CS446 Final Status Report - ClubWAT

GitHub Repository - Final Demo (Youtube)

Keshav Gupta (k44gupta) Abhinav Gupta (a363gupt)
Isshana Mohanakumar (imohanak) Maryam Afshar (m5afshar)
Anjali Gupta (a378gupt) Abhinit Patil (a33patil)

April 4, 2024

Status Report

Functional Properties

1. **User Authentication:** As demonstrated in the demo the user sees the login screen on startup of the application and can login with the email/password. The user can create their account with basic details and receives an email with a six digit code that they enter in the application to complete the registration and verify their identity.

2. **User Interface:** The system uses the popular four tab bottom bar layout that a user can click any tab to go to that view. The tab bar is visible at all times during the user session.

3. **Club and Event Browsing:** Clicking on the search tab in the bottom bar will go to a view that lets you search for clubs or events based on title or description. It sorts results alphabetically and each item is clickable to view more information about the club or event.

4. **Event Bookmarking and Scheduling:** From the event detail view a user can click the bookmark button in the top row of buttons, this will save the event to their home page to ensure they don't miss it. If they click the '+' button, it will mark in the system that the user is attending the event, making it visible in the home page and optionally prompting to add the event to their phone calendar (google calendar for example) automatically so they can get notified about when the event is starting.

5. **Club Registration and Management:** From the club details view, a user can register for a club by clicking the '+' button. If the club doesn't have a membership fee they will instantly get full access to the club. If it does have a membership fee, a club admin can click the settings icon in the top right of the club details view and manage members. This shows a full list of club members and allows the club admin to mark the user as 'Paid' for when they paid their membership fee. It also lets them remove a user from the club or promote them to a club admin. Note only club owners (the user who created the club) can promote users.

6. **Notifications and Alerts:** The application has many options to send emails to users for notifications and alerts. There is also an inbox view by clicking the inbox icon on the top right of the home page view. The user can manage whether they want to receive email notifications through the profile page and clicking edit profile. System events such as a new event being created, a club join request being approved, and sharing an event or club will send an email to the relevant user.

7. **Feedback System:** Users can 'like' or 'unlike' clubs and events. Liking a club/event will add to the like counter of that club/event. This is how users will give feedback on clubs and events.

8. **Personalized Recommendations:** In the bottom navigation bar, the user can click the 'For You' page which will suggest clubs based on the user's interests in a list. The user configures their interests through the profile view and selecting the edit interests option, then entering their relevant interests.

9. **Event Creation and Management:** Club admins can create an event for their club through the club settings page (settings button on the top right corner of the club details view). They enter all the event details and click the submit button. They can also edit an event by going to the event detail view and clicking the edit icon on the top right of the view.

10. **Interactive Club Discussions:** Users can share clubs/events with their friends by clicking the paper airplane icon in the club/event detail view. They manage their friends in the user profile view and clicking 'Manage Friends'. Users can also send messages in the dedicated club group chat if they're an approved member of the club. Club admins can delete any message in the club discussion, but regular members can only delete their own messages.

Non-Functional Properties

1. **Security:** Passwords are hashed with 2^{11} rounds and 32bytes of salt which is significantly more than the original target of 1000 rounds and 32 bits of salt. Passwords are also required to be strong with more than 8 characters, uppercase, lowercase, and symbol characters.

2. **Privacy:** The user can go to their profile tab and either click the download data button which sends them a file with all of their stored data in the application. They can also scroll down to the 'Danger Zone' and click the 'Delete Account' button to delete all of their data from the application.

3. **Usability:** We used a bottom navigation bar layout in our application, which is visible at all times once the user has logged into the application. Therefore, it will always be possible for the user to get to the For You page within three clicks.

4. **Safety:** We limited the application to only users with a valid **@uwaterloo.ca** email by using the popular code verification technique that's used in many enterprise applications. Since the user needs to enter a code that's sent to their email, and the API requires the email is an **@uwaterloo.ca** domain, the user will never be able to complete the registratiion unless they have access to that **@uwaterloo.ca** email.

Mitigate Harm

University of Waterloo Students

We mitigated the harm to students by requiring **@uwaterloo.ca** email code verification during registration which was described earlier. We also added a private/public functionality to club events. Events can be marked as private when they're created which will make those events hidden from all regular users and only visible to approved members of the club. Private events also cannot be shared with others. This will ensure that only the right users have access to certain club events and protect the student population from malicious actors.

Club Fairs

We mitigate the harm to club fairs by giving them a spotlight whenever the user opens the application. Super admins (for example, WUSA), can create club fairs by going to the profile section, scrolling to the 'Danger Zone' and clicking 'Create Club Fair'. Once they create the club fair, if any user opens the application they will instantly see a dialogue that they can click and see more about the club fair. They can either dismiss it or mark it as 'Don't Show Again' which will make it so they're not notified of club fairs until a new one is created. This will ensure club fairs aren't ignored and are instead given spotlights so users are still encouraged to attend them.

User Accounts

- **Super Admin:** clubwat.admin@uwaterloo.ca, Password1$
- **Regular User:** k44gupta@uwaterloo.ca, Password1$

Individual Contributions

NOTE: See the GitHub projects tab for a complete list of code changes by each member: Task Breakdown. The list in this document is not a complete list of team member contributions to adhere to page limit constraints.

Member	Deliverables	Code	Non-code
Anjali	Wrote out all functional properties for D1 and created mockups in Figma. Wrote out the whole section for Model-View-ViewModel in D4. Wrote out design descriptions for both the frontend and the backend in D6. Wrote my contributions for D7. Proofread all deliverables.	Modified Signup and Login page. Created all of the Verification Page. Added Navigation, Password Checks, and Factory Classes. Fixed how we save data to the User class. Added bottom navigation bar. Added basic home page view with tab rows. Added all of the user's clubs on the home screen. Added Events to the Home Page. Created all of the For You Page. Worked on allowing club admins to edit clubs. Worked on allowing club admins to edit events. Debugged DateTime Picker. Re-factored whole codebase to include dependency injection.	Helped with TechStack Finalization. Organized all tickets for the team in the backlog. Reviewed multiple PRs.
Keshav	D1 NFPs, Full D2, D3 except for component diagram, D4 Client-Server, D6 Arch Styles, D7 writeup except for individual contributions, proofread all deliverables	Feedback System Frontend/Backend, Revamp Profile View UI/UX, Delete/Download User Account Frontend/Backend, Home Page Backend, Club Fair Splash Page On Startup Flow Frontend/Backend, Joining Club Frontend/Backend, Club Management View, Club Membership Management Frontend/Backend, Event Details View + Interactions Frontend/Backend, Club Details View + Interactions Frontend/Backend, Club Browsing Frontend/Backend, Notifications/Inbox System via Email and App Frontend/Backend, Personalized Recommendations Backend, Authentication, Login, and Registration Backend	Project repo setup, DB and DigitalOcean setup, Gmail SMTP and push notification research, techstack finalization and initial project setup, recorded and edited final demo, reviewed PRs

Member	Deliverables	Code	Non-code
Maryam	Worked on the introduction section for D1 and helped with the Figma files. Wrote out the Strategy design pattern for D5. Created the architecture style design diagrams for D6. Recorded my part for the in class demo. Wrote my contributions for D7. Proofread the deliverables.	Created the initial frontend setup using the MVVM model. Worked on the login and sign up pages. Created the profile page which includes the user being able to change password, add friends, view all friends, accept friend requests, edit interests and view all interests. Added the 30 seconds wait time before sending another verification code. Worked on allowing users to delete clubs.	Researched the best UI Components library, looked into the MVVM frontend model set up, reviewed PRs
Isshana	Created user scenario diagrams and created Figma mockups in D1. Wrote the entire Implicit invocation section for D4. Wrote out key patterns, classes, abstractions, and algorithms found in our project for D6. Wrote my contributions for D7. Proofread all deliverables.	Created new Club Creation flow frontend. Implemented club approval flow frontend. Investigated and implemented Dependency Injection with Hilt for entire project. Implemented Delete User Messages in Club Discussions frontend. Implemented Club Discussions flow frontend for sending and populating discussions chats. Coded the backend APIs necessary for club discussions feature. Did a complete project refactor for best practices and design patterns (ex. authentication, architecture, API request handling). Created multiple reusable repositories for handling network calls.	Researched ways to refactor our project code for best practices. Investigation for dependency injection implementation with Hilt. Helped create presentation slides. Reviewed multiple PRs.
Abhinav	Added stakeholders to D1. Wrote out the Decorator design pattern for D5. Wrote down the design patterns we used, how we minimize coupling, and how the patterns can adapt to future software requirements in D6. Wrote down my contributions for D7. Proofread all the deliverables.	Built CRUD APIs clubs in the Backend. Backend APIs for user interests. Built the Frontend for browsing through events. ClubAdmins create a new event for their respective clubs frontend. Backend APIs for users being promoted to or demoted from the ClubAdmin role. Built Backend APIs for getting clubs approved. Introduced new middleware functions to the Backend code, like verifyIsClubAdmin. Fixed some bugs that appeared in our code throughout the term.	Designed the database schema for all SQL tables in the software. Reviewed multiple PRs.
Abhinit	Worked on functional properties, user scenarios, sequence diagrams, Human Values, stakeholders and target user population in D1. Created component diagram for D3. Reviewed D4. D5: write-up for Observer Pattern + diagram. D6: class diagram and sequence diagrams. Finally, added personal contribution in D7	Worked on the backend endpoints for Club approvals, Events management and user friendships. This included fetching, creating, updating and deleting records and performing checks accordingly. Coordinated with the frontend development for user friendships and events in app.	Helped design and review initial Database schema. Reviewed PRs; Suggested prisma integration