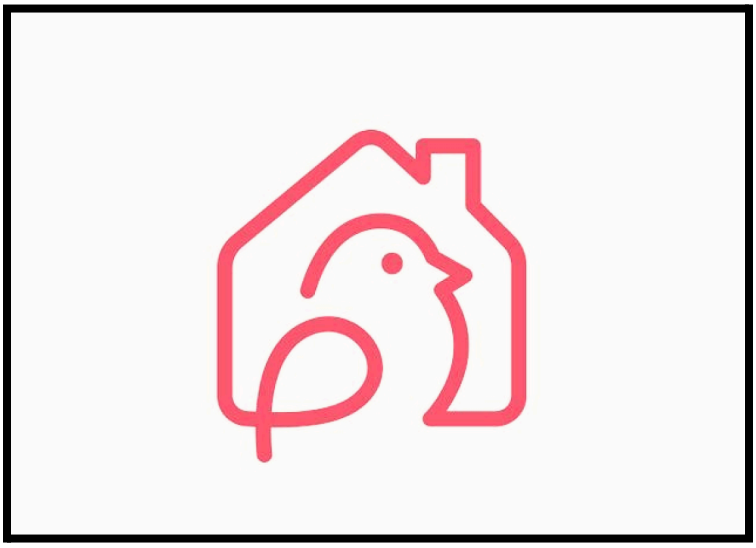
UNEST

CS 307 - Sprint #2 Retrospective

Team 13: Sudhanva Bharadwaj, Nivedha Kumar, Ram Laxminarayan, Grant McCord



What Went Well

Overall, the team completed all of the promised user stories for the second sprint. We created new pages such as the user profile, new listings form, messages, and chats pages. We also connected all of our pages to the database. Also, a big milestone was reached this sprint which was implementing the user context. The application functions more like a real website now by knowing the logged in user and carrying their info from page to page to load data accordingly.

Home Page:

1. As a user, I would like to be able to see various types of properties located in my area

Task #	Description	Estimated Time	Owner
1	Pull information about each of the properties from the database (name, date, price, and miles from campus) instead of hardcoding them.	4 hours	Nivedha
2	Create a unit test to make sure that properties are being pulled from the database	2 hours	Nivedha

Completed:

I made an axios call in the frontend that makes a http GET request to retrieve property information from the listing collection in the database when the homepage is mounted (started). It would then go to the corresponding app.get method in the backend index.js file by looking up the matching url. In that app.get method, I used the mongoose library Listings.find() method to retrieve all the property information from the database. res.json then converts that response into a JSON object of key value pairs and sends it back up to the frontend. After retrieving that information I displayed it in the property cards. I used the moment.js library to format the date so it is in a typical human readable format.

User Profile Page:

2. As a user, I would like to see a user profile page with some information about myself.

Task #	Description	Estimated Time	Owner
1	Pull in data from the database with the basic information text about the user like age, gender, pronouns, and university to display on the page.	3 hours	Nivedha

2	Pull in data from the database with the paragraph description about the user and display it under an "about me" section	3 hours	Nivedha
3	Pulls in data from the database to put in a details section about the user. It would have information like their year, major, minor, and hobbies to display on the page.	5 hours	Nivedha
4	Create a unit test to make sure the basic info section, about me section, and details about myself section are all rendering properly.	3 hours	Nivedha

Completed:

I made an axios call in the frontend that makes a http GET request to retrieve user information from the users collection in the database when the user profile page is mounted (started). It also retrieves that information anytime a change is made to the user's collection. It would then go to the corresponding app.get method in the backend index.js file by looking up the matching url. In that app.get method, I used the mongoose library User.findById() method to retrieve all the property information from the database. res.json then converts that response into a JSON object of key value pairs and sends it back up to the frontend. After retrieving that information I displayed it in three different sections on the user profile page: basic info, details, and about me sections.

3. As a user, I would like to be able to edit some information in the user profile page.

Task #	Description	Estimated Time	Owner
1	Implement an edit icon that allows users to edit and update basic information text about the user like age, gender, pronouns, and university to display on the page	3 hours	Nivedha
2	Using that same icon, implement functionality that allows users to edit and update the paragraph description about the user in the "about me" section	3 hours	Nivedha
3	Using that same icon, implement functionality that allows users to edit and update the details section. That section has information like their class, major, minor, and hobbies to display on the page.	3 hours	Nivedha
4	Send a query to the database to update the corresponding information for the user if they edit information	5 hours	Nivedha
5	Create a unit test to make sure the basic	2 hours	Nivedha

	information, about me, and detail sections on the user profile page are getting edited properly		

<u>Completed</u>:

I made an edit pencil icon that switches between regular viewing and edit mode. It either shows regular text or a list of text fields respectively. If the user enters new information in the text fields in edit mode and hits the save button, it makes an axios call in the frontend. That axios call makes a http PUT request to update user information in the user collection in the database when the homepage is mounted (started). It would then go to the corresponding app.put method in the backend index.js file by looking up the matching url. In that app.put method, I used the mongoose library Listings.findByIdAndUpdate() method to update the user information for the user with a specific id. res.json then converts that response into a JSON object (key value pairs) and sends it back up to the frontend. If the user hits the cancel button instead of the save button, then the mode changes back to regular viewing mode and the changes they made to the text fields go away.

New Listing Page:

1. As a lister, I would like to be able to fill out a form with all the specific information about a property including a short description of the property and a list of amenities.

Task #	Description	Estimated Time	Owner
1	Create form for listings that asks for basic information about the property	3 hours	Sudhanva
2	Create buttons for Create and Delete listings. Make buttons route to new pages based on design.	2 hours	Sudhanva
3	Connect listing page to database. Creating a new listing should add a new entry to the collections of the database.	3 hours	Sudhanva
4	Add an option to select start and end dates for listing that the interested tenant can see.	2 hours	Sudhanva
5	Test the functionality of the listing page with valid and invalid entries. Manual test cases written for various input combinations.	2 hours	Sudhanva
6	Test that the database is updated when the submit button is clicked with valid entries	1 hour	Sudhanva

<u>Completed</u>:

I implemented a working form that allows the logged in user to enter information about a new property. The data is stored in variables until the form is submitted by the user. Upon submission, the data is sent to the database with a post request and saved into the places collection. I also added a create new post button on the mainpage that takes the user to this newly created form.

2. As a lister, in the form, I would also be able to provide the property's exact location and pictures of the property to be displayed.

Task #	Description	Estimated Time	Owner
1	Create an address field for the listing page.	1 hour	Sudhanva
2	Add the upload images button for users to select pictures of their property. Multiple images should be allowed.	2.5 hours	Sudhanva
3	Allow for the amenities of the property to be selected via checkboxes.	1.5 hours	Sudhanva
4	Test the rendering of the page to ensure that the address and picture are shown on screen.	1.5 hours	Sudhanva
5	Test the network response of the listing page to ensure the proper fields are being sent	1.5 hours	Sudhanva

Completed:
I also added the ability for the user to upload images from their own device that will be displayed with their property. The images can be selected when clicking the upload button and render on the screen as they are uploaded. I faced some challenges in being able to save multiple images but eventually fixed the issue by creating an array that appends the images as they are added. Additionally, I added checkboxes for the amenities to be selected which required another array as all or none of the options could be selected.

My Listings Page:

3. As a lister, I would like to be able to see my listing from the My Listings page.

Task #	Description	Estimated Time	Owner
1	Create a button for creating a new listing. The button should render a new form and create a listing.	1 hour	Sudhanva
2	Add the entry from the database when the user selects to save. The listings should also be shown in the my listings tab for the user to view after creation.	3 hours	Sudhanva

3	Display an error/success message after creating a post.	1 hour	Sudhanva
4	Test that the fields in the database match the ones entered by the user.	1.5 hours	Sudhanva
5	Test that the creator of a listing is able to see their past listings with all the information they initially entered including images.	1.5 hours	Sudhanva

Completed:

For this story, I added the ability for each user to see the properties they have listed. I used the user context that I had implemented to get the id of the user that was currently logged in. Using that, I used the form submission to store the data fields of the listing as well as the ID of the logged in user. Then, on the my listings page, I looked for the ID of the user that was signed in to match the ones in the database. I then displayed the address and name of the properties that were returned as a result. I also added alert messages to notify the users if there were issues in creating their listing such as invalid fields.

4. As a lister, I would like users to be able to directly contact me through the listing page.

Task #	Description	Estimated Time	Owner
1	Create a contact button that shows the lister's information	1.5 hours	Sudhanva
2	Route button to the messages page so that user can connect with lister	1 hour	Sudhanva
3	Test that the lister's id/name is tagged to each listing as this is how the messages page will know who to connect to. Manual check in database if each listing is linked to a specific user.	2 hours	Sudhanva

Completed:

From the individual property pages, I added the ability for a user to click the message host. When that button is clicked, I made it so that it routes to the messages page with the ID of that specific lister. Since the user context allows us to distinguish each individual user, I was able to implement this and prepare the full messaging implementation the next sprint.

Home page:

1. As a potential tenant, I would like to sort the properties by the number of bathrooms.

Task #	Description	Estimated Time	Owner
1	Add a radio button to the Search Bar titled Bath Count	2 hours	Grant
2	When the search button is clicked, query the database and filter by number of bathrooms.	5 hours	Grant
3	Show all of the selected properties in a grid on the main page.	1 hours	Grant
4	Create unit tests to verify that the squares in the grid are populating with the correct information.	3 hours	Grant

Completed:

I added a button titled Bath Count that will filter the listings queried from the database by the number of bathrooms in the listing in ascending order. I included this in a new section of the search bar alongside the other user stories I completed this sprint. This button is part of a group of radio buttons with the Bedroom Count button that will only allow one of them to be selected at a time, so a user can only filter by one of the categories.

2. As a potential tenant, I would like to filter the properties by specifying a given price range.

Task #	Description	Estimated Time	Owner
1	Create a slider in the search bar called Price	4 hours	Grant
2	When the slider is updated, query the database and only return listings that satisfy that price range.	4 hours	Grant
3	Show all of the selected properties in a grid on the main page.	1 hours	Grant
4	Create unit tests to verify that the squares in the grid are populating with the correct information.	3 hours	Grant

Completed:

I added a slider named Price that will allow the user to adjust the max price of listings that they want to see. The slider has a value above it that tells the current value of the slider, and thus the max price of a listing that they will see. I included this in a new section of the search bar beneath the two radio buttons I completed this sprint.

3. As a potential tenant, I would like to sort the properties by the number of bedrooms.

Task #	Description	Estimated Time	Owner
1	Add a radio button to the Search Bar titled Bedroom Count	2 hours	Grant
2	When the button is clicked, query the database and filter by number of bedrooms.	5 hours	Grant
3	Show all of the selected properties in a grid on the main page.	1 hours	Grant
4	Create unit tests to verify that the squares in the grid are populating with the correct information.	3 hours	Grant

Completed:

I added a button titled Bedroom Count that will filter the listings queried from the database by the number of bedrooms in the listing in ascending order. I included this in a new section of the search bar alongside the other user stories I completed this sprint. This button is part of a group of radio buttons with the Bath Count button that will only allow one of them to be selected at a time, so a user can only filter by one of the categories.

4. As a user, I would like the ability to search for properties on different campuses.

Task #	Description	Estimated Time	Owner
1	When the search button is clicked, query the database and only show the results that match the search criteria and other filters.	3 hours	Grant
2	Allow the search bar to search for campus location or listing name.	4 hours	Grant
2	Show all of the selected properties in a grid on the main page.	1 hours	Grant
3	Create unit tests to verify that the squares in the grid are populating with the correct information.	3 hours	Grant

Completed:

I added variables to keep track of all of the elements filter values and search terms. Rather than only updating the values when the search button is clicked, I implemented a feature where whenever anything is updated it automatically updates the listings on the main page to reflect that. So if you move the slider to a max price of 1000, only properties with a max price of 1000 will be shown, without needing to click on the search button. Most filters work together, so you

can set the max price and search for a name and filter by something, but you can only filter by Bath Count or Bedroom Count.

Messages Page:

12. As a potential tenant, I would like to directly start a chat with the lister.

Task #	Description	Estimated Time	Owner
1	Create button redirection for potential tenants to message a lister for a specific property.	1 Hour	Ram
2	Create the frontend design of where features such as the name of the lister and the text tool are displayed when a potential tenant messages a specific lister.	4 Hours	Ram
3	Ensure that a message can be sent.	2 Hours	Ram
4	Write a test to ensure that the message a potential tenant sends is saved.	2 Hours	Ram

Completed: When a potential tenant goes to the property listings page and clicks on the message lister button on that page, they can redirect to a specific chat with that lister. This was done using useNavigate and routing from the react-router-dom package. When a potential tenant is on the specific chat with the owner, they can see the person they are chatting with such as their name and username, previous messages sent/received from them, and an input field to input messages. This styling was done in the MessageOwner.jsx and MessageOwner.css file to ensure that users could easily see the owner's name and messages displayed. When a potential tenant inputs a message in the input field, they can see that message displayed on their screen below their other messages, and can continue to see it from refreshing, ensuring that the message is sent and saved, which was done using styling in MessageOwner.jsx and MessageOwner.css and useState in the react package to ensure that the state of the message is still sent and saved.

13. As a user, I would like to easily view all my messages.

Task #	Description	Estimated Time	Owner
1	Design frontend of the entire messages page.	5 Hours	Ram

2	Write a test to ensure that specific parts are rendered on the messages page.	2 Hours	Ram
3	Include redirection of going back to the home page and your own profile from the messages page.	1 Hour	Ram
4	Display a search bar to allow users to search for people to contact.	1 Hour	Ram
5	Have the list of users you are messaging be displayed along with the most recent message sent/received below their name.	1 Hour	Ram
6	Ensure recent chat/messages not viewed are in bold, but messages already viewed are in plain text.	2 Hours	Ram

Completed: Designed the frontend of the messages page that users see when they want to view all the chats that they have. This includes being able to redirect to other pages such as the homepage and profile page, a search bar to search for users, and a list of chats that the user currently has at the bottom of the page, which was styled in the MessagesPage.css and MessagesPage.jsx files. Messages that are most recently sent/received in a specific chat are displayed below the name of the user that the specific user is messaging, and messages in a chat that have not been read yet are displayed in bold. When that chat ends up being clicked through, when you go back to the messages page, the chat now only shows up in plain text, which was implemented using a boolean variable to determine if the chat was clicked or not.

14. As a potential tenant, I would like to be able to contact listers/users about their property.

Task #	Description	Estimated Time	Owner
1	Allow potential tenants to search for a property lister through the search bar based on their name.	3 Hours	Ram
2	Allow potential tenants to click on the lister's name to redirect to chat with them.	1 Hour	Ram
3	Write a test to ensure that redirecting to a chat with a specific user works.	2 Hours	Ram

Completed: When a user clicks on the search bar in the messages page, they can see a list of users that they can message, which were fetched from the database using axios. When a potential tenant inputs a specific name in the search bar, they can see that the user's names are filtered based on what they input in the search field, as if they enter random characters, the user's names will no longer be displayed since no name matches the filter. This was done by comparing the input string with the name of the user's. On top of that, when a user's name was clicked on, a potential tenant would be redirected to a specific chat in which the right user's name and username would be displayed at the top of the page. This was done using encodeURIComponent to pass data to another page based on the url, and route based on name using the react-router-dom package.

15. As a lister, I would like to have my messages to other listers and my messages from potential tenants be separated.

Task #	Description	Estimated Time	Owner
1	Design frontend of displaying potential tenants messages and messages to other listeners on the same page but not at the same time.	3 Hours	Ram
2	Implement toggle functionality between messages with listers and messages with other potential tenants.	2 Hours	Ram
3	When a user clicks on a specific chat, they will only see their chat for that person.	2 Hours	Ram
4	Write a test to ensure that only one person's chat is displayed and that toggling is possible between messages with listers and messages with potential tenants.	2 Hours	Ram

Completed: Designed the messages page with a togging ability such that listers can see their messages with potential tenants and messages with other listers separately. Two headings are displayed on the messages page: My Properties and Other Properties, in which one heading is highlighted in gray to determine the chats that the user is seeing. My Properties refers to messages with potential tenants interested in the user's property, and Other Properties refers to messages with other lister's about their property, so different messages are displayed based on each heading highlighted, as clicking on one of the headings highlights that heading. This was done through styling in MessagesPage.css and MessagesPage.jsx, and was done using a boolean variable to determine which heading should be highlighted and which messages should

be displayed. It was also ensured that clicking on a specific chat in each section only displayed one chat, which was done using useNavigate and routing in the react-router-dom package.

16. As a potential tenant, I would like listers to have the names of their properties included to help differentiate them.

Task #	Description	Estimated Time	Owner
1	Include the name of the property next to the name of the lister that you are messaging in the messages page.	1 Hour	Ram
2	Clicking the lister's name/property name should redirect the potential tenant to the lister's profile.	1 Hour	Ram
3	Also create redirection towards property listing of a specific property that the lister owns.	1 Hour	Ram
4	Write a test to ensure that all the redirection works.	2 Hours	Ram

Completed: In the messages page, when the Other Properties heading is highlighted gray, each owner's name is displayed along with the property name that they own in each chat. This is to differentiate each lister for the potential tenant based on the property that they own. When a property tenant clicks on a specific chat, they can see the owner's name and property name at the top of the chat, and when they click on either name, they can visit the owner's profile. In the owner's profile there are options such as a message button and view property button. Clicking on the message button allows a potential tenant to go back and message with the owner, and the view property button allows a potential tenant to view a description of the owner's property. All of this functionality was done using useNavigate and routing with the react-router-dom package.

What did not go well?

For this sprint, the team felt that all the promised stores were met in terms of expectations. The team completed all the tasks to the level that was expected by the members of the team. However, one thing that the team would like to improve on for the next sprint is being consistent with the UI and design. Right now, pages like the signup page and homepage are not consistent with their design schemes as they were developed by different members of the team. While the colors and layout have been fairly consistent, things like the text and logos need to be formatted to be delivered for the final sprint.

Developer Story 7 (numbering from the revised sprint planning doc): As a user, I would like to be able to edit some information in the user profile page.

Task #	Description	Estimated Time	Owner
4	Send a query to the database to update the corresponding information for the user if they edit information	5 hours	Nivedha

Completed, but needs to be improved:

I did all the requirements I specified in the user story above. However, this page only becomes fully functional if the user context is applied. Right now, the id of the user is hardcoded, but it needs to have the id of the user that is currently logged in. Also, we need to allow the user to upload a new image for the profile picture, so that is also editable. Also, in the next sprint, the roommate preferences and personal habit sections should be added to finish off the user profile page.

Developer Story 2: As a lister, in the form, I would also be able to provide the property's exact location and pictures of the property to be displayed.

2	Add the upload images button for users to select pictures of their property. Multiple images should be allowed.	2.5 hours	Sudhanva

Completed, but could be improved:

I completed this task as promised by my story, however I feel like the functionality could be further developed to make the site easier to use. Right now, you can add one or multiple images to the listing that will show up on the property page. However, I also want to add the ability to pin specific pictures to the post so that they show up on the mainpage. Another thing that should be

added is a button on the frontend to delete uploaded images. Right now to delete an image, you have to reclick the upload button and deselect the images which takes multiple clicks.

Developer Story 16: As a potential tenant, I would like listers to have the names of their properties included to help differentiate them.

Task #	Description	Estimated Time	Owner
3	Also create redirection towards property listing of a specific property that the lister owns.	1 Hour	Ram

Completed, but could be improved: This task is completed, as from viewing the profile of the owner, a potential tenant can view their specific property by clicking on the property page. However, there are not enough listings and owners that we have created/pulled to ensure this functionality across all owners and listings. Therefore, more listings and owners/users should be added, and redirection to each property listing based on if the owner owns that property listing should be ensured as more and more listings and owners are added.

Developer Story 11: As a user, I would like the ability to search for properties on different campuses.

1	Create a slider in the search bar called Price	4 hours	Grant

Completed, but could be improved: This task is completed, and all of the functionality related to the slider works fine, but the formatting of the slider doesn't work. Right now the slider has the default styling, which definitely does not look as good as it would with styling. I believe the issue lies in a faulty import, or conflicted imports between different styling libraries.

How should we improve?

Upon completion of the second sprint, the team feels that we met all the expectations and promises made at the start of the project. We completed key components of the site and ensured that their functionality closely aligns with the final expected design. The team did a great job of communicating throughout the sprint and seeking help when needed. Unlike the previous sprint, the team did not really run into major roadblocks that halted the progress of the sprint. The team can definitely improve on focusing on the smaller details of the site that are not necessarily outlined by the story themselves. The smaller details help tie the whole website together and allow pages to be consistent and flow from one to another. The team can work better on agreeing upon a set styling and using it across all the pages so that the website looks like it was developed all by one person. This is a minor issue with respect to the functionality of the website so it will not be too difficult to go back and fix.

Another issue the team realized in the second week of this sprint was that the user stories had some gaps in them that needed to be fixed. For example, one of the user stories involved adding the ability for a user to see their listed properties. However, in order for this task to be achieved, the team had to first properly implement the user context and know the reference to the currently logged in user. Since this task was not previously accounted for, extra time had to be allocated for the development of the user context before the rest of the dependent features could be implemented. Luckily, the team identified this issue early in week 2 which gave us plenty of time to adjust our plans and execute the stories as promised.

Additionally, the team can also improve on providing each other with daily updates in the form of a standup. This was a point of emphasis from the last sprint that the team did a better job this time around. However, there is still room for improvement in this as it is an essential part of Agile development and used throughout the industry. The team members have had conflicting schedules with other classes so it has been difficult to meet daily for a standup but we can adjust by sending a text message of our progress to keep the rest of the members up to date.

Overall, the team did a great job collaborating and executing the work that was expected. We encouraged various challenges in this second sprint but helped each other out to navigate through it. The mentioned improvements above will be implemented into Sprint 3 to provide an even better result and a smoother process during development.