****

**Probos**

Team 14 - Sprint 3 Planning Document

*Alexa Baker, Daniel Raftery, Steven Daniel Vaughan, and Michael Woods*

**Sprint Overview**

During this sprint, we hope to wrap up the handful of incomplete stories that we have amassed over the course of the project and apply polishing touches to the app, while also still implementing a small number of additional features. In general, we will be adding List features, along with viewing the user’s own favorited messages and pinning messages to the top of their profile. On the client side, we will add customizability in the form of choosing the app’s primary and secondary color scheme.

**Scrum Master:** Michael Woods

**Meeting Plan:** Mondays/Wednesdays @ 7:00 PM

**Risks and Challenges**

Our group has a fair number of user stories that are either incomplete or buggy, so we will have to be sure to take those into consideration when deciding how much new content we wish to handle for this new sprint. Overwhelming ourselves with too much new work when we have old features to finish wrapping up would simply be unwise, especially considering that this is the final sprint, as we want to be sure that the work we do is finalized and presentable. Additionally, with the lack of helpful documentation for our API library, it is important that further research should be completed in search of answers, in particular for media submission requests. Finally, we will need to ensure that enough time is put towards establishing automated testing for demonstration.

**Current Sprint Detail**

1. As a user, I want to view a list of the “toots” I have favorited.
   1. Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 1 | Create a UI and button to display favorited “toots” | 3 hours | Alexa |
| 2 | Connect UI elements to API calls to retrieve statuses | 2 hours | Alexa |
| 3 | Test favorite list updating with new/removed favorited statuses | 0.5 hours | Alexa |

Acceptance Criteria:

* Given the UI is implemented correctly, when a user selects the Favorites option, a list of favorited messages will be displayed.
* Given the API calls are made correctly, when a user favorites/unfavorites a message, the list of their favorites will update when refreshed.
* Given the UI is implemented correctly, when the user leaves and re-enters the Favorites list, it should update and not use a cached list.

1. As a user, I want to be able to pin any of my “toots” to the top of my profile.
   1. Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 1 | Add another button to the status element UI for pinning; display only if the user owns the status | 3.5 hours | Steven |
| 2 | Connect button to a custom API call to have it pinned to the user’s profile | 6 hours | Steven |
| 3 | Test that pin/unpin requests are processed successfully | 0.5 hours | Steven |

Acceptance Criteria:

* Given the UI is implemented correctly, when the app displays one of the user’s own statuses, the pin button should be displayed somewhere on the status.
* Given the UI is implemented correctly, when the app displays a status not authored by the user, the pin button should not be visible or accessible.
* Given the API calls are made correctly, when the user presses the pin button on a status, its state as pinned/unpinned to their profile should be updated appropriately.

1. As a user, I want to be able to only view my “Lists”.
   1. Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 1 | Create a UI for a List page to display a user’s created Lists | 3 hours | Steven |
| 2 | Create a UI for statuses of a selected List to be displayed in | 3 hours | Steven |
| 3 | Test that lists retrieve statuses from the users in a List correctly | 0.5 hours | Steven |

Acceptance Criteria:

* Given the UI is implemented correctly, when a user selects the List option they will see the names of all their lists (if any).
* Given the UI and API calls are implemented correctly, when a user has selected one of their lists, it will display the messages of users in the list.
* Given the UI is implemented correctly, when the user refreshes the list message view, the previously-loaded messages are displayed below the newly-received messages (if any).

1. As a user, I want to create new “Lists”.
   1. Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 1 | Add a button to List UI and create a popup to name a List | 1 hour | Daniel |
| 2 | Connect button to popup and connect popup input to API calls to create a new list | 2 hours | Daniel |
| 3 | Test that new lists are created successfully, and with at least a one character name. | 0.5 hours | Daniel |

Acceptance Criteria:

* Given the UI is implemented correctly, when a user does not add a name or cancels the new list creation, no list should be created.
* Given the UI and API calls are implemented correctly, when a user confirms a list with a name at least one character long, a new list will be created.
* Given the UI is implemented correctly, when a user closes the app mid-creation of a list, no list should be created.

1. As a user, I want to edit “Lists.”
   1. Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 1 | Add button in list view to edit list and a UI to add/remove users to/from the list | 2 hours | Alexa |
| 2 | Connect UI to API calls to add/remove selected users to a list | 2 hours | Alexa |
| 3 | Test that users are added/removed from lists | 0.5 hours | Alexa |

Acceptance Criteria:

* Given the UI is implemented correctly, if a user does not confirm changes to the list no changes to the list should be made.
* Given the UI is implemented correctly, if a user saves the changes to the list any users added to the list will have all future messages appear in the list
* Given the UI is implemented correctly, if a user saves changes to the list any users removed from the list will have any messages in the list removed

1. As a user, I want to delete “Lists.”
   1. Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 1 | Add delete button and confirmation popup to List view UI | 3 hours | Steven |
| 2 | Connect delete button to API to remove selected List | 2.5 hours | Steven |
| 3 | Test that Lists are removed successfully | 0.5 hours | Steven |

Acceptance Criteria:

* Given the UI and API calls are implemented correctly, if a user confirms the deletion of a list, it should be removed.
* Given the UI is implemented correctly, if a user cancels the deletion of a list, no action will happen and the list should not be removed.
* Given the UI is implemented correctly, if the app session is closed before a selection is made, no action should be taken and the list should not be removed.

1. As a user, I want to customize the primary and secondary color schemes of the app’s user interface.
   1. Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 1 | Add customization option to dropdown menu and create UI for color schemes | 2 hours | Alexa |
| 2 | Connect code to the UI to apply user changes when confirmed | 3 hours | Alexa |
| 3 | Implement a “Reset” button to revert any applied user changes to the app’s original state | 1 hour | Alexa |
| 4 | Test that the various display elements the user can modify update correctly | 0.5 hours | Alexa |

Acceptance Criteria:

* Given no changes to the customization settings have occurred, the app should display in the current default dark blue
* Given the customization setting has changed, various menus, backgrounds, and/or text should change colors to reflect the new scheme
* Given the customization setting is reverted with the usage of the “Reset” button, the app should display the default dark blue once more.

1. As a user, I want to choose whether to remain logged in or not.
   1. Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 1 | Add a checkbox to the login/instance selection menu for the user to choose to remain logged in or not | 0.5 hours | Alexa |
| 2 | Connect the state of the checkbox to whether or not login information is saved between sessions | 1 hour | Alexa |
| 3 | Test that the user is logged in (or not) according to their selected preference | 0.5 hours | Alexa |

Acceptance Criteria:

* Given the UI is implemented correctly, if the user selects the save login checkbox, the user will be automatically logged in to the app on subsequent sessions.
* Given the UI is implemented correctly, if the user does not select the save login checkbox, the user will have to sign into the app again the next time it is opened.
* Given the UI is implemented correctly, if the user selects to remain logged in but the app session ends before actual login, the preference state should not be stored and default to log in again on next app launch.

**Developer Stories**

1. As a developer, I want to be able to run a test case to return an automated result.
   1. Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 1 | Research feasibility of touch input tests for Android Studio applications | 4 hours | Daniel |
| 2 | Write a simple test case in a separate project to test touch input as a proof of concept | 3 hours | Daniel |
| 3 | Take proof of concept and fully implement a version in Probos to measure a specific feature | 3 hours | Daniel |
| 4 | Make efforts to implement automated tests for other user stories | 3.5 hours | Daniel |
| 5 | Confirm whether tests properly work | .5 hour | Daniel |

Acceptance Criteria:

* Given a test is run, it should return with a message indicating whether or not the test was passed
* Given a test returned with a failure, it should return with a message indicating generally what went wrong, or the stack trace should be available
* Given a test passed, a confirmation message should be available announcing the test passed

***In Preparation for Final Presentation:***

Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 1 | ***Operation: Polish Probos***  Clean up existing code base, removing extraneous comments and adding minor improvements to existing functionality, appearance, and overall user experience | 15 hours | Everyone |
| 2 | Work towards preparing presentation materials and rehearsing our demo with dry runs | 2 hours | Everyone |

**Sprint 2 - Incomplete User Stories, Tasks, & Acceptance Criteria**

***(1 entire story, misc. worth another)***

As a user, I want to change my display name.

Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 1 | Create a duplicate, editable version of the profile UI for the user to interact with. | 3 hours | Michael |

Acceptance Criteria:

* Given the UI is implemented correctly, when the user quits updating their profile without submitting, all changes they made should be discarded.
* Given the UI is implemented correctly, when the user submits changes to the instance, their profile should refresh automatically to update with the changes.

As a user, I want to edit my account “bio” information for a specific instance.

Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 2 | Update the user profile UI to include up to 4 “metadata” fields. | 4 hours | Michael |
| 3 | Update “metadata” fields in editable profile UI to retrieve text correctly and create the appropriate API call(s). | 2 hours | Michael |

Acceptance Criteria:

* Given the UI is implemented correctly, when the user quits updating their bio and metadata without submitting, all changes they made should be discarded.
* Given the UI is implemented correctly, when the user views a profile, any metadata the account contains should be displayed appropriately.

As a user, I want to edit my profile icon and banner images in a specific instance.

Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 1 | Update code connected to both user profile and user banner images on the editable profile UI to open a media picker. | 3 hours | Michael |
| 2 | Ensure the media file selected is of a valid type to use for a Mastodon profile. | 3 hours | Michael |
| 3 | Generate the appropriate API call with the now appropriately-formatted images. | 2 hours | Michael |
| 4 | Test through both app and web client that the profile icon and banner image updates were processed correctly. | 1 hour | Michael |

Acceptance Criteria:

* Given the API calls are generated and processed successfully, when the user selects images for their profile icon/banner image and submits them, the change should be reflected from other clients.
* Given the UI is implemented correctly, when the user selects the image(s) to submit, they should be stored temporarily until either submitted or discarded on menu close.
* Given the system calls are made properly to retrieve media, when the user selects an image to use, it should be checked and converted (if needed) to an appropriate format to submit to the instance.

As a user, I want to attach a content warning to “toots” as needed.

Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 4 | Test functionality with redrafting content warnings, general bugs scrolling new messages | 2.5 hours | Steven |

Acceptance Criteria:

* Given content warnings are implemented correctly for message (re)drafting, when the user is composing a status, any text entered into the content warning field should be submitted appropriately and the message marked as sensitive.

As a user, I want to attach media to a “toot.”

Tasks Required

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 4 | Test that messages appear correctly locally and remotely. | 2 hours | Michael |

Acceptance Criteria:

* Given that the API calls for submitting media are used correctly, when a user submits a “toot” with media, it should be processed correctly and viewable from other clients.

As a user, I want to view “toot” messages sent by others.

Tasks Required:

|  |  |  |  |
| --- | --- | --- | --- |
| Order: | Task: | Estimated Time: | Assigned To: |
| 5 | Build UI for the expanded view of a “toot” | 3.5 hours | Steven |

Acceptance Criteria:

* Given the expanded message UI is implemented correctly, when the user taps to open a specific “toot,” the app should display previous and subsequent messages (if any) at regular message size in chronological order above/below the expanded message.

**Remaining Backlog:**

Functional Requirements

1. **Accounts**

As a user,

* 1. I want to register an account on a specific instance.
  2. ~~I want to log in to my account(s).~~
  3. ~~I want to be able to log out of or switch the account I am using.~~
  4. ~~I want to choose whether to remain logged in or not.~~
  5. I want to be able to go through my instance’s process for resetting passwords.

1. **Messaging in the Timeline / “Toots”**

As a user,

* 1. ~~I want to view “toot” messages sent by others.~~
  2. ~~I want to scroll to view older “toots”.~~
  3. ~~I want to view a list of the “toots” I have favorited.~~
  4. ~~I want to favorite “toots”~~
  5. ~~I want to reply to specific “toots”~~
  6. ~~I want to boost “toots”.~~

1. **Creating a Message/ “Toot”**As a user,
   1. ~~I want to send “toots.~~
   2. ~~I want to adjust the privacy setting of a “toot” (public, unlisted, followers only, direct message) on a message-by-message basis.~~
   3. ~~I want to show or hide media attached to a “toot.”~~
   4. ~~I want to attach a content warning to “toots” as needed.~~
   5. ~~I want to be able to delete and redraft my previous messages.~~
   6. ~~I want to save message drafts.~~
2. **Instance Settings**

As a user,

* 1. ~~I want to change which timeline I am looking at (personal, local, or federated).~~
  2. I want to carry over the list of profiles I follow to a new account on a different instance.
  3. ~~I want to edit my profile icon and banner images in a specific instance.~~
  4. I want to carry over my picture and banner to a new account on a different instance.
  5. ~~I want to edit my account “bio” information for a specific instance.~~
  6. I want to access custom emoji from my instance. (if time allows)

1. **Public Profile**

As a user,

* 1. ~~I want to be able to follow other users.~~
  2. ~~I want to see my list of followers.~~
  3. ~~I want to change my display name.~~
  4. ~~I want to be able to pin any of my “toots” to the top of my profile.~~
  5. ~~I want to mute/block other users.~~
  6. I want to change my notification settings.

1. **Feed / “List” Customization**

As a user,

* 1. ~~I want to create new “Lists.”~~
  2. ~~I want to edit “Lists.”~~
  3. ~~I want to delete “Lists.”~~
  4. ~~I want to be able to view my own “Lists.”~~
  5. I want to enable media only mode for the local and federated timelines. (if time allows)
  6. I want to change my filter for words/phrases. (if time allows)

1. **Notification Customization**

As a user,

* 1. ~~I want to change the frequency push notifications are generated.~~
  2. ~~I want to choose what types of notifications I receive (favorites, boosts, mentions, new followers).~~

1. **Client Appearance**

As a user,

* 1. ~~I want to customize the primary and secondary color schemes of the app’s user interface. (if time allows)~~
  2. I want the app to react properly to different font sizes. (if time allows)

1. **Server/Instance Control**

As a server admin,

* 1. I want to be able to ban accounts. (if time allows)
  2. I want to modify the instance’s default primary and secondary color schemes. (if time allows)
  3. I want to be able to read user-submitted reports. (if time allows)

1. **Security**

As a developer,

* 1. I want to implement a safe method of passing over login credentials to the instance without risking the personal information of users.
  2. I want to avoid storing raw personal login information.

Non-Functional Requirements

1. **Architecture and Design**
   1. The front end will be developed in Android Studio.
   2. Java API for Mastodon will be used for server-client communication.
   3. Multiple versions of Android will be supported.
   4. Application size will be less than 35 MB.
   5. Cache size of media, user lists, etc. will be less than 100 MB (if time allows)
   6. The client should load the instance with low latency as a priority of usability.
2. **Usability**
   1. The interface will be intuitive, sleek, and easy to navigate.
   2. Swipes and other gesture-related controls for touch screens will be implemented.
   3. The interface will adapt to multiple screen sizes. (if time allows)
   4. Portrait and landscape screen orientations will be supported. (if time allows)
3. **Security**
   1. We will build user trust and confidence by storing information in a secure manner and only if needed.
   2. Local storage backup of account information will be opt-in. (if time allows)
   3. We will limit/completely avoid handling of sensitive information by any third party or external service. (save Mastodon)