Team 14 - Sprint 2 Planning Document

**Hot TopiX**



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

**Sprint Overview:** For our second sprint, we have explicit goals for the frontend, backend, and database. For our front end, we plan to further build out a number of our existing UI pages to handle our user registration, user login, and profile customization functionality. We are also planning on creating additional UI pages to further our progress. Note that we are switching our implementation from Java and MySQL to Node.Js and MongoDB, since these will work much better with our program. Also, we plan to focus heavily on integrating our front end and back end, since we were unable to accomplish this during sprint one.

**Scrum Master:** Hunter David

**Meeting Plan:** Wednesday, 7PM, WALC 3060, Tuesday & Thursday 1:30-2:30 LWSN B155

**Risks and Challenges:** The main challenge we expect to face for this sprint is our pivot away from Java and MySQL toward Node.Js and MongoDB. Though these tools are a better fit for our scope and we already have tests with these platforms working, we still view one of our primary hurdles to be getting the whole team up to speed with these changes without losing time. To accomplish this, we’ve multiplied our meeting times by three for the second sprint, increasing the time that people working on different tasks have to collaborate face-to-face.

**Current Sprint Detail:**

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

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Estimated time | Owner |
| 1 | Store new User information and retrieve from MongoDB | 4 Hrs | Connor |
| 2 | Encrypt password prior to storage in the Database | 2 Hrs | Connor |
| 3 | Connect the Node.Js Server to the Bootstrap Create User Page | 4 Hrs | Steve |
| 4 | Implement a check for invalid email or existing email | 4 Hrs | Connor |
| 5 | Implement affirmative message for successful account registration | 1 Hr | Steve |
| 6 | Testing | 1 Hr | Connor |

Acceptance criteria:

* Given a user attempting to register an account with a valid email address, when they register, then their information will populate our database.
* The registered user will get redirected to the login page.
* Given a user attempting to register an account with an invalid email address (wrong format) they will receive an error message.
* Given a user registering with an already registered email, the user’s information will not populate the database and the registration page will refresh
* Given a logged in user, the sign up page will be inaccessible.

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

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Estimated time | Owner |
| 1 | Verify that the input email and password match values stored in the database | 7 Hrs | Connor |
| 2 | Implement error message given invalid information | 3 Hrs | Connor |
| 3 | Forward successfully logged in users to their Timeline | 2 Hr | Connor |
| 4 | Add database support for retrieving a user given email/password | 2 Hrs | Connor |
| 5 | Testing | 1 Hr | Connor |

Acceptance criteria:

* Given a user with a valid email address and password, when a user logs into Hot TopiX, then they are redirected to their timeline
* Given a user with an email address in incorrect format, when a user clicks login, an error message pops up stating it’s an invalid email address format
* Given a user with a valid email address and an invalid password, the login page refreshes and does not forward them to the timeline.
* Given a logged in user, the login page will be inaccessible.
* Given a user not yet logged in, the timeline will be inaccessible.

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

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Estimated time | Owner |
| 1 | Connect the database to enable sending and receiving calls to change aspects of user’s profiles | 8 Hrs | Victor |
| 2 | Setup the UI to connect to the backend and change account info in the database | 5 Hrs | Victor |
| 3 | Testing | 1 Hr | Victor |

Acceptance criteria:

* Given a logged in user, when they click the edit profile button in the navigation bar, they are presented with the edit profile page
* Given a logged in user ready to submit their profile edits, when they click ‘save’, their changes are saved to the database and their profile is updated
* Given a logged in user, when they want to delete information from their profile, they can make the corresponding info 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 | Display a user’s bio at the top of their profile page | 3 Hr | Steve |
| 3 | Unit tests | 1 Hr | Steve |

Acceptance criteria:

* Given a user with an existing profile, when they click edit profile, then they have the ability to add or edit a short bio
* Given a user with an existing profile, if they don’t want to save their changes, then they are redirected to their timeline
* Given a user with an existing profile, when they are ready to save their changes, then their changes are pushed to the database and their bio is updated

**User story #8:** As a user, I would like to be able to follow a user

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Estimated time | Owner |
| 1 | Implement table of users and their followers | 10 Hrs | Baxter |
| 2 | Connect follow button to database call | 4 Hrs (each) | Baxter, Ryan |
| 3 | Testing | 1 Hr | Baxter, Ryan |

Acceptance Criteria:

* Given that following a user works, I should be able to follow another user
* Given that following another user works, when a user clicks to follow, the followed user should show up in the original user’s database
* Given that following another user works, when a user has already followed another user, they should not be able to follow them again

**User story #9:** As a user, I would like to be able to unfollow a user

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Estimated time | Owner |
| 1 | Build User unfollowing method in server | 3 Hrs | Baxter |
| 2 | Implement unfollow button | 3 Hrs | Ryan |
| 3 | Implement unfollow data wipe in database | 4 Hrs | Baxter |
| 4 | Testing | 1 Hr | Baxter |

Acceptance Criteria:

* Given that unfollowing another user works, when a user clicks to unfollow, the unfollowed user should not show up in the original user’s database
* Given that unfollowing another user works, when a user has already unfollowed another user, they should be able to re-follow them.
* Given that unfollowing a user works, I shouldn’t be able to unfollow a user that I don’t already follow

**User Story #11:** As a user, I would like to be able to like a microblog that appears in my timeline

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Estimated time | Owner |
| 1 | Implement Like Button | 5 Hrs | Steve, Ryan |
| 2 | Implement display to show that a microblog has been liked | 6 Hrs | Ryan |
| 3 | Add ability to unlike a post | 4 Hrs | Ryan |
| 4 | Testing | 1 Hr | Ryan |

Acceptance Criteria:

* Given a microblog, a user should be able to “like” the post by clicking a button.
* Given a “liked” microblog, the user should see visual indication that the microblog has been liked.
* Given a “liked” microblog, the user should be able to unlike the post at any time.

**Total Hours of All Team Members**:

|  |  |
| --- | --- |
| **Name** | **Hours** |
| Steve | 33 |
| Hunter | 30 |
| Connor | 28 |
| Victor | 30 |
| Baxter | 31 |
| Ryan | 36 |

**Remaining User Stories from Backlog:**

* As a user, I would like to be able to quote another microblog
* As a user, I would like to be able to view Hot TopiX on both mobile and the web
* As a user with disabilities, I would like to be able to easily view the application
* As a user, I want to be able to customize the color palette of my display.
* As a user, I would like for the timeline to only contain [user, topic] combinations I choose.
* As a user, I would like to be able to see microblogs with new topics
* As a user, I would like to be able to accept or decline seeing more microblogs with new topic tags
* As a user, I would like to be able to view another user’s profile, with all posts
* (If time allows) As a user, I want to be able to report microblog posts, that I feel are offensive
* (If time allows) As a user, I would like to be able to report another user for inappropriate or offensive content
* (If time allows) As a user, I would like to be automatically logged out if no user activity is recorded for 5 minutes or more
* (If time allows) As a user, I would like to be able to create a private account, where others cannot view my microblogs unless they are following me
* (If time allows) As a user, I would like to be able to request to follow a private account to view their posts
* (If time allows) As a user with a private account, I would like to be able to accept or decline follow requests
* (If time allows) As a user, I would like to see a recommend list of accounts to follow based on what’s trending
* (If time allows) As a user, when I go to follow another user, I would like to be able to see how many times s/he used each topic tag

Non-Functional:

* Usability: The web application should be easy to navigate and edit settings. We want a clean UI with fast response time to avoid annoying lag for users. To insure this, if any action takes longer than 5000 ms, we will notify the user that the requested action is taking longer than usual. After 10000 ms, we will present the user with a timeout error.
* We also want to ensure that our application is accessible on all screen sizes and resolutions, as well as on all browsers. The application should be accessible to people with disabilities according to W3C’s accessibility guidelines version 2.
* Testing: (If time allows) We want to set up integration tests to make sure our backend can handle many users at once and existing users still see the filtered content they want