Team Project Deliverable 2:

UML, Design Patterns, Scrum, Risk Management

CSC 131 – Fall 2021

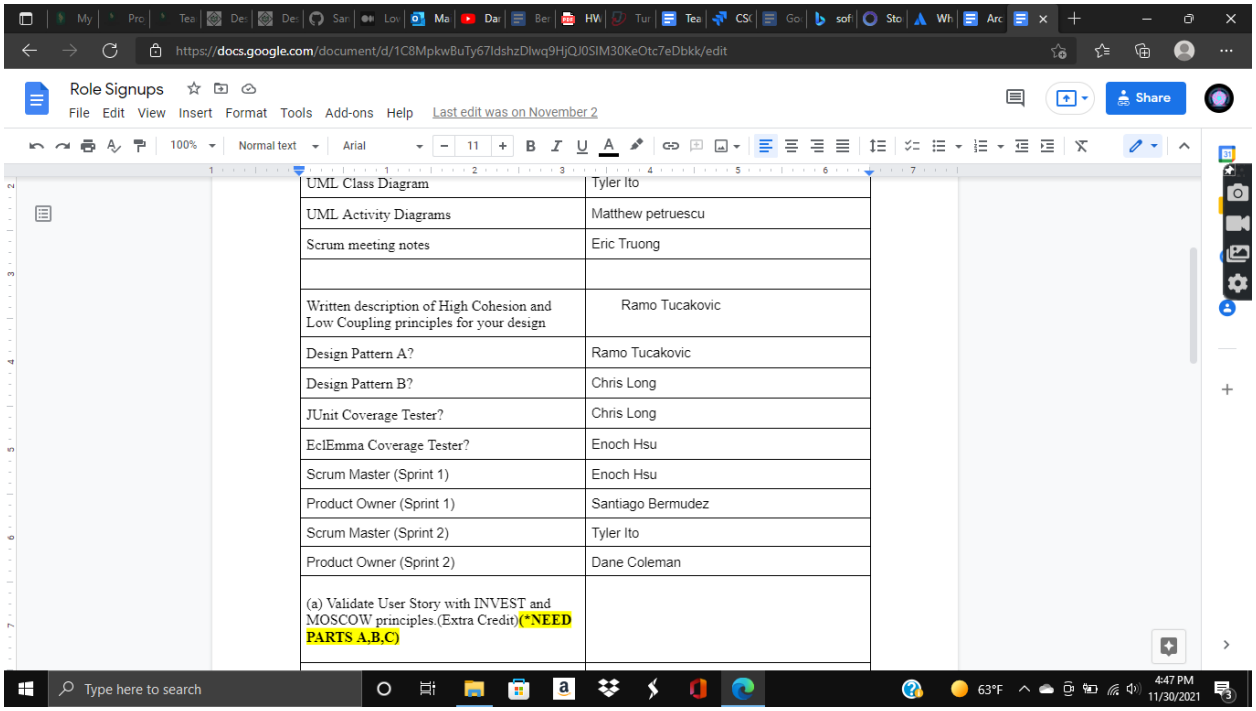
Dane Coleman
Santiago Bermudez
Enoch Hsu
Eric Truong
Tyler Ito
Ramo Tucakovic
Chris Long
Matthew Petruescu

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

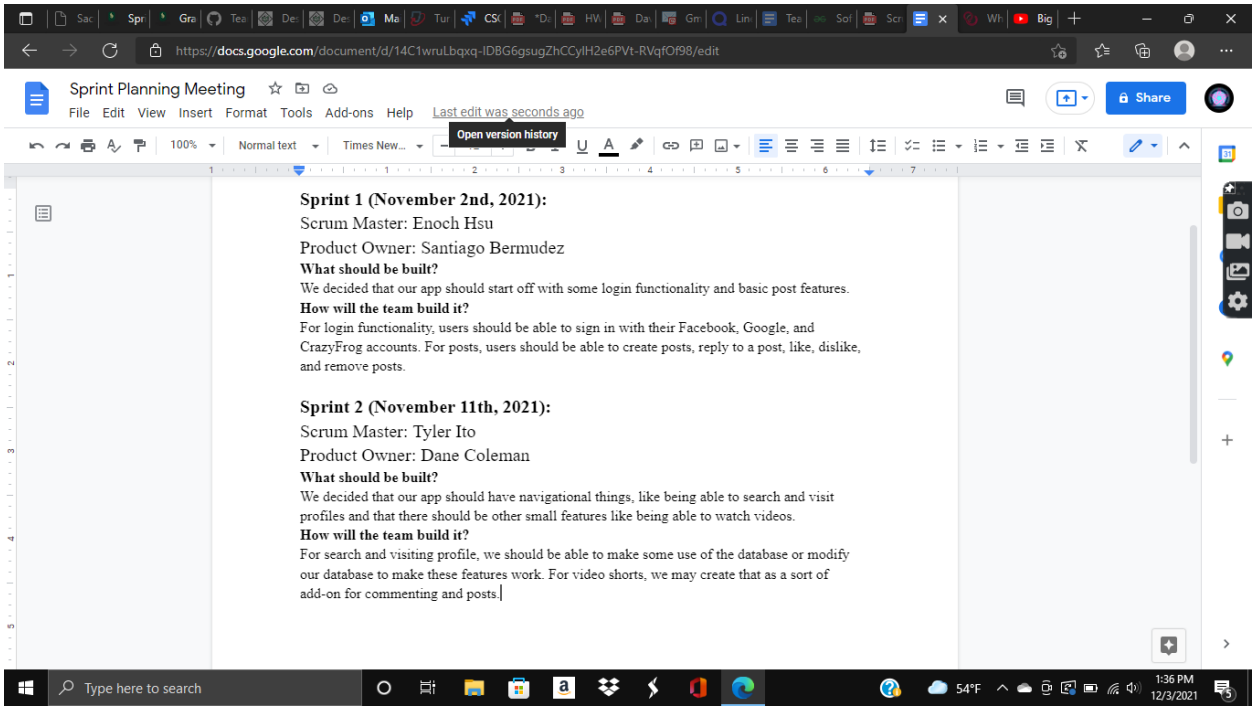
1(a) Assignment of Scrum Master and Product Owner (rotate roles among team members) (5 points)

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)

Sprint 1 Review

1. What went well during the Sprint?
   - We knew what we wanted to include in our project.
   - No one had issues accessing Jira and adding backlog items.
   - We were usually able to meet, otherwise we could communicate on Discord.
   - We were able to agree fairly easily over what to do on the project.

2. What problem did the team run into? How were those problems solved?
   - We had a hard time understanding Jira and user stories at first. We solved these issues by reading supplementary material and by working together on Jira.
   - Some members did not understand how to make changes to the repository. We decided to solve this by allowing others to use their preferred IDE and offering help if needed.
   - We had issues coordinating some of our meetings, so we would remedy this by agreeing on doing certain tasks by a certain date.

Sprint 2 Review

1. What went well during the Sprint?
    - Everyone was able to contribute by committing changes to our Github repository.
    - Everyone was able to pull changes rather easily from the Github repository.
    - We were able to meet more frequently than before, sometimes meeting twice a week.
    - We had lots of outside communication, regularly communicating via Discord.

2. What problem did the team run into? How were those problems solved?
    - We found that it was sometimes hard to delegate tasks and responsibilities, so we would just assign people tasks and have sign up opportunities to help remedy the issue.
    - Some of us did not completely understand design patterns, so we would go back and rereview the course material to ensure that we understood design patterns before implementing them in our code.
    - When it came to choosing design patterns, we would sometimes disagree on the best pattern to use, so we would settle things by vote.

1(d) Sprint Retrospective (5 points)

Sprint 1 Retrospective

1. Inspect how this Sprint went with regards to people, relationships, process, and tools;

People: We were able to communicate rather well, but our process was slow at times as it was to coordinate tasks at times.

Relationships: The relationships between the members on the project team is healthy and we have no issues so far.

Process: We're moving at a steady pace for the project and focusing on delivering a functional product that includes design patterns/architectural styles/patterns.

Tools: Everyone was able to understand how Jira works, but some people had issue with the GitHub repository.

2. Identify and order the major items that went well and potential improvements;
    1. We are consistently meeting every week. **improvement**: being more prepared for meetings and having a plan so we're more time efficient.

    2. Everyone was able to do their part in making diagrams.

    3. Everyone was able to fill in their user stories, but we need a more standardized process for writing and filling in user stories.

3. Create a plan for implementing improvements to the way your team does its work.
    - Having more strict deadlines
    - At the start of meetings, say what we are going to be doing for the day and make sure to finish those by the end of the meeting. (Do our best not to have these items for our next meeting so we are not wasting time)
    - Discuss what concepts need to be studied and understood for the next meeting.

Sprint 2 Retrospective

1. Inspect how this Sprint went with regards to people, relationships, process, and tools;

People: Some people were shy and afraid to pitch in at group meetings, so we may need to do a better job at facilitating cooperation and motivating others.

Relationships: People sometimes have a bit of a tough time understanding the process, so we may have to review things again at times.

Process: We found ourselves moving a little bit more slowly this time, but still making more progress. Perhaps, we are exerting ourselves a bit and making this more complicated than it needs to be?

Tools: This time around, everyone was able to make changes to the repository more easily. We also had people experiment with different IDEs and platforms for the project.

2. Identify and order the major items that went well and potential improvements;

    1. We were all able to look ahead and plan our tasks accordingly.

    2. Everyone was able to implement some aspect of their user stories in the repository, although it is possible that the quality of our work could be higher.

    3. Everyone was able to work on their own code, although we noticed that what we originally planned needed corrections, so we may be better off being more abstract when planning.

3. Create a plan for implementing improvements to the way your team does its work.

    - To increase productivity, we would need to work on reducing the amount of time spent delegating by holding ourselves accountable to knowing and understanding the material before coming to meetings.
    - Instead of having voluntary sign ups, we may need to assign people tasks randomly to save time and effort.
    - We may need to make sure that everyone has easier access to the documents we are working on and is on the same page to avoid lost productivity.

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

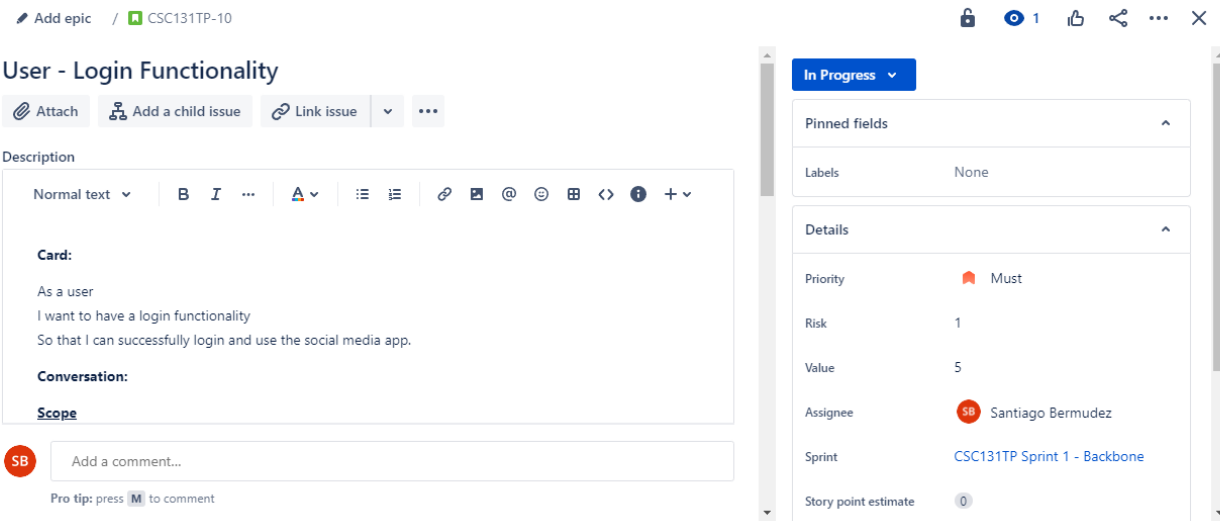
*You guys should have access hopefully, but if not, contact:

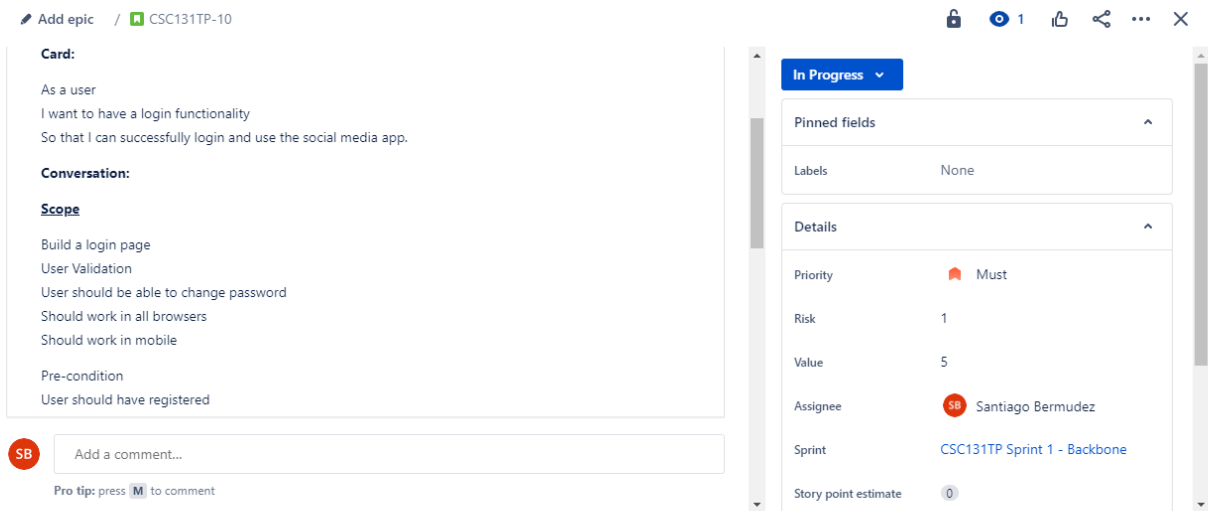
santiagoabermudez@gmail.com

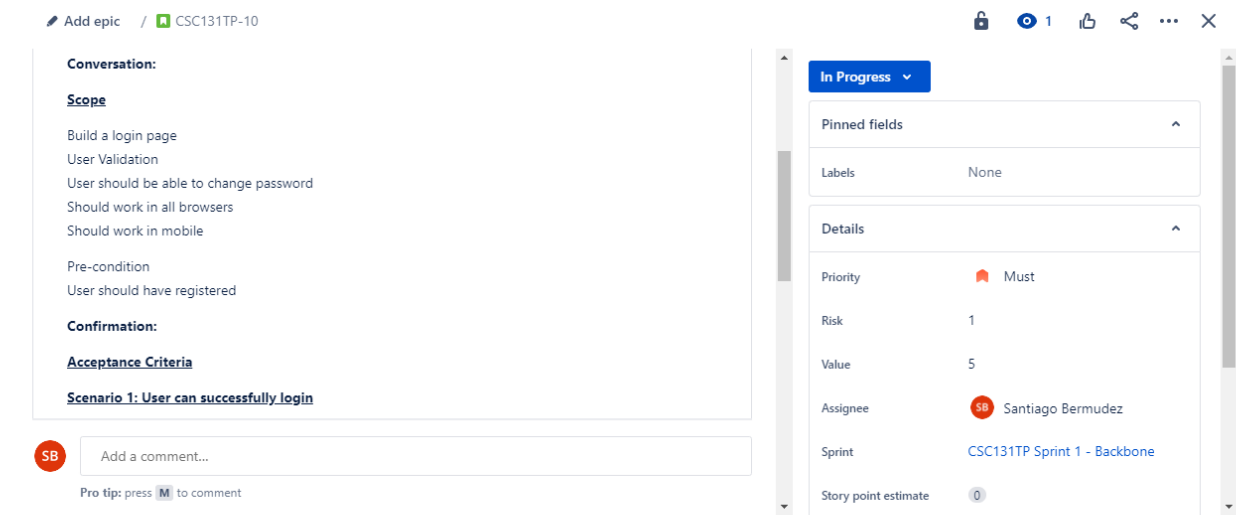
Use of any Scrum tool such as Vivify, Flying Donut or Atlassian/JIRA with:

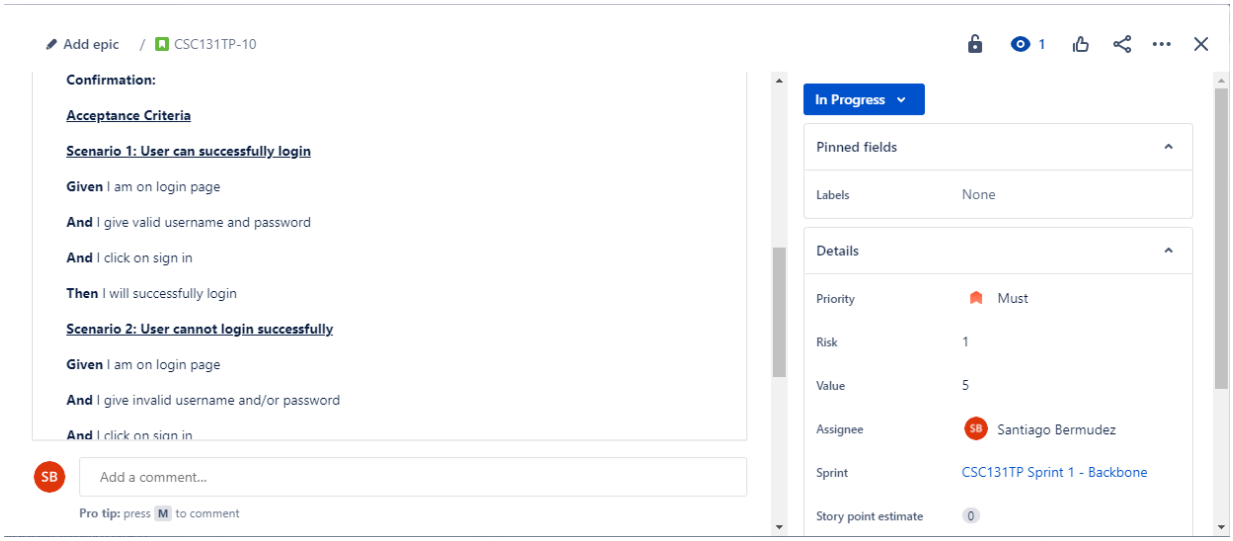
2(a) two examples of CCC (Card, Conversation and Confirmation) in PDF document: (2*2 = 4 points)

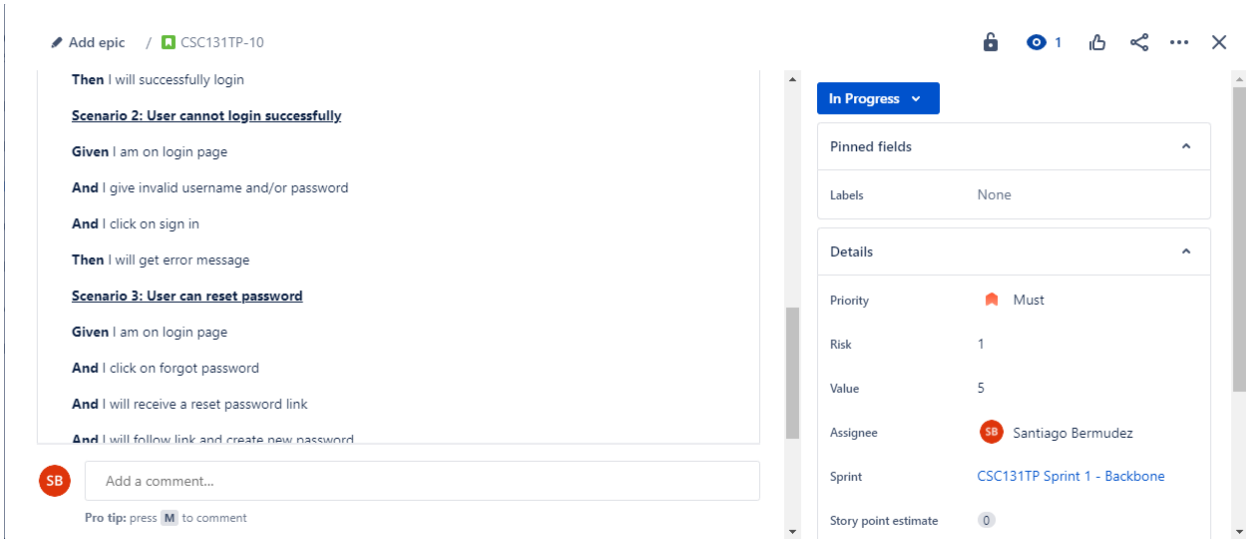
Example 1:

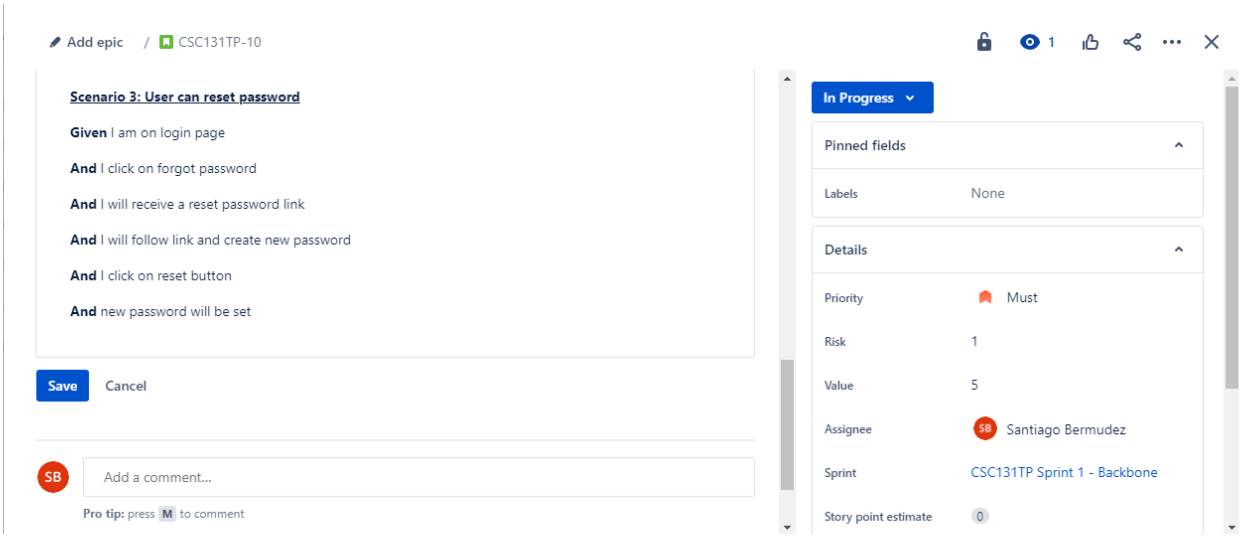




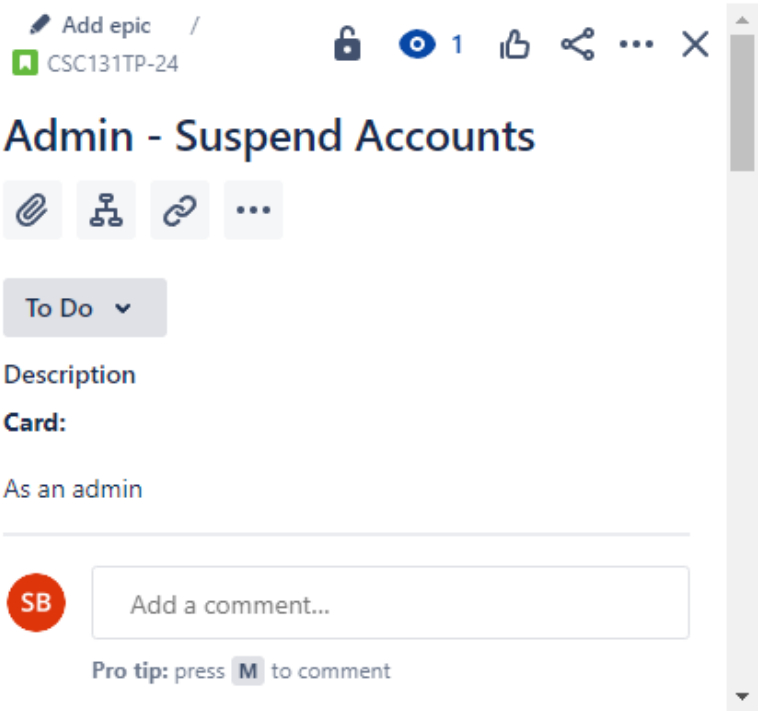

Add epic / CSC131TP-10
1
Conversation:
Scope
Build a login page
User Validation
User should be able to change password
Should work in all browsers
Should work in mobile
Pre-condition
User should have registered
Confirmation:
Acceptance Criteria
Scenario 1: User can successfully login
SB
Add a comment...
Pro tip: press M to comment
In Progress
Pinned fields
Labels
None
Details
Priority
Must
Risk
1
Value
5
Assignee
Santiago Bermudez
Sprint
CSC131TP Sprint 1 - Backbone
Story point estimate
0

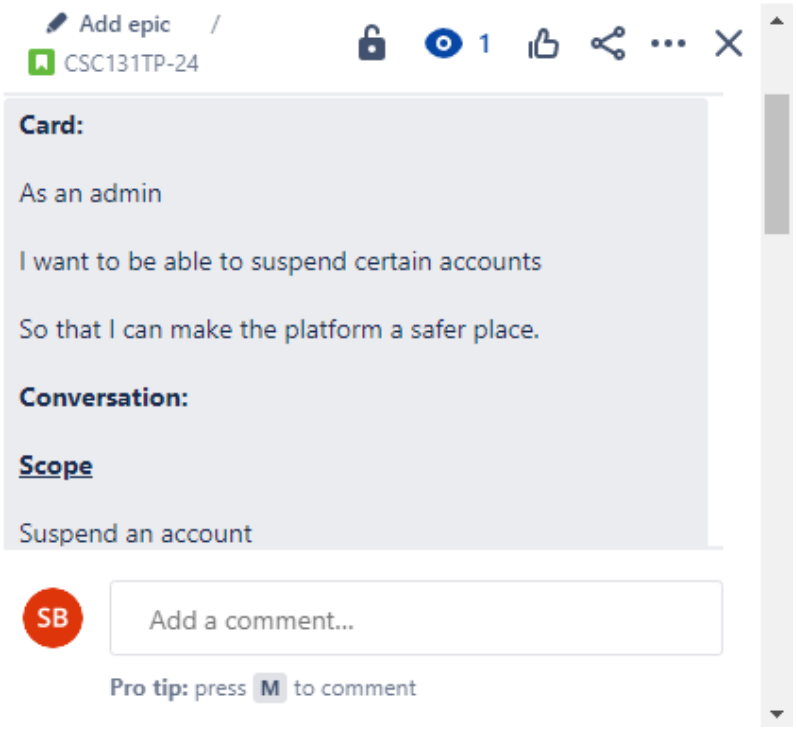

Add epic / CSC131TP-10
1
Confirmation:
Acceptance Criteria
Scenario 1: User can successfully login
Given I am on login page
And I give valid username and password
And I click on sign in
Then I will successfully login
Scenario 2: User cannot login successfully
Given I am on login page
And I give invalid username and/or password
And I click on sign in
SB
Add a comment...
Pro tip: press M to comment
In Progress
Pinned fields
Labels
None
Details
Priority
Must
Risk
1
Value
5
Assignee
Santiago Bermudez
Sprint
CSC131TP Sprint 1 - Backbone
Story point estimate
0

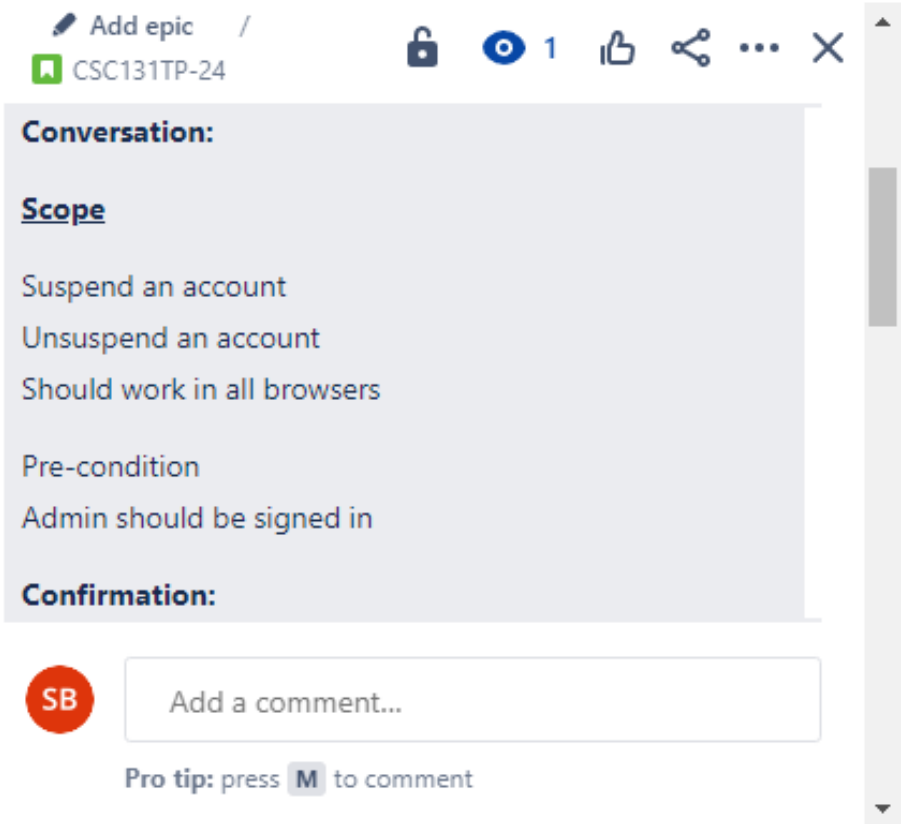

Add epic / CSC131TP-10
1
Then I will successfully login
Scenario 2: User cannot login successfully
Given I am on login page
And I give invalid username and/or password
And I click on sign in
Then I will get error message
Scenario 3: User can reset password
Given I am on login page
And I click on forgot password
And I will receive a reset password link
And I will follow link and create new password
SB
Add a comment...
Pro tip: press M to comment
In Progress
Pinned fields
Labels
None
Details
Priority
Must
Risk
1
Value
5
Assignee
Santiago Bermudez
Sprint
CSC131TP Sprint 1 - Backbone
Story point estimate
0

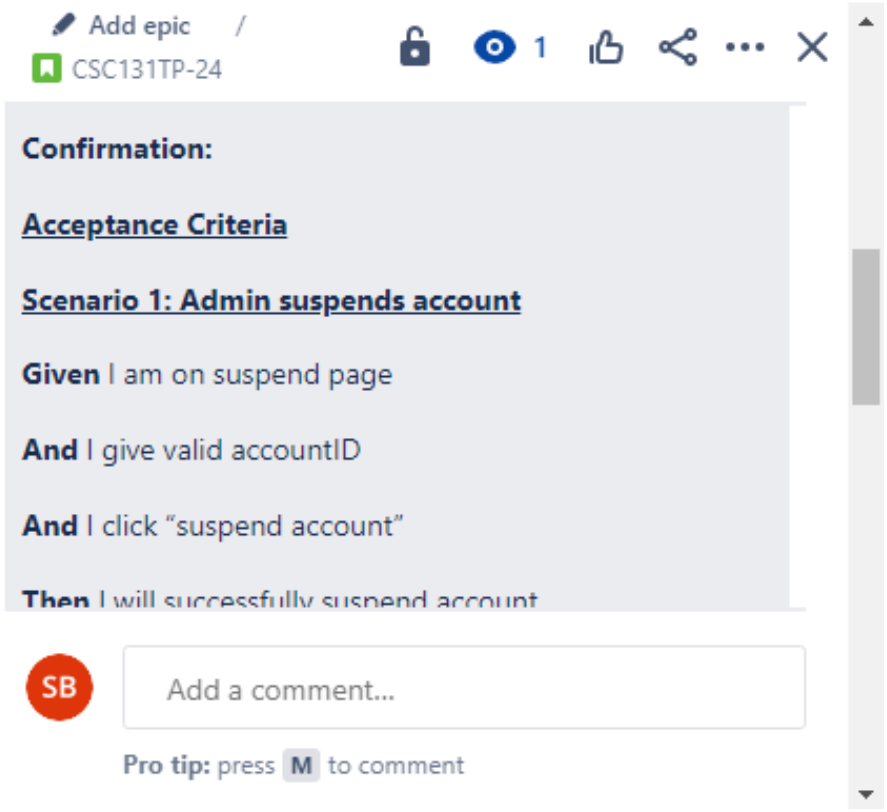

Add epic / CSC131TP-10
1
Scenario 3: User can reset password
Given I am on login page
And I click on forgot password
And I will receive a reset password link
And I will follow link and create new password
And I click on reset button
And new password will be set
Save
Cancel
SB
Add a comment...
Pro tip: press M to comment
In Progress
Pinned fields
Labels
None
Details
Priority
Must
Risk
1
Value
5
Assignee
Santiago Bermudez
Sprint
CSC131TP Sprint 1 - Backbone
Story point estimate
0

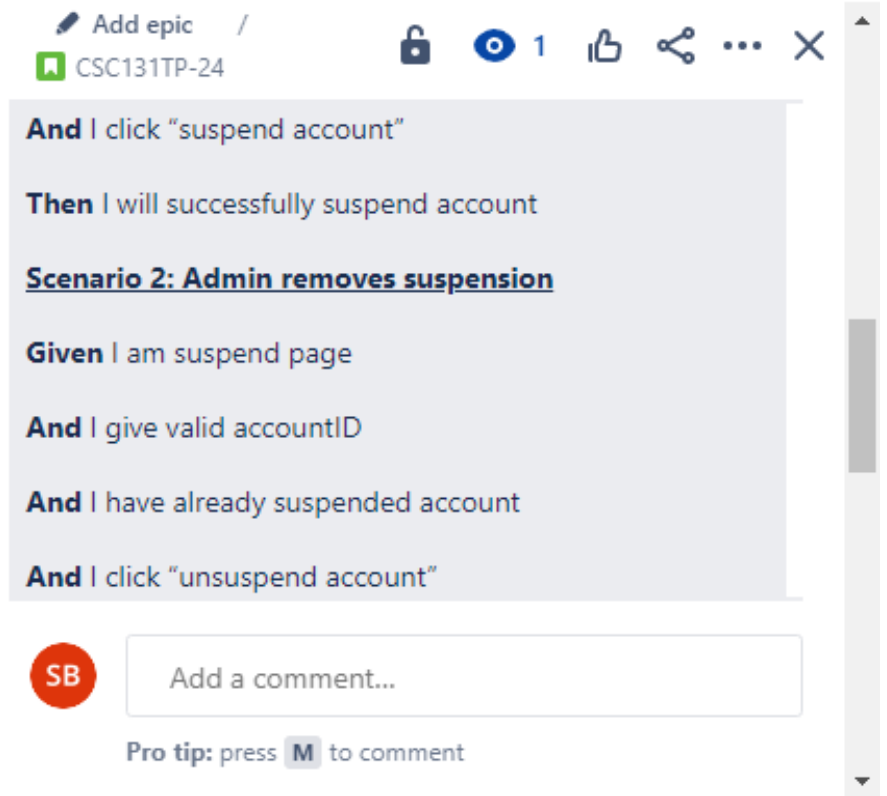
Example 2:

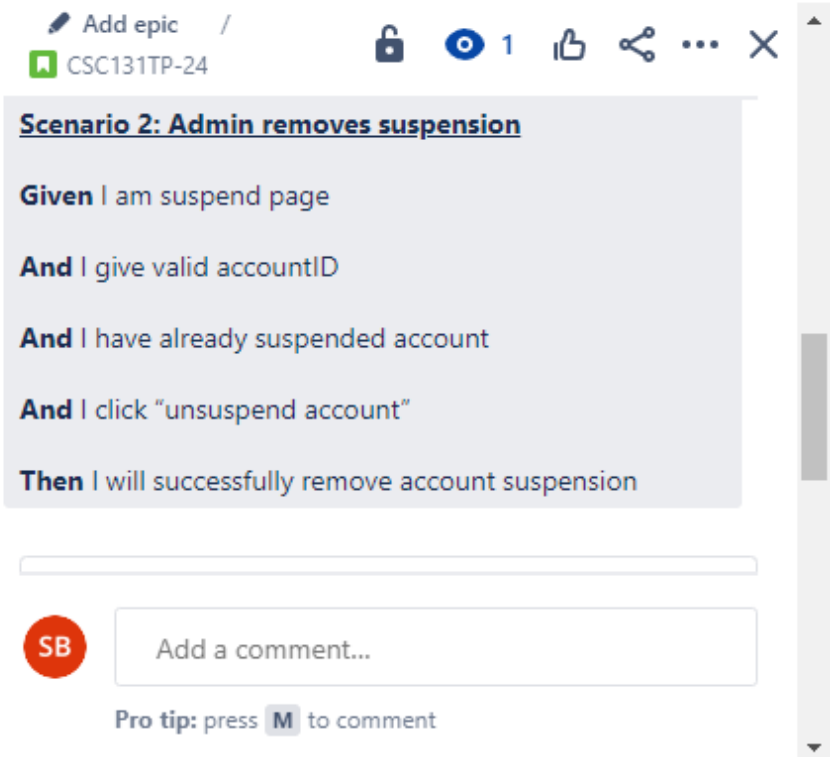

Add epic /
CSC131TP-24
1
Admin - Suspend Accounts
To Do
Description
Card:
As an admin
SB
Add a comment...
Pro tip: press M to comment


Add epic
/
CSC131TP-24
1
Card:
As an admin
I want to be able to suspend certain accounts
So that I can make the platform a safer place.
Conversation:
Scope
Suspend an account
SB
Add a comment...
Pro tip: press M to comment

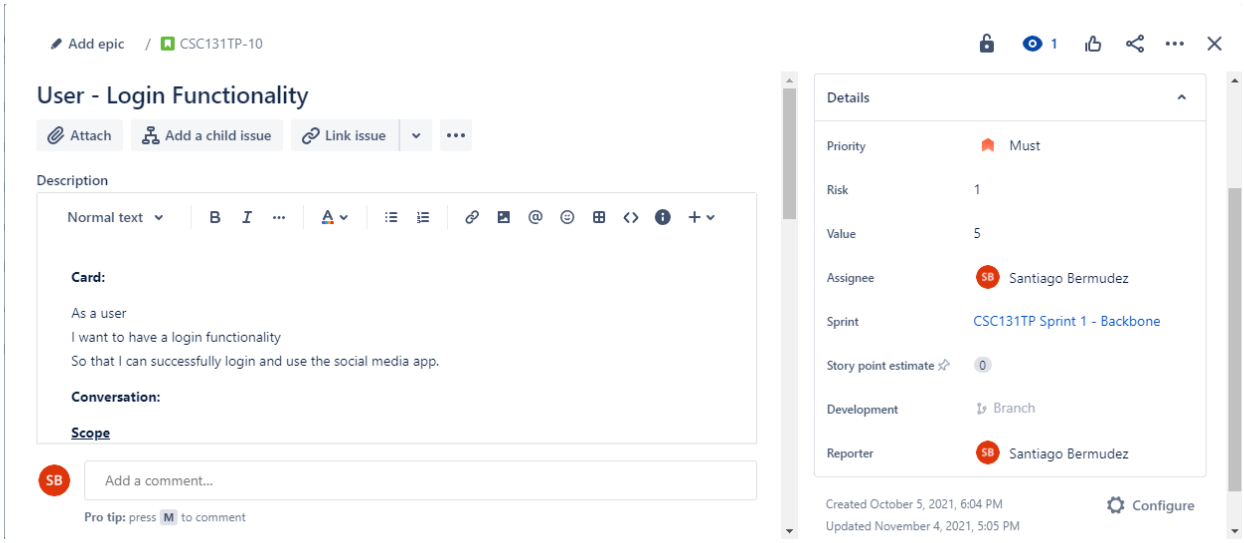

Add epic
/
CSC131TP-24
1
Conversation:
Scope
Suspend an account
Unsuspend an account
Should work in all browsers
Pre-condition
Admin should be signed in
Confirmation:
SB
Add a comment...
Pro tip: press M to comment

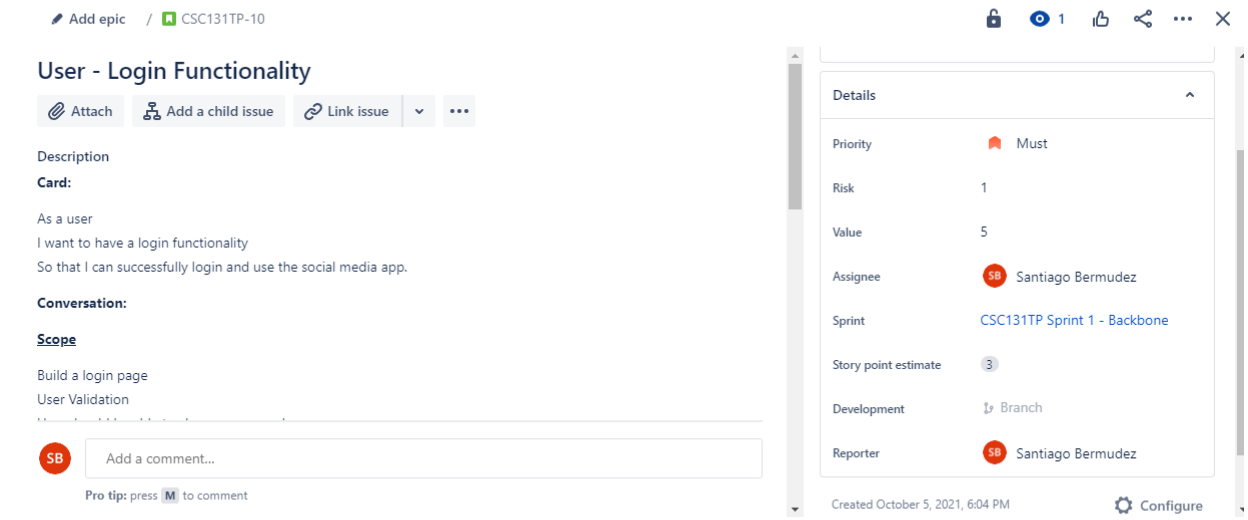

Add epic
/
CSC131TP-24
1
Confirmation:
Acceptance Criteria
Scenario 1: Admin suspends account
Given I am on suspend page
And I give valid accountID
And I click "suspend account"
Then I will successfully suspend account
SB
Add a comment...
Pro tip: press M to comment


Add epic
/
CSC131TP-24
1
And I click "suspend account"
Then I will successfully suspend account
Scenario 2: Admin removes suspension
Given I am suspend page
And I give valid accountID
And I have already suspended account
And I click "unsuspend account"
SB
Add a comment...
Pro tip: press M to comment



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





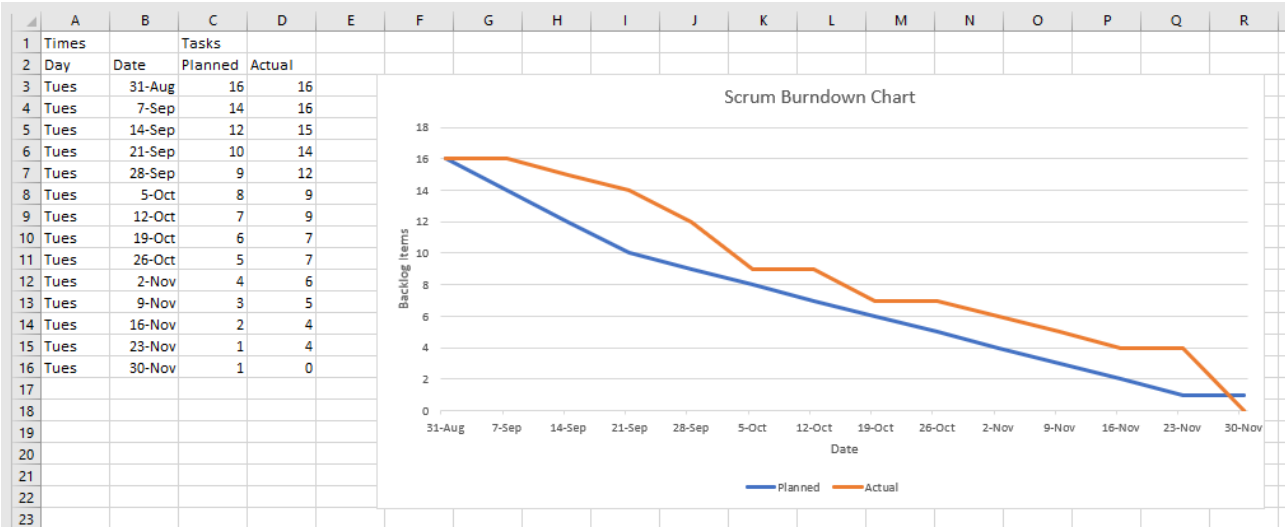
*In the details on the right, you can see our prioritization, task assignment, and story points. You can also see our risk and value estimates. You should see the same for the rest of our user stories in Jira.

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's don'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.	1. Unable to implement a feature in the backlog. 2. Financial costs start to creep up.	1. Poor user experience (slow loading app) 2. Website server crashing.

	3. User's accounts hacked		

2(d) and burndown chart (4 points)



	A	B	C	D
1	Times		Tasks	
2	Day	Date	Planned	Actual
3	Tues	31-Aug	16	16
4	Tues	7-Sep	14	16
5	Tues	14-Sep	12	15
6	Tues	21-Sep	10	14
7	Tues	28-Sep	9	12
8	Tues	5-Oct	8	9
9	Tues	12-Oct	7	9
10	Tues	19-Oct	6	7
11	Tues	26-Oct	5	7
12	Tues	2-Nov	4	6
13	Tues	9-Nov	3	5
14	Tues	16-Nov	2	4
15	Tues	23-Nov	1	4
16	Tues	30-Nov	1	0

3. Coding using GitHub with:

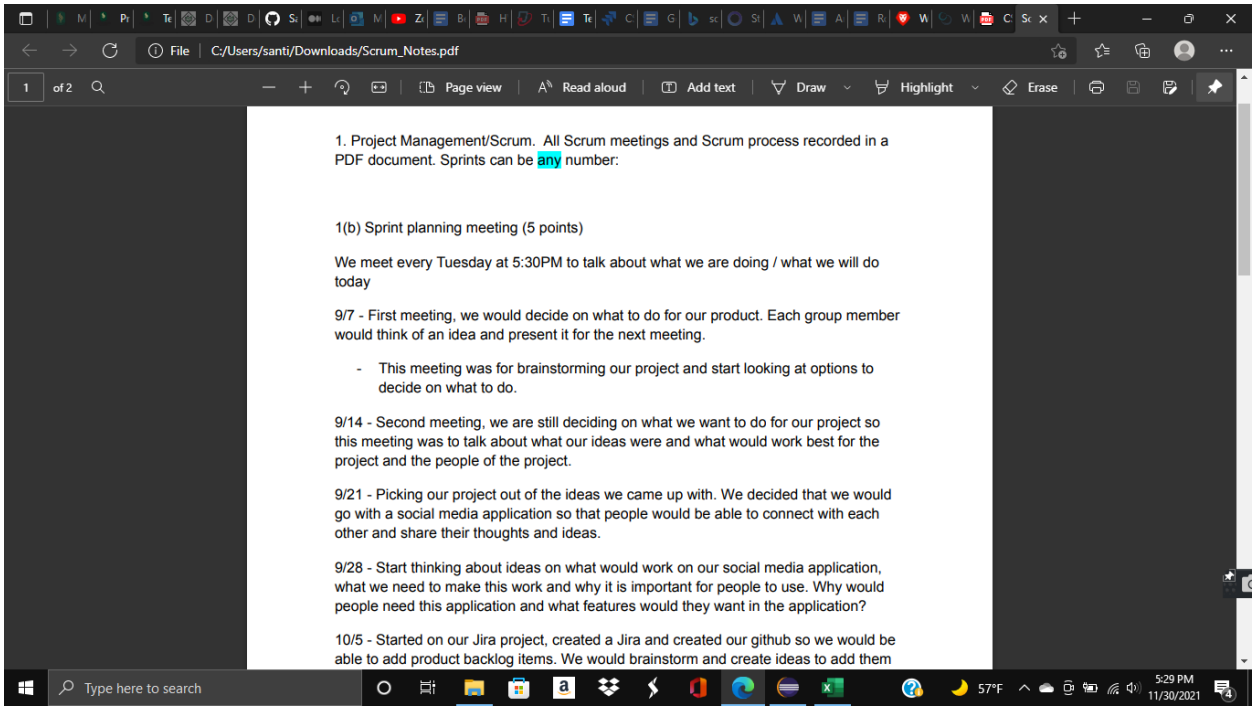
3(a) proof of collaboration (5 points) [GitHub or GitLab collaboration details and Scrum meeting notes).

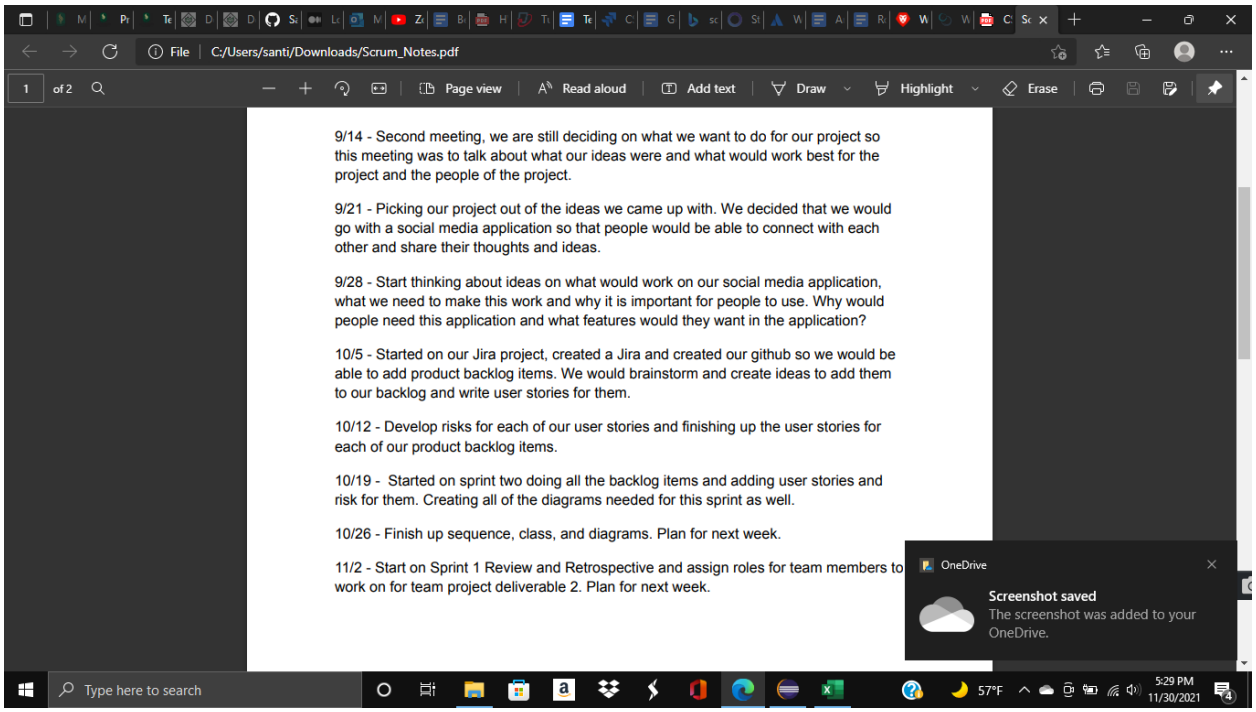
https://github.com/Santiago13225/TeamProjectCrazyFrogSocial

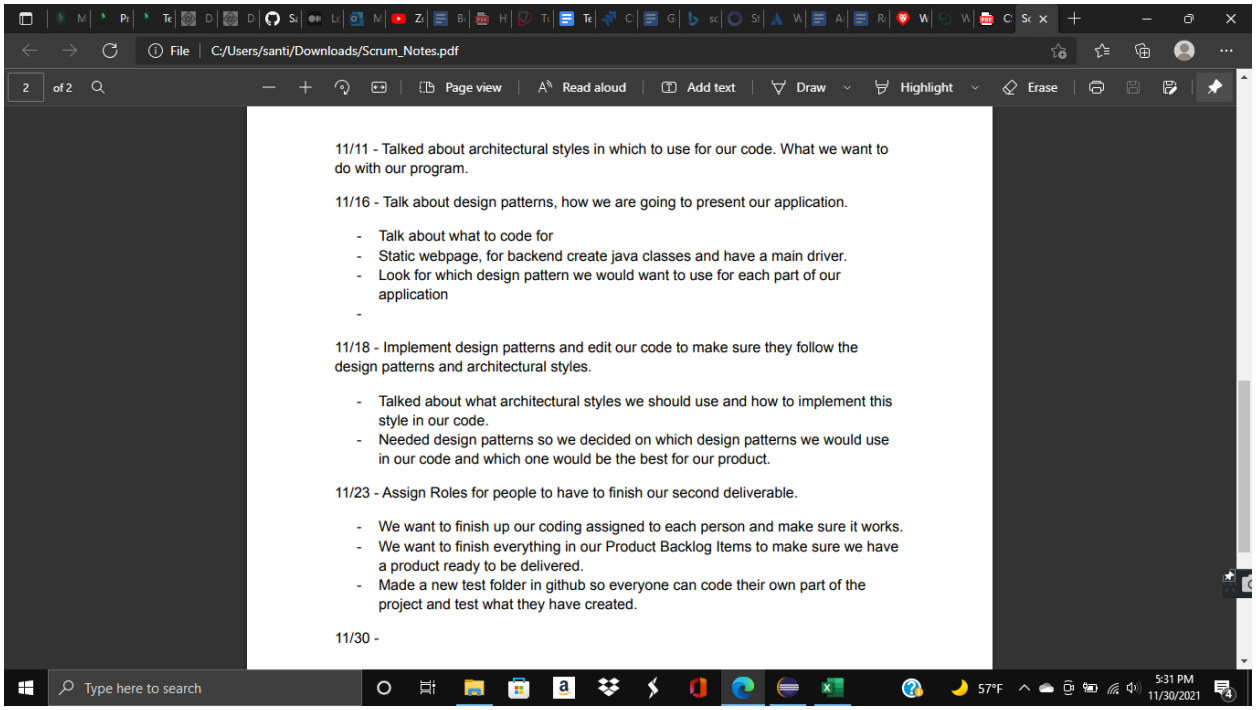
*Everyone should have access. You were all sent an invite by your Sac State email to be collaborators just so you can see our project. If you can't access anything, contact:

santiagoabermudez@csus.edu

*Scrum meeting notes below:


File C:/Users/santi/Downloads/Scrum_Notes.pdf
1 of 2
Page view
Read aloud
Add text
Draw
Highlight
Erase
1. Project Management/Scrum. All Scrum meetings and Scrum process recorded in a PDF document. Sprints can be any number:
1(b) Sprint planning meeting (5 points)
We meet every Tuesday at 5:30PM to talk about what we are doing / what we will do today
9/7 - First meeting, we would decide on what to do for our product. Each group member would think of an idea and present it for the next meeting.
- This meeting was for brainstorming our project and start looking at options to decide on what to do.
9/14 - Second meeting, we are still deciding on what we want to do for our project so this meeting was to talk about what our ideas were and what would work best for the project and the people of the project.
9/21 - Picking our project out of the ideas we came up with. We decided that we would go with a social media application so that people would be able to connect with each other and share their thoughts and ideas.
9/28 - Start thinking about ideas on what would work on our social media application, what we need to make this work and why it is important for people to use. Why would people need this application and what features would they want in the application?
10/5 - Started on our Jira project, created a Jira and created our github so we would be able to add product backlog items. We would brainstorm and create ideas to add them
Type here to search
57°F
5:29 PM
11/30/2021


File C:/Users/santi/Downloads/Scrum_Notes.pdf
1 of 2
Page view
Read aloud
Add text
Draw
Highlight
Erase
9/14 - Second meeting, we are still deciding on what we want to do for our project so this meeting was to talk about what our ideas were and what would work best for the project and the people of the project.
9/21 - Picking our project out of the ideas we came up with. We decided that we would go with a social media application so that people would be able to connect with each other and share their thoughts and ideas.
9/28 - Start thinking about ideas on what would work on our social media application, what we need to make this work and why it is important for people to use. Why would people need this application and what features would they want in the application?
10/5 - Started on our Jira project, created a Jira and created our github so we would be able to add product backlog items. We would brainstorm and create ideas to add them to our backlog and write user stories for them.
10/12 - Develop risks for each of our user stories and finishing up the user stories for each of our product backlog items.
10/19 - Started on sprint two doing all the backlog items and adding user stories and risk for them. Creating all of the diagrams needed for this sprint as well.
10/26 - Finish up sequence, class, and diagrams. Plan for next week.
11/2 - Start on Sprint 1 Review and Retrospective and assign roles for team members to work on for team project deliverable 2. Plan for next week.
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
Type here to search
57°F
5:29 PM
11/30/2021



11/11 - Talked about architectural styles in which to use for our code. What we want to do with our program.

11/16 - Talk about design patterns, how we are going to present our application.

- Talk about what to code for
- Static webpage, for backend create java classes and have a main driver.
- Look for which design pattern we would want to use for each part of our application
-

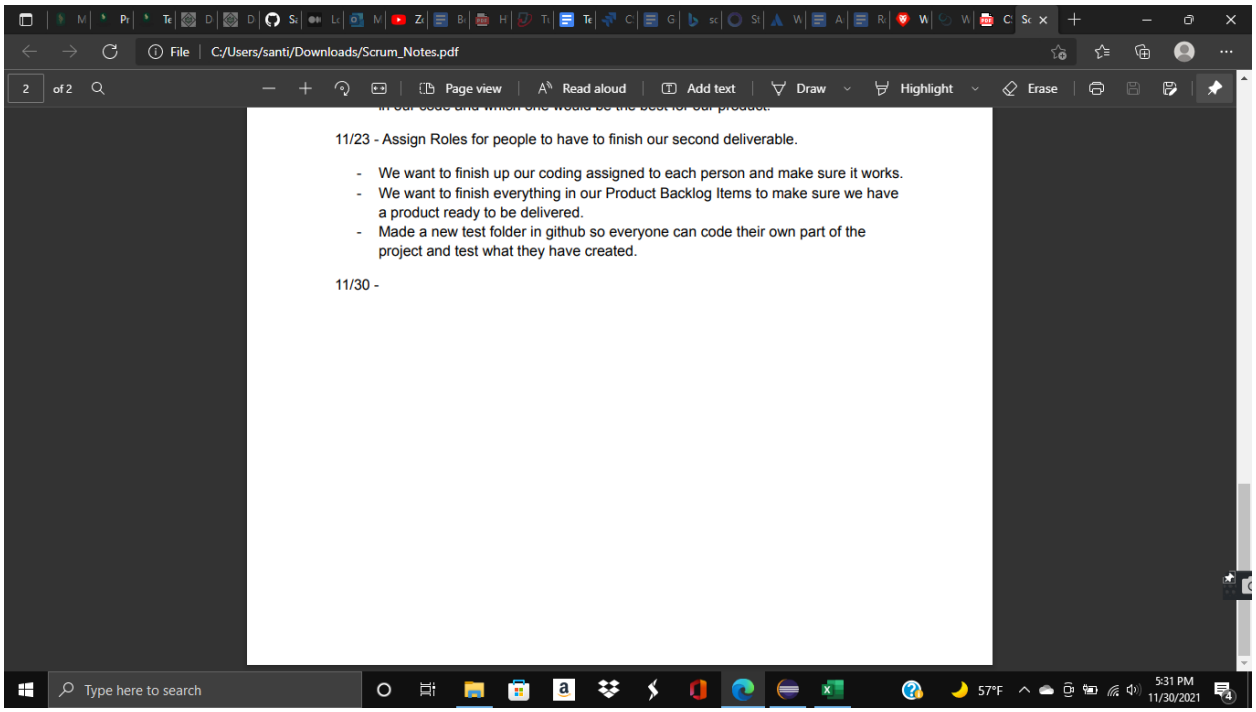
11/18 - Implement design patterns and edit our code to make sure they follow the design patterns and architectural styles.

- Talked about what architectural styles we should use and how to implement this style in our code.
- Needed design patterns so we decided on which design patterns we would use in our code and which one would be the best for our product.

11/23 - Assign Roles for people to have to finish our second deliverable.

- We want to finish up our coding assigned to each person and make sure it works.
- We want to finish everything in our Product Backlog Items to make sure we have a product ready to be delivered.
- Made a new test folder in github so everyone can code their own part of the project and test what they have created.

11/30 -



11/23 - Assign Roles for people to have to finish our second deliverable.

- We want to finish up our coding assigned to each person and make sure it works.
- We want to finish everything in our Product Backlog Items to make sure we have a product ready to be delivered.
- Made a new test folder in github so everyone can code their own part of the project and test what they have created.

11/30 -

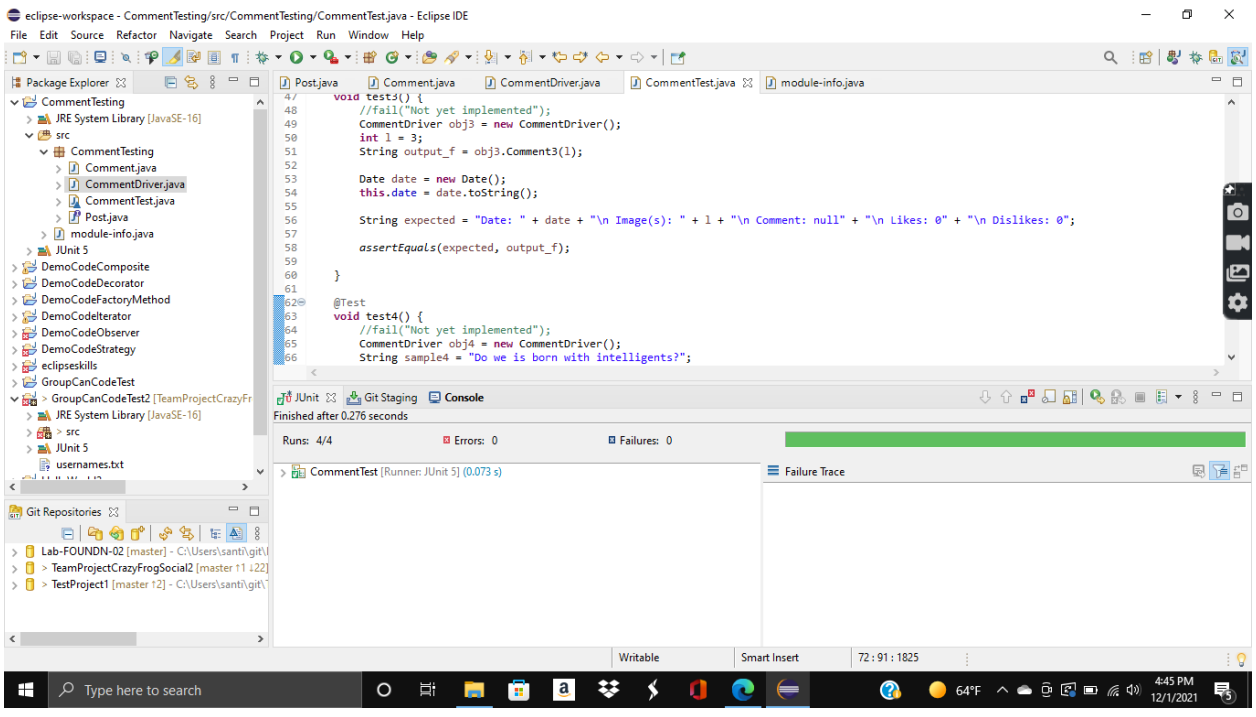
3(b) written description of High Cohesion and Low Coupling principles for your design (5 points)

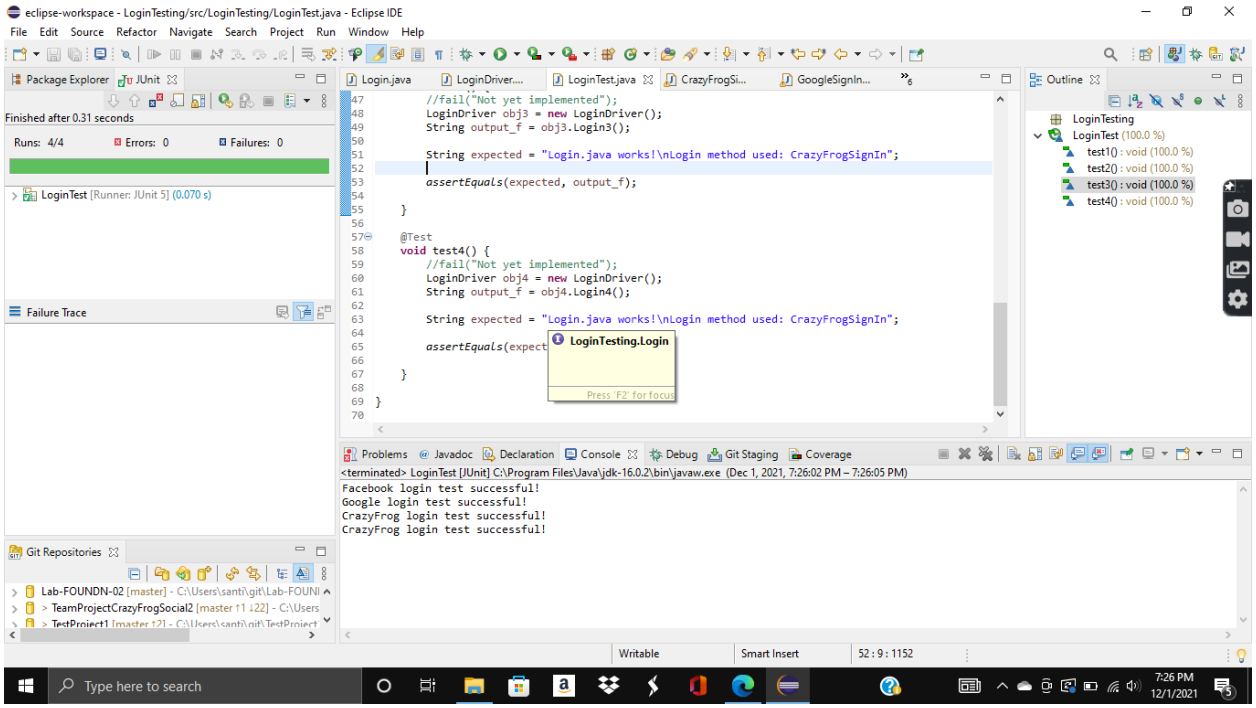
Low Coupling: For low coupling, we have classes of code that are split into certain groups related to their purpose like, a separate group for logging and a separate group of code for commenting and posts. For low coupling between these two modules, we made it so that our code uses data coupling as opposed 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)

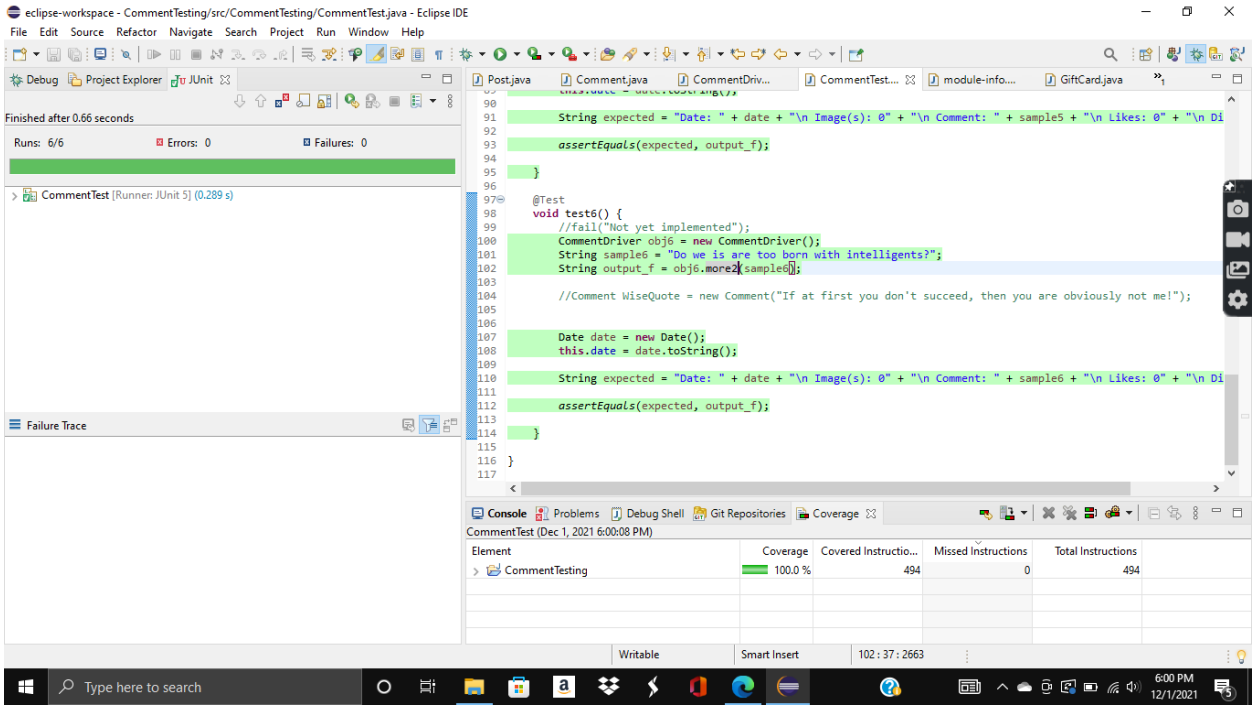
*We had a lot of code, so we had 2 groups of 4 people take on half of the code each.

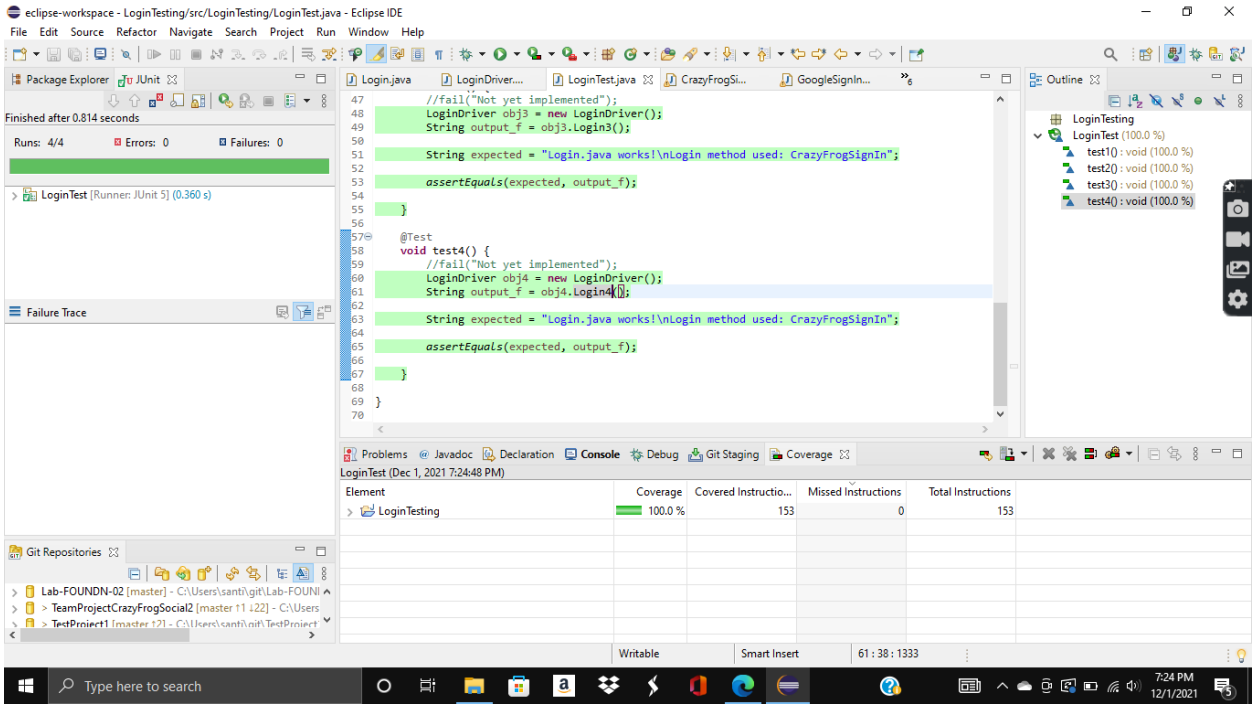




3 (d) using EclEmma (5 points) (100% code coverage)

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

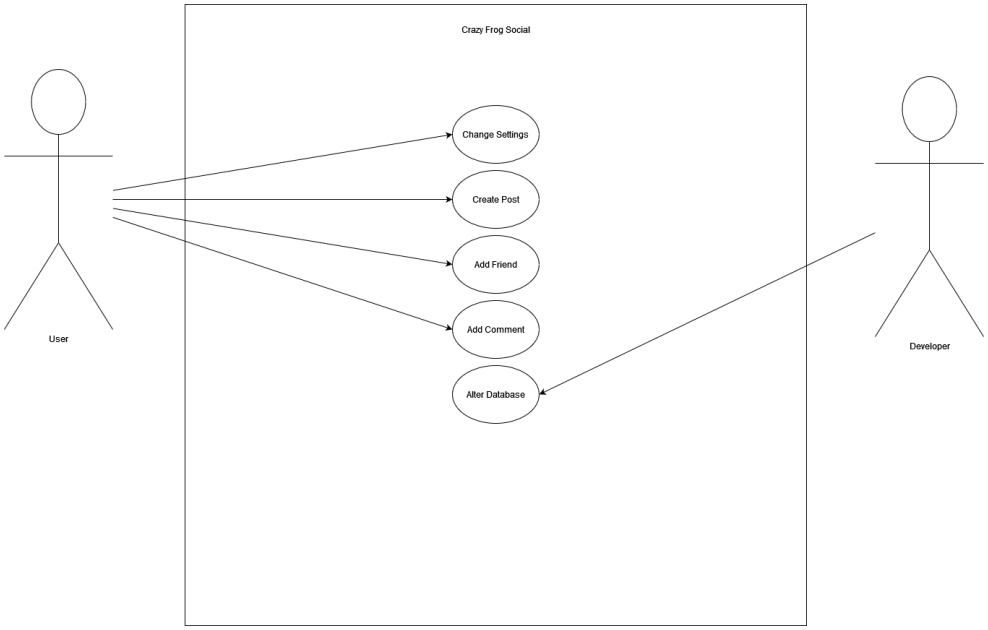




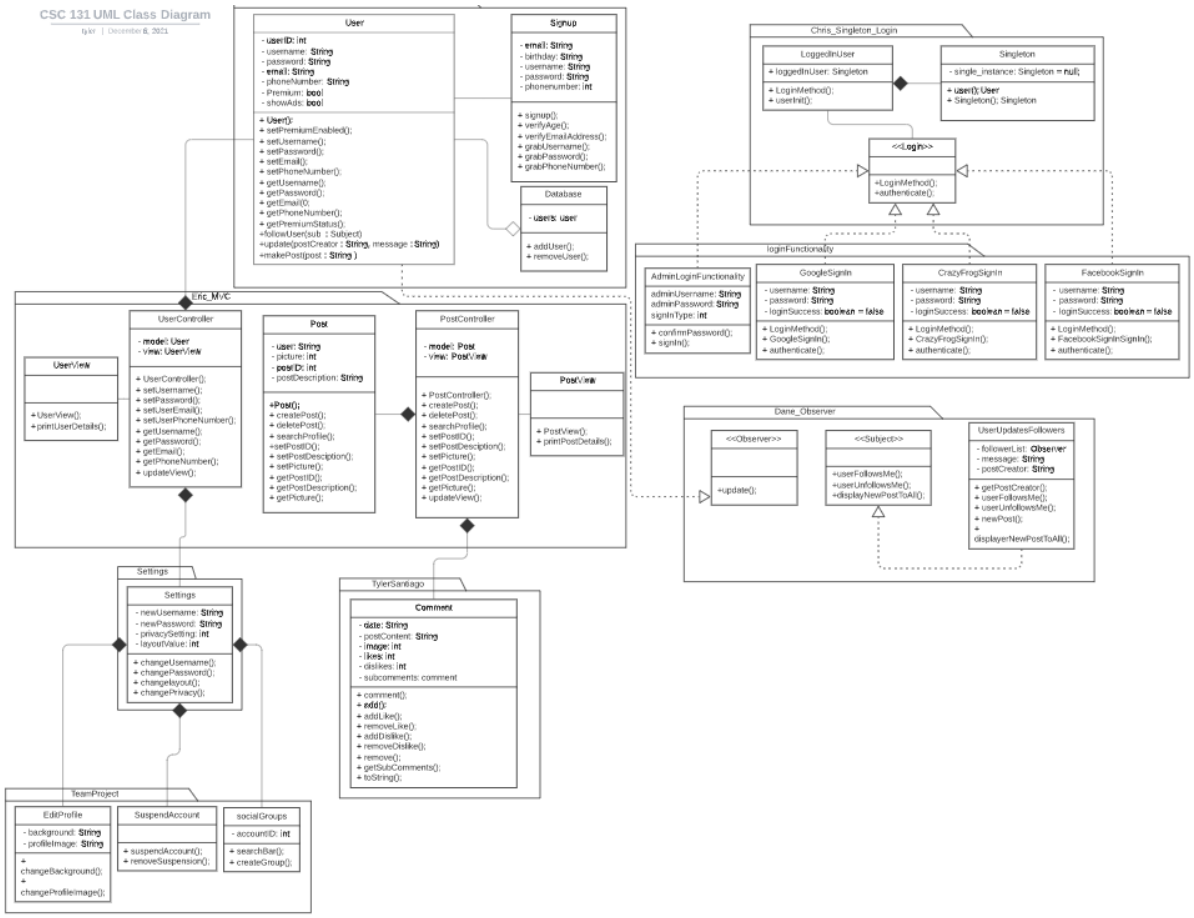
4. Four types of UML Diagrams:

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.

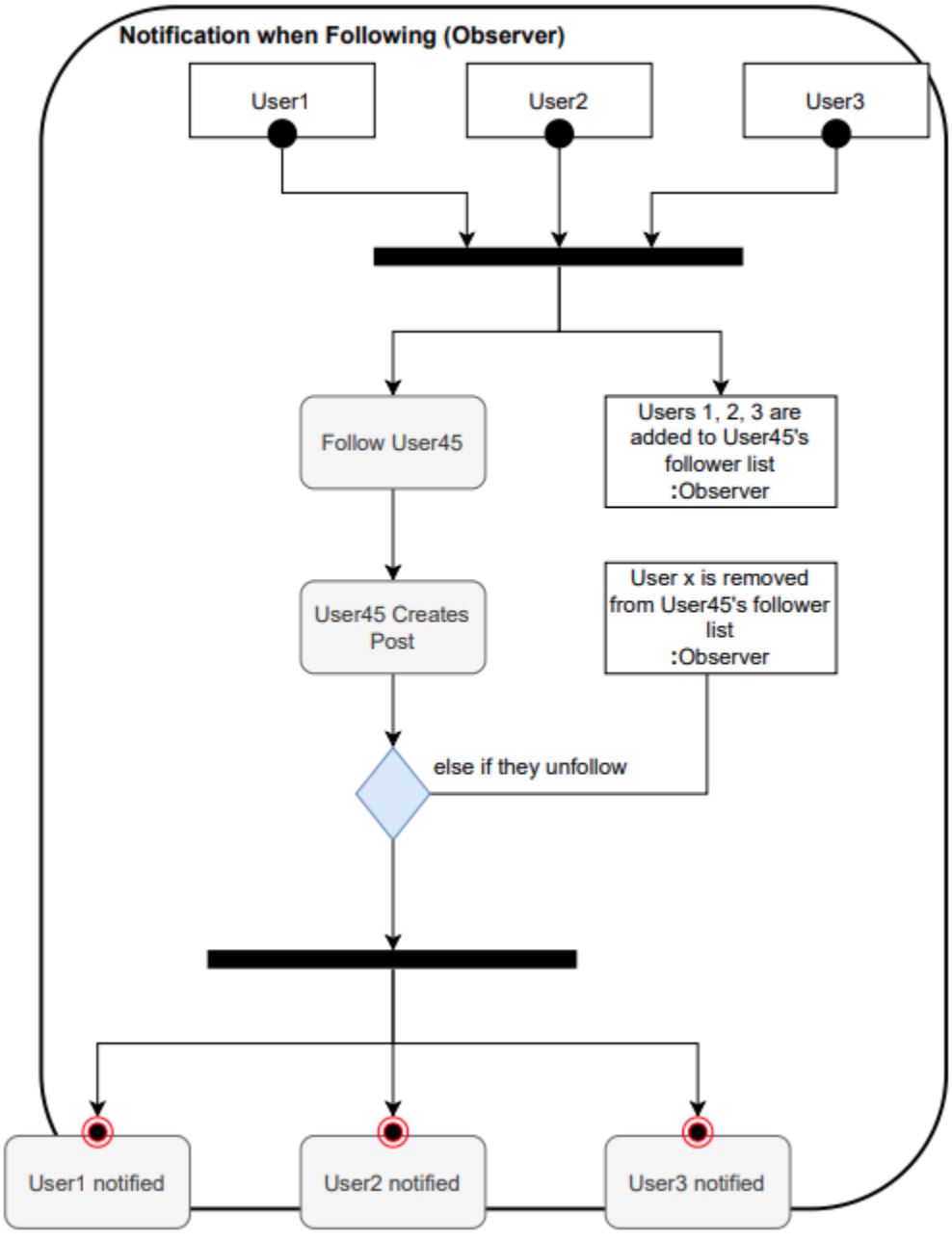
4(a) Use Case Diagram (5 points) (for part of code is ok)



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

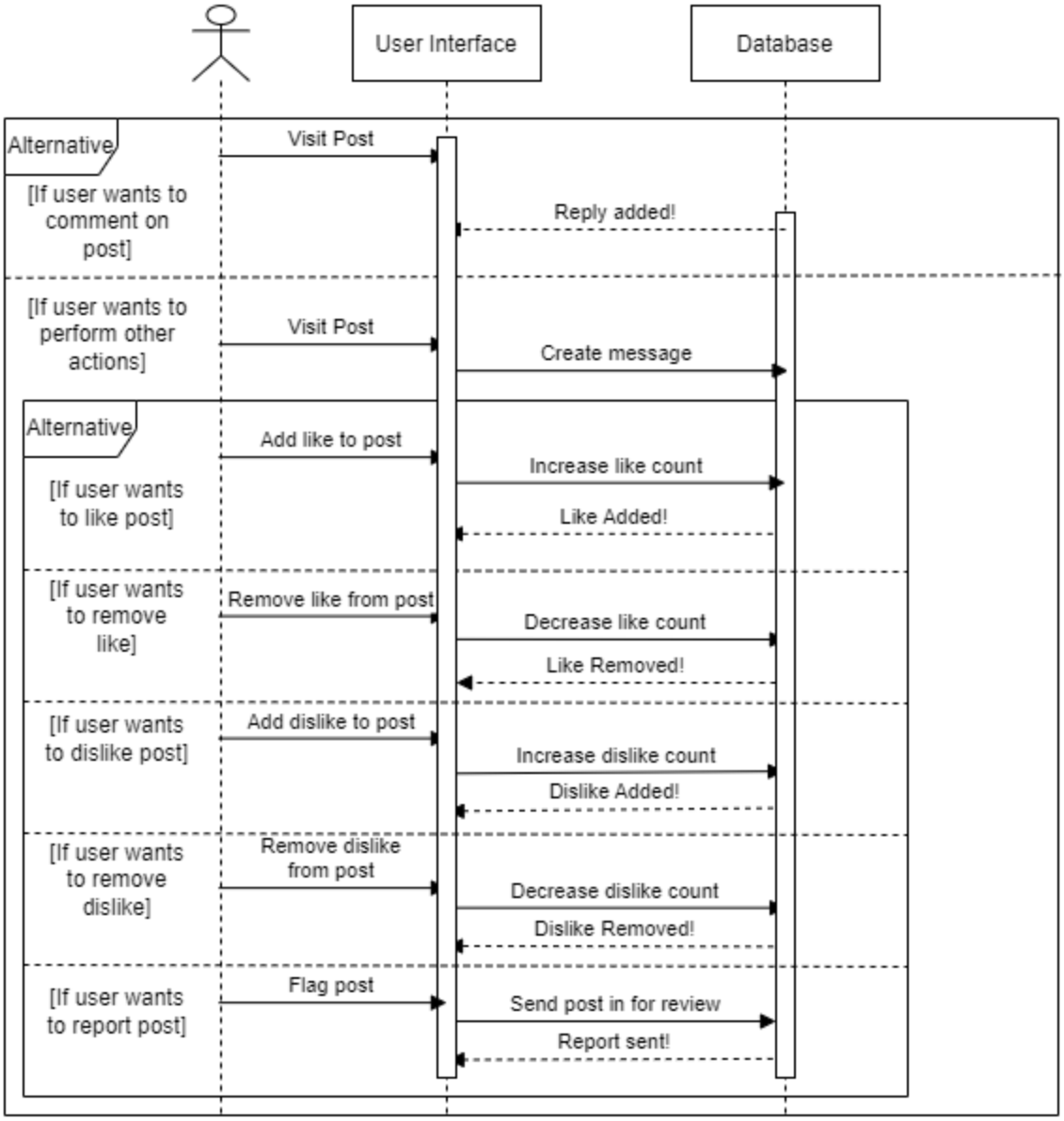


4(c) Activity Diagram (5 points) (for part of code is ok)



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

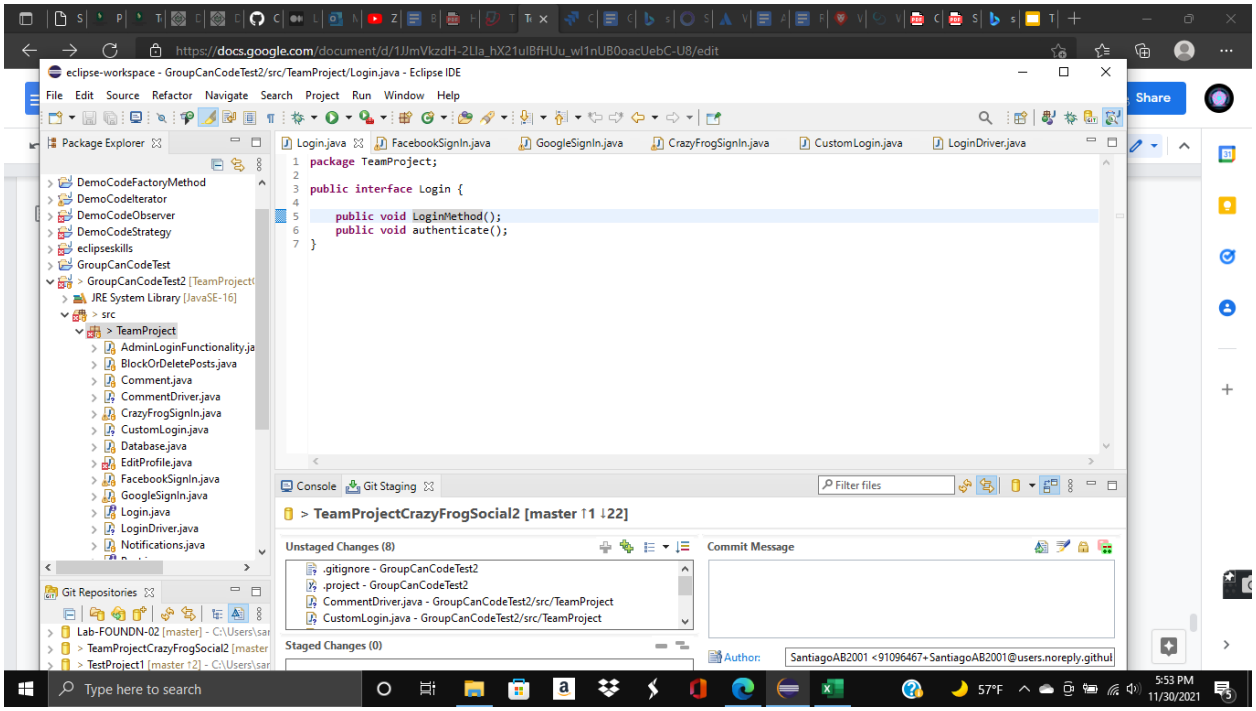
For replying to other posts:

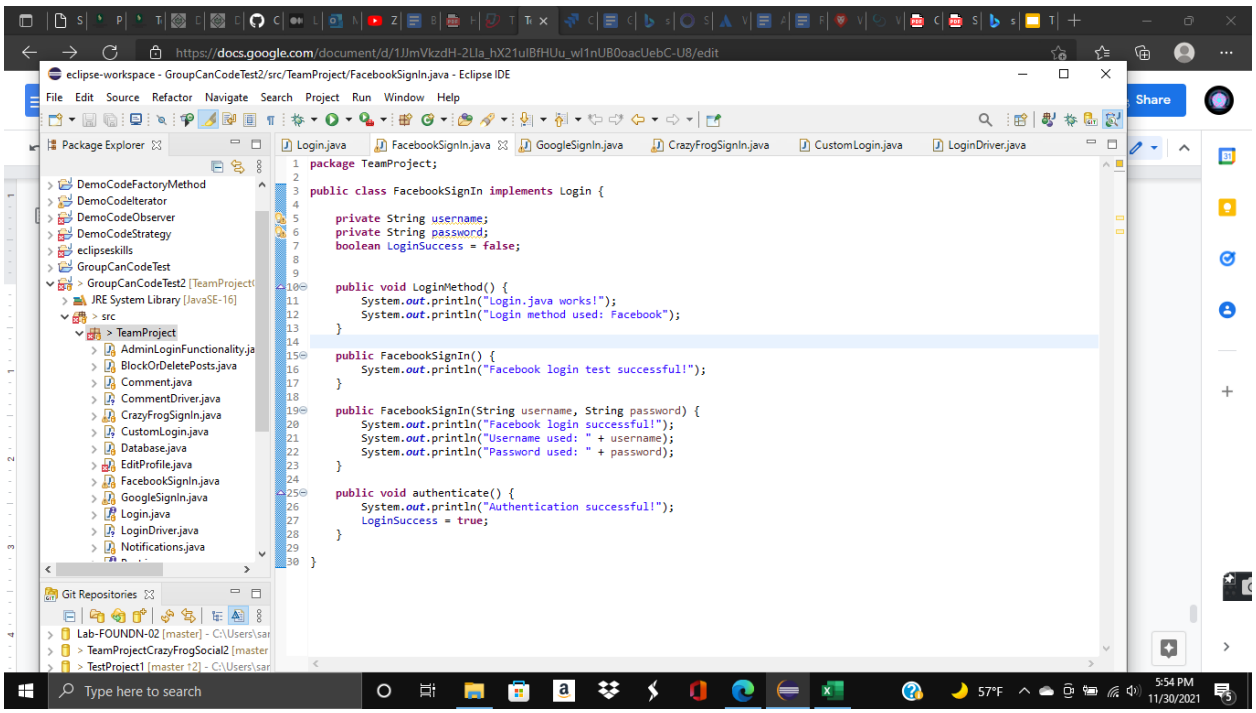


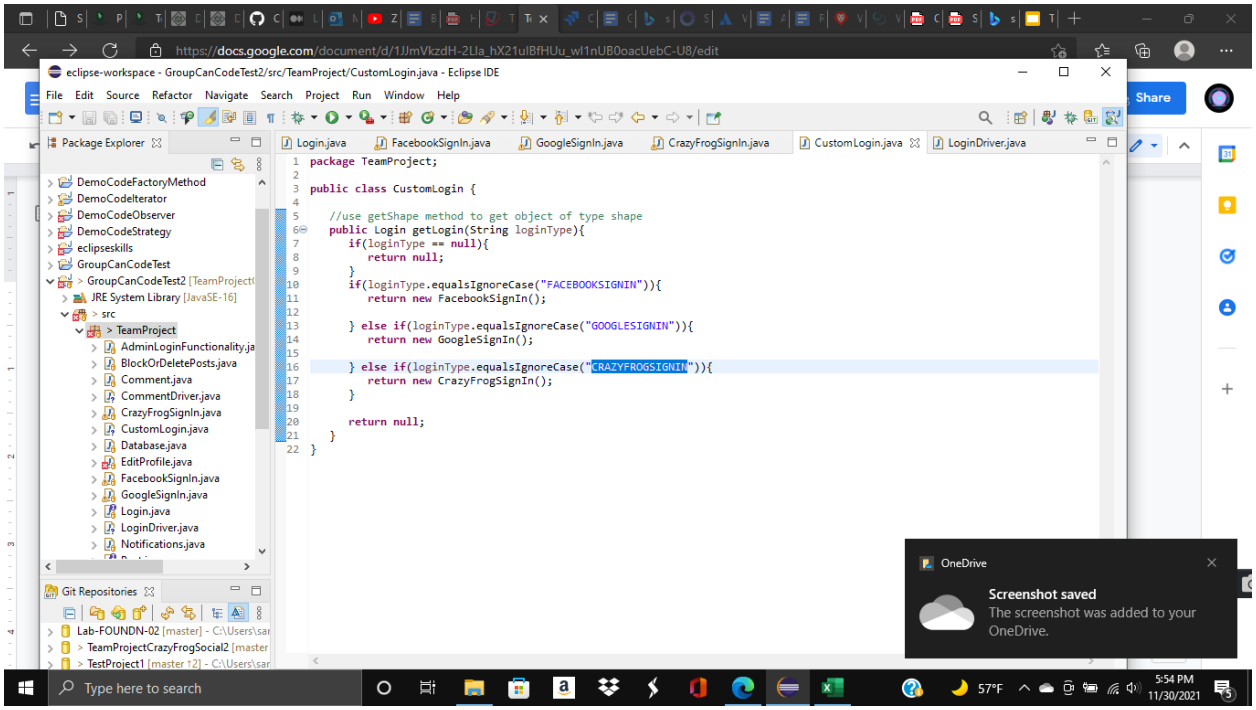
5. Minimum 2 Design Patterns: (If you are building your project on top of the labs then you cannot claim existing design patterns. You must include your own application of any two design patterns).

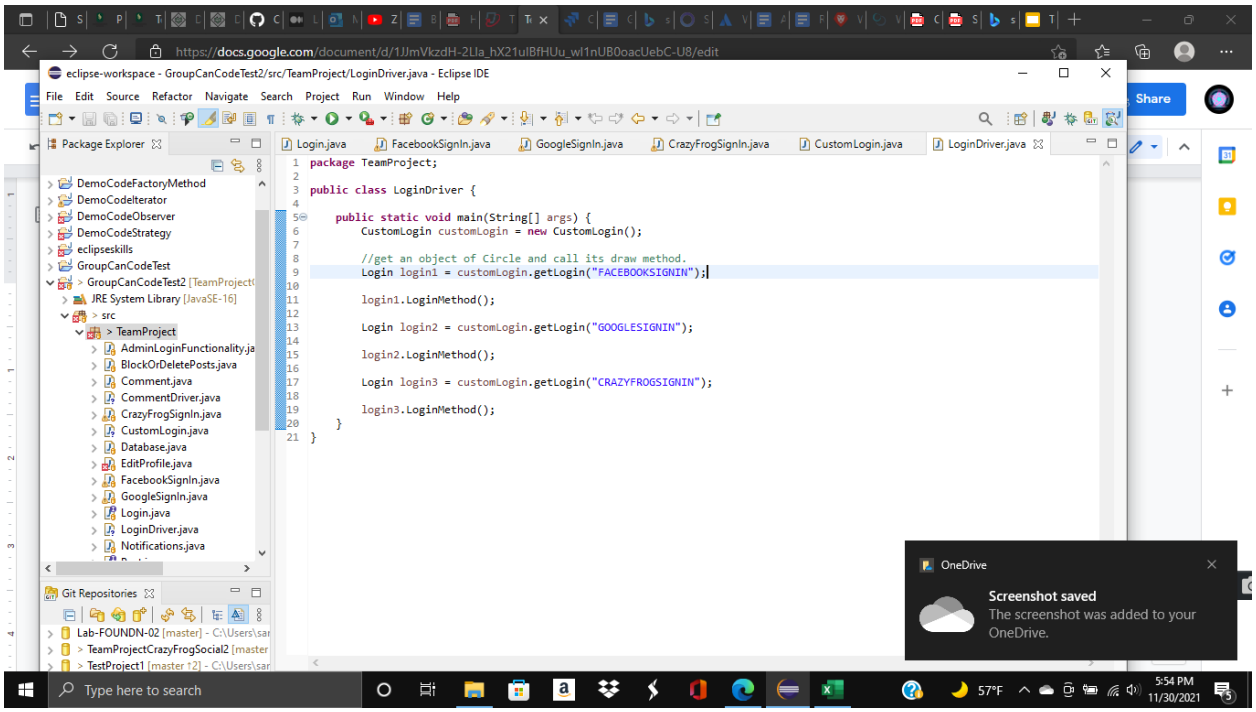
Design Pattern A: (10 points)

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)

driver.java | Observer.java | Subject.java | User.java | UserUpdatesFollowers.java

```java
//Making it so the user's followers act as observers to user (updates them when a new post is made)
package Dane;
import User.User;
public class driver{
    public static void main(String[] args) {
        User user1 = new User( userName: "jack",  password: "dskfsd",  email: "jack@outlook.com",  phoneNumber: "915534232");
        User user2 = new User( userName: "daneski",  password: "hfdsjoj432",  email: "dane@hotmail.com",  phoneNumber: "432234432");
        User user3 = new User( userName: "willy wonka",  password: "kohdsrjhjko",  email: "willy@gmail.com",  phoneNumber: "213321321"

        user1.followUser(user2);
        user3.followUser(user2);
        user2.makePost("hello everyone, check this out!"); //notifies user1&user3 of daneski's post(user2)

        user2.followUser(user1);
        user3.followUser(user1);
        user1.makePost("This is my cool post!"); //notifies user2&user3 of jack's post(user1)


    }


}
```

Run - TeamProjectCrazyFrogSocial

Run: driver

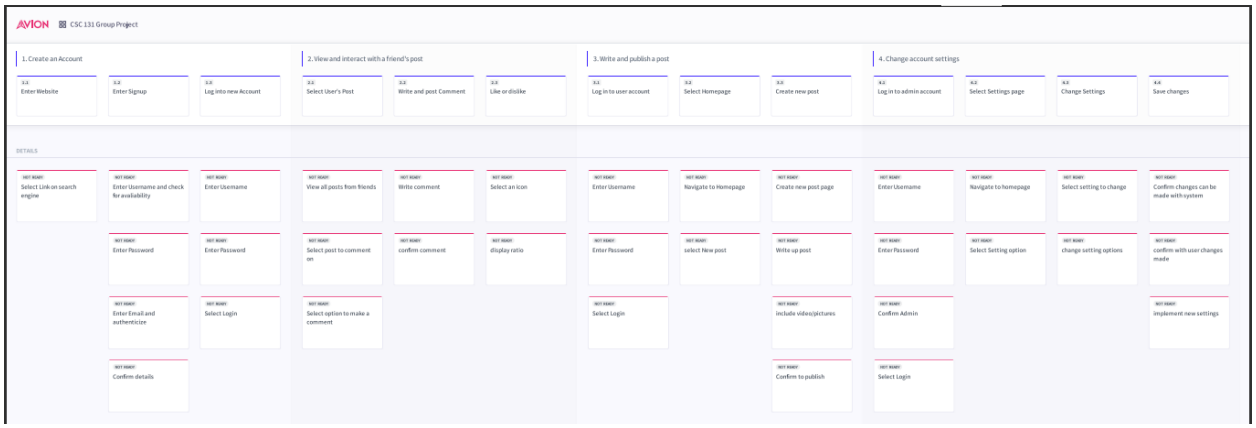
```
C:\Users\DaneC\.jdks\openjdk-17.0.1\bin\java.exe "-javaagent:C:\Program
daneski made new post titled: hello everyone, check this out!
daneski made new post titled: hello everyone, check this out!
jack made new post titled: This is my cool post!
jack made new post titled: This is my cool post!

Process finished with exit code 0
```

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

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

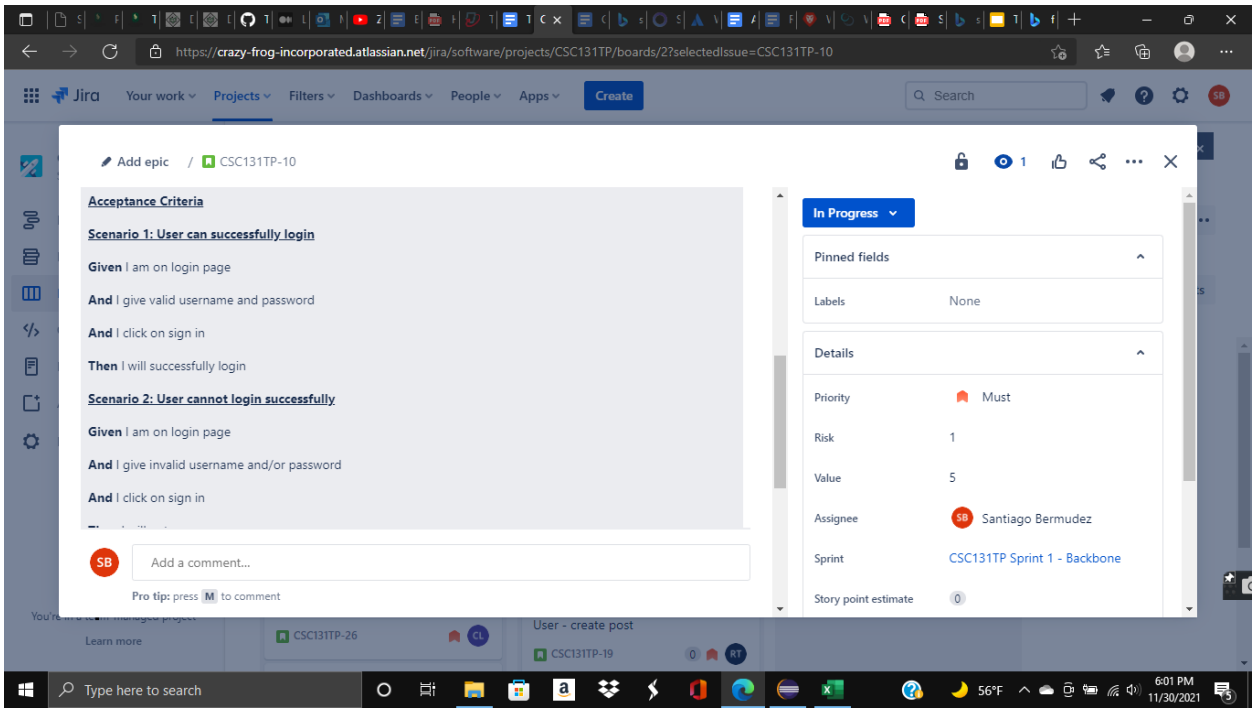
***We did MOSCOW principles, but we didn't do INVEST. We just did what we could. I don't know if we can get some extra credit for parts B and C.**

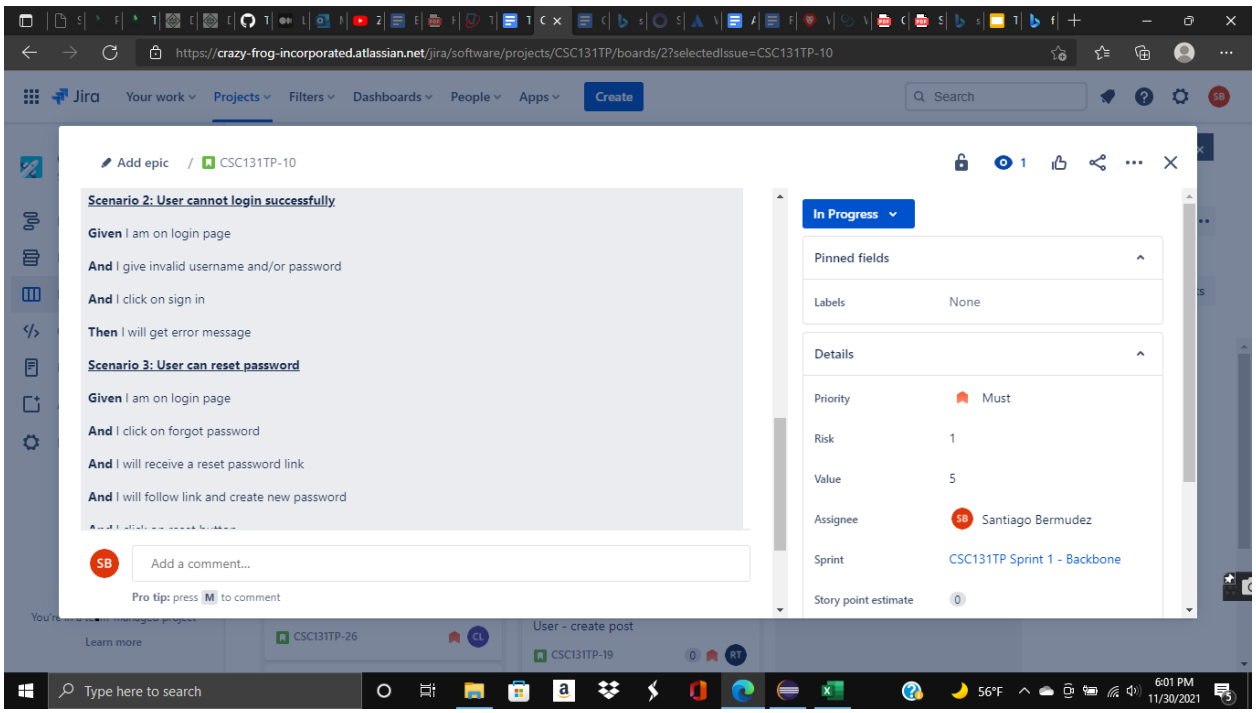
(b) Provide User Story Map.

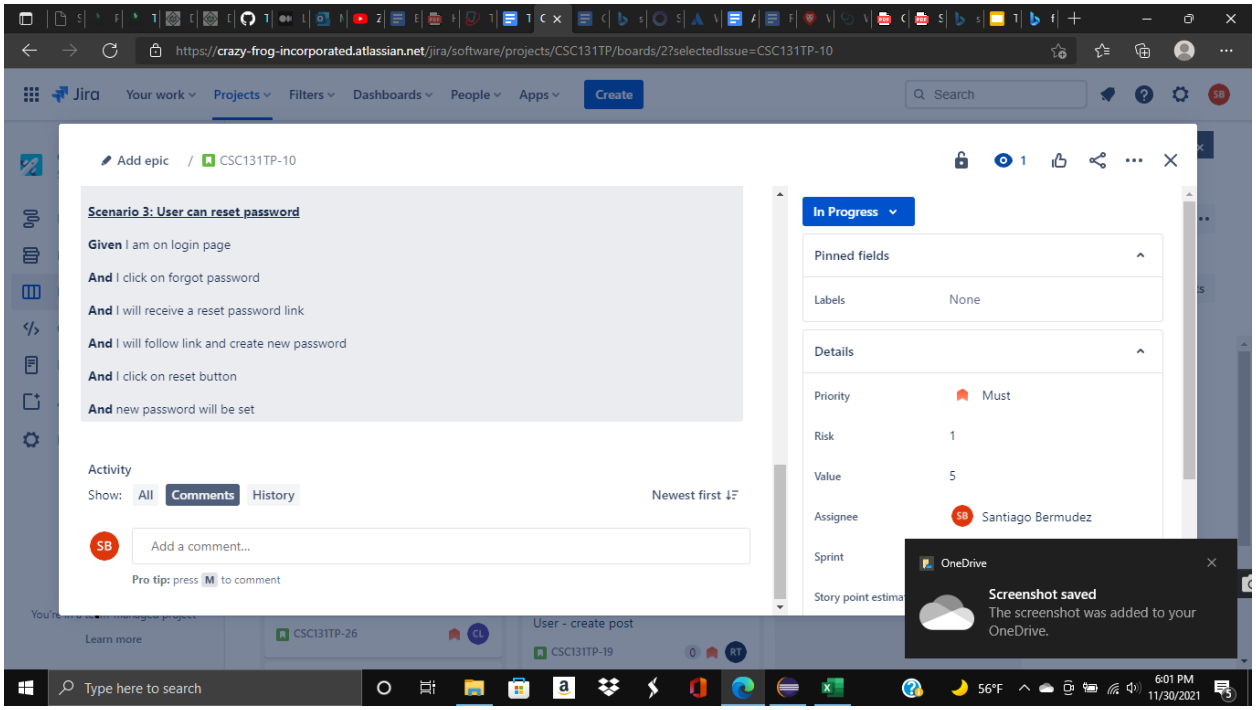


(c) Include Definition of Done with Acceptance Criteria.

*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.





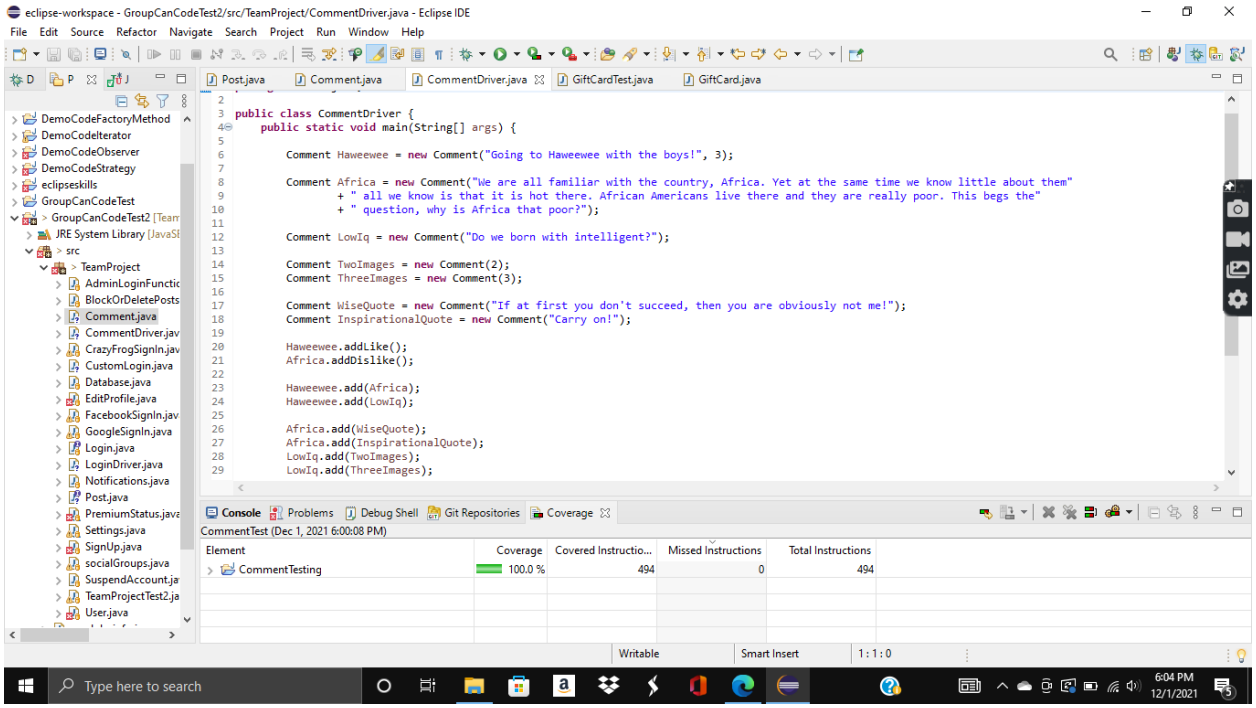


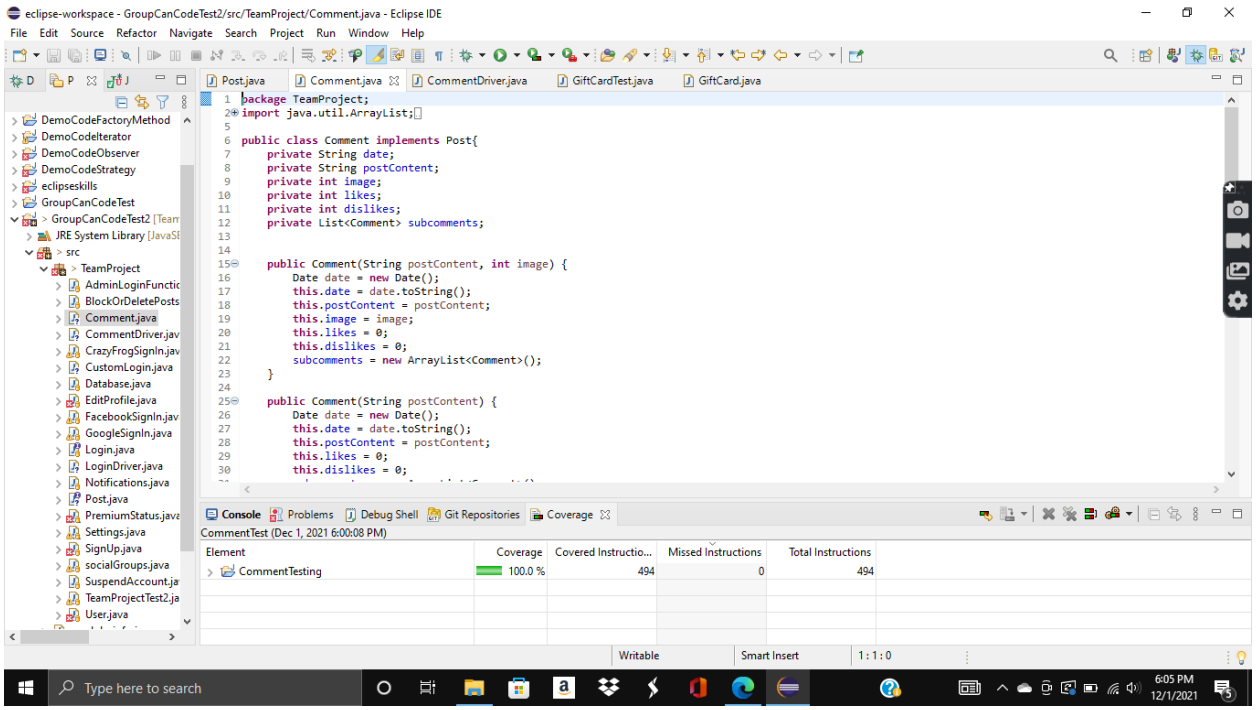
2. Bonus Points for use of additional 2 Design Patterns in Code relevant to your problem:

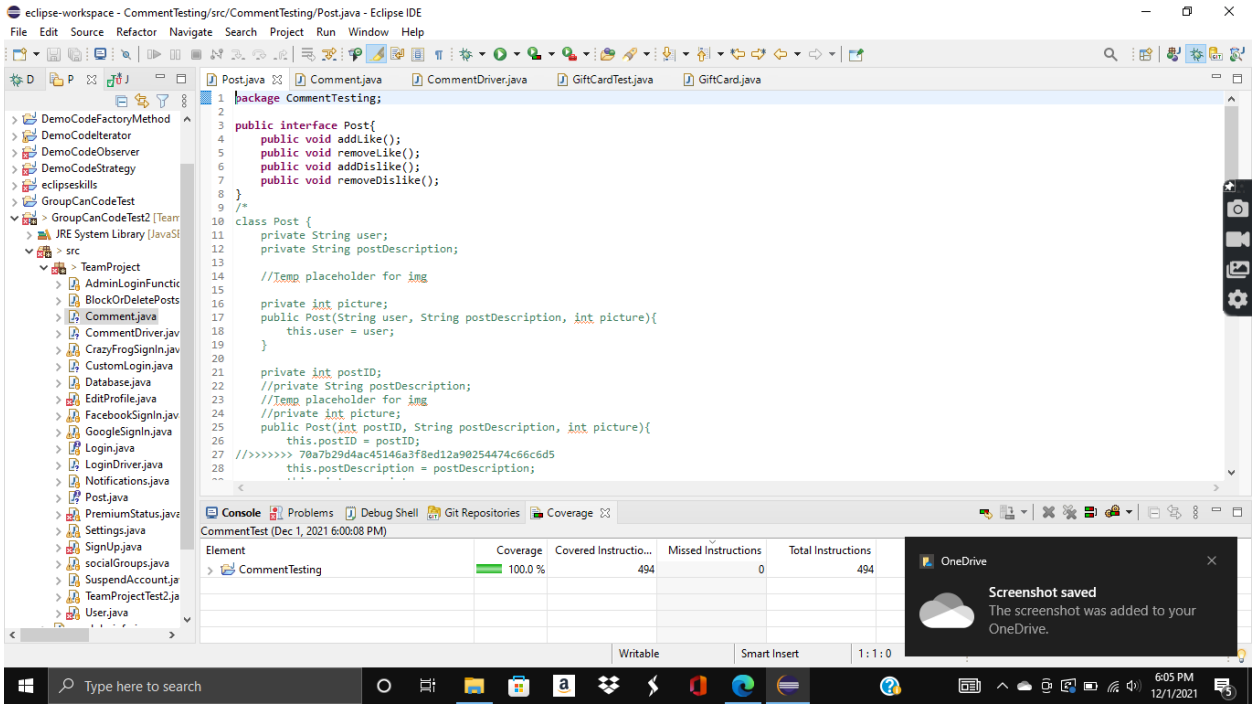
--->(5 bonus points)

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)


eclipse-workspace - GroupCanCodeTest2/src/TeamProject/CommentDriver.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Post.java
Comment.java
CommentDriver.java
GiftCardTest.java
GiftCard.java
public class CommentDriver {
public static void main(String[] args) {
Comment Haweewee = new Comment("Going to Haweewee with the boys!", 3);
Comment Africa = new Comment("We are all familiar with the country, Africa. Yet at the same time we know little about them"
+ " all we know is that it is hot there. African Americans live there and they are really poor. This begs the"
+ " question, why is Africa that poor?");
Comment LowIq = new Comment("Do we born with intelligent?");
Comment TwoImages = new Comment(2);
Comment ThreeImages = new Comment(3);
Comment WiseQuote = new Comment("If at first you don't succeed, then you are obviously not me!");
Comment InspirationalQuote = new Comment("Carry on!");
Haweewee.addLike();
Africa.addDislike();
Haweewee.add(Africa);
Haweewee.add(LowIq);
Africa.add(WiseQuote);
Africa.add(InspirationalQuote);
LowIq.add(TwoImages);
LowIq.add(ThreeImages);
Console Problems Debug Shell Git Repositories Coverage
CommentTest (Dec 1, 2021 6:00:08 PM)
Element Coverage Covered Instructio... Missed Instructions Total Instructions
CommentTesting 100.0 % 494 0 494
Writable Smart Insert 1 : 1 : 0
Type here to search
6:04 PM
12/1/2021


eclipse-workspace - GroupCanCodeTest2/src/TeamProject/Comment.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Post.java
Comment.java
CommentDriver.java
GiftCardTest.java
GiftCard.java
package TeamProject;
import java.util.ArrayList;
public class Comment implements Post{
private String date;
private String postContent;
private int image;
private int likes;
private int dislikes;
private List<Comment> subcomments;
public Comment(String postContent, int image) {
Date date = new Date();
this.date = date.toString();
this.postContent = postContent;
this.image = image;
this.likes = 0;
this.dislikes = 0;
subcomments = new ArrayList<Comment>();
}
public Comment(String postContent) {
Date date = new Date();
this.date = date.toString();
this.postContent = postContent;
this.likes = 0;
this.dislikes = 0;
Console Problems Debug Shell Git Repositories Coverage
CommentTest (Dec 1, 2021 6:00:08 PM)
Element Coverage Covered Instructio... Missed Instructions Total Instructions
CommentTesting 100.0 % 494 0 494
Writable Smart Insert 1 : 1 : 0
Type here to search
6:05 PM
12/1/2021



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!**