Team Project Deliverable 2:

**UML, Design Patterns, Scrum, Risk Management**

**CSC 131 – Fall 2021**

Dane Coleman

Santiago Bermudez

Enoch Hsu

Eric Truong

Tyler Ito

Ramo Tucakovic

Chris Long

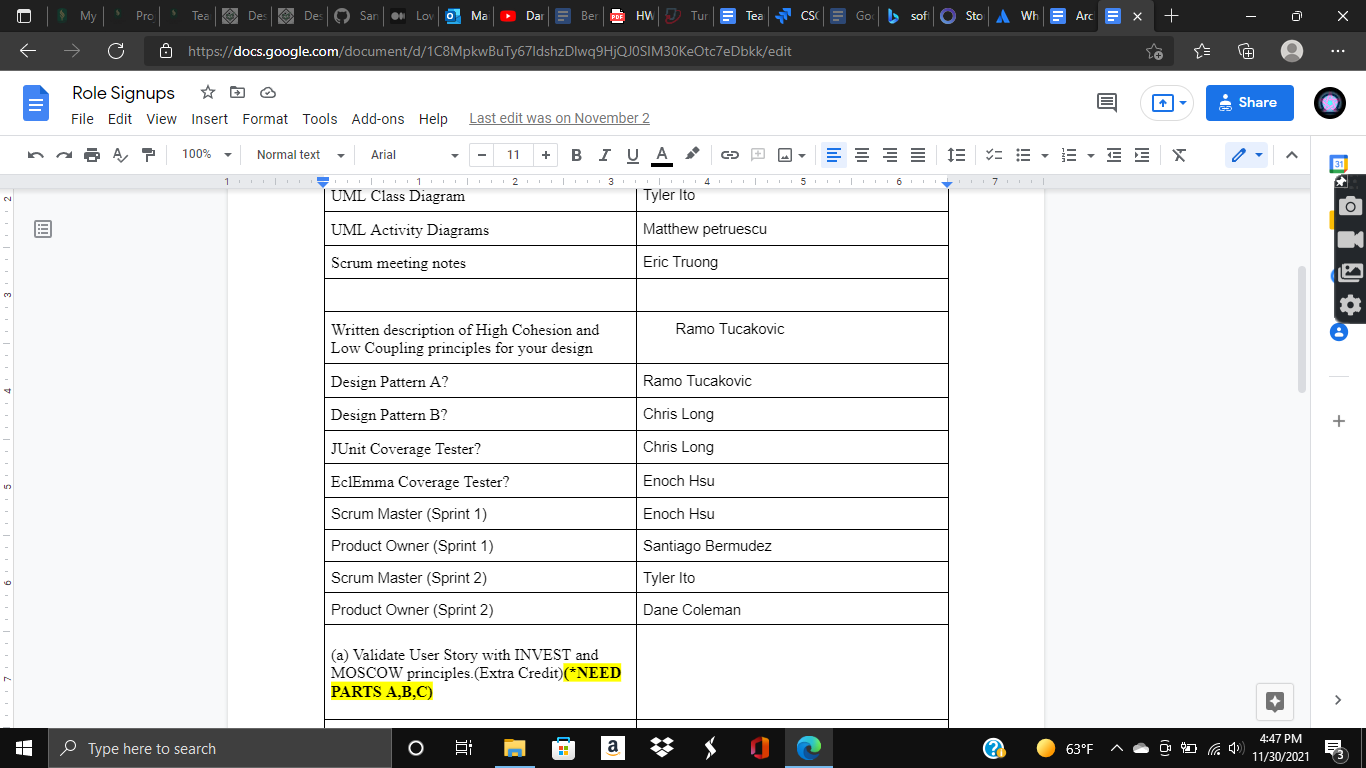
Matthew Petruescu

**1. Project Management/Scrum. All Scrum meetings and Scrum process recorded in a PDF document. Sprints can be any number:**

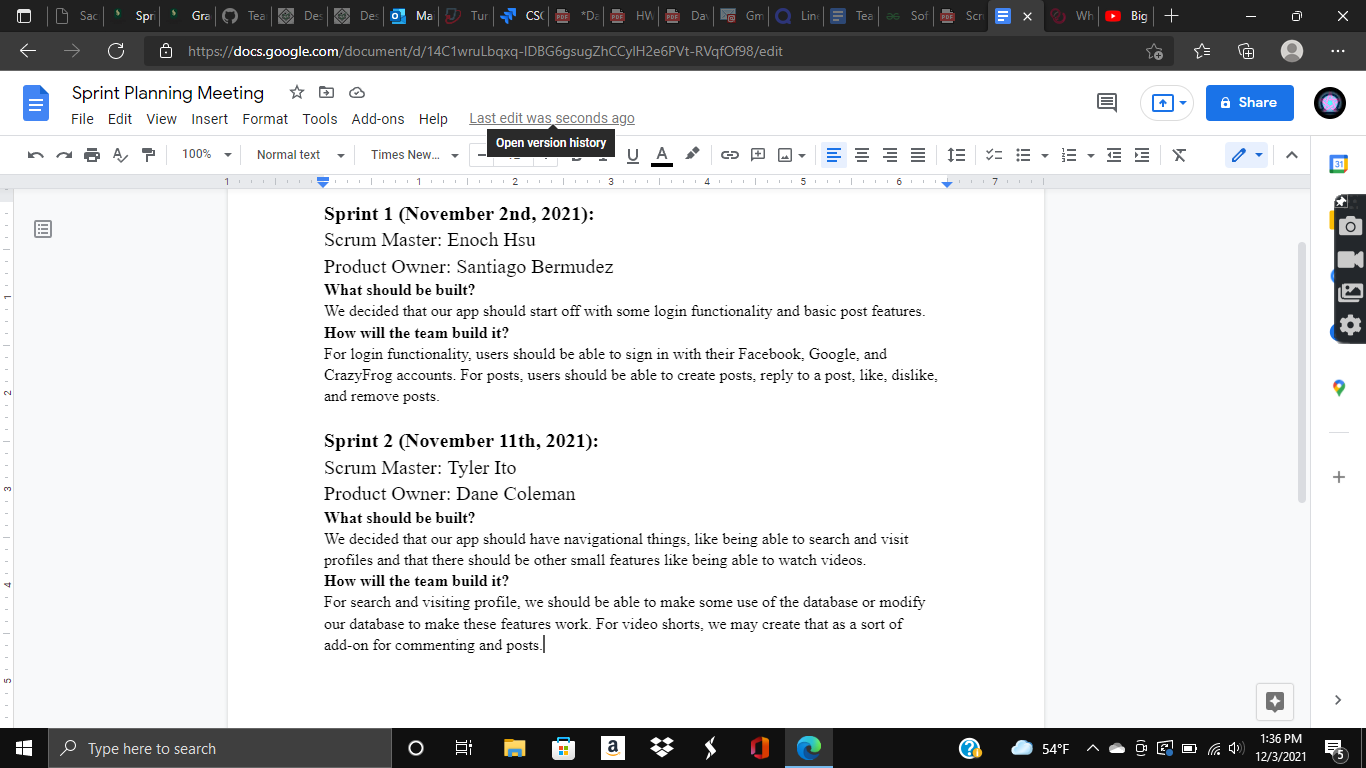
1(a) Assignment of Scrum Master and Product Owner (rotate roles among team members) (5 points)

| Scrum Master (Sprint 1) | Enoch Hsu |
| --- | --- |
| Product Owner (Sprint 1) | Santiago Bermudez |
| Scrum Master (Sprint 2) | Tyler Ito |
| Product Owner (Sprint 2) | Dane Coleman |

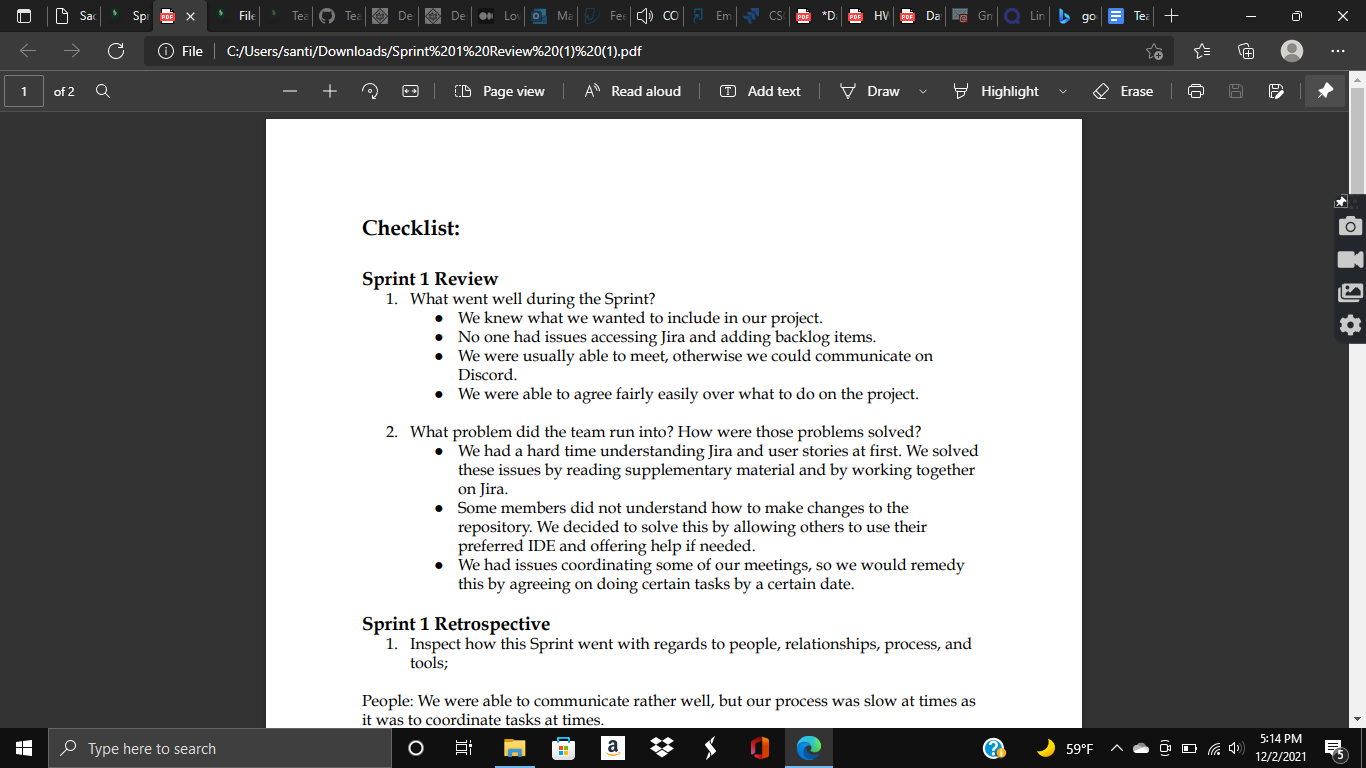
\*We actually had sign ups for a whole bunch of stuff as listed below, but you get the point.

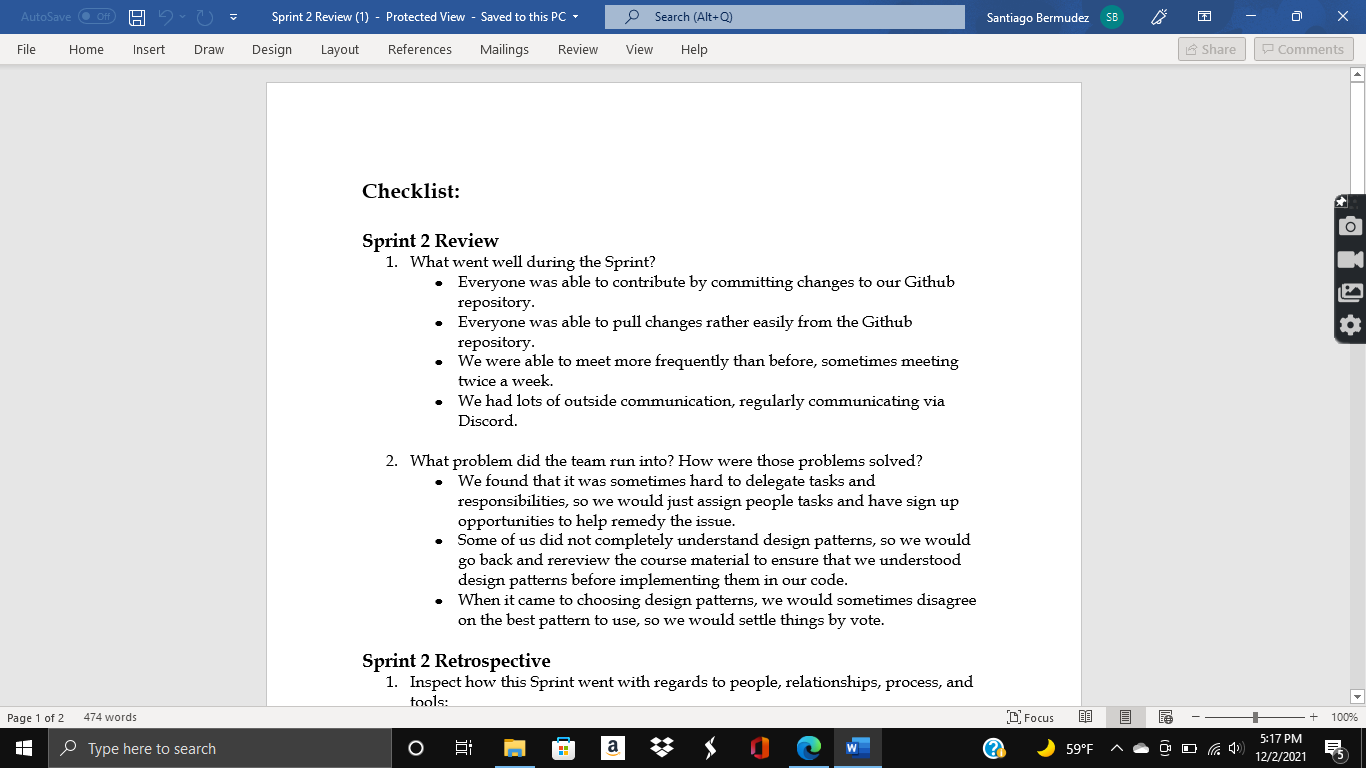


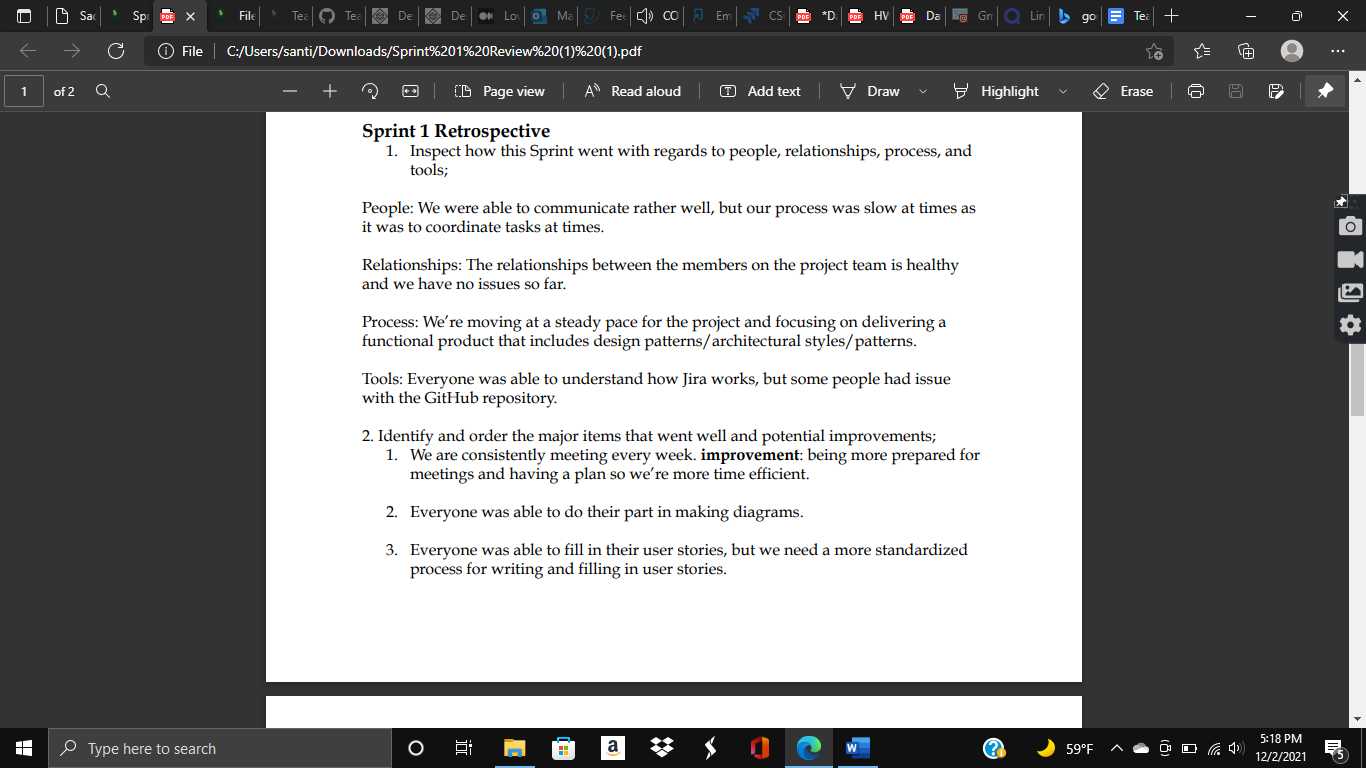
1(b) Sprint planning meeting (5 points)

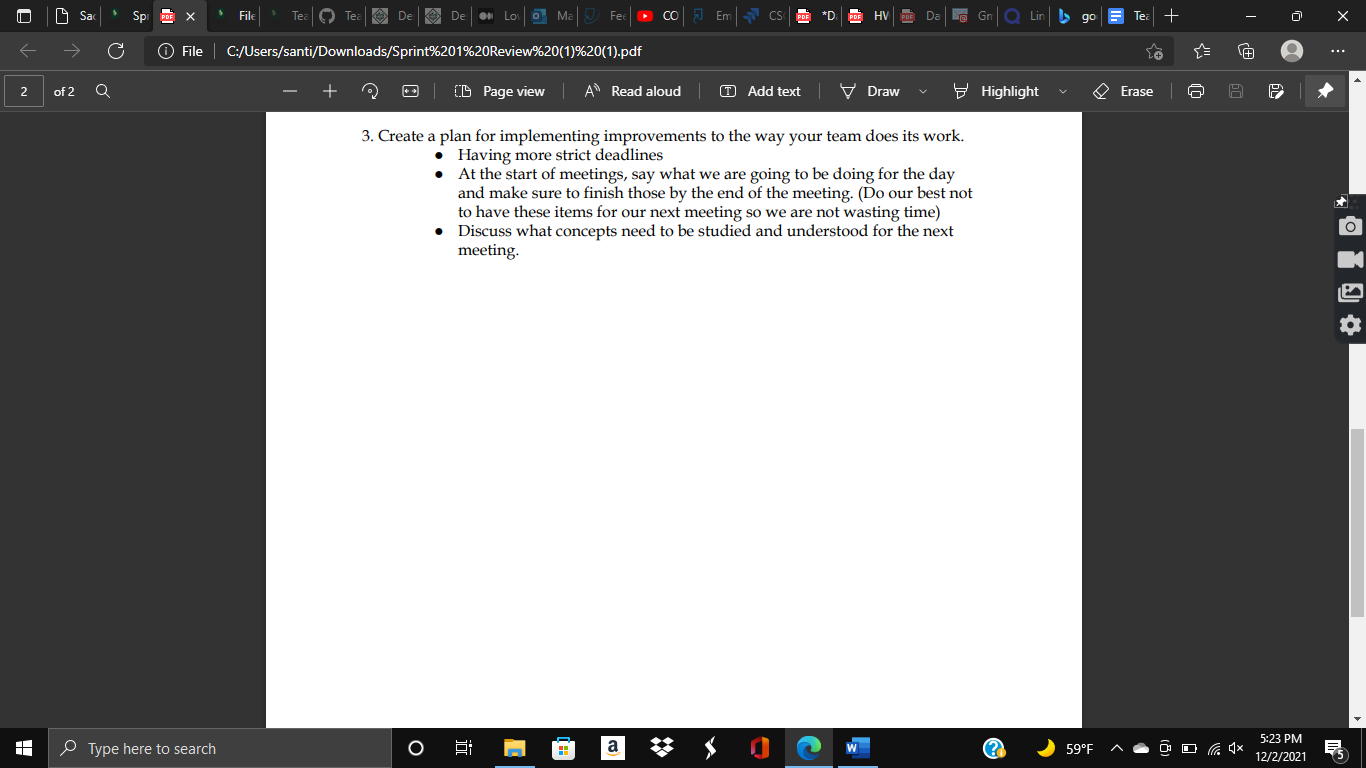


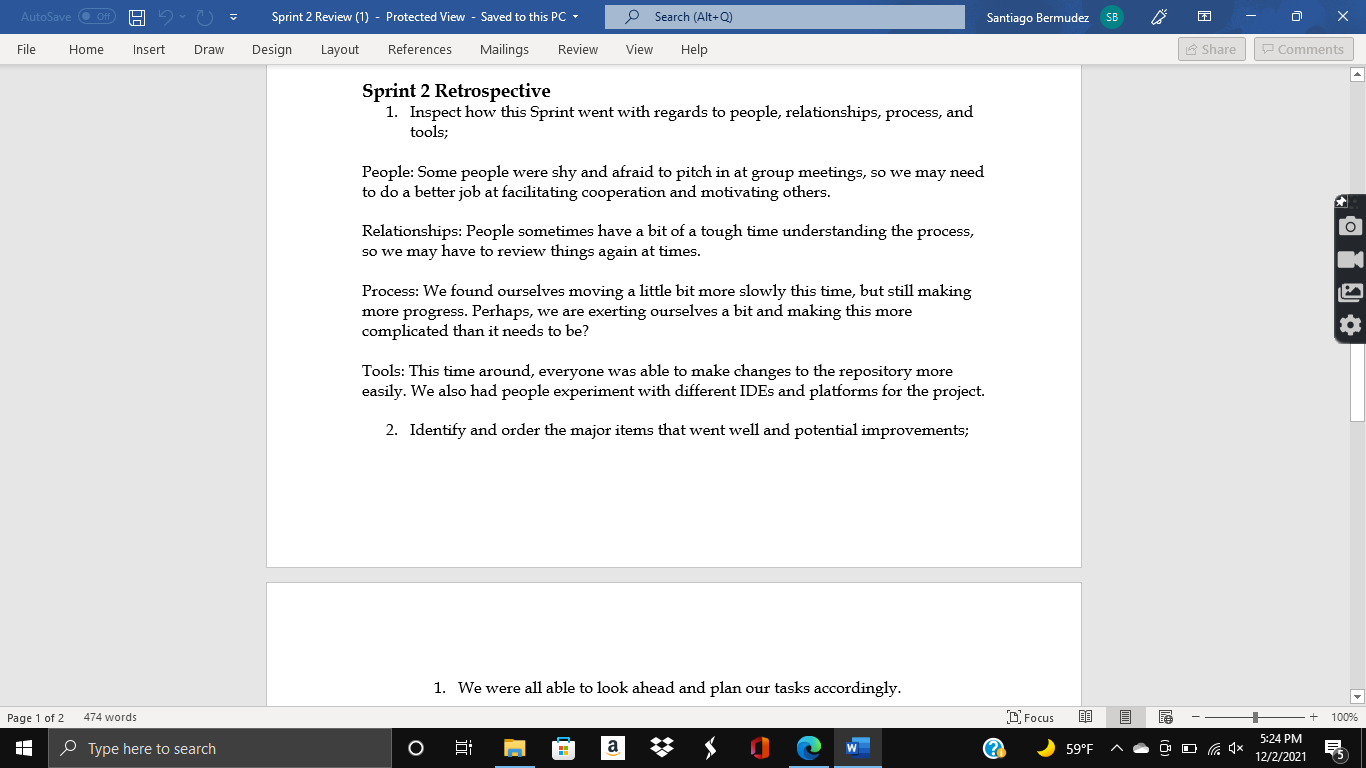
1(c) Sprint Review (5 points)

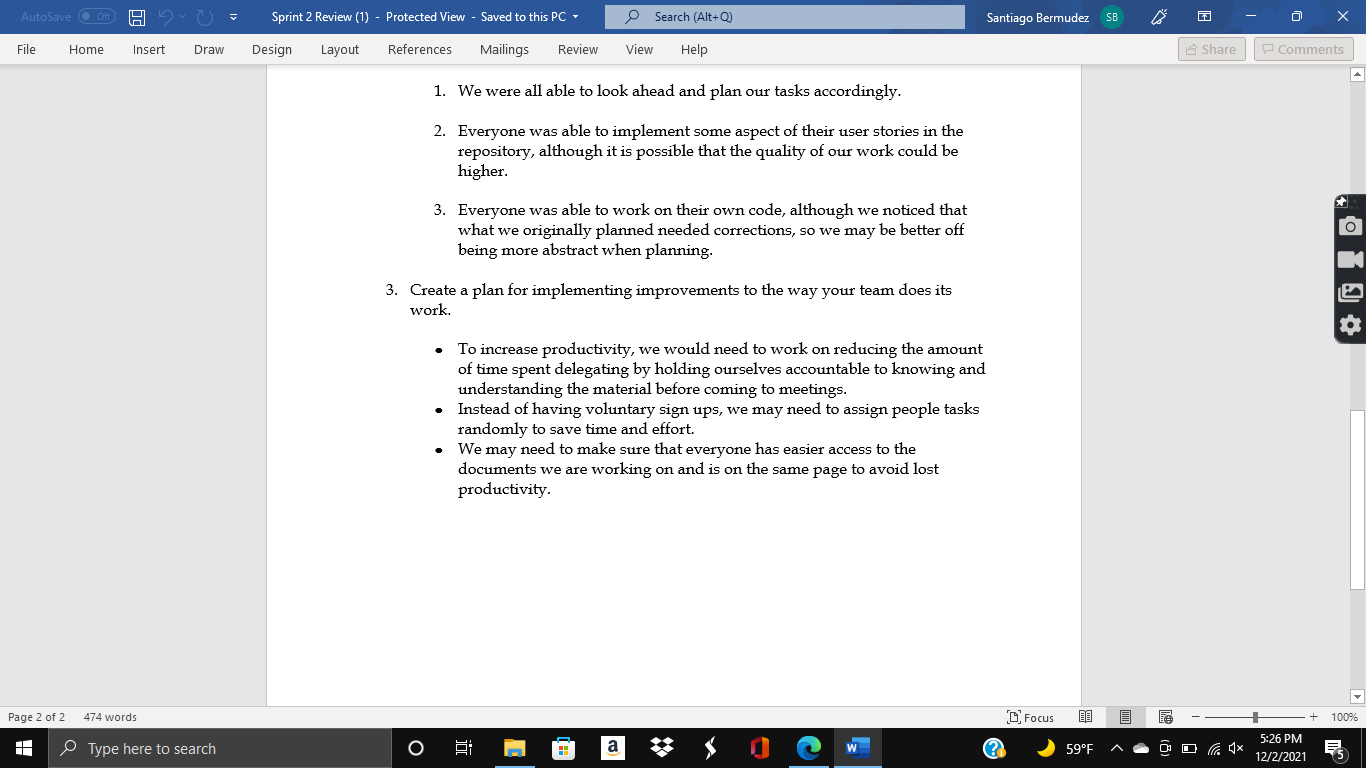




1(d) Sprint Retrospective (5 points)







**2. Handling User Stories (minimum 5 user stories) and use of unambiguous requirements in Scrum tool:**

**\*Link to our Jira below:**

[**https://crazy-frog-incorporated.atlassian.net/jira/software/projects/CSC131TP/boards/2/backlog**](https://crazy-frog-incorporated.atlassian.net/jira/software/projects/CSC131TP/boards/2/backlog)

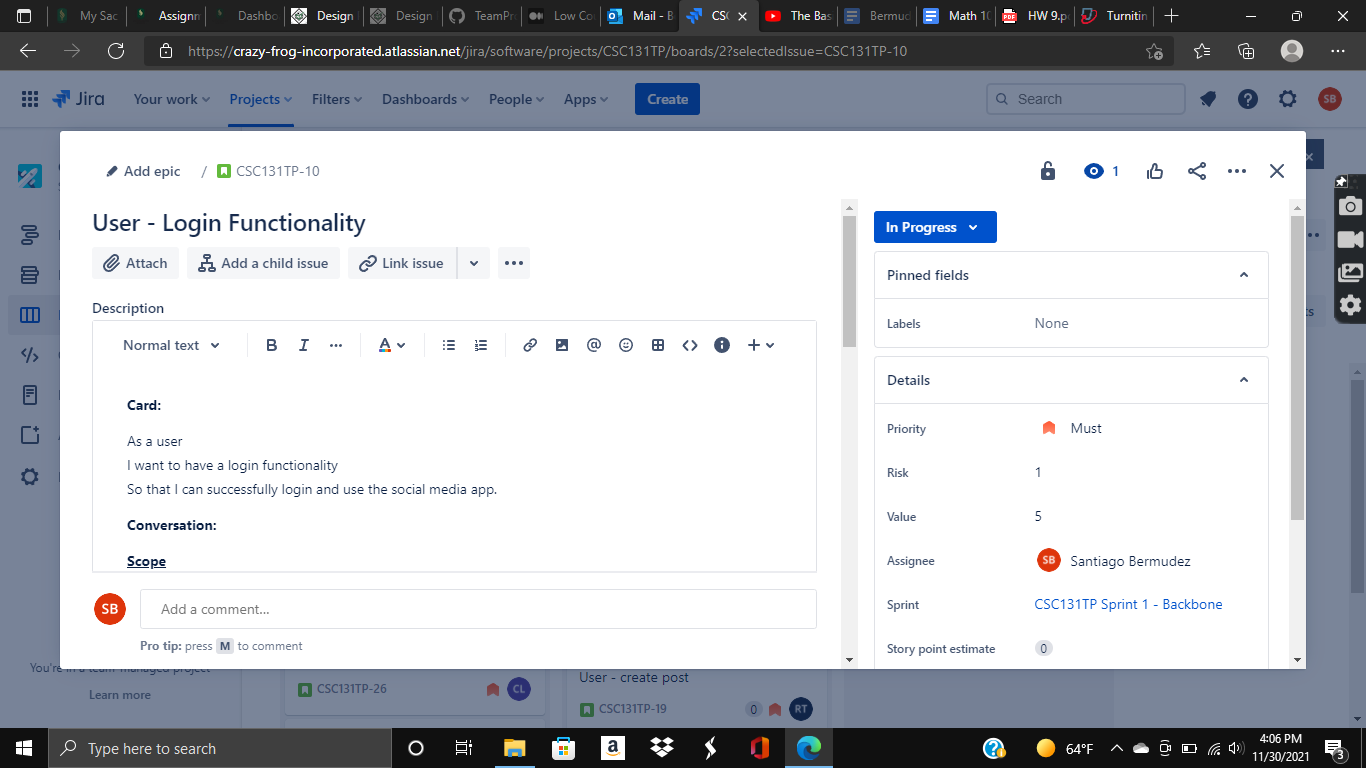
**\*You guys should have access hopefully, but if not, contact:**

**santiagoabermudez@gmail.com**

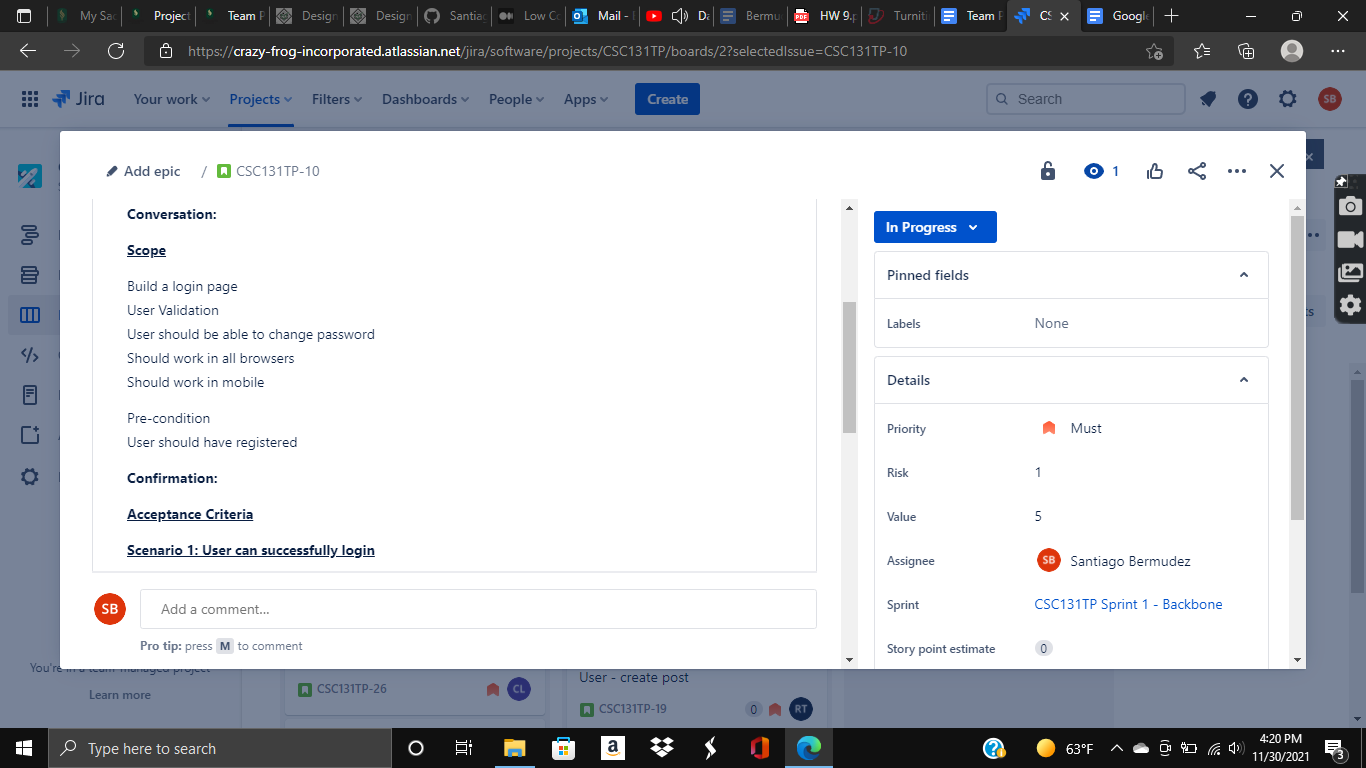
Use of any Scrum tool such as Vivify, Flying Donut or Atlassian/JIRA with:

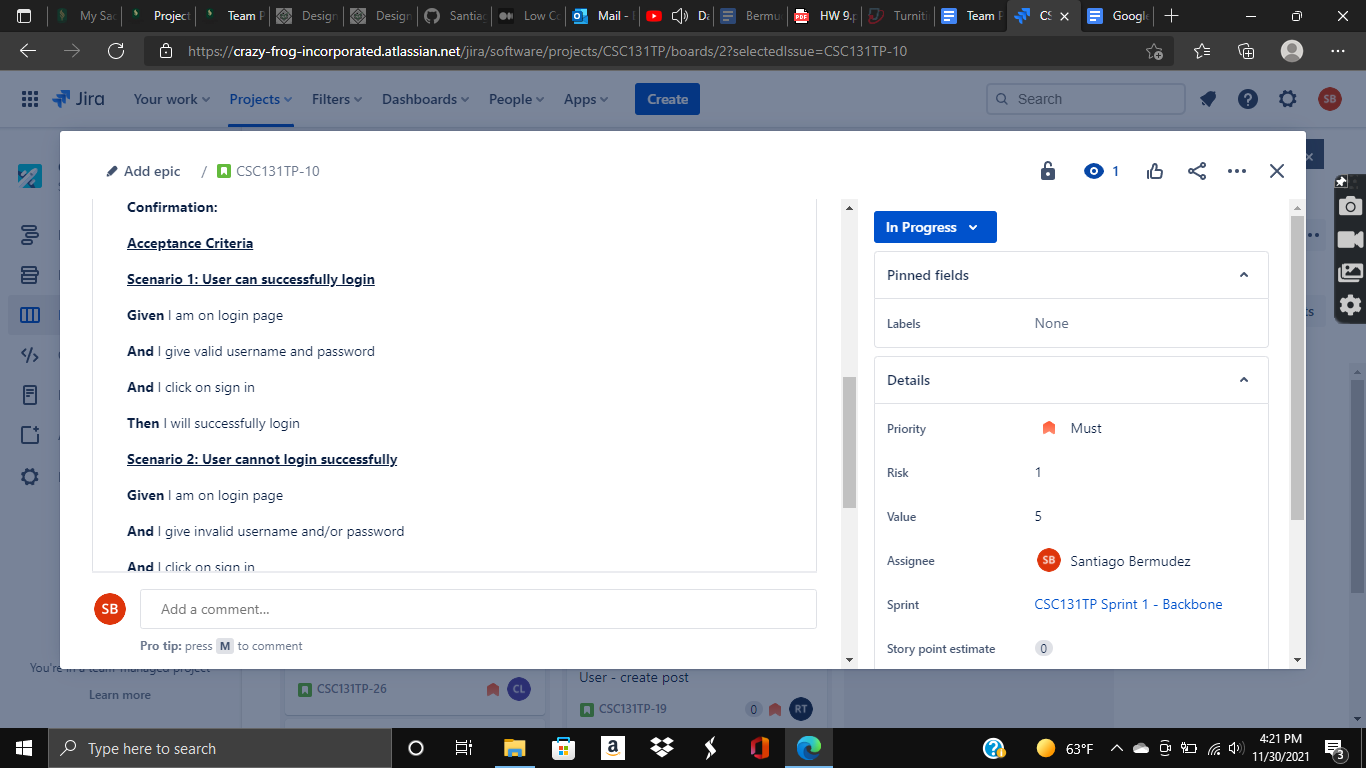
2(a) two examples of CCC (Card, Conversation and Confirmation) in PDF document: (2\*2 = 4 points)

Example 1:

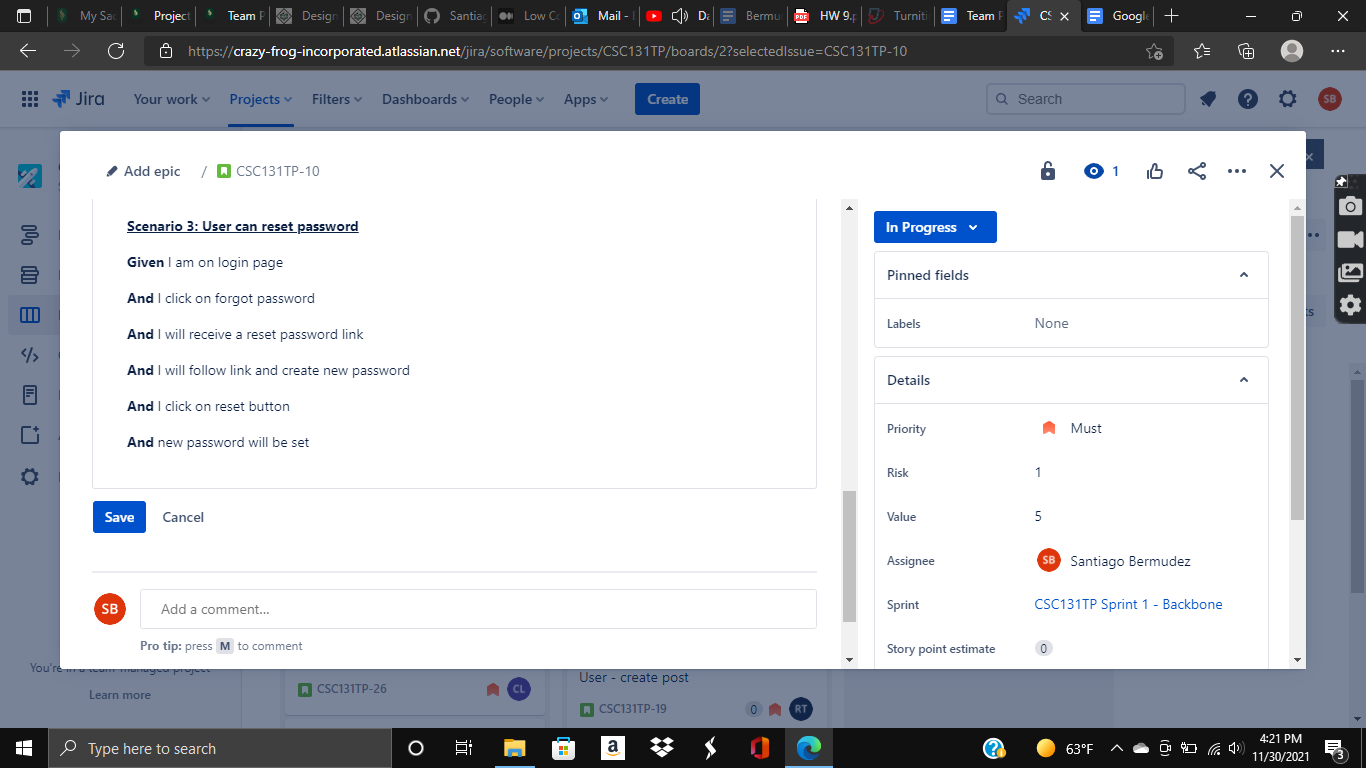




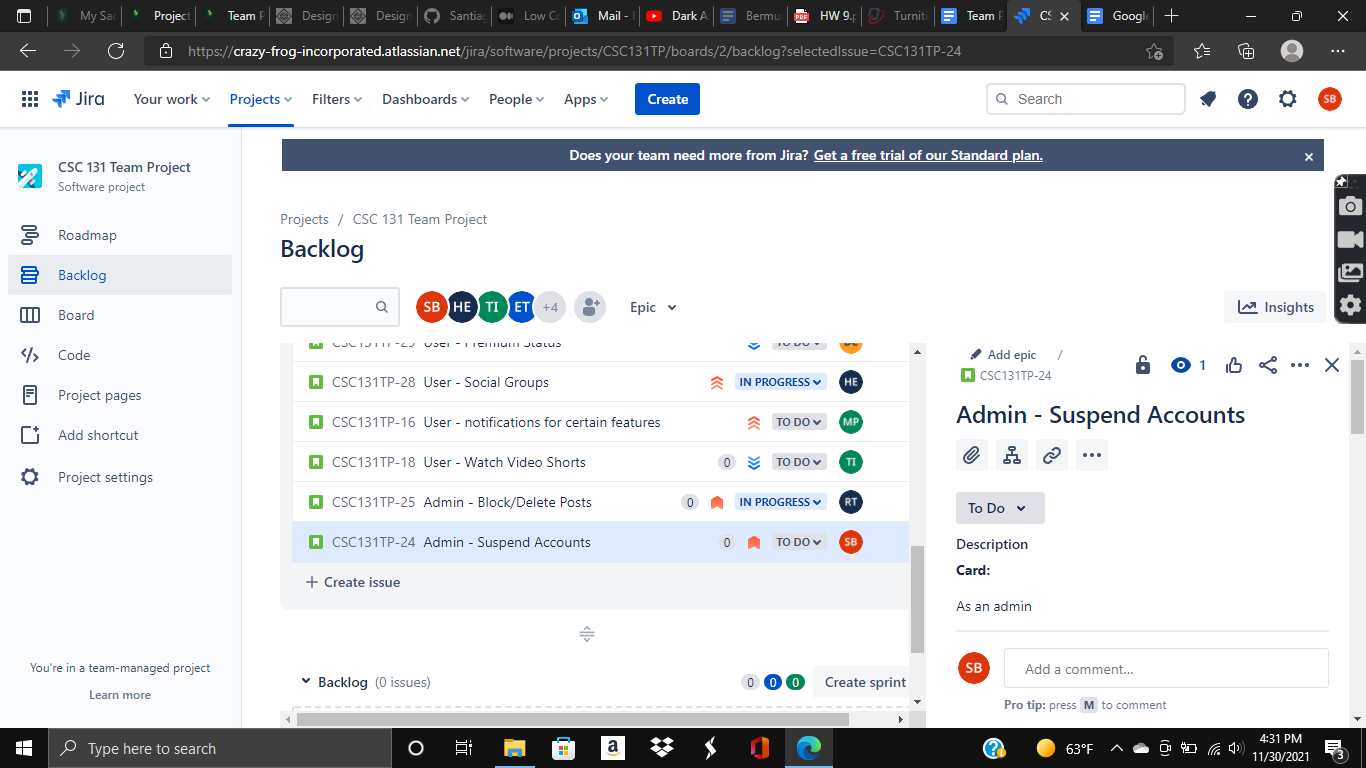


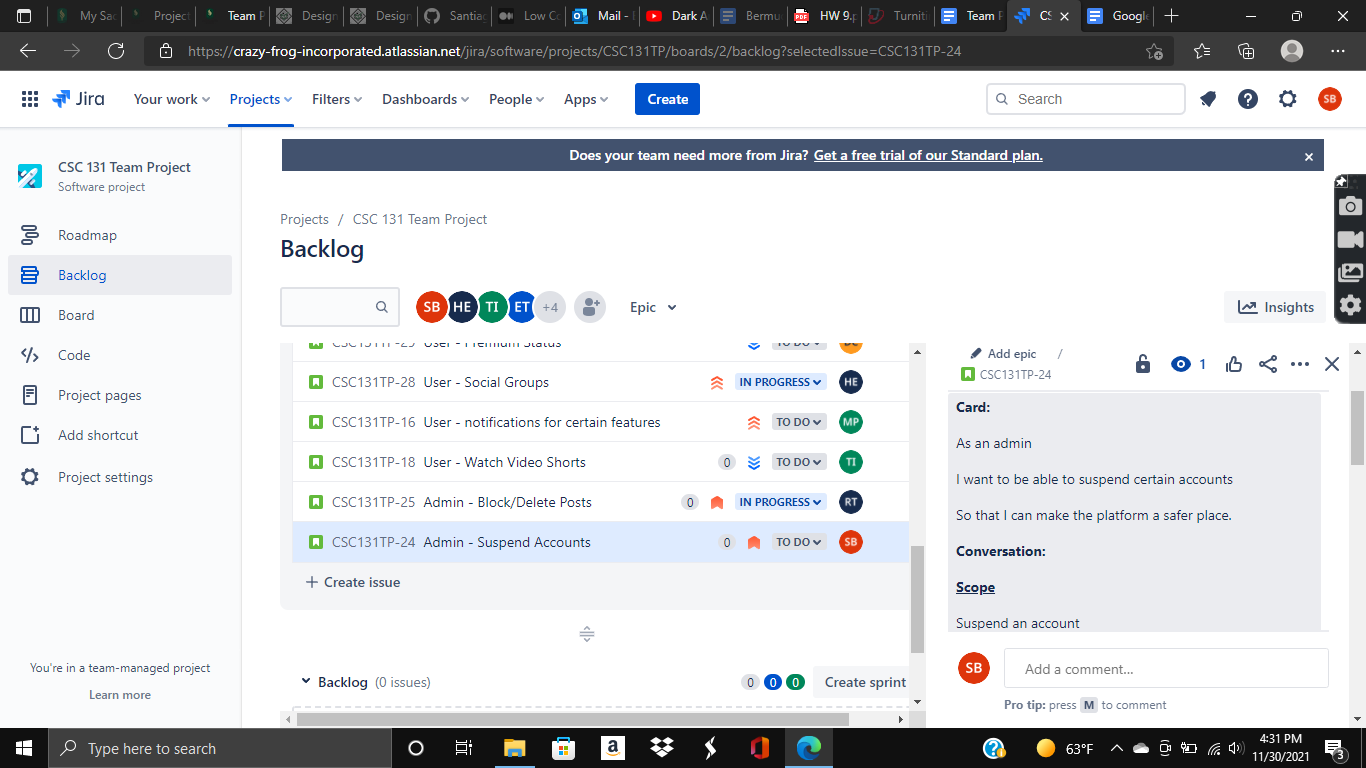


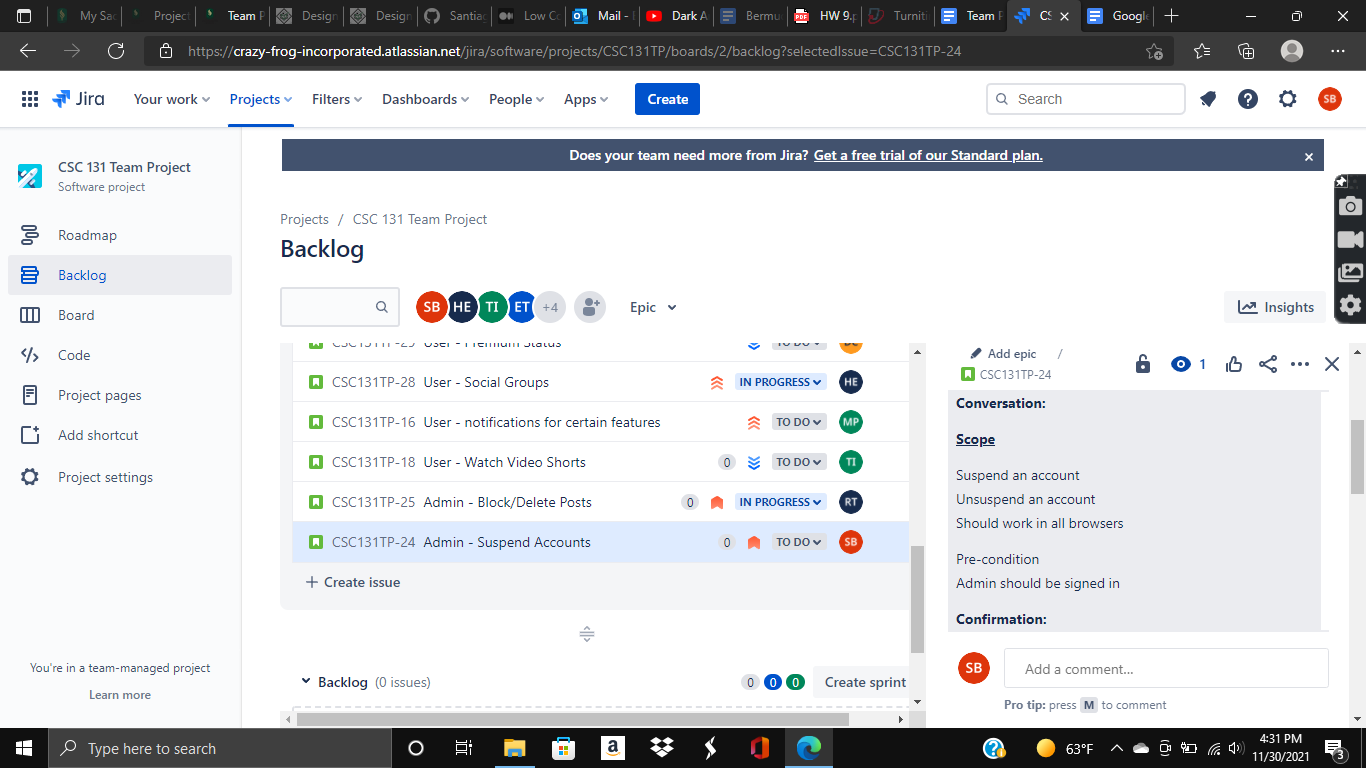


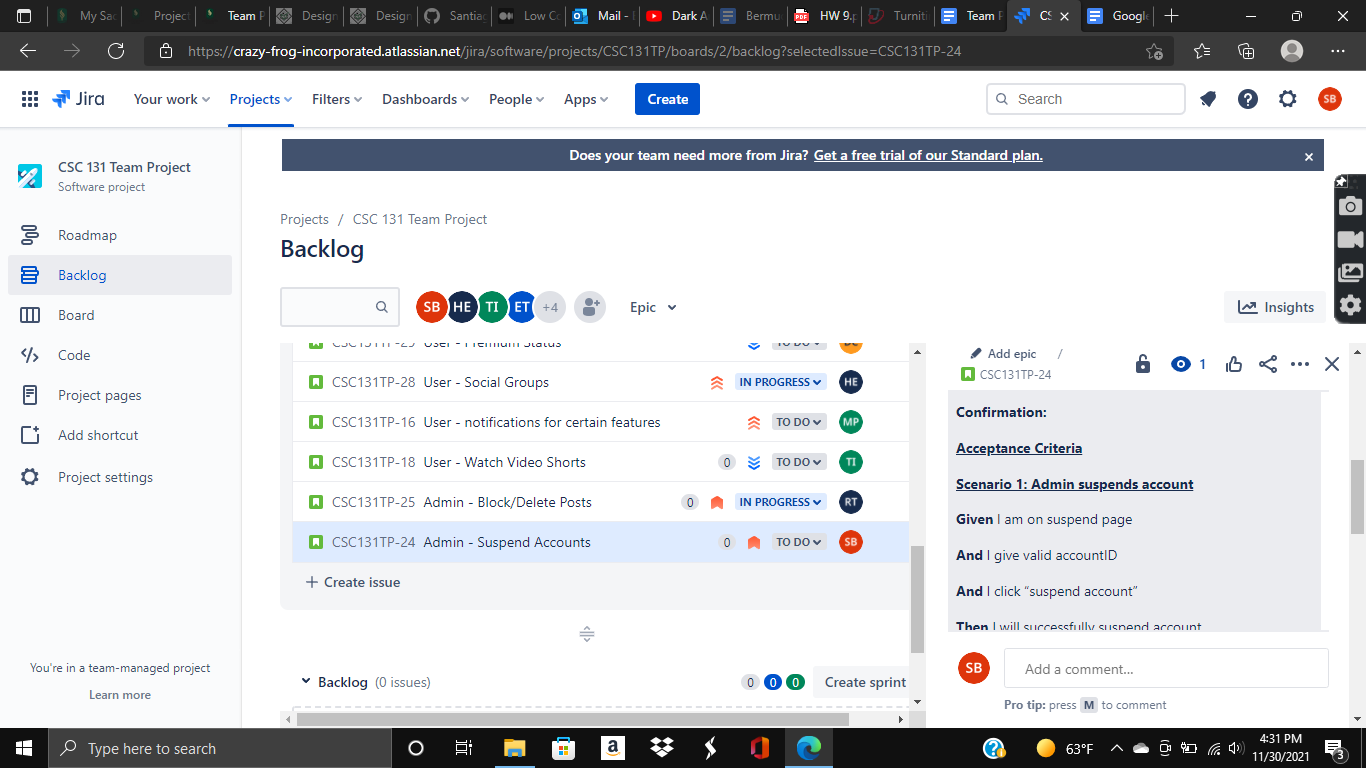


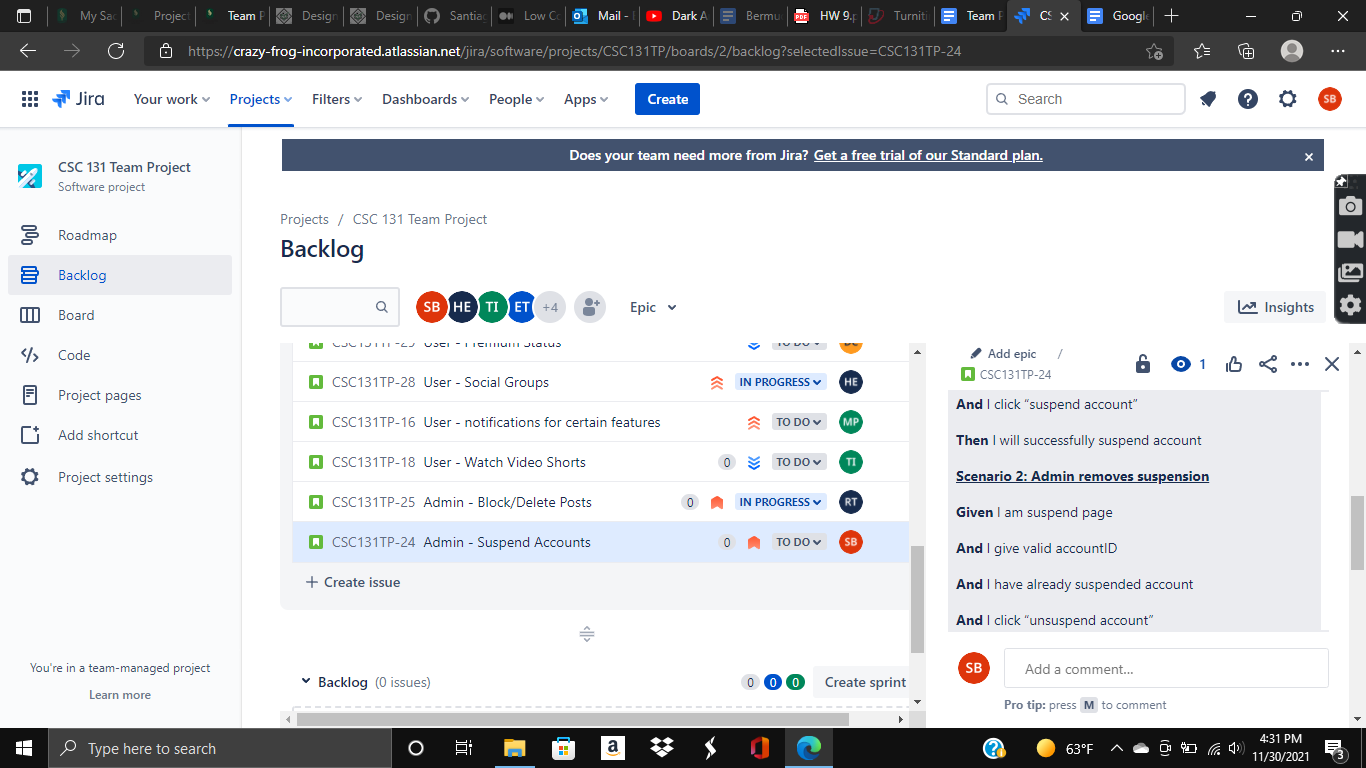
Example 2:

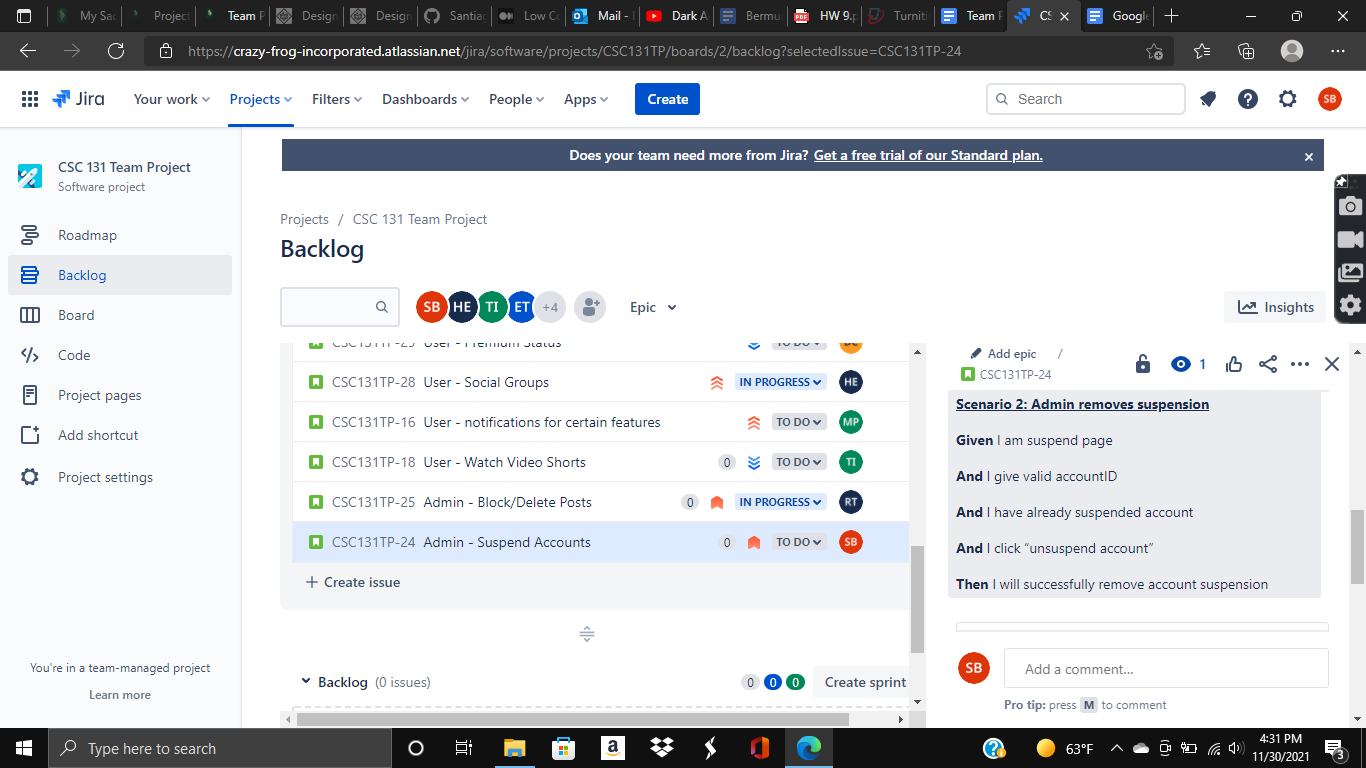




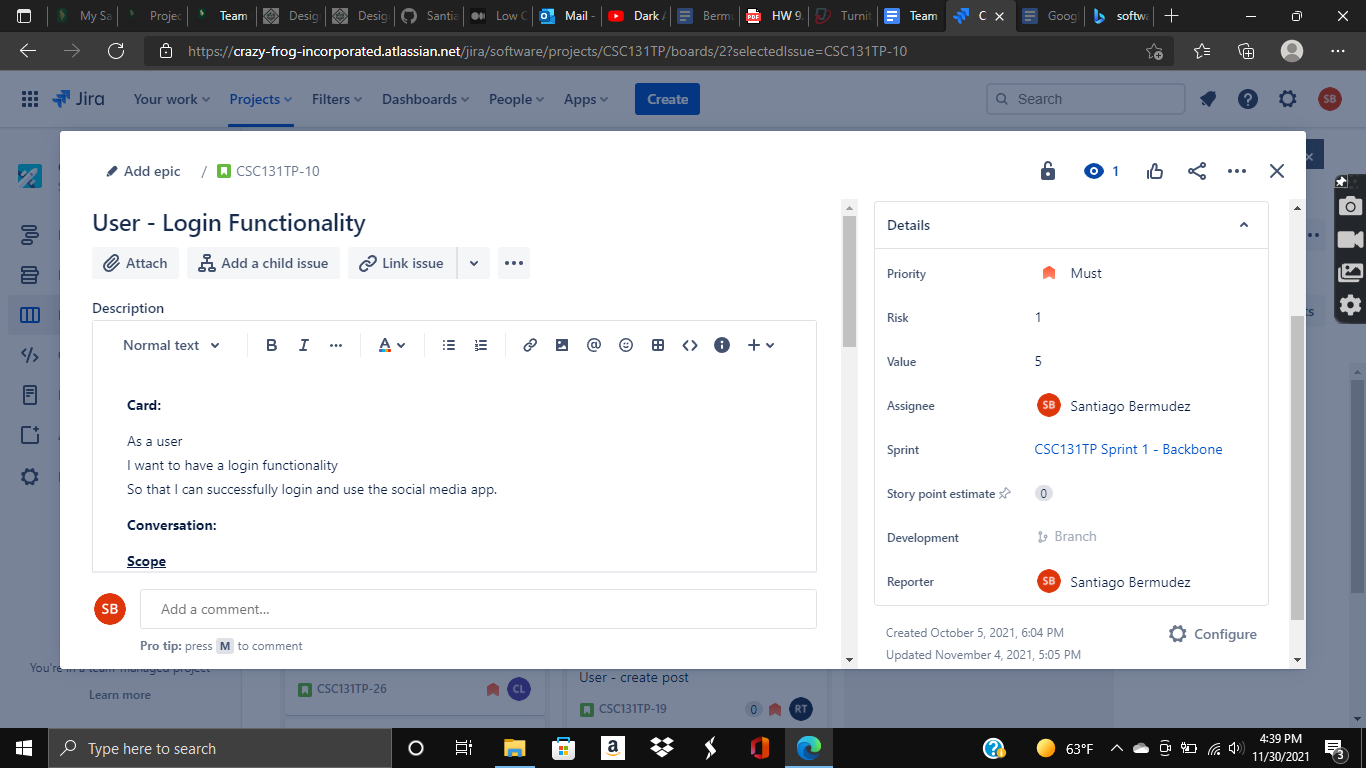


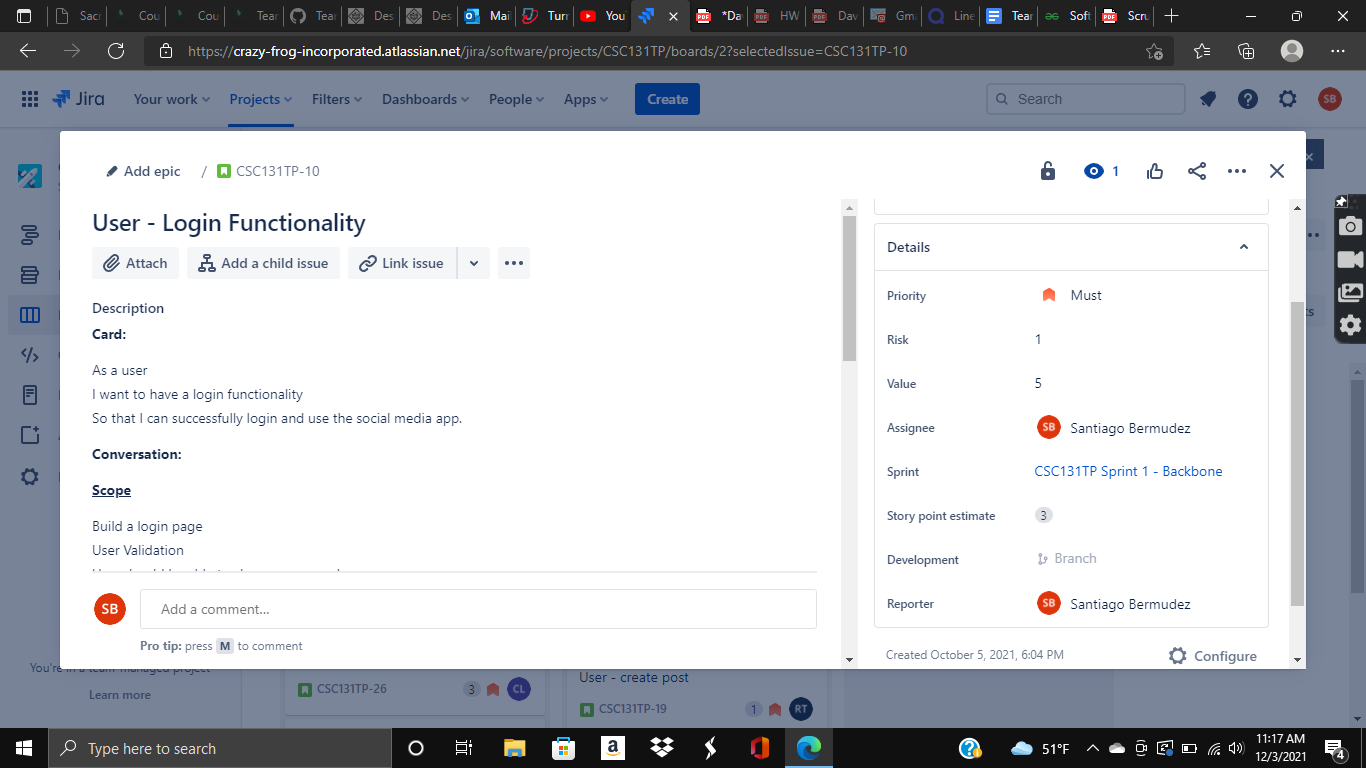






2(b) story points, prioritization and task assignment using tool (2\*3 = 6 points)



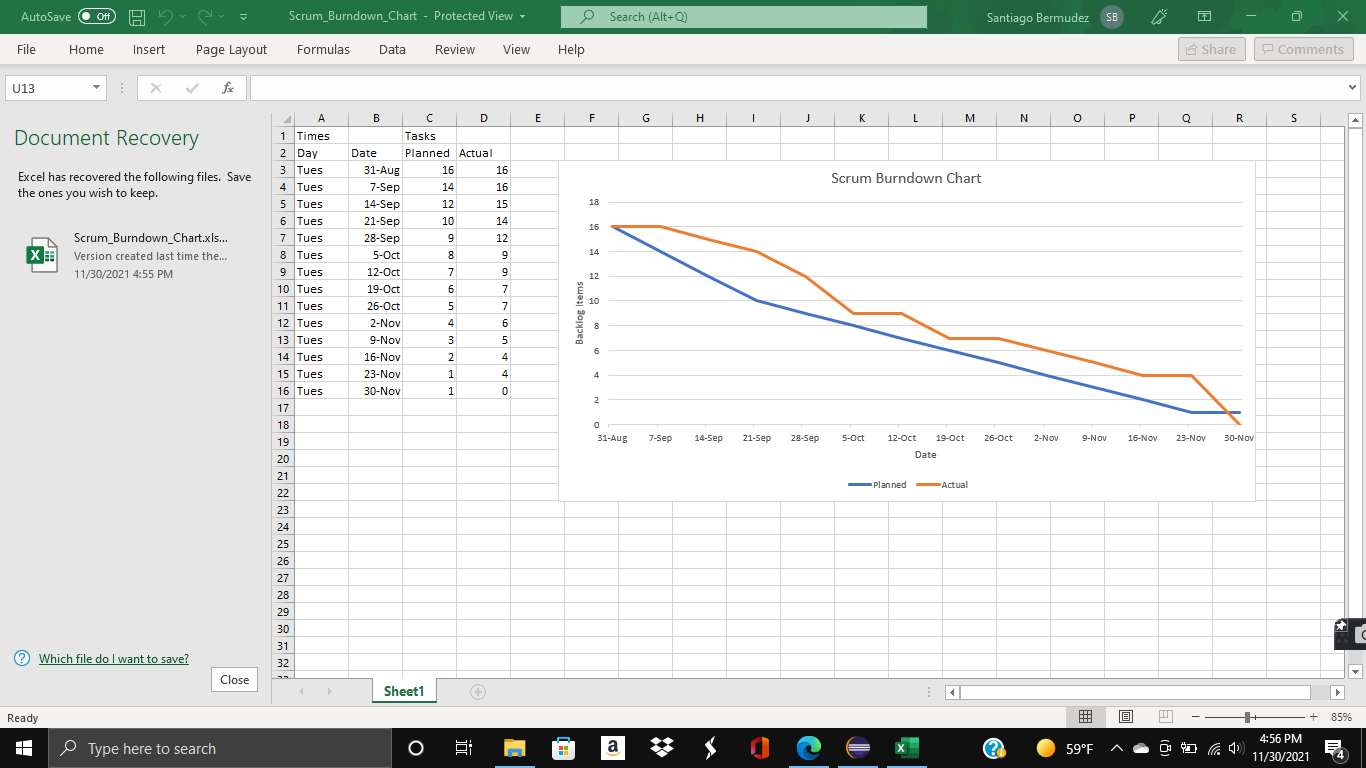


\*In the details on the right, you can see our prioritization, task assignment, and story points. You can also see our risk and value estimates. You should see the same for the rest of our user stories in Jira.

2(c) risk table on People, Process and Product - provide examples of minimum two risks for each of People, Process and Product pertaining to the problem and your risk management solution in same or separate PDF document or if possible in tool (2\*3 = 6 points)

| **Risk Type** | **People** | **Process** | **Product** |
| --- | --- | --- | --- |
| **Predictable** | 1. User’sdon’t enjoy the site.  2. People on the development team arguing over disagreements.  3. Difficulty switching users to our website. (ex. Preferring facebook) | 1. Having trouble delivering on-time.  2. Stakeholders suggesting information that will affect the site negatively. | 1. Functionality within the website failing such as posts not uploading, unable to login, profile not updating, etc.  2. Potential security issues if we don’t add features like 2FA. |
| **Unpredictable** | 1. Someone on the development team getting sick, slowing down the product creation.  2. People banned from accessing site in other countries.  3. User’s accounts hacked | 1. Unable to implement a feature in the backlog.  2. Financial costs start to creep up. | 1. Poor user experience (slow loading app)      1. Website server crashing. |

2(d) and burndown chart (4 points)



**3. Coding using GitHub with:**

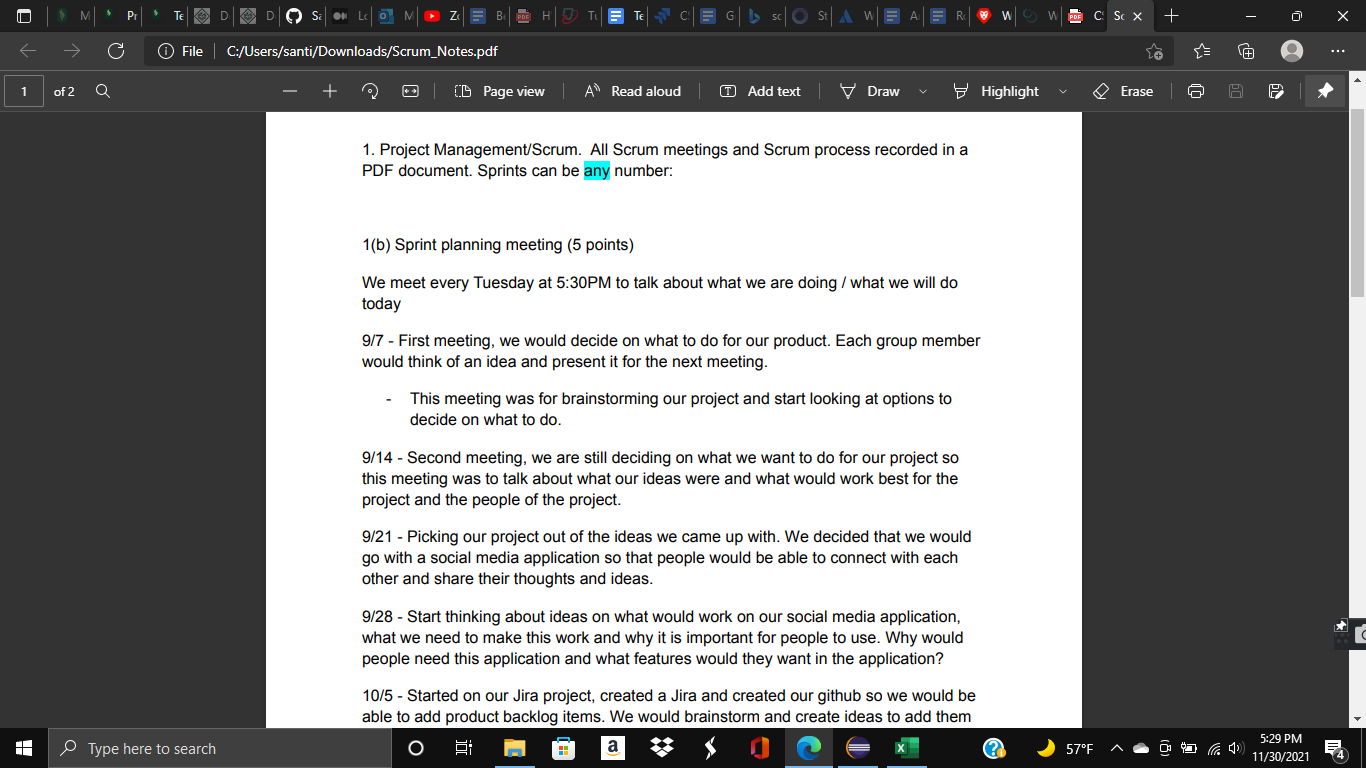
3(a) proof of collaboration (5 points) [ GitHub or GitLab collaboration details and Scrum meeting notes).

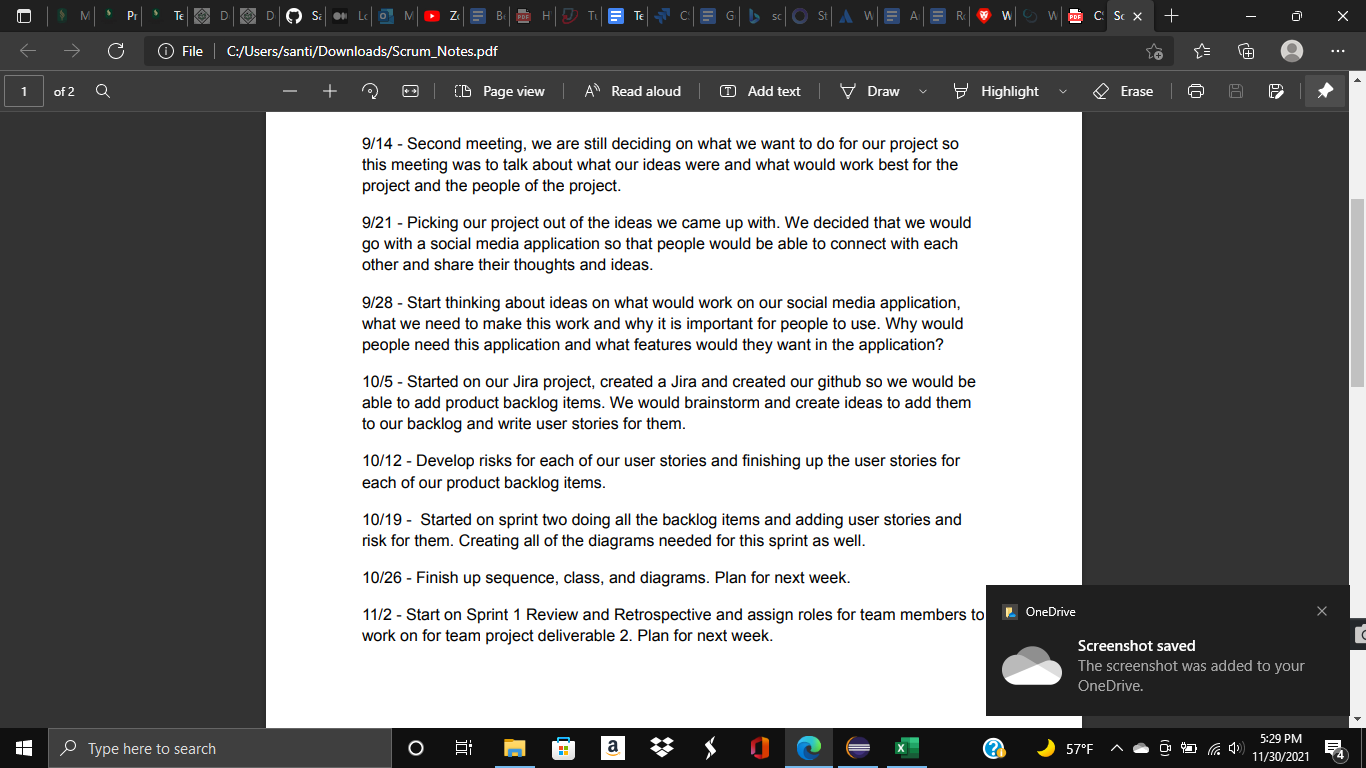
<https://github.com/Santiago13225/TeamProjectCrazyFrogSocial>

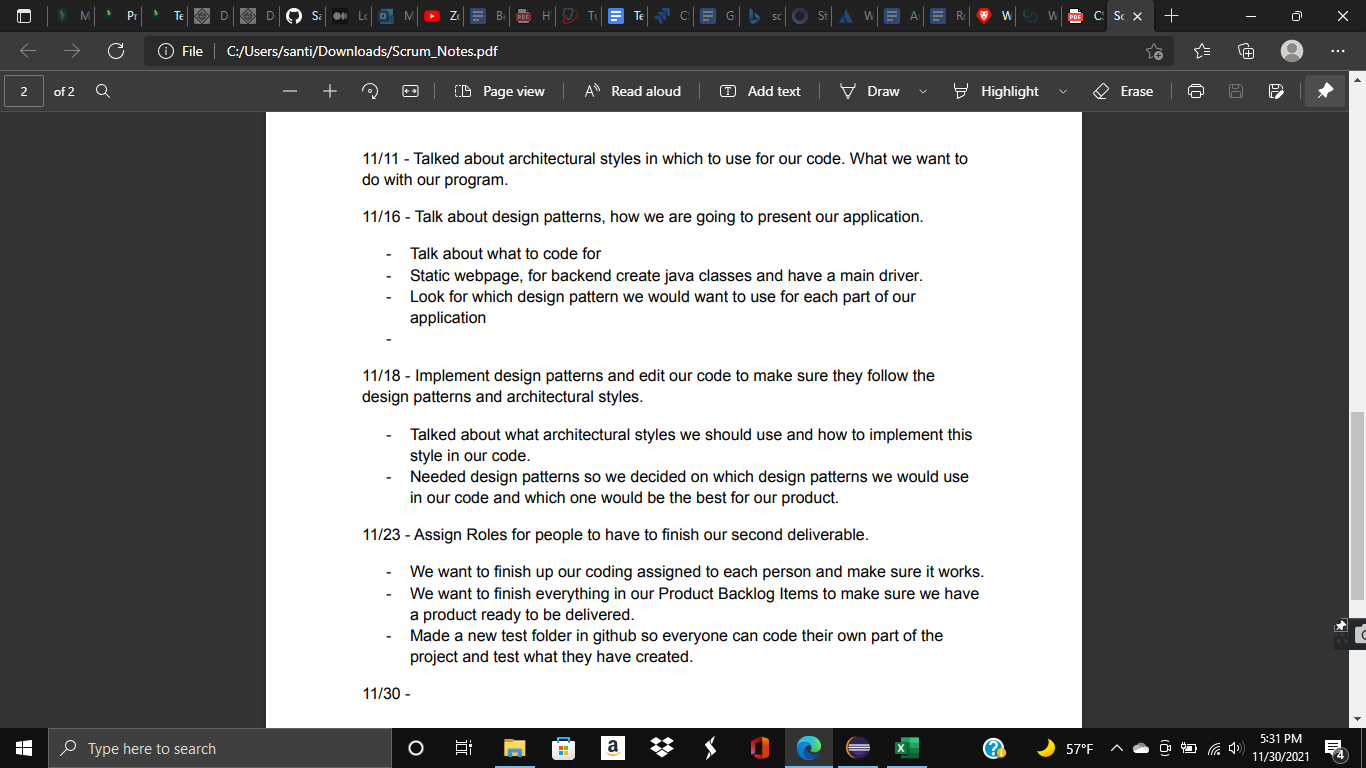
\*Everyone should have access. You were all sent an invite by your Sac State email to be collaborators just so you can see our project. If you can’t access anything, contact:

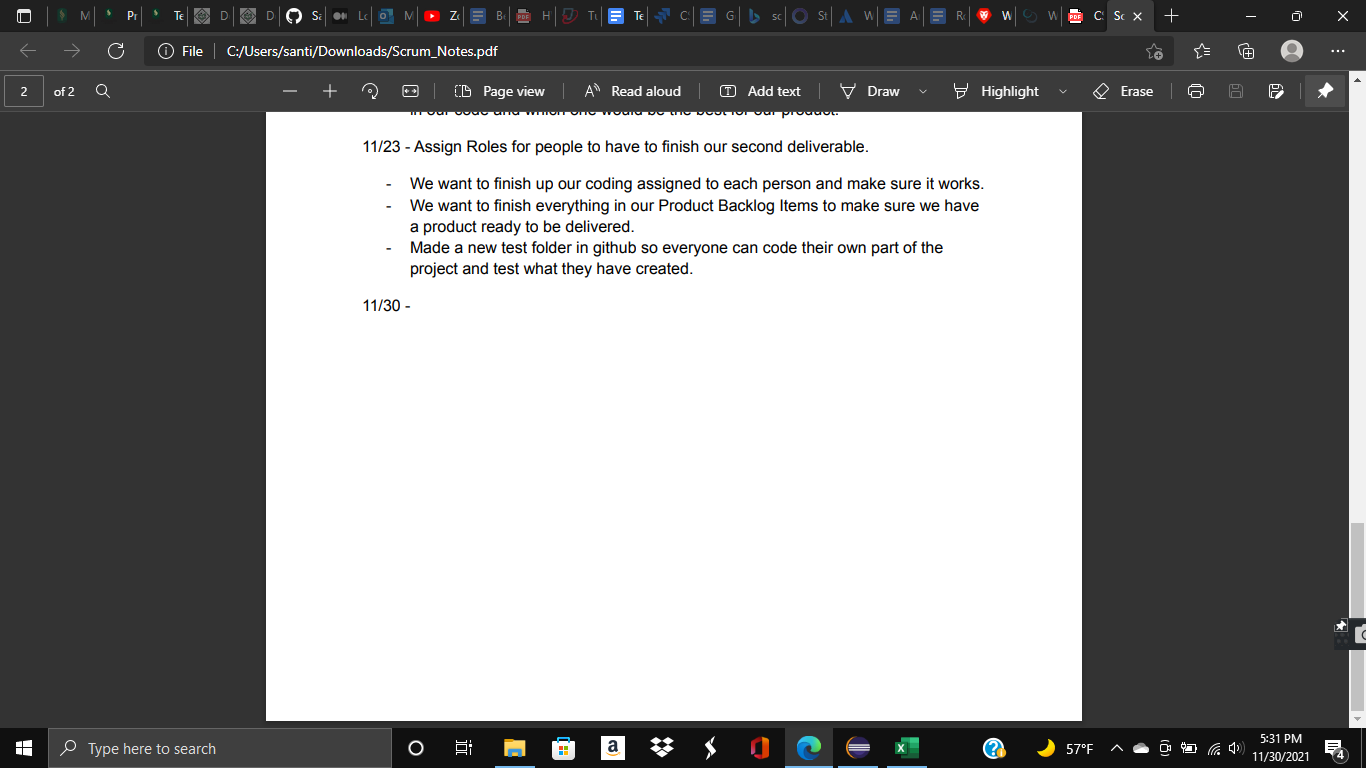
santiagoabermudez@csus.edu

\*Scrum meeting notes below:









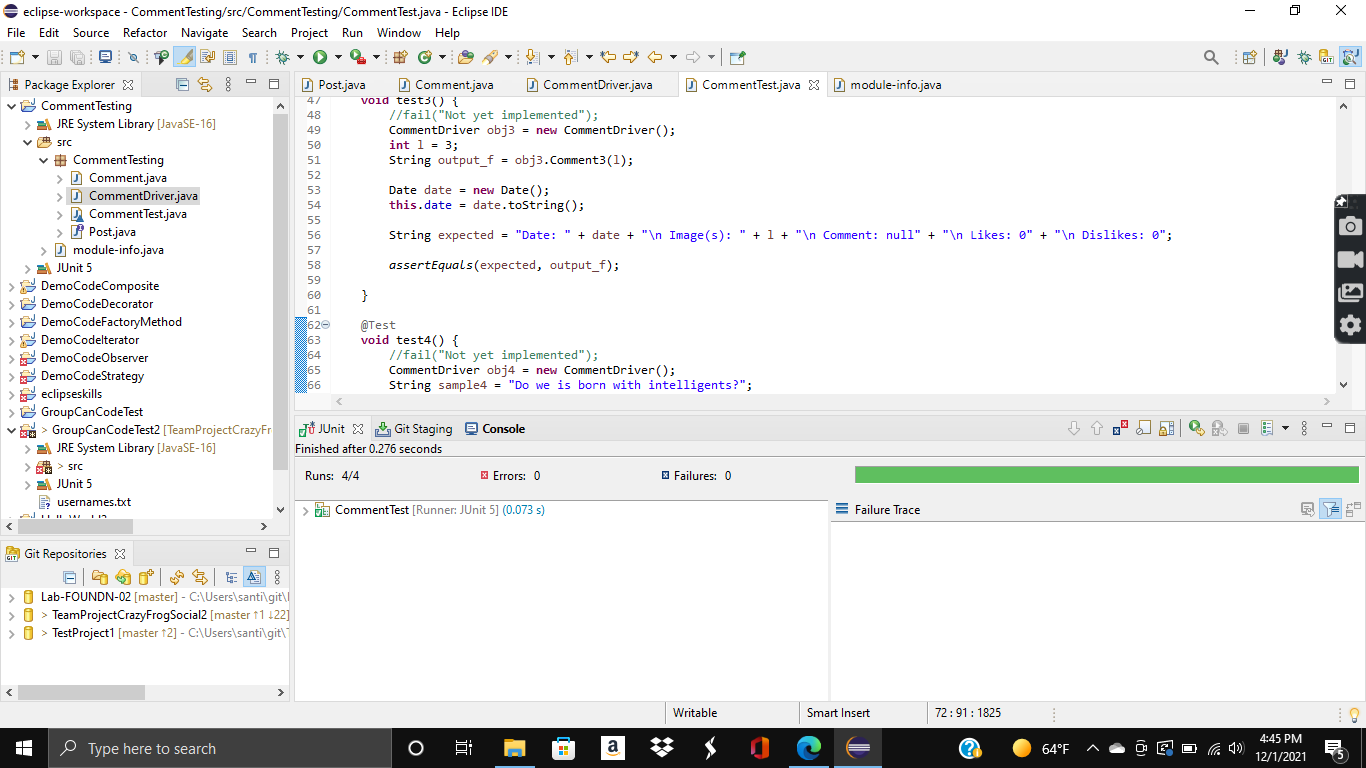
3(b) written description of High Cohesion and Low Coupling principles for your design (5 points)

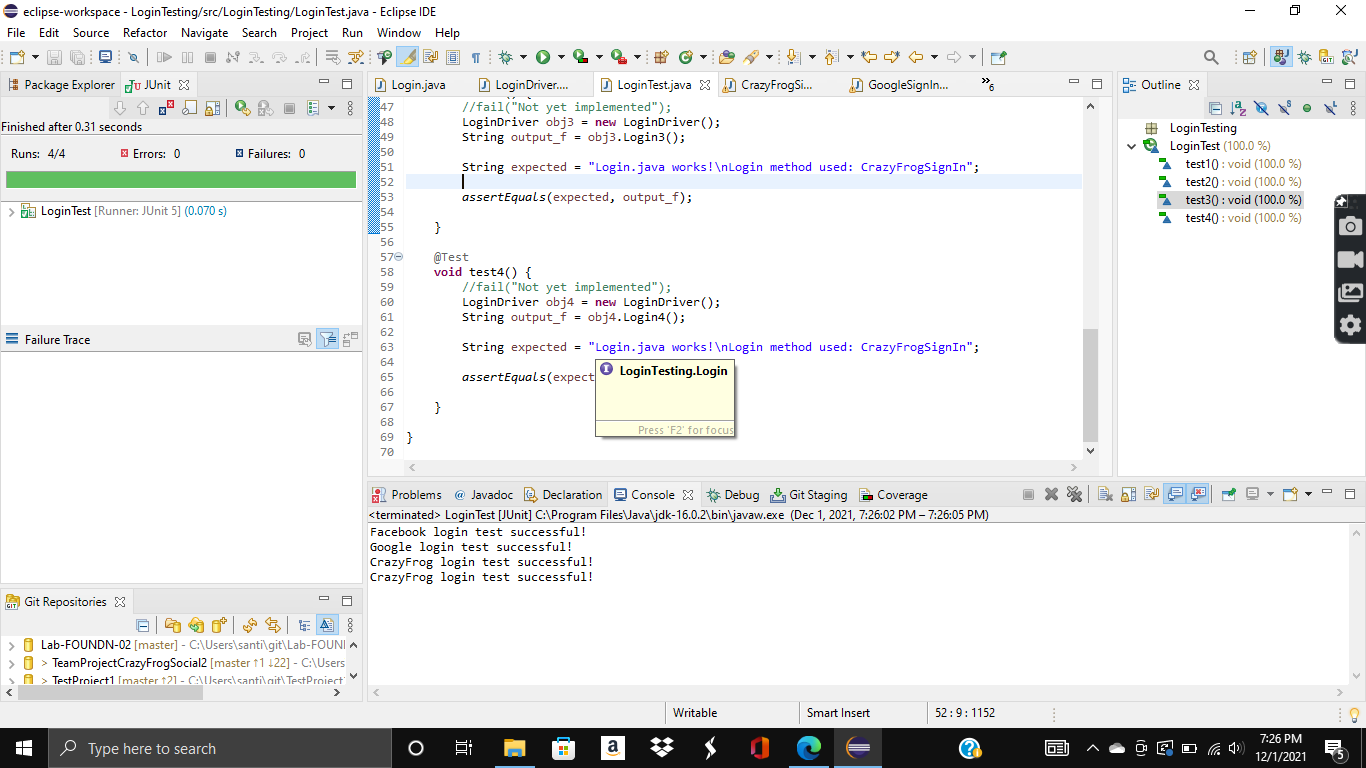
Low Coupling: For low coupling, we have classes of code that are split into certain groups related to their purpose like, a separate group for logging and a separate group of code for commenting and posts. For low coupling between these two modules, we made it so that our code uses data coupling as opposed to content coupling in that the only way the modules communicate is by passing data rather than having one module modify the data of another module.

High Cohesion: For high cohesion, we did our best to make it so that the classes in our modules are functionally related. Our code is somewhere in between the levels of sequential and communicational cohesion. This is because some of our classes might have output that could become the input for other classes and sometimes there might be classes that operate on the same input data or contribute towards the same output data.

3 (c) using JUnit (5 points) (100% tests are green)

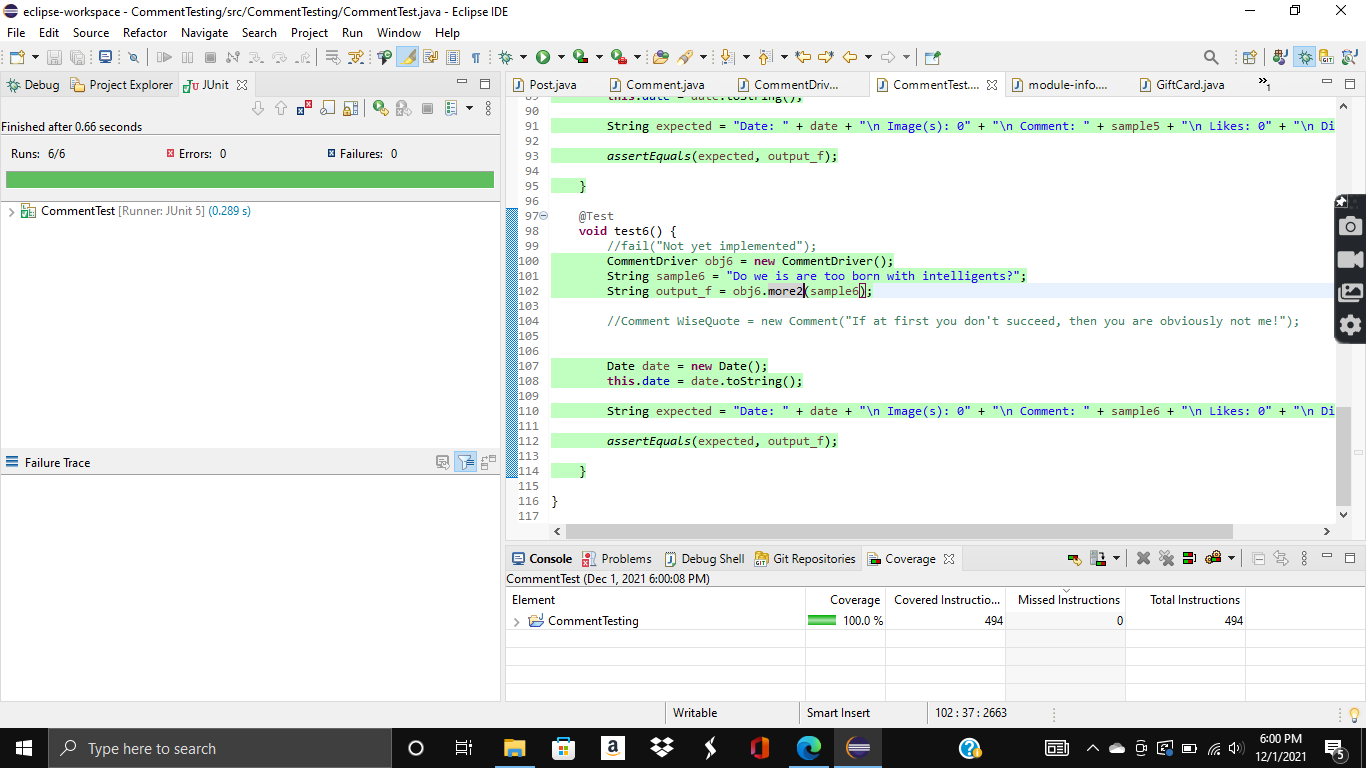
\*We had a lot of code, so we had 2 groups of 4 people take on half of the code each.

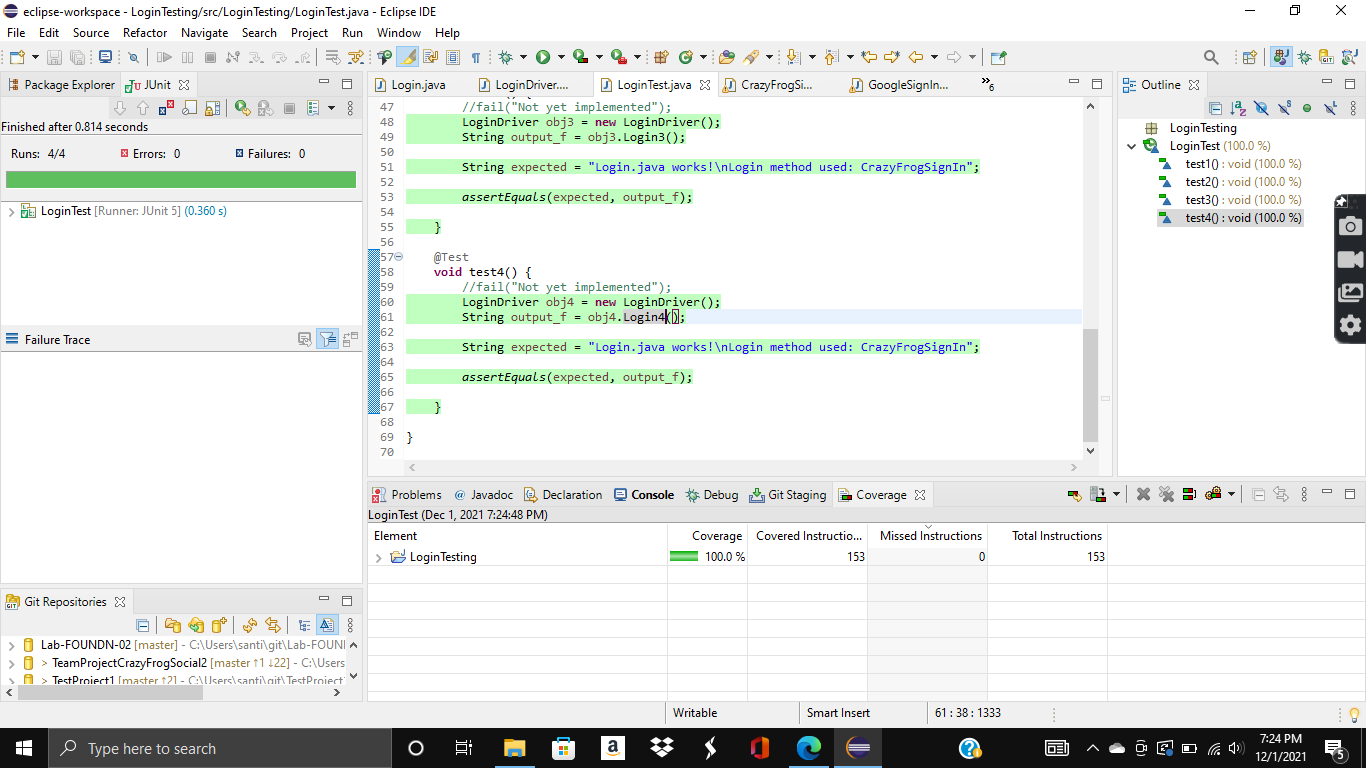




3 (d) using EclEmma (5 points) (100% code coverage)

\*We had a lot of code, so we had 2 groups of 4 people take on half of the code each.

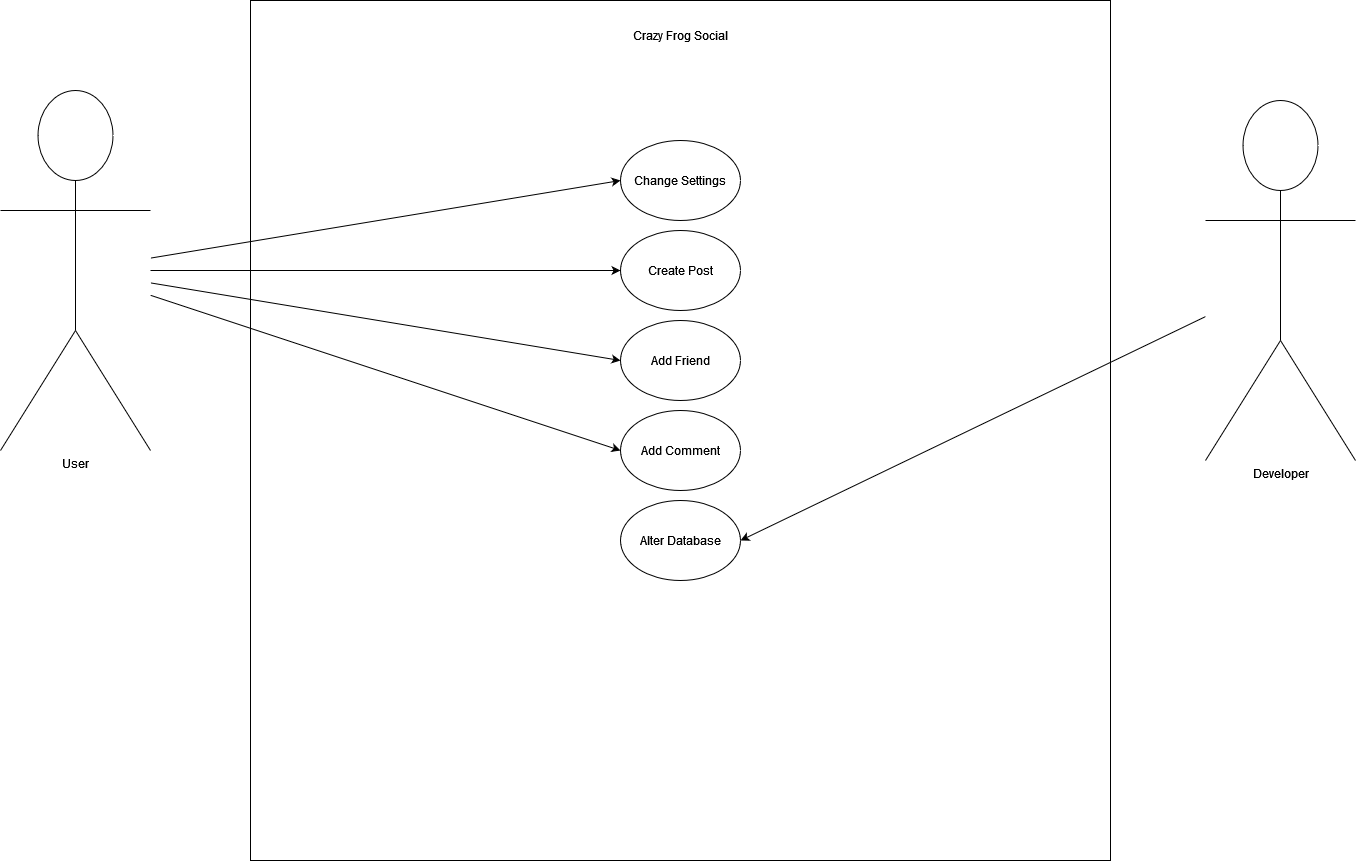




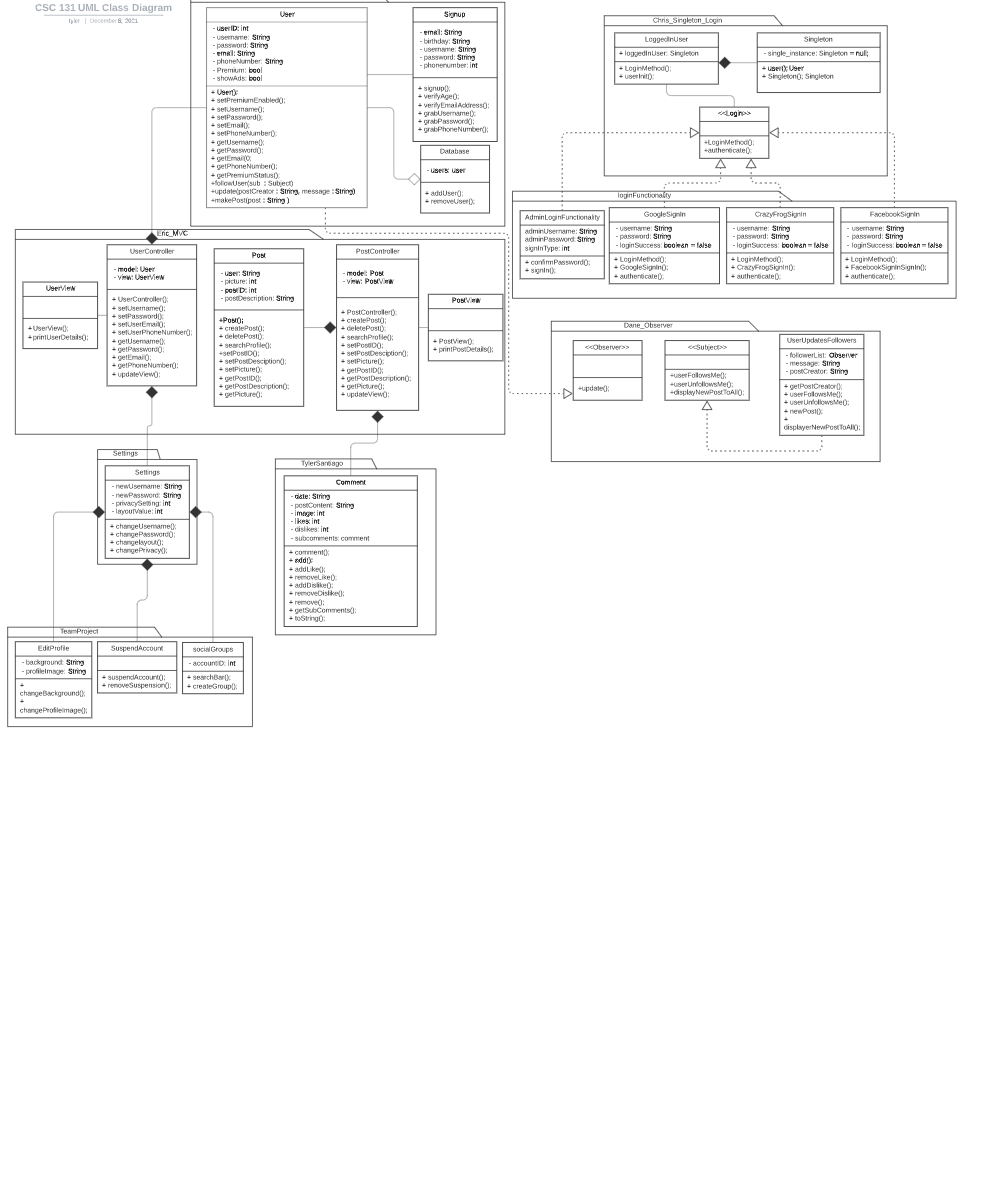
**4. Four types of UML Diagrams:**

Only one diagram each for examples of Use Case, Class, Activity and Sequence for portions of your code) for any part of code. It is not required to have a diagram for all aspects of your application. Please refer to Lucid Chart tutorials on YouTube. Lucid Chart is becoming popular.

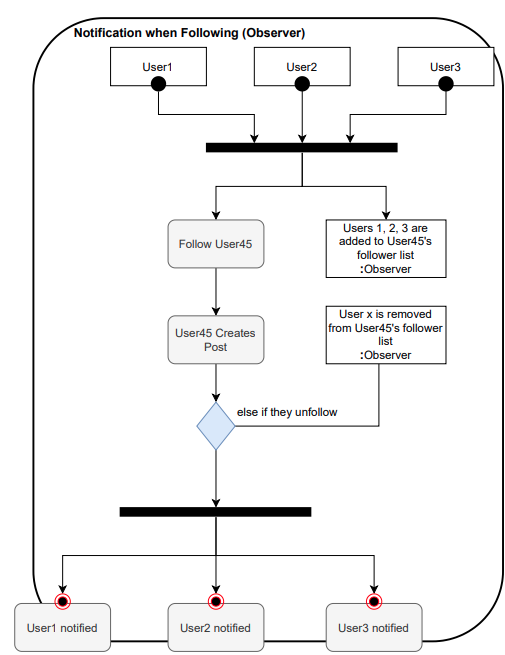
4(a) Use Case Diagram (5 points) (for part of code is ok)



4(b) Class Diagram (5 points) (typically is for all of your design)

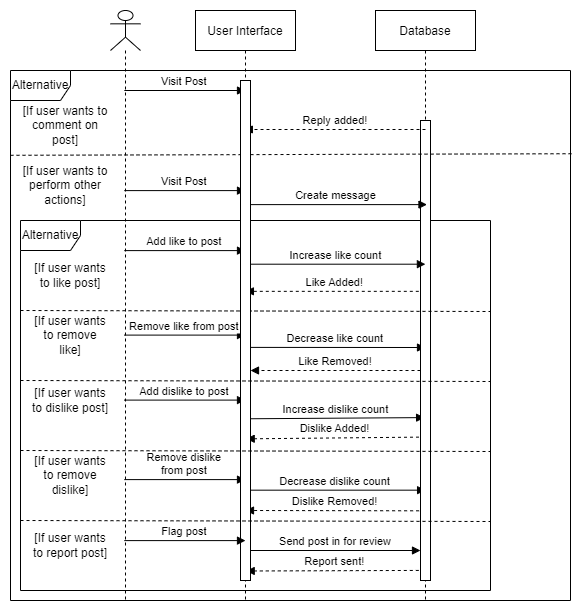


4(c) Activity Diagram (5 points) (for part of code is ok)



4(d) Sequence Diagram (5 points) (for part of code is ok)

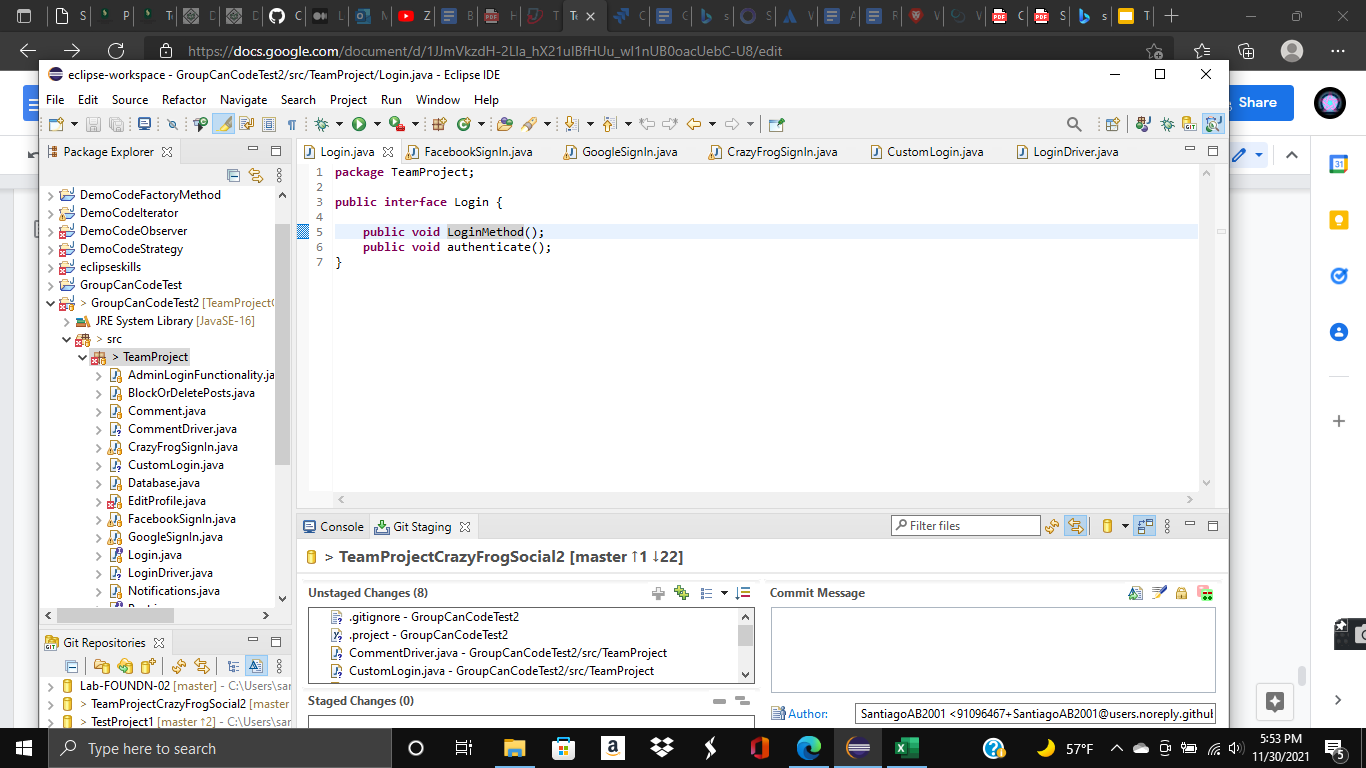
For replying to other posts:

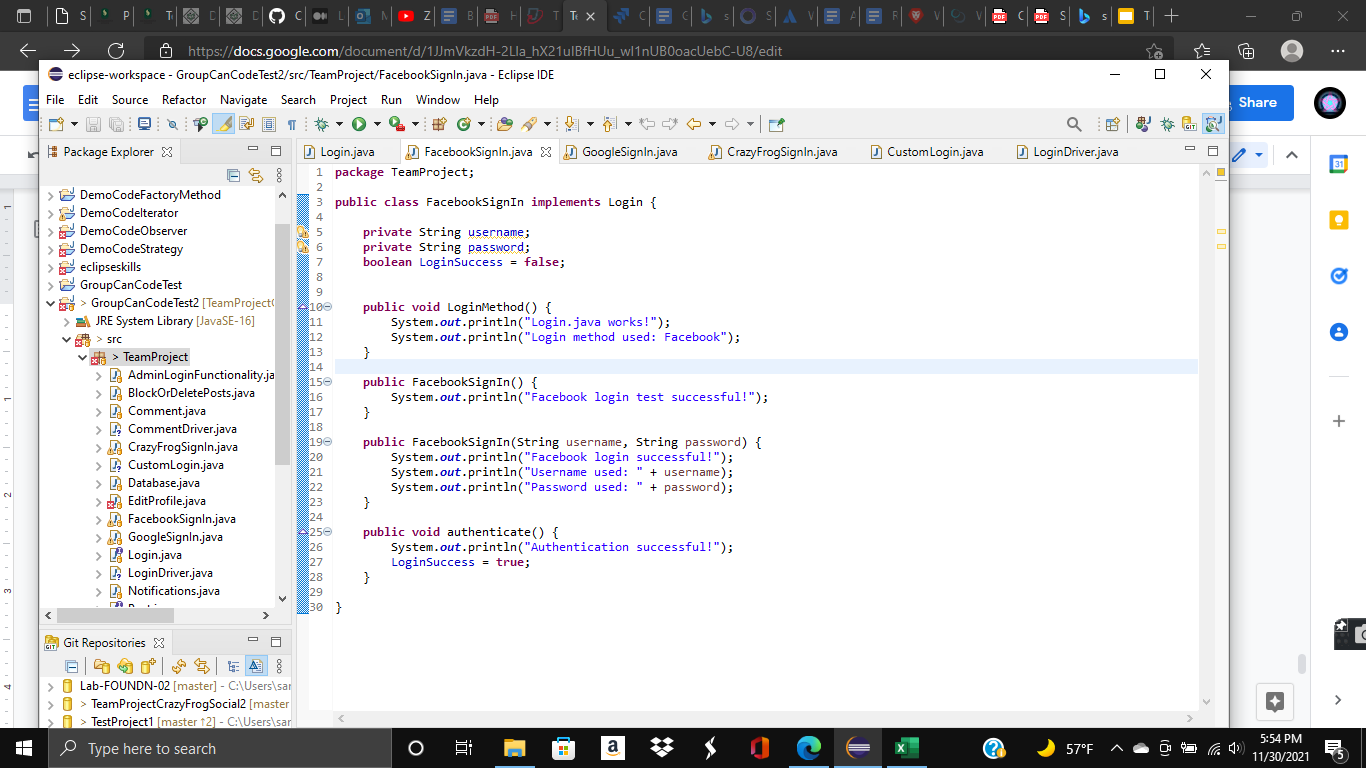


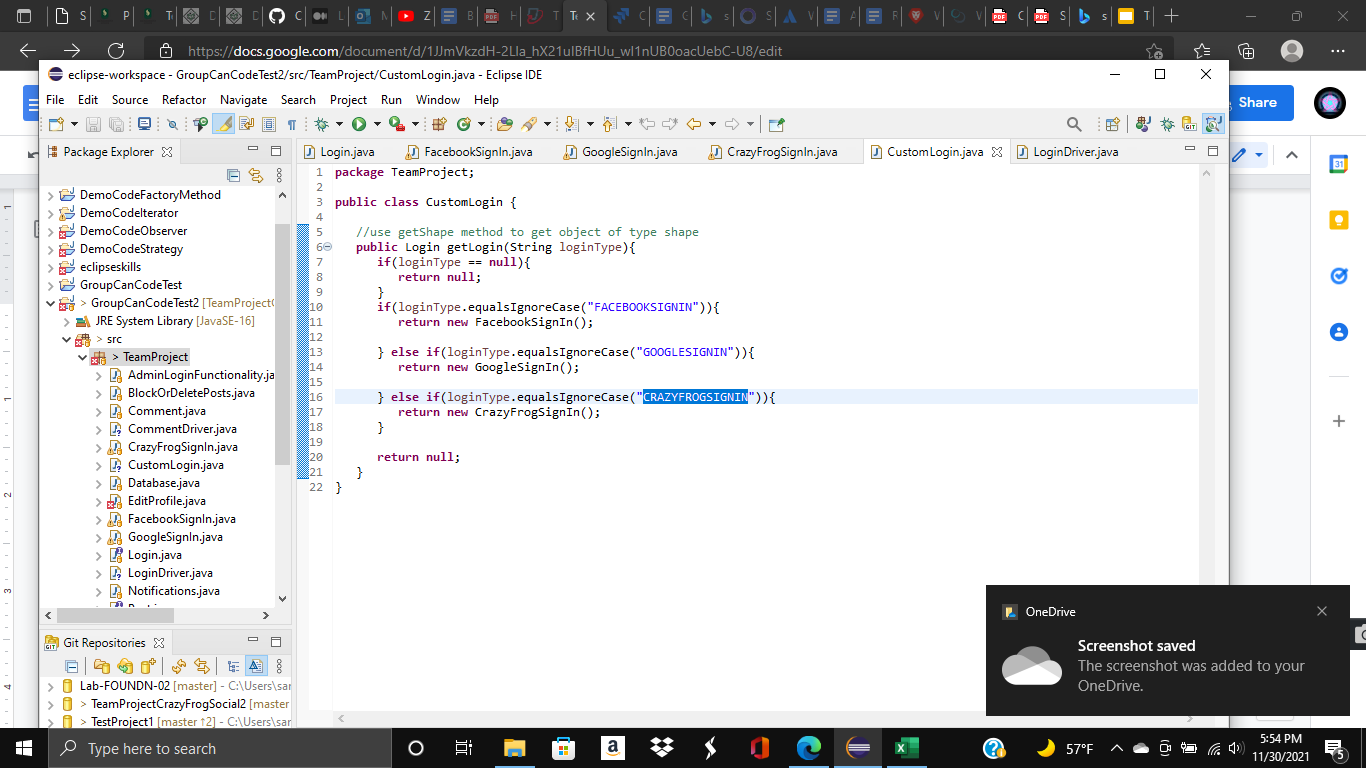
**5. Minimum 2 Design Patterns: (If you are building your project on top of the labs then you cannot claim existing design patterns. You must include your own application of any two design patterns).**

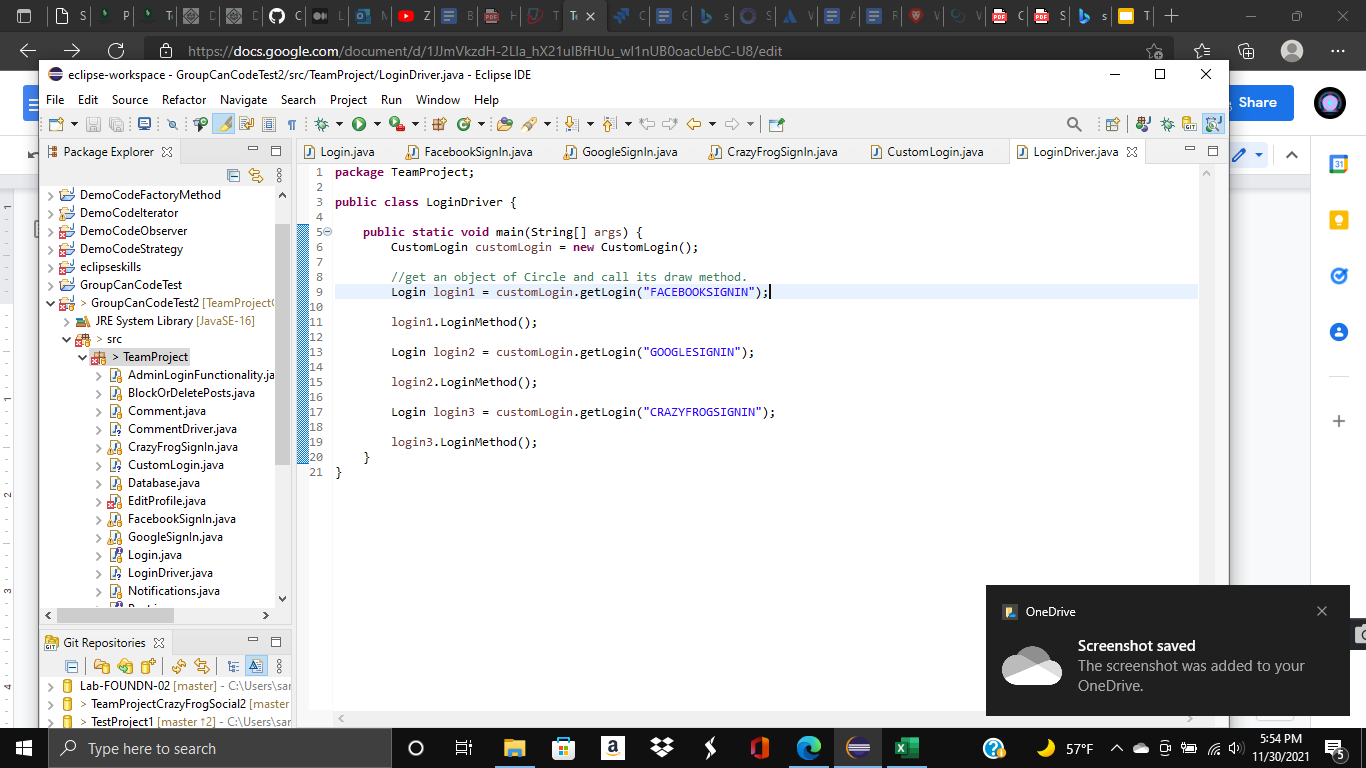
Design Pattern A: (10 points)

**Factory Method: Santiago - Login.Java and other related classes. In Factory pattern, we create an object without exposing the creation logic to the client and refer to the newly created object using a common interface. (Full code on GitHub)**



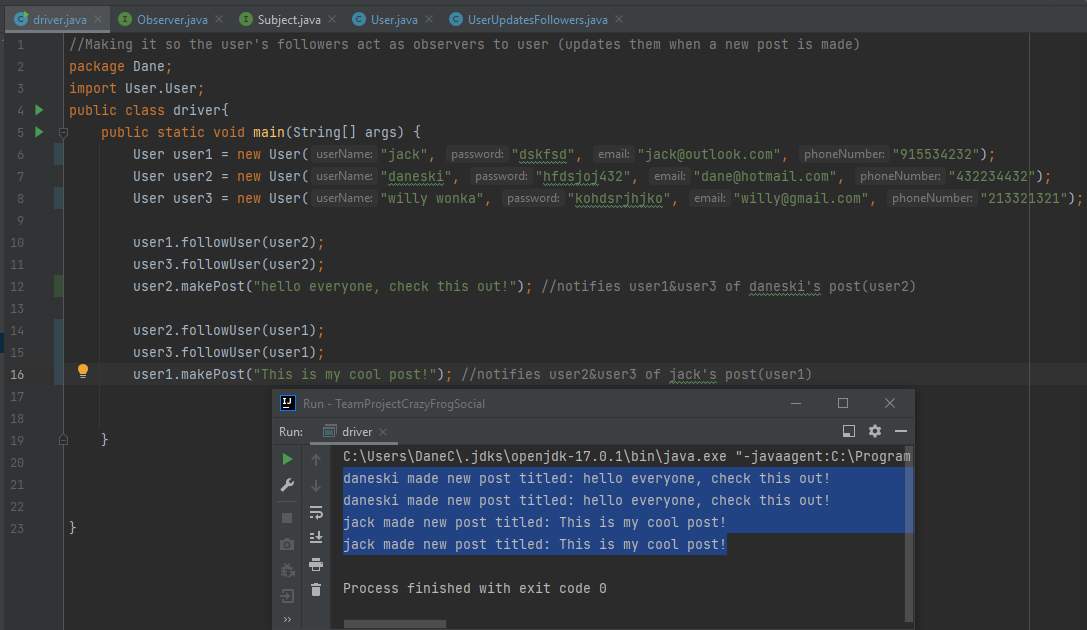






Design Pattern B: (10 points) :

**Observer : Dane -** Making it so the user’s followers act as observers so they are notified when a new post is made. (Full code on github)

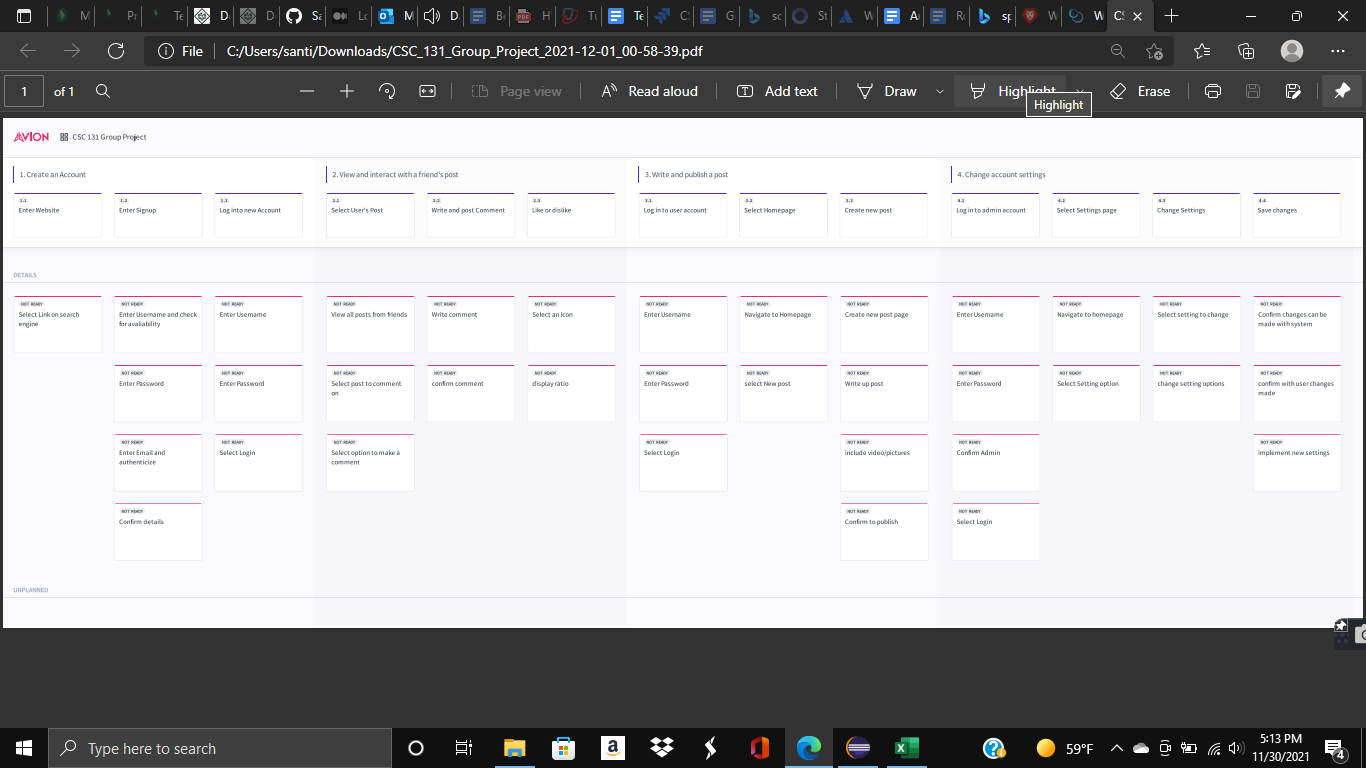


**1. Bonus Points for User Stories: ---> (5 bonus points)**

**(a) Validate User Story with INVEST and MOSCOW principles.**

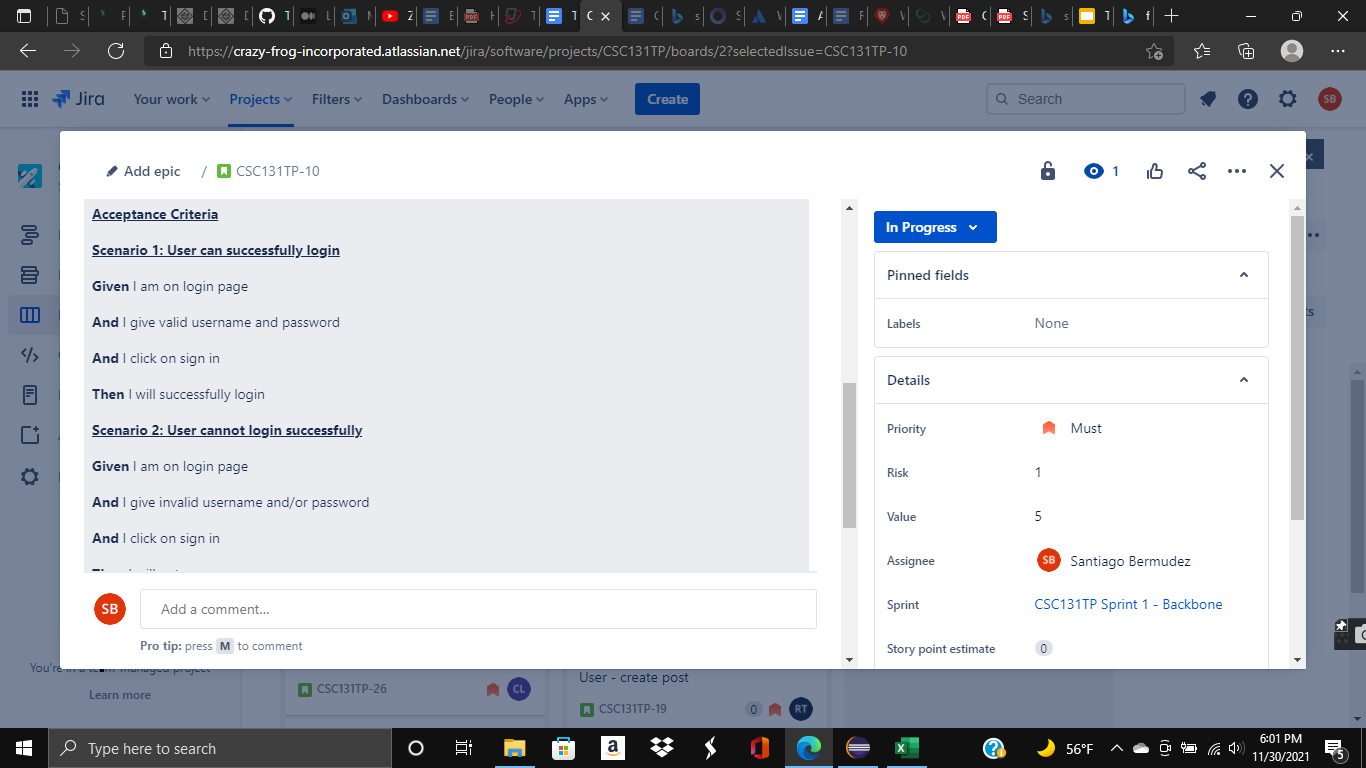
**\*We did MOSCOW principles, but we didn’t do INVEST. We just did what we could. I don’t know if we can get some extra credit for parts B and C.**

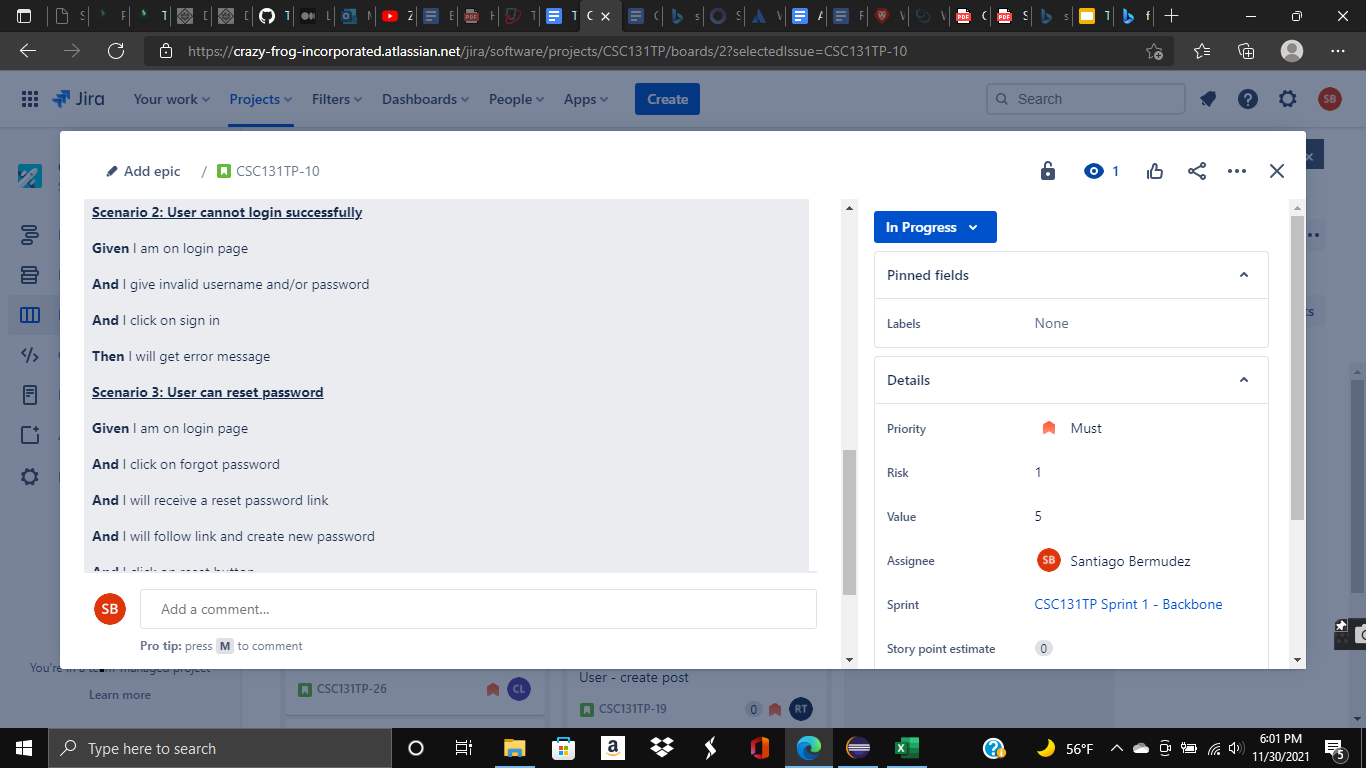
**(b) Provide User Story Map.**

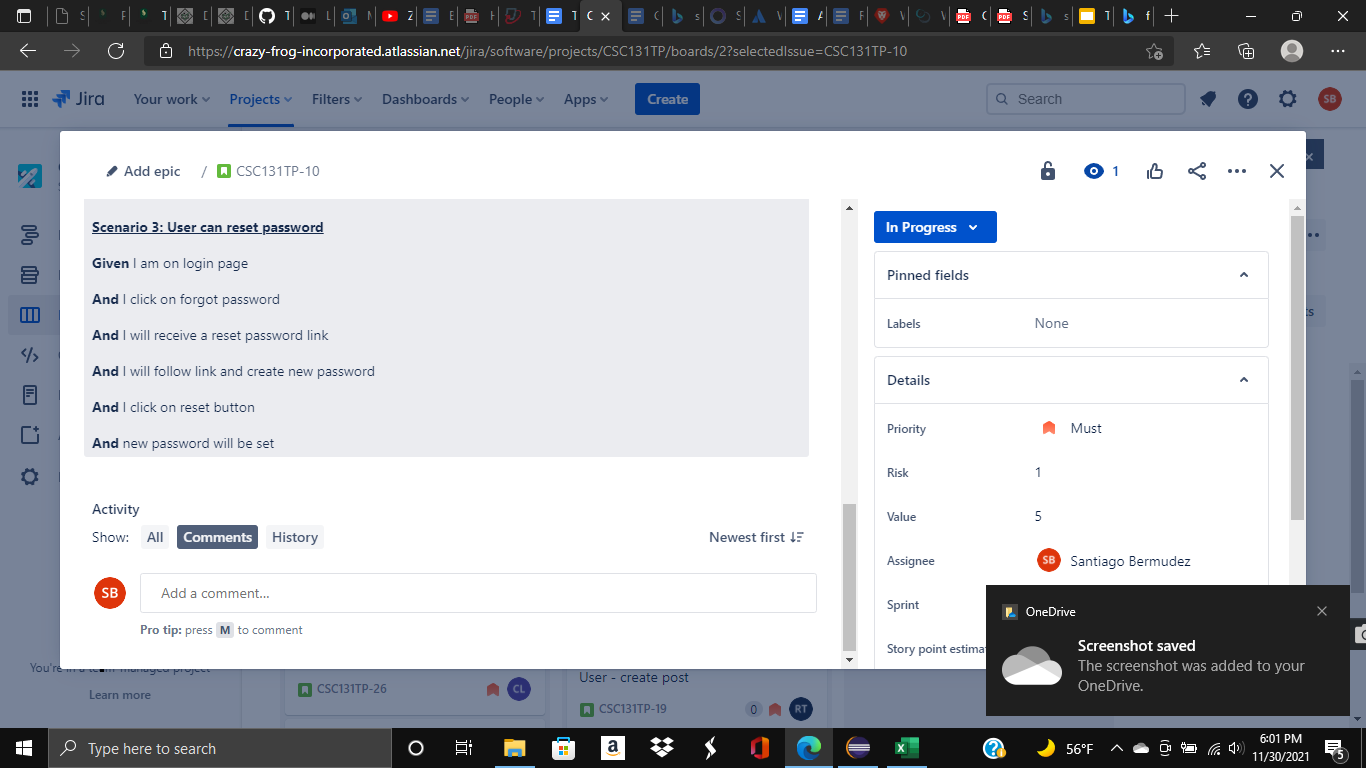
****

**(c) Include Definition of Done with Acceptance Criteria.**

**\***You can see an example of our definition of done with acceptance criteria below, but you should also see the same for the rest of our user stories in Jira.

****

****

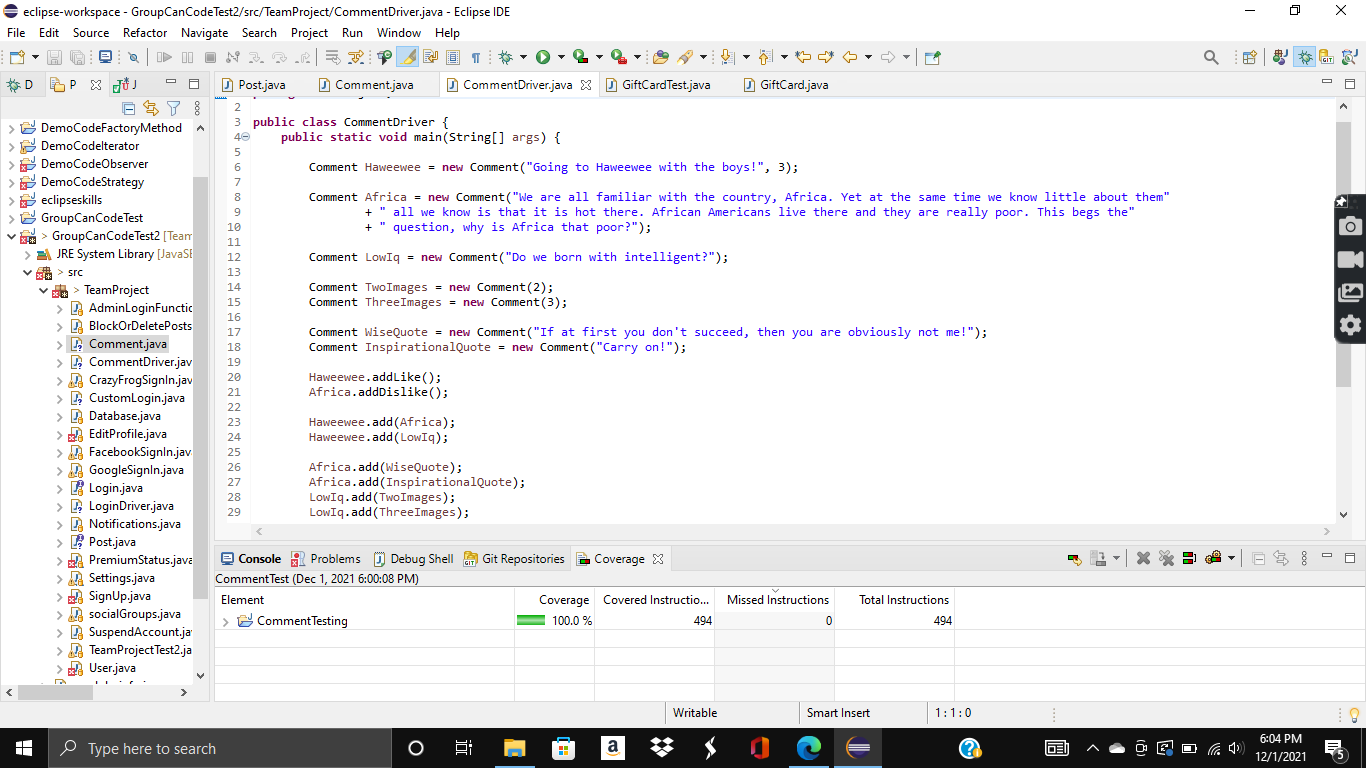
****

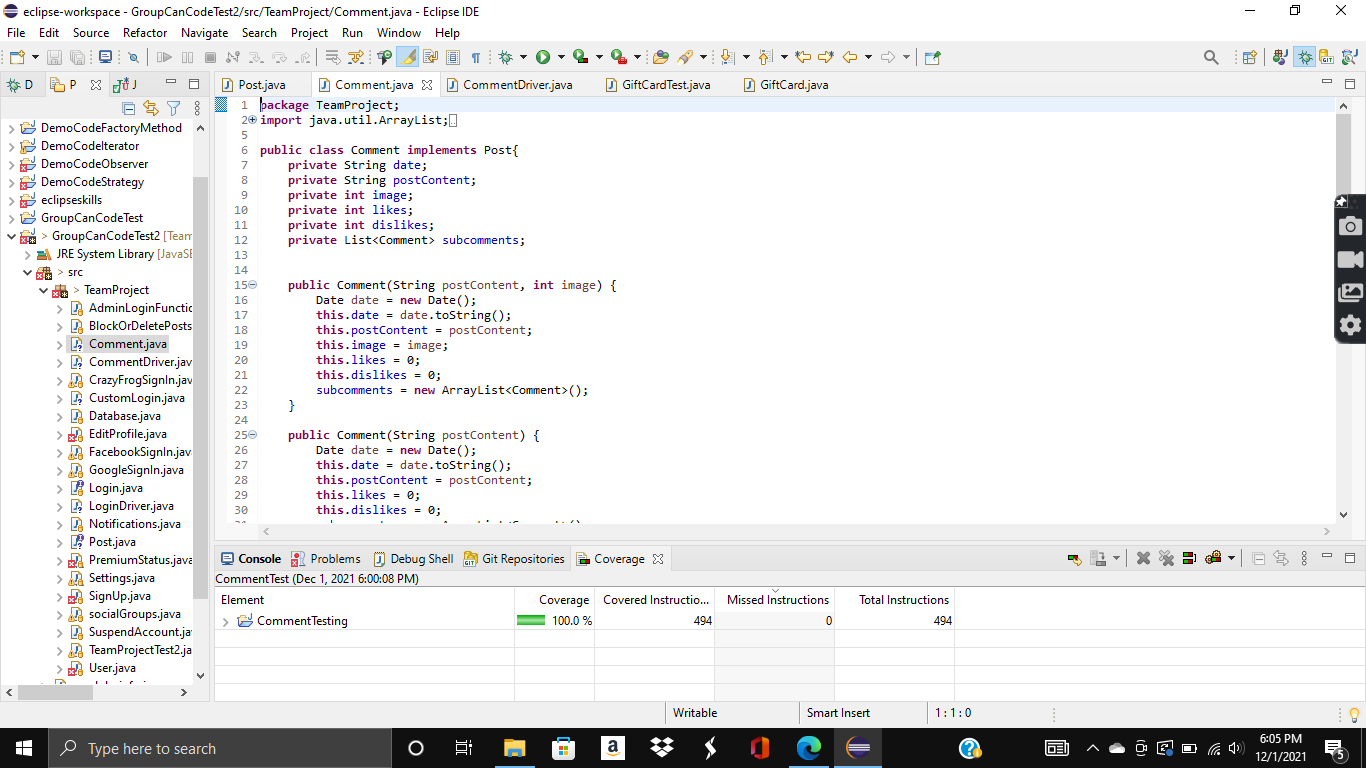
**2. Bonus Points for use of additional 2 Design Patterns in Code relevant to your problem:**

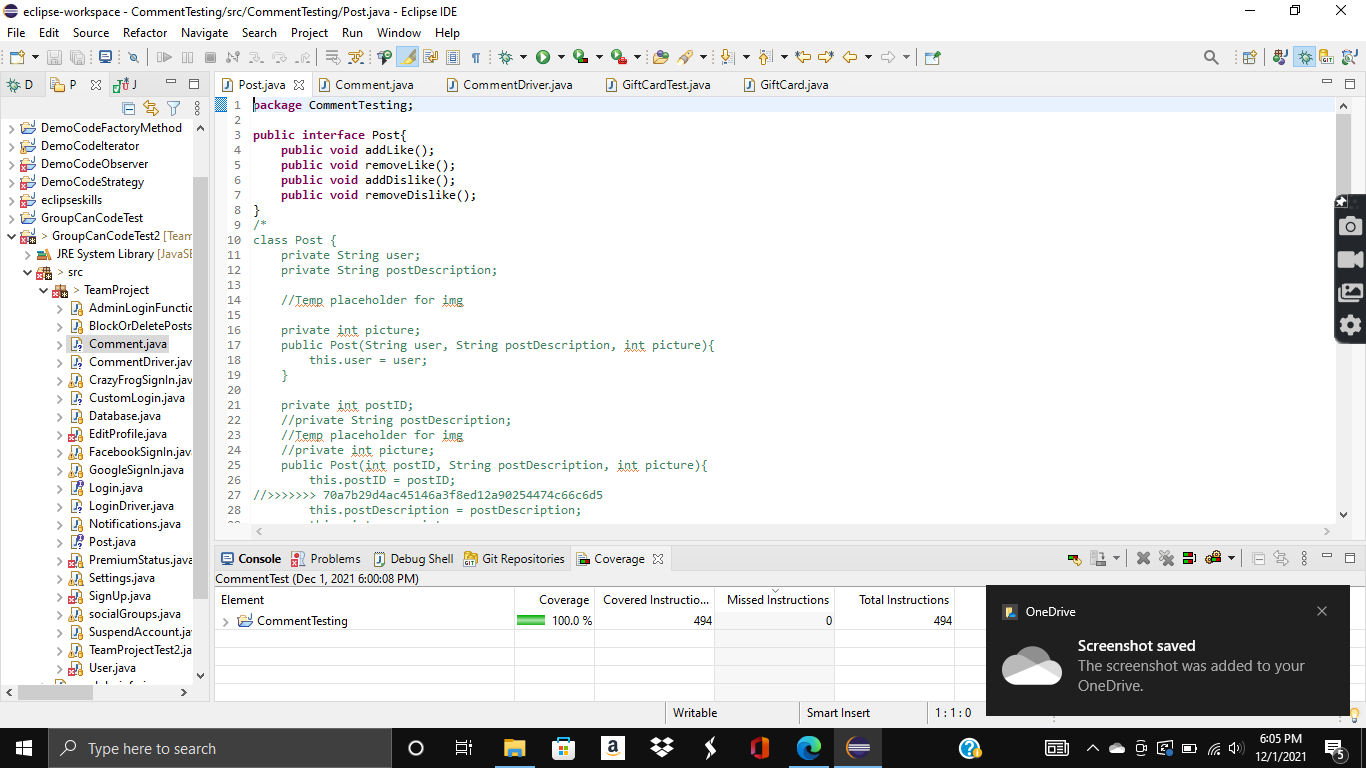
**--->(5 bonus points)**

**1.**

**Composite: Tyler, Santiago - Comment.Java and other related classes. Composite pattern composes objects in terms of a tree structure to represent part as well as whole hierarchy. This type of design pattern comes under structural pattern as this pattern creates a tree structure of a group of objects. We thought composite was most appropriate for comments as you can reply to a comment. (Full code on GitHub)**

****

****

****

**2.**

**\*We didn’t get to the 2nd one, but we figured that any extra credit is better than none. Plus, we get an extra design pattern in case one of our own doesn’t work.**

**3. Bonus Points for using any Interaction Design Principles and any User Interface (such as HTML/JavaScript or Java Swing) --> (5 bonus points)**

**\*We didn’t get to this part. Bummer!**