Sprint Plan #1

SEG4105 Group Project - G03

Note: most of the information presented in this document can be found in our Jira. In particular, the information **NOT** tracked in our Jira includes *Dependencies and Risks*, some parts of our *Definition of Done*, and our *Review and Approval* process.

Sprint Goal

Completion of frontend skeleton (search page, list of results, and preliminary movie details page) and search function.

Sprint Backlog

Below is a list of user stories and tasks scheduled for completion during this sprint.

- SCRUM-2: As a user, I want to search for movies by title so that I can find information on the movie I'm looking for.
- SCRUM-8: As a user, I want to see the title of the selected movie.
- SCRUM-3: As a user, I want to search for movies by genre so that I can see other movies within the same genre.
- SCRUM-10: As a user, I want to see the official synopsis of the selected movie so that I know what it's about.
- SCRUM-4: As a user, I want to search for movies by rating so that I can see the best rated movies.
- SCRUM-9: As a user, I want to see the poster of the selected movie so that I can see what it looks like.
- SCRUM-19: Implement search by Title API function
- SCRUM-20: Implement a search bar for the website
- SCRUM-21: Implement search by Genre API function
- SCRUM-25: Implement display page for selected movie
- SCRUM-28: Implement search by Rating API function
- SCRUM-29: Implement dropdown menu selection to select method of search.
- SCRUM-33: Implement a search results section for the website
- SCRUM-36: Implement movie title extraction from data received from API
- SCRUM-37: Implement movie synopsis extraction from data received from API
- SCRUM-38: Implement movie poster extraction from data received from API

Task Breakdown

Below is a list detailing how each user story is broken down into tasks.

- SCRUM-2
  - SCRUM-19
  - SCRUM-20
  - SCRUM-29
  - SCRUM-30
- SCRUM-8
  - SCRUM-25
  - SCRUM-36
- SCRUM-3
  - SCRUM-21
  - SCRUM-20
  - SCRUM-29
  - SCRUM-30
- SCRUM-10
  - SCRUM-25
  - SCRUM-37
- SCRUM-4
  - SCRUM-28
  - SCRUM-20
  - SCRUM-29
  - SCRUM-30
- SCRUM-9
  - SCRUM-25
  - SCRUM-38

Effort Estimates

Each story and task has been assigned an amount of story points on Jira. Each point is equal to one day of work. Story points are assigned at the task level, with the story points of user stories being the sum of story points of the tasks required for completion of the user story. User stories and tasks have been selected such that each developer is working to complete about 7 story points (days) worth of tasks for the sprint.

User Stories

- SCRUM-2: 7 Story Points
- SCRUM-8: 2.5 Story Points
- SCRUM-3: 5 Story Points
- SCRUM-10: 2.5 Story Points
- SCRUM-4: 6 Story Points
- SCRUM-9: 2.5 Story Points

Tasks

- SCRUM-19: 3 Story Points
- SCRUM-20: 2 Story Points
- SCRUM-21: 1 Story Point
- SCRUM-25: 2 Story Points
- SCRUM-28: 2 Story Points
- SCRUM-29: 1 Story Point
- SCRUM-33: 1 Story Point
- SCRUM-36: 0.5 Story Points
- SCRUM-37: 0.5 Story Points
- SCRUM-38: 0.5 Story Points

Task Assignments

Task assignments for each developer are shown below. Product Owner and Scrum Master do not have any outstanding tasks for this sprint and will be carrying on their responsibilities detailed in the Project Charter as usual.

Muizz Zafar: SCRUM-19, SCRUM-21, SCRUM-28, SCRUM-29
Azan Mubashar: SCRUM-20, SCRUM-25, SCRUM-33, SCRUM-36, SCRUM-37, SCRUM-38

Sprint Duration and Times

The sprint will be for 1 week, starting from October 20th, 12:00 AM and ends on October 27th, 2024, 12:00 AM.

Dependencies and Risks

Dependencies:

- **API Integration Dependency:**
    - SCRUM-19, SCRUM-21, SCRUM-28 rely on the functioning integration and implementation of the MoviesDatabase API functions for searching by title, genre, and rating. The frontend search functionality can't be fully implemented without the API responding with the data it needs.

- **UI Design Dependency:**
    - The implementation of SCRUM-20 (Search Bar) and SCRUM-25 (Display Page) depends on having a clear UI/UX design in place. Any discrepancies between teammates in the expected design could result in non-functional features.

- **Team Collaboration Dependency:**
    - Tasks are split between the team members, and progress on interdependent tasks needs regular communication. For example, Azan's work on the search bar (SCRUM-20) relies on Muiz's implementation of the search API functions (SCRUM-19, SCRUM-21, SCRUM-28).

Risks:

1. **API Connectivity Issues**
    a. **Impact**
        i. High
    b. **Likelihood**
        i. Medium
    c. **Description:**
        i. If the MoviesDatabase API faces connectivity issues and is not able to function, it would prevent tasks SCRUM-19, SCRUM-21, and SCRUM-28 from being implemented.
    d. **Mitigation Strategy**
        i. Ensure API Endpoint functionality with prior testing and potentially implement smaller database that is not dependent on the API
2. **Miscommunication in UI Design and Development**
    a. **Impact**
        i. Medium
    b. **Likelihood**
        i. Low
    c. **Description**
        i. Differences between the developers in what the UX/UI design is supposed to look like may cause issues with integrating front end components like the search bar not being functional
    d. **Mitigation Strategy**
        i. Schedule meetings for a design review among developers, so that everyone is clear on how each component will look and be integrated to ensure alignment.
3. **Time Constraints for Sprint**
    a. **Impact**
        i. High
    b. **Likelihood**
        i. Medium
    c. **Description**
        i. Considering a one week sprint and the amount of story points allocated to each task, it is possible that there may be unfinished tasks due to unforeseen complications.
    d. **Mitigation Strategy**
        i. It is effective to work on tasks based on higher priority to ensure that the most crucial user stories are completed to meet the sprint goal.

Definition of Done

For each user story:

- All tests are passed
- Code is reviewed and approved
- Results meets acceptance criteria

Acceptance Criteria for each user story and task can be found below.

User Stories

- SCRUM-2: User can input a string in some manner, after which the user is given a list of movies with titles containing the input string.
- SCRUM-8: When user selects a movie in some manner, the title of the movie is displayed in some manner visible to the user.
- SCRUM-3: User is able to input a desired genre in some manner, after which a list of movies of the inputted genre is displayed to the user.
- SCRUM-10: When user selects a movie in some manner, the official synopsis of the movie is displayed in some manner visible to the user.
- SCRUM-4: User can input a rating value, after which a list of movies with ratings equal to or greater than the input rating are returned.
- SCRUM-9: When user selects a movie in some manner, the poster of the movie is displayed in some manner visible to the user.

Tasks

- SCRUM-19: Function that takes an input string and returns a list of movies with titles containing the input string added.
- SCRUM-20: A search bar input is added somewhere on the site.
- SCRUM-21: Function that takes in an input string (genre) and returns a list of movies of the input genre is added.
- SCRUM-25:
    - Page that displays details of a movie added.
    - Code to load a selected movie and its details extracted from API response into the aforementioned page added.
- SCRUM-28: Function that takes a float (minimum rating) and returns a list of movies with ratings above the input minimum rating added.
- SCRUM-29: Dropdown to select method of search is added.
- SCRUM-33: An element to contain the list of movies returned by a search is found somewhere on the site.
- SCRUM-36: Code to get and display movie title from the received API response is added.

- SCRUM-37: Code to get and display movie synopsis from the received API response is added.
- SCRUM-38: Code to get and display the movie poster from the received API response is added.

Review and Approval

The sprint results will be reviewed towards the end of the sprint, on Friday, October 25th. This review will focus on the user stories and tasks of the sprint, and involve both Product Owner and stakeholders. For stakeholders, we will be presenting our results to Shabnam, acting on behalf of Professor Hamou-Lhadj.

The review process will involve a simple demonstration of our potentially shippable product of the sprint, and how it meets the acceptance criteria (definition of done) outlined by the sprint plan. The Product Owner and stakeholder will then express approval or disapproval over the results. Feedback gathered from Product Owner or stakeholder will be recorded for the next sprint.

The sprint retrospective and next sprint backlog planning will be conducted on Saturday, October 26th. These will involve a brief meeting during which the team will discuss three topics. First, how the sprint went process-wise and if adjustments need to be made. Second, what user stories and tasks will be added to the next sprint's backlog. Lastly, if there are any new user stories or tasks that need to be added to the product backlog, based on feedback obtained from the sprint review or required adjustments determined from the sprint retrospective.