|  |  |  |
| --- | --- | --- |
|  | | |
|  | **CMPS 350 Project Phase 1 – WebApp UI Design and Implementation**  **Conference Management System (ConfPlus)**  **(20% of the course grade)** | |
| **Group Id: t26** | | G -L52 |
| **Group Members: 2** | | Student1 Dalal Ali Nayem (dn1704728)  Student2 Ghada Al Naimi (Ga1513753)  **Emails:** dn1704728@student.qu.edu.qa;  ga1513763 @student.qu.edu.qa; |

**Grading Rubric - In the Functionality column please specify either: *Working (completed x%)*, *Not Working (completed x%)* or *Not done*.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **%** | **Functionality\*** | **Quality of the implementation** | **Your Grade** |
| **Application Design:** Entities, Repositories and Web API class diagrams | 6% |  |  |  |
| **Complete and correct implementation of the requirements:** | **88%** |  |  |  |
| * Login | 6 |  |  |  |
| * Submit paper | 18 |  |  |  |
| * Review paper | 20 |  |  |  |
| * Create/update conference schedule | 35 |  |  |  |
| * Get conference schedule | 9 |  |  |  |
| **Testing documentation** using screen shots illustrating the testing results.  - Discussion of the project contribution of each team member. Members should collaborate and contribute equally to the project. | 6% |  |  |  |
| **Total** | 100 |  |  |  |
| Copying and/or plagiarism or not being able to explain or answer questions about the implementation | -100% |  |  |  |

**\* Possible grading for functionality** - ***Working*** (get 70% of the assigned grade), ***Not*** ***working*** (lose 40% of assigned grade and ***Not done*** (get 0). The remaining grade is assigned to the quality of the implementation.

In case your implementation is not working then 40% of the grade will be lost and the remaining 60% will be determined based on of the code quality and how close your solution to the working implementation.

Solution quality also includes meaningful naming of identifiers (according to Android naming conventions), no redundant code, simple and efficient design, clean implementation without unnecessary files/code, use of comments where necessary, proper code formatting and indentation.

**Marks will be reduced** forcode duplication, poor/inefficient coding practices, poor naming of identifiers, unclean/untidy submission, and unnecessary complex/poor user interface design.

# Application Design

# Entities class diagram

Entities are data (json files) in the path "data".

# Repositories class diagram

Repositories are js files in the path "pages/api", where every repository access to appropriate data by reading/writing the related json file then processing some API methods by handler.

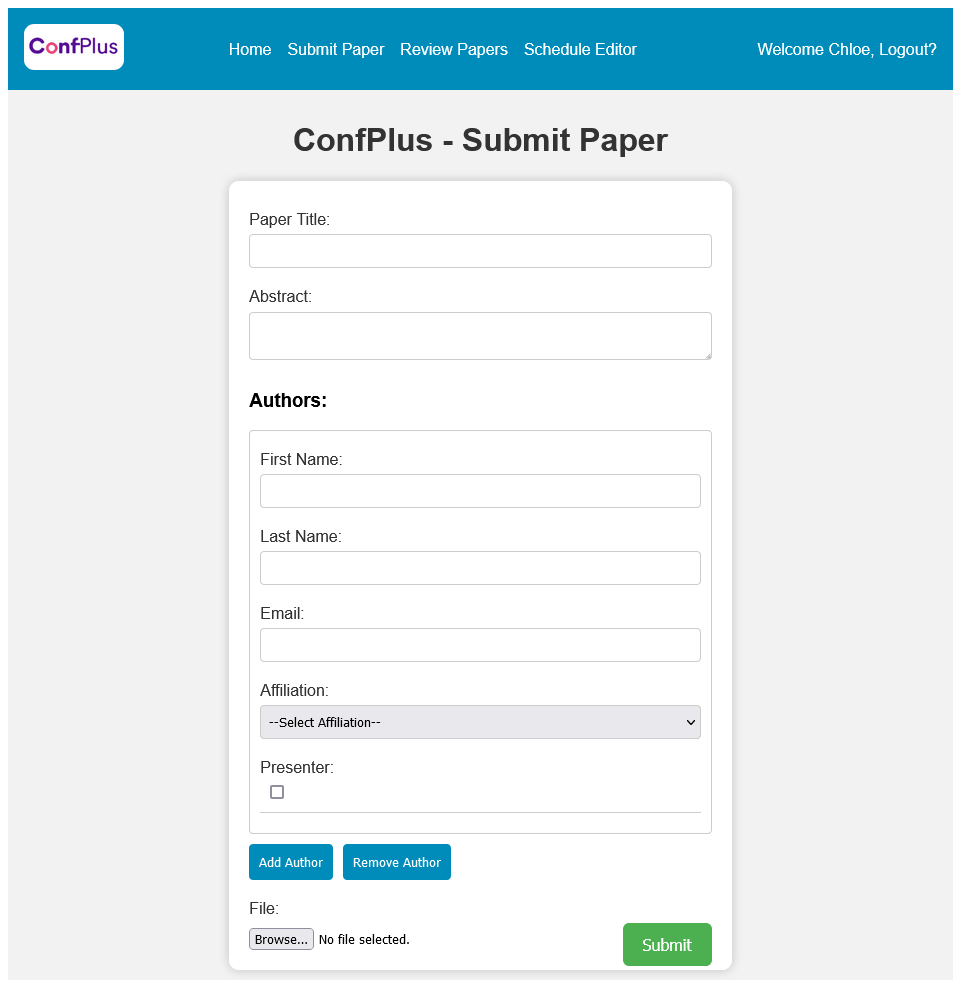
# Web API class diagram

* GET /api/conference-dates, for getting list of conference-dates.
* GET /api/institutions, for getting list of available institutions.
* GET /api/locations, for getting list of locations.
* POST /api/login, for validating credentials then login user.
* POST /api/papers, for saving paper with some details.
* GET /api/papers, for getting papers with strong accept.
* POST /api/schedule, to create/update conference schedule.
* GET /api/schedule, for getting the conference schedule.
* GET /api/schedule, for getting the conference schedule.
* GET /api/users?role=reviewer for getting reviewer users.
* GET /api/users for getting all users (option only for test).

# Testing

# Graphical user interface Description automatically generatedLogin

# Submit paper

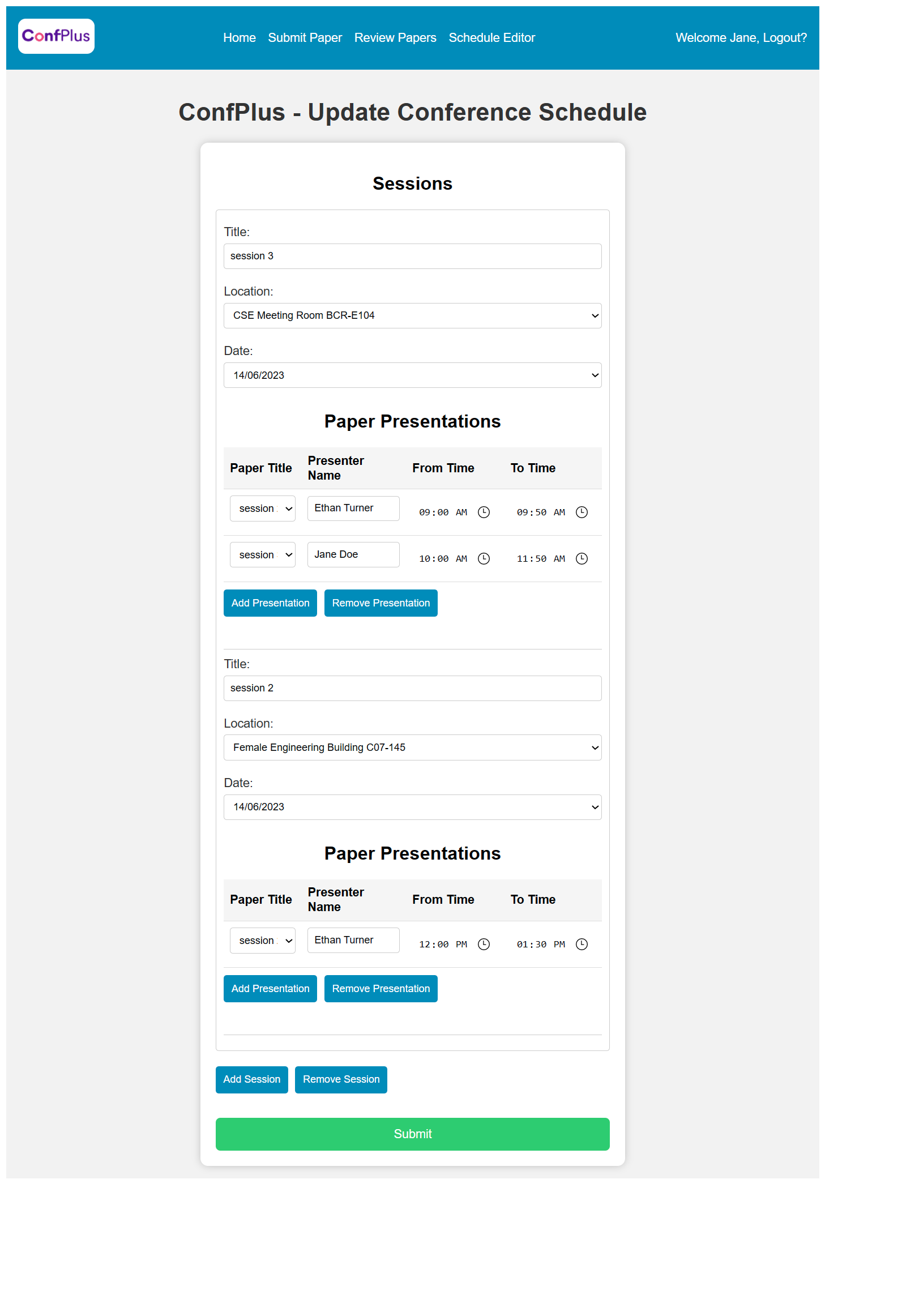


# Review paper

# Create/update conference schedule

Graphical user interface, application

Description automatically generated**Create:**

**update:**

# Get conference schedule

**Graphical user interface, text, application

Description automatically generatedafter flittering:**

# Discussion of the project contribution of each team member