Team Portfolio 8.D

Maria Beatriz Zanardo, Daniel Guerra
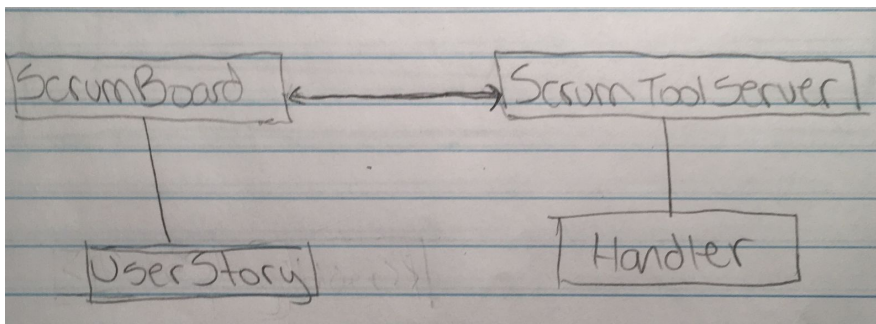
Project overview

The purpose of this project was to create a tool for the SCRUM agile framework. One of the main functionalities of this tool is the ability to be used by multiple users simultaneously and implementing shared memory among all users. The tool built by our team consists of a SCRUM board in which users can create user stories, add them to the current sprint, and change their status as the project progresses. This can be easily done by users by simply dragging and dropping the user stories to different lanes, each of which represents the different statuses of the user stories (i.e not started, in progress, testing, done, etc.). The user also has the ability to edit, add, delete, or see additional details about a given story.

User Stories

1. User 1 wants to add a new user story to the backlog and wants all the other users to be able to see it as soon as it is added.
2. User 2 wants to change the status of a given user story from "Backlog" to "Not Started" and wants all the user to have the updated status of this story.
3. User 3 wants to close the server and the running version of the ScrumBoard without losing the information that is currently stored in it.
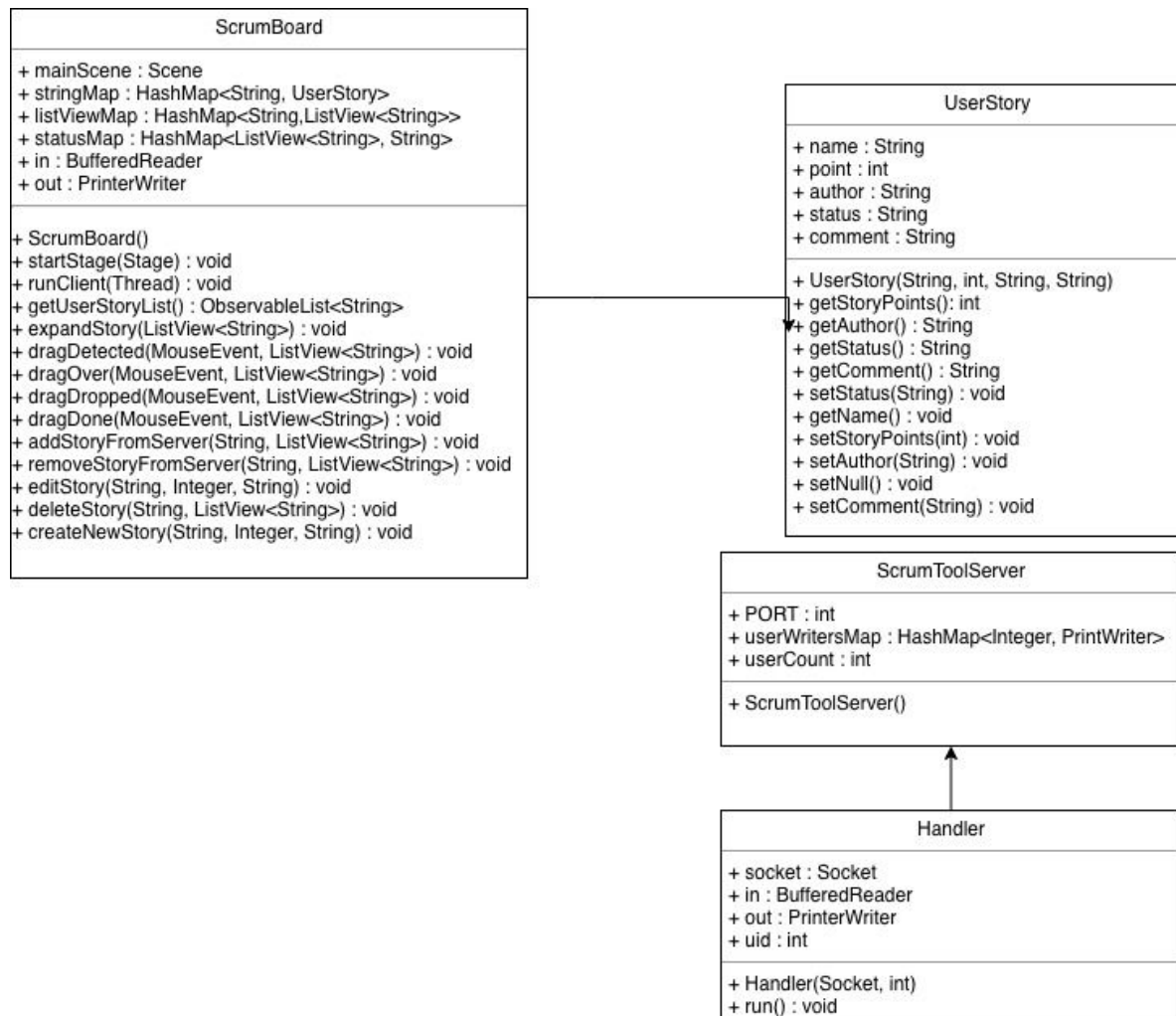
Architectural Design

The ScrumToolServer class is responsible for running the server at a given socket, to which multiple clients can connect. This class also uses the Handler class, which is responsible

for listening to a single client and broadcasting its messages. Each instance of the ScrumBoard class serves as a client for the server. The ScrumBoard class also is responsible for drawing the UI, recognizing when a user changes a story, and broadcasting the changes to the server. This class also receives messages from the server whenever another user changes a story in another instance of the ScrumBoard class and updates the UI accordingly. The UserStory class contains the information for each user story (such as author, story points, status, etc.). The ScrumBoard class has a HashMap of all updated UserStories saved in the program.

Class Diagrams

ScrumBoard
+ mainScene : Scene + stringMap : HashMap<String, UserStory> + listViewMap : HashMap<String,ListView<String>> + statusMap : HashMap<ListView<String>, String> + in : BufferedReader + out : PrinterWriter
+ ScrumBoard() + startStage(Stage) : void + runClient(Thread) : void + getUserStoryList() : ObservableList<String> + expandStory(ListView<String>) : void + dragDetected(MouseEvent, ListView<String>) : void + dragOver(MouseEvent, ListView<String>) : void + dragDropped(MouseEvent, ListView<String>) : void + dragDone(MouseEvent, ListView<String>) : void + addStoryFromServer(String, ListView<String>) : void + removeStoryFromServer(String, ListView<String>) : void + editStory(String, Integer, String) : void + deleteStory(String, ListView<String>) : void + createNewStory(String, Integer, String) : void

UserStory
+ name : String + point : int + author : String + status : String + comment : String
+ UserStory(String, int, String, String) + getStoryPoints(): int + getAuthor() : String + getStatus() : String + getComment() : String + setStatus(String) : void + getName() : void + setStoryPoints(int) : void + setAuthor(String) : void + setNull() : void + setComment(String) : void

ScrumToolServer
+ PORT : int + userWritersMap : HashMap<Integer, PrintWriter> + userCount : int
+ ScrumToolServer()

Handler
+ socket : Socket + in : BufferedReader + out : PrinterWriter + uid : int
+ Handler(Socket, int) + run() : void

Test Plan and Results

Acceptance Tests:

1.

    a. Pre-conditions: ScrumToolServer is running, two different users are connected to the server and seeing the same Scrum Board.

    b. Steps:

        i. User 1 clicks the "Add New Story" button.

        ii. User 1 fills out the "Add New User Story" form, setting the story name to "Integration Test 1", author to "User 1" and story points to 2, and then clicks the "Create New Story" button.

    c. Post-conditions: The "Integration Test" story appears in both User 1 and User 2 "Backlog" lane and when User 2 clicks on the story, all the information matches the information entered by User 1 in step ii.

    d. Result: Passed

2.

    a. Pre-conditions: ScrumToolServer is running, two different users are connected to the server and seeing the same Scrum Board.

    b. Steps

        i. User 2 drags the "Integration Test" story (created in previous test) from the "Backlog" lane to the "Not Started" lane.

    c. Post-conditions: The "Integration Test" story appears in both User 1 and User 2 "Not Started" lane.

    d. Result: Passed

3.

    a. Pre-conditions: ScrumToolServer is running, two different users are connected to the server and seeing the same Scrum Board.

    b. Steps:

        i. User 1 clicks on the "Integration Test" story.

        ii. The text box on the bottom of the UI get populated with "Integration Test" story's information.

        iii. User 1 clicks the "Edit Story" button, and changes the story points filed from 2 to 3 and clicks save.

    iv. User 2 clicks on the "Integration Test" story

  c. Post-conditions: The text box on User 2's UI shows the information about the "Integration Test" story, and it shows that the story points are equal to 3.

  d. Result: Passed

4.

  a. Pre-conditions: ScrumToolServer is running, one user is connected to the server.

  b. Steps:

    i. The user creates a story following the same steps as test 1.

    ii. User closes the UI and stops the server.

    iii. User restarts the server and reopens the Scrum Board.

  c. Post-conditions: The "Integration Test" story is still present in the "Backlog" lane.

  d. Result: Passed