



## Sprint 1 Planning Document

Team 38: Hareum Lee, Xavier Dextra, Sparkle Biswas, Divay Gupta, Joseph Jackson, Rudis Sprenne

### **Sprint Overview**

During this sprint, we hope to implement the front-end and back-end functionality of login/registration page for students and instructors. Users (students) should be able to register/login and view all the courses they are registered in. Users (instructors) should be able to register/login and view all the courses they are teaching, create/delete courses and add/remove students from their courses. We will also implement the navigation system so that users can easily switch between pages.

**Scrum Master:** Rudis Sprenne

**Meeting Plan:** Fridays/Sundays @ 6:00pm

### **Risks and Challenges**

For Sprint 1, the biggest challenge for some of us is going to be creating the UI interface, the frontend functionality, and the backend without any prior full-stack development experience. It will likely take us longer to implement these front-end and back-end functionalities than mentioned, since some of us will have to get used to React JS and Firebase. We will also have learn how to navigate this project as a group and divide tasks efficiently.

## Current Sprint Detail

### User Story #1

As a user, I want to be brought to the login page upon visiting the website.

#	Description	Estimated Time (Hours)	Owner
1)	Initialize the Firestore database	2	<b>Divay</b>
2)	Implement basic Firebase backend functionality	2	<b>Divay</b>
3)	Test backend functionality	2	<b>Divay</b>
4)	Create a basic “Login” page	4	<b>Divay</b>
5)	Verify that the website can be accessed on a web browser	1	<b>Divay</b>
<b>Total Hours</b>		<b>11</b>	

### Acceptance Criteria:

- Given that a user tries to visit the website for the first time, they should be taken to the Login page.
- Given that the backend and database are initialized, the developers should be able to access and query the backend.
- Given that the database is initialized and basic backend functionality has been implemented, when the backend inserts data in the database, the data should be present in the database.
- Given that the database is initialized and basic backend functionality has been implemented, when the backend deletes data in the database, the data should no longer be present in the database.

### User Story #2

As a user, I want to register for a CampusLink account.

#	Description	Estimated Time (Hours)	Owner
1)	Create a “Registration” page	3	<b>Rudis</b>

2)	Implement backend functionality to handle registration	3	<b>Rudis</b>
3)	Create frontend functionality so that user-entered information is sent to the backend.	2	<b>Rudis</b>
4)	Integrate with the backend	1	<b>Rudis</b>
5)	Test frontend-backend integration	1	<b>Rudis</b>
<b>Total Hours</b>		<b>10</b>	

#### Acceptance Criteria:

- Given that a user doesn't have an account, when a user creates an account with their email and password, a success message is displayed.
- Given that a user attempts to create an account with duplicate information (such as, a duplicate email), an error message is displayed.
- Given that an account has been successfully created, user's account should be added to the database.

#### User Story #3

As a user, I want to log in to my account on CampusLink.

#	Description	Estimated Time (Hours)	Owner
1)	Implement complete "Login" page frontend functionality	3	<b>Divay</b>
2)	Implement backend functionality to handle authentication with JWT for the various user types	4	<b>Divay</b>
3)	Implement frontend functionality	3	<b>Divay</b>
4)	Integrate with the backend	1	<b>Divay</b>
5)	Test frontend-backend integration	1	<b>Divay</b>
<b>Total Hours</b>		<b>12</b>	

#### Acceptance Criteria:

- Given that a user's account information exists in the database, the user should be able to log in and a success message should be displayed.

- Given that a user's account information does not exist in the database, the user should not be able to log in and an error message should be displayed.
- Given that the user has successfully logged in, if the user is a student or an instructor, the user should be sent to the appropriate Courses page (student/instructor view).
- Given that the user is an instructor, the user should not be able to login until the instructor account request has been approved by an admin, and an error message should be displayed.

#### User Story #4

As a user, I want to log out from my CampusLink account.

#	Description	Estimated Time (Hours)	Owner
1)	Create front-end Logout functionality	2	Hareum
2)	Implement backend functionality to handle log-out	3	Hareum
3)	Implement navigation to Login page	2	Hareum
3)	Integrate with the backend	3	Hareum
4)	Test frontend-backend integration	1	Hareum
<b>Total Hours</b>		<b>10</b>	

#### Acceptance Criteria:

- Given a user is logged in, when the user clicks the log out button, they should be logged out and a message should be displayed.
- Given a user has successfully logged out, the user should be sent to the login page.
- Given a user is successfully logged out, the user should not be able to access any page other than the login or registration pages without first logging back in.

#### User Story #5

As a user, I want to request for an instructor account on the registration page.

#	Description	Estimated Time (Hours)	Owner
1)	Create frontend functionality to request for an	4	Joseph

	instructor account on the registration page.		
2)	Implement backend functionality.	2	<b>Joseph</b>
3)	Test that data is being sent correctly	1	<b>Joseph</b>
3)	Integrate frontend and backend.	2	<b>Joseph</b>
4)	Test frontend-backend integration.	1	<b>Joseph</b>
<b>Total Hours</b>		<b>10</b>	

#### Acceptance Criteria:

- Given that a user is creating a professor account, the user should be able to request for an instructor account.
- Given that a user is creating a student account, no request for an instructor account should be sent.
- Given that a user has successfully registered for an instructor account, the request should be sent to the backend.

#### User Story #6

As a user (student), I want to be able to login and have my courses be in student view.

#	Description	Estimated Time (Hours)	Owner
1)	Create a student view "Courses" page	4	<b>Sparkle</b>
2)	Implement navigation to the Courses page upon student login	2	<b>Sparkle</b>
3)	Implement front-end functionality for displaying courses in student view	3	<b>Sparkle</b>
4)	Integrate with backend	2	<b>Sparkle</b>
5)	Test frontend-backend integration	1	<b>Sparkle</b>
<b>Total Hours</b>		<b>12</b>	

#### Acceptance Criteria:

- Given that a student user successfully logs in, the user should be redirected to the Courses page.

- Given that a student user is on the Courses page, the correct list of registered courses should be displayed.
- Given that a student user is on the Courses page, they should not be able to access any instructor-specific Courses page functions (such as creating or removing courses).

### User Story #7

As a user, I want to be able to navigate the website using a navigation bar.

#	Description	Estimated Time (Hours)	Owner
1)	Create a Navigation Bar	3	<b>Sparkle</b>
2)	Implement front-end functionality to navigate between pages	3	<b>Sparkle</b>
3)	Implement session authentication functionality to keep user logged in upon navigation	3	<b>Sparkle</b>
4)	Test that page navigation works as intended	1	<b>Sparkle</b>
<b>Total Hours</b>		<b>10</b>	

### Acceptance Criteria:

- Given a user clicks on a navigation button, then the user should be brought to the requested page.
- Given a user is logged in and the logout functionality is implemented, when a user clicks the logout button, the user should be logged out and a log out message should be displayed.
- Given the user is logged out, the navigation bar should no longer be visible to the user.

### User Story #8

As a user, I want to be able to view my account settings page.

#	Description	Estimated Time (Hours)	Owner
1)	Create an “Accounts Settings” page	3	<b>Hareum</b>
2)	Implement frontend functionality	3	<b>Hareum</b>

3)	Implement backend functionality	2	<b>Hareum</b>
4)	Test the backend to ensure that data is being sent correctly	1	<b>Hareum</b>
5)	Integrate the frontend and backend	1	<b>Hareum</b>
6)	Test frontend-backend integration	1	<b>Hareum</b>
<b>Total Hours</b>		<b>11</b>	

#### Acceptance Criteria:

- Given that a user is logged in and the user clicks on the account settings button, the user should be able to access the account information page.
- Given that a user is on the account settings page, the page should display the correct user information.
- Given that a user is on the account settings page and the account settings editing functionality has been implemented, the user should not be able to edit the information until they click the edit button.

#### User Story #9

As a user, I want to be able to access and search a frequently asked questions page

#	Description	Estimated Time (Hours)	Owner
1)	Create an “FAQ” page	3	<b>Divay</b>
2)	Implement frontend functionality	2	<b>Divay</b>
3)	Implement backend functionality to allow people to ask questions	3	<b>Divay</b>
4)	Test that the data is being sent correctly	1	<b>Divay</b>
5)	Integrate frontend and backend	1	<b>Divay</b>
6)	Test frontend implementation	1	<b>Divay</b>
<b>Total Hours</b>		<b>11</b>	

#### Acceptance Criteria:

- Given that a user is logged in, they should be able to visit the FAQ page.

- Given that a user visits the FAQ page, they should be able to view the Frequently Asked Questions.
- Given that a user successfully submits a question, it should be sent to an admin and a success message should be displayed.
- Given that a user unsuccessfully submits a question, an error message should be displayed.

### User Story #10

As a user (teacher), I want to be able to login and have the page be in an instructor view.

#	Description	Estimated Time (Hours)	Owner
1)	Create an instructor view “Courses” page	4	<b>Sparkle</b>
2)	Implement navigation to the Courses page upon instructor login	1	<b>Sparkle</b>
3)	Implement front-end functionality	3	<b>Sparkle</b>
4)	Integrate with backend	1	<b>Sparkle</b>
5)	Test implementation	1	<b>Sparkle</b>
<b>Total Hours</b>		<b>10</b>	

### Acceptance Criteria:

- Given that a user’s input does not match with information in database, an error message should be displayed.
- Given that a user has the instructor account type, the user should be logged in as an instructor
- Given that an account has been successfully logged in, user should navigate to main page that shows currently teaching courses.

### User Story #11

As a user (teacher), I want to be able to create or delete courses I am teaching

#	Description	Estimated Time (Hours)	Owner
---	-------------	------------------------	-------



1)	Implement front-end functionality to create a course on the “courses page”	2	<b>Joseph</b>
2)	Implement front-end functionality to delete a course on the “courses page”	2	<b>Joseph</b>
2)	Implement backend functionality to handle course creation	2	<b>Joseph</b>
3)	Implement backend functionality to handle course deletion	2	<b>Joseph</b>
4)	Integrate frontend and backend	1	<b>Joseph</b>
5)	Test frontend-backend integration	1	<b>Joseph</b>
<b>Total Hours</b>		<b>10</b>	

#### Acceptance Criteria:

- Given that the user successfully creates a course, the course should be added to the database, and the page should be updated to display the newly created course.
- Given that the user unsuccessfully creates a course, an error message should be displayed.
- Given that the user successfully deletes a course, the course should be removed from the database, and the page should be updated to display the correct courses.
- Given that the user unsuccessfully deletes a course, an error message should be displayed.

#### User Story #12

I want to have exclusive access to an admin page that displays all registered accounts and requests for instructor accounts.

#	Description	Estimated Time (Hours)	Owner
1)	Create an “Admin Page”	4	<b>Rudis</b>
2)	Implement admin page frontend functionality (such as displaying information and login)	3	<b>Rudis</b>
3)	Implement backend functionality	2	<b>Rudis</b>
4)	Integrate the frontend and backend	2	<b>Rudis</b>

4)	Test that frontend functionality works as intended	1	<b>Rudis</b>
5)	Test that the integration works as intended	1	<b>Rudis</b>
<b>Total Hours</b>		<b>13</b>	

#### Acceptance Criteria:

- Given that a user has the admin account type, when the user navigates to the “/admin” URL, the user should be able to access the admin page and admin login functionality.
- Given that an admin user has successfully logged in, the admin page should be displayed.
- Given that an admin user is unsuccessful in login in, an error should be displayed.

#### User Story #13

As a user, I want to be able to edit my information on the account settings page.

#	Description	Estimated Time (Hours)	Owner
1)	Implement front-end functionality to edit information	3	<b>Hareum</b>
2)	Implement backend functionality to update settings	3	<b>Hareum</b>
3)	Test that the data is being sent/updated correctly on the backend	1	<b>Hareum</b>
4)	Integrate frontend and backend	2	<b>Hareum</b>
5)	Test frontend-backend integration	1	<b>Hareum</b>
<b>Total Hours</b>		<b>10</b>	

#### Acceptance Criteria:

- Given that a user is on the account settings page and has clicked the edit button, the user should be able to edit their account information.
- Given a user is on the account settings page and has not clicked the edit button, the user should be unable to change their account settings.
- Given a user is editing their information and has successfully submitted the changes, the user information should be correctly updated on the backend.
- Given a user is editing their information and has unsuccessfully submitted the changes, an error message should be displayed.

- Given that the user navigates away from the account settings page without submitting any changes made, the changes should be discarded.

#### User Story #14

As a user (student), I want to be able to view my courses.

#	Description	Estimated Time (Hours)	Owner
1)	Implement backend functionality to send registered course information	3	<b>Xavier</b>
2)	Test that data is being sent correctly	1	<b>Xavier</b>
3)	Implement frontend functionality to update list of courses on Courses page	3	<b>Xavier</b>
4)	Integrate frontend and backend	2	<b>Xavier</b>
5)	Test frontend-backend integration	1	<b>Xavier</b>
<b>Total Hours</b>		<b>10</b>	

#### Acceptance Criteria:

- Given that a user is a student and logged in, when the user navigates to the Courses page, then the course information for that user should be displayed.
- Given that a user is a student and has been added to a course, when the user navigates to the Courses page then the course they have been added to should be displayed.
- Given that a user is a student and is enrolled in a course, when the user clicks on the course then they should be redirected to the course home page.

#### User Story #15

As a user (student), I want to be able to see the names and emails of other students in the course.

#	Description	Estimated Time (Hours)	Owner
1)	Create a “Class list” course sub-page in student view (no student management functionality)	2	<b>Xavier</b>
2)	Implement frontend functionality	2	<b>Xavier</b>

3)	Implement backend functionality to send class list data	3	<b>Xavier</b>
4)	Test that the data is being sent correctly	1	<b>Xavier</b>
5)	Integrate backend and frontend	1	<b>Xavier</b>
6)	Test frontend-backend integration	1	<b>Xavier</b>
<b>Total Hours</b>		<b>10</b>	

#### Acceptance Criteria:

- Given that the student user is on a course page, when the user clicks on the Class List button then the user should be redirected to a page displaying the name and email of the instructor and every student enrolled in the course.
- Given that the student user is on the Class List page, the backend should send a list of the instructor and every student in that course to the frontend to display to the user.
- Given that the student user is on the Class List page, the user should be unable to see instructor view functions.

#### User Story #16

As a user (teacher), I want to be able to see a list of all courses I am teaching.

#	Description	Estimated Time (Hours)	Owner
1)	Implement backend functionality to retrieve created course information.	2	<b>Xavier</b>
2)	Test that data is being sent correctly	1	<b>Xavier</b>
3)	Implement frontend to view information on the courses	3	<b>Xavier</b>
4)	Integrate frontend and backend	2	<b>Xavier</b>
5)	Test frontend-backend integration	1	<b>Xavier</b>
<b>Total Hours</b>		<b>10</b>	

#### Acceptance Criteria:

- Given an instructor is logged in, they should be able to access the Courses page.

- Given an instructor is on the Courses page, the instructor should be able to view all the courses they are teaching.
- Given an instructor is on the course view page, when an instructor clicks a course, information amount the course should be displayed

### User Story #17

As a user (teacher), I want to be able to add students to the course sections I am teaching.

#	Description	Estimated Time (Hours)	Owner
1)	Implement frontend functionality to add students on the “Class list” course sub-page	3	<b>Rudis</b>
2)	Implement backend functionality to handle adding students	1.5	<b>Rudis</b>
3)	Implement backend functionality to handle removing students	1.5	<b>Rudis</b>
4)	Test the backed to see that the data is being sent correctly	1	<b>Rudis</b>
5)	Integrate backend and frontend	2	<b>Rudis</b>
6)	Test frontend-backend integration	1	<b>Rudis</b>
<b>Total Hours</b>		<b>10</b>	

### Acceptance Criteria:

- Given a user is an instructor and is entering the Class List page, the user has access to the instructor-only functions.
- Given a user is an instructor and is on the Class List page, when the user clicks on Add Student, the user is able to add students to the class by entering the email of the student
- Given a user is an instructor and is adding a student, when the user attempts to add a student that does not exist, then an error message should be displayed.

### User Story #18

As a user (teacher), I want to be able to see the list of students registered for the course.

#	Description	Estimated Time (Hours)	Owner
1)	Create a “Class list” course sub-page in instructor view (with student management functionality)	5	<b>Joseph</b>
2)	Implement frontend functionality	1	<b>Joseph</b>
3)	Test that data is being sent correctly to the frontend	1	<b>Joseph</b>
4)	Integrate frontend and backend	2	<b>Joseph</b>
5)	Test integration	1	<b>Joseph</b>
<b>Total Hours</b>		<b>10</b>	

#### Acceptance Criteria:

- Given a user is an instructor and in a course, when the user clicks on the Class List button then the user should be directed to a page displaying the name and email of every student in the course.
- Given a user is an instructor and is viewing the Class List, the user should have the option to remove a student from the course.
- Given a user is an instructor and is viewing the Class List, the user should have the option to send an email to all students in the course.

## Remaining Backlog

- ~~1. As a user, I want to be brought to the login page upon visiting the website.~~
- ~~2. As a user, I want to register for a CampusLink account.~~
- ~~3. As a user, I want to request for an instructor account on the login page.~~
- ~~4. As a user, I want to log in to my account on CampusLink.~~
- ~~5. As a user, I want to log out from my CampusLink account.~~
6. As a user, I want to be able to reset my password if I forget it.
7. I want to be able to access the website from multiple devices and have the pages scaled properly for each device.
- ~~8. As a user, I want to be able to navigate the website using a navigation bar.~~
- ~~9. As a user, I want to be able to view my account settings page.~~
10. As a user, I would like to easily navigate to features common to all courses, such as the calendar.
- ~~11. As a user, I want to be able to access and search a frequently asked questions page.~~
- ~~12. As a user, I want to be able to edit my information on the account settings page.~~
13. As a user, I want to be able to post on the discussion board.
14. As a user, I want to be able to respond to a discussion post.
15. As a user (student), I want to receive update emails on any discussion posts I am a part of or have interacted with.
- ~~16. I want to have exclusive access to an admin page that displays all registered accounts and requests for instructor accounts.~~
17. As an admin, I want to be able to approve or deny professor account requests.
18. As a user (student), I want to be able to view past or submitted quizzes after the due date.
19. As a user (student), I want to be able to view my grades.
20. As a user (student), I want to be able to calculate my “what if” grade for all courses.
- ~~21. As a user (student), I want to be able to view my courses.~~
22. As a user (student), I want to be able to view course content.
23. As a user (student), I want to be able to take quizzes.
24. As a user (student), I want to be able to submit files.
25. As a user (student), I want to be able to enter the weights (temporarily) for the different graded items for each course.
- ~~26. As a user (student), I want to be able to login and have my courses be in student view.~~
- ~~27. As a user (student), I want to be able to see the names and emails of other students in the course.~~
28. As a user (student), I want to be able to bookmark assignments and lecture notes in a special bookmark tab.
29. As a user (student), I want to be able to view calendar page which shows due dates.
30. As a user (student), I want to get an email notification when the announcements page is updated.
31. As a user (teacher), I want to be able to create blank discussion boards.

32. As a user (teacher), I want to be able to create private or restricted discussions.
33. As a user (teacher), I want to be able to enter the assignment weights to the “what if grade” system.
- ~~34. As a user (teacher), I want to be able to create or delete courses I am teaching.~~
35. As a user (teacher), I want to be able to modify the courses I created.
- ~~36. As a user (teacher), I want to be able to see a list of all courses I am teaching.~~
- ~~37. As a user (teacher), I want to be able to add students to the course sections I am teaching.~~
- ~~38. As a user (teacher), I want to be able to see the list of students registered for the course.~~
39. As a user (teacher), I want to create quizzes, such as multi-choice or short answer quizzes.
40. As a user (teacher), I want to create assignment submissions.
41. As a user (teacher), I want to set due dates for assignment and quiz submissions.
42. As a user (teacher), I want to set the number of submissions for assignments.
43. As a user (teacher), I want to set the duration for quizzes.
44. As a user (teacher), I want to set the number of attempts for quizzes.
45. As a user (teacher), I want to see how many students submitted assignments.
46. As a user (teacher), I want to be able to post lecture slides or notes to the course.
47. As a user (teacher), I want to be able to upload videos to the course (such as lectures or other media).
48. As a user (teacher), I want to be able to embed videos to the course content (such as lectures or other media) using a URL.
- ~~49. As a user (teacher), I want to be able to login and have the page be in an instructor view.~~
50. As a user (teacher), I want to be able to enter grades for the students in each course.
51. As a user (teacher), I want to be able to compute and display a class average for each assignment and an overall grade.
52. As a user (teacher), I want to be able to update an announcements and calendar page.
53. (*If time allows*) As a user (teacher), I want to be able to view how many times a student has viewed the course page/content.