ApartMate

Team 3
Group members - Adrian Raj, Akshay Srinivasan, Ian Rettig, Risheek Narayanadeverekere, Siddhant Patel

Sprint 1 Retrospective

What went well?

Initially we hit a couple of roadblocks with our first design choice to use NativeScript (which is a cross platform development language), however we made the decision to switch to an Android application because none of our team members knew NativeScript and although we tried learning it, we were not able to do it fast enough to keep up with the sprint. We did not have anything concrete in terms of an app at the end of our first week, but we were still able to finish all our user stories for Sprint 1. Most of the team was well versed with Android app development and those that weren’t picked it up quickly during the course of the sprint.

User Story 1:
As a user, I would like to login, so that the app is person centered

Task Table:

Task Number	Description	Time	Task Owner
1	Create a login page	3	Siddhant
2	Encrypt the User’s password	1	Ian
3	Unit Test - correct and incorrect input, unknown email, incorrect password attempt limits	2	Risheek

Completed:
We completed the login page and encrypted the user password using our server. We stored each user’s encrypted password on our FireBase Database.

User Story 2:
As a user, I would like to register for an ApartMate account, so that I can use the app.

Task Table:

Task Number	Description	Time	Task Owner
1	Create a User Model on the Node.js backend	2	Adrian/Ian
2	Create User JSONs in Firebase	1	Adrian/Ian
3	Create Sign Up page in the UI	2	Akshay
4	Store the encrypted user data on the database	1	Adrian/Ian
5	Connect the frontend to backend and Firebase to complete the signup process	2	Adrian/Ian
6	Unit test with correct and incorrect input	2	Adrian/Ian

Completed:
We completed creating a register page to create a apartmate account. We were able to connect it with the backend securely. User was able to view the registration page and all the information was linked to the database for future use.

User Story 3:
As a user, I would like to send/receive messages to potential roommates,so that I can find a roommate

Task Table:

Task Number	Description	Time	Task Owner
1	Create a Message page	3	Risheek
2	Store time for messages received/sent	1	Adrian
3	Store message history on the device	2	Risheek
4	Unit test - Use two devices to see if messages are sent/received properly	1	Siddhant

Completed:
We completed the friend chat which stores the messages locally and on the database. The chats were being sent and received dynamically.

User Story 4:
As a user, I would like to change my password, so that I don't have to remember a complicated one from recovered password.

Task Table:

Task Number	Description	Time	Task Owner
1	Create a Reset password button in profile.	1	Akshay
2	Create a reset password form	2	Akshay
3	Create a reset password function in server.	2	Ian/Adrian
4	Update the database	1	Ian/Adrian
5	Test to see if the passwords get changed.	1	Ian/Adrian

Completed:
We completed this user story by having a reset password button in the user's profile page. We also updated the user's password on our database along with encrypting the new password.

User Story 5:
As a user, I would want to set themes for my account, so that I can reflect my world view.

Task Table:

Task Number	Description	Time	Task Owner
1	Create a toggle button in profile to switch between a dark and light theme	1	Risheek
2	Store the theme preference locally	4	Siddhant, Akshay
3	Unit test to check whether the themes are being switched properly	3	Siddhant, Akshay, Risheek

Completed:
After we created the UI for ApartMate, and frontend team set up a toggleable button that changed the colors to a different color theme. This choice is stored locally, so that when the application is closed and re-opened, the theme color is consistent.

User Story 6:
As a user I would like to verify my email, so that nobody else can use my email in that app.

Task Table:

Task Number	Description	Time	Task Owner
1	Verify user’s email	1	Adrian
2	Disallow users to use an already verified email	1	Siddhant
3	Unit test - Verify using a test email account	1	Risheek

Completed:
We were able to complete the verification feature. This made sure that no one else can use the user’s email and the user’s email is legitimate.

User Story 7:
As a user, I would like to upload profile picture, so that I can attract possible roommates.

Task Table:

Task Number	Description	Time	Task Owner
1	Put interface option on the app.	1	Akshay/Risheek
2	Create location to store the image on a file server.	2	Ian/Adrian
3	Link the database and server to the file server.	1	Ian/Adrian
4	Create data transfer to send the image to viewers of the account.	2	Ian/Adrian

Completed:
The user was able to upload picture using camera or gallery. The picture was stored in the database through the server. When the user opens the app again, the user is able to view the picture he/she had uploaded previously.

User Story 8:
As a user, I would like to recover my password, so that I do not need to create a new account every time I forget my password.

Task Table:

Task Number	Description	Time	Task Owner
1	Create a button for *Forgot Password*	1	Akshay
2	Create a page that asks for the registration email ID	1	Risheek
3	Verify that the email ID is valid and has an account associated with it	1	Ian/Adrian
4	Send an email to the user with a link to reset the password	1	Ian/Adrian
5	Store the new password in the database after encryption	1	Ian/Adrian
6	Test the feature and make sure the password can be reset and updates in the database	2	Adrian/Siddhant

Completed:
We created a button on the login page to allow the user to receive a temporary password in case he/she forgot the password to their account. Upon clicking this button the user will receive an email containing a randomly generated password which the user will have to login with before changing the password.

User Story 9:
As a user, I would like to make my schedule, so that my roommates can see it.

Task Table:

Task Number	Description	Time	Task Owner
1	Make schedule using Google calendar Api	4	Akshay
2	Create UI for the schedule	1	Risheek
3	Test the schedule and see if it updates on the UI component	1	Siddhant

Completed:
We completed this user story by having a schedule in each user’s profile page. The schedule showed a list of events that the user had scheduled. We used the Google calendar Api to implement the events on the calendar.

User Story 10:
As a user I would like to create a profile, so that I can personalize my account.

Task Table:

Task Number	Description	Time	Task Owner
1	Create the class/object to store the profile information on the server memory.	2	Ian/Adrian
2	Create relations in the database for the profiles.	2	Ian/Adrian
3	Create user interface for creating the profile.	1	Akshay
4	Create mechanism to send the profile information to the server	2	Ian/Adrian
5	Let server parse the information, update the database, and sent confirmation back to user.	1	Ian/Adrian

Completed:
The users were able to view a profile for themselves with basic information such as first name, last name and email populated from the their registration information along with a default profile picture to begin with.

User Story 11:
As a user, I would like to start a new chat, so that I can talk to someone that I know is using the app.

Task Table:

Task Number	Description	Time	Task Owner
1	Create button in message center to start a new chat.	1	Risheek/Siddhant
2	Create page to search for a user to chat with.	2	Risheek/Siddhant
3	Search the database for the user.	1	Ian/Adrian
4	Return confirmation if that user exists, and a button to start chat.	2	Ian/Adrian
5	Test to see if the search works and and starts a new chat with the user.	2	Ian/Adrian

Completed:
We created an option to search for a username that the current user knows is using the app. From that area, the user can start messaging the person if they are using the application. This part links to the sending a message feature in order to send the messages. We then made sure that it worked by searching for a user on a different account and starting to chat with them.

User Story 12:
As a user, I would like to edit my profile, so that I can reflect recent changes

Task Table:

Task Number	Description	Time	Task Owner
1	Create UI to edit the use profile	6	Siddhant, Akshay, Risheek
2	Update the database with the updated profile changes	2	Adrian
3	Unit Test to see if the users are able to make changes to their profiles	1	Siddhant

Completed:
The users are able to edit their personal information in the profile and including “Latest Achievement” and “Greatest Achievement” fields and also have and option to change their profile picture.

User Story 13:

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	2 each	Adrian, Akshay, Ian, Risheek, Siddhant
2	Progress reports	2 each	Adrian, Akshay, Ian, Risheek, Siddhant
3	Planning Documents	2 each	Adrian, Akshay, Ian, Risheek, Siddhant
4	Code documentation	2 each	Adrian, Akshay, Ian, Risheek, Siddhant

Completed:

To make sure anyone can understand our code, we documented our code. And, if anyone wants to view our progress, they can look at our progress report and planning documents.

What did not go well?

Generally, the planning of the sprint 1 was a bit of a mess. We needed to reorganise our user stories after a week and a half as we realised that some of the user stories involved parts of the app that we hadn’t built yet or were planning on building in the same sprint. The specific example is that we were making features that depended on the apartment group existing before we had created the groups in our program.

Another part that was a mess was our decision to use NativeScript to build the app. We spent a week trying to learn how to use it before deciding to change to Android only development. Unfortunately, this wasted a lot of time and put us in a time crunch in order finish the sprint on time.

How should you improve?

We need to spend more time in deciding our design choices to know what will work and what will not work depending on the current architecture. Spending time in research and trying to figure out the pros and cons of using a particular library or framework can go a long way with the execution of the user stories.

Another thing we can improve from the first sprint planning is estimating the amount of time it would take for us to work on user stories, and tasks within it. We underestimated the time it would take us for certain user stories and that can be improved on now that we have a better idea of our actual implementation and overall architecture.

Finally, we need to be more careful in deciding what user stories we pick for each sprint keeping in mind that we don’t choose to implement user stories that would require functionalities that we haven’t yet implemented as a part of our user stories.