**MEETING #7 2/25/2021 -** The team met for close to an hour

- Wellness check

* The team met to discuss where they are at in the project and work on the deliverable for this week.
* We uploaded everything we had and are getting everything ready to send in.
* The last thing we are working on together is system architecture.
* Updated Trello
* Michael and Diego will regroup Saturday to make sure everything for deliverable 4 is all set

**Meeting #6 2/21/2021** – The group met for an hour and a half.

- Wellness Check

- Discussion of UI design

Sim showed the group what he had for java code, design is the main goal right now. Getting a good skeleton set up is important to us.

The group is looking at videos of connecting SQL and java.

- Currently discussing system architecture

- Discovered how to store username and passwords safely in SQL database, we were debating if we needed another database. We now only need to stay with one.

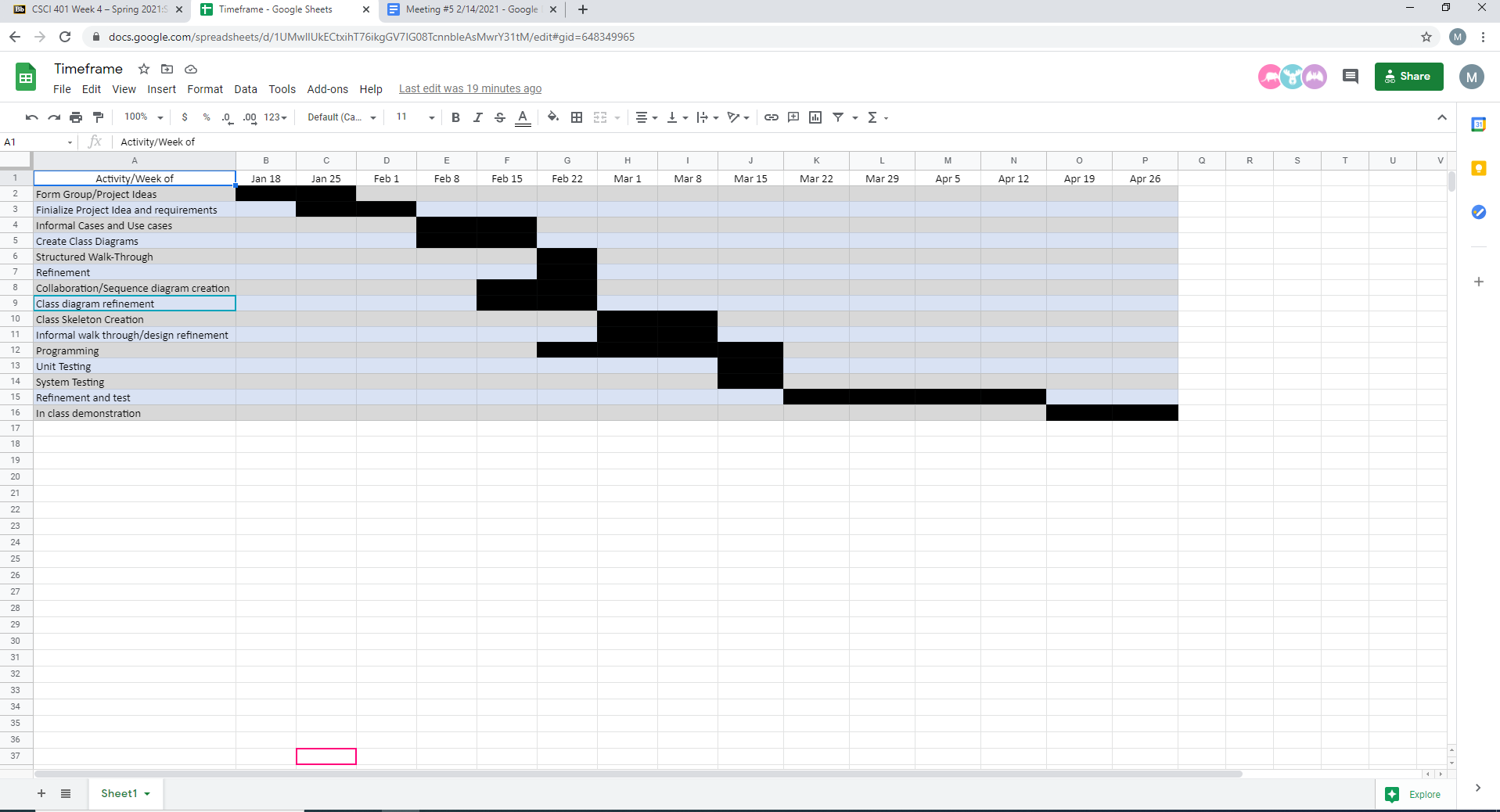
- Made a word doc to fill with bonus feature ideas.

**MEETING 5 2/14/2021---**

The team met for an hour and made progress on deliverable 3

-Wellness check

-Team worked on Deliverable 3



SRS Document below

https://docs.google.com/document/d/1-laRvPKKZClX\_Hak0yoiAMYdnbRLlepzPL1OmA2qbNE/edit

**MEETING #4 2/11/2021**

Minutes: The group met for about an hour and a half.

Events:

Wellness check

We worked on Deliverable 3, we worked on informal scenarios and use cases.

* Updated trello.

INFORMAL SCENARIO #1: RIC Student tries to add a class that is full.

Current State: Diego logs in and is looking for a class to add. The list of classes are displayed to him.

Informal Scenario: Diego selects the class and hits the button to add the class. Diego didn’t notice that the class is already full.

Next Scenario: A message pops up notifying Diego that the class is full. He will be able to choose a different class to add.

INFORMAL SCENARIO #2: Time conflicts

Informal Scenario: Student wants to add a class to their schedule.

Current state: Student has selected a class to add to their schedule.

Informal scenario: The student attempts to add the class to their schedule, however there is a scheduling conflict. with a class already added to their schedule, they cannot attend two classes at the same time.

Next scenario: An error message is displayed, indicating that there is a scheduling conflict between the class they are adding and the other conflicting class. For the student to add this class, they will have to make sure there are no scheduling conflicts with other classes.

# **Use Case: Logging in**

Precondition: User starts the program

Main flow of events: Use case starts when a student attempts to log in, they input a user name and password. The system checks if the username and password are both valid. If valid, they are allowed into the application, thus ending this use case.

Exceptional flow of events: If the user enters a invalid username and a valid password, the system prompts: invalid username and doesn’t allow login, and thus this use case ends.

Exceptional flow of events: If the user enters a valid username and a invalid password, the system prompts: invalid password and doesn’t allow login, and thus this use case ends.

# **Use Case: Drop Class**

Pre condition: User has logged in with a valid username and password.

Main flow of events: User wants to remove a class from their schedule, they navigate to the Drop class section and selects the class they want to drop, the system confirms that they would like to drop a class. The student confirms and the class is removed from the schedule, thus ending the use case.

Exceptional flow of events: Student tries to remove a class, and there is no class in the schedule, thus ending the use case.

Exceptional flow of events: Student cancels the request to drop the class, thus ending this use case.

# **Use Case: Add class**

Precondition: User starts application. User successfully validates their identity with a valid username and password. User applies desired filters to available classes

Main Flow of Events: User selects a class from the filtered list. User attempts to add class by clicking add button.

Exceptional of Events: The class is successfully added to the user's schedule

Exceptional Flow of Events: An error message displays telling the user that the class is already at full capacity.

Exceptional Flow of Events: An error message displays telling the user that the class that there is a time conflict with another class they has already added

Exceptional Flow of Events: An error message displays telling the user that they do not meet the requirements to take that class

**MEETING #3 2/04/2021**

Team met for a half hour to double check Deliverable 2 before sending it in. We met for a half hour.

-Updated Trello.

**MEETING #2 1/31/2021**

Meeting Time: 7:50-9

Next Meeting – Thursday February 4th – Check in, Double check deliverable 2

Team Wellness check

- All members check in and break the ice.

TEAM NAME

Settled on “Elite 4”

TEAM PROJECT

RIC Scheduling Application

Use SQL with PHP myAdmin

**Meat and Potatoes:**

**User interface**

Java - Eclipse

**Database**

MYSQL

PHPMyadmin

xxAMP

Our team’s goal is to make a MYRIC scheduling application using Java, SQL, and PHPMyadmin. This allows students to create and modify their schedule.

Project Goals/Main Functionality:

- Add class

-Drop class

-View Schedule

- Pre-requisite courses met

-Dates and Times

- Handle Scheduling Conflicts

Extra add Ons (If time allows)

- Credits

- Built in advisor

- Registration Period

**Team Roles:**

Diego: Project Manager, Research

Mike: Writer, Research

Callum : PHPMyAdmin and SQL

Simeon : Java and UI

**Things to learn:**

PhpMyAdmin/SQL

Build upon Java skills

Our team’s goal is to make a MYRIC application using Java, SQL, and PHPMyadmin.

Learn to build Java UI

Learn how to connect PHPMyAdmin and java

**Meeting 1:**

We started off introducing ourselves to each other and discussed roles and skills

The group met for a little over an hour

**ROLES:**

Diego Gomez - Project Manager

Callum Moe - Coder

Simeon Sabo - Coder

Michael Goldberger - Writer

*Note: Additional roles may be added to each team member.*

**(All members have compatible hardware)**

**CODING SKILLS**

MICHAEL: *JAVA AND SQL*

SIM - *JAVA AND C# C C++ PYTHON*

CALLUM MOE - *JAVA SQL*

DIEGO - *JAVA AND C# C C++ PYTHON CYBER SECURITY*

***Note: All members are to interact with/understand the code in some way. No free rides through the course. The team is only as strong as the weakest link.***

**Possible Projects**

Website - Flashcard program/app

Program - Java/SQL based project

Applications - MyRic Mobile / Mobile RIC website

JAVA & SQL - RIC Class picking project

**Note: We aren’t sure about the scope/requirements of the project yet. This will impact the decision making process.**

**Meeting Times**

**Default meeting time: Sunday - 8pm**

**Secondary meeting time: Thursday - 9pm**

**Will Ramp up meeting times to accommodate workload**

**MTG Time - ??? Not sure what this means yet**

Professor was invited to both Google Drive & Trello

For Monday: More project ideas

**IDEA DUE + Team Name: 2/3/21**