CS 307

Software Engineering

Team 17

Bryan Duffy, Michael Reed, Simon Smith
Derek Schrag, Hari Mantena, Akshit Gudoor

Roomedy

Sprint One Retrospective

September 28th - October 16th

What went well?

Our first user story was: As a developer, we'd like to have a common development environment between all team members. In order to achieve this, we had to setup a Ruby on Rails environment on all of our computers, and create a basic version of our site using Bootstrap. This was our simplest task of the sprint, and was achieved in a mostly timely manner.

User Story: As a user, I want to be able to create an account and edit my information.

Task Description	Estimated Time	Owner
Create "Create Account", "Sign In", "Account Information", and "Sign Out" UI elements on the webpage	2	Hari
Create account creation process, client side forms	6	Akshit
Setup account creation process server side	5	Mike
Send email to confirm user registration	3	Simon
Account details should be saved in Postgresql database.	5	Mike

The tasks listed above were completed according to our acceptance criteria. Users are able to create accounts using their email, a password, and a password confirmation. Their emails are checked for their validity as potential emails, but not for whether or not they are real emails. The user registration emails are generated, but not currently sent, as sending requires our server code to be on Heroku. Once a user has registered their account, it is saved in our database, with a hashed password.

User Story: As a user, I want to be able to make a House

Task Description	Estimated Time	Owner
Before user has joined a house but after they've created an account, show "Build House" UI element.	2	Simon
Create House details in database, allow assignment of users to house	10	Derek

The above tasks were completed according to our acceptance criteria. Once the user has created an account, they are shown a page that requires them to create a house before having access to any features. The house creation page requires the user to enter a house name and house address (the address should not actually be required which will be handled in the next sprint). Upon successful completion of the house form, the information of the house is sent to the database. The user and the house are then linked using a join table allowing the page for house information to display the users living there and also for the user profiles to display the information of which house they belong to.

User Story: As a user, I want to be able to create and share notes with other users.

Task Description	Estimated Time	Owner
Create a button on user's dashboard.	2	Hari
Make the button functional to send requests to the controller, including create, delete, and edit.	6	Hari
Allow users to edit share properties of their notes.	8	Akshit
Setup dashboard to show multiple notes.	8	Akshit

Users are able to create, delete, and edit notes to be shared with the house. What is partially incomplete at this point, is their ability to edit who these notes are shared with. The backend code for creating these note relationships are there, we just need to add a filter and GUI that will allow users to pick and choose who to share notes with.

User Story: As a user, I want to be able to create and manage our inventory.

Task Description	Estimated Time	Owner
Create a button on user's dashboard.	2	Hari
Create the model for User and House inventories.	6	Derek
Make the button functional to send requests to the controller, including create, delete, and edit.	6	Hari
Allow users to edit values and quantities of their inventory items.	6	Akshit
Conglomerate individual inventories into the House inventory.	6	Akshit

Users are able to create, edit, and delete items which are then displayed on the inventory page. The inventory page contains all of the items of all of the users in the house. This page also displays the owners of the items as well as the cost and quantity of the items. Once the items are added, the user that owns the item is able to edit delete that item in the list.

What went wrong?

(Partially) Incomplete

Manage account details, changing password, profile photo.	10	Derek

We are able to manage account details, but we are missing some potential account details that we might want users to have, such as a profile photo. With our current implementation it should be easy to add these additional user details. What we do collect so far, their email, their name, and their password, have a page where users can edit this information.

Create an account recovery system	2	Duffy

The account recovery system was not completed this sprint. We have some design decisions to make still about how best to do the recovery system. One system would be to send the user an email with a link to a form where they update their password themselves. Another system would be to send the user a temporary password, which they use to login and then change their own password. This decision will be finalized and implemented for Sprint 2.

Allow User to invite other users via email to join their house.	6	Simon

We were successful in creating a email service to create emails. However as we are still running the project locally, we do not have an available SMTP server to actually send these emails. As for now these created emails are stored in /tmp/mail. We also successfully created the invitation system, so users can generate links to allow others users to join their house.

Give house admin the ability to remove other users	3	Duffy

We did not finish the House Admin page for editing the house information. We keep track of the house admin currently, so in the next sprint we will have to give the house admin a page in order to edit their house's information. This page will be very similar to the User Account information page, and should not take long now that we have done the similar task before.

How can we improve?

For the second sprint, we need to improve our team organization. To accomplish this, we are going to make better use of our Trello, taking advantage of their card labels and checklist features. As well, we are going to try and organize more group coding meetings. On the last two days of the sprint we spent many hours physically together working on code, which increased our productivity and understanding of the many different parts of the code.