SCRUM MEETING WEEK (11)

**:white_check_mark: Sprint planning checklist**

| **Preparation** | **Meeting** | **Follow up** |
| --- | --- | --- |
| ​​​​​​ Discussed previous tasks and division of this week's tasks. | ​​During the lab section. | ​​Spoke about everyone finishing their tasks from the previous week. Mentioned goals for this week's sprint. |

**:busts_in_silhouette: Sprint team members**

| **Name** | **Role** |
| --- | --- |
| ​​ Oliver Medgyesi | ​​ |
| Mariya Putwa |  |
| Stefan Spataro | ticket selection |
| Rory Hazlett | Checkout Back end |
| Prateek Balani | confirmation page ui |
|  |  |

** Sprint planning meeting items**

**Previous sprint summary**

| **Sprint theme** | ​​UI/Backend |
| --- | --- |
| **Issues completed** | ​​ 11 |
| **Issues left** | 0 |
| **Team Capacity** | 5 |
| **Summary** | ​​ All issues were completed. |

**Details Current sprint**

| **Start date** | ​​ 2024/03/18 |
| --- | --- |
| **End date** | 2024/03/25 |
| **Sprint theme** | ​​ UI/Backend |
| **Team capacity** | 5 |
| **Issues capacity** | > 20 |
| **Individual capacity** | **Oliver Medgyesi:**   1. Created Review UI 2. Count and display Reviews based on Figure 3. Dynamically display the converted User Rating to a Star rating w/ "starRange" python method. 4. Dynamically display Reviews with images on top row 5. Dynamically display Reviews with no images on bottom row 6. Updated Review model to associated with Figure instead of event 7. Updated test\_models to 100% coverage 8. Wrote test case for Figure View to 100% Coverage 9. Create "Write a Review" UI 10. Reviews are set to correspond with specific figures 11. - /review/drake/ will return a review for "Drake" 12. - /review/foo%20fighters/ will return a review for "Foo Fighters" 13. Users Select Star Rating -> Converted to reviewRating (Integer) 14. Wrote JS to Dynamically highlight stars (3rd star highlighted, all previous will highlight) 15. Users can enter custom reviewTitle, reviewDescription, reviewDate 16. Users have the option to upload or not upload images. 17. Review Form Validiation for Inputs 18. Homepage to only display events w/ Images 19. Wrote TESTS for Review View (100% Coverage) 20. Created a pop-up for REDIRECTION to homepage POST review 21. Created Photo Gallery UI 22. Dynamically Adds Photos from Reviews w/ Images to the Photo Gallery 23. Dynamically Add newly created artists to the pool 24. Dynamically Add artists with similar query strings to the pool 25. Dynamically display Related Figures (Based on Genre) under "RELATED FIGURES" 26. If user searched "Drake" (Genre = Hip-Hop), Related = "Genre" displays other hip-hop artists... 27. User selection redirects to Figure page (Figure selected, we return /figure/"figureName") 28. Dynamically display and sort Figures based on Query ->   - Example: Frank Ocean and Frank Sinatraa, If a user searches "Frank", we display both.  - Use the first initialFigure = searchedFigures[0] for the base of our Related Figures   1. Related figures will display according to figureGenre, Frank = Pop, all Related = Pop 2. Adjusted Navbar to redirect to "home" on logo press 3. Dynamically Add events to the search pool 4. Dynamically display events related to the initialFigure 5. User searches "Drake" events will display for "Drake" 6. Display the amount of available events + corresponding figure 7. Search Functionality: Handling ONLY KEYWORD (DONE) 8. User searches by keyword: Example, we have two figures "Frank Ocean" and "Frank Sinatraa" 9. Display both figures in "Searched Figure" section as they both contain "Frank" 10. Related Figures will display based on initialFigure's Genre, so in this case Frank Ocean is first and genre=Pop so other "Pop" Artists 11. ONLY Search by keyword should show searched figures, rest should display related figures. 12. Search Functionality: Handling ONLY DATE (DONE) 13. User searches by only date: Displays events on that date + Related Figures for the events 14. Search Functionality: Handling Only City (DONE) 15. Display events based on Event Location, Display Related Figures to these events. 16. Search Functionality: Handle City + Date (DONE) 17. Display events only on the EXACT DATE INPUT + Related Figures for those exact events 18. Search Functionality: Handle City + Date + Keyword 19. Display EXACT Events for the EXACT Date for the EXACT Figure  * i.e City: Vancouver Date: 2024-05-30 Keyword: Drake  1. Displays events for this exact city, date, and person 2. Will display other events for other figures if they are also on the same date 3. Search Functionality: Handle Date + Keyword 4. Display events only on the EXACT DATE INPUT for the Figure Searched 5. - i.e User puts in 2024-05-30 "Drake", ONLY Drake Concerts show for this date 6. Empty Searched Figure to only show specific EVENT for specific FIGURE 7. Search Functionality: Handle City + Keyword 8. Display events ONLY related to the specific CITY and the Related Figures to these events 9. Empty Searched Figure to only show the related figures to the events. 10. Dynamically Display Carousel Images 11. Dynamically Display Figures for (HipHop, Pop, and Basketball) 12. - Removed all static data on the homepage. 13. Wrote TEST to cover aspects in search\_results VIEW (97% Coverage): 14. TEST: Handle City + Date + Keyword 15. TEST: Handle City + Date 16. TEST: Handle City + Keyword 17. TEST: Handle Date + Keyword 18. TEST: Handling Only City 19. TEST: Handling ONLY DATE 20. TEST: Handling ONLY KEYWORD 21. TEST: Handle NO INPUTS   **Mariya Putwa:**   * Navbar UI: Drop-down menu for profile page, logout and creating event functionality. * Tests for Organiser views. * Profile page functionality. * Edit profile page button functionality. * Enhance UI for login/reg pages.   **Prateek Balani:**   * Confirmation: Created the page * Confirmation: Added Ui * Confirmation: added test for views * Confirmation: added test for url   **Stefan Spataro:**   * Ticket Selection: Create UI * Add UI: arena, concert, etc. * Ticket Selection: Create sidebar to display selected seats * Ticket Selection: Checkout re-direction * Ticket Selection: Link to Event “find tickets” * Ticket Selection: Dynamic markup for selected seats   **Rory Hazlett:**   * Fix Register UI * Logout Functionality * Login Functionality * Make test for Login/Logout Views * Checkout UI |
| **Potential risks** | Not completing issues |
| **Mitigations** | Communication |