Team 14 - Sprint 1 Retrospective

**Hot TopiX**



**Team Members:** Connor Brown, Hunter David, Steve Eisner, Baxter Govan, Ryan Klinedinst, Victor Pan

**What did we do well?**

**User story #1:** As a user, I would like to register a new account.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Create account/profile table in SQL database | 2.5 Hrs | Hunter |
| 2 | Create query for creating a new account with all the associated info | 2.5 Hrs | Hunter |
| 3 | Create query for retrieving an account that already exists | 3 Hrs (Each) | Connor/Baxter |
| 4 | Create a “landing page” for non-logged in users | 5 Hrs (Each) | Victor, Connor |
| 5 | Create a “Create new account” page for users wanting to make an account | 4 Hrs (Each) | Victor, Connor |
| 6 | Implement email address verification | 2.5 Hr | Steve |

**Completed:** The queries were built out, the pages were implemented, the commands to call the queries when signing up were made, and the email address verification was made, but since they all weren’t connected, we didn’t have working acceptance criterias

**User story #2:** As a user, I would like to register a new account.

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | Create login page UI, with forms for email and password | 5 Hrs | Ryan |

**Completed:** We created a working login page that would send requests to the client handler to run the login queries. It included forms for username and password.

**User story #3:** As a user, I want to be able to edit my profile

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Estimated time | Owner |
| 1 | Create query for updating an existing account’s profile, including bio | 3.5 Hrs | Hunter |
| 2 | Setup the UI to connect to the backend and change account info in the database | 5 Hrs | Victor |
| 3 | Create forms, ~~e~~tc. for editing account info | 6 Hrs | Ryan |
| 4 | Testing | 1 Hr | Victor |

**Completed:** We implemented a UI page for editing the user’s profile. We also implemented a message to send to the server a message with the updated information that could then be sent to the database. Because the only information required in an account is the username, email, and password, the user is able to leave the rest of their account information blank.

**User story #4:** As a user, I want to be able to customize a brief bio for my profile

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Estimated time | Owner |
| 1 | Write call to edit the profile field in the database | 5 Hrs (each) | Hunter, Baxter |
| 2 | Create a way to edit the bio from the profile page | 5 Hrs (each) | Steve, Ryan |
| 4 | Unit tests | 1 Hr | Steve |

**Completed:** We were able to implement this page in the UI. We also implemented a message to send to the server and a confirmation message to send back to the client.

**User Story #6** As a user, I would like to be able to publish a microblog to the public

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Estimated time | Owner |
| 1 | Create an interface for users to type microblogs, and publish. | 7.5 Hrs (each) | Ryan, Steve |
| 4 | Testing | 1 Hr | Steve, Victor |

**Completed:** We implemented a UI to display and publish all microblogs, including the functionality to like and unlike microblogs.

**User story #7:** As a user, I would like to be able to delete my account

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | Implement a “User not Found” page for links to deleted users | 5 Hrs | Connor |

**Completed:** We implemented a User not Found page that showed the client the user isn’t available anymore.

**User story #10:** As a user, I would like to be able to search for a user to follow

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | Build and implement Search User page | 4 Hrs (each) | Baxter, Victor |

**Completed:** We implemented a Search User page that sends a request to the client handler to pull the list of users matching the name.

Our team maintained good communication throughout this sprint, particularly when it came to delegating tasks. Even when certain tasks took longer or certain team members had trouble implementing their tasks, we still made sure to keep everyone in the loop as to what was going wrong and assigned new tasks to work around problem areas. Obviously, nothing is perfect and we plan to improve in these areas. However, we do feel that communication was a strength for this sprint.

**What didn’t go well?**

**User story #1:** As a user, I would like to register a new account.

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | Implement error message for invalid email / existing email | 1 Hr | Steve |
| 8 | Implement affirmative message for successful account registration | 1 Hr | Steve |
| 9 | Testing | 1 Hr | Connor |

**Not Completed:** Most of the tasks were completed for this user story. The problem was, the database queries were not connected to the front end, so therefore we weren’t able to complete the above tasks. These required that we could access the database.

**User story #2:** As a user, I would like to be able to log into my existing account.

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | Implement error message given invalid information | 1 Hr | Steve |
| 3 | Create login page UI, with forms for email and password | 5 Hrs | Ryan |
| 4 | Forward successfully logged in users to their Timeline | 1 Hrs | Steve |
| 5 | Testing | 1 Hrs | Connor |

**Not Completed:** Because the database was not connected to the server we were unable to verify a user’s login information, therefore we could not have an error message that would display given incorrect information. This also hindered our ability to have an error message since we didn’t know if there was an error without the database. Our Timeline was also not integrated and it was in a separate maven build, so we redirected to the homepage instead.

**User story #5:** As a user, I want to be able to select a display image for my profile.

**Not Completed:** We crossed out the tasks for this user story for our updated sprint overview, but we didn’t know we had to delete the acceptance criteria. We will keep this in mind for future sprints. Because the database was not connected to the server, we were unable to implement the connection between the server and the database to save a user’s profile image. Because we learned early on that the database would not be in a state to accomplish this task, we didn’t implement any feature in the UI to save an image either. Therefore, we missed all acceptance criteria for this task.

**User story #7:** As a user, I would like to be able to delete my account

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Estimated time | Owner |
| 3 | Add a way to delete user’s account from settings page | 4 Hrs | Connor |
| 4 | Testing | 1 Hrs | Connor |

**Not Completed:** Since our database was not hooked up to our server, we could not store any account data. Because of that, we were not able to implement the ability to delete an account.

**User story #8:** As a user, I would like my sensitive information to stay private

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | Implement encryption for passwords | 2 Hrs | Steve |
| 3 | Implement private fields for User Classes | 2 Hrs | Steve |
| 4 | Testing | 1 Hrs | Steve |

**Not Completed:** Since the database was not connected to the server we could not encrypt and save the passwords. For the same reason we could not implement private fields for account information.

**User story #9:** As a user, I would like to be able to follow a [user, topic] combination

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Estimated time | Owner |
| 1 | Build User following and in Database | 3 Hrs | Baxter |
| 3 | Implement data parse in backend | 3 Hrs | Baxter |

**Not Completed:** Since the database was not connected to the server we could not build a user follow list that is saved in the database. As a result we also could not parse data in the backend for user information and topic information.

**User story #10:** As a user, I would like to be able to search for a user to follow

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Build User searching method | 4 Hrs (each) | Baxter, Victor |
| 3 | Implement Searched User list pop up | 4 Hrs (each) | Baxter, Victor |

**Not Completed:** We were not able to search for users, again because our database was not fully functional.

Though it would be easy to simply say that nothing went well, we believe that it is a more valuable use of our time to analyze specific areas upon which we wish to improve. Still, if we had to pinpoint our greatest weakness on the previous sprint we would say that our team, although maintaining good communication throughout the sprint regarding tasks in general, did not do a good job of communicating problems and dependencies between the tasks. Combining different aspects of our application’s functionality presented substantial problems for our team, as the communication of how they all fit together was not thoroughly enough discussed.

**What can we improve on?**

Our team has many things on which we can improve. We think that meeting more than once a week will help with overall communication, since we’ll all be in person and can help each other out with dependencies and tasks. Our plan moving forward is to meet at least twice a week minimum, which we believe will help with overall communication regarding execution of completing tasks and user stories.

We were very ambitious for our goals of the first sprint. We listed all the user stories that we could think of. This led to the goal of the first sprint to basically be to finish the whole application. We should have realized the time commitment this would be and divided the work up evenly for each sprint.

Also, the learning curve for setting up a spring boot application took a lot of time from the beginning of the sprint, so we had a lot less time to churn out all the ambitious user stories we decided upon for one sprint. Spring boot in general seems to target a much more large-scale application than what we intend to build, so a component of what we will work to change involves transitioning to a framework more suited to the scale of our project like Node.js and MongoDB. Though this seems like a large task, stubbornly continuing forward with our flawed execution plan may take more time than this pivot we have planned.