

Facilitating the search for Off-Grounds Housing at UVa

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Pragnya Pandrate

Spring, 2022

Sion Kim

Chloe Estrada

Andrew Gilikin

Pragnya Pandrate

Kiyan Zewer

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Dr. Rosanne Vrugtman, Department of Computer Science

Abstract

Off-Grounds Housing at UVa is something that all upperclassmen need, but have difficulty finding because there is no central application that makes the search personalized and easy. My fellow engineering students and I developed a website that allows students to browse and sort different housing options with accurate information and to communicate with others regarding housing. In the process of creating this website we gathered requirements from students through surveys and continually improved our website through Beta testing. This website utilized Django as the web framework and involved the use of UI/UX design, unit testing, databases, and Heroku deployment. After fine-tuning, our project enabled students to look for housing through our map search and sorting features. Students were also able to communicate with each other through publicly available reviews and a public forum. In the future, although all of the features of the website are functional, the UI/UX design can be improved to reflect the changing needs of the students.

1. Introduction

While On-Grounds housing options are detailed on UVa's website, it falls short as it does not accommodate the many students looking for Off-Grounds housing as an upperclassman, transfer student, or graduate student. The current UVa website that deals with this has a limited database which makes it hard for students to search for a residence without consulting multiple other sources, a process that is time-consuming.

It also does not offer the option to rate/review properties. Considering that many students make decisions on where to live based on others' experience, this feature is vital to a housing search website. By limiting the information about properties to only what the property management chooses to display, the website gives a skewed view on what the properties might actually be like.

Additionally, many students find it difficult to find roommates when looking for Off-Grounds housing. Although quite a few students find roommates through friends, there are still many students who are in need of roommates when searching for housing in order to be more economically conservative when paying for rent.

Many of these students resort to social media to solve this issue, however this is only because the current housing website doesn't provide a platform for interacting with other students.

The lack of vital features on the current website is further exacerbated by the fact that most leases off-grounds need to be signed almost 1 year in advance. For transfer students and 1st years, this means they only have approximately 2 months to look for housing and sign a lease. This is already a difficult thing to do but the lack of features on the website make it even more difficult to go through the process of finding housing, roommates, and viewing reviews/ratings of properties. The lack of these features means that students might not be able to sign a lease in time therefore risking their housing options.

2. Related Works

The features that were included in the website we created were based on a requirements elicitation in which we surveyed and interviewed undergraduate students on their experiences in the search for Off-Grounds Housing (Surveys and Interviews of undergraduate students, Personal Communication, September 2021). From these interviews and surveys we concluded that the most important features were related to being able to view ratings and reviews of properties and to find roommates. This requirements elicitation helped us prioritize what features to implement in our website in order to give students the most value when searching for housing.

We also consulted the current official UVa Off-Grounds Housing website was also consulted in order to improve upon features that were lacking and incorporate some of the features that were useful based on students' experiences (University of Virginia, Apartments.com, and CoStar, 2022). For example, the map feature that allowed students to view properties' locations on a map proved to be useful as many students wanted to use this feature to find properties close to Grounds. This feature was also incorporated in our website using Google Maps. Additionally, students wanted to be able to favorite properties to view them later. This was already a feature on the official website which we re-implemented as it proved to be useful for users as they pursued their search for Off-Grounds Housing at UVa.

3. Process Design

After defining the requirements for the project through surveys and analysis of other related works, we created a plan for what tasks needed to be done throughout the 3 months. We divided up the tasks into 2 week “sprints” in order to check in with the team every 2 weeks to discuss our progress.

3.1 Setting up the project and its architecture

One of the project’s requirements was to use Django which is a web framework that makes it easy to write code for websites. Django uses a model-view-controller framework in order to organize the code that is written for the website. The “model” refers to the database that is used to store all the data, the “view” refers to what the users physically sees on the displayed webpage, and the “controller” connects the model and view so that the view is updated based on the data. Fortunately, all of this initial set up is done by default when a Django web app is created so most of the code organization was already done for us.

Next we had to set up the general architecture of the website, which included a database, Heroku, and GitHub. In order to be able to store data about reviews and housing options we needed to use a database. Heroku was used to deploy our website to the internet so that it was publicly viewable. The hardest part of this step was to make sure that the database we were using in Django (which was SQLite by default) connected to Heroku. Heroku did not support SQLite so we had to switch our database to Postgres in order to deploy our app using Heroku. Lastly, GitHub was used as a version control system so we could keep track of different versions of the code used for the website as we all collaborated on it.

3.2 Implementing the main features of the website

One of the first features we had to implement was a sign in page. We decided to allow users to sign in with their Google accounts, if they did not have a Google account they would have to create one. We chose to do this because we would not have to save username and password information in the database and we assumed that most of the students who would use our website would have a Google account. In order to implement this feature we used OAuth (open authorization). This is an internet protocol that allows our website to access Google’s sign in resource in order

to grant users access to our website. Rather than implementing this whole protocol ourselves which would have been tedious, we used a Python library extension that had already implemented OAuth. Adapting their code allowed us to use OAuth to allow users to sign in through Google without having to implement the whole thing ourselves. Once signed in we had the following features available to users: the listings page, the listing details page, adding reviews and ratings, the map page, the forum page, and the home page

3.2.1 Implementing the Listings page

We started with the listings page which was meant to show all the housing options that were available to students. Admin users were allowed to update this page with new listings that they found. This page allowed users to get main details of each property that was listed such as the price, location, number of bedrooms/bathrooms, etc. This page also allowed users to sort the listings using different criteria such as the price, distance from the University of Virginia, number of bedrooms/bathrooms, etc. Rather than implementing the sorting feature manually, we used a Python extension for this which made implementing the feature much easier.

This page also allowed users to “Favorite” certain listings so that they could revisit them in the future. For each listing we added a button that would allow the user to either add to favorites or remove from favorites. Every time a user updated their favorites using the buttons that were on each listing, we would update the database to reflect the user’s favorite listings. This way we could keep track of each user’s favorite listings. This is shown in the figure below.

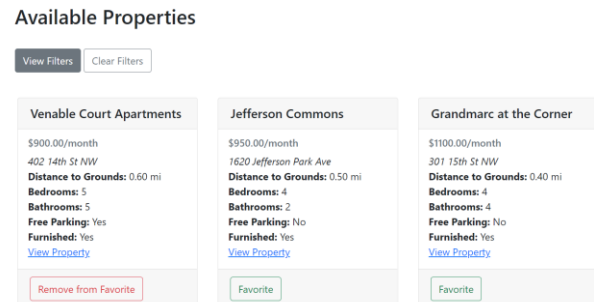


Figure 1. Screenshot of Listings page

3.2.2 Implementing the Listing details page

For each listing on the listings page, we linked it to a details page that would show details about that individual listing. This details page had more information about the property such as amenities, floor plans, average ratings that other residents had given, and reviews that other residents had given.

This information was inputted in the database by the admin users and retrieved from the database whenever a user clicked on a property. This allowed for only admin users to edit information about a property which ensured the accuracy of the information.

3.2.3 Allowing users to submit reviews and ratings for each property

As mentioned in the previous section, each listing's detail page also displayed average ratings for the property and all the reviews that were submitted. The average ratings were separated into categories such as overall rating, services rating, amenities rating, and noise rating as shown in Figure 2 below.



Figure 2. Example of how ratings are displayed for a given property

By giving both the overall rating and categories' ratings, the user was able to view as much as information about a property as possible through other residents' ratings.

Reviews of a given property were also displayed at the bottom of the details page. These consisted of text formatted reviews from residents so that users looking for housing could read other's opinions of the housing. One of the biggest drawbacks of the current Off-Grounds housing website is the fact that users could not gather information from residents who had lived there, which is why implemented the reviews and ratings feature for our website.

In order for current or past residents of a property to submit a review or a rating about a property, they could go to the home page to fill out a

form inputting this information. The form has a dropdown text box where they can select the property they want to rate/review and then fill out the rest of the form according to their opinions about the property. Whenever the form was submitted, the information was stored in the database. Then when the listing details page is loaded, all the reviews are retrieved and displayed while all the ratings for that property are retrieved from the database to display the averages.

3.2.4 Implementing the Map Page

One of the features that was already present in the official Off-Grounds Housing website that many students found useful was the map feature. The aim of this feature was to map all the properties on a map of Charlottesville so that students could easily visualize where all the properties were.

On our website we incorporated Google Maps through the Google Cloud SDK which allowed us to include some Javascript code that rendered Google Maps. Users could then use the search bar to search for addresses. They could also view the house markers that were placed on the map which indicated the locations of all the properties on the Listings page.

3.2.5 Implementing the Forum Page

A common request from many users when we were eliciting requirements for the project was to have a way to communicate with other students looking for housing. This is why implemented the Forum page where students are able to make posts about any topic they want and other students can also follow up on the discussion.

We did not store any of the forum posts in the database as we felt like this information did not need to be retrieved at any time. Rather, whenever a post was made it would immediately render on the screen through some simple HTML and CSS formatting. Each post had a Discussion Topic and included the replies from users. If a user wanted to reply to an existing Discussion Topic they would just have to fill out the reply text box which would trigger the reply to render on the screen after it was posted. Similarly, to create a new Discussion Topic the user just had to click a button and fill out the appropriate text boxes for the post to render after it was submitted. An example of how the forum renders is shown below.

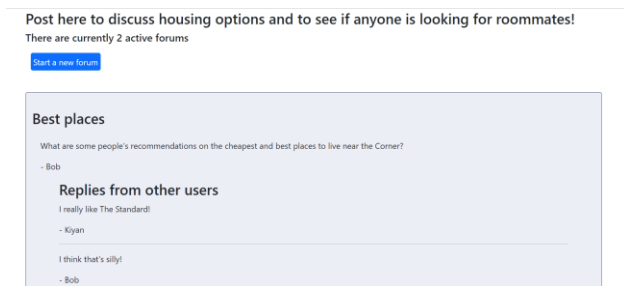


Figure 3. Screenshot of the Forum Page

3.2.6 Implementing the Home Page

We implemented the Home Page last after implementing all the above features, because we did not know what information would be important to display on the Home Page until all the features were implemented. Once the other features were implemented, we decided that the only two important things to include would be the Favorited properties for the user that is logged in and a button so that the user could publish a review/rating. We could have included more information on the Home page but we decided against it in favor of a cleaner, minimalistic design which is shown below.

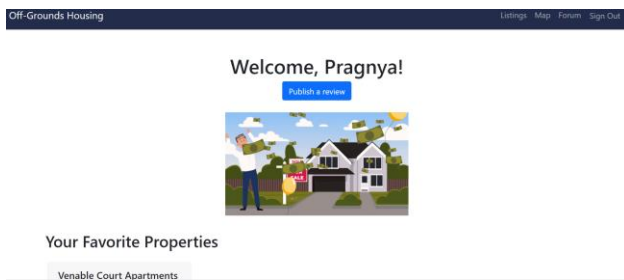


Figure 4. Screenshot of Home page

3.3 UI/UX Design

Although we worked on UI/UX design throughout the project by making sure the color schemes were not distracting and text/images were organized, we worked on more of it after implementing all the features. One of the main parts of the website that we improved in this aspect was the Listing details page. Originally, the page was very cluttered with images and text interspersed throughout the page. This made information hard to locate and the user had trouble navigating the page. Therefore, we opted to have a more organized look by putting all the text information into a table format in HTML. This way all margins were aligned in terms of the text was displayed. We also decided to have a picture of the property to the

right to avoid too much whitespace and make the beginning of the page more visually appealing, as can be seen in Figure 5 below.



Figure 5. Example of the layout of the Listing details page that shows detailed information about Grandmarc

3.4 Testing

Including testing for our website was also important as we wanted to make sure that an error didn't cause our website to become non-functional. We included unit tests which allowed us to test different parts of our codebase. For example, one of our unit tests tested to make sure that whenever a rating was making sure that the filters on the Listing page correctly filtered the properties. Other examples of unit tests we included were testing to make sure a user was logged in before they could access any of the private pages and testing to make sure that properties that were added correctly rendered on the Listings page.

Whenever a change was made to our codebase and pushed to GitHub, all these unit tests were run to make sure that the code that was changed passed all the tests. This way we ensured that any time code was changed it did not break the website and inconvenience the user.

3.5 Challenges

One of the main challenges we faced throughout the development of the website was UI/UX design. Everyone on my team had experience with coding but were not as familiar with UI/UX Design. One of the main ways we combatted this was through user testing. Once we had finished a beta version of our website, we had users test it and give us feedback on how the website looked and how navigable it was. We based a lot of the changes we made to website's design based on their feedback.

Another challenge we faced was implementing test driven development throughout the process. Test driven development is when tests are written first and code for the application is written based off of that. We struggled with writing tests throughout the process

since we did not prioritize it as much. Towards the end of the process, we did write more tests but this was definitely a challenge and an area for improvement.

4. Results

The end product of this process was a website that allowed students to more efficiently look for Off-Grounds Housing. During the requirements elicitation some of the main concerns students had were not being able to view residents' opinions on properties and not being able to interact with other people that might be looking for housing. These helped drive our project in a direction that solved both these problems as we implemented a reviews/ratings mechanism and a forum where students could interact with each other. Compared to the official Off-Grounds Housing website we were able to provide a better platform for students at UVa to find housing.

Through Beta Testing we also gathered that some of parts of our website were not navigable so we improved our UI/UX Design by making certain parts of the website less cluttered and designing pages in a way that drew attention to the main features. This made our website more navigable and useful compared to the official website.

5. Conclusion

Before starting this project, many students struggled with finding Off-Grounds housing as it was a time confusing and confusing process, especially for first-timers. We aimed to make the process of finding housing smoother by having a website with all the features students might need to find housing. We gathered around 50 responses from students through interviews and surveys to implement features they thought would be helpful in the search for housing. The aim was to have one website for all their house search needs. We achieved this goal as we were able to implement a website with features like favoriting properties, sorting properties, viewing properties on a map, reading reviews/ratings, and discussing properties on a forum. This gave students a much easier platform to search for housing and made the process less time consuming.

6. Future Work

Although the website has all the functionalities that help make searching for housing easier, the UI/UX

design can still be improved. For example, right now the user has to go to the home page in order to submit a review/rating rather than just going to the corresponding listing details page. This does not make intuitive sense and might be confusing for the user. In order to find other parts of the website that are not intuitive, more Beta testing should be done. Beta testing will allow the developers to gather more information straight from the user about what works and what doesn't.

Additionally, another avenue for improvement would be to change how properties are added to the website. Currently only admin users can add information to the website about new properties. However, this could get cumbersome as the website grows and more properties need to be added. Therefore, having a system where users can submit properties to admin users for review so that they can be added to the website would be helpful. This way admin users can still maintain accuracy while getting information easily.

References

Pandrate, Pragnya, Kim, Sion, Estrada, Chloe, Gilikin Andrew, Zewer, Kiyana. 2021. Requirements Elicitation.

CoStar Group. 2022. University of Virginia Off-Grounds Housing Website. Retrieved from: <https://offgroundshousing.student.virginia.edu/>