Team 9 Sprint 1 Planning Document

CS 307 Fall 2022

Purdue Course Finder

Alex Kobus, Alex Plump, Tommy Lane, Peter Zong

Sprint Overview

During this sprint, we want to set up the front end of our website using React. The backend setup will use Spring Boot. We want to have the database initialized so that data can be stored. When users create an account, we want the necessary classes and server interactions to pass data exist. Last, we hope to create a dedicated page for signing up and logging in this sprint.

Scrum Master: Alex Kobus

Meeting Plan: 7 PM – Monday/Wednesday

Risks and Challenges:

Since this is the first Sprint, there might be some issues with dependencies. Since frameworks might need to be chosen before direct coding, testing of frameworks at the beginning may take some time in the first week. Additionally, figuring out how to connect and configure some of the working parts may cause delays. We will need the front end of the website working and functional before we can start displaying our interactive map.

2. Current Sprint Detail

User story 1

As a User, I would like to see current Purdue courses, and their section times, professors, and locations.

#	Task Description	Estimated Time	Developer
1	Forward information from Purdue API in backend	4 hours	Tommy
2	Create a prototype UI to display the data for each course	2 hours	Alex K
3	Connect frontend to backend	1 hour	Alex K

Acceptance Criteria

- Given the backend forwards information from the Purdue API, when calling defined endpoints in the backend, course information should be returned correctly.
- Given the API forwarding is implemented correctly, when querying the database, previously requested classes will be found to be cached in the database
- Given the frontend is finished correctly, when viewing courses, the information will be displayed accurately and obviously

User story 2

As a Professor, I would like to see the number of students registered in a section.

#	Task Description	Estimated Time	Developer
1	Create an endpoint for the number of registered students in a section	1 hour	Tommy
2	Create a prototype UI to display the number of students registered in each section of each course	2 hours	Alex K
3	Connect frontend to backend	1 hour	Alex K

Acceptance Criteria

- Given the backend can correctly access each department's courses, when a professor selects a course, then a legitimate request is sent to the backend.
- Given the backend forwards information from the Purdue API, when calling defined endpoints in the backend, the number of registered students should be returned correctly.
- Given the frontend is finished correctly, when viewing course registration, the information will be displayed accurately

User story 3

As a User, I would like a dedicated page for signing up.

#	Task Description	Estimated Time	Developer
1	Initialize React Project & setup packages	2 hours	Alex K
2	React Tutorials and setup local React environments	3 hours (each)	Peter, Tommy, Alex P
3	Lay out and finalize page design	1 hour	Alex K
4	Create signup page UI	3 hours	Alex K
5	Connect input fields and buttons to send credentials to server	3 hours	Alex K
6	Implement email/password format checks	2 hours	Alex K
7	Setup placeholder actions depending on server login response	2 hours	Alex K
8	After user story 5 is done, connect server to database (user credentials)	2 hours	Tommy
9	Create unit tests and perform manual tests to ensure correct page functionality	3 hours	Alex P

Acceptance Criteria

- Given that the input fields and buttons correctly connect to the server, when the Sign-Up button is pressed, then text in the email and password fields should be sent to the server.
- Given that the server sends back correct status codes, when user signup info is sent to the server, the login page should display either an “email already exists” error or placeholder text for a page redirect.
- Given that the email/password format checks are implemented correctly, when the email or password is in the wrong format or when the password field doesn’t match the confirm password field, then the page displays the appropriate error message.
- Given that user story 5 is complete, when user credentials are sent to the server, then the server sends the info to the database and either the info is stored as a new record or the email already exists, returning an error code.

User story 4

As a User, I would like a dedicated page for logging in.

#	Task Description	Estimated Time	Developer
1	Lay out and finalize page design	1 hour	Peter
2	Create login page UI	3 hours	Peter
3	Setup placeholder actions depending on server login response	2 hours	Peter
4	After user story 5 is done, connect server to database to check user credentials	2 hours	Peter
5	Create unit tests and perform manual tests to ensure correct page functionality	3 hours	Alex K

Acceptance Criteria

- Given that the page design is finalized, when the login page UI is created, then it should look almost identical to the mockup.
- Given that the input fields and buttons correctly connect to the server, when the Log In button is pressed, then text in the email and password fields should be sent to the server.
- Given that the server sends back correct status codes, when user login info is sent to the server, the login page should display either an email/password not fount error or placeholder text for a page redirect.
- Given that user story 5 is complete, when user credentials are sent to the server, then the server sends the info to the database and either a match is found, returning user profile info or a match is not found, returning an error code.

User story 5

As a Developer, I would like to store user account information in a SQL database.

#	Task Description	Estimated Time	Developer
1	Setup initial Spring Boot backend configuration	2 hours	Tommy
2	Spring Boot tutorials and setup local Spring Boot environments	3 hours (each)	Alex K, Alex P, Peter
3	Docker tutorial, testing, prototyping	4 hours (each)	Alex P, Alex K, Peter
4	Sample various databases within Docker Containers	3 Hours	Alex P
5	Setup database as Docker container	3 hours (each)	Tommy, Alex P
6	Integrate Spring Boot back-end with the Database	4 hours	Tommy
7	Define database schema in backend	4 hours	Tommy
8	Learning SQL Format, Commands: Selecting, Adding, Modifying, and Removing entries.	2 hours (each)	Alex P, Alex K, Peter, Tommy
9	Adding and removing items from Database using Backend	3 hours (each)	Tommy, Alex P
10	Containerize the backend using Docker	3 hours	Tommy
11	Create docker compose file	2 hours	Tommy

Acceptance Criteria

- Given the initial Spring Boot configuration is done correctly, when the backend is set up, it will start correctly.
- Given the integration with database is done correctly, when the backend is started, it will continue when the database is running but give an error when the database is not running.
- Given the database is setup correctly, when starting the Docker container, the backend will be able to connect to it.
- Given the database schema is defined correctly, when utilizing the schema, all the expected values and relationships will be present.

- Given the database can be modified correctly, when calling the appropriate sample endpoint, data will be added or removed from the database
- Given the backend has been containerized correctly, when starting the Docker container, the backend can be connected to by a client
- Given the docker compose file is created correctly, when running docker compose, then the backend and database start and operate together correctly.

User story 6

As a User, I would like to see a tutorial that explains how to use the software.

#	Task Description	Estimated Time	Developer
1	Lay out and finalize page design	1 hour	Peter
2	Create page UI	2 hours	Peter
3	Add UI Mockups from the Design Document and describe how to use them	1 hour	Peter
4	Add images of the completed pages for logging in and creating accounts and describe how to use them	1 hour	Peter

Acceptance Criteria

- Given the page design has been finalized, when viewing the design, it should be clear how to proceed with the implementation.
- Given the page UI has been correctly created, when viewing the tutorial page, it should be clear how to use the existing functionality.
- Given the images have been correctly added, when viewing the tutorial page, the images of the pages being described should be visible.

User story 7

As a User, I would like to delete an account.

#	Task Description	Estimated Time	Developer
1	Create a page for account deletion	1 hour	Alex P
2	Add buttons and pop-up when pressed	2.5 hours	Alex P
2	Search and delete account credentials from database	1.5 hours	Alex P
3	Unit tests	1 hour	Alex P

Acceptance Criteria

- Given the page is set up correctly, when a user redirects to the account deletion page, they are displayed a delete account button.
- Given the button's actions are configured correctly, when a user presses the delete account button, then another pop-up appears asking if they are sure they want to delete their account.
- Given the pop-up is working correctly, when a user confirms, the server sends a delete query to the database.

User story 8

As a User, I would like to change my account email and password.

#	Task Description	Estimated Time	Developer
1	Create a page for account modification	2 hours	Peter
2	Find and change database email/password entry	2 hours	Peter
4	Unit tests	1 hour	Peter

Acceptance Criteria

- Given the page is set up correctly, when a user redirects to the account modification page, they are shown their email and asterisks as their password.
- Given the page is set up correctly, when a user redirects to the account modification page, they are given 2 buttons, the option of changing either the email or password.
- Given the button actions work correctly, when a user clicks the buttons, they input their new email or password in a text field.
- Given the server-database interaction works correctly, when the server gets a request to change email or password, the server queries the database to modify the user's info.

User story 9

As a Developer, I would like to save encrypted account login information in a database.

#	Task Description	Estimated Time	Developer
1	Learn and employ encryption to client-side message passing	4 hours	Tommy
2	Test to ensure data is properly encrypted and saved	1 hour	Tommy

Acceptance Criteria

- Given the account information is properly encrypted, when viewing the emails stored in the database, then we should not be able to tell what the original emails are.
- Given the account information is properly encrypted, when viewing the passwords stored in the database, then we should not be able to tell what the original passwords are.
- Given the encryption logic is implemented correctly, when a user views their own email, then it should display the original, unencrypted version.

User story 10

As a Developer, I would like to save backend logs of what pages are accessed most frequently.

#	Task Description	Estimated Time	Developer
1	Log each page visited by users to the server	2 hours	Tommy
2	Save logs to a file	2 hours	Tommy
3	Test that logs are being generated and saved correctly	1 hour	Tommy

Acceptance Criteria

- Given that all pages load correctly, when they are accessed, then the client should send a request to load to the server.
- Given client-server interactions are working, when the server receives a message that a page is being loaded, the server should log what page is being accessed.
- Given the server can save page logs to a file, when the server wants to log a page, then it stores the page to a file on the backend.

3. Backlog

1. Usage

As a User,

a. ~~I would like to see a tutorial that explains how to use the software.~~
b. I would like to see an option to select a specific semester to view the courses offered that semester.

2. Map

As a User,

a. I would like to see a birds-eye view map of Purdue.
b. I would like to search the map for a building's location.
c. I would like to see a sidebar that shows a list of all classes and buildings.
d. I would like to see campus buildings highlighted on the map.
e. I would like to see campus buildings labeled on the map.
f. I would like to be able to click a highlighted building on the map to see more information about it, the rooms in it, and the classes in it.
g. I would like to filter the buildings that I see highlighted on the map by name, class location, and section location.

3. Account

As a User,

~~a. I would like a dedicated page for signing up.~~
~~b. I would like a dedicated page for logging in.~~
~~c. I would like to create an account.~~
~~d. I would like to delete an account.~~
~~e. I would like to change my account email and password.~~

4. General Filter

As a Student,

a. I would like to filter the map and sidebar to show only relevant courses, course locations, course sections, and section locations.

b. I would like to filter the map and sidebar to show only relevant buildings, classrooms, and meeting times.
c. I would like the option to manually input parameters to search for a course/building in case the filters return no results.

5. Filter by Favorites

As a User,

a. I would like to create a list of my favorite buildings and classrooms.
b. I would like to create a list of saved (favorite) classes and sections.
c. I would like to filter the map by my favorite buildings and rooms.
d. I would like to filter the map by my favorite classes and sections.

6. Personal Schedule

As a Student,

a. I would like to create a personal schedule for all my classes in a week.
b. I would like to see the estimated time it takes to walk between my classes.
c. I would like to see the estimated time it takes to bike between my classes.

7. Courses and Classrooms

As a User,

~~a. I would like to see current Purdue courses, and their section times, professors, and locations.~~
b. I would like the sidebar to show classrooms in a building after selecting that building.
c. I would like to select a classroom from the sidebar to see more information about it and the classes in it.

As a Student,

d. I would like to see a schedule page that shows all class meetings in a selected room each week.
e. I would like to see a schedule page that shows all meetings of a selected class in each week.

As a Professor,

f. I would like to see the number of seats in a classroom.
g. ~~I would like to see the number of students registered in a section.~~
h. I would like to see statistics that show how often a classroom or building is used. (if time allows)

8. Miscellaneous

As a User,

a. I would like to see a suggestion page to request features (if time allows).

9. Developer

As a Developer,

a. ~~I would like to store user account information in a SQL database.~~
b. ~~I would like to save encrypted account login information in a database.~~
c. ~~I would like to save backend logs of what pages are accessed most frequently.~~
d. I would like to restrict schedule creation specifically to users that are logged in.
e. I would like the application to be hosted on AWS.

Non-Functional Requirements

1. As a User, I would like the application to be easy to understand and intuitive.
2. As a User, I would like the application to be fast.
3. As a User, I would like the application to be secure.
4. As a Developer, I would like to optimize the web application, so that response times are lower than 1 second.
5. As a Developer, I would like to optimize the web application, so that page loading times are lower than 500 ms.
6. As a Developer, I would like to create the application in a way that allows it to be adapted to include other universities. (if time allows)
7. As a Developer, I would like to have developer roles on accounts for testing and any account resolutions (like resetting passwords). (if time allows)