

# **Sprint 1 Planning**

Team #18

Project Name: DSA VS

Group Members: Lawrence Liu, Josh Cho, Tyler Gantry, Joshua Jackson

## **Sprint Overview**

During this sprint we hope to get the UI of our website set up along with linking it to the backend. Our goal is so that once the UI is set up it will facilitate easier testing for future sprints. Our backend also needs to be set up to make the testing and implementation of our docker deployment for running of problem code possible.

**Scrum Master: Joshua Jackson**

**Meeting Plan: Tues/Thurs 2:30 - 3pm**

## **Risks and Challenges:**

- Sanghyun Joshua Cho will be attending a Hackathon (ETH Denver) and will have to work virtually.
- Connecting the backend to the frontend as both work will have to get done or at least have limited implementation at similar times.
- Finding possible problems and test cases online for our database that are open-source and readily accessible.

# Current Sprint Detail

All user stories to be implemented in this sprint.

## User Story #1:

As a user, I would like to create an account and be able to set a username on [DSA.VS](#).

#	Description	Estimated Time	Owner
1	Create a database backend	6 hours	Lawrence Liu
2	Create a server for concurrent tasks	6 hours	Josh Cho
3	Implement the GUI frontend for signing up for an account	5 hours	Josh Jackson
4	Implement a query for storage of account when signed up	2 hours	Josh Cho

Acceptance criteria:

1. Given I am not logged in, when I submit a valid username and password, then my account is created and I receive a confirmation
2. Given the username exists, when I try to register with it, then I see a ‘username taken’ error and no change occurs
3. Given I submit an invalid username/password (empty or not matching requirements), when I submit the form, then I receive an error and the request is rejected

## User Story #2:

As a user, I would like to be able to login using a password on [DSA.VS](#).

#	Description	Estimated Time	Owner
1	Implement a GUI for login page	4 hours	Josh Jackson
2	Implement a query for retrieval of user info	5 hours	Lawrence Liu
3	Implement password checker for user	5 hours	Lawrence Liu

Acceptance criteria

1. Given I have a valid account, when I enter the correct username/password, then I am logged in and can access my personal page and information
2. Given I enter an incorrect password, when I try to log in, then I remain logged out and see an “invalid login” message
3. Given I am logged out, when I try to open a private page, then I am redirected to login or shown an access denied message

## User Story #3:

As a user, I would like to be able to change the password connected to my account on [DSA.VS](#).

#	Description	Estimated Time	Owner
1	Implement a GUI for password changing	3 hours	Tyler Gentry
2	Implement deletion and appending query for password change	4 hours	Tyler Gentry

1. Given I am logged in, when I enter the current correct password and a valid new password, then my password is changed
2. Given I enter the wrong current password, when I submit the change, then the update is rejected and I receive an error
3. Given my password was changed, when I log in with the old password, then the login fails.
4. Given my password was changed, when I login with the new password, then the login succeeds

## User Story #4:

As a user, I want the ability to customize my profile for my account.

#	Description	Estimated Time	Owner
1	Implement the GUI frontend for viewing and changing account details	5 hours	Josh Jackson
2	Implement a query for retrieval, deletion, and appending onto account details	4 hours	Josh Cho

Acceptance Criteria:

1. Given I am logged in, when I open my profile page, then I can view my current profile info
2. Given I edit my profile, when I click save, then the change persists and reloading the page shows the new data
3. Given I edit my profile with an invalid change, when I save, then I receive an error and nothing is changed

## User Story #5:

As a user, I want the ability to “friend” other users for later contact.

#	Description	Estimated Time	Owner
1	Implement a GUI for managing and viewing friends	6 hours	Josh Jackson
2	Implement a query for appending to and removing from a list of friends for each account	4 hours	Lawrence Liu

Acceptance Criteria:

1. Given I am logged in, when I send a friend request to another user, then the request is recorded as pending
2. Given I have a pending friend request, when I accept it, then I appear in their friends list and they appear in mine
3. Given I have sent a friend request to a user, when I send another request, then the system prevents a second friend request and shows a ‘request pending’ message

## User Story #6:

As a user, I want the ability to “block” other users from communication.

#	Description	Estimated Time	Owner
1	Implement a GUI for managing and viewing blocked users	6 hours	Josh Jackson
2	Implement a query for appending to and removing from a list of blocked users for each account	4 hours	Lawrence Liu

Acceptance Criteria:

- Given I have blocked a user, when they try to message me, then the message is not delivered and the request is denied
- Given I have blocked a user, when they try to challenge me, then the challenge will not be created
- Given I unblock a user, when they message or challenge me, then the action works normally again

## User Story #7:

As a user, I want the ability to message other users.

#	Description	Estimated Time	Owner
1	Implement a GUI for a messaging box with history	4 hours	Josh Jackson
2	Implement a concurrent client handler to ensure messaging is updated real time	5 hours	Josh Cho
3	Implement a query for chat history	5 hours	Lawrence Liu
4	Implement a query for appending to chat history	5 hours	Lawrence Liu
5	Implement a query for receiving new chats	5 hours	Lawrence

Acceptance Criteria:

- Given I am logged in, when I send a message to another user, then it is stored and appears in the conversation for both users
- Given I open an existing conversation, when messages exist, then I can load and view the message history
- Given either user has blocked the other, when a message is sent, then it is not delivered

## User Story #8:

As a user, I would like to be able to see my current Elo on [DSA.VS](#).

#	Description	Estimated Time	Owner
1	Implement a GUI for seeing ELO	4 hours	Tyler Gentry
2	Implement a query for updating ELO	2 hours	Lawrence Liu

Acceptance Criteria:

- Given I am logged in, when I view my profile, then my current Elo is visible
- Given my Elo changes, when I refresh my profile, then the Elo displayed reflects the new value
- Given I am not logged in, when I request to view my Elo, then access is denied

## User Story #9:

As an administrator, I would want the ability to monitor chats.

#	Description	Estimated Time	Owner
1	Create “admin” mode on serverside	5 hours	Josh Cho
2	Create GUI for “admin” mode	5 hours	Josh Jackson
3	Implement a query for admin to retrieve chats between 2 users	5 hours	Lawrence Liu

Acceptance Criteria:

- Given I am an admin, when I open the admin view of chats, then I can view chat logs
- Given I am not an admin, when I try to access admin chats, then I receive an authorization error
- Given chat logs exists, when I filter/search by user, then I can see their messages

## User Story #10/11:

As a user, I would like to be able to enter the queue for matchmaking on [DSA.VS](#).

As a user, I would want to be able to queue for an “AI free for all mode”.

#	Description	Estimated Time	Owner
1	Implement a global, concurrent queue for matchmaking/AI matchmaking	10 hours	Josh Cho

Acceptance Criteria:

1. Given I am logged in, when I join the queue, then I become queued and the UI shows I'm queued
2. Given I am queued, when I leave the queue, then I'm removed from the queue and the UI displays that
3. Given I click ‘join queue’ multiple times, when requests repeat, then I am only in the queue once (no duplicate users in the queue)
1. Given I am logged in, when I join the FFA queue, then I appear in that queue and not the ranked queue
2. Given I am in the FFA queue, when I leave, then I am removed and the UI shows that change
3. Given I am queued in the ranked mode, then when I try to queue for FFA, then I receive an error message and stay in the ranked queue

## User Story #12:

As a user, I would want a working report system to ban users that are abusing the platform.

#	Description	Estimated Time	Owner
1	Implement GUI for report system	5 hours	Tyler Gentry
2	Implement Admin ability to ban certain users/deleting their profile query on database	5 hours	Tyler Gentry

Acceptance Criteria:

1. Given I am logged in, when I submit a report against a user, then the report is saved and I receive a confirmation
2. Given I am an admin, when I review a report and ban a user, then the user is marked as banned and they receive a ban message when they try to log into their account
3. Given a user is banned, when they try to login/message/etc., then the system denies them access and displays a message

### User Story #13:

As a user, I would want to report any issues/bugs I find to ensure they're fixed.

#	Description	Estimated Time	Owner
1	Implement GUI for user bug reporting	4 hours	Tyler Gentry
2	Implement a query to store bug reports in the backend	4 hours	Tyler Gentry

Acceptance Criteria:

1. Given I am on the bug report page, when I submit a report, then it is saved and I see a confirmation
2. Given I submit a report missing required fields, when I submit, then I see validation errors and the report isn't saved
3. Given I'm an admin, when I open the bug reports list, then I can view submitted reports

## Testing Needed

1. As a user, I would like to create an account and be able to set a username on [DSA.VS.](#)
  - a. test for registration success and duplicate usernames
2. As a user, I would like to be able to login using a password on [DSA.VS.](#)
  - a. test a valid login and invalid password
3. As a user, I would like to be able to change the password connected to my account on [DSA.VS.](#)
  - a. test using the wrong old password and the correct new one
4. As a user, I want the ability to customize my profile for my account.
  - a. test for updated profile
5. As a user, I want the ability to “friend” other users for later contact.
  - a. test for user friend request and acceptances
6. As a user, I want the ability to “block” other users from communication.
  - a. test that blocked user can’t message/challenge the person who blocked them
7. As a user, I want the ability to message other users.
  - a. test for sending/fetching message
8. As a user, I would like to be able to challenge a specific user.
  - a. test for challenge creation and acceptance
9. As a user, I would like to be able to get matched with another user with a similar Elo score in real time.
  - a. test for match pairing logic and see if matches can be created
10. As a user, I would like to be able to see my current Elo on [DSA.VS.](#)
  - a. test if correct elo is returned for authenticated users
11. As an administrator, I would want the ability to monitor chats.
  - a. test admin access to chats
12. As a user, I would like to be able to enter the queue for matchmaking on [DSA.VS.](#)
  - a. test if you can join the queue/leave and rejoin with no errors
13. As a user, I would want to be able to queue for an “AI free for all mode”.
  - a. test if you can join the queue/leave and rejoin with no errors
14. As a user, I would want a working report system to ban users that are abusing the platform.
  - a. test if reports are properly logged
15. As a user, I would want to report any issues/bugs I find to ensure they’re fixed.
  - a. test if bug reports are properly logged

# Remaining Backlog

## Functional Requirements:

1. As a user, I would like to create an account and be able to set a username on [DSA.VS](#).
2. As a user, I would like to be able to login using a password on [DSA.VS](#).
3. As a user, I would like to be able to change the password connected to my account on [DSA.VS](#).
4. As a user, I want the ability to customize my profile for my account.
5. As a user, I want the ability to “friend” other users for later contact.
6. As a user, I want the ability to “block” other users from communication.
7. As a user, I want the ability to message other users.
8. As a user, I would like to be able to enter the queue for matchmaking on [DSA.VS](#).
9. As a user, I would like to be able to challenge a specific user.
10. As a user, I would like to be able to challenge a specific user.
11. As a user, I would like to be able to see my current Elo on [DSA.VS](#).
12. As a user, I would like to be able to get matched with another user with a similar Elo score in real time.
13. As a user, I would want to view the history of my previous matches to review and see how I’ve improved.
14. As a user I would want the matches to last no longer than 25 minutes and be able to see a timer of my remaining time left.
15. As a user, I would like to be able to sort my past problems by category.
16. As a user, I would like to be able to see the number of test cases my opponent has passed at any given time during a match on [DSA.VS](#).
17. As a user, I would want the code that I have written to execute properly when given test cases.
18. As a user, I would like for my Elo to increase or decrease depending on the outcome of a match.
19. As a user, I would want a variety of problems within the problem set that are correct and able to be solved within the given time period.
20. As a user, I would want to be able to queue for an “AI free for all mode”.
21. As a user, I would want a working report system to ban users that are abusing the platform.
22. As a user, I want the ability to see my statistics for different modes and types of problems.
23. As a user, I want the ability to sort and visualize my different stats in different formats.
24. As a user, I would want to see how my current Elo stacks up to others with a ranking system through an active leaderboard.
25. As a user, I would want to report any issues/bugs I find to ensure they’re fixed.
26. As a user, I would want to be able to spectate a match (if time allows).

27. As a user, I would want a whiteboard-like scratch area to help visualize my solution (if time allows).
28. As an employer, I would like to see the current level of programmers to be able to gauge the requirements for jobs.
29. As an instructor, I would like to evaluate a student's coding performance without the use of AI.
30. As an employer, I would like to see the relative focus students have on certain problems.
31. As an employer, I would like to see what programming languages students excel at.
32. As a contributor, I would like to be able to submit problems to the [DSA.VS](#) problem set.
33. As a contributor, I would like to submit individual test cases for a given problem.
34. As a user, I would want the ability to select a preferred language before a match.
35. As a user, after a match I would want the ability to view problem solutions.
36. As a user, I would want the ability to report suspected cheating.
37. As an instructor, I would want the ability to create private matches with a password.
38. ~~As an administrator, I would want the ability to monitor chats.~~
39. As an administrator, I would want the ability to review submitted problem sets and solutions before they are added to the pool.
40. As a user, I would want the ability to receive notifications when a new chat pops up, a friend comes online, or I am challenged by a user.
41. As a user, I would want to earn achievements or profile badges when certain milestones are passed.
42. As a user, if I disconnect during a match I would want to be able to reconnect and continue working if the match is still ongoing.
43. As a user, I would want the ability to save my current work if I am disconnected or the page refreshes.
44. As a user, I would want the ability to see the difficulty rating of current problems.
45. As a contributor, I would want the ability to edit or update previous problems submitted.

## **Nonfunctional Requirements:**

### **Security**

- All sensitive user data (emails, match results, etc.) should be encrypted to protect privacy and secure data
- Multi-factor authentication should be implemented for 95% to prevent unauthorized access, ensuring user accounts are secure
- User-submitted code should be properly sanitized in order to prevent vulnerabilities like SQL injection
- Sessions should time out after 15 minutes of inactivity to prevent unauthorized access

### **Architecture**

- Code execution should be done in a sandboxed environment to ensure any malicious code has no effect

### **Performance**

- The site should be able to handle up to 20 simultaneous users and matches at a time without lags or crashes
- The platform should have < 1 second of latency to prevent disruptions to the competitive experience
- The system should be able to handle up to 20 concurrent matches or more, allowing performance to remain stable as user demand increases

### **Usability**

- User satisfaction with the interface should be 4.5/5 or higher to allow for easy navigation and coding without issues
- Accessibility features like a screen reader will allow the platform to be used by individuals with disabilities