CS 307

Software Engineering

Team 17

Bryan Duffy, Michael Reed, Simon Smith

Derek Schrag, Hari Mantena, Akshit Gudoor

Roomedy

Sprint One Planning Document

September 28th - October 16th

Sprint Overview

Overview

Below, we have listed user stories that we intend to complete this sprint. Our first sprint will focus on getting the backbone of the website up and functional and then begin to add the extra features that define our product. A large portion of the start of our project will be spent getting comfortable with the different tools we are using and getting accustomed to the working relationships between our team. We estimated the time to complete each user story based on a modified Fibonacci sequence of 0, 1/2, 1, 2, 3, 5, 8, 13, 20.

Scrum Master: Bryan Duffy

Scrum Meeting Time: 12:30, MWF.

Risks and Challenges

As our first sprint, there will be a lot of learning the technologies we chose to for our project and implementing the basic framework for our system. We understand that due to our inexperience there may be unforeseen consequences or problems with our design, which may delay some completion of the work listed below until Sprint Two.

Current Sprint Details

User Story: As a developer, we'd like to have a common development environment between all team members.

Task Description	Estimated Time	Owner
Setting up a Ruby on Rails Environment to serve web content.	1	Bryan
Setting up a basic Bootstrap HTML/CSS/JS website	2	Bryan
Assist team members in setting up identical development environments.	1	Bryan

Acceptance Criteria: All team members will be able to access a development environment, that includes access to the basic Bootstrap website and Ruby on Rails development tools. If we're able to view and interact with the website on a variety of devices and browsers then this user story will be implemented satisfactorily.

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
Manage account details, changing password, profile photo.	10	Derek
Account details should be saved in Postgresql database.	5	Mike
Create an account recovery system	2	Duffy

Acceptance Criteria: If this user story is implemented satisfactorily, then we will be able to successfully create and edit accounts without bugs.
Given a guest visitor to the site, when they choose to “Create Account”, then their account will have an entry created in our database with their given information.
Given a current user logged into the site, when they choose to “Edit Account Profile”, then their accounts information will be updated in our database.

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
Allow User to invite other users via Sendgrid email to join their house.	6	Simon
Give house admin the ability to remove other users	3	Duffy

Acceptance Criteria: If this user story is implemented satisfactorily, then we will be able to test the creation of a house and the invitation/removal of other users into said house.
Given a user account that is not bound to a House, When the user chooses to “Build a House”, then a new House will be created in our database, containing the user who created it.
Given a user account that is the owner of a House, when the user chooses to “Invite Roommates”, then an email will be sent with an invitation link to other potential users to have their new account created and added to the House.

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

Acceptance Criteria: If this user story is implemented satisfactorily, then we will be able to test the creation and viewing of notes with editing and viewing permissions correctly implemented and testable.
Given a logged in user who is part of a House, when the User chooses to “Add new Note”, then a new note will be created in our database, and displayed to all Users who are listed as having read or modify rights.
Given an already created note, when a user chooses to edit the contents of the note, then this change will be reflected in our database.
Given a user has read/modify rights on more than one note, their dashboard will be able to display each note in a way that aesthetically makes sense.

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	6	Hari

requests to the controller, including create, delete, and edit.		
Allow users to edit values and quantities of their inventory items.	6	Akshit
Conglomerate individual inventories into the House inventory.	6	Akshit

Acceptance Criteria: If this user story is implemented satisfactorily, then we will be able to test the creation, viewing, and editing of inventory lists.

Given a logged in user who is part of a House, when the User chooses to "Add Item to Inventory", then the user will be prompted to give the item a Name, Monetary Value, and Quantity, then this item will be put into our database in the user's list.

Given individual users who have created inventory lists, when a user updates his list, then the master list will reflect the same changes.

Given a user who has set an item not to show up on the master list, when the master list is updated, then the item will be excluded.

Backlog

Functional

- As a user, I want to be able to pay or request money from roommates.
- As a user, I want to be able to keep track of how much my roommates owe me
- As a user group, we want to be able to pay our bills by splitting up the total cost
- As a user, I want to be notified when bills are due
- As a user group, we want to be able to pool money together for group expenses
- As a user, I want to keep track of my monthly expenses and be given financial statistics
- As a user, I want all the above to be done with a familiar financial service
- As a user group, we want to have a common shopping list
- As a user group, we want an inventory list to keep track of food & supplies
- As a user group, we want both the grocery and inventory lists to have a value attached
- As a user, I want to be reminded of commonly purchased items
- As a user, I want to be suggested cheaper alternatives to my purchases
- As a user, I want events synced to my calendar
- As a user, I want to receive alerts about events
- As a user group, we want a common event calendar
- As a user group, we want financial notifications
- As a user, I want to opt out of group notifications
- As a user group, we want a rotating schedule of chores

- As a user group, we want a voting system for group decisions
- As a user, I want to be able to send private messages

Non-Functional

- As a user, I want consistent uptime of the service
- As a developer, I want to be able to deploy builds easily
- As a developer, we will require dedicated hosting
- As a sysadmin, I want to be able to scale the service easily
- As a user, I want a secure service to protect my account and payment information
- As a user, I want a simple and effective service
- As a developer, I want modularity to ease the creation of new features
- As a user, I want transactions to process quickly
- As a user, I want timely notifications