SNAPSHOTS

Sprint 1 Planning Document

Group 28: Alex Goswick, Rachel Hawley, Samara Saquib, Evan Shaw

Sprint Overview

During this sprint, we aim to create a very basic version of the desktop app with the basic functionality from our user stories. These include creating an account, being able to upload photos, being able to view the photo library, sorting methods for easy browsing of photos, photo sharing, and photo library editing. These will need modification and debugging in the future as more features are added in future sprints.

Scrum Master: Rachel Hawley

Meeting Plan: Our group will have biweekly meetings on Wednesdays at 4pm and Saturdays at 3pm.

Risks and Challenges:

The main challenge with this sprint will be learning the techstack as we go. This will make it difficult to implement features in a timely manner and could be a setback in our progress. This will make future sprints more labor intensive, so it is important that we take the time to implement our user stories correctly now to avoid future problems.

Current Sprint Detail

User Story #1

As a user, I would like to be able to upload photos from the desktop app.(7)

#	Description	Estimated Time	Owner
1	Create UI for uploading photos	3 hours	Rachel
2	Create an algorithm for adding photos to the database	10 hours	Rachel
3	Debug and test algorithm using unit tests	5 hours	Rachel

Acceptance Criteria:

- Given that the connection between the Photo Library page UI and the Upload page UI is implemented correctly, when the user clicks on the “upload photo” tab, the UI for uploading photos is displayed.
- Given that the UI for uploading photos is implemented correctly, when the user attempts to upload a photo, the UI will display a success or error message.
- Given that the Upload page UI is implemented correctly, when the user receives a success message, the UI will return to the Photo Library page with an updated display to include the newly uploaded photo.

User Story #2

As a user, I would like to be able to view photos from the desktop app.(8)

#	Description	Estimated Time	Owner
1	Create UI for photo library	3 hours	Rachel
2	Create an algorithm for retrieving photos from the database and displaying them in the UI	12 hours	Rachel
3	Debug and test algorithm using unit tests	5 hours	Rachel

Acceptance Criteria:

- Given that the connection between the Home page UI and the Photo Library page UI is implemented correctly, when the user clicks on the “photo library” tab, the UI for the photo library is displayed.

- Given that the algorithm for retrieving photos from the database is implemented correctly, the UI should display the photos for viewing and browsing.
- Given that the UI for the photo library is implemented correctly, when the user attempts to view more photos than can fit on the screen, they should be able to scroll down the page to view the rest of the photos.

User Story #3

As a user, I would like the photos to be stored in a central location, and accessible at any time from any computer. (1)

#	Description	Estimated Time	Owner
1	Set up the schema for the SQL database	2 hours	Evan
2	Define the client-facing API for the server	3 hours	Evan
3	Define basic server architecture + multi-user behaviors	4 hours	Evan
4	Implement basic arch. + API	12 hours	Evan
5	Debug and write test cases	6 hours	Evan

Acceptance Criteria:

- Given that the database schema is set up, the server should be able to interact with the database on the basic data objects of the application
- Given that the client-facing API is defined for the server, there should be a consistent model on paper for client apps to interface with the server. Group members should be able to use this to build the client before all API features are implemented
- Given that the basic server architecture is defined, and multi user behaviors are defined, there should be a model on paper that details how the server will interact with the clients and the database
- Given that the basic server architecture and client facing API is implemented, the client app should be able to interact with the server based on the previously defined API specification
- Given that all these tasks are correctly done, photos should be able to be stored and retrieved from the server

User Story #4

As a user, I would like the system to be easy to set up with minimal install. (4)

#	Description	Estimated Time	Owner
1	Learn about Docker	2 hours	Evan
2	Put together a basic Docker container with the server	1 hours	Evan
3	Implement a continuous integration pipeline using (Jenkins?) and Docker	8 hours	Evan
4	Debug CI pipeline	4 hours	Evan

Acceptance Criteria:

- Given that a continuous integration pipeline has been set up, changes made to the server code should trigger a build and deployment process
- Given that the build process is functional, the server should be automatically packaged in a Docker image for easy installation
- Given that the CI pipeline is functional, a Jenkins interface should be functional containing the past and present builds

User Story #5

As a user, I would like to be able to browse photos by date from the desktop app.(11)

#	Description	Estimated Time	Owner
1	Create UI button in the Photo Library UI for sorting by date	2 hours	Samara
2	Create an algorithm that sorts photos by date	6 hours	Samara
3	Debug and test the algorithm with unit tests	4 hours	Samara

Acceptance Criteria:

- Given that the “Sort” button in the sidebar is properly programmed, when the user clicks on the “Sort” button, an option for “Sort by Date” will be available
- Given that the date sorting algorithm is correctly written, when the user selects “Sort by Date,” the photos will be arranged in order by their date, with most recent photos at the top
- Given that the photos that are uploaded contain metadata with the date they were *taken*, the photos will be sorted by the date they were taken rather than their upload date

User Story #6

As a user, I would like to be able to browse photos by geolocation from the desktop app.(12)

#	Description	Estimated Time	Owner
1	Create UI button in the Photo Library UI for sorting photos by location into their own albums	3 hours	Samara
2	Create an algorithm that sorts photos by geolocation	8 hours	Samara
3	Debug and test the algorithm with unit tests	4 hours	Samara

Acceptance Criteria:

- Given that the “Sort” button in the sidebar is properly programmed, when the user clicks on the “Sort” button, an option for “Sort by Location” will be available
- Given that the “Sort by Location” button is set up correctly, when the user clicks on “Sort by Location,” the photos will be sorted into albums with their locations as the title of the album.
- Given that the photos are sorted into their appropriate location albums, the location albums will be sorted in alphabetical order by their location name

User Story #7

As a user, I would like to be able to view videos from the desktop app.(13)

#	Description	Estimated Time	Owner
1	Create UI for displaying videos	3 hours	Alex
2	Create algorithm for uploading videos	10 hours	Alex
3	Create algorithm for retrieving videos from the database and displaying them on the UI	10 hours	Alex
4	Debug and test the algorithms with unit tests	5 hours	Alex

Acceptance Criteria:

- Given that the UI for displaying videos is implemented correctly, the user should be shown their video library when selecting to view videos
- Given that the algorithm for uploading videos is implemented correctly, the user should be given a success or error message when a video is uploaded
- Given that the algorithm for retrieving videos is implemented correctly, videos should be populated in the video library when the UI is shown.

User Story #8

As a user, I would like to be able to delete photos and videos from the desktop app.(14)

#	Description	Estimated Time	Owner
1	Create a “Select” button to select multiple photos or videos	5 hours	Samara
2	Create a “Delete” button for the user to delete the selected photos	5 hours	Samara
3	Create an algorithm to allow users to right click on specific photos and then delete them	5 hours	Samara
4	Create an algorithm for removing photos from the database	5 hours	Samara
5	Create an algorithm for removing videos from the database	5 hours	Samara
6	Reorder the photos and videos after deletion	5 hours	Samara
7	Debug and test algorithms using unit tests	4 hours	Samara

Acceptance Criteria:

- Given that the Home page of the photo library displays all photos, a “Select” button will be available on the sidebar for the user to select multiple photos
- Given that the “Select” button is properly implemented, when the user clicks “Select,” the user can select multiple photos or videos at once
- Given that the “Select” button can successfully allow the user to select multiple photos or videos, the icon for each selected file will be highlighted with an outline to illustrate that it is selected
- Given that the “Delete” button is properly programmed, the user can click on the “Delete” button to delete the selected files

- Given that the “Delete” button is properly programmed, when the user hits “Delete,” a confirmation message will appear on the screen asking the user to confirm the deletion
- Given that users will be given the ability to right click on photos to see options for that photo, when the user right clicks on a photo, an option to “Delete” that photo directly will appear in a small menu
- Given that users can right click on photos to see options, when the user clicks “Delete,” it will call the delete function and display the same confirmation message
- Given that the “Delete” function is successful, the photos and videos on the Home page will be reordered to fill up the spaces left by the deleted files

User Story #9

As a user, I would like to be able to create and edit albums/folders of photos that are shared with authenticated users.(20)

#	Description	Estimated Time	Owner
1	Create UI for creating albums of photos	3 hours	Alex
2	Create UI for displaying photo albums	3 hours	Alex
3	Create an algorithm for adding photos to an album	10 hours	Alex
4	Create an algorithm for removing photos from an album	8 hours	Alex
5	Debug and test algorithms using unit tests	5 hours	Alex

Acceptance Criteria:

- Given that the UI for displaying albums is implemented correctly, when selecting the album tab, the user should be shown all of their albums
- Given that the UI for creating albums is implemented correctly, when selecting the corresponding button the user should be shown options for creating an album
- Given that the algorithm for adding photos to albums is implemented correctly, the user should receive a confirmation when adding photos to an album
- Given that the algorithm for removing photos is implemented correctly, the user should receive a confirmation when removing photos from an album

User Story #10

As a user, I would like the metadata formats of many photos to be recognized.(10)

#	Description	Estimated Time	Owner
1	Research photo metadata formats	2 hours	Evan
2	Implement a standard metadata format in the database	3 hours	Evan
3	Write server code to extract and parse data from EXIF	8 hours	Evan
4	Write test cases for metadata parsing	4 hours	Evan

Acceptance Criteria:

- Given the successful implementation of the metadata parsing, when a user uploads a photo, it should be searchable/organized by metadata
- Given the successful implementation of the metadata parsing, the date taken and gps location should be stored in the database,
- Given the successful implementation of the metadata parsing, the metadata between different file formats should be standardized in the database

User Story #11

As a user, I would like to share photos with other people.(33)

#	Description	Estimated Time	Owner
1	Create UI for sharing photos with other users	3 hours	Alex
2	Create algorithm for adding photos to another user's photo library	10 hours	Alex
3	Debug and test algorithm using unit tests	5 hours	Alex

Acceptance Criteria:

- Given that the UI for sharing photos is implemented correctly, the user should be given fields to input recipients and photos to be shared with others.
- Given that the algorithm for adding photos to another user's library is implemented correctly, the user should be given a confirmation or error when attempting to add photos
- Given that the algorithm for adding photos to another library is implemented correctly, the user who received the photos should get a notification that photos were added

User Story #12

As a user, I would like to be able to create and login into an account. (36)

#	Description	Estimated Time	Owner
1	Create UI for login upon opening the app	3 hours	Rachel
2	Create UI for creating an account	3 hours	Rachel
3	Create UI for password recovery	3 hours	Rachel
4	Create an algorithm for storing account data in the database	10 hours	Rachel
5	Create an algorithm for retrieving account information and authenticating a login	10 hours	Rachel
6	Create an algorithm for retrieving account information if password is forgotten	10 hours	Rachel
7	Debug and test the algorithms with unit tests	5 hours (each)	Rachel

Acceptance Criteria:

- Given the UI for login is correctly implemented, when the user opens the application, then they should be shown the Login page UI.
- Given the connection between the Login page UI and Home page UI is implemented correctly, when the user attempts to login, then the Home page UI is displayed.
- Given the algorithm for retrieving account information and authenticating a login is implemented correctly, when the user attempts to login, then an error message or a success message is displayed.
- Given the connection between the Login page UI and the Forgot Password page UI is implemented correctly, when the user presses the “forgot password” button, the Forgot Password page UI will be displayed.
- Given the algorithm for retrieving account information if the password is forgotten is implemented correctly, when the user forgets their password and enters in their email address, then they will receive an email with instructions on changing their password.

Developer	**Number of Hours**
Alex Goswick	75 hours
Rachel Hawley	82 hours
Samara Saquib	61 hours
Evan Shaw	59 hours

Remaining Backlog

Functional Requirements:

1. ~~As a user, I would like the photos to be stored in a central location, and accessible at any time from any computer.~~
2. As a user, I would like to be able to set up the system on my own computer and not be tied to a cloud service.
3. ~~As a user, I would like the system to be easy to set up with minimal install.~~
4. As a user, I would like this data to be password protected and available to authenticated users.
5. As a user, I would like to be able to have a library that can scale to 10+ terabytes.
6. As a user, I would like to be able to use the desktop app on multiple operating systems.
7. ~~As a user, I would like to be able to upload photos from the desktop app.~~
8. ~~As a user, I would like to be able to view photos from the desktop app.~~
9. ~~As a user, I would like the metadata formats of many photos to be recognized.~~
10. ~~As a user, I would like to be able to view the entire metadata from a single photo in the desktop app.~~
11. ~~As a user, I would like to be able to browse photos by date from the desktop app.~~
12. ~~As a user, I would like to be able to browse photos by geolocation from the desktop app.~~
13. ~~As a user, I would like to be able to view videos from the desktop app.~~
14. ~~As a user, I would like to be able to delete photos and videos from the desktop app.~~
15. As a user, I would like to be able to fix red-eye effects from the desktop app.
16. As a user, I would like to be able to adjust the color curves from the desktop app.
17. As a user, I would like to be able to retouch photos from the desktop app.
18. As a user, I would like to be able to edit the photos without changing the original photo (non-destructive) from the desktop app.
19. As a user, I would like to be able to undo and redo edits of photos.
20. ~~As a user, I would like to be able to create and edit albums/folders of photos that are shared with authenticated users.~~
21. As a user, I would like to be able to view photos from a mobile device.
22. As a user, I would like to be able to manually upload photos from a mobile device.
23. As a user, I would like to be able to automatically upload photos from a mobile device in the background without any user intervention.
24. As a user, I would like the system to automatically recognize duplicate files and stop their upload to avoid clutter.
25. As a user, I would like the system to automatically recognize near duplicate files (like barely cropped files) and stop their upload to avoid clutter.

26. ~~As a user, I would like to be able to download photos in multiple file formats.~~
27. As a user, I would like the system to load quickly and be responsive.
28. As a user, I would like the system to support Apple Live Photos (JPEG + HEIC).
29. As a user, I would like to be able to easily backup my photos.
30. As a user, I would like this backup system to keep track of the state of the photo library at different dates and be able to roll back to a specific date in time.
31. As a user, I would like this backup system to not take up too much more space than the actual photo library.
32. As a user, I would like the system to be compatible with third-party apps that anyone could write.
33. ~~As a user, I would like to share photos with other people.~~
34. As a user, I would like to enable multi-factor authentication.
35. As a user, I would like to share photos directly to social media applications.
36. ~~As a user, I would like to store an email and phone number with my account in case I forget my password.~~
37. ~~As a user, I would like to be able to create and login into an account~~

Non-Functional Requirements:

1. Interface should have a simple layout and be easy to understand and navigate
2. Application should be usable on a desktop computer
3. (If time allows) Should be able to access photos on Android products
4. (If time allows) Should be able to access photos on Apple products