Scholasticate

Team 8 - Sprint 1 Planning Document

Daniel Karagory, Andrew Lanham, Jackson Rosenberg, Colston Streit, Devin Vering, Ethan Zhu

Sprint Overview:

During this sprint, our group is prepared to hit the ground running and set up the architecture for the project. Our primary goals include configuring the database design and most of the server-side logic, as we want to reach a state of testability before we shift our focuses to the client-side UI / UX. We want to establish a solid foundation for the structure of our code to allow for the user stories to be easily fulfilled after minor additions in future sprints. Also, it is imperative for the group to determine the most efficient and effective ways to complete tasks and find a solid method for ensuring all necessary tasks are well-crafted and completed in a timely manner. Strong, methodical architecture design will be the main theme of this sprint and a key to ensuring all future sprints are equally successful.

Scrum Master: Jackson Rosenberg

Meeting Plan: Mondays at 7:30pm, Wednesdays at 7:30PM

Risks and Challenges:

One major software challenge will be coordinating the development environments of all of the contributors on their personal machines; with users on Windows and Linux operating systems, we must make sure no conflicts in the software arise due to these differences. Besides this, most of the challenges will fall under a non-software category - rather, they will focus on the group as a whole. Being the first sprint, there may be some friction in starting the project completely from scratch and appropriately delegating the work. Since we have no previous experience developing as a group to reflect on, there may be a bit of a challenge figuring out the habits and abilities of each individual team member and how they may affect the group as a whole.

Current Sprint Detail:

User Story #1

As a user, I would like to register for, login to, and access a Scholasticate account.

#	Description	Estimated Time	Owner
1	Create a database to contain all user login information, including a hashing algorithm for passwords	6 hours	Daniel, Jackson
2	Create UI Panels for registering new accounts and logging into existing accounts	6 hours	Devin
3	Create algorithm to receive login information from database and check entered password	4 hours	Jackson
4	Create algorithm / functionality for forgotten password retrieval	4 hours	Jackson
5	Create algorithm for verifying password strength	2 hours	Colston
6	Create unit tests for password strength algorithm	2 hours	Colston
7	Create unit tests for the algorithm that checks passwords against those stored in the database	2 hours	Jackson

Acceptance Criteria:

- Given that the database configuration and retrieval algorithm is implemented correctly, when the user enters their log-in information correctly, the user should be brought to their home page.
- Given that the database configuration and retrieval algorithm is implemented correctly, when the user enters their log-in information incorrectly, the user should be prompted to re-enter their credentials with a warning attached.
- Given that the database configuration and retrieval algorithm is implemented correctly, when a user enters in an email that does not exist in the database, the user should be asked to register a new account.
- Given that the database configuration is implemented correctly, when a user registers for an account, the data should be properly stored and the user should be redirected to a profile settings page.
- Given the forgotten password algorithm is implemented correctly, when a user requests a password change, the user should receive a redirection link to update it in an email associated with their account.
- Given the password strength algorithm is implemented correctly, when a user's desired password does not meet the application standards, the application should not allow the password to be entered as a request (and vice versa).

User Story #2

As a user, I would like to edit my personal and academic information on my profile and see those changes reflected publicly.

#	Description	Estimated Time	Owner
1	Create UI panels for displaying people's profiles	4 hours	Devin
2	Create UI panels for editing your own profile	4 hours	Devin
3	Create unit tests for ensuring the UI panels function properly	2 hours	Devin
4	Create algorithm / functionality for submitting profile updates to database	2 hours	Daniel
5	Create algorithm / functionality for requesting profile information from the database and displaying it appropriately	4 hours	Ethan

Acceptance Criteria:

- Given that the UI is implemented well, I should be able to quickly and easily understand the structure of users' profiles.
- Given that the algorithm to request profile information from the database is implemented correctly, I should see the correct information displayed on the UI.
- Given that the algorithm to submit profile changes to the database is implemented correctly, when I refresh the page after submitting changes, the profile page should reflect those changes.

User Story #3

As a user, I would like to select my school and courses from the database or, if they are not currently in the database, request that they be added to the database.

#	Description	Estimated Time	Owner
1	Create UI panels for selecting school and courses within the profile editing page	4 hours	Devin
2	Create algorithm / functionality for retrieving current schools and courses from the database	2 hours	Ethan
3	Create algorithm for filtering the current schools and courses from the database based on the contents of a search bar	6 hours	Ethan
4	Create unit tests for filtering algorithm	3 hours	Ethan
5	Create functionality for requesting that a school or course be added to the database	3 hours	Ethan
6	Create algorithm that either approves or denies these requests to add entries to the database (with regex)	2 hours	Jackson
7	Create unit tests for the request approval / denial algorithm.	2 hours	Ethan

Acceptance Criteria:

- Given that the school and course list retrieval algorithm is implemented correctly, when I go to select what school I attend and what courses I study, all options from the database should be visible and available to choose from.
- Given that the school and course list requesting algorithm is implemented correctly, when I press the button to request something new to be added to the database, the server should receive a notification and begin undergoing the approval algorithm.
- Given that the school and course list approval algorithm is implemented correctly, if my request for something new to be added to the database follows the guidelines set forth by the app, my request will be accepted and it will be added to the database for later selection.
- Given that the filtering algorithm is implemented correctly, when I filter database options by a certain string, only options in the database that somehow match that string will appear in the result.
- Given that the UI is implemented correctly, when I go to search for a certain school or course to select, the options available to me will shrink in number as they are correctly filtered by my search.

User Story #4

As a user, I would like to interact with other users through direct messaging systems and create connections with other users by friending them (or blocking them).

#	Description	Estimated Time	Owner
1	Create UI panels for direct messaging between two (or many) users	8 hours	Devin
2	Create socket initialization functionality when one student sends a starting message	8 hours	Daniel, Ethan
3	Create functionality for requesting a user to be added to / removed from the friend list stored on the user profile	1.5 hours	Jackson
4	Create functionality for requesting a user to be added to / removed from the blocked list stored on the user profile	1.5 hours	Jackson
5	Create UI features to distinguish types of relationships between users (friends, neutral, blocked)	2 hours	Devin
6	Create UI panel to show friend list as a section of user profiles	4 hours	Devin
7	Create unit tests for the friend / block request approval / denial algorithm.	1 hour	Jackson

Acceptance Criteria:

- Given that the UI is implemented correctly, when I go to view my messages, I should be able to see a list of users I have direct messages with.
- Given that direct messages are implemented properly, when I visit the messaging pages, I should be able to send a message to a user, receive messages from other users and view all past messages.
- Given that the friend list feature is implemented properly, whenever I enter the application I should be able to send and receive friend requests, which should show up in the friends list after accepting.
- Given that the block list is implemented properly, when I visit a troublesome user's profile, I should be able to add the user to my block list, which would hide their messages and prevent further communication with them.
- Given that the UI is implemented properly, when I visit my profile I should be able to access and see a list of friends.
- Given that the UI is implemented properly, when I use either the map or user list UI panels, I should be able to distinguish friend users from other users.

User Story #5

As a user, I would like to view the updating locations of (and distances from) other users, as well as have my own location update periodically.

#	Description	Estimated Time	Owner
1	Implement JS Geolocation API / establish location for every user currently logged in the application	5 hours	Colston
2	Create functionality to automatically update the location of every user appearing online	6 hours	Colston
3	Create algorithm to compute current distances between user and all other students	1.5 hours	Colston
4	Create algorithm to sort all current distances between the user and all other students in order of magnitude (smallest to largest)	1.5 hours	Colston
5	Create functionality to allow user to change preferences on location updates (automatic timing, manual refreshing)	2 hours	Colston
6	Create unit tests for the distance sorting algorithm	2 hours	Colston

Acceptance Criteria:

- Given the API is correctly implemented, when I view another profile, I should be able to view their location information if they are choosing to appear online.
- Given the updating location algorithm is correctly implemented, when I am idle in either of the searching portals, the location should automatically update at a frequency specified in my settings.
- Given the updating location algorithm is correctly implemented, when I choose to manually update all user locations, I should see a change in distances in the user list and user icon locations on the map.
- Given the distance computation and sorting algorithms are correctly implemented, when I view the user list UI panel, I should see a list of all online users in order of their relative distance away from me.

User Story #6

As a user, I would like to search for other users in the database with certain criteria.

#	Description	Estimated Time	Owner
1	Create UI panels for user list, including a search feature	6 hours	Colston
2	Create algorithm / functionality for retrieving current online users from the database.	2 hours	Colston
3	Create algorithms for filtering the current online users based on name, distance radius, or current course information	6 hours	Jackson
4	Create unit tests for filtering algorithm	2 hours	Jackson

Acceptance Criteria:

- Given that the filtering algorithm is implemented correctly, when I search for users with certain attributes in the list, only users matching the query will appear in the search result list.
- Given that the UI panels for the filtering are implemented correctly, when I filter the search by distance radius, I should see a dropdown menu with selections instead of a standard text input box.
- Given that the filtering algorithm is implemented correctly, when I search for users with certain attributes on the map, only users matching the query will be shown on the map while other users will not.
- Given that the algorithm to retrieve current online users from the database is implemented correctly, when I go to view who is online, I should only see people whose status is online rather than everyone in the database.

User Story #7

As a user, I would like to create a study group and I would like to create invitations for the group.

#	Description	Estimated Time	Owner
1	Create UI panels for creating a new study group	6 hours	Andrew
2	Create UI panels for creating study group invitations	6 hours	Andrew
3	Create UI panels for editing a study group profile	6 hours	Andrew
4	Create unit tests for ensuring the UI panels function properly	2 hours	Andrew
5	Create algorithm / functionality to add a new study group to the database	2 hours	Daniel
6	Create algorithm / functionality for submitting study group profile updates to database	2 hours	Daniel
7	Create algorithm / functionality to send invitations to the specified users	4 hours	Daniel
8	Create unit tests to verify proper study group requests to database	2 hours	Daniel

Acceptance Criteria:

- Given that the algorithm to add a new study group profile to the database is implemented correctly, when I create a new study group, its information should be persistent and last across page refreshes.
- Given that the algorithm to add a new study group profile to the database is implemented correctly, when I create a new study group, I should be distinguished with a UI feature to allow users to see I am the owner.
- Given that the algorithm to submit study group profile changes to the database is implemented correctly, when I refresh the page after submitting changes, the study group's profile page should reflect those changes.
- Given that the algorithm to send study group invitations to other users is implemented correctly, when I send out an invitation, other users should be notified and be given the opportunity to accept or reject the invitation.
- Given that the algorithm to send study group invitations to other users is implemented correctly, when I have an invitation to another user pending, I should be notified of the other user's decision to either accept or reject the offer.

User Story #8

As a user, I would like to view a study group profile and join an existing study group.

#	Description	Estimated Time	Owner
1	Create UI panels for accepting or rejecting study group invitations	4 hours	Andrew
2	Create UI panels for displaying a study group profile	6 hours	Andrew
3	Create algorithm / functionality to accept or reject study group invitations	1.5 hours	Ethan
4	Create unit tests to verify proper study group information updates when invitations are accepted	0.5 hours	Ethan
5	Create algorithm / functionality for requesting study group profile information from the database and displaying it appropriately	4 hours	Daniel

Acceptance Criteria:

- Given that the algorithm to request study group profile information from the database is implemented correctly, when I select to view a study group, I should see the correct information displayed on the UI.
- Given that the algorithm to accept study group invitations is implemented correctly, when I accept an invitation to join a study group, I will be formally added to the list of members for that study group.
- Given that the algorithm to accept study group invitations is implemented correctly, when I accept an invitation to join a study group, other users should be able to see my name automatically be updated into the members section of the study group profile.
- Given that the algorithm to reject study group invitations is implemented correctly, when I reject an invitation to join a study group, the invitation will be deleted and I will not be added to the list of members for that study group.

Remaining Backlog:

Functional

1. **Users can create, edit, and manage their accounts.**
   As a user,
   a. ~~I would like to register for a Scholasticate account.~~
   b. ~~I would like to login to my Scholasticate account.~~
   c. ~~I would like to access my Scholasticate account from web browsers on both mobile devices and desktop computers.~~
   d. ~~I would like to reset my password if necessary.~~
   e. I would like to delete my account and all my information if desired.
   f. ~~I would like to select which course(s) I'm currently studying.~~
   g. ~~I would like to edit my user profile's personal information (name, pronouns, profile picture, etc.).~~
   h. ~~I would like to edit my user profile's academic information (expected graduation date, major, minors, etc.).~~
   i. ~~I would like to add a small biography to my profile to describe study habits and personality traits.~~
   j. I would like to set times where I am available and unavailable.
   k. I would like to have my course history saved to my profile to inform other users who may request tutoring assistance.
   l. I would like to update wardrobe information in order to help find other people more seamlessly.
   m. I would like to have the option to display my profile as offline after finding another student to work with.

2. **Users can customize their preferences regarding location services.**
   As a user,
   a. ~~I would like to manually refresh the location services in addition to having it be automatically updated at regular intervals.~~
   b. ~~I would like to customize the location service's refresh rate.~~

3. **Users can view other active users both on a map and in a list.**
   As a user,
   a. ~~I would like to see a list of users who are currently online.~~
   b. ~~I would like to select and view individual user profiles within the list.~~
   c. I would like to see a map displaying everyone's locations.
   d. I would like to be able to zoom in and out on a map.
   e. I would like to select and view individual user profiles as a pop-up window on the map portal.
   f. ~~I would like to see a list of online users sorted by distance.~~
   g. ~~I would like to see a list of all friends who are currently online.~~
   h. ~~I would like to see a distinction between my friends and other users.~~

4. **Users can manage their relationships and communications with other users.**
   As a user,
   a. ~~I would like to send a message to another user.~~
   b. I would like to screen messages from non-friend users before beginning a direct message chain.
   c. I would like to delete a message chain so that I can't see it anymore.
   d. ~~I would like to friend another user to potentially reconnect during another study session.~~
   e. ~~I would like to remove users from my friends list.~~
   f. ~~I would like to block another user if they are bothering me in some way.~~
   g. I would like to report another user if they are being inappropriate.

5. **Users can search for other users with whom to study using a variety of filters.**
   As a user,
   a. I would like to search for people studying the same course using keywords.
   b. I would like to search for people studying the same course using specific course number information.
   c. ~~I would like to filter nearby students by school.~~
   d. ~~I would like to filter nearby students by course.~~
   e. ~~I would like to filter nearby students by course section.~~
   f. ~~I would like to filter nearby students by my friends list or if I belonged to a shared study group in the past.~~

6. **Users can help contribute to the database by adding their own courses and schools so future users can find them.**
   As a user,
   a. ~~I would like to request that a new school be added to the database for later searches.~~
   b. ~~I would like to request that a new course be added to the database for later searches.~~
   c. ~~I would like to request that a new course section be added to the database for later searches.~~

7. **Users can interact with existing study groups.**
   As a user,
   a. I would like to have the option of publicly displaying active meetups, or study groups, I participate in.
   b. I would like to see all study groups distinctly marked on the map and list interfaces.
   c. ~~I would like to view the study group profile and the individual profiles of all of the members present.~~
   d. ~~I would like to request to join an active study group and send direct messages to all current active members.~~

e. I would like to receive messages from individuals from shared study groups in a group chat format.
f. ~~I would like to discern the owner of an active study group.~~

8. **Study group owners can manage their own study groups.**
   As an owner of a study group,
   a. ~~I would like to name the study group and have the name displayed on the user interfaces.~~
   b. ~~I would like to edit a study group "profile" with a similar structure to the user profiles.~~
   c. ~~I would like to create invitations for the group.~~
   d. I would like to restrict the number of members in the group, and remove members if necessary.
   e. I would like to have settings to schedule recurring iterations of the same study group at a certain time and location.
   f. I would like to transfer ownership of the group to another member.
   g. I would like to delete the study group from the system if it is no longer relevant.

9. **Users can receive and customize notifications.**
   As a user,
   a. I would like to receive pop-up notifications when a message is sent to my profile.
   b. I would like to receive pop-up notifications when a new member joins a current study group I participate in.
   c. I would like to receive pop-up notifications when a scheduled study group meeting is about to begin.
   d. I would like to change notifications preferences in my user profile and receive email notifications if desired.

Non-Functional (ongoing throughout project)

1. **Performance Requirements**
   As a developer,
   a. I would like the back-end server to be able to handle at least 1,000 simultaneous requests.
   b. I would like each user's location to be updated at regular intervals.
   c. I would like the application to be fluid and stutter-free on all platforms.

2. **Client Requirements**
   As a developer,
   a. I would like the application to be available on an Android phone.
   b. I would like the application to be available on an iOS device.
   c. I would like the application to be available on a generic web platform.

3. **Server Requirements**
   As a developer,

a. I would like the server to be able to support real-time communication with all connected clients.
b. I would like the server to be able to save relevant data (users, current study groups, etc.) to a database.

4. **Usability Requirements**
   As a developer,
   a. I would like the front-end to be easily understandable to the average user.
   b. I would like the front-end to be aesthetically pleasing for all screen resolutions.

5. **Security Requirements**
   As a developer,
   a. I would like the users' account information to be encrypted.
   b. I would like the users' current location data to be encrypted.
   c. I would like the application to be resistant to SQL injection and other common exploits.

6. **Deployment Requirements**
   As a developer,
   a. I would like to use Docker to allow for easy compartmentalization of the back-end server.
   b. I would like rebuilding and testing to be done every time new code changes are committed to the repository to support continuous delivery and integration.