Team 12 - CS 307 Sprint 2 Retrospective

Project Name: Capsule

Nathan Schneider, Kevin Jones, Evan Zimmerman, Peter Henwood

What went well?

During this sprint we finished the vast majority of our user stories in exactly the way we intended before the sprint started. We took the skeleton of our app and turned it into a functional product with tons of new working features. Some of these features include additional profile settings, completed capsule generation (with all the features from the planning document), and completed storyboard/history pages.

User Stories

User Story #1

(3) As a user, I would like to be able to pick a profile picture during registration, so that I can better identify myself on the platform.

#	Description	Estimated Time	Owner
1	Create add picture page for registration, and create add picture button	1 hr	Evan
2	Implement photo selection	2 hrs	Evan

3	Save that photo in MongoDB	1 hr	Evan
4	Implement profile photos throughout app	2 hrs	Evan
5	Debug and complete unit testing for selecting a picture	2 hrs	Evan

Completed: Fully integrated the functionality to allow users to pick and upload a profile picture during registration. This included adding the selection option to the registration page, implementing photo selection, saving the photo to MongoDB, and ensuring profile pictures are consistently displayed throughout the app.

User Story #2

(6) As a user, I would like to be able to delete my account, so that I can remove my information from the platform.

#	Description	Estimated Time	Owner
1	Add delete account button profile page	1 hr	Nathan
2	Delete account from database	3 hrs	Nathan
3	Navigate user back to main page	1 hr	Nathan
4	Delete user data from all friend pages	2 hrs	Nathan
5	Debug and complete unit testing to ensure user data is correctly deleted	2 hrs	Nathan

Completed: Successfully implemented a feature that enables users to delete their accounts. This involved adding a delete account button to the profile page, removing the account from the database, navigating users back to the main page, ensuring deletion of user data from all friend pages, and conducting thorough debugging and testing to confirm the accurate removal of user data.

User Story #3

(9) As a user, I would like to be able to link my spotify, link my instagram, and change my profile picture, on the ‘my profile’ page, so that I can update any information linked to my account.

#	Description	Estimated Time	Owner
1	Add those buttons corresponding with each functionality in profile page	1 hr	Peter
2	Create API call to link Spotify	2 hrs	Peter
3	Create API call to link Instagram	2 hrs	Peter
4	Create option for user to change profile picture	3 hrs	Peter
5	Debug and complete unit testing for aspects of the profile page	2 hrs	Peter

Completed: Added functionality for users to link their Spotify and Instagram accounts and change their profile pictures from the 'my profile' page. This involved creating buttons for each action, establishing API calls for Spotify and Instagram linking, and providing an option for profile picture updates.

User Story #4

(16) As a user, I would like to have a history page where I am able to see previous capsules, so that I can reflect on previous months.

#	Description	Estimated Time	Owner
1	Create History page which can be navigated too from profile page	2 hrs	Evan
2	Save all capsules into MongoDB	2 hrs	Evan
3	Display all past capsules in scrollable list	2 hrs	Evan
4	Check functionality in MongoDB and on interface stays up to date	1 hr	Evan
5	Add back button and swipe feature to return to profile page	1 hr	Evan
6	Debug and complete unit testing for history page capsules	2 hrs	Evan

Completed: Developed a history page where users can view their past capsules, from previous months. This included creating a history page accessible from the profile page, saving capsules in MongoDB, displaying past capsules in a scrollable list, ensuring up-to-date functionality, adding navigation features, and conducting debugging and testing.

User Story #5

(17) As a user, I would like to be able to click on a snapshot to enlarge it, so that I can view more options regarding the snapshot.

#	Description	Estimated Time	Owner
1	Add page for enlarged snapshot	2 hrs	Evan
2	Add a way for user to return to scrollable history page	2 hrs	Evan
3	Resize picture to fill as much of the screen as possible	4 hrs	Evan
4	Debug and complete unit testing for enlarging snapshot	2 hrs	Evan

Completed: Added functionality for snapshots to be clicked and enlarged for a more detailed view. This involved adding a dedicated modal for enlarged snapshots, implementing navigation back to the history page, resizing pictures for optimal screen use, and debugging and testing to ensure it works properly.

User Story #6

(23) As a user, I would like the Story Board to be reset every month, so that I can only view the most recent capsules.

#	Description	Estimated Time	Owner
1	Develop Mechanism to clear Story Board	3 hrs	Peter
2	Display the most recent capsule for all friends that have shared their new capsule	2 hr	Peter

3	Automatically clear Story Board once the countdown hits 0	4 hrs	Peter

Completed: Implemented a mechanism to automatically reset the Story Board every month, ensuring that users only see the most recent capsules. This feature included developing the reset functionality, displaying the latest capsule for friends, and automatically clearing the Story Board at the end of the countdown (done by only displaying capsules from the current month). Debugging and testing ensured reliable performance.

User Story #7

(25) As a user, I would like to be able to view my snapshot at the end of the month, so that I can reminisce on the past month

#	Description	Estimated Time	Owner
1	When the countdown reaches 0, trigger an event to begin photo choosing.	3 hrs	Nathan
2	Display a “generating capsule” message on the main page while the algorithm creates the capsule.	2 hrs	Nathan
3	Combine all chosen photos and the song into a presentable capsule that can be displayed.	10 hrs	Nathan
4	Display generated capsule in main page.	2 hrs	Nathan
5	Debug and complete unit testing for generating capsules.	3 hrs	Nathan

Completed: Enabled users to view their snapshots at the end of the month, offering a moment of reflection on past experiences. This required triggering an event when the countdown reaches zero, displaying a generating capsule message, combining chosen photos and the song into a presentable capsule and displaying the capsule on the main page.

User Story #8

(26) As a user, I would like my snapshot to have my most played song of the month, so that I have another reminder of the month.

#	Description	Estimated Time	Owner
1	Retrieve Spotify refresh token from user's access token	2 hrs	Kevin
2	Retrieve the user's most played song from the user's refresh token.	3 hrs	Kevin
3	Write MongoDB requests to post and delete spotify songs from an associated capsule	2 hrs	Kevin
4	Debug and complete unit testing for retrieving most played spotify song	3 hrs	Kevin

Completed: Integrated Spotify to include users' most played songs of the month in their snapshots, adding a personal music touch to their reflections. This involved retrieving Spotify tokens, fetching the most played song, writing MongoDB requests for song management, and debugging and testing for reliable music retrieval.

User Story #9

(28) As a user, I would like Time Capsule to intelligently choose photos for my snapshot, such as landscapes and people smiling, so that I have quality photos for my capsule.

#	Description	Estimated Time	Owner
1	Implement functionality to collect photos from the month from the user's camera roll.	5 hrs	Kevin
2	Figure out how to upload photos to Google's Vision API and then retrieve the JSON file	8 hrs	Kevin
3	Write an algorithm to search the JSON file for keywords dictating a quality photo. This	6 hrs	Kevin

	would include people smiling, photos of nature, etc.		
4	Debug and complete unit testing for intelligently choosing snapshot photos	4 hrs	Kevin

Completed: Implemented intelligent photo selection for snapshots using Google's Vision API, ensuring high-quality photos in capsules. This involved collecting photos, interfacing with the Vision API, analyzing the JSON for quality indicators like smiles and nature scenes, and thorough debugging and unit testing for accurate photo selection.

User Story #10

(30) As a user, I would like my snapshot to not contain identical pictures, so that I have a variety of photos on my snapshot.

#	Description	Estimated Time	Owner
1	Write an algorithm to analyze the JSON file from Google Vision API and compare it to already chosen photos.	6 hrs	Peter
2	Debug and complete unit testing to verify there are no identical pictures.	2 hrs	Nathan

Completed: Developed an algorithm to prevent identical pictures in snapshots, ensuring a variety of photos. This included analyzing the JSON file from Google Vision API, comparing chosen photos, and conducting debugging and testing to verify the absence of duplicates.

User Story #11

(31) As a user, I would like my snapshot to pick default photos / songs if I have not taken enough pictures / linked my spotify account , so that I always have enough information for the capsule.

#	Description	Estimated Time	Owner

1	Organize a collection of photos in MongoDB to choose from if enough quality photos can not be found.	2 hrs	Nathan
2	Organize a collection of songs to choose from if Spotify songs can not be found.	1 hr	Evan
3	Implement functionality to retrieve default photos from MongoDB in the case that user data is insufficient.	2 hrs	Evan
4	Debug and complete unit testing for retrieving default information if there isn't enough information from the user.	2 hrs	Evan

Completed: Ensured a complete snapshot by picking default photos or songs when user data was insufficient. This involved organizing default photos and songs in MongoDB, implementing functionality to retrieve these defaults, and debugging and testing to guarantee a full snapshot experience.

User Story #12

(34) As a user, I would like to be able to share my snapshot to the Story Board, so that my friends can see it.

#	Description	Estimated Time	Owner
1	Add a publish button with good styling to the Main Page.	1 hr	Peter
2	When the publish button is pressed, it adds the user's current capsule to the storyboard for all friends to see.	2 hrs	Peter
3	Write MongoDB requests to update the published status of the selected capsule.	1 hr	Peter
4	Debug and complete unit testing for sharing snapshots to the Story Board.	2 hrs	Peter

Completed: Facilitated sharing snapshots to the Story Board, allowing friends to view each other's capsules. This included adding a publish button, updating the

storyboard with the current capsule upon publishing, writing MongoDB requests for published status updates, and debugging and testing for smooth snapshot sharing.

What did not go well?

For this sprint, we completed almost everything we set out to accomplish. Similar to last sprint, the only thing we weren't able to fully complete is linking Instagram accounts. This is only because of a problem with Facebook's API, and unfortunately we can't do anything to fix it for now. Linking the instagram account isn't a big part of our project at all, so it doesn't actually set us back.

I think the main thing that didn't go great this sprint is our time management. Many of the User Stories took us a lot longer than we expected them to take, and we also all had several exams and other projects due during this sprint. This made it difficult for us to spread out the work evenly over the course of the sprint, resulting in a lot of the functionality being implemented in the final week.

In terms of complete user stories, there weren't any that we didn't finish during the sprint.

How should we improve?

As mentioned in the previous section, time management was a slight issue with this sprint. Most of it was out of our control though, so for the most part I think we did a good job working through the sprint considering our other commitments. However, we can still improve on this by setting specific story completion goals for each week. This will be better than setting aside blocks of time, since we can quantify how much progress we are making per week based on the number of user stories completed.

Another cause of our time management issues were running into trouble with some of the technology for our app. We are using something called Expo Go to test our app on our phones, and this service limits some of the normal libraries that we would be able to use. This made it hard to implement several user stories, and it took longer than we expected. This issue will also most likely be fixed for the next sprint because the vast majority of the remaining user stories don't require significant new libraries to be used.

I think this sprint was a great success, and besides the couple things we mentioned, for the most part we plan on replicating our flow from last sprint to hopefully have a smooth final sprint!