ApartMate

Team 3

Group members - Adrian Raj, Akshay Srinivasan, Ian Rettig, Risheek Narayanadevarakere, Siddhant Patel

Sprint 2 Retrospective

What went well?

This sprint was focused on building the group part of our app. We added most of the functionality that revolved around being in a group in our app. As this is a major feature, there were a lot of things that needed to be created. We started out working with the server and the calls that would be sent to it in order to create groups and send messages. After that, we went to the client side of things and started to make all the new views that would be required for the new information we wanted to display. Since we had all protocols for merging frontend and backend set up, integrating them was not too hard.

Current Sprint Detail

User Story 1:
As a user, I would like to send/receive group messages to roommates, so that I can communicate with my roommates.

Task Table:

Task Number	Description	Time	Task Owner
1	Create a chat within a group containing all the members within the group	4	Risheek
2	Store the messages in the database	3	Adrian
3	Automatic Testing - test for whether messages are being sent to and received by all the members in the group	3	Adrian

Completed:
We completed the group chat. The user is able to send and receive group messages. All the messages are stored in that database.

User Story 2:
As a user, I would like to view my roommate’s schedule, so that I can plan things accordingly.

Task Table:

Task Number	Description	Time	Task Owner
1	View the roommate’s schedule	4	Siddhant
2	Click on events and see the description	3	Adrian
3	Manual testing using 2 devices	1	Ian

Completed:
The user is able to view the roommate’s schedule. They are also able to click on the events to more information about the events. We also tested it with two devices.

User Story 3:
As a user, I would like to set up chores around schedules, so that I can share chores evenly.

Task Table:

Task Number	Description	Time	Task Owner
1	Set up chores	4	Akshay
2	Set the time and description for each chore	2	Akshay
3	Store these chores on the database	2	Ian
4	Unit test - Use two devices to see the chores	1	Ian

Completed:
We were able to add chores and have them show up on the schedule for the user. This was manually tested by us.

User Story 4:
As a user I would like to have a chore list on one of the group tabs, so that I can see them quickly.

Task Table:

Task Number	Description	Time	Task Owner
1	Set a table view for the list of chores	4	Akshay
2	View chores on the list	1	Akshay
5	Manual testing to see the chore list	1	Adrian

Completed:
The user is able to view the chores list in a table view. We manually tested this.

User Story 5:
As a user, I would like to add chores to the chore list, so that all the chores can be completed.

Task Table:

Task Number	Description	Time	Task Owner
1	See the current chore list	3	Risheek
2	Have a button that lets the user add a chore	2	RIsheek
2	Add a specific chore to it	2	Risheek
3	Manual testing	1	Adrian

Completed:
We are able to add chores to the current list. We have a button that let the user add the specific chore. We manually tested this.

User Story 6:
As a user, I would like to remove chores from the chore list.

Task Table:

Task Number	Description	Time	Task Owner
1	See the chore list and remove specific chores from it	3	Risheek
2	Have a button that lets the user add a chore.	2	Risheek
3	Remove a specific chore	2	Risheek
4	Manual testing	1	Ian

Completed:
We are able to remove chores from the current list. We have a button that let the user add the chore. The user can click on a chore that already exists to remove a chore. We manually tested this.

User Story 7:
As a user, I would like to add a new receipt within a group and control the members within the receipt, so that we can split bills.

Task Table:

Task Number	Description	Time	Task Owner
1	Add a button that lets us add a receipt	1	Siddhant
2	Add text fields to add the item and group	2	Siddhant
3	Pick which group member needs to pay for a certain item	2	Siddhant
4	Link the information to the database	3	Ian

Completed:
The user can add a new receipt, provide a title to the receipt, decide who it should be split between and adds the amount due for each person on the home screen. This was manually tested.

User Story 8:
As a user, I would like to see the total I am required to pay for the month, so that I can pay my share.

Task Table:

Task Number	Description	Time	Task Owner
1	See the total amount in one of the group tabs	2	Akshay
2	View changes to the amount if a bill has been paid	2	Akshay
3	View changes to the amount if a bill has been added	2	Akshay
4	Manual testing	1	Adrian

Completed:
The user is able to view the total amount the user to has to pay. If the amount is negative, it means the user needs to receive money. The user is able to view changes when a bill is added or paid. We manually tested this.

User Story 9:
As a user, I would like to add items to the grocery list, so that I can purchase them.

Task Table:

Task Number	Description	Time	Task Owner
1	View the current list	2	Risheek
2	Create fields to add items	3	Risheek
3	Send the item to the server and database	1	Ian
4	View the updated list.	2	Akshay

Completed:
The user is able to view the current list and add an item to it. The item is updated to the database through the server. And the user is able to view the updated list.

User Story 10:
As a user, I would like to remove items to the grocery list, so that no one purchases them by mistake.

Task Table:

Task Number	Description	Time	Task Owner
1	View the current list	2	Risheek
2	Add button to remove items from the list.	2	Akshay
3	Remove item from the database	1	Ian
4	Manually test that the items appear on the list	1	Ian

Completed:
We are able to remove grocery items from the current list. We have a button that let the user remove the grocery item. The user is able to see the updated list. We manually tested this.

User Story 11:
As a user, I would like to know my roommate’s interests, so that I can know what they enjoy doing.

Task Table:

Task Number	Description	Time	Task Owner
1	Add interest section to user profile UI	2	Akshay
2	Add page for adjusting interests	2	Akshay
3	Create message to send the information to the server to be stored on the database.	2	Adrian
4	Create message to retrieve the information from the server and display on profile.	2	Adrian
5	Manually test setting interest, seeing them on own profile, and on other user’s screen.	2	Adrian

Completed:
We were able to have an Interests button show up in a profile section in one of the group tabs. The user is able to click on the Interests button and view their roommate’s interests. This was manually tested.

User Story 12:
As a user, I would like to know what possessions of my roommate can be shared, so that I can use them.

Task Table:

Task Number	Description	Time	Task Owner
1	Add possessions to the list	3	Risheek
2	Remove possessions from the list	2	Risheek
3	View which possessions can be shared	3	Risheek
4	Manual testing	2	Ian

Completed:
We were able to have a Shared Possessions button show up in a profile section in one of the group tabs. The user is able to click on the Shared Possessions button and view which possessions of their roommates can be shared. This was manually tested.

User Story 13:

As a user, I would like to know birthdays of my roommates, so that I can celebrate with them.

Task Table:

Task Number	Description	Time	Task Owner
1	Add birthdays to profile	2	Siddhant
2	Birthday notification to roommates	3	Siddhant
3	View roommate’s birthday	1	Siddhant
4	Store birthdays on the database	1	Ian
5	Manual testing	1	Ian

Completed:
We were able to have a Birthdays button show up in a profile section in one of the group tabs. The user is able to click on the Birthdays button and view their roommate’s birthdays. This was manually tested.

User Story 14:

As a user, I would like to be able to log out of my account, so that other users can log in from my device.

Task Table:

Task Number	Description	Time	Task Owner
1	Logout of the app	2	Risheek
2	Clear current user’s session	4	Adrian
3	Notify the server	2	Risheek
4	Manual testing	2	Adrian

Completed:
We were able to have a Logout button in the options menu on the home screen. When the user clicks on it, it resets the local environment and directs the user back to the login page.

User Story 15:

As a user, I would like to create an apartmate group, so that everything related to the group can be accessed by all group members.

Task Table:

Task Number	Description	Time	Task Owner
1	Create a group	3	Akshay
2	Add members to the group	3	Ian
3	Give a title to the group	1	Akshay
4	Automatic Testing	2	Ian

Completed:
The user is able to press a floating action button on the home screen of the app and create a new group. When they press the button a dialog comes up asking for a group name and a button to submit. When they submit that, a new group is created with them as the member. We ran automatic tests to make sure this works.

User Story 16:

As a user, I would like to know normally how late my roommate comes home everyday, so that I can plan around it.

Task Table:

Task Number	Description	Time	Task Owner
1	Check my roommate's schedule and see his daily meetings/appointments/events	3	Siddhant
2	Manual testing	1	Ian

Completed:
We added time as a criteria for user scheduled events each day, for which the roommates can check their schedule and see what the last appointment for the day is. This was manually tested by us.

User Story 17:

As a user, I would like to have a “Do Not” section, so that I can set boundaries with my possessions and respect my roommates’ possessions.

Task Table:

Task Number	Description	Time	Task Owner
1	Remove possessions from the list	2	Risheek
2	Mark possessions which cannot be shared	3	Risheek
3	Store the selections/marking on the database	2	Adrian
4	Manual testing	2	Adrian

Completed:
We were able to have a Unshared Possessions button show up in a profile section in one of the group tabs. The user is able to click on the Unshared Possessions button and view which possessions of their roommates cannot be shared. They can add to this list by clicking a button and remove from the list by clicking on the possession. This was manually tested.

User Story 18:

As a user, I would like to have emergency contact information of my roommates, so that I know who to call in an emergency.

Task Table:

Task Number	Description	Time	Task Owner
1	Add emergency contact information to user profile page.	2	Siddhant
2	Add a page in group menu where all user's emergency contact information is displayed.	3	Siddhant
3	Add listener to collect information and send to the server and database for storage.	3	Ian
4	Be able to receive the contact information for everyone in a group from server and display in proper place.	3	Ian

Completed:

We were able to have a Emergency Contacts button show up in a profile section in one of the group tabs. The user is able to click on the Emergency Contacts button and view emergency contacts of their roommates. They can add to this list by clicking a button and remove from the list by clicking on the contact. This was manually tested.

User Story 19:

As a developer, I would like to document the process and code.

Task Table:

Task Number	Description	Time	Task Owner
1	Code reviews and coding standards	6	Adrian, Akshay, Ian, Risheek, Siddhant
2	Progress reports	6	Adrian, Akshay, Ian, Risheek, Siddhant
3	Planning Documents	6	Adrian, Akshay, Ian, Risheek, Siddhant
4	Code documentation	6	Adrian, Akshay, Ian, Risheek, Siddhant

Completed:

We commented our code and worked on planning documents, progress reports and also spent time reviewing code standards.

What did not go well?

The biggest thing that went wrong this sprint was when we tried to make fragments for the views in the client. This completely messed up our communication with the server. The client stopped waiting for the socket to be written to before continuing execution, rendering the communication useless. In order to get around this problem, we decided to move some of the functionality from the server to the client. Mainly the calls to the database for simple information was changed. It was directly integrated into the client in order to solve the issue communicating through sockets.

Another small issue that we only learned of during the review is that we did not complete one of the acceptance criterias for creating a group. We left out the ability to change the group name. This was simply an oversight.

How should you improve?

The easiest place for us to improve is to make sure that we complete every task that we set ourselves. We need to take special care to include testing of some sorts in every user story, and to make sure that we do everything that we say we are going to do for each user story.

Another place for us to improve is time management. For the past two sprints, we have somewhat left most of the merging and testing for the last day or two before the final review. We should try to complete everything a day or two ahead of time instead, so that any last minute bugs are not devastating. We attempted to this sprint, but we were unable to finish everything by the goal we had set. We will try to set a goal ahead of time again and try harder to meet it.