

CS 30700: Sprint Planning Document

Team: Group 26

Team Members: Gokul Yenugadhati, Vishaal Bommena, Harsha Lingareddy, Dhiman Swadia, Sanat Mouli and Vijay Viswan

Sprint Overview:

For the the first sprint our plan to focus most of our energy on getting the web-based subsection of our project up. Our goal is that by the end of this sprint we will have the user be able to create an account, which can log in using 2 factor authentication. In addition we want the user to have basic site functionality such as forum to input past class and view the list of different courses. The backend/ml team will be working on the initial algo which will enable a user to see if a course is recommended or not.

Scrum Master: Dhiman Swadia

Meeting Schedule: Monday, Wednesday, Friday 7-9 PM

Risks/Challenges:

A risk is that some team members are not yet familiar with the languages, frameworks or technologies we are using such as Python, React, Flask, Heroku, and MySQL. In this sprint a majority of the time will be to get most of the team members familiar with these technologies. A challenge we will have as team this is that this is the first time most of the group has worked with a group of this size and it will take some to get used to work as group and adjust/dealing with git related issues.

Current Sprint Detail

User Story #1

As a user, I would like to create an account.

Description

#	Description	Time (hrs)	Owner
1	Set up Web Client	6	Harsha
2	Set up backend server (Flask)	6	Gokul
3	Set up Database	5	Sanat
4	Setup frontend server (Node)	5	Dhiman
5	Setup production server and Database services	10	Sanat
6	Implement backend ability for account creation	8	Dhiman
7	Testing to check functionality and boundary cases	1	Vijay

Acceptance Criteria

- Given the backend database for the user accounts is setup successfully, when there is a request for a new account, a new user profile will be added to the database if the given email does not exist in the database and the password is strong.
- Given the account was successfully created, the user will be prompted to a page where they can input their information.
- Given that the user inputted their information the frontend service should process the information and send it to the backend.
- Given that the backend has received the new user profile, it will create a new user profile in the database.

User Story #2

As a user, I would like two-factor authentication.

Description

#	Description	Time (hrs)	Owner
1	Implement Frontend UI to input the verification code	5	Dhiman
2	Implement API service send text message	5	Vijay
3	Implement backend logic to process the text message	5	Vijay
4	Implement backend service to verify the verification code	5	Vijay
5	Testing to check functionality and boundary cases	1	Vishaal

Acceptance Criteria

- Given that the user has an account, they should be prompted to enter their phone number. (First time log-in)
- Given that the user has given their phone number, a code should be sent to their phone.
- Given that user has a code, user should enter code on browser to log in to their account.
- Given that the user has logged in, they should be redirected to the homepage.

User Story #3

As a user, I would like to input the courses I have previously taken.

Description

#	Description	Time (hrs)	Owner
1	Backend Service to implement this functionality	6	Vijay
2	Setup Frontend Client to send to backend	4	Harsha
3	Store results in database	5	Sanat
4	Frontend to provide searchable course list	3	Harsha

Acceptance Criteria

- Given that the backend contains all valid courses, it should be able to send out information for all courses from the database.
- Given that the user wants to add courses, the frontend service should be able to retrieve available courses from the backend database.
- Given that the user inputted information to their profile, the backend should be able to handle these updates in the user database.
- Given that the frontend receives the list of valid courses, it should be able to conveniently display them when the user searches the list.

User Story #4

As a user, I would like to view my profile information.

Description

#	Description	Time (hrs)	Owner
1	Implement Backend service to fetch/update profile information	6	Gokul
2	Frontend Client to implement functionality	6	Harsha

Acceptance Criteria

- Given that the user has created an account, the backend service should be able to fetch the information from the database.
- Given that the backend service is able to fetch the information, the frontend should be able to retrieve the information.
- Given that the front end retrieves the information it should be able to display the information on the profile screen.
- Given that the user needs to update information, the database should be able to handle all updates.

User Story #5

As a user, I would like to view the list of different courses.

Description

#	Description	Time (hrs)	Owner
1	Frontend Client to display courses	7	Dhiman
2	Backend Service to implement this functionality	5	Vijay
3	Set up Database to store results	5	Sanat

Acceptance Criteria

- Given that the user requests to view the course list, the frontend should process this request and forward it to the backend server.
- Given that the database is complete and filled with all courses, the backend service should be able to retrieve list of courses.
- Given that the backend is able to retrieve the list of courses, the frontend service should be able to retrieve the courses from the backend service.
- Given that the frontend has all the data from the backend service, the frontend should be able to display the list of all courses.

User Story #6

As a user, I would like to see if a course is recommended or not.

Description

#	Description	Time (hrs)	Owner
1	Set up backend service to connect to machine learning data pipeline.	8	Gokul
2	Set up training database	4	Sanat
3	Normalizing data	4	Vishal
4	Machine Learning service to generate models	10	Vishal
5	Classification of data for machine learning	10	Vishal
6	Frontend to display machine learning data result	4	Harsha
7	Testing to check functionality and boundary cases	2	Vijay

Acceptance Criteria

- Given that the user has inputted a list of course with their opinion on the course, the data should be stored in a database for training.
- Given that the data has been stored in the database, the machine learning service should have data to train on.
- Given that the machine learning service has training data, the model should be able to normalize the data for proper classification
- Given that the machine learning service has normalized the data, the service should classify a course into recommended or not.
- Given that the machine learning service has classified a course into recommended or not, the classified tag should be edited in the database.

User Story #7

As a user, I would like to contact the webmaster to report bugs.

Description

#	Description	Time (hrs)	Owner
1	Frontend Client to report bugs	4	Harsha
2	Backend Service to handle the request	4	Dhiman
3	Frontend Client to forward response from webmaster to user	4	Harsha
4	Testing to check functionality and boundary cases	1	Gokul

Acceptance Criteria

- Given that there is a web prompt to enter details that need to be fixed with the webpage, these details will be sent to the webmaster.
- Given the prompt exists and is able to send text, the user will also be able to send a image regarding the problem.
- Given the user sent a bug report, this prompt should be sent to the webmaster.
- Given the webmaster receives the report, the webmaster should be able to send a response back to the user.

User Story #8

As a user, I would like to view a recommended course list without inputting my own data.

Description

#	Description	Time (hrs)	Owner
1	Set up backend service to connect to machine learning data pipeline.	4	Vijay
2	Set up training with general variables	4	Gokul
3	Normalize variables	5	Gokul
4	Machine Learning service to generate models	4	Vishal
5	Classification of general variables for machine learning	3	Vishal
6	Testing to check functionality and boundary cases	1	Sanat

Acceptance Criteria

- Given the user logs in without inputting data, the training variables should be set to general variables (vote, class size, etc.).
- Given no input in the database, the model should work on training on the general variables.
- Given training on general variables, the model should classify whether the course is recommended or not.
- Given classification, the predictions should be set in the database.

Team Hour Chart Per User Story

User Story #	1	2	3	4	5	6	7	8	Total Hours
Harsha	6		7	6		4	8		31
Gokul	6			6		8	1	9	30
Dhiman	13	5			7		4		29
Sanat	15		5		5	4		1	30
Vijay	1	15	6		5	2		4	33
Vishaal		1				24		7	33

Remaining Backlog (12/31 complete)

1. As a user, I would like to create an account.
2. As a user, I would like to use my social accounts to sign up. (If time permits)
3. As a user, I would like to create a handle for my account. (If time permits)
4. As a user, I would like two factor authentication.
5. As a user, I would like to input the classes I have previously taken
6. As a user, I would like to view my profile information.
7. As a user, I would like to upload a profile picture.
8. As a user, I would like to view the list of different courses
9. As a user, I would like to sort the list of courses
10. As a user, I would like to view the courses of different degree requirements
11. As a user, I would like to see if a course is recommended or not
12. As a user, I would like to see a recommendation score for each course
13. As a developer, I would like to calculate the recommendation score for each course
14. As a user, I would like to see improved recommendation after using the platform for some time.
15. As a user, I would like to get a predicted grade for a specific course.
16. As a user, I would like to see how others have upvoted the class.
17. As a user, I would like to sort according to the upvoted of the class.
18. As a user, I would like to differentiate the recommended courses according to the degree requirements.
19. As a user, I would like to view a recommended course list without inputting my own data.
20. As a user, I would like to get specific course information.
21. As a user, I would like to get course timings for the course for specific semesters.
22. As a user, I would like to post a comment for a specific course.
23. As a user, I would like to get recommended professors for the specific course.
24. As a user, I would like to view other student's comments for a specific course.
25. As a user, I would like to search courses. (If time permits)
26. As a user, I would like to view courses from other majors.
27. As a user, I would like to be able to contact the webmaster to report bugs.
28. As a user, I would like to share the specific course information with peers.
29. As a user I would like to communicate with other students taking the specific course.
30. As a user, I would like a conversational UI to communicate with the recommendation system. (If time permits)
31. As a user, I would like to see push notification for comments on recommended course. (If time permits)