CMPS 350 Project Phase 1 – WebApp UI Design and Implementation

Conference Management System (ConfPlus)

(15% of the course grade)

The project phase 1 submission is due by 8am Tuesday 21st March 2023. Demos during the same week

1. Requirements

You are required to design and implement a Conference Management System named ***ConfPlus*** to manage the process of organizing and running an academic conference. It should be designed to meet the needs of all stakeholders involved in the conference, including organizers, presenters, attendees, and reviewers. The app use cases include:

- Conference public website:

Use Case 1: Conference public website

A visitor can visit the conference public website to get information about:

- Conference information including the conference name, description, conference dates, venue, contact details.
- Call for papers including important dates (submission, decision, and camera-ready dates), formatting guidelines, and submission link.
- Program committee chairs and members.
- Registration packages and fees, and registration link.
- Conference Program to display the schedule.
- Sponsors details.
- Social events.

The public website should be generated from static json data (such as Program Committee members) or from data created by other app use cases (such as the conference schedule).

- Paper Submission Management: The system should allow authors to submit their papers and manage the review and acceptance process.

  ### Use Case 2: Submit a paper

  Precondition: User is logged in as an author has a paper to submit.

  - Author navigates to the paper submission page.
  - Author enters paper details, such as title and abstract.
  - Author uploads the paper and submits the paper for review.

  Postcondition: Paper is submitted for review.

- Review Management: The system should allow reviewers to access the submitted papers, review them, and provide feedback to the authors.

  ### Use Case 3: Assign papers to reviewers

  - Organizer can assign papers to reviewers.

  ### Use Case 4: Review paper

  Precondition: User is logged in as a reviewer and has been assigned a paper to review.

  - Reviewer navigates to the paper review page.

- Reviewer selects the paper to review and enter the review details.
- Reviewer submits the review.

Postcondition: Review is submitted to the system and can be accessed by the paper author and conference organizers.

Use Case 5: Get completed/pending paper reviews

- Organizer can get the completed/pending paper reviews then make accept and reject decisions.

Use case 6: View paper review

- Author can view the reviews and the decision of their submitted paper(s).

- **Registration Management**: The system should allow attendees to register for the conference and provide personal information, dietary preferences, and payment details.

Use Case 7: Register for conference

- Attendee navigates to the registration page.
- Attendee enters personal information, such as name, email, and affiliation.
- Attendee selects dietary preferences from a list of options
- Attendee selects conference package and payment method.
- Attendee enters payment details and completes registration.

Postcondition: Attendee is registered for the conference and payment is processed.

Use Case 8: Get list of pending/completed registrations

- Organizer can get of completed / pending paper registration.
- Organizer can send registration reminders to authors who did not register yet.

- **Conference Program**: The system should allow organizers to manage the conference schedule and agenda, including session timings, chairs, presenters, and locations.

Use Case 9: Create/update conference schedule

Precondition: User is logged in as an organizer and has access to paper and presenter information.

- Organizer navigates to the conference agenda page.
- Organizer creates sessions and assigns papers and presenters to each session.
- Organizer sets session timings and locations.
- Organizer publishes the conference schedule.

Postcondition: Conference schedule is created and published and can be accessed by attendees.

Use Case 10: View conference schedule

Visitor can view the conference schedule with the ability to filter by date.

- **Manage the conference Social Events**
  - Add/update/remove a social event.
  - Social Event registration.
  - Get the list of attendees of a social event.

2. Deliverables

Seek further clarification about the requirements/deliverables during the initial progress meeting with the instructor. Note that further important clarifications maybe modified/added to the project requirements.

1) Design the App Web UI and navigation.

   You may design the UI wireframe (sketch) to decide the UI components and the layout either on paper or use a design tool such as https://www.figma.com/
   During the weekly office hours, you are required to present and discuss your design with the instructor and get feedback.

2) For each use case, implement the app Web UI and navigation using HTML, CSS and JavaScript. The pages should comply with Web user interface design best practices. Also remember that 'there is elegance in simplicity'.

   Design and implement the app navigation to allow the user to navigate from one page to another in intuitive and user-friendly way to achieve the app use cases.

3) For each use case, implement the server-side data access repositories using JavaScript to read/write the app data from/to a data store.

   Also, you should initialize the data store with data from JSON files (if the data store is empty).

4) Application design documentation including the Entities Class Diagram and the Repositories Class diagram.

5) Document the app testing using screen shots illustrating the results of testing.

- Every team member should submit a description of their project contribution. Every team member should demo their work and answer questions during the demo.
- Push your implementation and documentation to your group GitHub repository as you make progress.

Note that this phase will be focused only **a fully working client-side implementation** using data stored in json files and a simple data store.

3. Grading rubric

Criteria	%	Functionality*	Quality of the implementation
1) Implement the app Web UI and navigation using HTML, CSS and JavaScript. Including designing the App Web UI and navigation.	50		
2) Implement the server-side data access repositories to read/write the app data from/to data store. Also, initialize the data store with data from JSON files.	40		
3) Application Design: Entities Class Diagram and Repositories Class diagram.	5		
4) Testing documentation using screen shots illustrating the testing results. - Discussion of the project contribution of each team member.	5		
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 for code duplication, poor/inefficient coding practices, poor naming of identifiers, unclean/untidy submission, and unnecessary complex/poor user interface design.