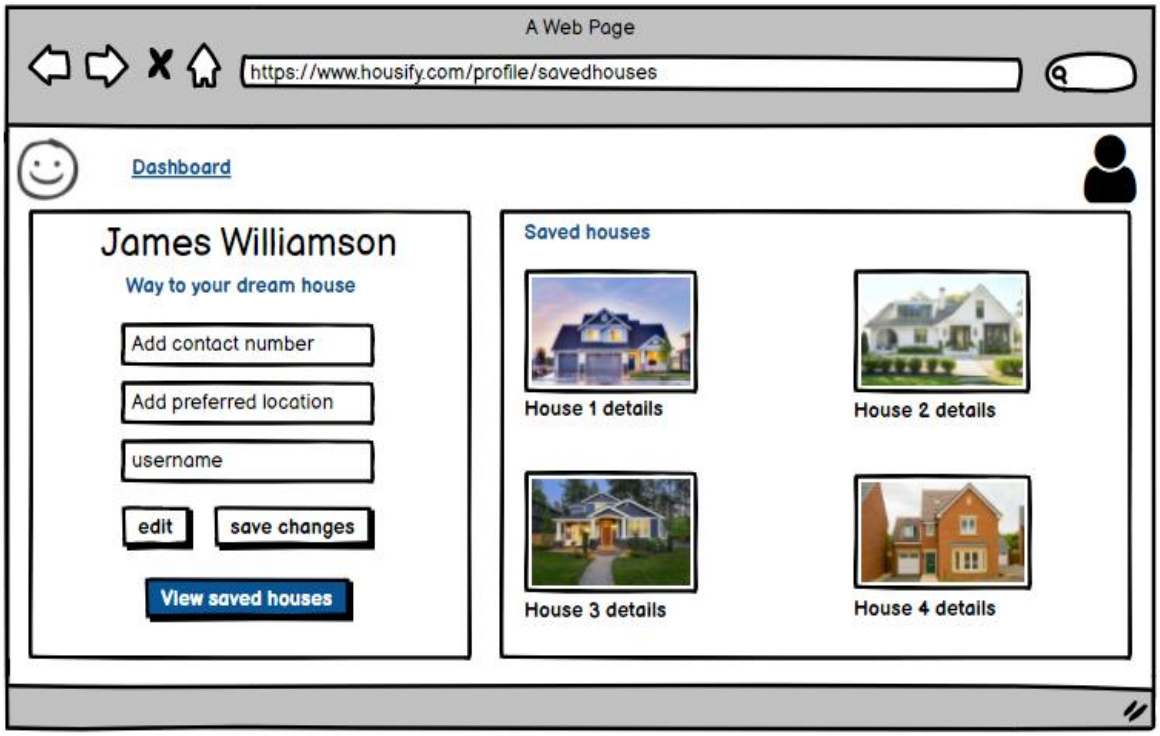

Figure 4 : Wireframe – Profile page [1]

Feature 2: Payment Module

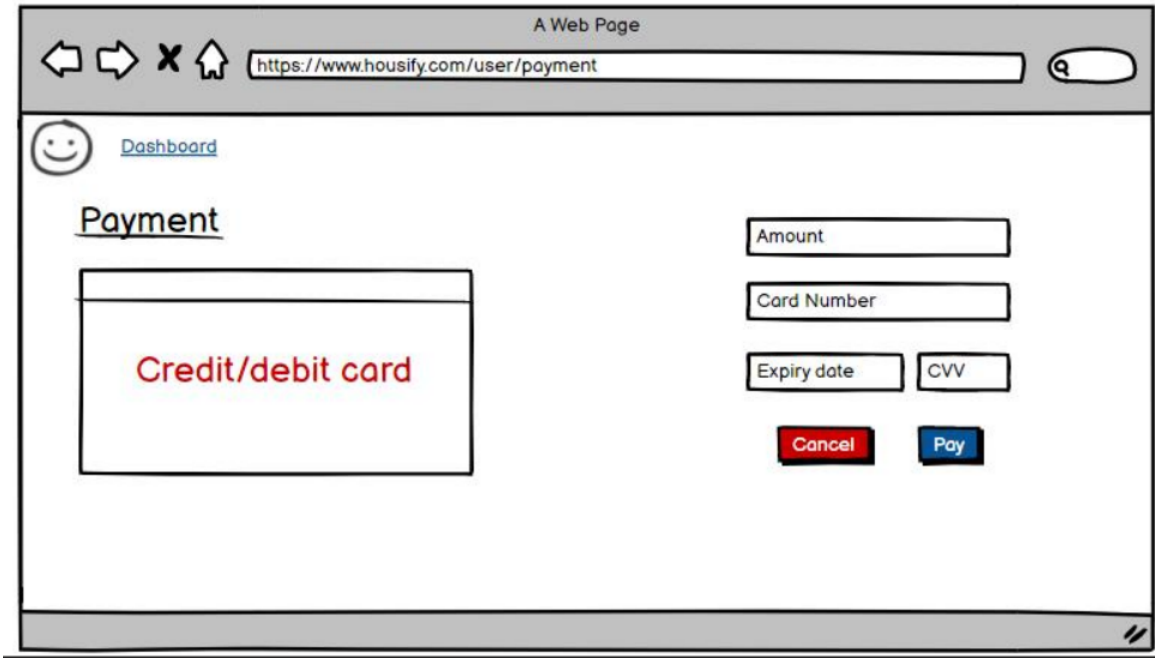

Figure 5: Wireframe – Payment Page [1]

Feature 3: Search for the house using filter option

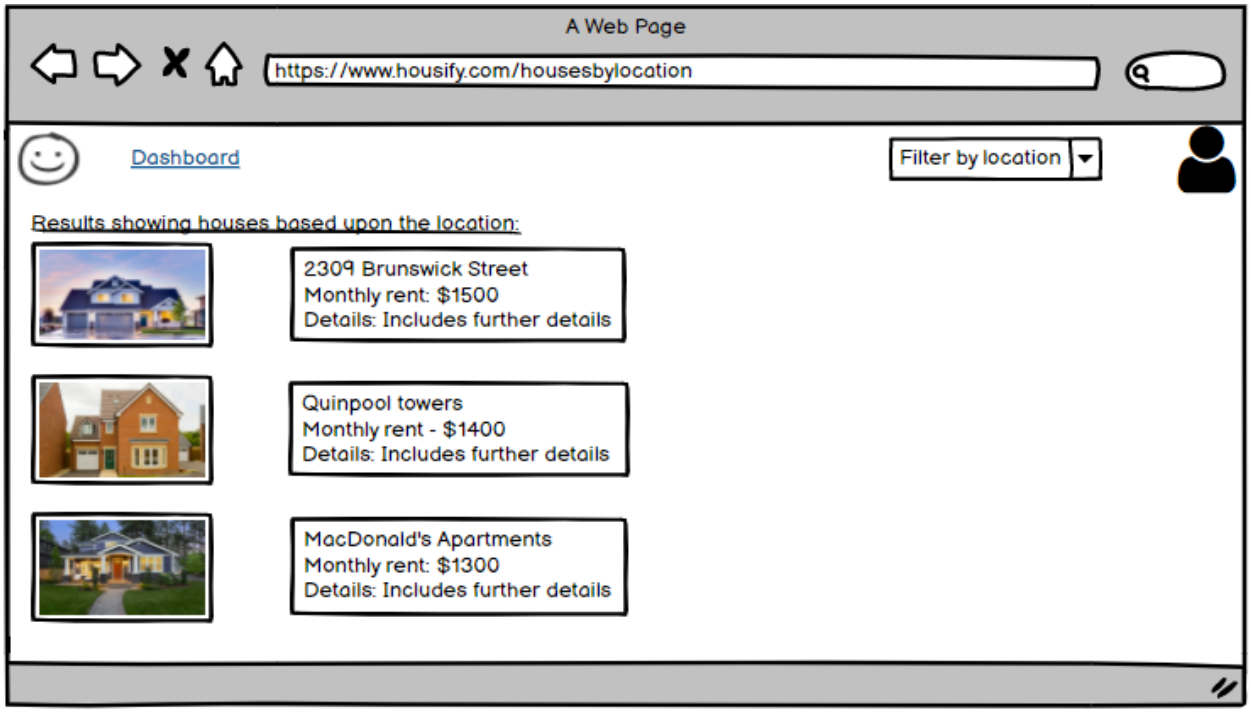

Figure 6: Wireframe – Search for houses screen, based on location [1]

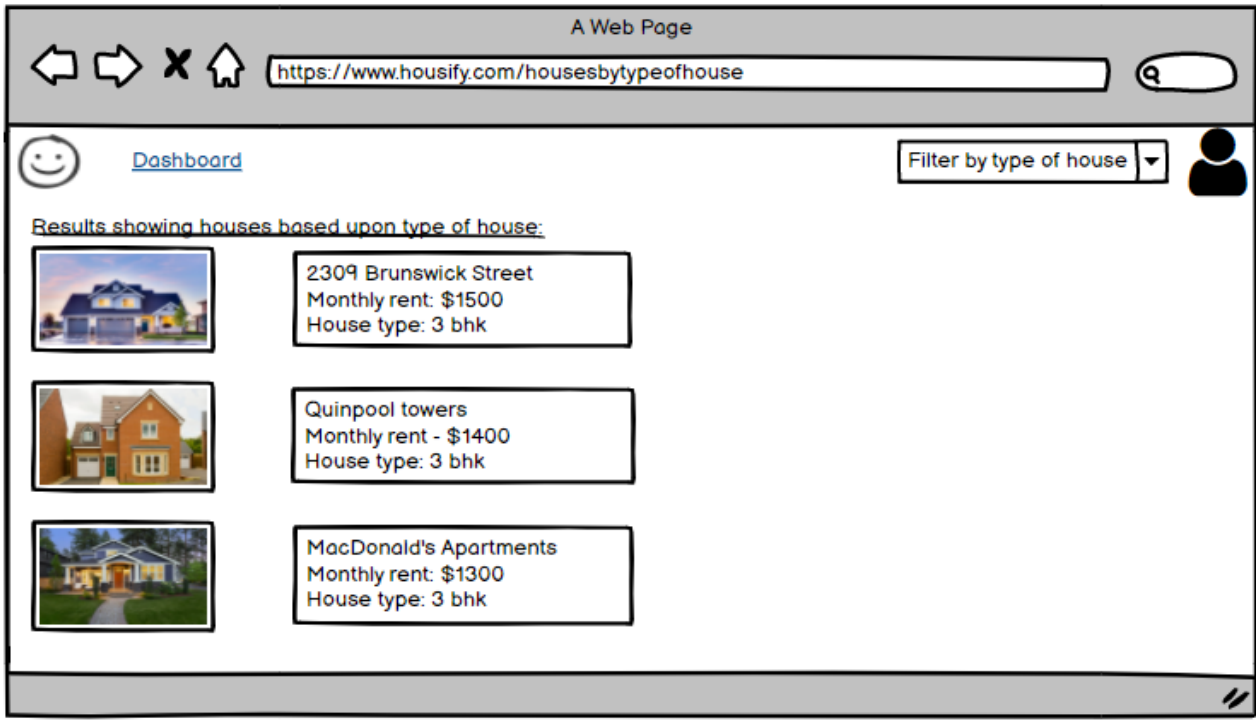

Figure 7: Wireframe – Search for houses screen, based on type of house [1]

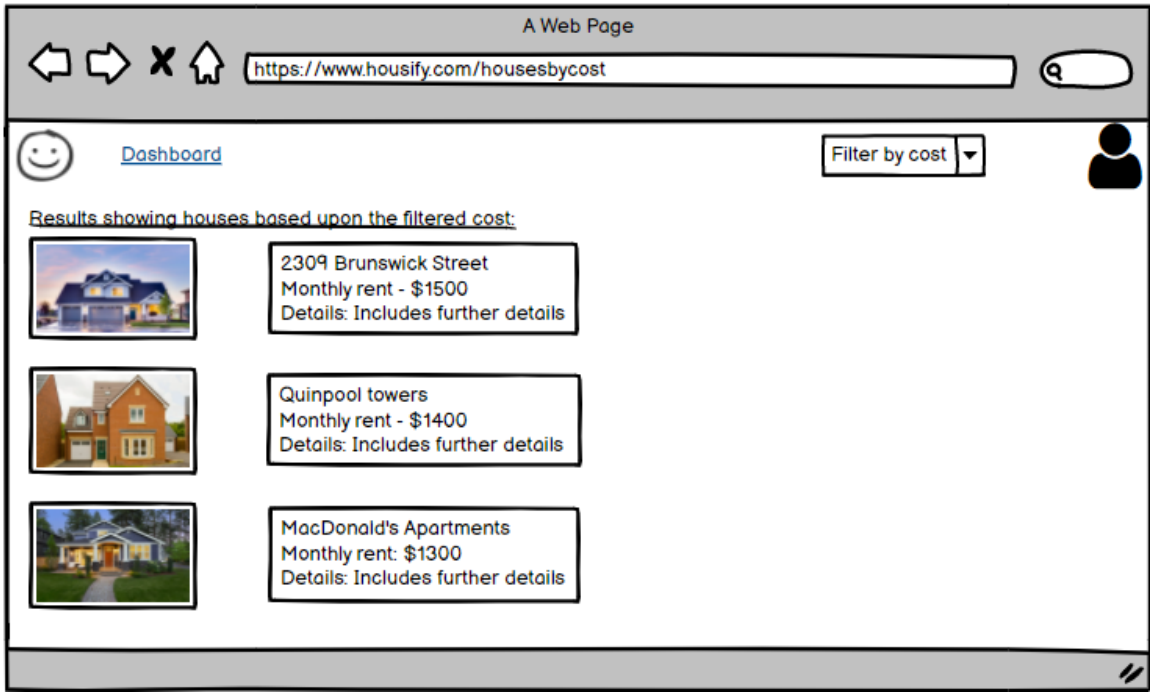

Figure 8: Wireframe – Search for houses screen, house type by cost [1]

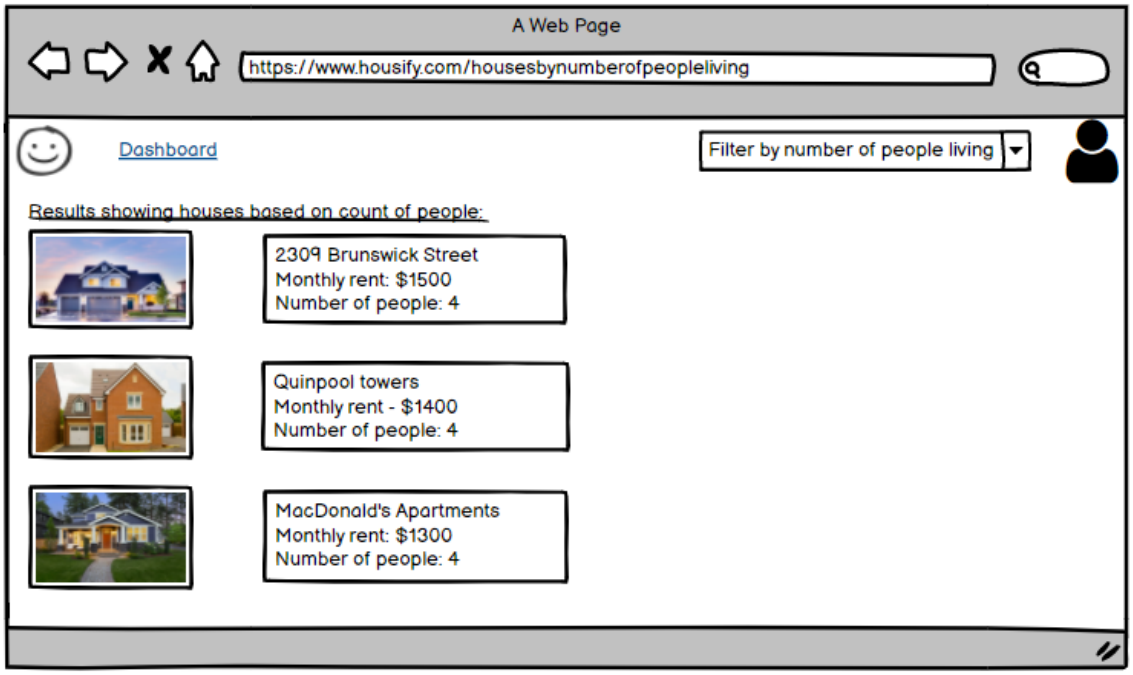

Figure 9: Wireframe – Search for houses screen, based on number of people living [1]

Feature 4: Application Dashboard

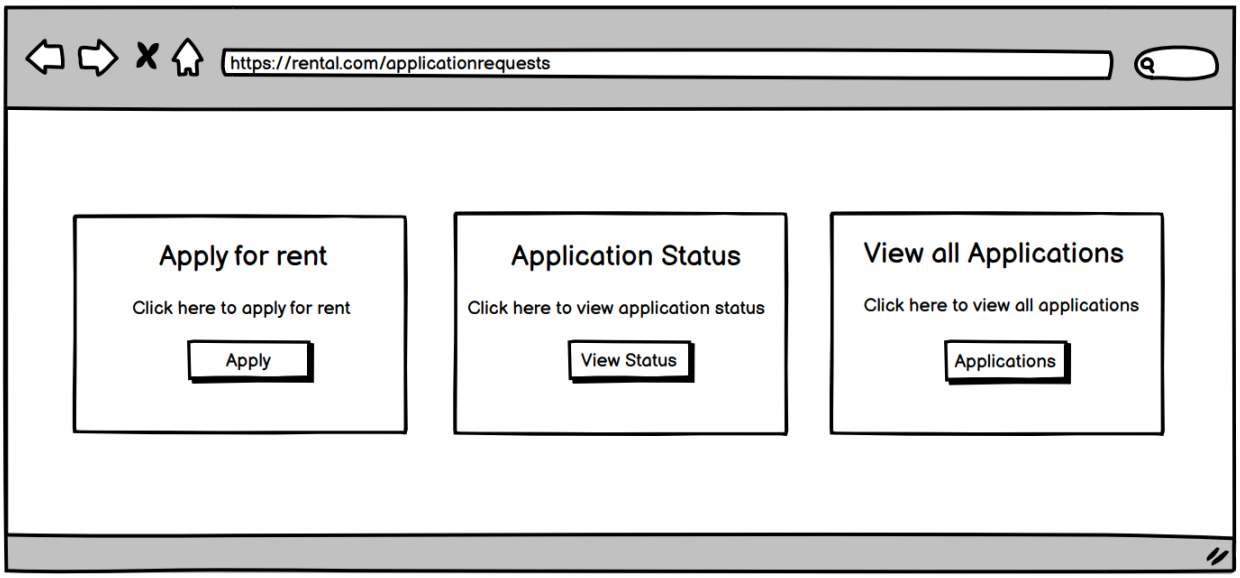

Figure 10 : Wireframe – Application Dashboard [1]

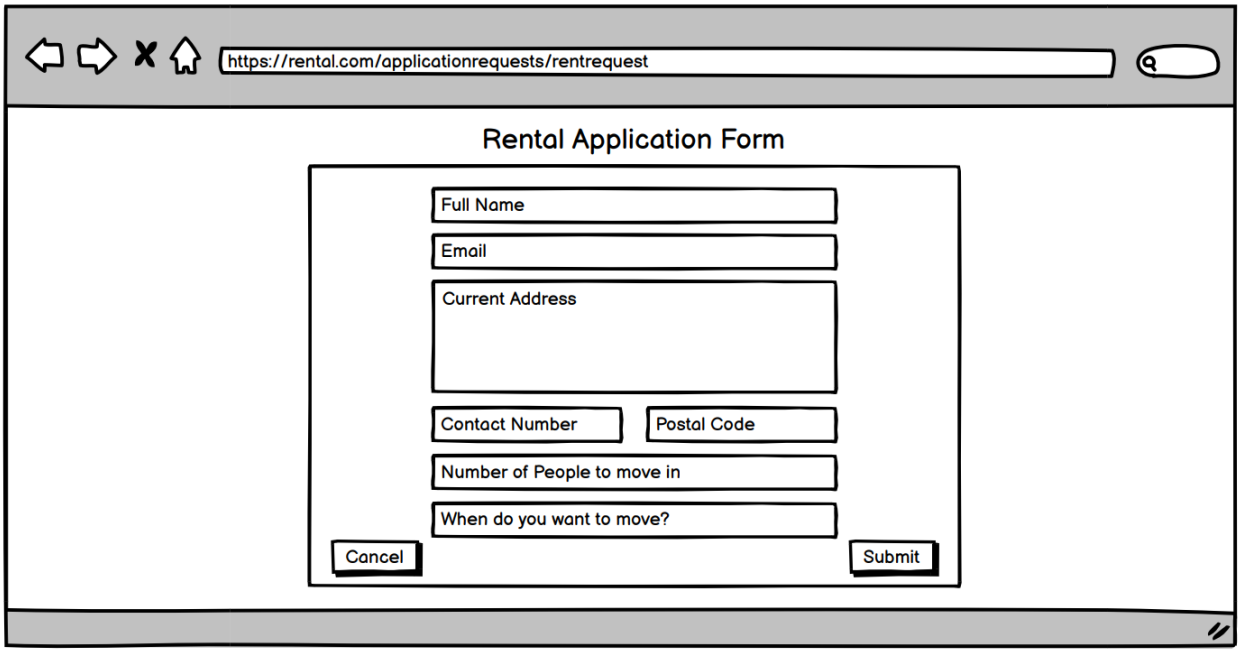

Figure 11 : Wireframe – Rental Form [1]

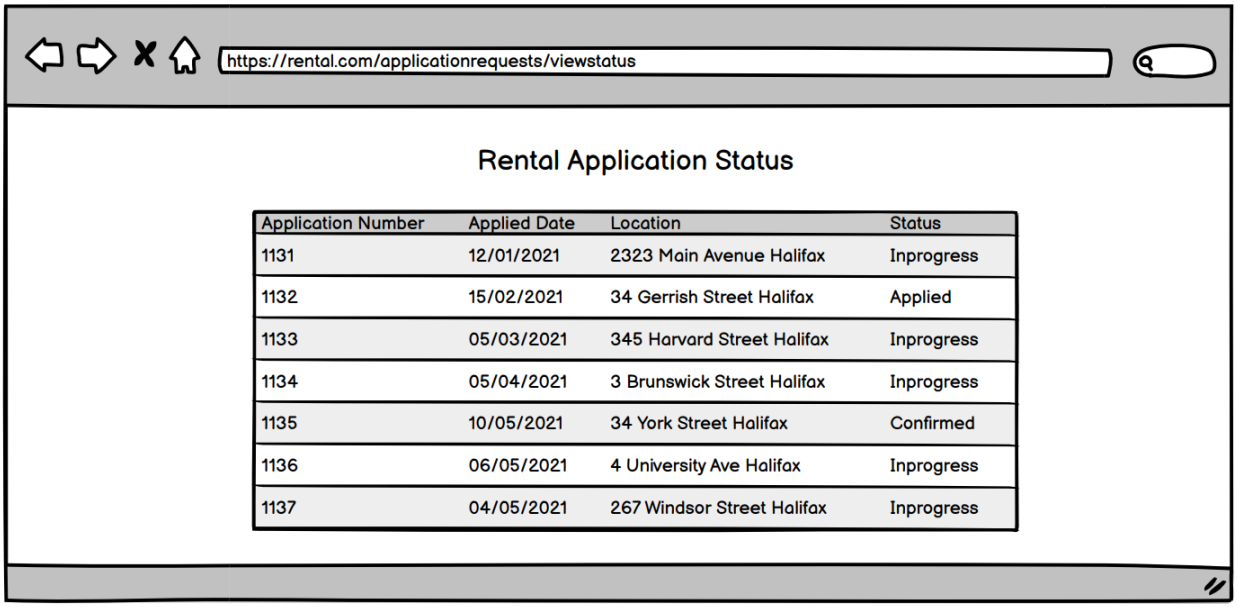

Figure 12 : Wireframe – Application Status [1]

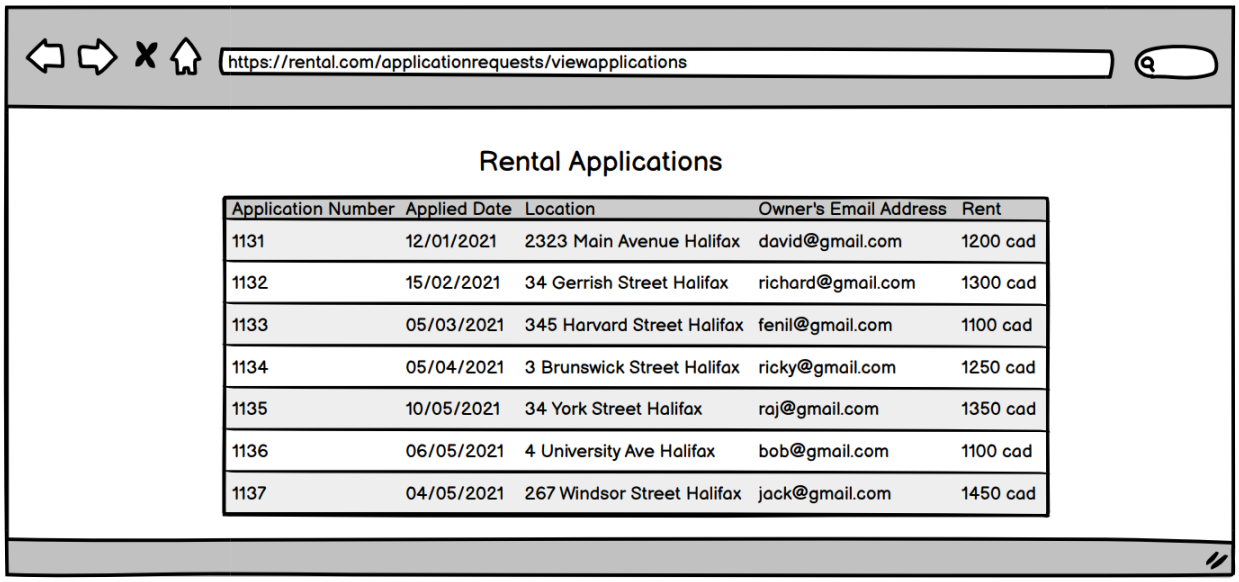

Figure 13: Wireframe – List of all applications [1]

Feature 5: Discussion Forum

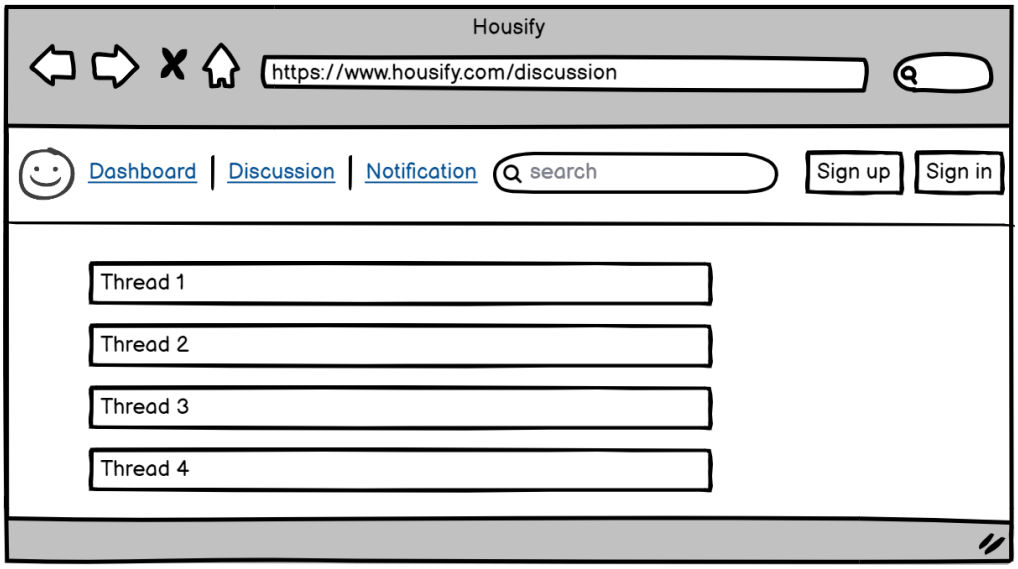

Figure 14 : Wireframe – Discussion forum [1]

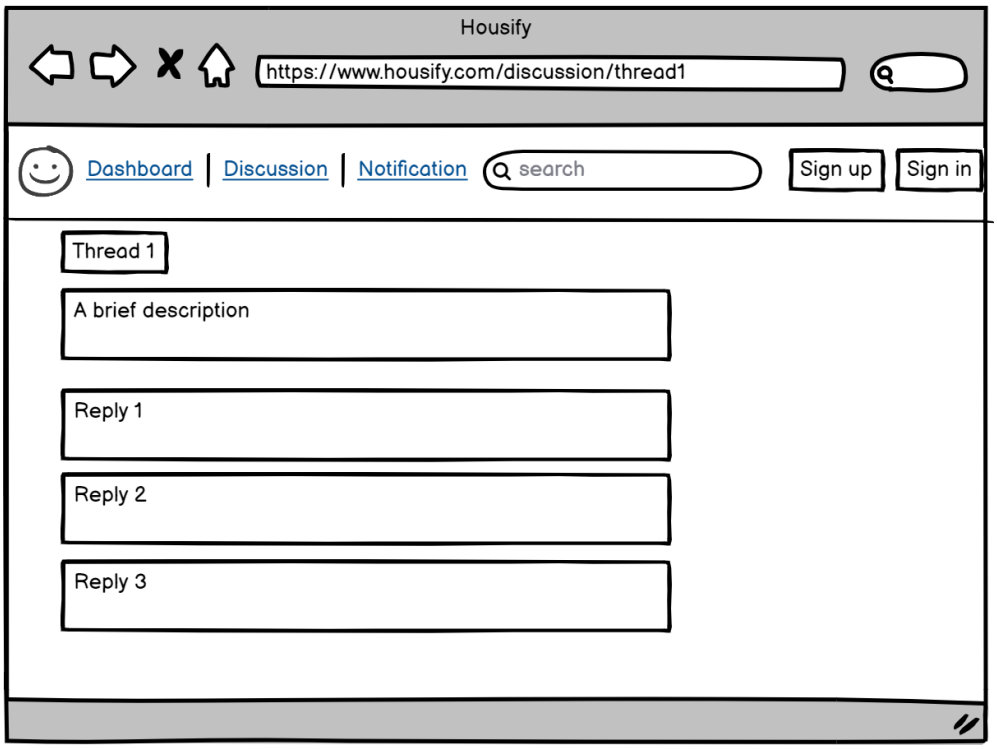

Figure 15: Wireframe – Discussion thread [1]

Feature 6: Manage Rentals/Houses

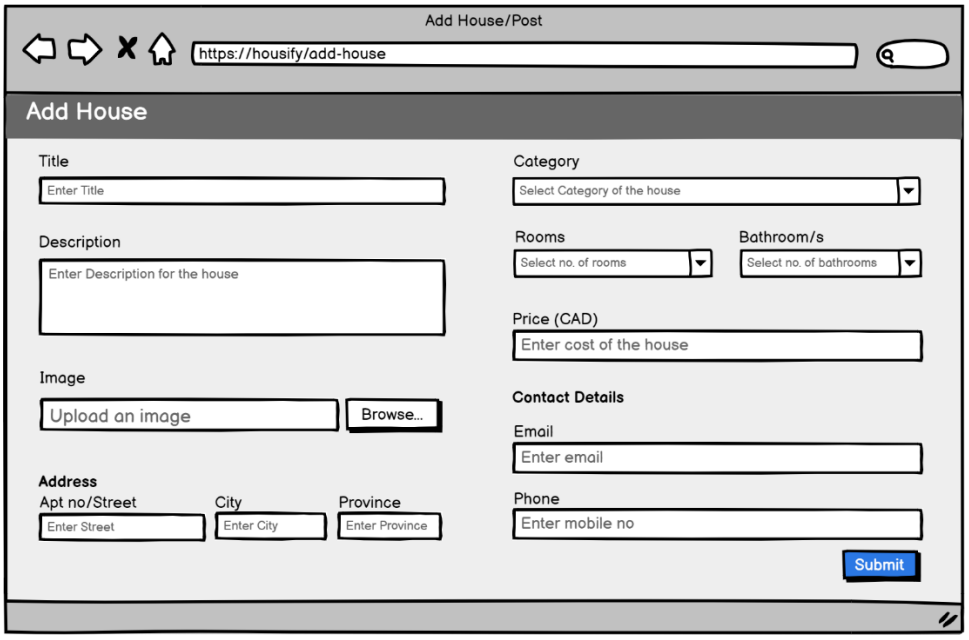

Figure 16: Add post/ad for house – Wireframe [1]

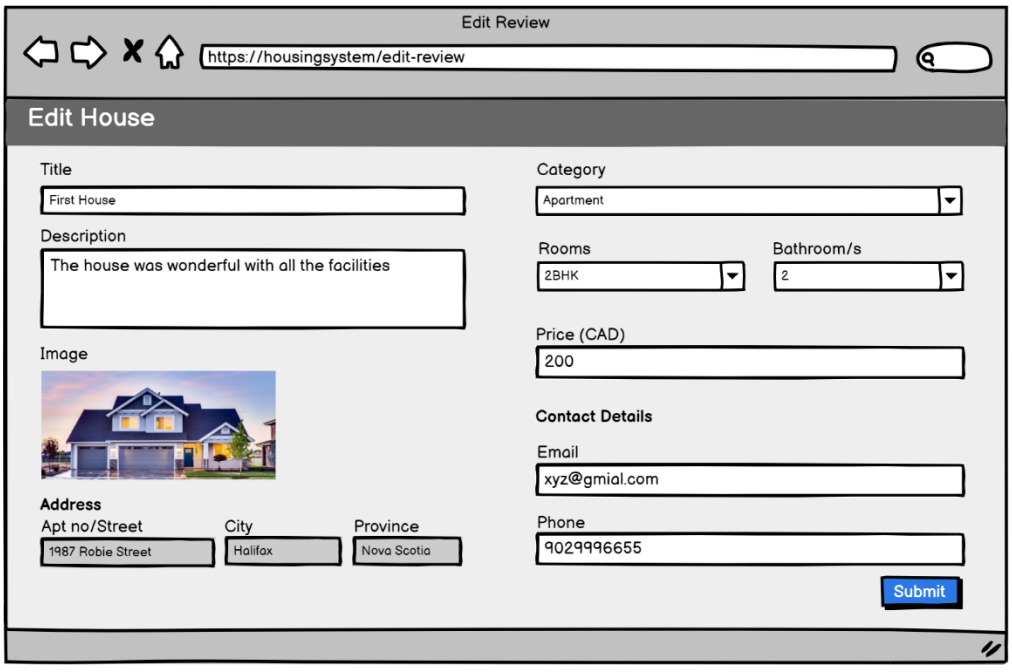

Figure 17: Edit post/ad for house – Wireframe [1]

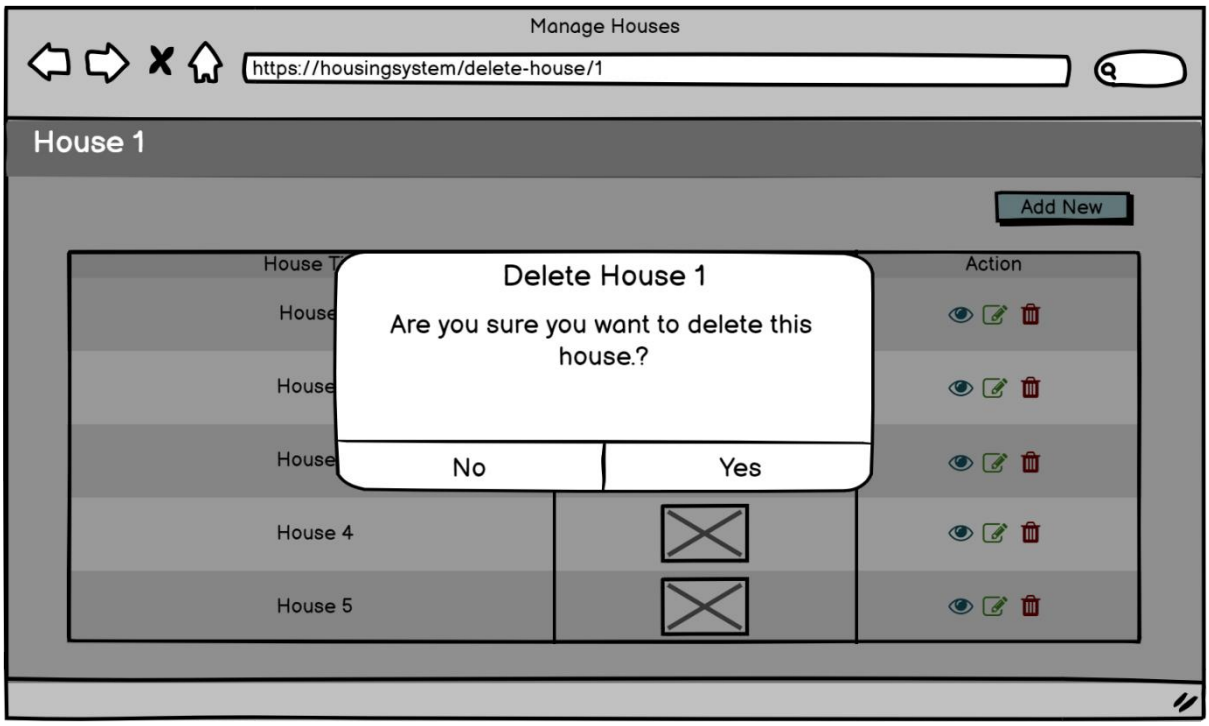

Figure 18: Delete post/ad for house – Wireframe [1]

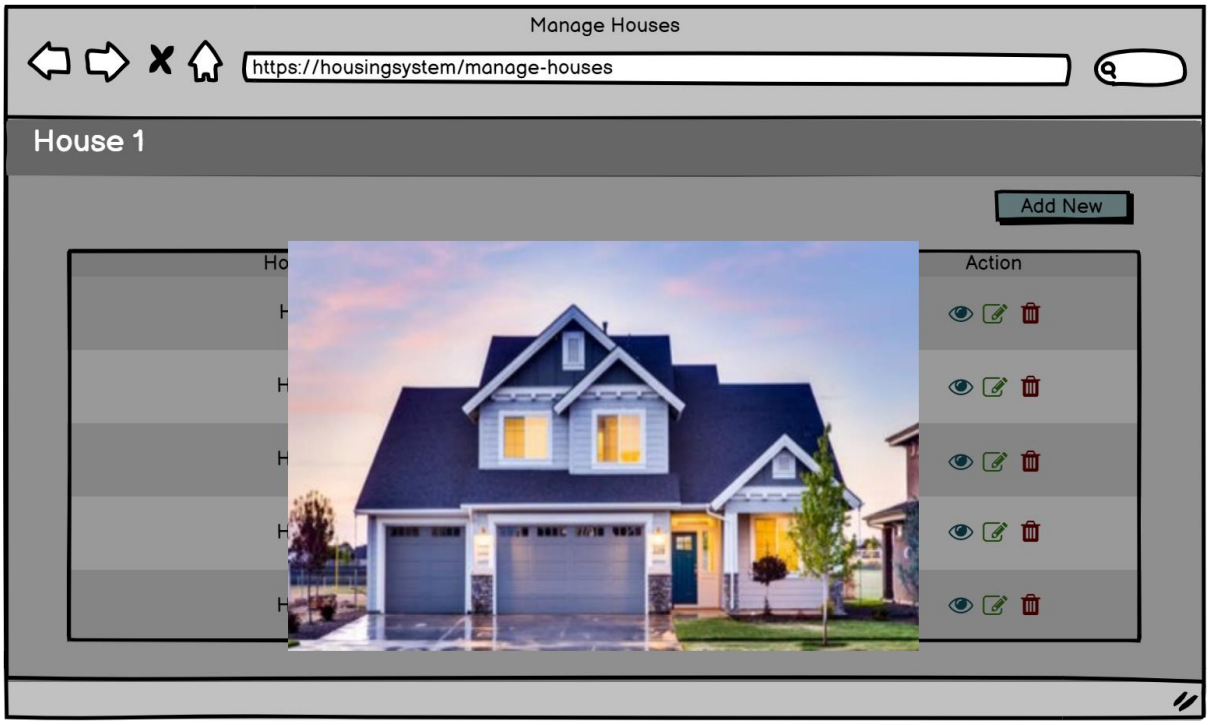

Figure 19: View post/ad for house – Wireframe [1]

Feature 7: Housing analysis

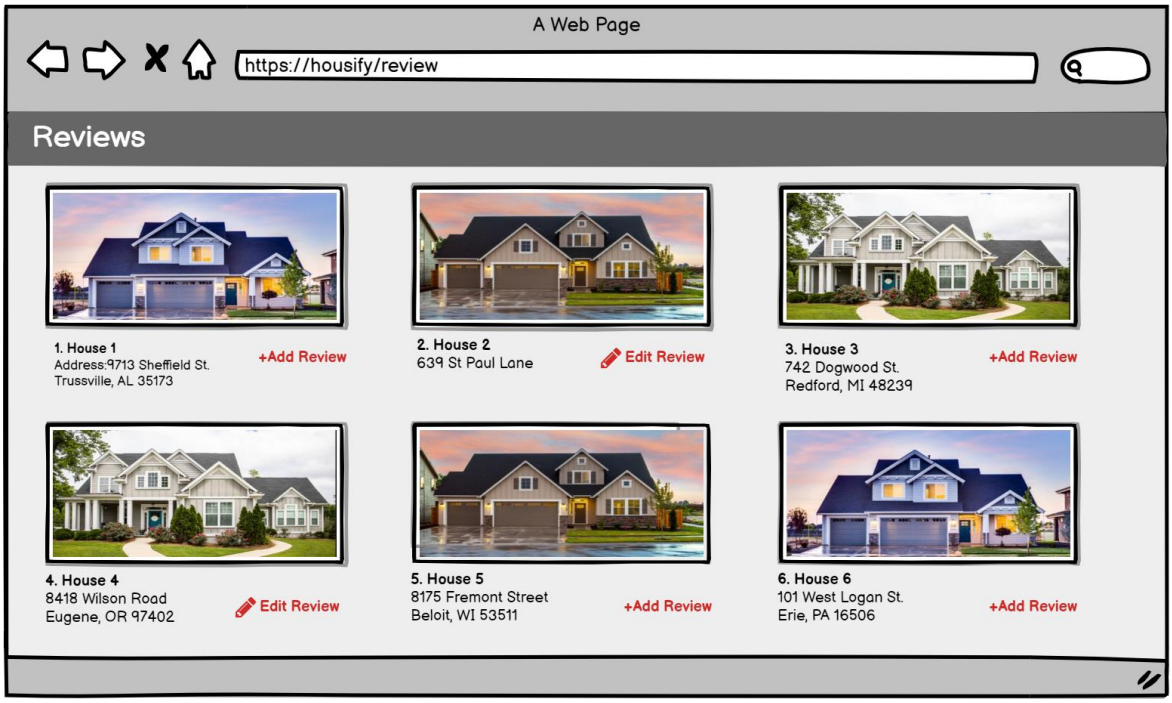

Figure 20: Wireframe – Housing analysis review [1]

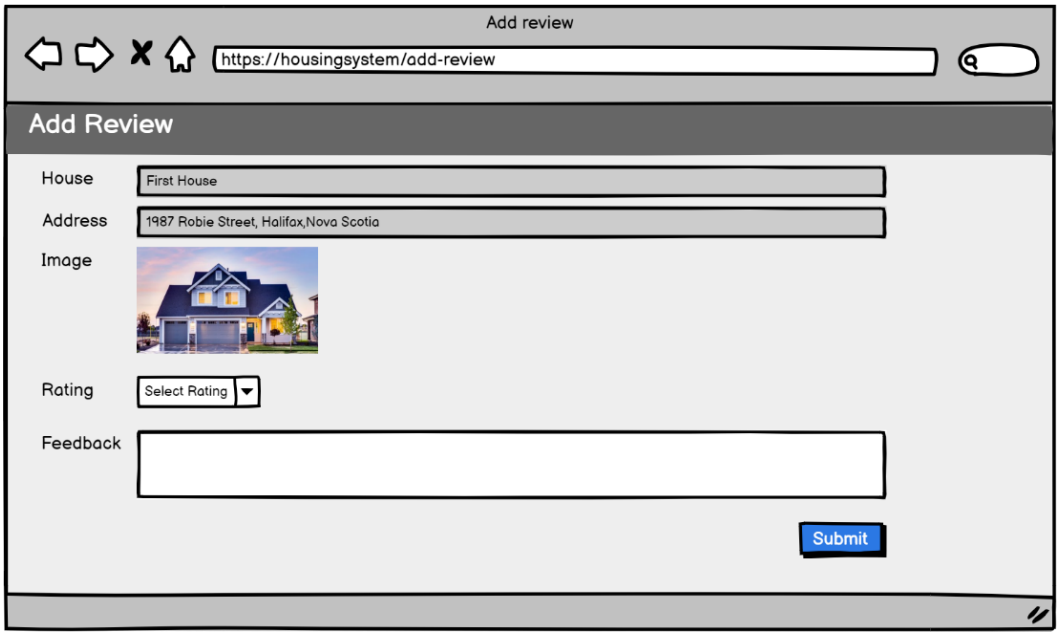

Figure 21: Wireframe – Adding review of property [1]

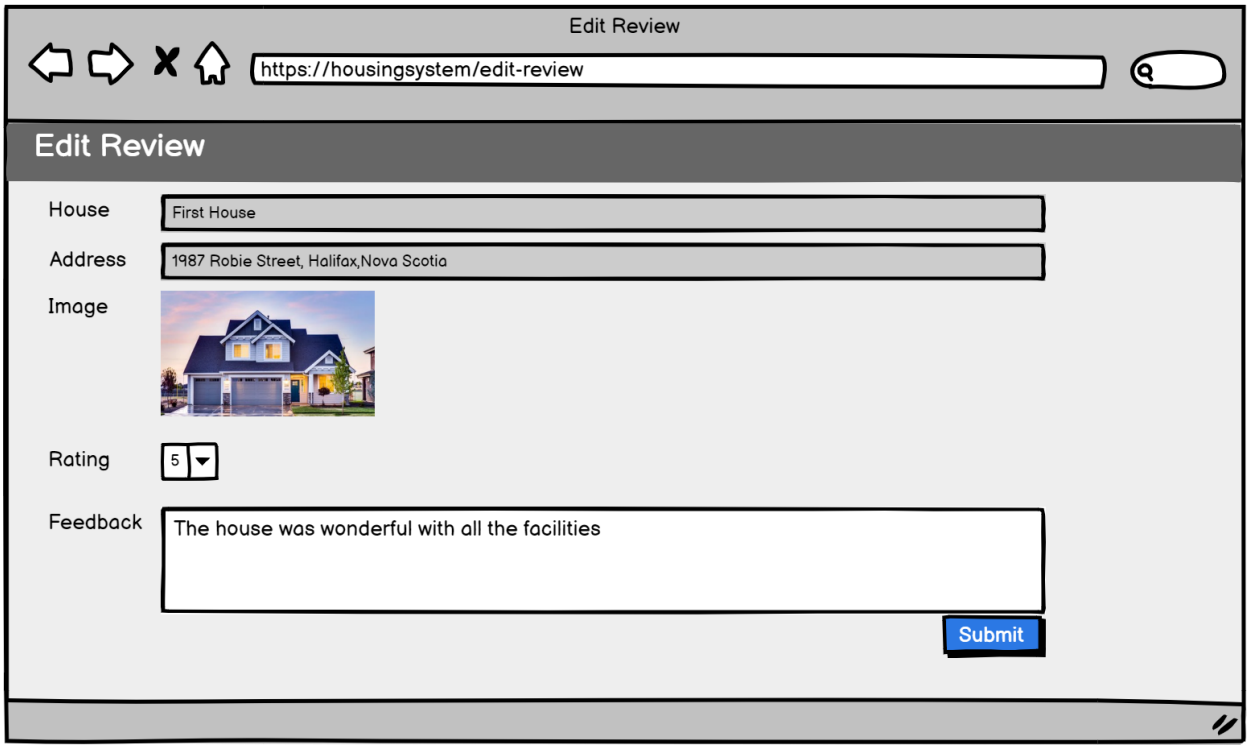

Figure 22: Wireframe – Editing review/feedback [1]

3.2.2 Design and Layout

Design of a web application plays an important role on creating impact on a user. A good design and layout can make a web application more appealing and attractive. It also draws the attention of customers to use the application thereby promoting to good user interface. In this project, we have developed all features and web pages with a simple and yet fascinating design look.

User management feature

The user who lands onto our application, needs to register and login to our application in order to access the functionality of our 'Housify' web application. In order to register, a user needs to provide some basic details like first name, last name, email id etc., in a proper valid format. A registered user will be able to login to our application if he enters correct email id and password. A successfully logged in user will have an opportunity to view his profile, edit his profile and see the saved houses that he has wish listed from the dashboard of our application. Below are the screenshots of the layouts that are designed for tasks involved in this feature.

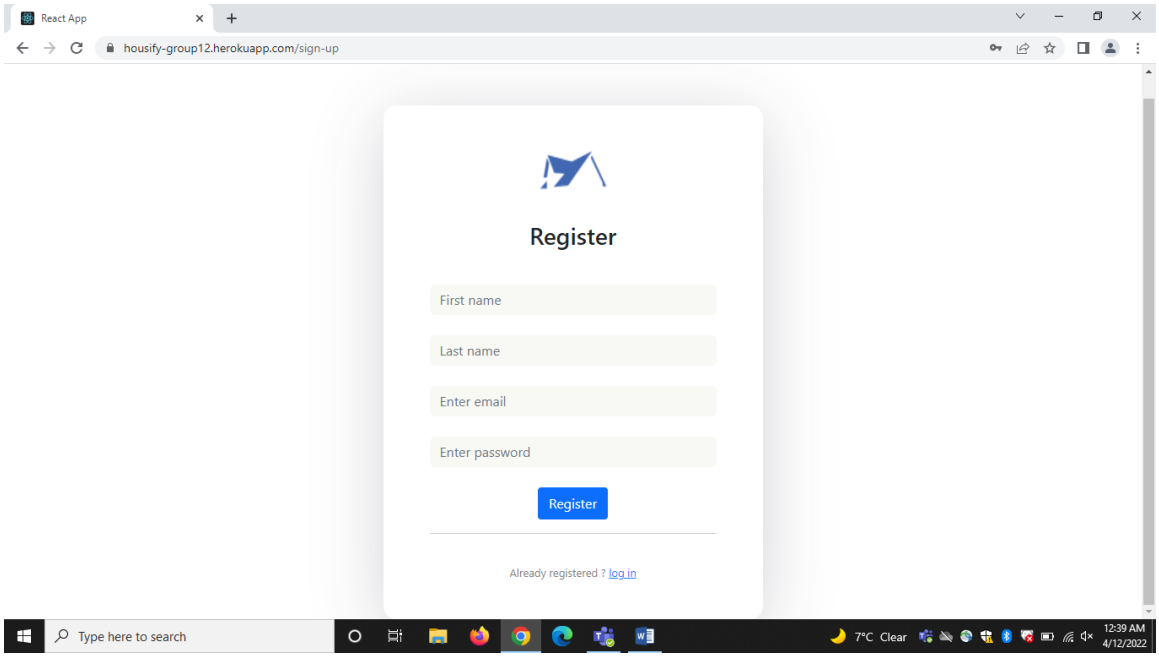

Figure 23: Application Design – Register

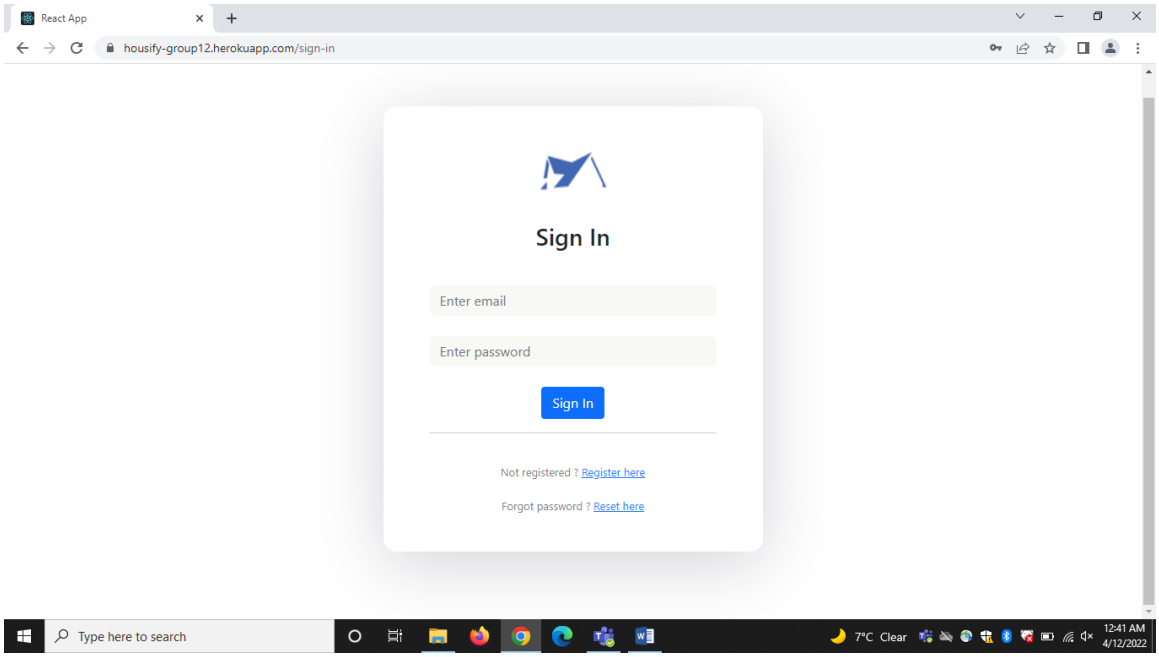

Figure 24: Application Design – Sign In

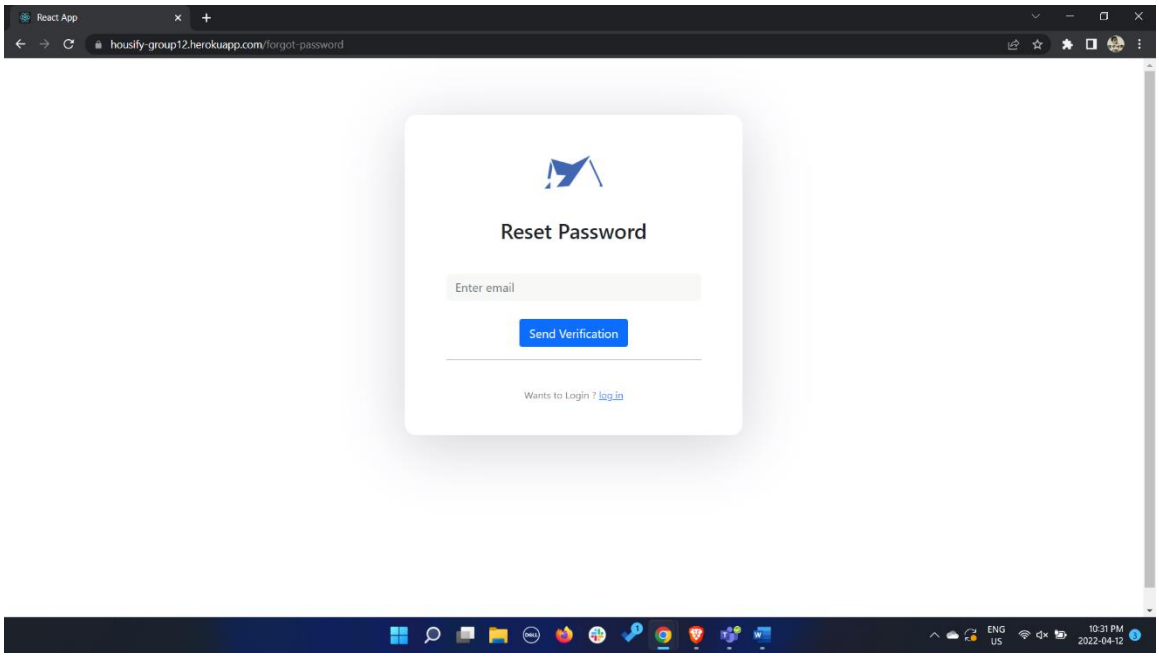

Figure 25: Application Design – Reset Password

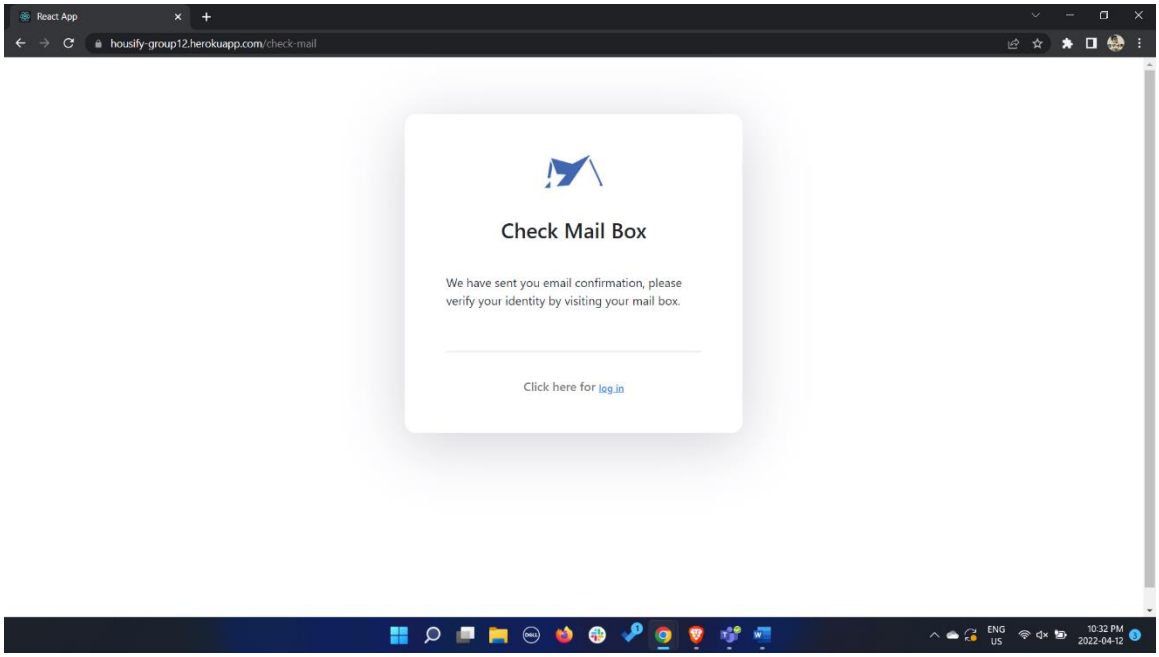

Figure 26: Application Design – Reset Password Verification

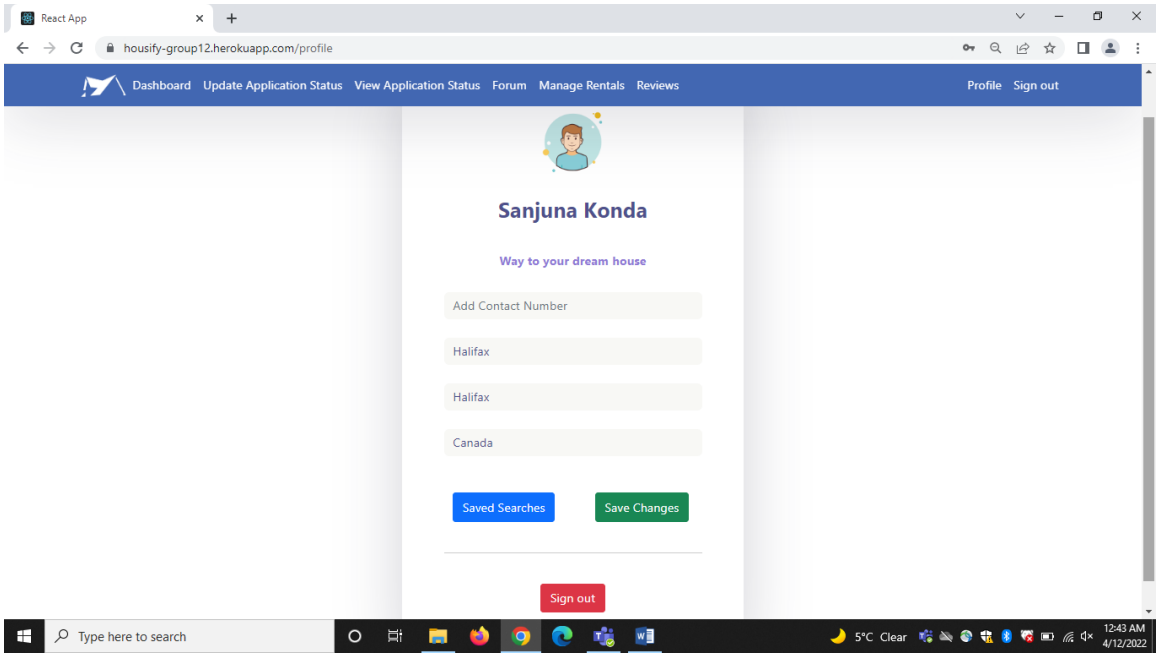

Figure 27: Application Design – User Profile

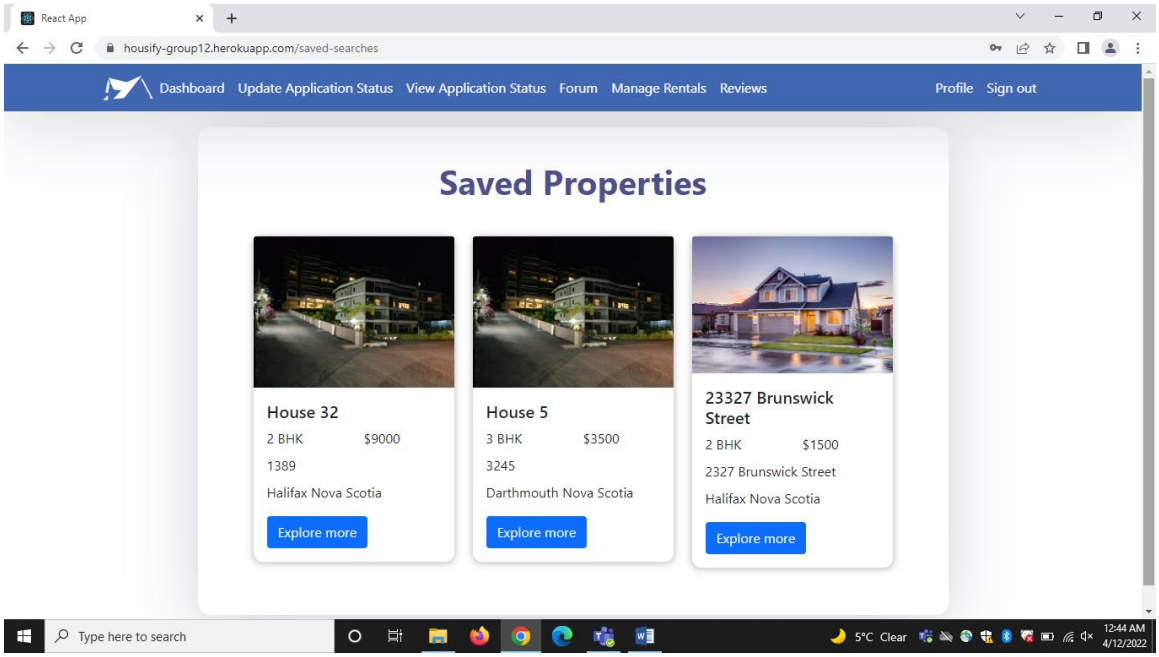

Figure 28: Application Design – User Saved Properties

Search houses using filter option

A successfully logged in user wants to navigate through our application and view all the available houses for rent or sale. So, sometimes he needs to filter houses depending upon his choice and convenience. To meet this requirement, we have come up with a feature of filtering houses depending upon city, cost, type of house (1 bhk, 2 bhk etc) and number of people that can share the house. Below are the screenshots of the layouts that are designed for tasks involved in this feature.

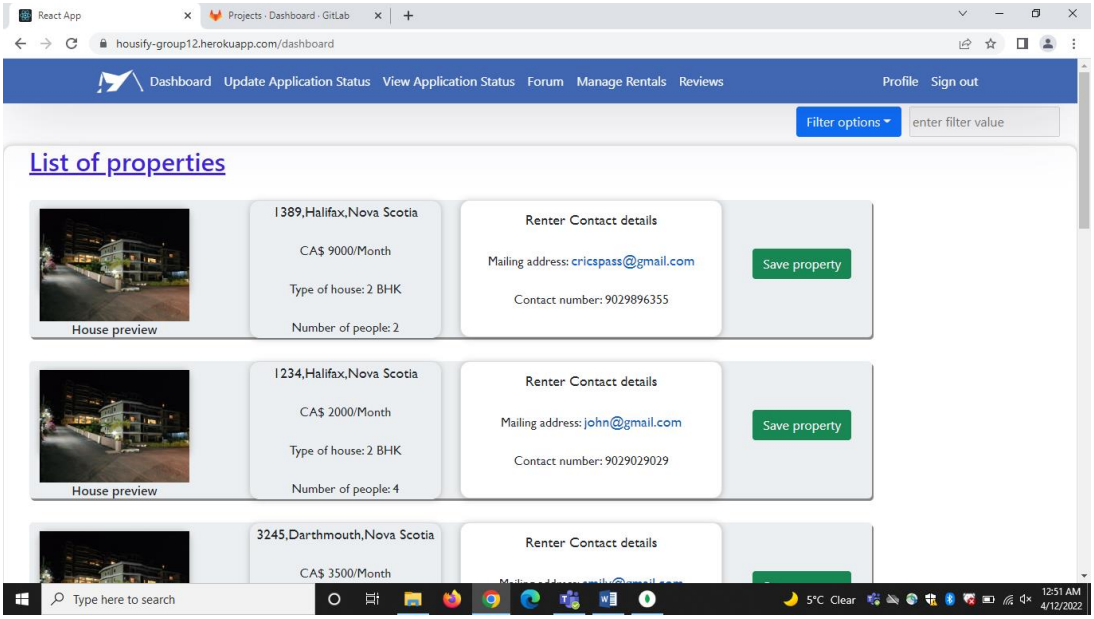

Figure 29: Application Design – Dashboard

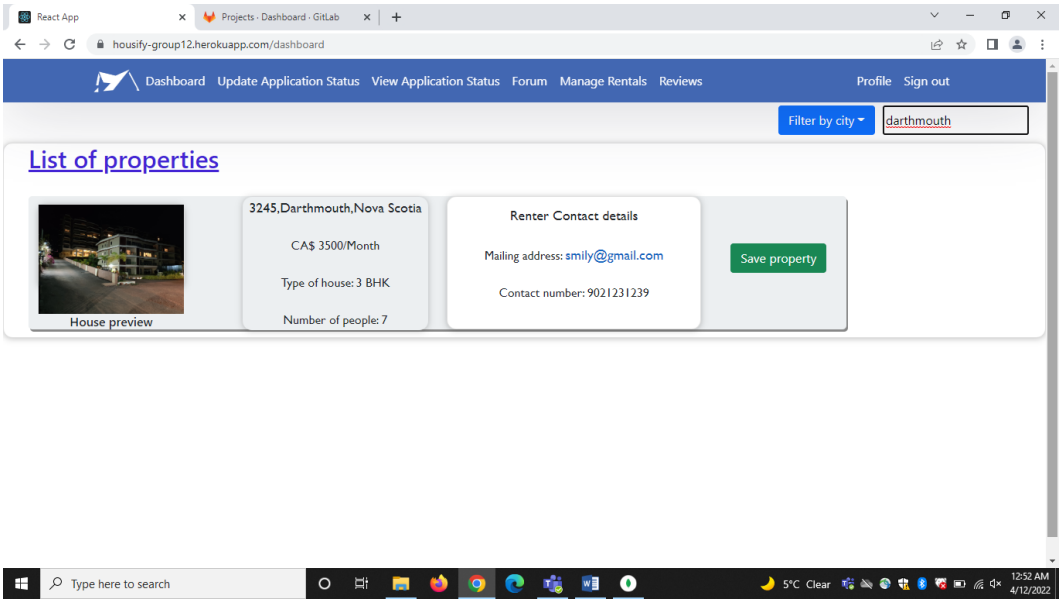

Figure 30: Application Design – Filtered properties in Dashboard

Application dashboard (application form)

A user who uses our application may want to look for some medium that allows him to apply for a particular house. This feature is all about providing user with an opportunity to apply for a house, view its application status whether its approved, in progress etc., and list all his applications. Below are the screenshots of the layouts that are designed for tasks involved in this feature.

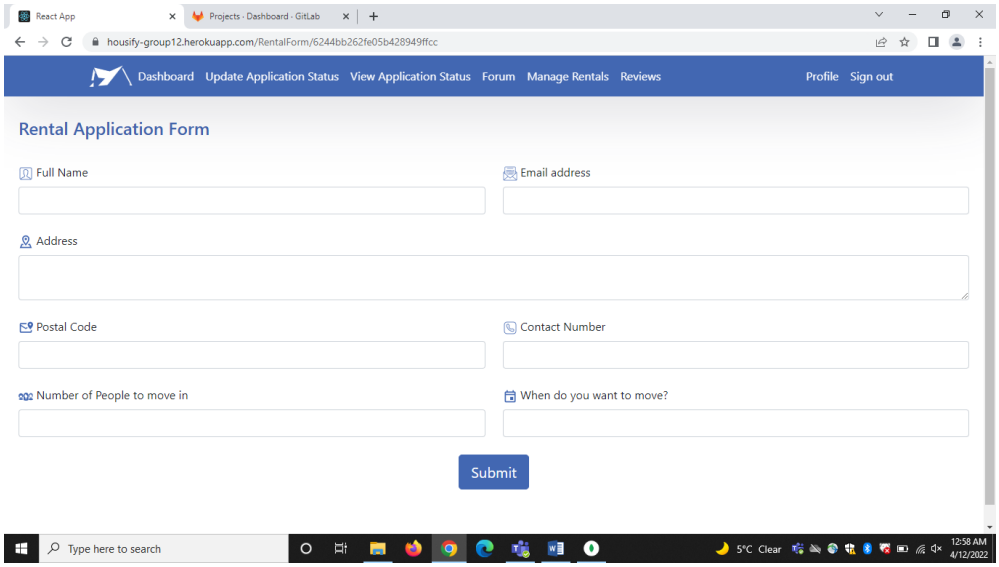

Figure 31: Application Design – Rental Application Form

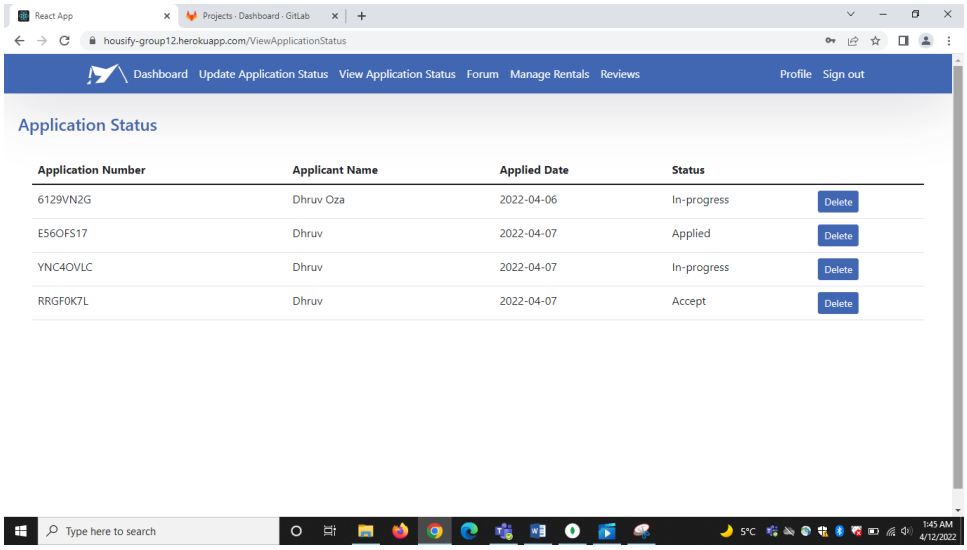

Figure 32: Application Design – Applications List (Renter)

Community dashboard

A user who is using our application will be provided with an opportunity to create a thread in which he can describe about a house or community and can be displayed to all other users who use our application. The user can also comment on previously posted threads thereby sharing his opinion on it. In this way, this particular feature enables users to share their opinions about several rental properties and comment on them thereby building healthy community relationships. Below are the screenshots of the layouts that are designed for tasks involved in this feature.

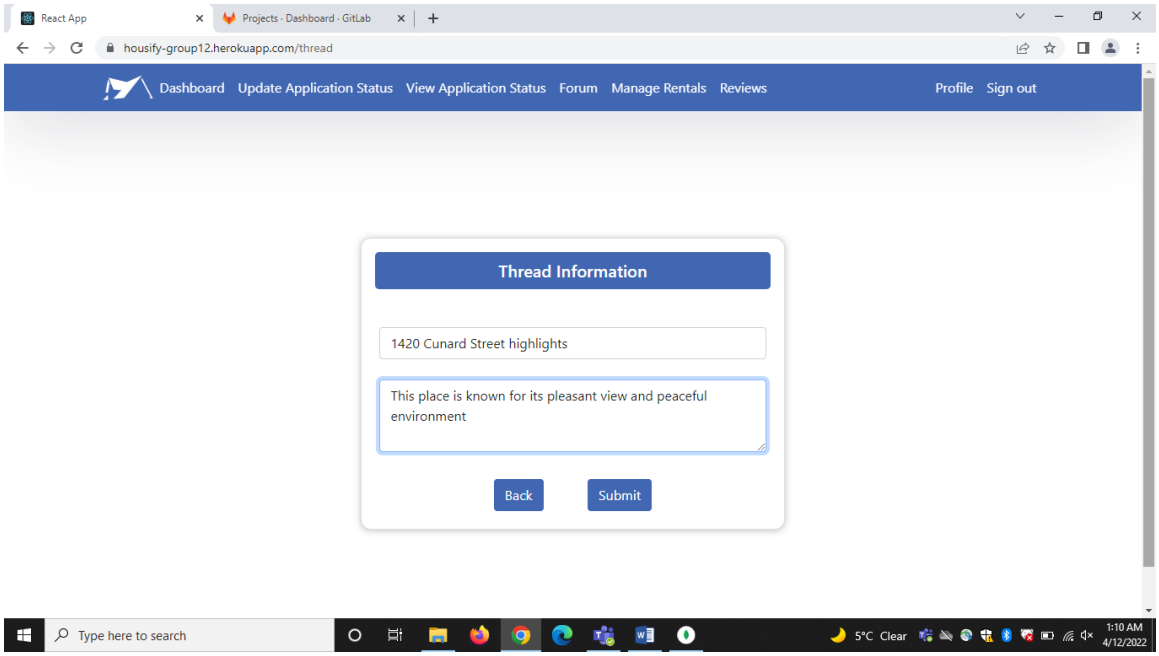

Figure 33: Application Design – Discussion/Community Dashboard

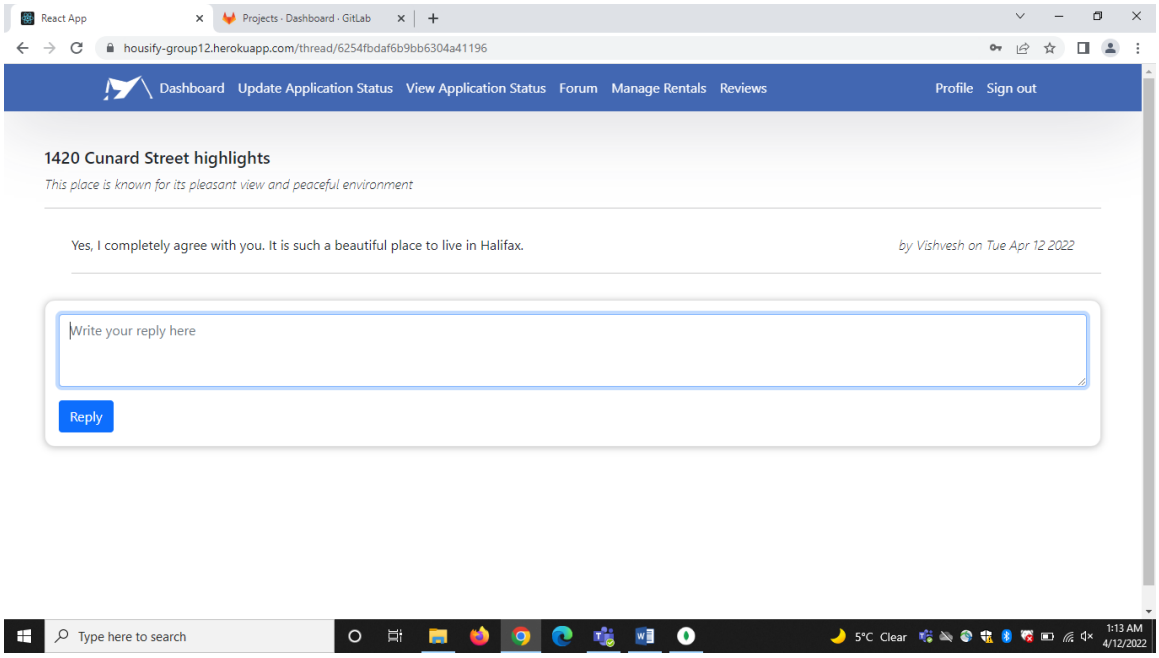

Figure 34: Application Design – Single Discussion Forum

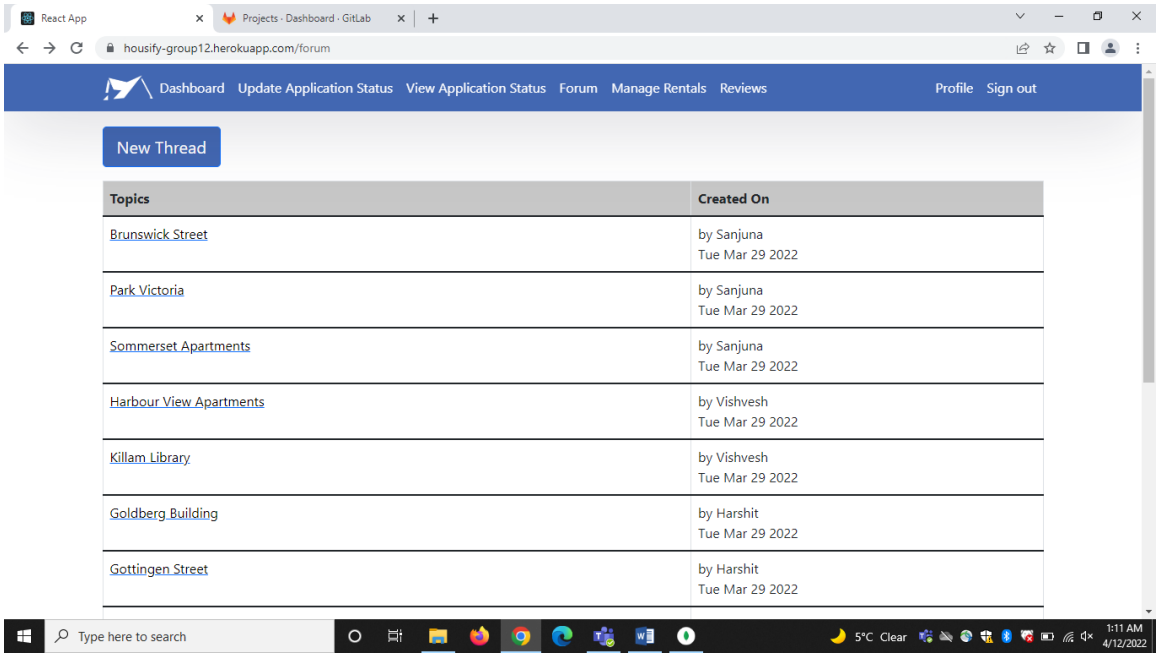

Figure 35: Application Design – List of threads/discussions

Manage rentals/houses

With this feature, the user has the ability to list out the house for rent out the house for other users. The user will create new post/ad for renting out the house which would be viewed by all other tenants. Furthermore, the house listed by the renter can be updated for editing the details of the house with the help of this feature. The renter is also provided with the option of deleting the post, if the house is rented or in any other situation. All-inclusive, this feature is more inclined towards the renters. Below are the screenshots of the layouts that are designed for tasks involved in this feature.

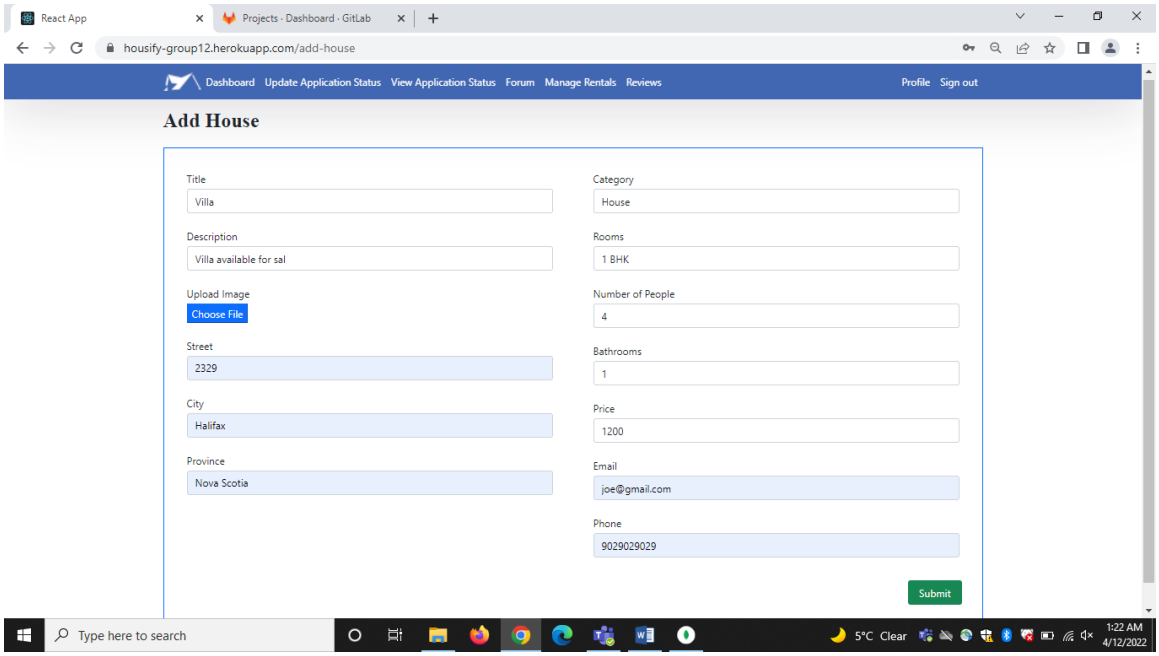

Figure 36: Application Design – Add Property/House

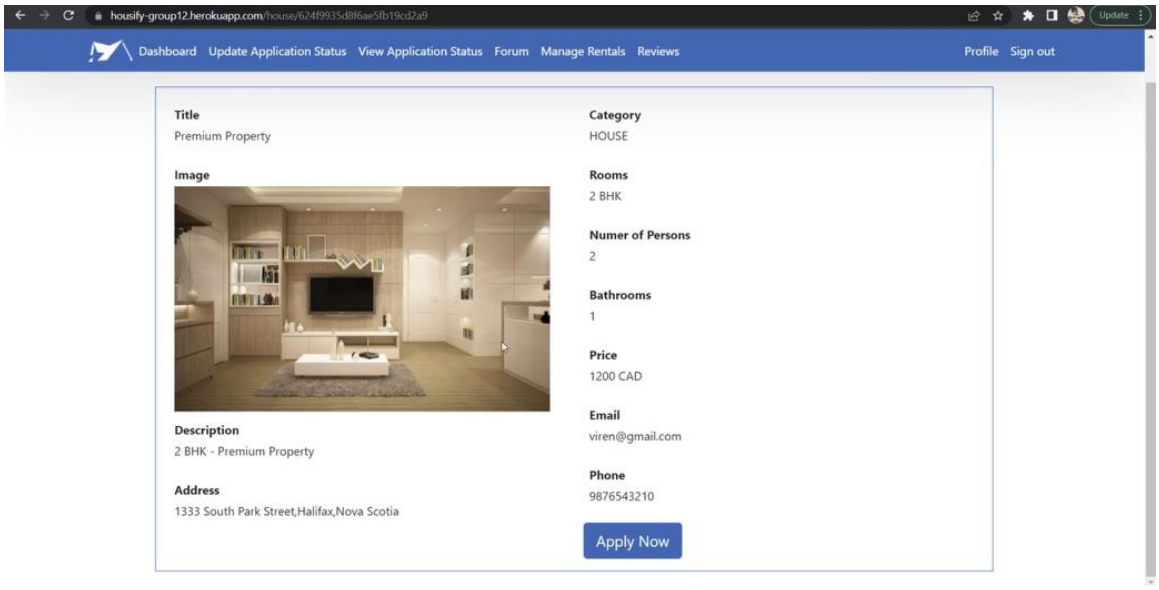

Figure 37: Application Design – View property/House

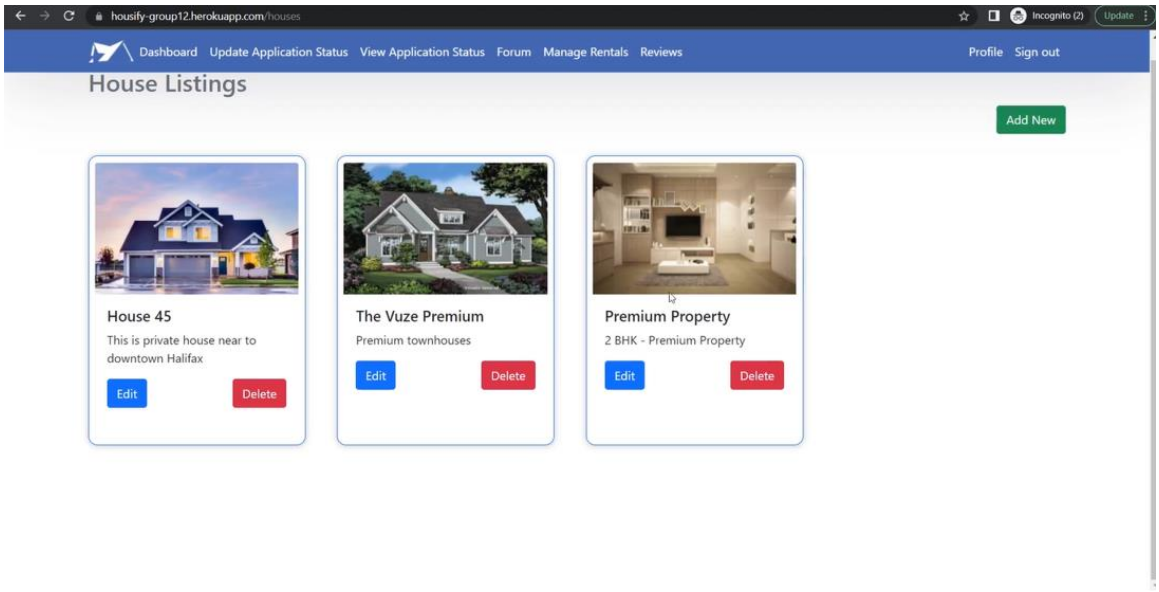

Figure 38: Application Design – List of Properties/Houses (Renter)

Housing analysis

This feature enables the tenants to provide review for the house they are currently living in or have previously lived in. The tenants can add the rating and feedback for the houses, and also, they can edit the same.

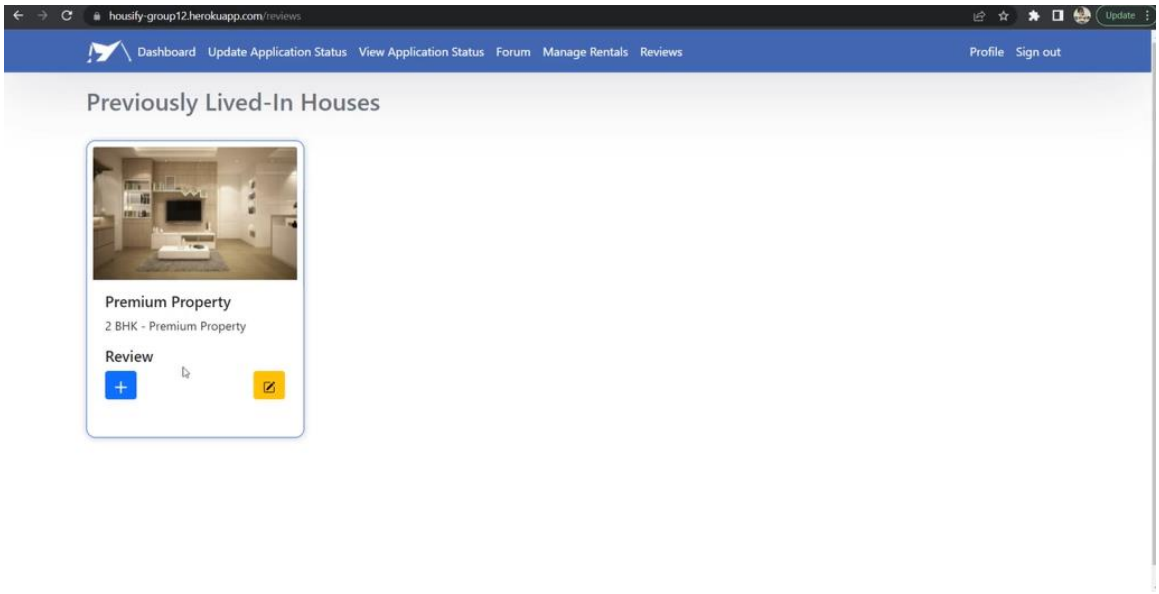

Figure 39: Application Design – Register

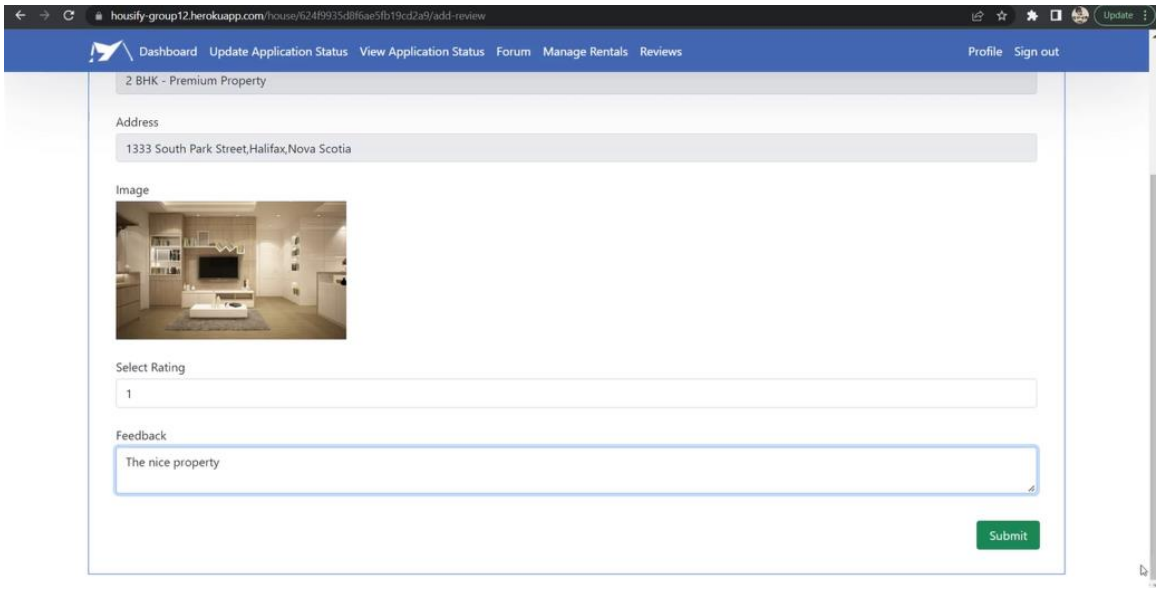

Figure 40: Application Design – Add House

4. APPLICATION WORKFLOW

4.1 Interaction Design

User persona

User persona below defined the user situation and why there is need of application considering the current market scenario and need, as user wats to search for rental properties quick and landlords wants to advertise their property on the platform. Below are the screenshots of the layouts that are designed for tasks involved in this feature.

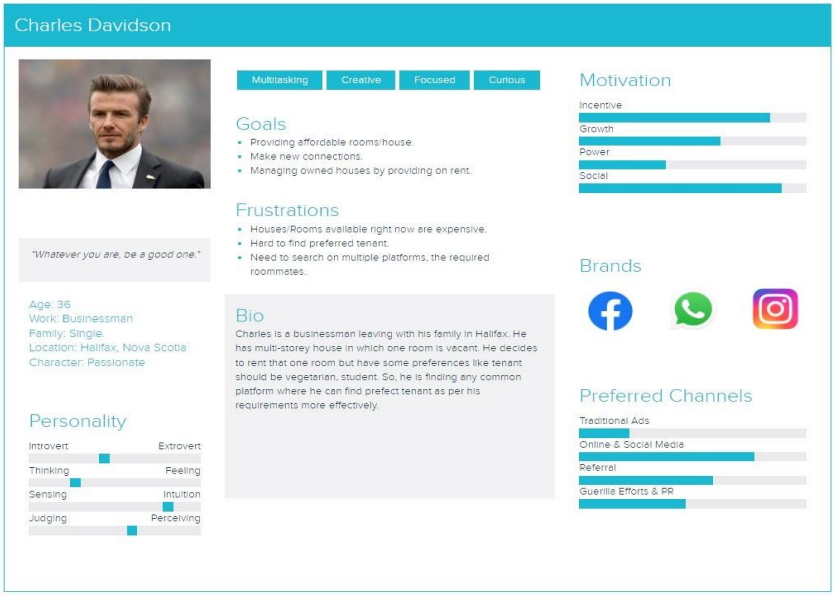

Figure 41 : Charles Davidson - User Persona [2]

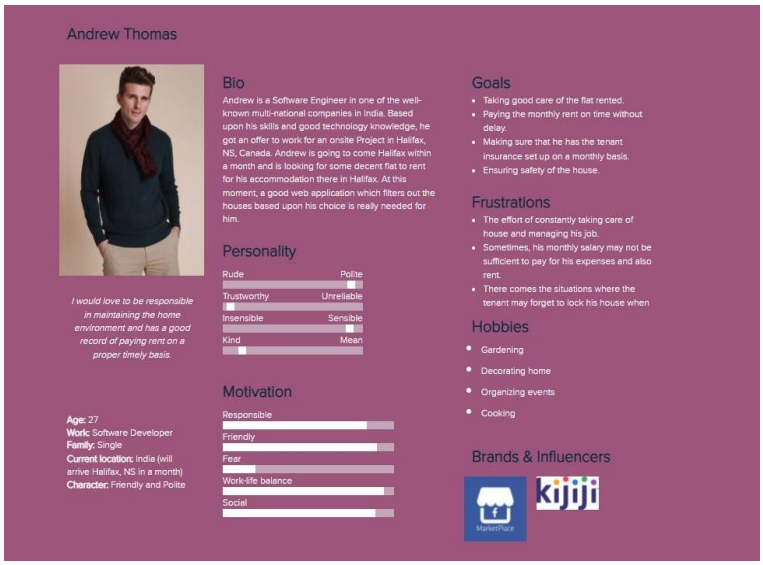

Figure 42: Andrew Thomas - User Persona [2]

User Scenario

Profile Management:

Register a user

- Scenario: John a tenant looking for specific properties in the nearby area he wants to live in Halifax, he wants to use ‘Housify’ application to access all the information about information of available houses in nearby area, John wants to Register to the ‘Housify’ to access the platform [1].
- User Persona: Tenant [1]
- Feature: Profile Management [1]
- Need: A user wants to access the ‘Housify’ platform [1]
- Context: John wants to access the platform so that he can fulfill the aim as he is finding the house to rent nearby [1]

Login a user

- Scenario: John a tenant looking for specific properties in the nearby area he wants to live in Halifax, he is already registered with ‘Housify’ application, however john wants to access the platform once again, as he logged out from the last time, so that he can securely access the platform once again [1]
- User Persona: Tenant [1]
- Feature: Profile Management [1]
- Need: A user wants to access the ‘Housify’ platform [1]
- Context: John wants to access the platform so that he can fulfill the aim as he is finding the house to rent nearby [1]

View or edit profile

- Scenario: John a tenant registered with ‘Housify’ and now moving to some other city, John wants to change his housing location preferences, so that he can access the information about house rentals from that particular area [1].
- User Persona: Tenant [1]
- Feature: Profile Management [1]
- Need: A user wants to change the location preferences on the ‘Housify’ platform [1]
- Context: John wants to change the location so that he can explore rental properties of the desired place [1].

View saved properties

- Scenario: John a tenant registered with ‘Housify’ and he has already listed some of the properties by clicking on the save button. He now wants to access properties [1]

- User Persona: Tenant [1]
- Feature: Profile Management [1]
- Need: A user wants to access the previously visited saved house / properties.
- Context: John wants to view saved properties [1].

Payment Module

Pay a deposit

- Scenario: Claire as a new tenant wants to pay her deposit fees to the landlord and wants to access digital payment options. Moreover, Claire also wants to save this payment information for future payment purposes [1].
- User Persona: Tenant [1]
- Feature: Payment Module [1]
- Need: A user wants to pay the deposit [1]
- Context: Claire wants to pay deposit to the landlord [1]

Accessing payment history

- Scenario: Claire as a tenant wants to access the previous payment history made through the website/platform. This is the confirmed payment from both parties, one payment released, and another payment received [1].
- User Persona: Tenant [1]
- Feature: Payment Module [1]
- Need: A user wants to access the payment history [1]
- Context: Claire wants to access the payment history to analyse [1]

Monthly rent payment

- Scenario: Claire as a tenant wants to pay the monthly rent of the property she is living, mentioned that she already paid deposit and now she wants to pay the monthly rent either as a auto-deposit or as a manual payment option [1].
- User Persona: Tenant [1]
- Feature: Payment Module [1]
- Need: A user wants to pay monthly rent [1]
- Context: Claire wants to pay her monthly rent of the property [1].

Search for houses using filter option

Filtering the houses based upon location

- Scenario: John is arriving to Halifax in a week in order to do his master's at Dalhousie University. So, he would like to search for a house near downtown

Halifax so that he can enjoy city's culture and also take up number of buses to reach university [1]

- User Persona: Tenant [1]
- Feature: Filtering the houses based upon location [1]
- Need: To search for a house near downtown [1]
- Context: As John is arriving to Halifax soon, he can search for some house near Downtown for his accommodation in Halifax [1]

Filtering the houses based upon the type of house (1-bedroom, 2-bedroom etc.)

- Scenario: John is looking for an apartment nearby Downtown Halifax and successfully found some of houses using filter by location. Later, he wanted to go for only those houses which includes only one bedroom such that he can manage rent and live alone maintaining his privacy [1]
- User Persona: Tenant [1]
- Feature: Filtering the houses based upon the type of house [1]
- Need: To search for a 1-bhk house [1].
- Context: After filtering houses near Downtown Halifax, John wants to filter only 1-bhk houses so that he can live comfortably throughout his masters [1] .

Filtering the houses based upon the cost

- Scenario: John got the deals of some 1-bhk houses near Downtown, but now he wants to filter among those houses which are expecting less rent so that he can save his money for future expenses [1]
- User Persona: Tenant [1]
- Feature: Filtering the houses based upon the cost [1].
- Need: To search for a house with low rent [1].
- Context: John wants to search for a 1-bhk house near Downtown which is expecting less monthly rent [1].

Filtering the houses based upon number of persons willing to live in that particular house

- Scenario: Thomas is arriving Halifax in a week for his onsite project and looking for an accommodation in Halifax. He wants to share the house with one other person so that the rent can be shared among both of them equally, thereby reducing the entire rent burden on him [1]
- User Persona: Tenant [1]
- Feature: Filtering the houses based upon number of persons willing to live in that particular house [1].
- Need: To search for a house in which 2 people can live [1].
- Context: As Thomas is arriving to Halifax soon, he searches for a house on sharing basis with one other person to decrease the rent burden [1].

Application Dashboard:

Apply for rent

- Scenario: John would want to share a room near his university with a bachelor student, and he would also like to rent this room from any rental who is staying with family at an affordable rate [1].
- User Persona: Tenant [1]
- Feature: Apply for rent [1]
- Need: To apply for room near university and share it with preferred bachelor student [1].
- Context: Tenant wants to apply for room available near university [1].

View Application Status

- Scenario: John would want to see application status for which he had applied a month ago [1].
- User Persona: Tenant [1]
- Feature: View application status [1]
- Need: To view application status of any request made before [1].
- Context: Tenant wants to track application by going through application status [1].

List our applications

- Scenario: John would want to see all applications he applied before to rent a room [1].
- User Persona: Tenant [1]
- Feature: List out Applications [1]
- Need: To list out all applications for which request were made [1].
- Context: Tenant wants to see all applications for which request has been made [1].

Rental House Service and maintenance:

Create Service Request

- Scenario: John wants to replace his refrigerator and electric stove because he believes they are too old and are increasing his electric cost [1].
- User Persona: Tenant [1]
- Feature: Create Service Request [1]
- Need: To create service request for replacement of refrigerator and electric stove [1].
- Context: Tenant wants replacement of refrigerator and electric stove by creating service request [1].

View Service Requests

- Scenario: John would want to see all service request for which he had create service request a month ago [1].
- User Persona: Tenant [1]
- Feature: View Service Requests [1]
- Need: To view all service request made before [1].
- Context: Tenant wants to track service request through view service request feature [1].

Notification Management:

Viewing Notifications

- Scenario: Charles has posted a house for a rent, and he wants to view who are interested on his housing post [1].
- User Persona: Landlord [1]
- Feature: Notification Management [1]
- Need: Charles want to check who are interested on his housing post [1]
- Context: Charles get a notification from another user who is interested on his housing post [1]

Creating notifications

- Scenario: Charles is looking for a house or apartment to live. Charles find a perfect house and he want to know more about the housing. So, he clicks on interested button [1].
- User Persona: Tenant [1]
- Feature: Notification Management [1]
- Need: Showing interest on a housing post [1]
- Context: Charles creates a notification to notify other users about his interest [1].

Community Dashboard:

Create a thread

- Scenario: Charles wants to discuss about a particular housing area. Charles wants to know about the area and all amenities that are close to the area. All the benefits and drawback of living in that housing area [1].
- User Persona: Tenant [1]
- Feature: Discussion [1]
- Need: Wants to discuss about a particular topic [1]
- Context: Charles wants to discuss and know more about the topic that he/she is posting [1].

Commenting on a thread

- Scenario: Charles is already living in that housing area and has additional knowledge on the topic [1].
- User Persona: Tenant [1]
- Feature: Discussion [1]
- Need: Have some knowledge on the topic and wants to share it with others [1].
- Context: Charles comments on the topic and shares his views and other users will be able to view his point of view [1].

Manage Houses:

Add Post/Ad

- Scenario: A renter has a house other than he lives in, which he wants to rent to other tenants. The user creates an ad/post with images to rent out his house [1]
- User Persona: Renter [1]
- Feature: Add post/ad [1]
- Need: To rent out the house and make it available for the tenants [1].
- Context: Renter wants tenants to live in the house by viewing the ad that he created [1].

View Post/Ad

- Scenario: A renter has created a post/ad(s) for his house. He needs to scan the display of the ad/post. To get the idea about how it will be displayed to the tenants [1]
- User Persona: Renter [1]
- Feature: View post/ad [1]
- Need: To view post/ad according made by the renter [1].
- Context: Renter takes overview of the ad/post visible to the other tenants [1].

Delete Post/Ad

- Scenario: A renter has rented out his house to the tenants. He needs to delete the ad/post, so that there it would not cause confusion to other tenants [1]
- User Persona: Renter [1]
- Feature: Delete post/ad [1]
- Need: To delete the ad/post of the house after it is rented [1].
- Context: Renter receives inquiries about the house, although it is rented out. The renter wants to delete the post/ad [1]

Housing Analysis:

Add review

- Scenario: A tenant who has previously lived at house or is currently living there wants to give review about the house so that other persons can see and take help for making the decision of residing [1]
- User Persona: Tenant [1]
- Feature: Add review for the houses [1]
- Need: To give proper review and feedback of the house which is useful for other customers [1]
- Context: Tenant wants to live in a house, and he uses reviews and feedback to strengthen his choice [1].

Edit Review

- Scenario: A tenant provides review about the house. In winter, he comes across situation where the heat is not working since a long time. He edits reviews about the house [1]
- User Persona: Tenant [1]
- Feature: Edit review for the houses [1]
- Need: Edit the submitted review according to the situations [1].
- Context: The Tenant can edit his review if he/she first provided a positive review, and after some dispute or problems, it becomes a negative review or vice versa [1].

Analytics

- Scenario: A renter has listed an ad/post of his house. He is curious about the statistics of the house such as how many persons have provided reviews about the house, number of tenants living/lived in the house [1].
- User Persona: Renter [1]
- Feature: View analytics of the house [1]
- Need: To get an idea about the statistics of the house [1].
- Context: The renter wants to add another post/ad of house. He compares the number of tenants and other statistics to post the ad [1].

Meet with us:

Book an appointment

- Scenario: Joe recently came across our application and wanted to make use of it in order to search for a house as he is currently living in a temporary accommodation. So, he wants to schedule an appointment with the tenant to get the detailed information about the property and clarify his doubts so that he can take a wise decision in the house selection process [1].
- User Persona: Tenant [1]
- Feature: Meet with us [1]
- Need: To meet the renter and have a conversation in order to get an idea about the houses being rented [1].
- Context: Tenant wants to avoid confusion and hence decides to schedule an appointment in order to meet the renter [1]

Edit/Modify a scheduled appointment

- Scenario: Joe booked a appointment with the renter and the appointment is on February 25th. But he got some urgent work to do at that time and so he decided to change the appointment date and time to February 26th as he will be free that time [1].
- User Persona: Tenant [1]
- Feature: Meet with us [1]
- Need: To update the scheduled appointment [1]
- Context: The tenant wants to update the appointment date and time as he/her got some urgent work or unavailable at that time [1].

Cancel an appointment

- Scenario: Joe booked an appointment with the tenant on a specific day and time. But later, he got the entire information about the apartment from one of his friends who lived there a month ago. So, now Joe wants to cancel the appointment as it is waste of time as he already has enough knowledge about the house [1].
- User Persona: Tenant [1]
- Feature: Meet with us [1]
- Need: To cancel the appointment [1]
- Context: The tenant wants to cancel the appointment as he is no longer interested in attending it [1].

Use Cases:

Feature 1: Profile Management

Task 1: Register a user to 'Housify' [1]

(Assume user discovered app for the first time and want to continue)

Scenario:

A person who is looking for an affordable housing property in with specific choices such as, a property should include all the extra amenities (hydro, heater) and more filters like this. For this purpose, a user wants to explore the 'Housify' web-app, and as a next step to get started with the web-app user needs to register first [1].

Use Case: Register for the web-app for further access [1]

1. User visits 'Housify' web-app
2. System displays homepage / housing property postings
3. User clicks on register button
4. System displays the registration page asking full name, email id, password and confirm password.
5. User enters full name, email id, password and confirm password
6. User clicks on the 'Register' button
   - 6.1. System displays error message for empty full name
     - 6.1.1. User enters valid full name
     - 6.1.2. User clicks on 'Register' button
   - 6.2. System displays error message for invalid full name (name without valid string)
     - 6.2.1. User enters valid full name
     - 6.2.2. User clicks on 'Register' button
   - 6.3. System displays error message for empty email id
     - 6.3.1. User enters valid email id
     - 6.3.2. User clicks on 'Register' button
   - 6.4. System displays error message for invalid email id
     - 6.4.1. User enters valid mail id as required
     - 6.4.2. User clicks on 'Register' button
   - 6.5. System displays error message for invalid password (password does not follows minimum requirements – minimum 8 characters, at least one upper case and one special character)
     - 6.5.1. User enters valid password as per the requirement
     - 6.5.2. User clicks on 'Register' button
   - 6.6. System displays error message as password and confirm password are not same
     - 6.6.1. User enters same password
     - 6.6.2. User clicks on 'Register' button

7. System displays positive response, user registered successfully.
8. System displays dashboard to the user

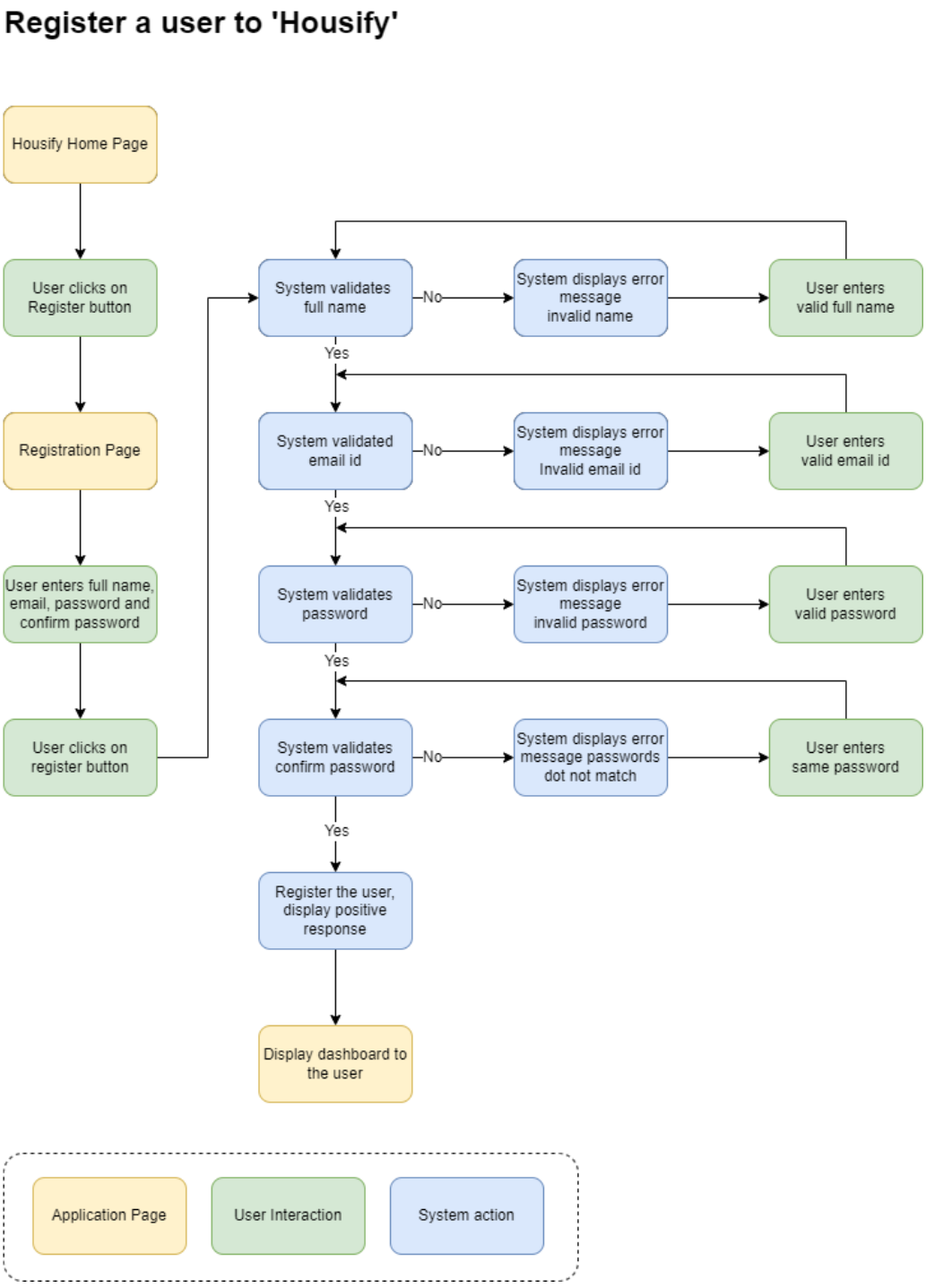

Figure 43: Task flow for Register a User [1]

Task 2: Login a user to ‘Housify’ [1]

(Assume user already registered and want to continue)

Scenario:

A person who is already registered for the ‘Housify’ web-app and want to access all the listings and house properties available in nearby or specific areas, for that a person wants to login to the system to continue to the web-app [1].

Use Case: Login for the web-app for further intractability and actions [1]

1. User visits ‘Housify’ web-app
2. System displays homepage / housing property postings
3. User clicks on login button
4. System displays the login page asking email id and password.
5. User enters email id and password
6. User clicks on the ‘Login’ button
    - 6.1. System displays error message for invalid email id
        - 6.1.1. User enters valid mail id as required
        - 6.1.2. User clicks on ‘Login’ button
    - 6.2. System displays error message for invalid password
        - 6.2.1. User enters right password
        - 6.2.2. User clicks on ‘Login’ button
    - 6.3. System displays error message for invalid password
        - 6.3.1. System displays forgot password link
        - 6.3.2. User clicks on forgot password link
        - 6.3.3. System displays field to enter mail id
        - 6.3.4. User enters the email id
            - 6.3.4.1. System displays error message for invalid email id
            - 6.3.4.2. User enters valid email id
            - 6.3.4.3. System accepts the email id
        - 6.3.5. User clicks on link, reset my password
        - 6.3.6. System sends code to the registered email id
        - 6.3.7. System displays box to enter sent code (OTP)
        - 6.3.8. User enters the code (OTP)
        - 6.3.9. User clicks on ‘Verify’ button
            - 6.3.9.1. System displays error message – invalid code (OTP)
            - 6.3.9.2. User enters valid code
            - 6.3.9.3. User clicks on ‘Verify’ button
        - 6.3.10. System displays new password and confirm password fields
        - 6.3.11. User enters new password and confirm password
        - 6.3.12. User clicks on set new password button

6.3.12.1. System displays error message for not following minimum password requirements (minimum 8 characters, at-least one capital and special character).

6.3.12.2. User enters valid password

6.3.12.3. User clicks on set new password button

6.3.13. User gets successful password reset message

6.3.14. System displays login screen

7. System displays positive response, user logged in successfully.
8. System displays dashboard to the user
9. User interacts with the 'Housify' web-app

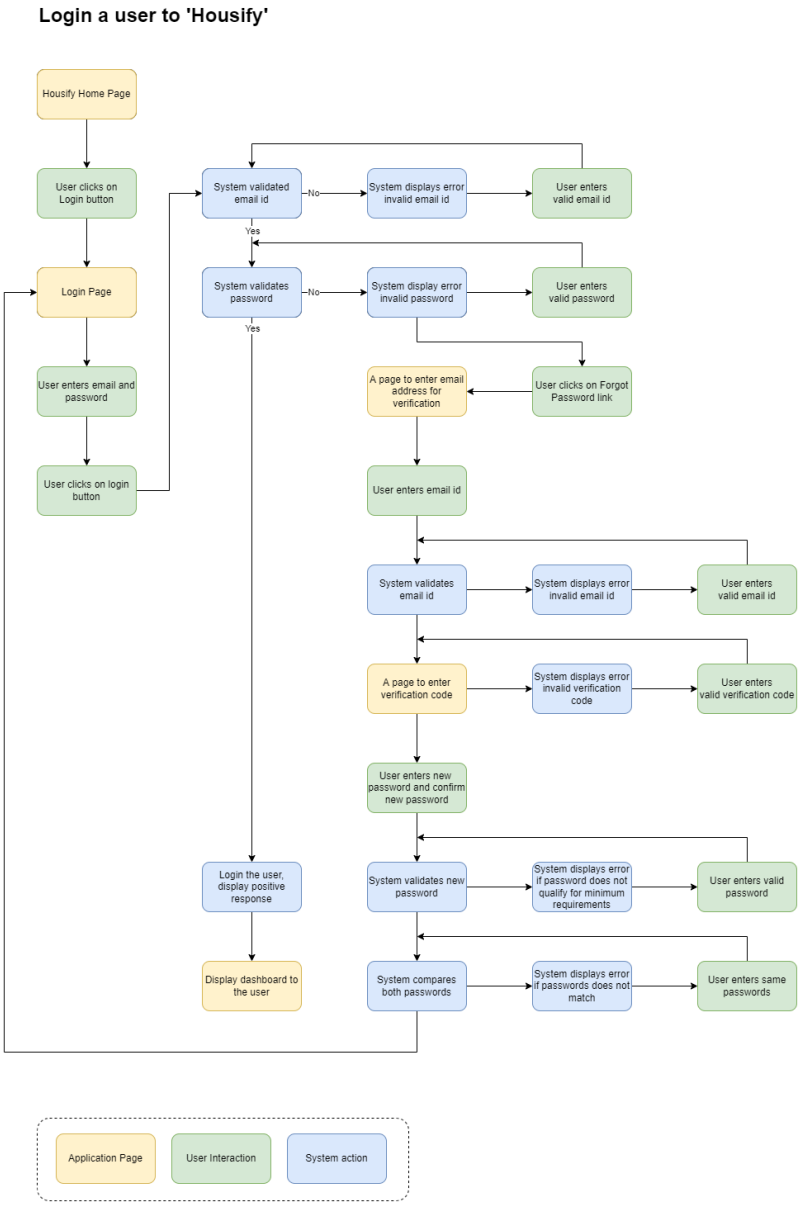

Figure 44: Task flow for Login a User [1]

Task 3: View and edit user profile [1]

(Assume user is logged in to ‘Housify’)

Scenario:

A user who is exploring the ‘Housify’ web-app and wants to update their profile information, and general preferences such as email id, preferred location country, user wants to go the profile section and do all these activities [1].

Use Case: View and edit information under user profile section [1]

1. User clicks on the profile button
2. System displays profile to the user
3. All user details are available under the user profile section
4. User clicks edit name icon
5. User enters new name
6. System displays update button to the user
7. User clicks on update button
   7.1. System displays error message ‘Invalid Name’ (as there might be numbers)
   7.2. User enters proper name
   7.3. User clicks on update button
8. User clicks on change location icon
9. User enters new valid location
10. User clicks on update button
    10.1. System shows error message ‘Invalid Location’
    10.2. User enters valid location
    10.3. User clicks on update button
11. System displays updated profile information
12. User can view updated profile as per entered information

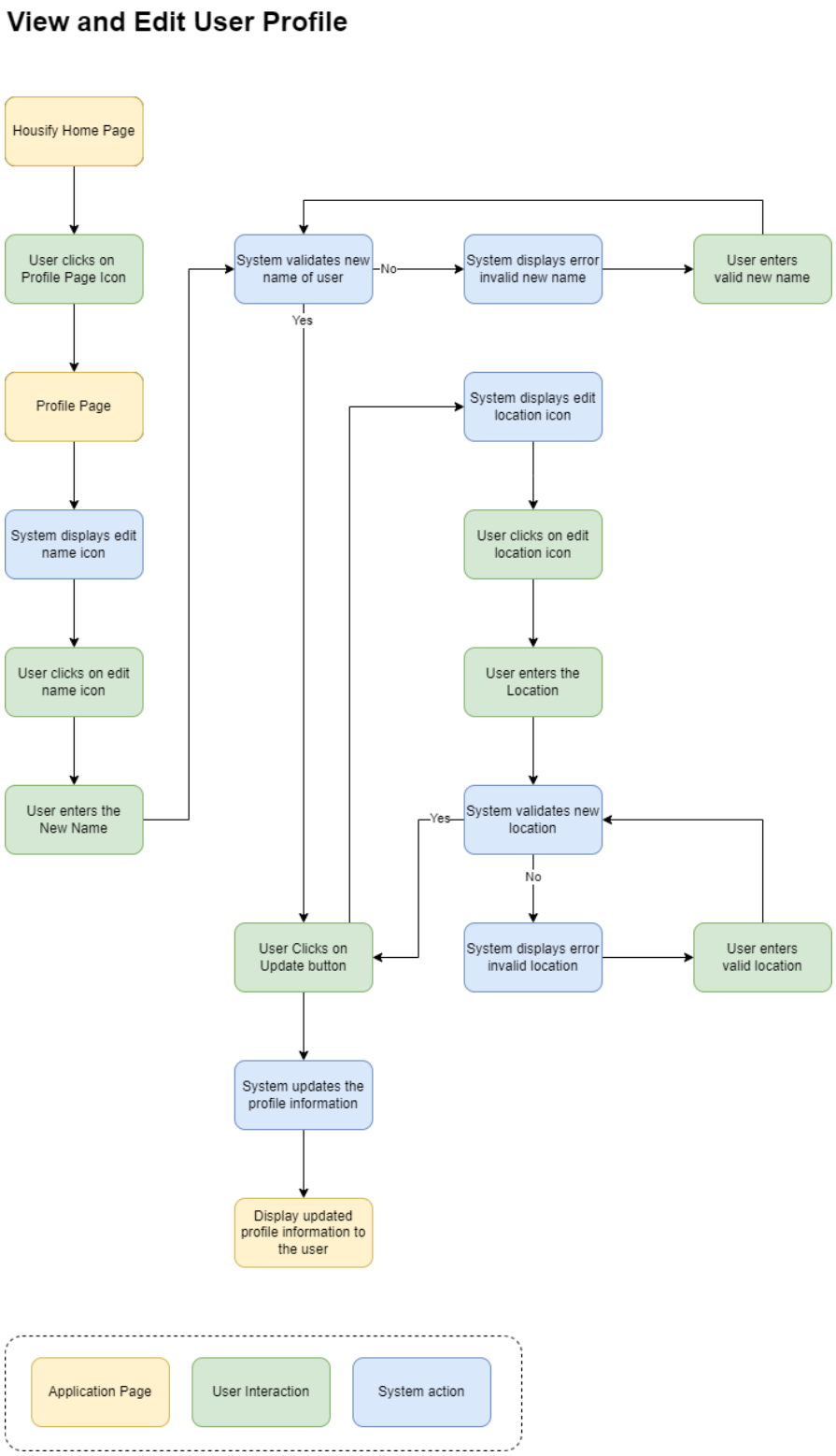

Figure 45: Task flow for View and Edit User Profile [1]

Task 4: View Saved Searches [1]

(Assume user is logged in to ‘Housify’)

Scenario:

A user, who is searching for houses and going through the listing one by one, he wants to visit saved houses/properties again which he was shortlisted earlier. In the profile, user has button to view the saved items [1].

Use Case: View saved house in profile [1]

1. User visits ‘Housify’ web-app
2. System displays homepage / housing property postings
3. User clicks on the profile button
4. System displays profile to the user
5. All the profile activity information is available under the profile section
6. User clicks on Saved Listing button
   6.1. User is unable to click on saved listing button
   6.2. Button is not visible as no listing is there saved by the user
   6.3. System shows no listing has been saved
   6.4. User clicks on see houses/properties button
   6.5. User inspects the house and click on save for later
   6.6. System shows positive sign that visited house listing is saved
   6.7. User goes back to the profile
   6.8. System shows user profile
   6.9. Steps repeats from number 2
7. System returns the list/view of the saved listings
8. User can scroll to view more listings saved in the profile
9. User clicks on the listing and see all the information
10. Users click on save button again (un-save) to remove that listing form the saved items
11. System gives positive response, removes that listing from the saved items.

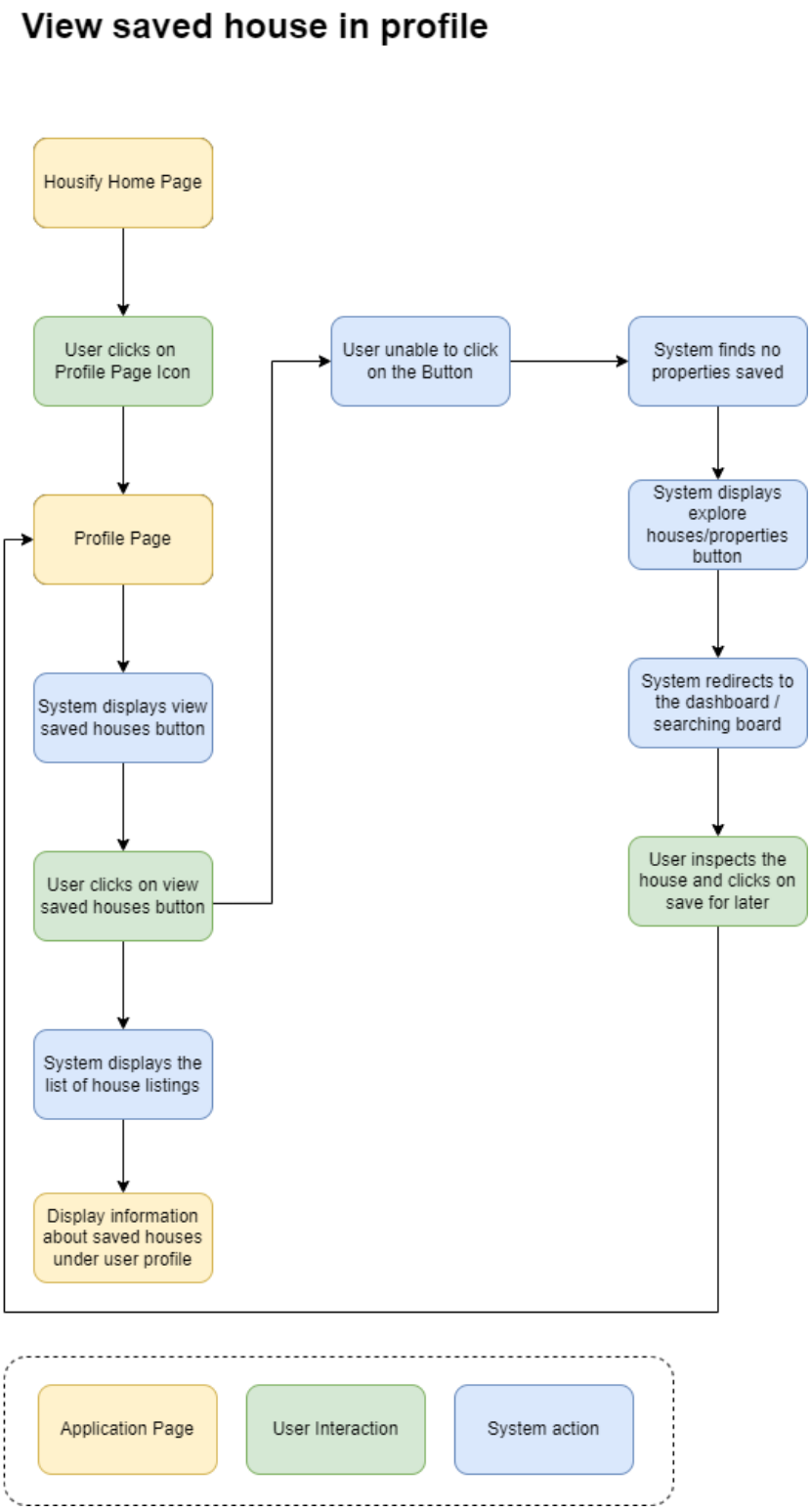

Figure 46: Task flow for View Saved House [1]

Feature 2: Payment Module

Task 1: Tenant wants to make a payment to the landlord [1]

(Assume user is logged in to the system and both tenant and landlord agreed for property rental)

Scenario:

A person who is looking to make a payment, i.e., a tenant wants to pay amount for the reason such as deposit or the monthly rent payment, will be using the feature [1].

Use Case: Pay amount to landlord (as a tenant) [1]

1. User visits 'Housify' web-app
2. System displays homepage / housing property postings
3. User clicks on profile button
4. System displays the profile page and can see submitted application
5. System displays positive sign and the approved property rental application
6. User clicks on the application
7. System displays pay deposit/rent as a next step
8. User clicks on the pay now button
9. System displays user a payment processing page
10. User enters banking details, and clicks on 'Pay' button
    - 10.1. System displays error message for empty card field
        - 10.1.1. User enters valid banking details
        - 10.1.2. User clicks on 'Pay' button
    - 10.2. System displays error message for invalid card field (invalid date/numbers)
        - 10.2.1. User enters valid banking details
        - 10.2.2. User clicks on 'Pay' button
11. System displays success message, as payment is processed
12. Confirmation and positive response is displayed to the user
13. Payment history is created, and user can view that

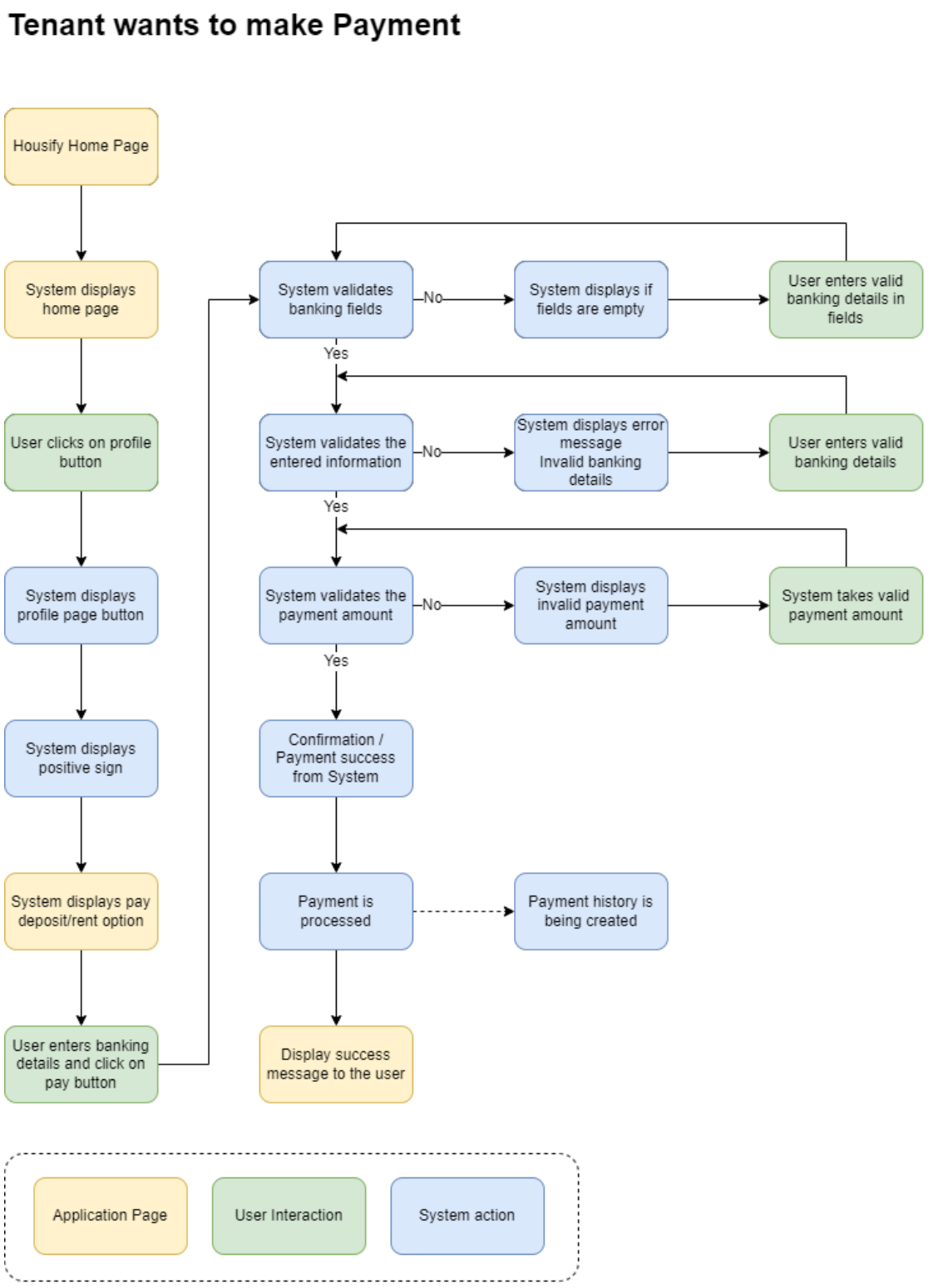

Figure 47: Task flow for Tenant want to make Payment [1]

Task 2: Tenant wants to view payment history [1]

(Assume user is logged in to the system and both tenant and landlord agreed for property rental, tenant has occupancy of the property and paying their monthly rents)

Scenario:

A person who paid rental deposit as well as a person who is paying monthly rents in regular basis want to view their payment history for analytical purposes [1].

Use Case: View payment history (as a tenant) [1]

1. User visits 'Housify' web-app
2. System displays homepage / housing property postings
3. User clicks on profile button
4. System displays the profile page and user can see rented property option
5. System displays rented property and brief information about the property.
6. User clicks on the property
7. System displays various options such as payment, view payment history
8. User clicks on view payment history option
9. System displays user a payment history page
10. User views payment history
11. User analyses the payment history with all details
12. User clicks on back button to go back to the rental property option

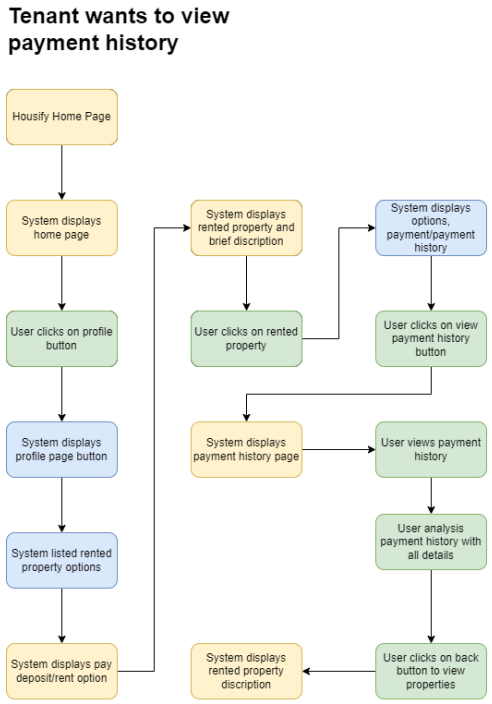

Figure 48: Task flow for Tenant want to view payment history [1]

Feature 3: Search for the house using filter option

Task 1: Filtering the houses based upon location [1]

Scenario:

John is arriving in Halifax in a week in order to do his master's at Dalhousie University. So, he would like to search for a house near downtown Halifax so that he can enjoy the city's culture and take several buses to reach university [1].

Use Case [1]:

1. User logins into the application and clicks on the Filter houses tab.
2. System displays the Filter options as location, type of house, count of persons, cost and amenities.
3. User selects location filter option.
4. Systems asks the user to choose the required location.
5. User chooses the location as Downtown Halifax.
6. System displays houses or flats located in or near Downtown Halifax.

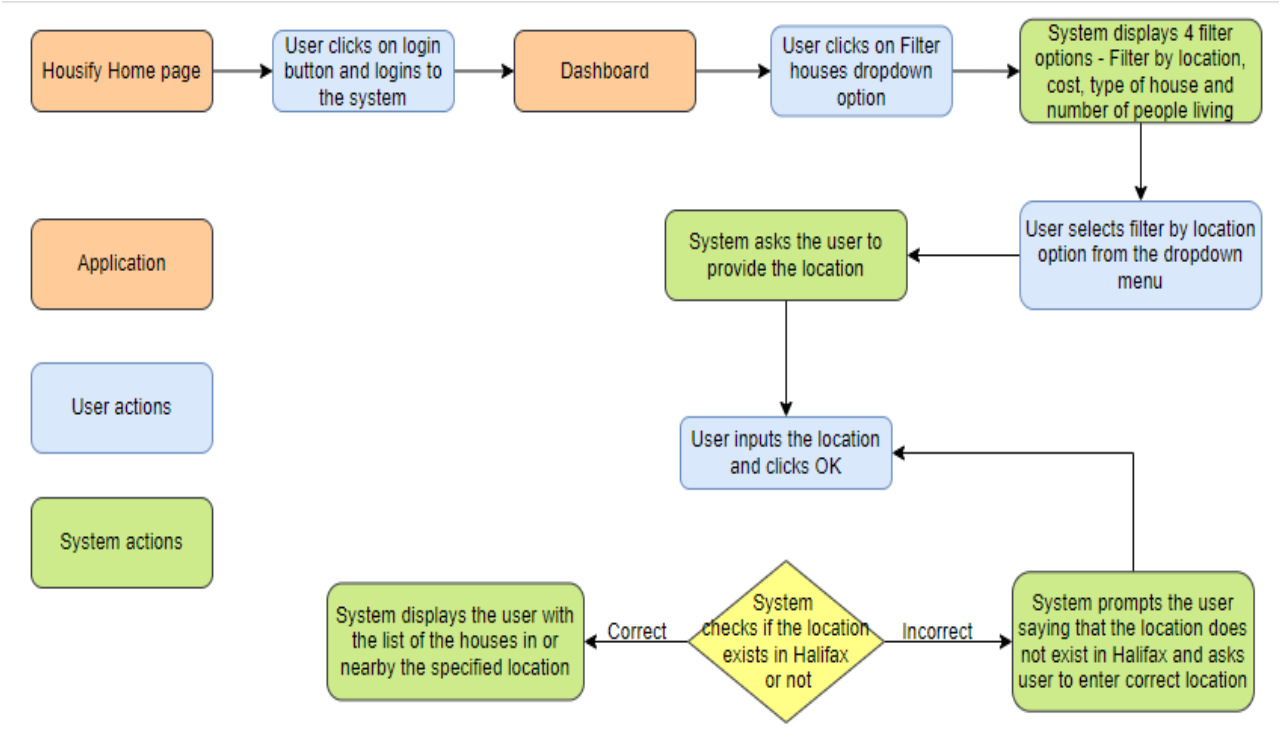

Figure 49: Task flow for House filter by using filter by location [1]

Task 2: Filtering the houses based upon the type of house (1-bedroom, 2-bedroom etc.) [1]

Scenario:

John is looking for an apartment near Downtown Halifax and successfully found some houses using filter by location. Later, he wanted to go for only those houses which include only one bedroom such that he could manage rent and live alone maintaining his privacy [1].

Use Case-2 [1]:

1. User logins into the application and clicks on the Filter houses tab.
2. System displays the Filter options as location, type of house, count of persons, cost and amenities.
3. User selects type of house filter option.
4. Systems asks the user to choose the required house type as per his/her convenience.
5. User chooses the 1-bedroom type of house as his/her choice.
6. System displays houses or flats belonging to 1-bedroom category.

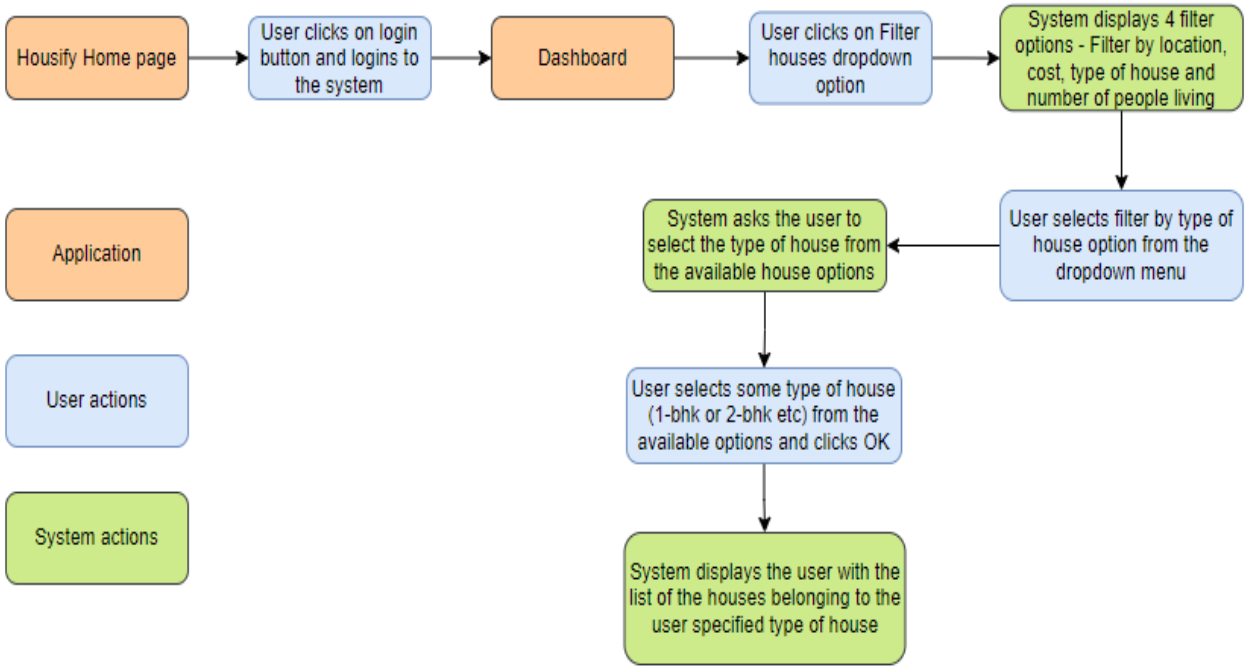

Figure 50: Task flow for Search house using filter option [1]

Task 3: Filtering the houses based upon the cost

Scenario:

John got the deals of some 1-bhk houses near Downtown, but now he wants to filter among those houses which are expecting less rent so that he can save his money for future expenses [1].

Use Case [1]:

1. User logins into the application and clicks on the Filter houses tab.
2. System displays the Filter options as location, type of house, count of persons, cost and amenities.
3. User selects filter by cost option.
4. Systems asks the user to choose the approximate cost range for the rent.
5. User specifies the cost to be a minimal one and clicks OK.
6. System displays houses or flats having rent approximately equal to the low cost chosen by the user.

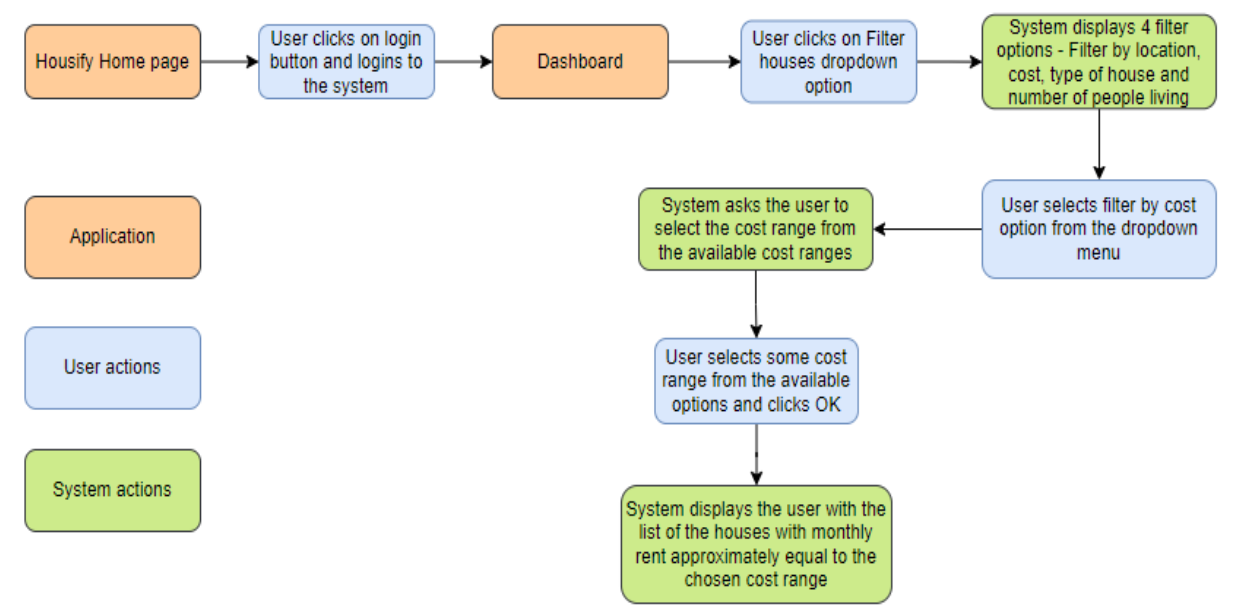

Figure 51: Task flow Housify filtering using filter by cost [1]

Task 4: Filtering the houses based upon number of persons willing to live in a house.

Scenario:

Thomas is arriving in Halifax in a week for his onsite project and looking for accommodation in Halifax. He wants to share the house with one other person so that the rent can be shared among both equally, thereby reducing the entire rent burden on him [1].

Use Case [1]:

1. User logins into the application and clicks on the Filter houses tab.
2. System displays the Filter options as location, type of house, count of persons, cost and amenities.
3. User selects filter by number of persons sharing the house option.
4. Systems asks the user to choose the approximate count of people he wants to live in the house.
5. User specifies the number of people to live as 2 and clicks OK.
6. System displays houses or flats in which 2 people can live.

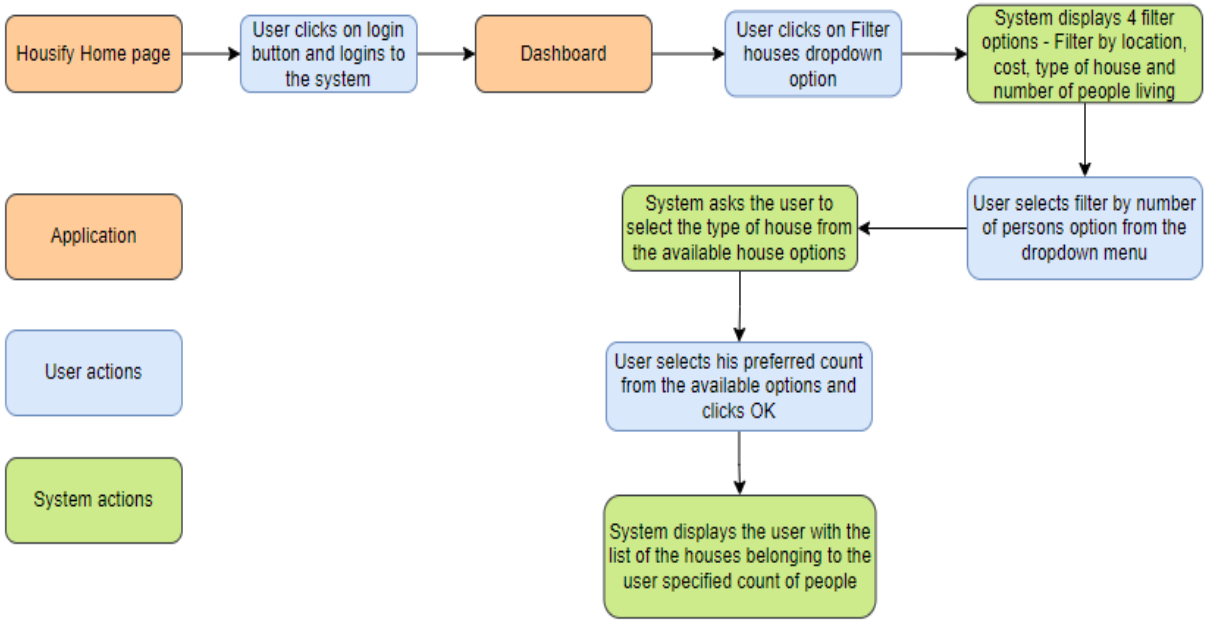

Figure 52: Task flow House filtering using filter by count [1]

Feature 4: Application Dashboard

Task 1: User apply for rent [1]

1. User opens the web application and logs into it.
2. Once logged in, user will be able to see his dashboard.
3. User selects Application Request button.
4. System redirects user to Application Request page.
5. User selects to apply for rent.
6. System redirects to page to fill details (full name, email, current address etc.).
7. User fills out all details
   7.1.User clicks on submit after filling details
   7.2.System sends this request as a notification to owner
   7.3.If user clicks on submit without filling details.
   7.4.System will prompt to fill out all details before submitting.
   7.5.If user clicks on cancel
   7.6.System will redirect to Application Request page.
8. Once submitted request, system will send this request as notification to owner and redirect to Application Request page.

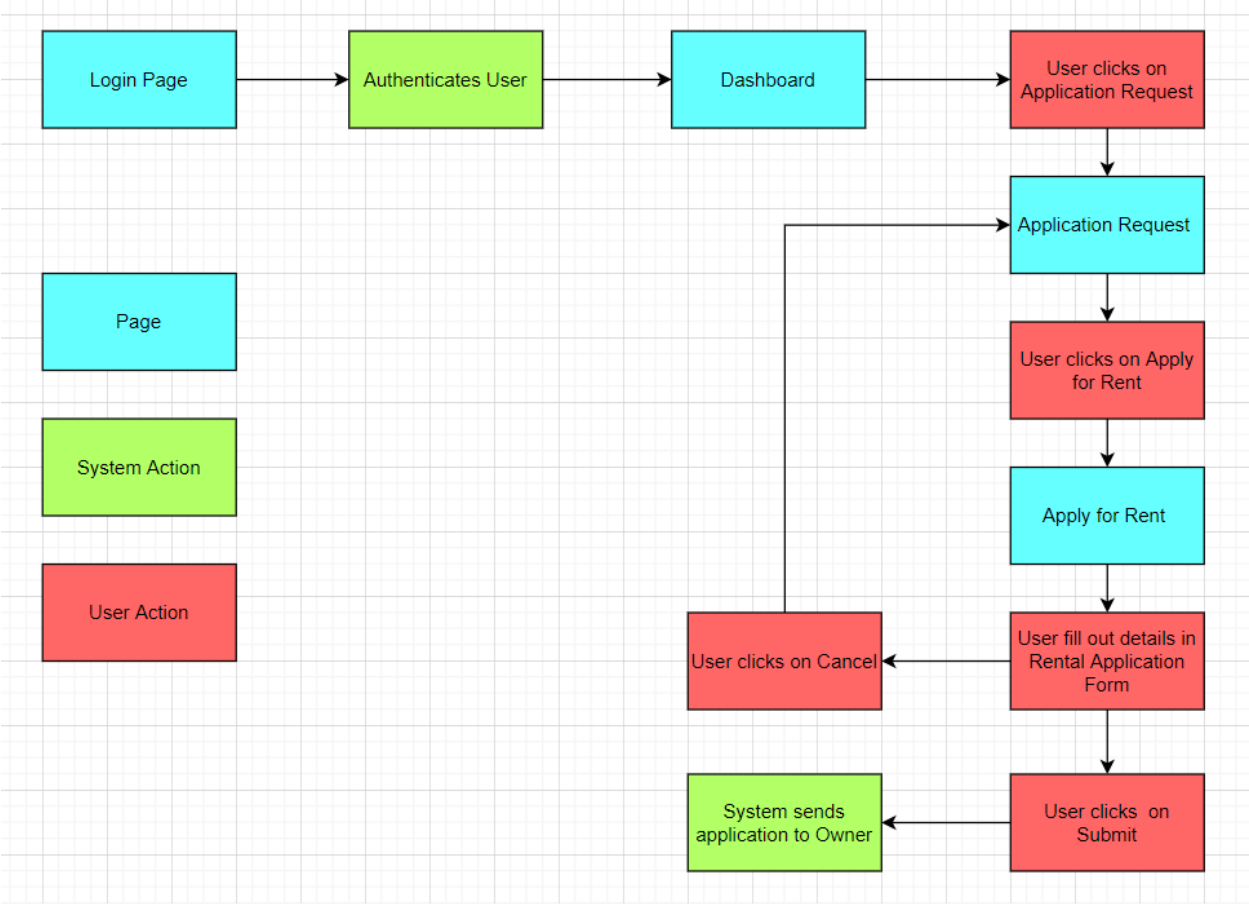

Figure 53: Task flow for Application Request [1]

Task 2: User wants to view application status [1]

1. User opens the web application and logs into it.
2. Once logged in, user will be able to see his dashboard.
3. User selects Application Request page.
4. System redirects user to Application Request page.
5. User selects to view status.
6. System redirects to page where list of applications with status will be shown.
7. Once user go through the applications with status, he/she clicks on back button
8. System redirects to Application Request page.

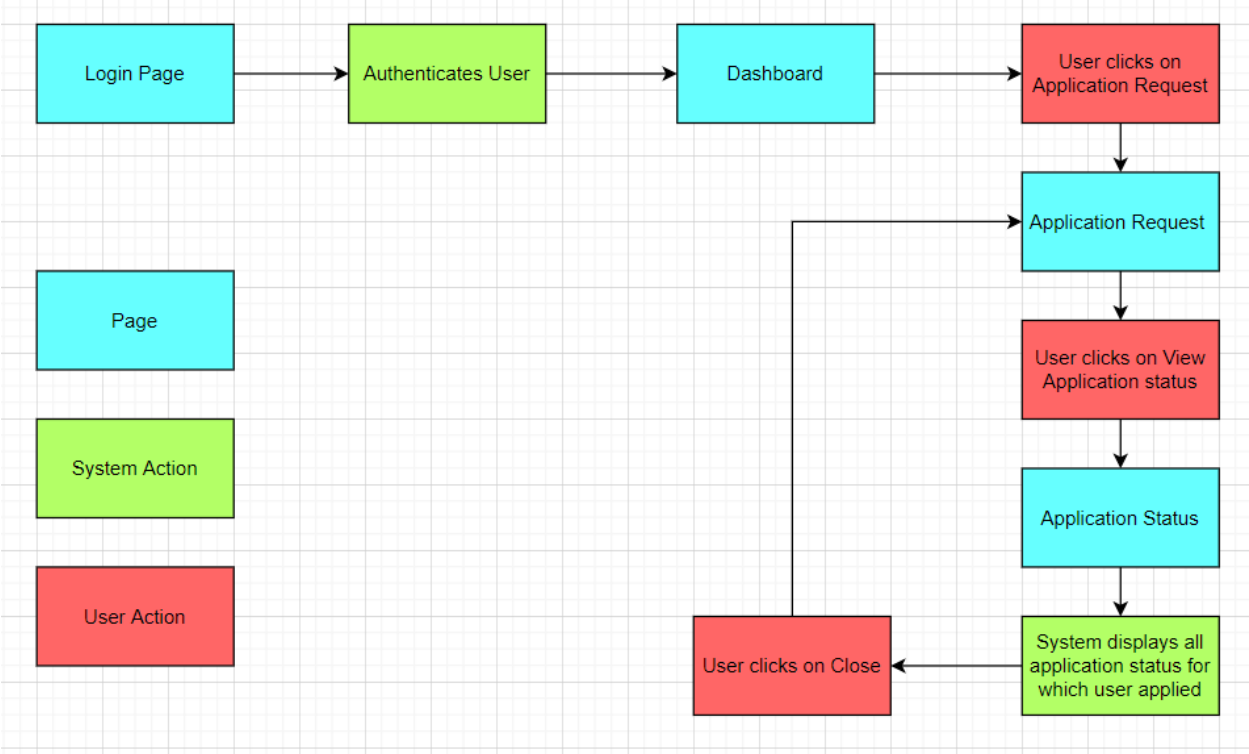

Figure 54: Task flow to view application status [1]

Task 3: User wants to view all applications [1]

1. User opens the web application and logs into it.
2. Once logged in, user will be able to see his dashboard.
3. User selects Application Request page.
4. System redirects user to Application Request page.
5. User selects to Applications button.
6. System redirects to page where list of applications for which user applied will be shown.
7. Once user go through all applications, he/she clicks on back button
8. System redirects to Application Request page.

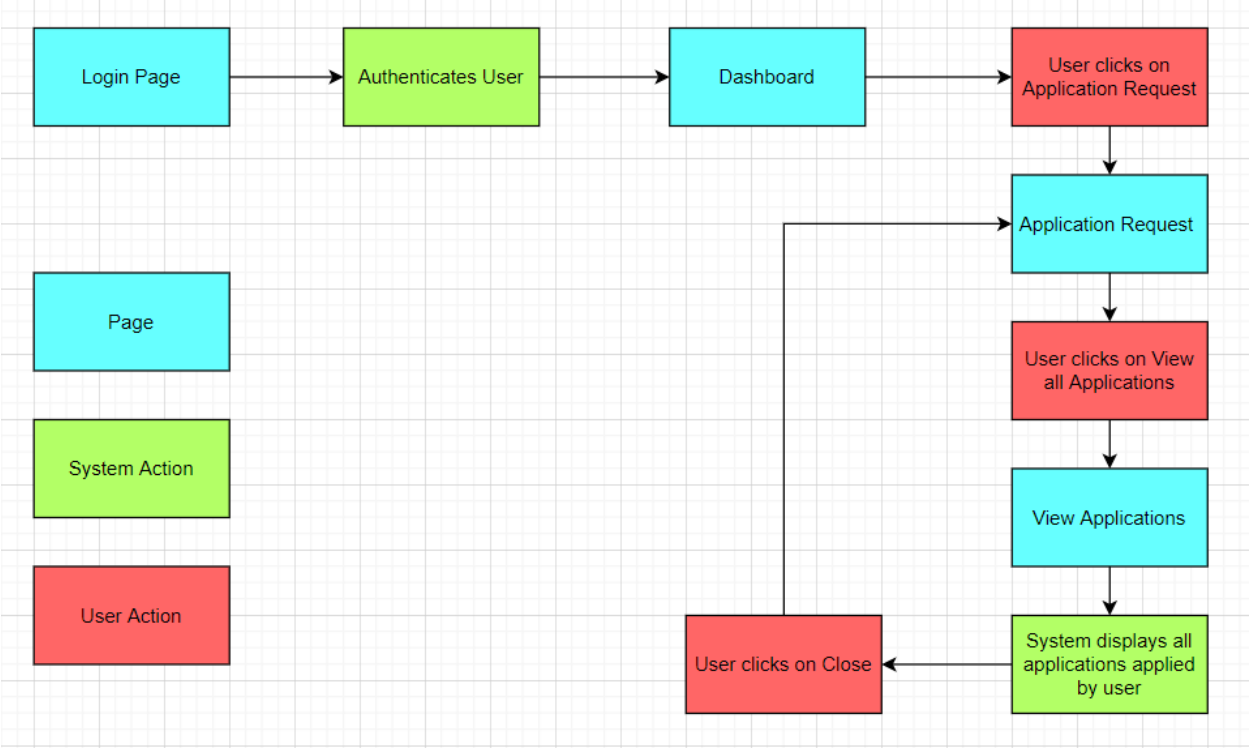

Figure 55: Task flow to view all applications [1]

Feature 5: Discussion Forum

Task 1: Creating a thread on the Discussion Forum [1]

Scenario:

A user, who is searching for houses and going through the list. After various listings he might find somethings confusing, it might raise few queries or just curiosity of a user. So, there is a discussion forum page in which queries like this will get resolved [1].

Use Case: Create thread (as a user) [1]

1. User visits ‘Housify’ web-app
2. System display homepage/housing property posting
3. User clicks on Discussion page
4. System displays already create threads and a button where you can create a thread
5. User will start creating a thread
6. System displays a form where user can choose a title and description
7. User inserts the details in the form
8. System posts the thread to the main discussion page
9. A thread is created, and other user can interact with by viewing and commenting on the thread.

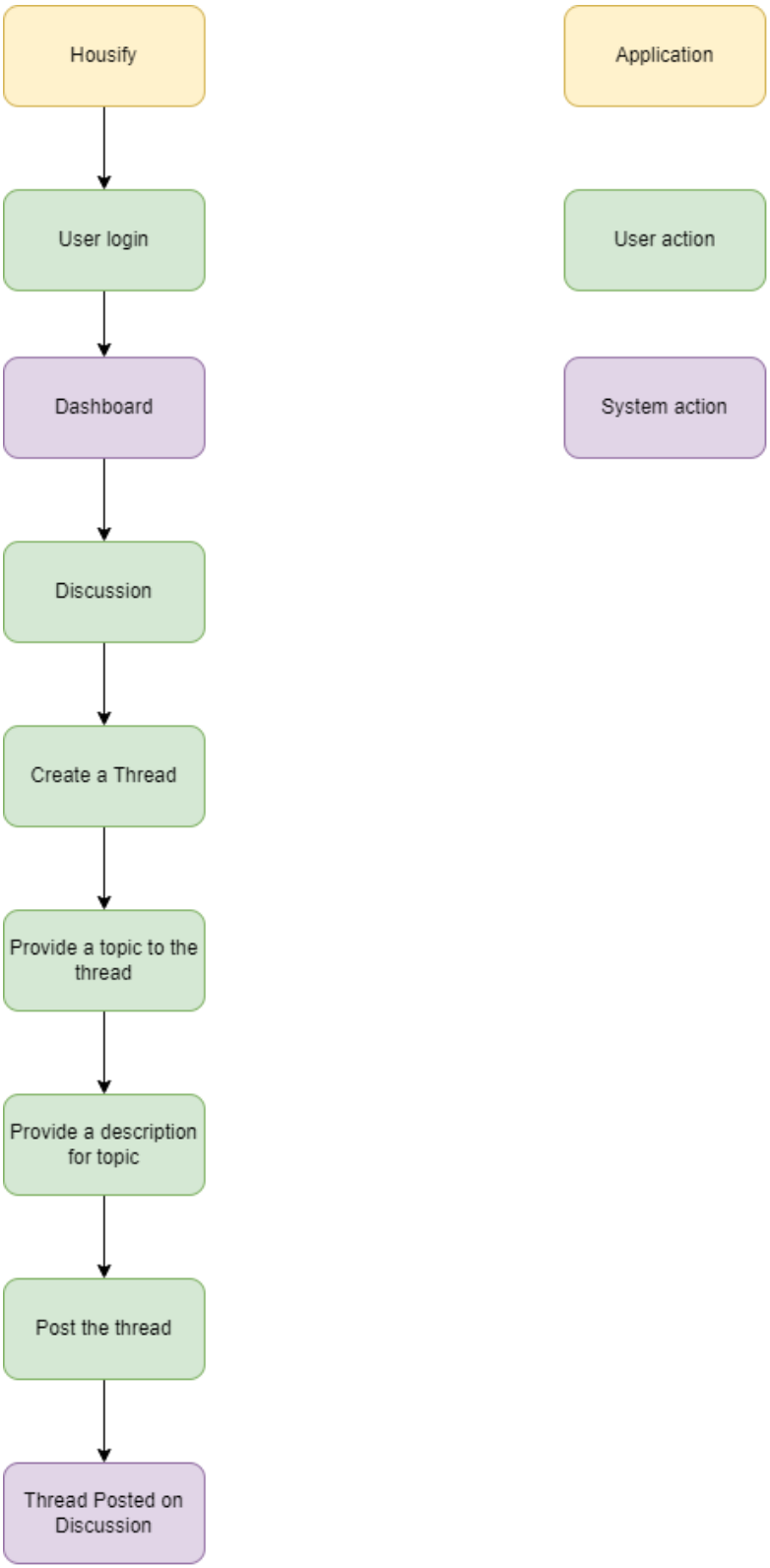

Figure 56: Task flow for creating thread [1]

Task 2: Commenting on a thread [1]

Scenario:

John is arriving in Halifax in a week. He is looking for a place to rent. He found a good house that fits most of his needs except the general vibe of the environment and close amenities near them. He sees there is already a discussion going on for this housing area. He asks his specific question about the area [1].

Use cases [1]:

1. User visits ‘Housify’ web-app
2. System display homepage/housing property posting
3. User clicks on Discussion page
4. System displays already create threads and a button where you can create a thread
5. User searches for the thread
6. System displays a result for the searches
7. User interacts with the most appropriate thread that he likes
8. System opens all the comments already on the threads.
9. User inserts his comments on the thread asking their queries or general comment.
10. A comment is created and now it is displayed under that specific thread.

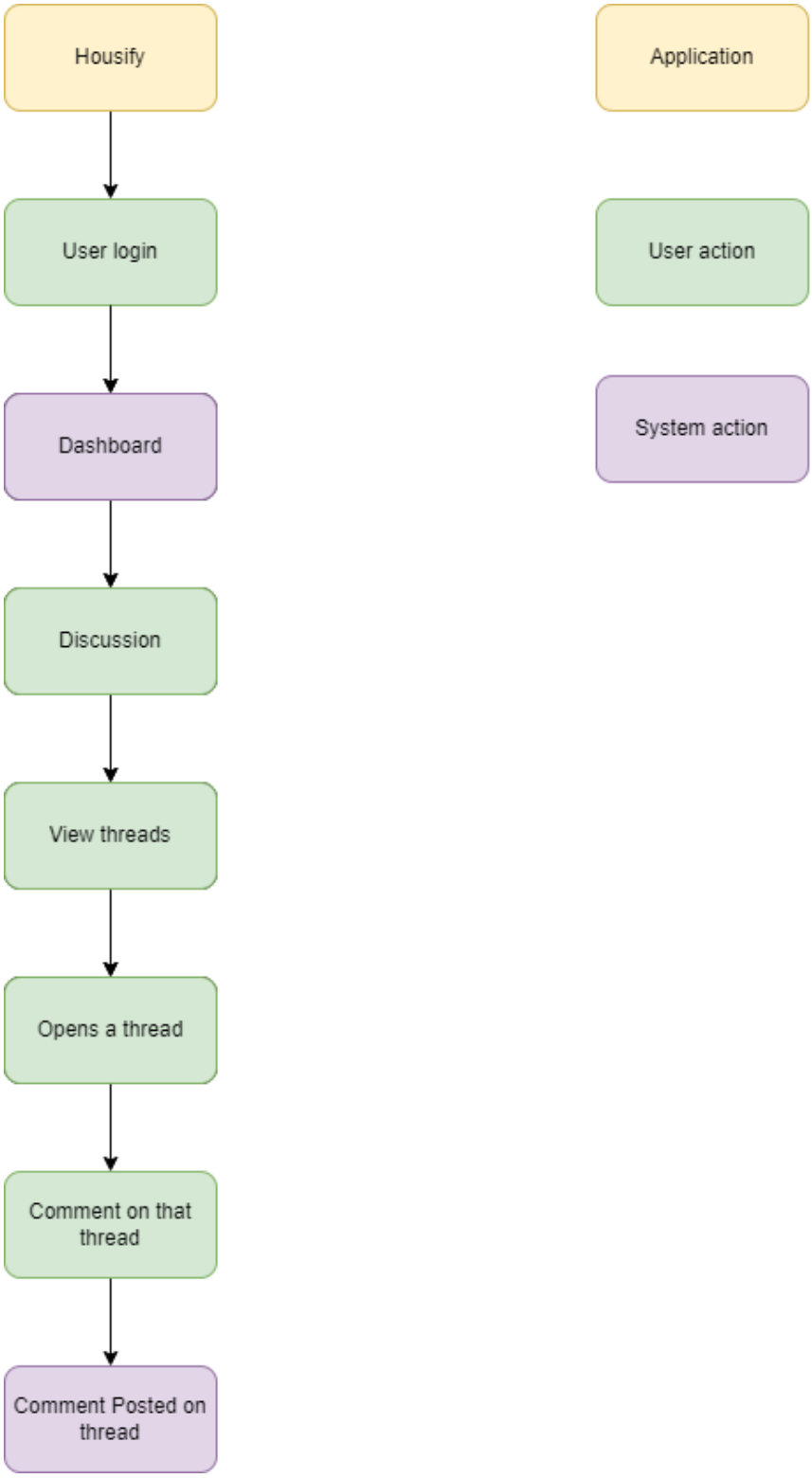

Figure 57: Task flow for commenting on thread [1]

Feature 6: Manage Rentals/Houses

Assuming **Renter** already a registered user and logged into the system [1].

Task 1: Create New Post/Ad for house [1].

1. System displays dashboard of the logged-in user.
2. User selects Manage Rentals/Houses button in the navigation bar.
3. System displays all the house advertisements/posts that the **renter** has previously posted in the form of grid.
4. User clicks on the "Add new" button for creating a new post/ad for the house he wants to rent.
5. System displays the "Create Ad/Post" form for the **renter** to fill.
6. **Renter** fills out the form fields i.e., house title, description, image, address, category, contact details and submits the form.
7. System validates all the fields with corresponding its corresponding validations, for example if any field is empty which is required or contact number is should be 10 digits.
   7.1 System displays "The required field is empty" message in case of former validation and "Contact should be 10 digits long" message in case of latter validation.
   7.2 System prompts the user to try again.
      7.2.1 The user fills out the fields with validations and requirements and submits the form.
8. System displays "Advertisement/Post created successfully" after all the validations and saving the ad/post into database and redirects to Manage Rentals/Houses page.
9. **Renter** sees their new ad/post of the house.

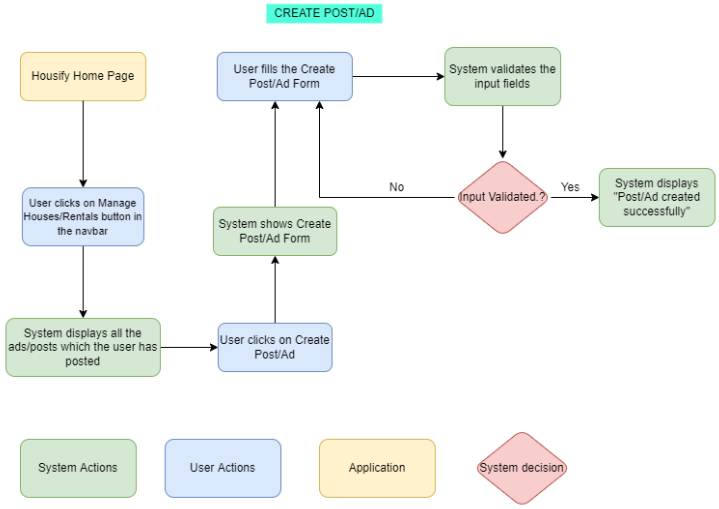

Figure 58: Create Post/Ad for the house – Task flow [1]

Task 2: Update an existing Post/Ad of house [1].

1. System displays dashboard of the logged-in user.
2. User selects Manage Rentals/Houses button in the navigation bar.
3. System displays all the house advertisements/posts that the **renter** has previously posted in the form of grid.
4. User clicks on the "Edit" button in the action column of the grid for editing a existing post/ad of the house.
5. System displays the "Edit Ad/Post" form for the **renter** to edit.
6. **Renter** updates the form fields i.e., house title, description, image, address, category, contact details and submits the form.
7. System validates all the fields with its corresponding validations, for example if any field is empty which is required or contact number is should be 10 digits and many more.
   - 7.1 System displays "The required field is empty" message in case of former validation and "Contact should be 10 digits long" message in case of latter validation.
   - 7.2 System prompts the user to try again.
     - 7.2.1 The user fills out the fields with validations and requirements and submits the form.
8. System displays "Advertisement/Post updated successfully" after all the validations and updating the ad/post into database and redirects to Manage Rentals/Houses page.
9. **Renter** sees their updated ad/post of the house.

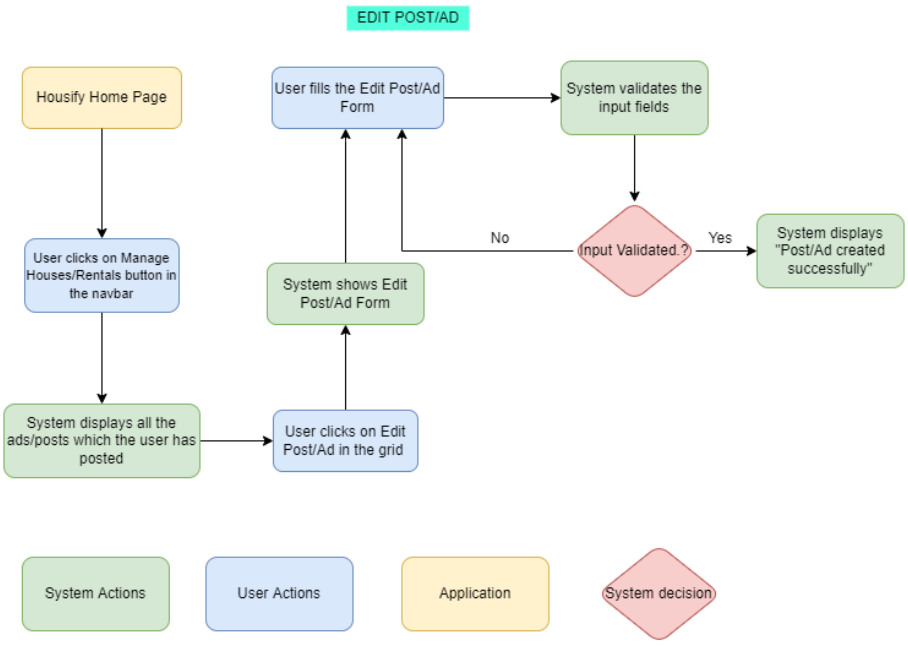

Figure 59: Update post/ad of the house – Task flow [1]

Task 3: Delete an existing Post/Ad of house [1] .

1. System displays dashboard of the logged-in user.
2. User selects Manage Rentals/Houses button in the navigation bar.
3. System displays all the house advertisements/posts that the renter has previously posted in the form of grid.
4. User clicks on the “Delete” button in the action column of the grid for deleting an existing post/ad of the house.
5. System displays the prompt box for the renter confirmation to delete the ad/post.
   5.1 Renter clicks on “Cancel” button of the prompt box.
      5.1.1 System displays the Manage Rentals/Houses page after removing the prompt box.
   5.2 Renter clicks on “Confirm” button of the prompt box,
6. System deletes the corresponding post/ad of house.
7. System displays “Post/Ad deleted successfully” message.
   7.1 System displays “The required field is empty” message in case of former validation and “Contact should be 10 digits long” message in case of latter validation.
   7.2 System prompts the user to try again.
      7.2.1 The user fills out the fields with validations and requirements and submits the form.
8. System displays “Advertisement/Post updated successfully” after all the validations and updating the ad/post into database and redirects to Manage Rentals/Houses page.
9. Renter sees their updated ad/post of the house.

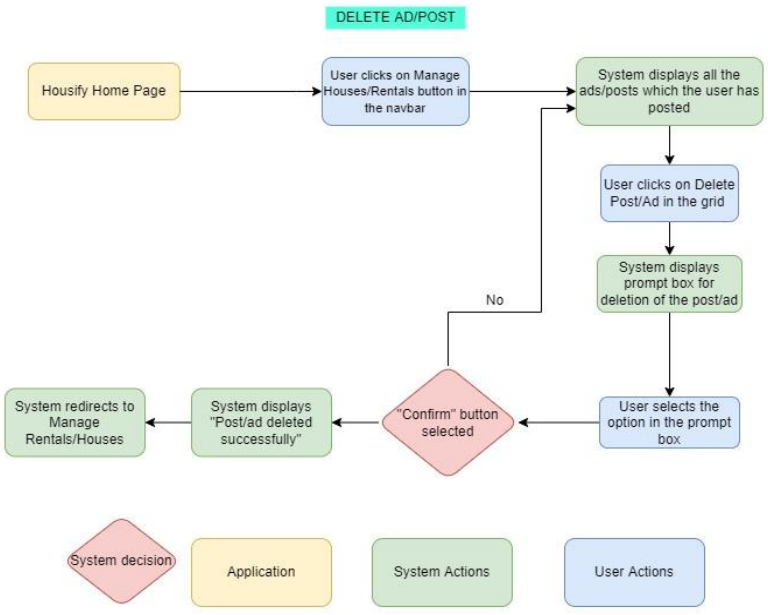

Figure 60: Delete an ad/post of the house – Task flow [1]

Task 4: View an existing Post/Ad of house [1].

1. System displays dashboard of the logged-in user.
2. User selects Manage Rentals/Houses button in the navigation bar.
3. System displays all the house advertisements/posts that the **renter** has previously posted in the form of grid.
4. User clicks on the "View" button in the action column of the grid for viewing an existing post/ad of the house.
5. System displays the selected ad/post of the house with all the details.

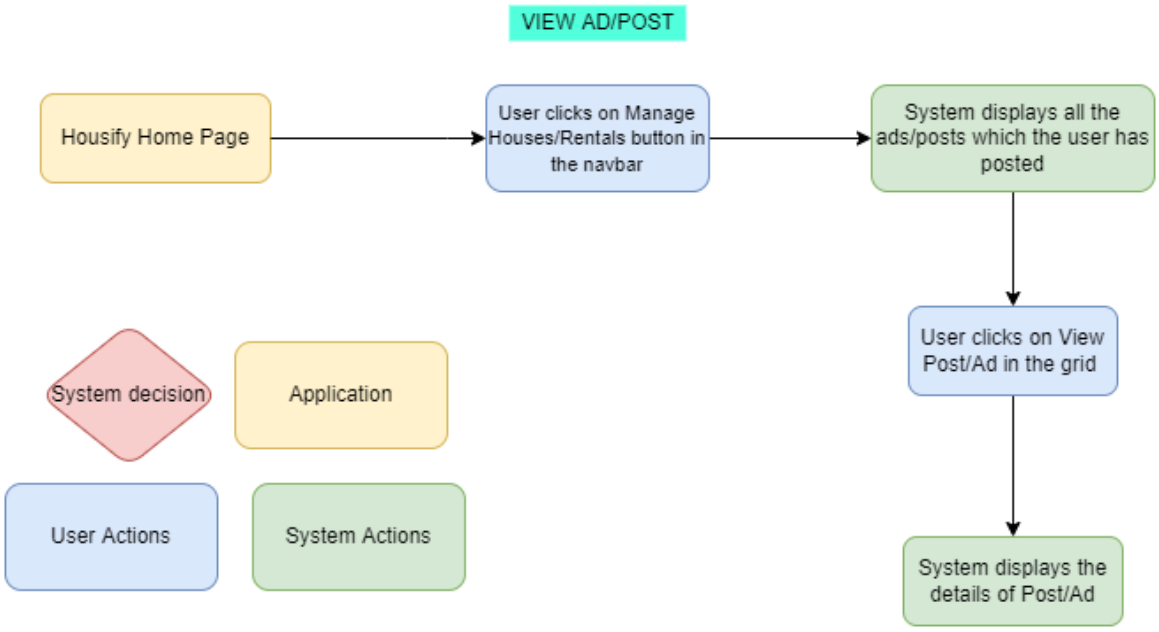

Figure 61: View an ad/post of the house – Task flow [1]

Feature 7: Housing analysis

Task 1: Add Review [1]

1. System displays dashboard of the logged-in user.
2. User selects Reviews button in the menu.
3. System displays all the houses that the user has previously lived in andhis reviews.
4. User clicks on the add review button for the house he wants provide review.
5. System displays the review form for the user to fill.
6. User fills out the form fields i.e., name, mobile, select rating, and provide feedback and submits the review.
7. System validates all the fields such as if any field is empty which is required.

    7.1 System displays “The required field is empty” message.

    7.2 System prompts the user to try again.

        7.2.1 The user fills out the fields with validations and requirements and submits the review.

8. System displays “Review added successfully” after all the validations and saving the review into database.
9. User sees their review of the house.

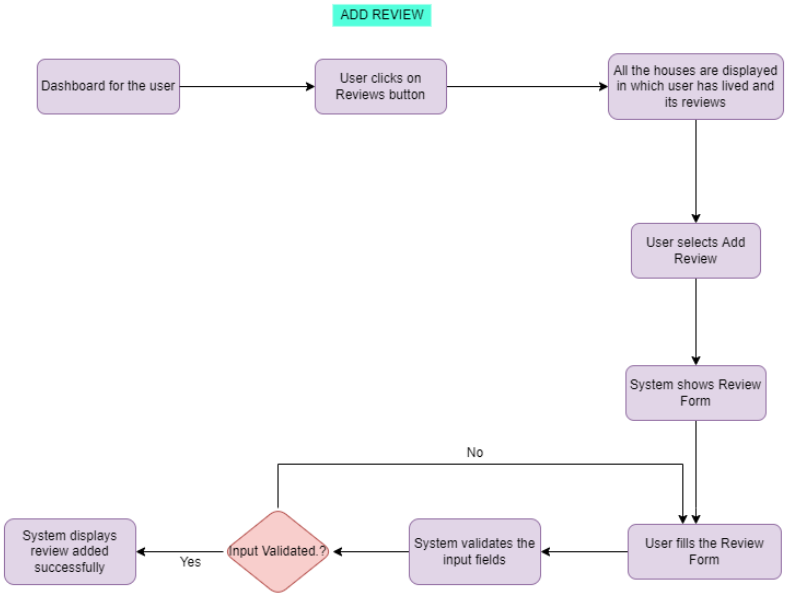

Figure 62: Task flow for Adding Review [1]

Task 2: Edit Review [1]

1. System displays dashboard of the logged-in user.
2. User selects Reviews button in the menu.
3. System displays all the houses that the user has previously lived in andhis reviews.
4. User clicks on the edit review button for the house he wants to edit thereview he provided.
5. System displays the review form for the user to edit. The review form contains the current review rating, and feedback from the user.
6. User updates the review form fields according to the need.
7. System validates all the fields such as if any field is empty which is required and many more.
    7.1 System displays “The required field is empty” message.
    7.2 System prompts the user to try again.
        7.2.1 The user fills out the fields with validations and requirements and submits the review.
8. System displays “Review updated successfully” after all the validationsand updating the review into database.
9. User sees their updated review of the house.

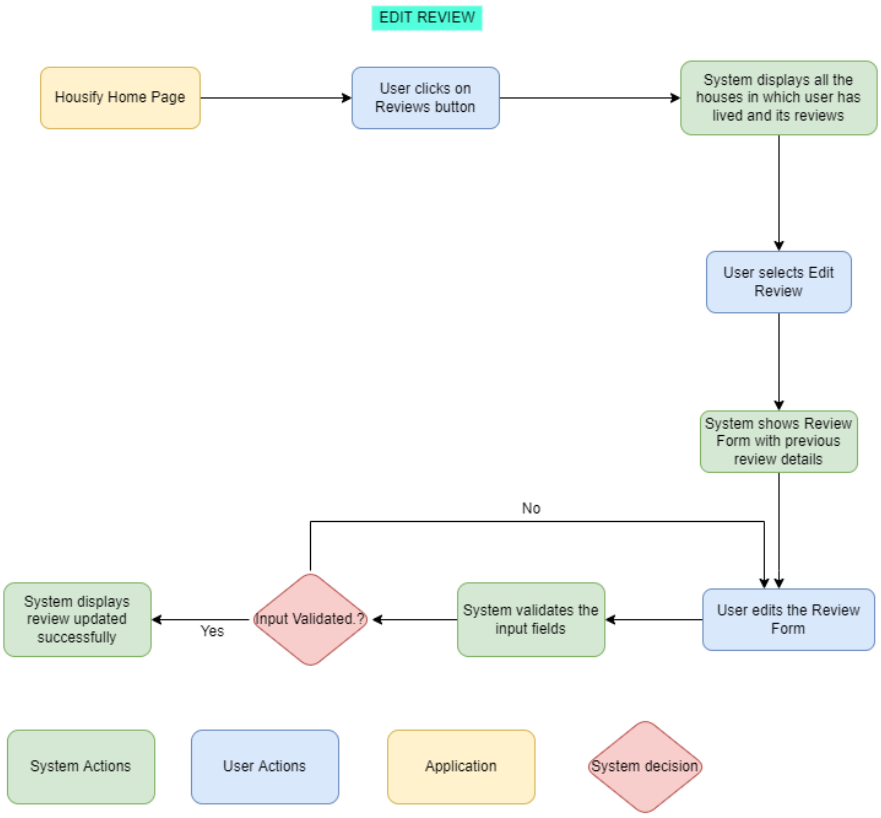

Figure 63: Task flow for Editing Review [1]

4.2 Process and Service Workflow

This figure represents that the application architecture of “Housify” is using model view controller design pattern where the model handle operations with MongoDB database, all business logic is handled in the controller, and the view represents the front-end part which is the user interaction part [3].

React.js will handle the front-end part including bootstrap, CSS, javascript while for backend development node.js, express.js frameworks are to be used. For storing purposes MongoDB database is going to be used due to its flexibility in fetching and storing data [3].

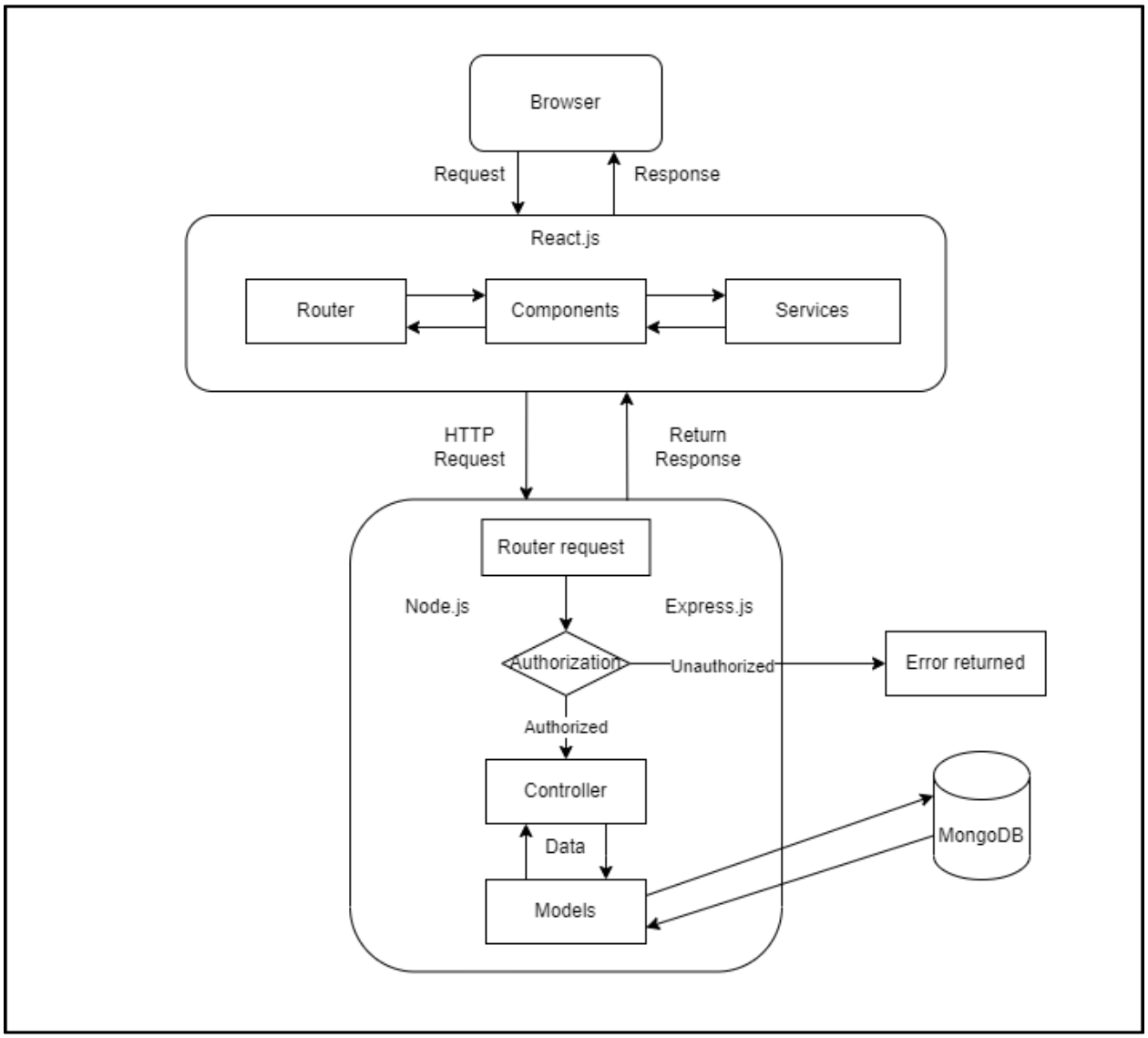

Figure 64: Application Architecture [3]

From the below figure, it can be explained how the workflow of the “Housify” web application is. User request from the browser is interacted with using react components which are then called using REST API services that interact with node.js. Then, Node.js handles a request from MongoDB to fetch and store data into the database [3].

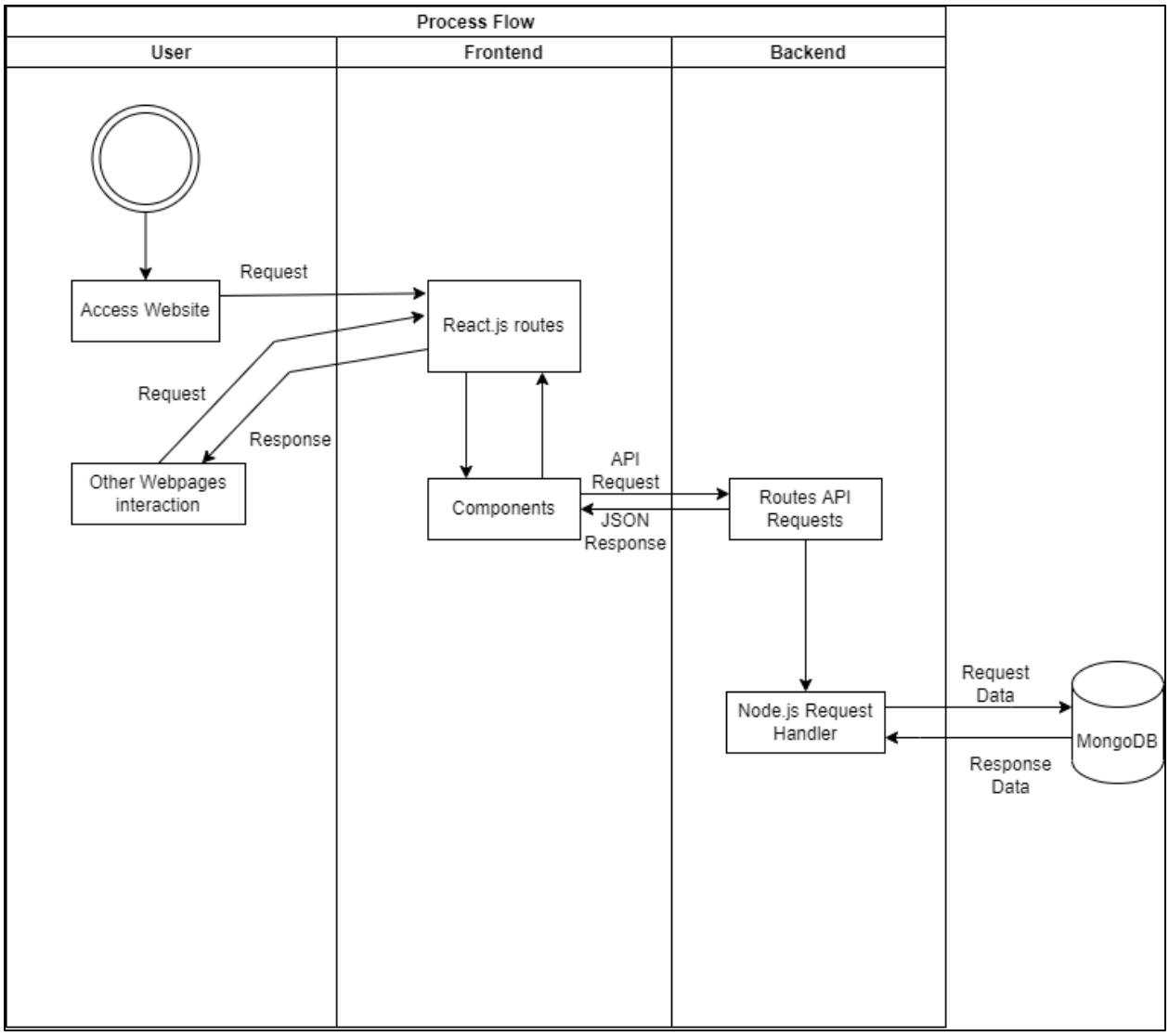

Figure 65: Process Flow [3]

5. FUTURE WORK

We would be working on remaining features which will give user's better experience. Till now we have completed 70 % of our work (which includes 7 features proposed) [1].

1. **Rental House Service and maintenance [1]:**
   The user will be able to generate service requests that will be sent to the owner, as well as view any requests that have been submitted and their status [1]

2. **Notification Management [1]:**
   The users will get notified when other users are interested in their posts. As this application focuses on renter and tenant, we are creating a gateway where the renter will see who is interested in their housing post. They can further connect through various mediums such as calls and email [1].

3. **Meet with us [1]:**

   After successful login, a user will be provided with an opportunity to schedule an appointment with the tenant to clarify his/her doubts regarding the house or to proceed with further process if decided to move into the house belonging to the tenant [1].

6. CONCLUSION

Our website has a simple user interface for all users to use easily. Users start by creating an account on the website and then can look for housing options on the home page. The user can also post an ad if they have space empty to rent or sub-lease the room or apartment. A filtration search option is also available for the user to search for a particular request or requirement that they might need or have for the tenant. One major challenge is payment when renting a house, and our website also mitigates that. Analysis of a particular location and housing is as necessary as living there because beforehand, the user has an idea of the thoughts of the tenants residing. Furthermore, there is a discussion forum where the users can start a thread about any particular topic related to housing and region. In those threads, users can reply with their opinions. Notification is a future development plan where we will notify the users when there be reactions to their post, ad or even thread. Moreover, housing service and maintenance is also part of the plans where we will add an option where current users can raise a ticket about anything related to their rented house. For meeting with us, users can schedule a meeting with each other and clarify their doubts and questions. Finally, our website is simple, user-friendly and fulfil an essential need.

7. RECOMMENDATIONS

For future developments and support from technical point of view are, it is recommended that the rest conventions be strictly followed. We recommend setting up database connections pools for the backend to improve application latency. For better availability for the project, we also recommend switching to a different cloud provider, GCP. In AWS, get a proper dedicated server to host the website. Cloud platforms have many other servers that can make the application even better. From feature point of view, still user interaction and user experience could be enhanced and make better from user point of view, however, to take advantage of full potential of the project it is recommended to have more features such as sharing the property and some built in management of the property rentals.

8. REFERENCES

[1] D. Oza, K. Mistry, R. Domadiya, S. Konda and V. Naik, "Project Proposal", Dal.brightspace.com, 2022. [Online]. Available: https://dal.brightspace.com/d2l/lms/dropbox/user/folders_history.d2l?db=140429&grpid=211879&isprv=0&bp=0&ou=203602. [Accessed: 09- Apr- 2022].

[2] D. Oza, K. Mistry, R. Domadiya, S. Konda and V. Naik, "Assignment 1", Dal.brightspace.com, 2022. [Online]. Available: https://dal.brightspace.com/d2l/lms/dropbox/user/folders_history.d2l?db=140431&grpid=211879&isprv=0&bp=0&ou=203602. [Accessed: 09- Apr- 2022].

[3] D. Oza, "Assignment 2", Dal.brightspace.com, 2022. [Online]. Available: https://dal.brightspace.com/d2l/lms/dropbox/user/folders_history.d2l?db=140434&grpid=0&isprv=0&bp=0&ou=203602. [Accessed: 09- Apr- 2022].

[4] D. Oza, K. Mistry, R. Domadiya, S. Konda and V. Naik, "Demo Video", Dal.brightspace.com, 2022. [Online]. Available: https://dal.brightspace.com/d2l/lms/dropbox/user/folders_history.d2l?db=140433&grpid=211879&isprv=0&bp=0&ou=203602. [Accessed: 09- Apr- 2022].

[5] "Flowchart Maker & Online Diagram Software", *App.diagrams.net*, 2022. [Online]. Available: https://app.diagrams.net/. [Accessed: 13- Apr- 2022].