Summer 2017

Home Away from Home

Santosh Kumar Ampolu
Governors State University

Devi Prathyusha Manchella
Governors State University

Divya Swarup Nadella
Governors State University

Prashanth Goud Yarlagadda
Governors State University

Follow this and additional works at: http://opus.govst.edu/capstones

Part of the Computer Sciences Commons

ABSTRACT

Home Away From Home is an e-Commerce application for short-term rentals of homes or apartments. This application presents a Web portal for property owners to advertise their properties and lease them to renters for short-term stays.

As the global sharing economy grows more owners of premium properties discover the benefit of sharing their furnished homes. Making use of your home as an asset, when travelling or working interstate or overseas for extended periods, makes perfect sense to savvy owners.

Maintaining your home and keeping it safe is very important to us and we make sure we leave nothing to chance. Before a check-in as well as after check-out, we are methodical in our preparation to ensure your home is perfectly presented for the guest as well as pristine when welcoming you home. We only list the best properties in the best spots. Knowing how special your property is, we make sure we always find you the perfect guests.

This Web application provides features and functionalities to meet the following requirements:
- 3 types of accounts: renter, property owner and administrator
- Renter can manage its setting and rental transactions.
- Properties owner can manage its properties, rental transactions, payment records, etc.
- Admin can manage all properties, all rental activities, and all accounts.
- Front page will present sections of homes under different themes.
- A search box allows users to find particular homes to their interests.
- Search result listings show hyperlinked thumbnails of properties.
- Full details of properties, availability, amenities are shown on the detail pages.
- The application enables bookings and payment transactions.
- User survey and comment will be available.
- The website will be mobile friendly and responsive to different screen resolutions.
- Use of Web mapping technology to show interactive map of properties.

Table of Content

1   *Project Description*

Home Away from Home or Rental Homes plays a vital role in every aspect. For example if we are planning to go outside where we don't have any home to live in then we can opt for this home rental service. Nowadays there are many websites which suggest home rentals but among them only few are better websites, and also show the best properties that are available. The quality of a website can be determined by viewing its reviews, the number of properties posted and the number of renters.

A Website is said to be well-maintained only when the website is frequently updated. As such, the website should be maintained properly like removing all the old and already rented properties, and the website should be updated with all the new properties and relevant information. In the Home Away from Home portal, the Admin handles all the activities of the website. Home Away from Home website is designed using the homes rental website as examples, which contain all of the information regarding properties, owners and renters. This Home Away from Home website is used by owners to post properties. All the information about properties, owners and renters are stored in a database. The Home Away from Home website is kept up to date for all daily activities. All these activities are handled by the Admin. Once the owner uploads a property, then the renter will be able to view the property online.

The Admin has all the rights to add, delete & update the owner and renter accounts at any time. The owners have the right to edit the property that was added by him/her. The Admin has the right to view all the owner's and renter's details who registered into the website and also keep

their details up-to-date. Based on the requirement of the renters can search all the available properties.

1.1   Competitive Information

A website or a project or a business has a competitor. A website may be an extension to another website or it can be an alternative to the existing website. For a website to be successful we have to follow a few steps which are developed by a criteria. They are:

- ➢ Gathering of Information.
- ➢ **Through Research**: We have to do a research on all the possible rental websites and collect the information about those websites.
- ➢ **Study Competitors**: Once the information is gathered a detailed study must be performed.
- ➢ Determining and creation of the website in a way it is a competitor to other websites.

1.1.1 Relationship to Other Websites/Applications

Home Away from Home website is identical to RentalHomes.com and Rent.com websites. Difference between these two websites is the way how renter can view the details of property and the owner. These websites portray the properties and a renter can rent it. Owner has to register himself and then he can add the property.

1.2   Dependencies and Assumptions.

- ➢ We made an assumption that for every property that is added into the website has an owner.
- ➢ The details of both the owner and the renter are genuine and valid.

1.3   Future Enhancements

In the present model the gallery is known to be static and shows only some of the sample Images. In future we will create a website where it shows the images of all the newly added Properties in the Gallery.

1.5   Definitions and Acronyms

Admin:

Admin has the right to view, delete, update and also make the necessary changes to the owner, renter and property data.

Owner:

Owner uploads a property for rent. An owner can add as many properties as he can and also delete the property . At first owner needs to register him/herself into the website. After that owner need to login and then he will be able to add the property. Owner can view all his/her personal details and can also update the details.

Renter:

Renter is the one who rents a property. To rent a property renter need to register him/herself into the website. After that they need to login and can rent the available property that they are looking for.

Payment:

Once after the renter books for a property he/she need to pay for that appartment/home. And this payment details can view by renter, owner, and also admin. This is one way of confirmation that the property has been booked or already rented.

Registration:

To login into the website both the owner and the renter need to register and then they can Login into the website. To Register into the website they need to provide all the required details and create a password. After that they can easily login into the website.

UML   – Unified Modeling Language

HTML- Hyper Text Markup Language

SQL   - Structured Query Language

MVC - Model View Control

2.   *Project Technical Description*

Plan of the site pages and required UI is performed utilizing bootstrap classes and HTML5. At first the required page configuration is finished and afterward the entire database is developed with tables and required put away methods. After that utilizing C#.NET code is composed to associate with database and play out the required operations as for this site. All these are done in an arrangement. We have utilized shape outline for this venture.

2.1   *Application Architecture*

Relations:

Registration – Payment (Many to One)

All the registered owners and renters can view the payment details.

Registration – Property (One to Many)

Once you register into the website you can add many properties.

Admin – Property (One to Many)

Admin can delete the properties.

Admin – Registration (One to Many)

Admin can view all the details of the registered owners and renters.

 Registration – Feedback (One to Many)

All the registered owners can view the feedback.
### 2.2   *Application Information Flows*

The application flow starts at homepage. User can enter the website in three different roles

Admin: Admin can access all the properties, renters and owners. All the data is stored is stored in database.

Owner: Owner can upload, edit or delete property, review the payment details and feedback from renter.

Renter: Renter can search the property and selects the property they want, review the payment details and feedback given to the owner.

Data Flow Diagram

5
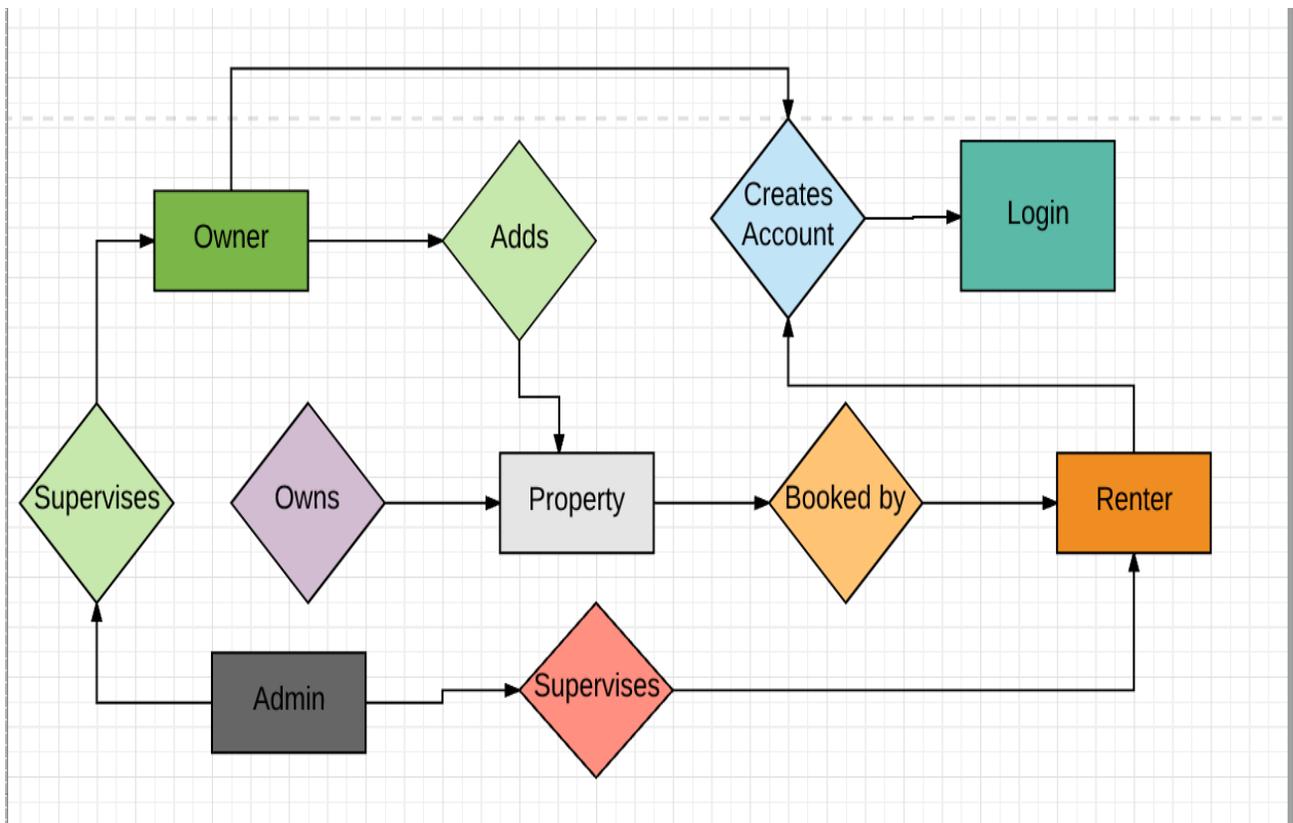
2.3 Interactions with other Projects (if Any)

There are no interactions with any projects.

2.4 Interactions with Other Applications

Rental Homes is one of the websites quite similar to Home away from Home. The only difference between these two websites is that in rental homes we need to provide both check-in and check-out details to search for a home or apartment whereas in Home away from Home we need to just provide the basic details like area, apartment type. Then you can automatically find all the apartments that are available and inorder to book that you need to login into the website.

2.5 Capabilities

Capabilities of Admin:
- ➢ add/delete/update the owner information
- ➢ add/delete/update the renter information
- ➢ delete the property
- ➢ view all the feedbacks.

Capabilities of Owner:
- ➢ add/delete/update the property information
- ➢ update own profile
- ➢ view the payment details of renter.
- ➢ view the feedback given by the renter.

Capabilities of Renter:
- ➢ update own profile
- ➢ view the payment details.

2.6   Risk Assessment and Management

Risk: The principle risk is renter should check in the event that he has given a working email id as the confirmation of property booked will be sent to the email.

Mitigation: In event that on the off chance that the renter gives a wrong email id they can transform it by updating their profile**.**

Risk: Technical hardware failure of server

Mitigation: 24/7 server maintenance

3. Project Requirements

3.1 Identification of Requirements

<GSU-HomeAwayFromHomePortal_SU-2017 Payment- 01 >

By using this requirement allows us to add the personal details of a customer by giving the attributes owner_id, property_id, renter_id, card_type, amount, name, card_number, cvv & expiry. this functionality in database can add the details to the database and we can retrieve at any point of time.

<GSU-HomeAwayFromHomePortal_SU-2017 Payment- 02 >

By using this requirement allow us to delete the details of a customer by giving the attributes owner_id or property_id or renter_id. this functionality in database can delete the details from the database and we can retrieve at any point of time.

<GSU-HomeAwayFromHomePortal_SU-2017 Payment- 03 >

By using this requirement allow us to update the payment details of a customer by giving the attributes owner_id, property_id, renter_id, card_type, amount, name, card_number, cvv &

expiry. this functionality in database can update the details to the database and we can retrieve at any point of time.

<GSU-HomeawayFromHomePortal_SU-2017 Payment- 04 >

By using this requirement allows us to change the customer card details by giving the attributes card_type, card_number, cvv, and expiry. this functionality in database can change the user data in database and we can retrieve at any point of time.

<GSU- HomeawayFromHomePortal_SU-2017 Payment - 05 >

By using this requirement allows us to track status of payment by giving the attributes property_id, renter_id. this functionality in database that track the details in database and we can retrieve at any point of time.

<GSU- HomeAwayFromHomePortal_SU-2017 Property- 06 >

By using this requirement allow us to add the property details to the renters by giving the attributes id, owner_id, property_type, name, room_type, area, cost, description, address. this functionality in database will store the details of the data and we can retrieve the data at any point of time.

<GSU- HomeAwayFromHomePortal_SU-2017 Property- 07 >

By using this requirement allow us to delete the client details by giving the attributes id or owner_id. this functionality in database will delete the stored data and we can retrieve the data at any point of time.

<GSU- HomeAwayFromHomePortal_SU-2017 Property- 08 >

By using this requirement allow us to update the property details by property owner to the customer by giving the attributes id, owner_id, property_type, name, room_type, area, cost, description, address. this functionality in database will update the data and store the data and retrieve the data at any point of time.

<GSU- HomeAwayFromHomePortal_SU-2017 Property- 09 >

By using this requirement allow us to find the property is available or not by giving the attributes is id and owner_id. this functionality in database that can store the details of the data and we can retrieve the data at any point of time.

<GSU- HomeAwayFromHomePortal_SU-2017 Property- 10 >

By using this requirement allow us to find the property owner details by giving the attributes property_type, room_type, and area. this functionality in database will store the details of the data and we can retrieve the data at any point of time.

<GSU- HomeAwayFromHomePortal_SU-2017 Registration- 11 >

By using this requirement allow us to add the details of the customer by giving the attributes id, first_name, last_name, email_id, password, mobile, address, type. this functionality in database will add the details into database and we can retrieve at any point of time.

<GSU- HomeAwayFromHomePortal_SU-2017 Registration - 12 >

By using this requirement allow us to delete the details of the customer by giving the attribute id. this functionality in database will delete the stored details in database and we can retrieve at any point of time.

\<GSU- HomeAwayFromHomePortal_SU-2017 Registration - 13 \>

By using this requirement allow us to update the details of the customer by giving the attributes id, first_name, last_name, email_id, password, mobile, address, type. this functionality in database will update the new details into database and we can retrieve at any point of time.

\<GSU- HomeAwayFromHomePortal_SU-2017 Registration - 14 \>

By using this requirement allow us to track the status of the customer by giving the attributes id. this functionality in database will track the status details into database and we can retrieve at any point of time.

\<GSU- HomeAwayFromHomePortal_SU-2017 Registration - 15 \>

By using this requirement allow us to update the details of the notes by giving the attribute id. this functionality in database can add the details into database and we can retrieve at any point of time.

\<GSU- HomeAwayFromHomePortal_SU-2017 Registration - 16 \>

By using this requirement allow us to find the details of the customer who registered either as owner or renter by giving the attribute id. this functionality in database that can track the details into database and we can retrieve at any point of time.

\<GSU- HomeAwayFromHomePortal_SU-2017 Feedback - 17 \>

By using this requirement allow us to add the details of feedback by giving the attributes user_name, feedback. this functionality in database will add the details of the feedback in database and we can retrieve at any point of time.

\<GSU- HomeAwayFromHomePortal_SU-2017 Feedback - 18 \>

By using this requirement allow us to delete the details of feedback by giving the attributes user_name. this functionality in database that delete the data in database and we can retrieve at any point of time.

\<GSU- HomeAwayFromHomePortal_SU-2017 Feedback - 19 \>

By using this requirement allow us to update the details of proposal by giving the attributes user_name, feedback. this functionality in database will update the details of new data in database and we can retrieve at any point of time

\<GSU- HomeAwayFromHomePortal_SU-2017 Feedback - 20 \>

By using this requirement allow us to check the feedback all the renters of project by giving the attribute user_name. this functionality in database will check store the data in detail and we can retrieve at any point of time.

\<GSU- HomeAwayFromHomePortal_SU-2017 Feedback - 21 \>

Using this requirement allow us to update the details of notes by giving the attributes user_name, feedback. This functionality in database that update the details of the notes in database and we can retrieve at any point of time

\<GSU- HomeAwayFromHomePortal_SU-2017 Admin - 22 \>

By using this requirement allow us to add the details of the admin by giving the attributes user_name, password. This functionality in database will add the details of the data and we can retrieve the data at any point of time.

<GSU- HomeAwayFromHomePortal_SU-2017 Admin - 23 >

By using this requirement allow us to delete the details of admin by giving the attribute user_name. This functionality in database that can delete the details of the data and we can retrieve the data at any point of time.

<GSU- HomeAwayFromHomePortal_SU-2017 Admin - 24 >

By using this requirement allow us to update the details of project by giving the attributes user_name, password. This functionality in database that can add the details of the data and we can retrieve the data at any point of time.

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)
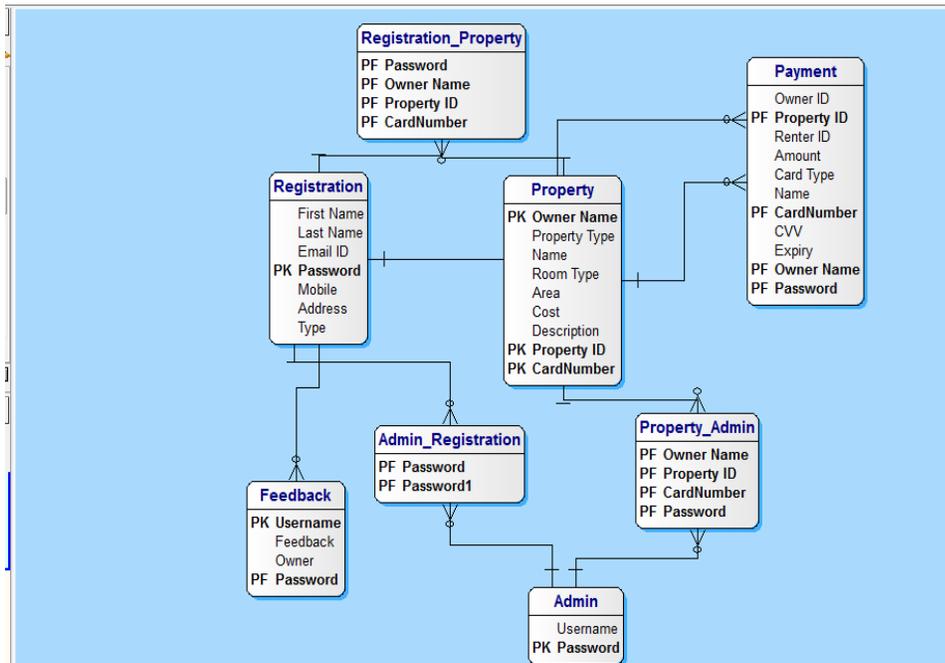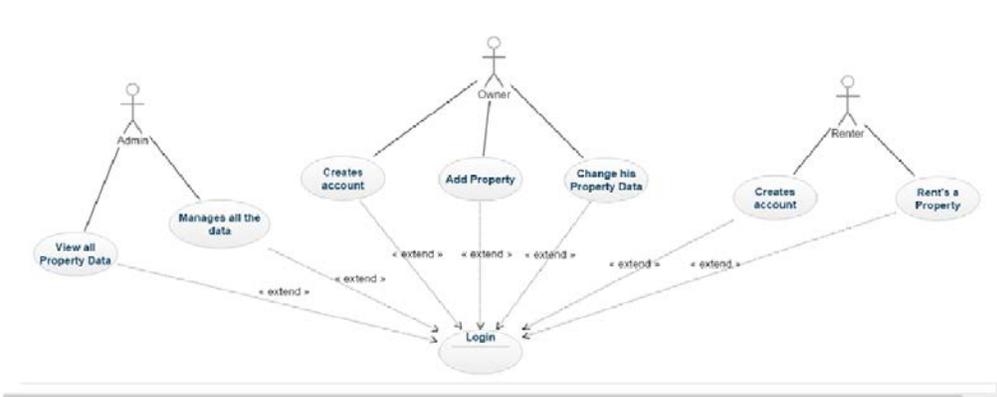
1. Administrators have the privilege to see, alter and refresh, erase the proprietors and tenant's record/postings. Essentially, administrator is the person who deals with every one of the information.

2. Owner has the privilege to include his property, refresh and remove his/her property.

3. Renter can lease the property.

3.3   Security and Fraud Prevention

Security is a must for any website. In fact, customers trust us when they are confident about the security.

1. To add the property owner needs to mention add the required details.
2. Even the renter needs to provide all the necessary details to register into the website.
3. Secure network connection should be used.
4. All employees are given training on security compliances.
5. All the activities done by owners and renters should be monitored by the admin.
6. Data is backed up time to time and have disaster recovery plan.

3.4   Release and Transition Plan

- Buy a domain name and maintain DNS service.
- Functional testing of the website and make sure the owners and renters can register.

- Make sure backup happens time to time.

- Monitor the website for few days before it goes live.

4. *Project Design Description*

Home away from home website homepage looks like this:
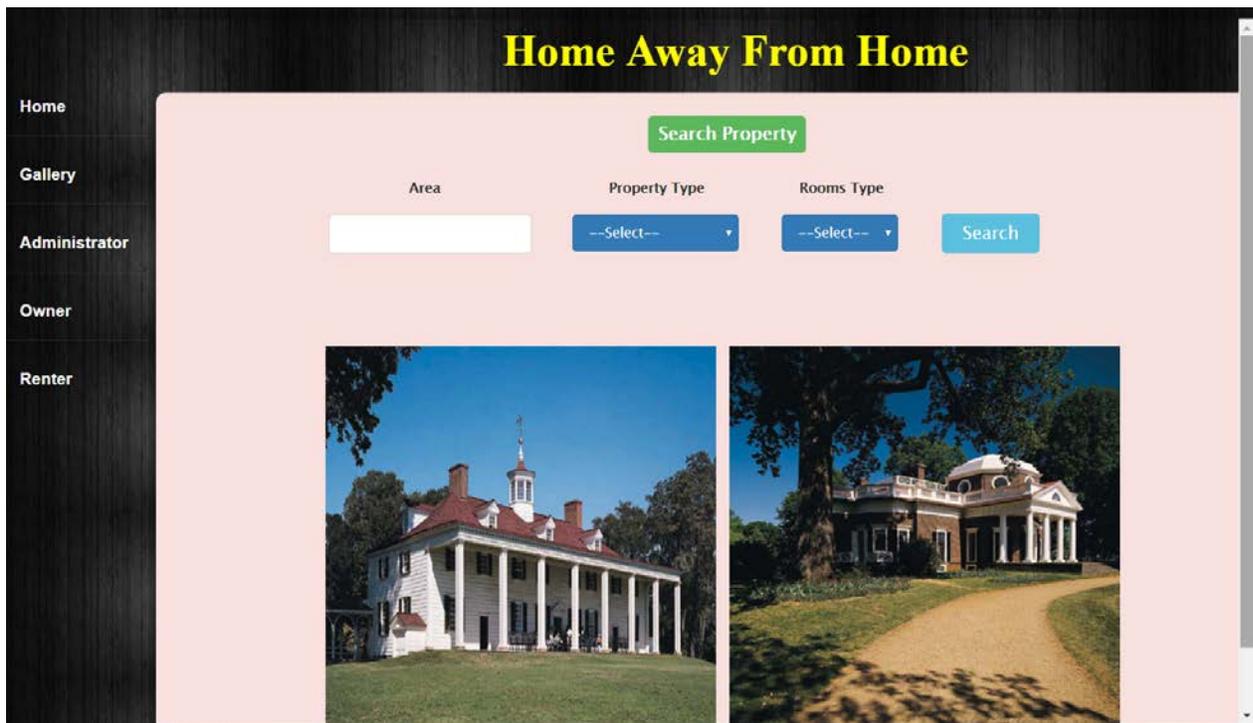
Home page has diverse alternatives like gallery, administrator, owner and renter. we utilized bootstrap assets to plan the login page. We have asp textboxes tolerating the username and secret word and one server control catch to deal with the occasion for all the login pages for administrator , proprietor and tenant.

Manager can see every one of the exercises done by proprietors and tenants.

4.1 Admin Login:

Admin login page looks like the following:

At the point when the "login" catch is clicked, the certifications which are entered are approved against the qualifications which are put away in database. In the event that they coordinate the administrator can see the landing page.

Once signed in, it would appear that this:

Administrator can see proprietors, tenants and every one of the properties recorded. Notwithstanding these, administrator can likewise observe feedback which are given by the renters to owners which thusly is utilized for the improvement of this site.

4.2 Owners Login Page:

Owner registration looks like the following:

Owner should specify his points of interest like initially name, last name, versatile no, email address and address. This data is required to enlist and after that he can include property by signing in. Once the owner taps on "Enroll" catch, all the data is put away in the database and the proprietor would then be able to enter the site with his qualifications that he made. We have utilized text boxes to enter the points of interest and bootstrap to outline the page. We additionally utilized JQuery for the plan of this page. For putting away every one of the points of interest we utilized the local database.

Once enlisted and signed in, proprietors landing page resembles the accompanying:
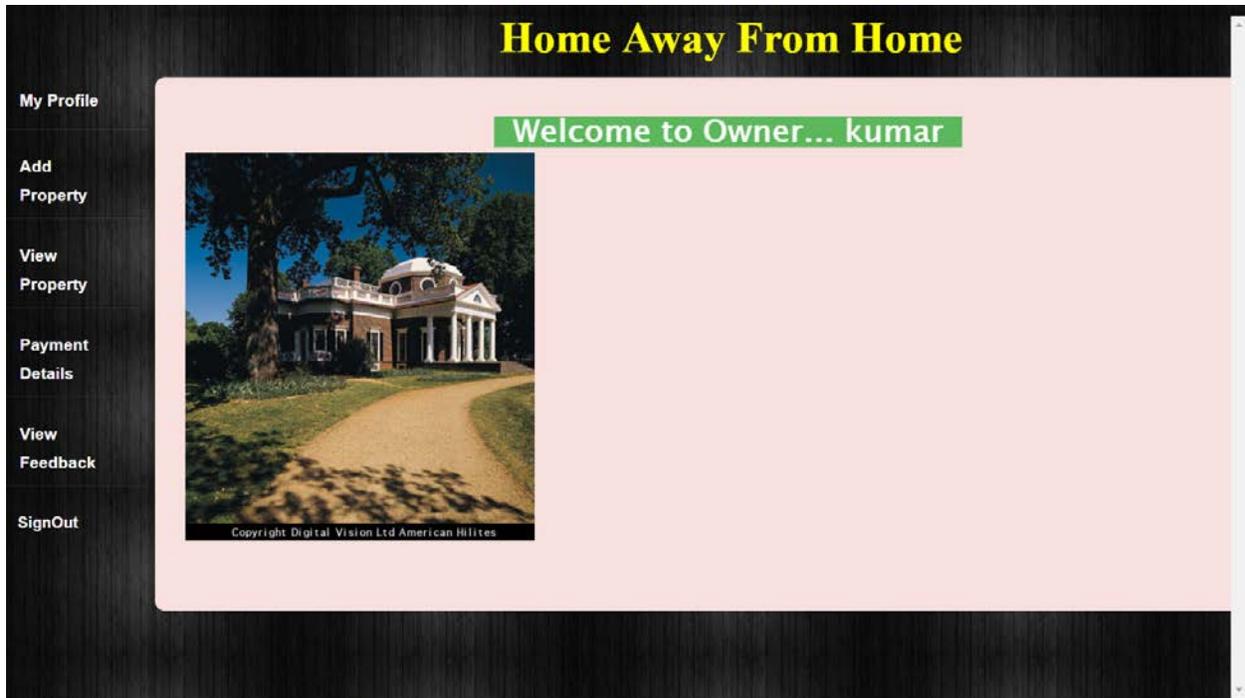
Proprietor can include property, see/refresh his property.

At the point when the client taps on "include property" he/she can include the property. when he includes the property, the data is put away in database and is accessible for survey by all tenants. in this way, the renters can book the property. We have utilized text boxes, drop down records to enter the points of interest and furthermore property subtle elements and bootstrap to outline the page. We likewise utilized jquery for the plan of this page. For putting away every one of the points of interest we utilized the local database.

4.3 Renters Login page:

Renters registration looks like the following:
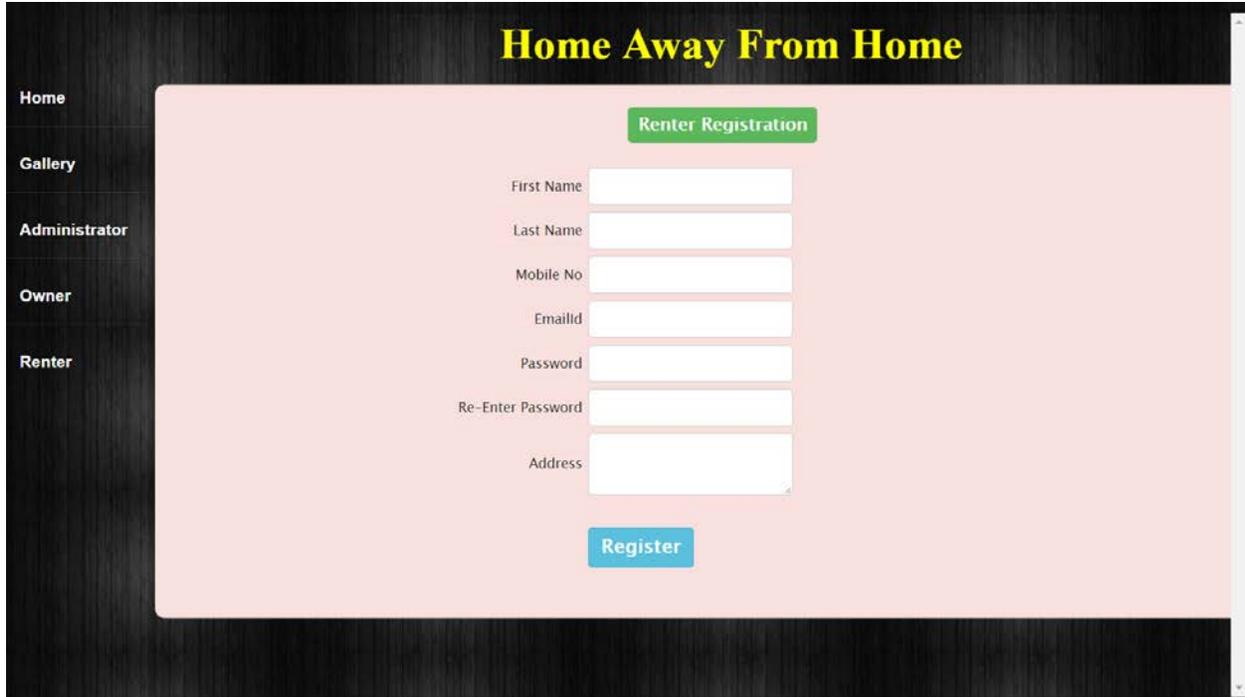
Renter ought to likewise give his first and last names, email address, telephone number and deliver to enroll and view and lease the property. Once the tenant ticks on "enroll", the data is put away in the database and the tenant can utilize this data to login and seek/lease any property. We have utilized text boxes to enter the points of interest and furthermore property subtle elements and bootstrap to plan the page. We additionally utilized jquery for the plan of this page. for putting away every one of the subtle elements we utilized the local database.

Once registered and logged in, the renter's homepage looks like the following:

The renter can search for the property in the coveted zone by saying they are in the "Area" field. He/she can choose property sort. Renter can likewise have a channel on number of rooms (like 1bhk, 2bhk and so on). We have utilized text boxes, drop down records and catch to enter the subtle elements and to look through the property points of interest and bootstrap to outline the page. We additionally utilized JQuery for the plan of this page. For putting away every one of the subtle elements we utilized the local database.

5. *Internal/external Interface Impacts and Specification*

There are a few contrasts among administrator, proprietors and leaseholders when we consider useful region. Administrators can see every one of the information (property postings, proprietor's information and leaseholder's information). Owner can just refresh/erase/include their own particular property and furthermore refresh his/her own particular profile. The owner can likewise see the feedback that is given to them by a renter. Renter can lease the property that is as of now

recorded and furthermore can refresh their own profile. We must be cautious about these situations as tenants shouldn't approach proprietor's information and proprietors ought not have consent to refresh/erase other proprietor's postings and renter's. In the event that there are any progressions to the useful necessities, the entire venture will be affected.

6. Design Unit Impacts

6.1 Functional Overview

While implementing this application we have tried many different cases like if a person books a home on the website so the property will automatically get disappeared so it is the sign that home is booked. User in this application can give feedback about the owner and this feedback is checked by the admin if to many bad reviews it may result in termination.

No	Page	Action	Expected Result
1.	Renter's Page	can look for the properties and book the properties and give feedback.	Booked property will be stored
2.	Owner's Page	add properties and remove properties and can alter the properties and manage the properties	It will be stored in database
3.	Admin's Page	controls the application can remove the properties and manage the owners and renters list	Results will be stored

6.2 Impacts

Impacts between the owners and renters and admins are show clearly in below use case diagram and they are clearly stated with the arrows.
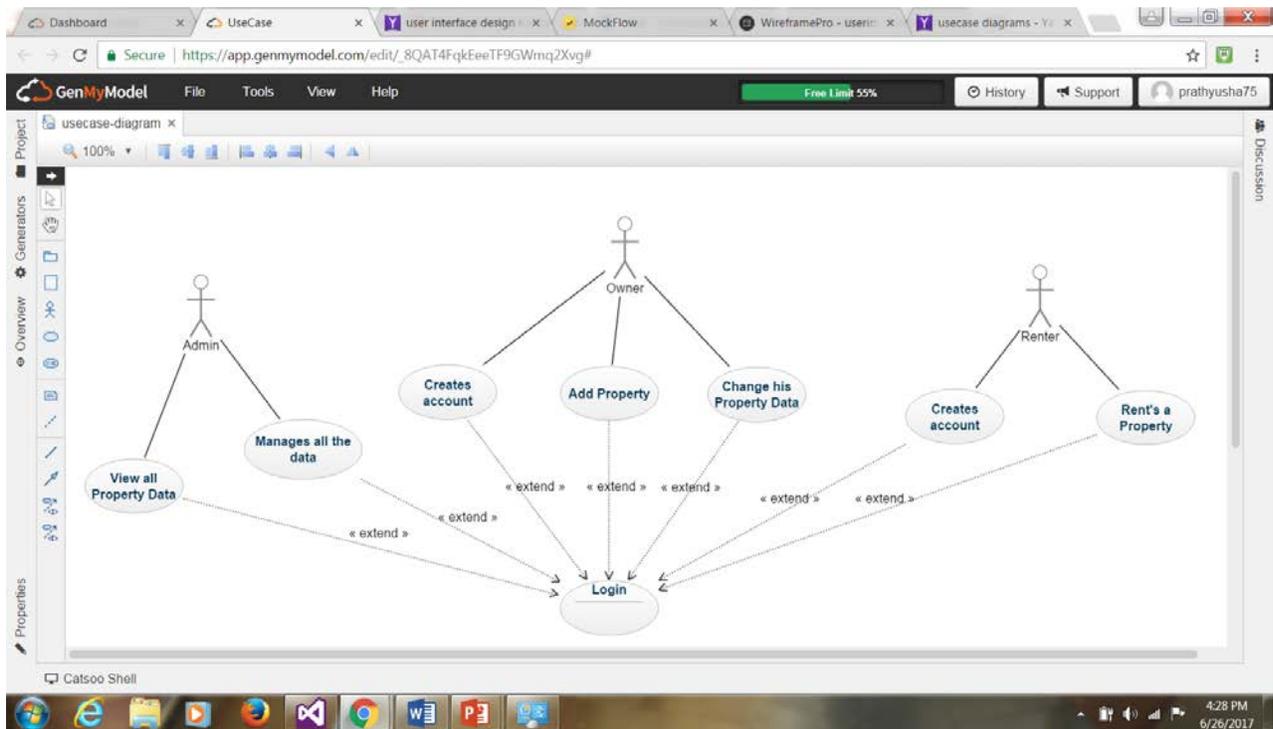
6.3 Requirements

Software Requirements:
- ➢ Microsoft visual studio 2015.
- ➢ Microsoft SQL Server Express edition

Hardware Requirements:
- ➢ Processor : Intel dual core or above
- ➢ Processor Speed : 1.0 GHZ or above
- ➢ RAM : 1 GB RAM or above
- ➢ Hard Disk : 20 GB hard disk or above

➢ Platform : Windows, Mac , Linux

7. Open Issues

In this project we had a few issues while building up the venture like generally in the website page outline and database association and we have understood it experiencing on the web instructional exercises and some course readings so we ran over the issues.

8. Acknowledgements

I would like to thank professor Liu Alex for helping our project and I would like to thank each and everyone in the team for making this project successful .

9. References

1. Web applications textbook by "Carles Mateu" https://archive.org/details/ost-computer-science-fta-m4-web_development

2. W3schools for developing the web pages https://www.w3schools.com/html/default.asp

3. For the database and web pages connectivity we used

   https://downloads.mysql.com/docs/apis-php-en.pdf

4. W3schools for sql queries https://www.w3schools.com/sql/default.asp

5. Sql and web pages connectivity

   http://home.hit.no/~hansha/documents/database/Visual%20Studio/Database%20Communication%20using%20ASP.NET%20WebForms.pdf

6. Visual studio https://www.visualstudio.com/downloads/

10. *Appendices*

No Appendices.