Carl-Michel Fevrius, Harry Gebremedhin, Brenden Judson

Professor Jane Cleland-Huang

CSE 40793 - Software Development Practices

December 12th 2018

Project overview

For this project we were tasked with designing and implementing a collaborative SCRUM tool that could keep track of User Stories and contained an interactive scrum board with multiple different lanes to track the state of the project. The SCRUM tool was meant to use the client-server model so that multiple users can modify information simultaneously. The tool was also tasked with being able to save the internal state using Java Serialization

Three user stories

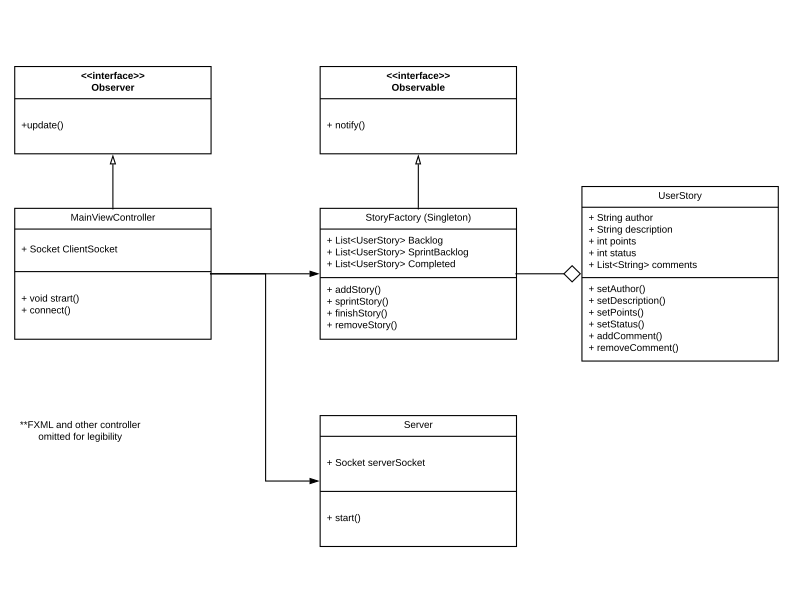
1. John is a senior engine programmer for a well known video game company and he is looking for a new web application that can help the artists and programmers collaborate better on future projects. He has been working in the industry for a long time and the company has developed a specific culture over the years that he wants to continue with the addition of any new tools. John wants the web app to be 1) highly customizable, with the ability to 2) backup files on a database and 3) edit files through an online interface. The web app should also be written in a language that has support for concurrency, as the the system should be able to do many asset processing in the background.
2. Katie is a senior tech lead at a booming startup. She is looking for a SCRUM based task management application to help her team become more efficient and organized with their work. She wants the application to have user friendly design with good guidance, intuitive organization, persistence of the state . It also needs to have minimal to no errors as they need to get their product to market as soon as possible.
3. Jamie is a web developer that does freelance work. Jamie needs a way to manage the tasks he has to work on . He needs to update his remote team on progress made on different tasks and be able to create tasks and assign it to different team members. He also wants the ability to comment on progress made by his teammates.

Architectural design (a diagram and a discussion of design decisions)

The architectural design and the different design patterns we implemented are below:

1. Factory: We want a StoryFactory class that is responsible for managing all user stories and has the data structures instantiated to hold the stories that are in different stages.
2. Singleton: We want there to be only one story factory and we want multiple other classes to access it.
3. Observer: We will also make the Story factory observable so when the logs change the UI will update.
4. Client/Server: We will build a client server model in order to broadcast one centralized scrum board across several clients(maximum < 10).

Diagram



Test plan and results (Unit tests, acceptance tests). Acceptance tests that evaluate your user stories.

We wrote some unit and integration tests in order to test the different components that make up our system. We used some unit tests to test our smaller class level components and wrote integration tests that involved testing these classes working together. We also ran the following acceptance tests manually as a way of assessing the readiness of our code as compared to our user story requirements.

**Acceptance Test 01 (Ran by Harry Gebremedhin 12/14/2018)**

* This acceptance test was ran to see if our serializable feature was functioning properly and saved the state of our gui.
* Started the program
* Added a new user story
* Moved some user stories to completion log
* After observing the state our scrum board was in
* Closed our application and restarted it
* Verified that our state matched the state previous to exit
* Test Passed

**Acceptance Test 02 (Ran by Harry Gebremedhin 12/14/2018)**

* This acceptance test was ran to see if our scrum gui correctly responded to user actions and gave the proper messages if illegal actions were being attempted.
* Started the program
* Added a new user story
* Moved the added user story across the board
* Moved other user stories that were loaded from serialized file on launch
* Edited some details of the user stories already present
* When updating time and move things out of completion we see changes in burndown chart
* GUI buttons such as add, remove and details show alerts when items are not selected
* Test Passed

**Acceptance Test 03 (Ran by Harry Gebremedhin 12/14/2018)**

* This acceptance test was ran to see the commenting feature which allows us to add comments to a certain task
* Started the program
* Clicked on an entry from the sprint backlog
* And clicked on details
* Using the bottom add comment button
* Prompt with text box appears and allows editing
* After typing in comment , we click save
* Our comment gets added to comments list view
* Test Passed