Critical Design Review Document

IWAC Conference Application for Heather Falconer

PenUltimate

Brett Palmer, Monica Agneta, George Pitt, Rebecca Sonnemann, Ben Caras

19 December 2025

Abstract

The IWAC Application is being developed for the purpose of allowing in-person and virtual attendees of the International Writing Across the Curriculum Conference (IWAC) to be able to navigate conference sessions, information, and materials. This application is wanted by the client, Dr. Falconer due to the fact that past applications were missing certain functionalities that the PenUltimate team aims to provide in the IWAC Application. The Critical Design Review Document (CDRD) contains a summary of the System Requirements Specification (SRS), System Design Document (SDD), and User Interface Design Document (UIDD). These documents establish a shared understanding of project requirements, system architecture, and user interface design for the IWAC Application. The PenUltimate team has created a prototype of the application in order to visualize the user interface design and to ensure it meets the standards of Dr. Falconer. The CDRD serves as documentation for the development of the IWAC Application as the PenUltimate team begins developing system features, components, and testing these fundamental components.

Preface

The IWAC Application is being completed in partial fulfillment of the requirements for the B.S. in Computer Science degree for the University of Maine. Dr. Falconer is part of the Association for Writing Across the Curriculum (AWAC), which hosts the International Writing Across the Curriculum Conference. The IWAC Application is meant to provide a way for

in-person attendees to navigate session schedules and find the information they require. The application further provides a way for virtual attendees to have a conference experience being able to access session information, presentation materials, and recordings of presentations. The PenUltimate team members have prior experience with the software being used to develop this application, as well as with similar databases. This Critical Design Review Document (CDRD) is meant to provide an understanding of requirements, and how the PenUltimate team has developed the IWAC Application in order to fulfill these requirements. The CDRD serves as documentation of the development process for Dr. Falconer as well as those who update or maintain the IWAC Application in the future.

Table of Contents

1 Introduction…….……...…….5

1.1 Purpose……..…….……6

1.2 Methods…….……...…….…..6

2 Project Status Update…..…….…….…..7

2.1 Completed Documents…….…….…..…7

2.1.1 System Requirements Specification……....8

2.1.2 System Design Document……..……11

2.1.3 User Interface Design Document……...15

2.2 Current Prototype Status…….…….…….…18

3 Milestones……..…….…….……..18

3.1 Project Timeline……...…….…….…20

4 Equipment…….…….22

5 Conclusion…….……...…….23

5.1 Recommendations…….…….…24

6 References…….……...……..25

7 Bibliography…….……...……..26

7.1 Documentation…….…….…….26

7.2 PenUltimate Documentation…….…….…26

7.3 PenUltimate Repository…….……26

8 Acknowledgement…….……...…….27

Appendix A – Agreement Between Customer and Contractor……...…..28

Appendix B – Team Review Sign-off……...……29

Appendix C – Document Contributions…….……...30

Appendix D – System Requirements Specification……...…….……...32

Appendix E – System Design Document……...…….……...73

Appendix F – User Interface Design Document ..……..…….……..95

List of Figures

Figure 1 - Top-Level Use Case and Context Diagram ……..…..…….……10

Figure 2 - Layered Architecture Diagram ..……..……..……..12

Figure 3 - MVC Logical Architecture Diagram ..……..…….……..13

Figure 4 - IWAC Conference App Class Diagram ..……..……..…….……14

Figure 5 - AWAC Style Guide……...……..…….…16

Figure 6 - Sessions, Agenda, and Messages Screens ..……...…….…17

List of Tables

Table 1 - Project Timeline ..……..…….……...……20

Summary

The CDRD summarizes the first phase of the IWAC Application development by describing key components from the System Requirements Specification, System Design Document, and User Interface Design Document documents; it defines requirements and how the system features and components satisfy these requirements. As well as a status update on where the PenUltimate team is currently within the development process, and outlines milestones and future deliverables. The introduction outlines the key features and goals that the IWAC Application aims to achieve as well as defines the purpose of this document. Within the sections summarizing the SRS, SDD, and UIDD documents it focuses on defining and further providing descriptions of the requirements, design of the system, and design of the user interface. The Equipment section outlines the software that was chosen in order to develop this application as well as providing background information as to why each of these development tools were chosen.

1 Introduction

Academic conferences such as the one hosted by IWAC, rely on clear communication and accessible schedules to support engagement among attendees. Our client, who is part of AWAC, expressed the need for an application which is designed for easy use and allows attendees to browse the full conference schedule, create a personalized agenda, and communicate with fellow attendees. The IWAC Conference Application has been developed with long-term reuse in mind

as the IWAC conference reoccurs every other year. This project aims to create the application as a tool that can be adapted for the future conference with minimal additions. This critical design review ensures that the application's prototype architecture, interface, and proposed database structure supports the requirements for the next IWAC conference and the future ones.

1.1 Purpose

The purpose of the CDRD is to define the progress established in developing the IWAC Conference Application and determine if it meets the technical requirements established by the client. This document specifies attendee usability and administrative capabilities, a detailed overview of the key behaviors, and the project timeline. This document also identifies recommendations to ensure success in developing the final product. The CDRD serves as a centralized reference for the complete design of the application and guides the next phase of its development.

1.2 Methods

Methods used to evaluate the IWAC Conference Application design included a review of all created diagrams to ensure consistency among them. We also reviewed our established functional and non-functional requirements against the design to ensure requirements could be mapped to a design component and that none had been disregarded. An overarching evaluation was done to ensure that the application's interface has a clear and easy to use interface. Our final

method was integrating client feedback obtained from every review meeting into improving our application design.

2 Project Status Update

In this chapter, we provide a detailed update on the current status of the IWAC Application. We summarize the completed documentation, the meetings held with our client Dr. Falconer, and the progress made towards developing the final product. This chapter highlights the accomplishments at each stage of the project so far, including the SRS, SDD, and UIDD. Additionally, we describe the corresponding review meetings and approvals received, as well as the current state of the application prototype.

2.1 Completed Documentation and Reviews

The IWAC project has successfully progressed through all planned documentation review stages. We have completed the SRS, SDD, and UIDD, and held the corresponding review meetings: the System Requirements Review (SRR), System Definition Review (SDR), and Software Specification Review (SSR) with our client, Dr. Falconer. During these meetings, Dr. Falconer reviewed and approved all stages of our project, providing feedback and confirmation that our progress aligns with her expectations.

2.1.1 System Requirements Specification (SRS)

We wrote the SRS to identify and detail the core requirements for the IWAC Application to establish its desired functionality. In Section 2, we expressed functional requirements as use cases, which specify how the system should help conference attendees navigate an IWAC conference. In Section 3, we established non-functional requirements that set performance standards for how the IWAC Application should perform, regardless of operating system. Furthermore, in Section 6 we acknowledged open issues with the requirements that were resolved throughout the semester. By defining the requirements and discussing them with Dr. Falconer during our SRR, we were able to ensure a common understanding of the application's necessary features.

This application has several key requirements, which include reusability for consecutive conferences, allowing in-person attendees to easily navigate session schedules, allow in-person attendees to find the necessary information they require, allowing for virtual attendees to have more of a conference experience, such as being able to communicate with conference participants and ask questions to presenters, and ensuring that this application is available on both Apple and Android mobile devices. The main features that are being developed based on the requirements include: messaging between conference participants, accessibility of presentation materials and recordings, filtering of sessions by theme, presenter, or day when searching the list of sessions, creation of personal agendas with each attendees personal selection of sessions, ability for administrators to send push notifications, and the ability for virtual attendees to ask questions to presenters.

The non-functional requirements on section 3 of our SRS lists different performance and security standards that our application should meet. These standards include aspects such as loading time, minimal accepted data loss, how user passwords are encrypted, how quickly a user is able to learn how to use the application, minimum uptime during the conference itself, and more. This ensures that data for the IWAC Application is readily available to be accessed and changed by users in approved manners only.

Figure 1 is a top-level use case and context diagram that displays all requirements that our client requested. Each use case, labeled with a number above it, is described in-depth under its own table in the SRS. Each table is labeled with the associated use case number, name, priority, brief description, a step-by-step description of how these use cases are meant to function, and any open issues associated with them. For more information on the application's requirements, see the SRS in Appendix D.

Figure 1 - Top-Level Use Case and Context Diagram

2.1.2 System Design Document (SDD)

We wrote the SDD to establish the system architecture of the IWAC Application and to illustrate how our system will satisfy the project requirements. The SDD contains the architectural design of the application, as well as the descriptions of the classes within the object-oriented IWAC Application system. As well as descriptions of how the data will be stored within the database and what types of data will be associated with certain features.

Figure 2 is a layered architecture diagram. The top layer is the User Interface Layer, which contains all of the different pages that attendees and system administrators are able to view. The Application Layer contains all of the actions users are able to perform, which are able to modify the information within the database on the Data Layer of the diagram. Next to the Data Layer are external applications that are integrated within our application, such as the IWAC User Registration database for verifying conference enrollment.

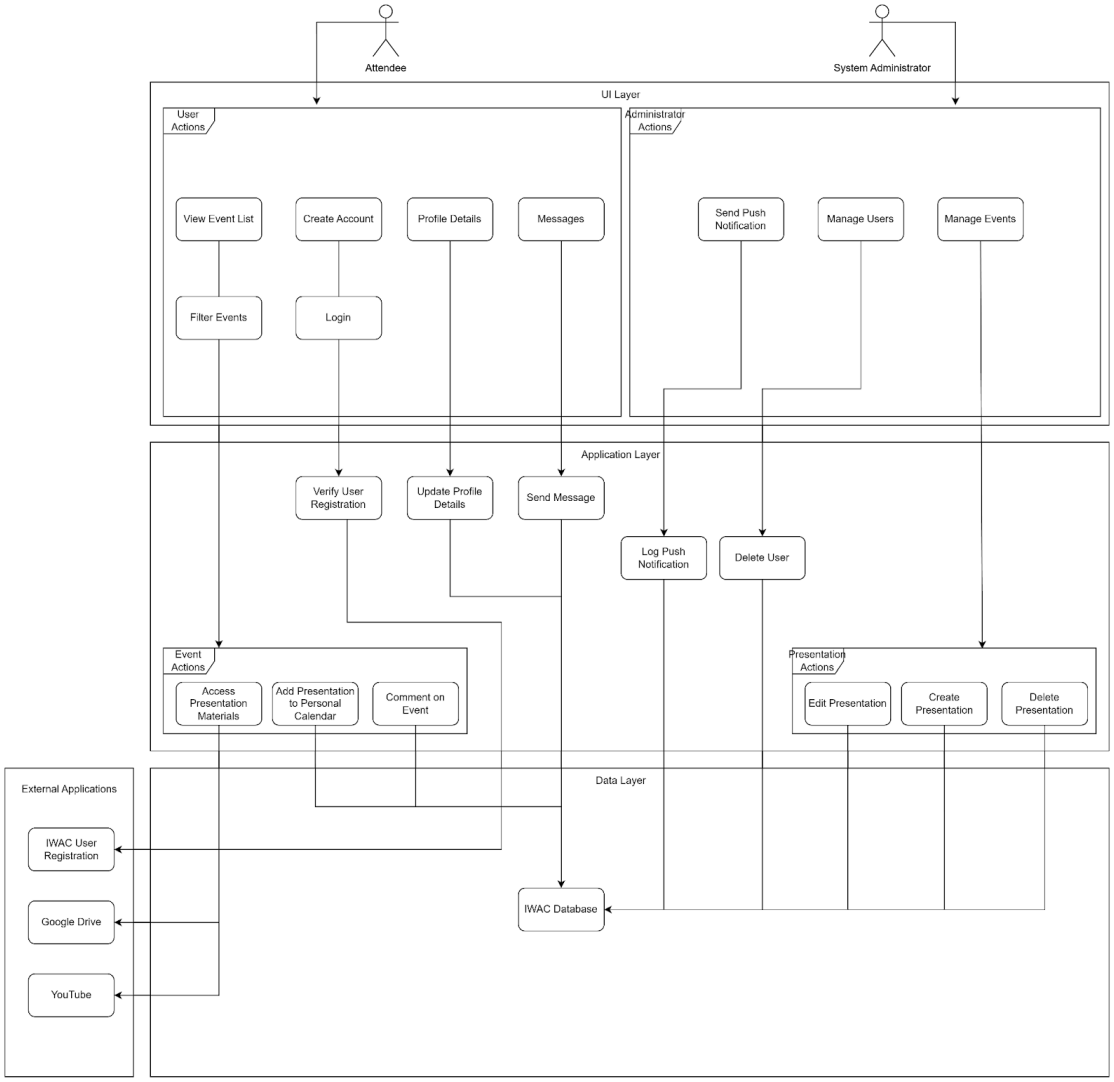

Figure 2 - Layered Architecture Diagram

We designed our system to implement a Model-View-Controller (MVC) structure as illustrated in Figure 3 which is most appropriate for a React Native mobile application. Figure 3 shows how the attendees interact with the UI which triggers API calls that interact with our database. This diagram also shows how the system administrator will input data through Supabase's external data entry UI. The model layer is a Supabase database that stores all data necessary to manage conference information. The view layer consists of visual screens and

components that conference attendees will interact with in the IWAC Application. The controller layer facilitates communication between the other two layers, primarily via Supabase API calls. This illustration provides a high-level overview of the system's data structure and inner table relationships.

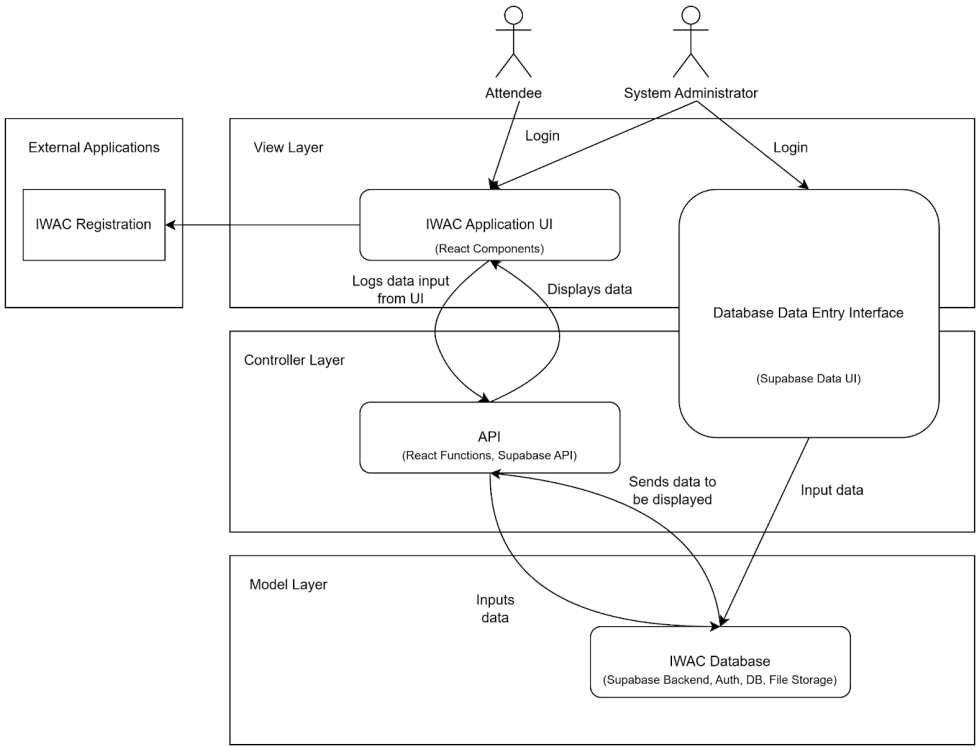

Figure 3 - MVC Logical Architecture Diagram

We also created the App Class Diagram Figure 4 to identify the main classes within the object-oriented IWAC Application and described their qualities as attributes and capabilities as methods. Figure 4 illustrates how all of the functional requirements that were listed in the SRS document have a corresponding method present within the App Class Diagram. This ensures that the system architecture is designed to address all functional requirements of the IWAC Application.

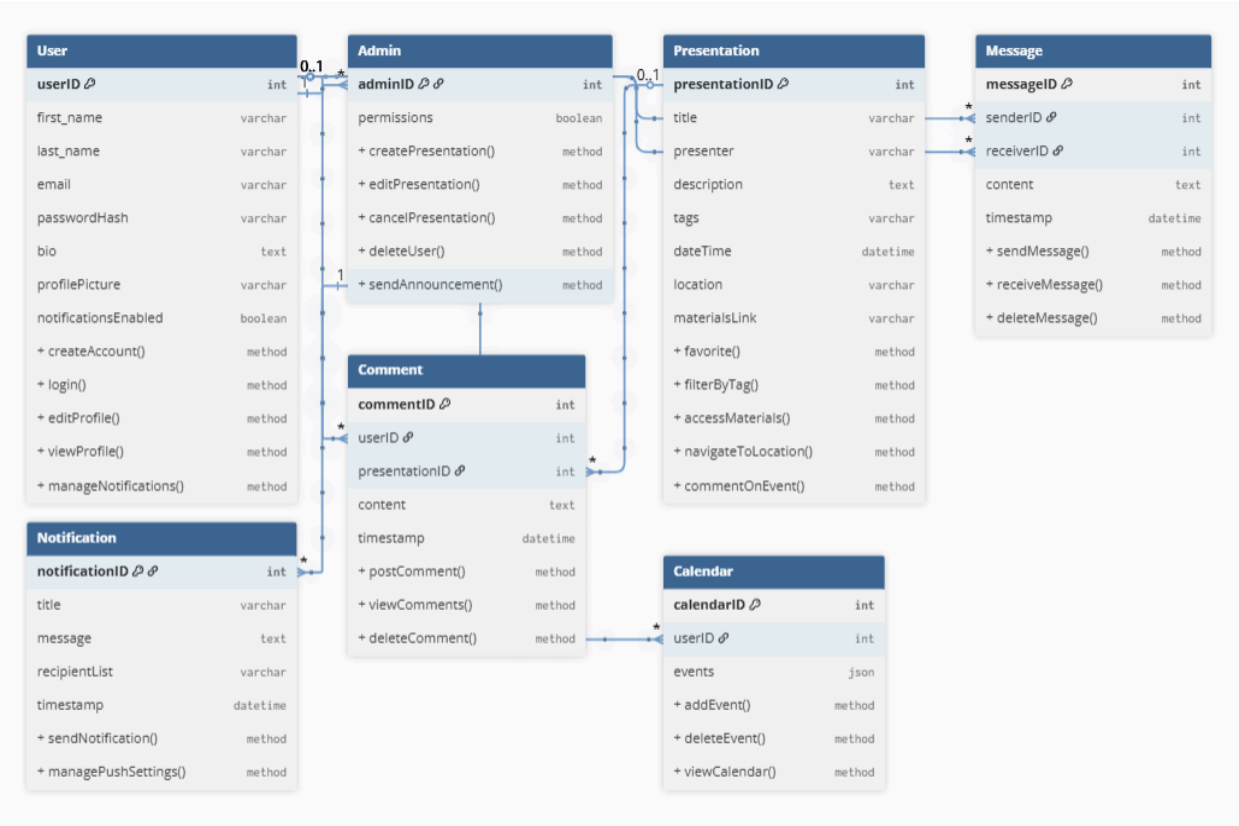

Figure 4 - IWAC Conference App Class Diagram

The application will primarily be written in TypeScript using the Expo (2025) full-stack React Native (2025) framework, due to the fact that the application needs to be usable on both Android and Apple mobile devices. The system will be tested using Detox (2025), a testing framework for React Native that multiple team members have prior experience with. Additionally, we outlined all database tables that will be required to reliably store and update user and conference information. The application will use Supabase (2025) for a PostgreSQL database, Supabase is an open source database software. The data for the application will be entered directly into Supabase's interface by the system administrator. Users will enter data in the app which will then be updated using Supabase API calls. Supabase has been chosen given

the fact that Supabase's UI and data entry aligns with Dr. Falconer's expertise. During our SDR with Dr. Falconer, we obtained approval that our system design addressed all of the functional requirements that she requested. For more information on the application's architecture, see the SDD in Appendix E.

2.1.3 User Interface Design Document (UIDD)

The UIDD was created to define, describe, and prototype the components of the IWAC Application that are visible to the user. Its purpose is to demonstrate how conference attendees will interact with the IWAC Application to navigate a conference, as well as how an administrator will interact with the database management tool to organize a conference. The UIDD also ensures that the user interface supports the functional requirements defined in the SRS. Lastly, the UIDD serves as an early prototype of the application by presenting an implementation of the interface made with React Native, providing both visual and functional validation of the design decisions.

In section 2 of the UIDD, we elaborated on the overarching design components that establish a consistent and intuitive user interface. These components include common interface elements such as the bottom tab navigator, session containers, and standardized error messages. We also described how our code implements the style guide provided by Dr. Falconer (AWAC, 2025b), shown in Figure 5. The guide encourages the consistent usage of beige, blue, and orange throughout the application, as well as the fonts Alata and Open Sans Condensed.

Figure 5 - AWAC Style Guide

In section 3, we provided a full walkthrough of the application interface. Each screen was prototyped in React Native, allowing us to illustrate the user interface while simultaneously building an early version of the IWAC Application. Following each screen, we explained how users can interact with the interface to complete tasks that satisfy the functional requirements from our SRS. Figure 6 highlights the overall style and structure of our application through three sample screens.

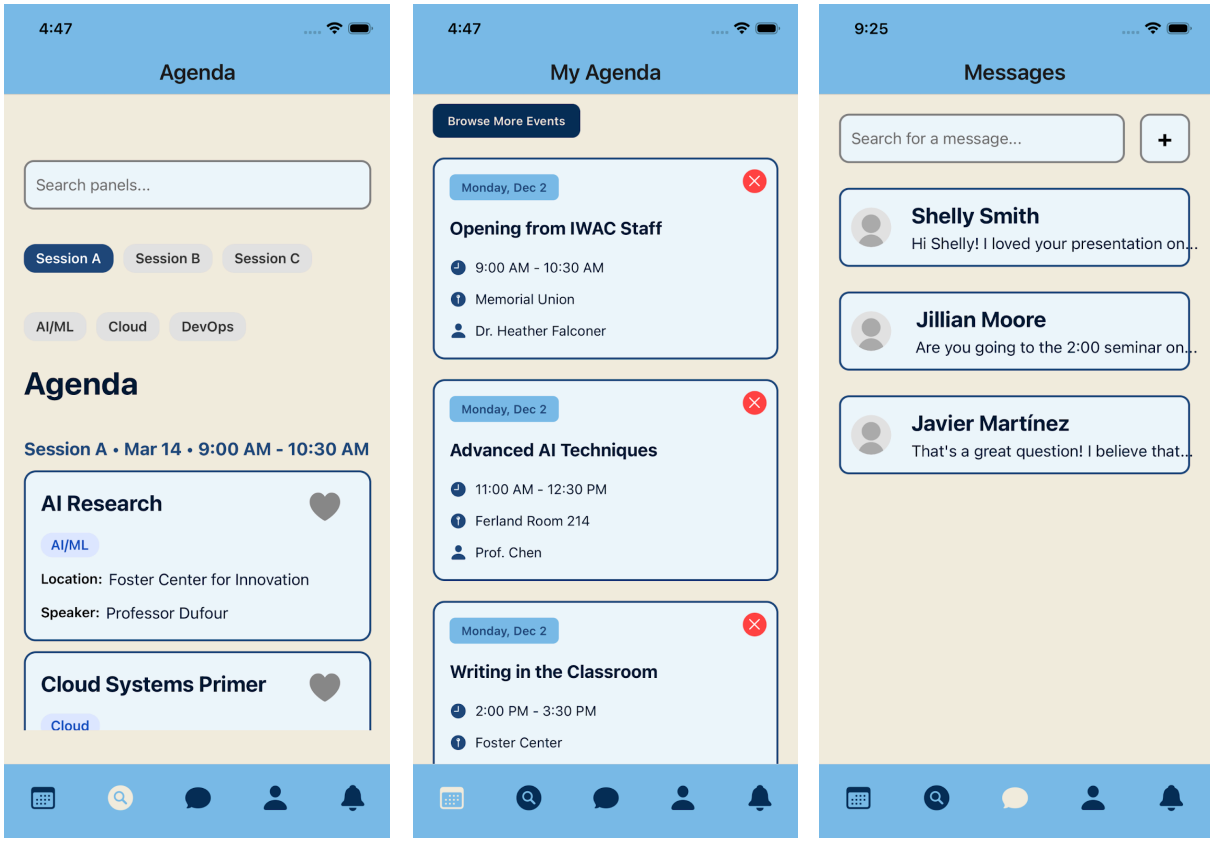

Figure 6 - Agenda, My Agenda, and Messages Screens

In section 4, we outlined how information is added to and managed within the system, with a clear distinction between administrative and user permissions. We presented the Supabase interface used to input conference and presentation data, which is only accessible by system administrators such as Dr. Falconer. We also described which components of the user interface would allow attendees to enter and update their information within the application. This section ensures consistency between user interface components and the underlying database design.

During our SSR, Dr. Falconer approved of the user interface prototype and complimented its clear and straightforward design. This approval confirms that the user interface effectively satisfies the project requirements. For more information on the application's user interface design and implementation, see the UIDD in Appendix F.

2.2 Current Prototype Status

As mentioned previously, we have completed a fully functional user interface prototype for the IWAC Application. Every primary screen has been built in React Native with components designed to support the user workflows defined in our SRS. In addition, we have created a GitHub repository with a project structure that reflects the MVC system architecture outlined in our SDD. The resulting prototype is a cohesive interface that reflects the layout, navigation patterns, and visual standards established in our UIDD. Furthermore, we developed a sample Supabase table to demonstrate how conference sessions and presentations will be entered and managed by the conference administrator through Supabase's built-in UI. Dr. Falconer has reviewed and approved our documentation, interface prototype, and the database workflow. Collectively, these accomplishments set a strong foundation for the next phase of development for the IWAC Application during the Spring 2026 semester.

3 Milestones

In Fall 2025, our team began development of the IWAC Conference Application. We assigned team roles, them being Monica Agneta as the client liaison, Rebecca Sonnemann as the Scrum Master, and Brett Palmer as the Repository Master. In our first meeting with our client - Heather Falconer, we specified the functional requirements for the application, and used that information to then choose an appropriate technology stack including Supbabase for our database. We then went on to design the system architecture and layout of our application, as well as mockups of the user interface and how the user will interact with it. Finally, we set up the

development environment with React Expo and coded the UI components for each screen in our application.

The project is well positioned to move into the implementation and testing stages in Spring 2026. Our next steps begin with completing the Test Readiness Review and integrating live data retrieval from Supabase into the application by the end of February. We will integrate this controller logic into screens within our prototype. In early March, we will train Dr. Falconer on how to input and manage conference information through the database, and will write the User Manual. The remainder of the semester will focus on the testing pipeline, beginning with unit testing progressing through integration, system, and acceptance testing. Finally, we will complete the Final Project Report and deliver the Final Project Presentation in late April 2026.

3.1 Project Timeline

Table 1 - Project Timeline

Deliverable	**Date**
System Requirements Specification	October 29, 2025
Systems Requirements Review	October 30, 2025
System Design Document	November 17, 2025
Systems Definition Review	November 20, 2025
User Interface Design Document	December 3, 2025
Software Specification Review	December 4, 2025
Critical Design Review Document	December 18, 2025
Critical Design Review	December 18, 2025
Prototype Presentation	December 19, 2025
Input conference data into database	Early February 2026
Integrate external IWAC registration into Create Account screen	Early February 2026
Code Inspection Report	Early February 2026

Preliminary Software Review	Mid February 2026
Administrator Manual	Mid February 2026
Test Readiness Review	End February 2026
Populate database data in the application	End February 2026
Code functionality of conference panels	End February 2026
Teach client how to input data into database	Early March 2026
User Manual	Early March 2026
Functional Configuration Audit	Mid March 2026
Code the functionality of messages, profile, & notifications	Mid March 2026
Unit Testing	End March 2026
Integration Testing	Early April 2026
System Testing	Mid April 2026
Acceptance Testing	End April 2026
Final Project Report	End April 2026
Final Project Presentation	End April 2026

4 Equipment

When building our application, we used React Native (2025) with Expo (2025). React Native allows us to code for both iOS and Android devices using only typescript, rather than learning each operating system's own language. Expo further simplifies this process by using file-system routing, allowing us to create screens without needing to deal with the navigation stack one would otherwise have to work with. In addition, Expo also provides several libraries to further facilitate development, such as the camera, maps, and even notifications. Expo also has integrations with Supabase (2025), which we will be using for our back-end data storage. Supabase is a database where the presentation and user information will be stored. For testing, we will use Detox (2025), which tests user inputs, button presses, page navigation, and other user interface-related components. Detox begins automatically on app launch, and is directly integrated with our application. Not only does this make our tests faster and more reliable, but it also allows for cross-platform testing, meaning that we don't need two separate tests for iOS and Android devices. All of this is stored in a shared GitHub repository (PenUltimate, 2025), allowing us to establish version control for our application that can be easily accessed by the developers and client.

While our database holds the presentation and user information, our application will also need to connect to the IWAC registrant list, which holds the information of all individuals who have purchased tickets for attendance. As this is a paid conference, only those who have purchased tickets will be able to create an account with our application. To do this, our application will connect to the IWAC registrant list any time someone attempts to create an

account. If they are properly registered for this conference, their account will then be created, and stored in Supabase.

The last piece of equipment needed is a smartphone using iOS or Android, which must also have reliable internet access, not only for downloading our application itself, but also for connecting to the IWAC registrant list when registering for an account, connecting to Supabase to load and view any presentation or user details, making changes or edits to the user's personal details, and anything that requires accessing information not stored directly on the smartphone.

5 Conclusion

In this Critical Design Review Document, we summarized the progress made on the IWAC Application during the Fall 2025 semester. We completed all major project documents for the semester: the SRS, SDD, and UIDD. Also, we successfully obtained approval for each document in its corresponding review meetings with Dr. Falconer. Collectively, these documents establish a shared understanding of project requirements, system architecture, and user interface design for the IWAC Application. Furthermore, we developed a complete React Native user interface prototype, structured it according to the MVC architecture of our application, and demonstrated how conference data will be managed through Supabase. With all milestones achieved and client approval, the project enters next semester with a strong foundation and a clearly defined path to producing a fully functional conference application.

5.1 Recommendations

Based on the current prototype and feedback received during design reviews, several recommendations are proposed to guide the next phase of development. The application should prioritize the integration of live data from Supabase into the existing user interface. Continued usability testing with both administrative users and conference attendees is recommended. A lack of sufficient user testing presents the risk of usability issues going undetected until later stages of development, which could delay deployment. Clear documentation should also be continued to support long-term reuse of the application.

6 References

Association for Writing Across the Curriculum. (2025a). *IWAC 2025*. https://iwac2025.lineupr.com/iwac-2025/

Association for Writing Across the Curriculum. (2025b). *AWAC social media kit*.

Bushnell, C., Costley, S., & Daniel, A. (2023). *WAC for transitions: the next 50 years.* [Conference program] International Writing Across the Curriculum Conference, Clemson, SC, United States.

dbdiagram.io. (2025). Dbdiagram.io: Database relationship diagrams design tool. Holistics Software. https://dbdiagram.io/d

draw.io. (2025). *Draw.io: Free flowchart maker and diagrams online.* draw.io Ltd. https://app.diagrams.net/

Fowler, M. (1997). *UML distilled: A brief guide to the standard object modeling language*. Addison-Wesley. http://ci.nii.ac.jp/ncid/BA65029497

LineUpr. (2025). *The event application solution to boost your event communication – LineUpr*. LineUpr GmbH. https://lineupr.com/en

Whova. (2025). *Whova: Award-winning event apps and event management*. Whova, Inc. https://whova.com/

Wireframe.cc | The go-to wireframing tool. (2025). Wireframe.cc. https://wireframe.cc/

7 Bibliography

7.1 Documentation

Detox. (2025). https://wix.github.io/Detox/

Expo. (2025). Expo. https://expo.dev/

React Native. (2025). *React Native · A framework for building native apps using React.* Meta Platforms, Inc. https://reactnative.dev/

Supabase. (2025). *The Open Source Firebase Alternative*. Supabase Inc. https://supabase.com/

7.2 PenUltimate Documents

- System Requirements Specification - Appendix D
- System Design Document - Appendix E
- User Interface Design Document - Appendix F

7.3 PenUltimate Repository

PenUltimate (2025). *IWAC Conference App*. GitHub. https://github.com/umaine-capstone-iwac/IWAC-Conference-App

8 Acknowledgement

We would like to thank our client Dr. Falconer, who has consistently met with us to provide validation throughout the project's development. We would also like to thank our professor Terry Yoo and teaching assistant Johnny Sylvain for their guidance throughout the project.

Appendix A - Agreement Between Customer and Contractor

The PenUltimate team agrees to design and develop the IWAC Conference Application in accordance with the specifications and requirements defined in this Critical Design Review Document (CDRD) and its supporting artifacts, including the System Requirements Specification (SRS), System Design Document (SDD), and User Interface Design Document (UIDD).

The scope of development includes application functionality, user interface requirements, attendee and administrator actions, data storage, and database management. Future developers will have access to our code and the database to be able to update it in the future as needed, as our involvement with building this application will end after May 2026.

Any requested modifications to this agreement must be discussed and mutually approved by both parties prior to implementation. If changes are requested by the client, Dr. Heather Falconer, such requests should be communicated to the team's designated client liaison, Monica Agneta. Likewise, if the PenUltimate team proposes changes, Monica Agneta will communicate these requests to the client for review and approval.

Signature: _____ Date: 12/18/2025

Signature: _____ Date: 12/18/2025

Signature: _____ Date: 12/18/2025

Signature: _____ Date: 12/18/2025

Signature: _____ Date: 12/18/2025

Signature: _____ Date: 12/18/2025

Appendix B - Team Review Sign-off

All team members have signed to acknowledge they have reviewed this document and agreed on both its content and format. If team members have minor disagreements, they may state them in the comments area.

Rebecca Sonnemann
Signature: _____ Date: 12/18/2025
Comments: _____

Brett Palmer
Signature: _____ Date: 12/18/2025
Comments: _____

Monica Agneta
Signature: _____ Date: 12/18/2025
Comments: _____

Ben Caras
Signature: _____ Date: 12/18/2025
Comments: _____

George Pitt
Signature: _____ Date: 12/18/2025
Comments: _____

Appendix C - Document Contributions

This section outlines the individual contributions of each team member to the development of this document. Each subsection details the specific sections written, figures created, and revisions made by each contributor.

Brett

Brett wrote the section 2 introduction, section 2.1 introduction, section 2.1.3, section 2.2, and the conclusion. He contributed to the writing for section 2.1.1, section 2.1.2, section 3 introduction, bibliography, and acknowledgements.

Monica

Monica wrote the report introduction, section 1.1, section 1.2, and section 3.

George

George wrote most of section 2.1.1, as well as section 4. He estimates that he did roughly 20% of the work for this document.

Rebecca

Rebecca wrote section 5.1, 6, 7, 8, and the appendices. She also filled out the table of contents.

Ben

Ben wrote section 2.1.2, the preface, abstract, and summary. Ben estimates he did 20% of the work.

Appendix D - System Requirements Specification Document

System Requirements Specification

IWAC Conference Application for Heather Falconer

PenUltimate

Brett Palmer, Monica Agneta, George Pitt, Rebecca Sonnemann, Ben Caras

30 October 2025

Table of Contents

1 Introduction……..…….……3

1.1 Purpose of This Document……..……..……3

1.2 References.……..……...…….4

1.3 Purpose of the Product…….……..……...4

1.4 Product Scope……..……..……4

2 Functional Requirements……..…….……..6

2.1 Account Management Requirements…….6

2.1.1 Create Account……..……..……6

2.1.2 Login to Account…….....….……..8

2.2 User Profile and Communication Requirements……..…….9

2.2.1 Edit Profile Details……...……..…….9

2.2.2 View Other User Profiles……..……10

2.2.3 Message User……..……..……11

2.2.4 Manage Push Notifications…….……..12

2.3 Presentation and Calendar Requirements..……..……13

2.3.1 Filter Presentations……..…….14

2.3.2 Access Presentation Materials……..14

2.3.3 Add Presentation to Personal Calendar……15

2.3.4 Access Personal Calendar……..…...16

2.3.5 Navigate to Event Location……..17

2.3.6 Comment on Event……..…….18

2.3.7 Access Previous Years' Presentations……...19

2.4 Administrator Requirements……21

2.4.1 Create Presentation……...21

2.4.2 Edit Presentation……...22

2.4.3 Delete Presentation……...22

2.4.4 Delete User……23

2.4.5 Send Push Notification……24

2.5 Functional Requirement Tests……..26

3 Non-Functional Requirements……...30

4 User Interface……32

5 Deliverables……...32

6 Open Issues……...33

6.1 Scalability……....……33

6.2 Administrator Data Management…….……33

6.3 Data Privacy…….…34

Appendix A – Agreement Between Customer and Contractor……..35

Appendix B – Team Review Sign-off……36

Appendix C – Document Contributions……....37

Revision History

Date	Reason For Changes	Version
10/9/25	Initial creation of template document.	0.0.1
10/16/25	The group collectively revised the document during our group meeting.	0.0.2
10/19/25	The group collectively revised the document for the SRS draft submission.	0.0.3
10/20/25	Final version of the SRS draft.	0.1.0
10/29/25	Final version of the SRS, modified in response to the Peer Review.	1.0.0

10/30/25	Reviewed and signed by our client during our SRR.	1.1.0

1. Introduction

This capstone project focuses on building and developing an application for the IWAC (International Writing Across the Curriculum) conference. Held every two years by the Association for Writing Across the Curriculum, this four day conference has hundreds of attendees, many attending in person, and many attending remotely. To make the conference experience better for all parties involved, the AWAC decided to start using scheduling applications. Although they have used other applications and subscriptions in the past, they have all had some form of issue. To solve this problem, we have been requested to build a new scheduling application for the 2027 IWAC conference. This application shall allow registered attendees to view and filter the conference schedule, as well as add certain presentations to their personal schedule, and show their interest for that specific panel. All panels on the application will include a link to a Google doc containing all of the presentation materials, as well as a link to access the presentation livestream in order to assist all remote attendees. Additional desired features include direct messaging and notifications.

1.1 Purpose of This Document

The purpose of this System Requirements Specification document is to clearly define the functional and non-functional requirements for the IWAC Conference Application being developed by the PenUltimate team. The functional requirements describe the essential capabilities the system must support while the non-functional requirements define how the system will perform these functions. Together, the requirements establish a guide for the design and development process. The System Requirements Specification is intended for the PenUltimate development team and organizers of the IWAC conference to ensure a shared understanding of the product.

This document begins by introducing the objectives of the project, then defines the system's functional and non-functional requirements. It goes on to describe the intended user interface and expected deliverables. The document concludes by identifying any open issues that have been raised and outlining the details of the client agreement.

1.2. References

Association for Writing Across the Curriculum. (2025). *IWAC 2025*. https://iwac2025.lineupr.com/iwac-2025/

draw.io. (2025). *Draw.io: Free flowchart maker and diagrams online.* draw.io Ltd. https://app.diagrams.net/

Fowler, M. (1997). *UML Distilled: A brief guide to the standard object modeling language*. Addison-Wesley. http://ci.nii.ac.jp/ncid/BA65029497

LineUpr. (2025). *The Event application solution to boost your event communication – LineUpr*. LineUpr GmbH. https://lineupr.com/en

Whova. (2025). *Whova: Award-winning Event Apps and Event Management*. Whova, Inc. https://whova.com/

1.3. Purpose of the Product

This client is requesting an application for the Association for Writing Across the Curriculum, which holds a four day conference called the International Writing Across the Curriculum Conference (IWAC). In 2027, UMaine is hosting this conference. The client is interested in having a mobile conference application that could be repurposed for future conferences. The client noted having issues with last year's application, such as having to scroll through too many events and being unable to filter through them (Whova, 2025). The client also mentioned that when a previous app was used on mobile devices, it opened up a web page that was not optimal (LineUpr, 2025). The problem that the client is trying to solve is finding an effective way to allow in-person attendees to navigate the session schedule and find the info they need, as well as creating a way for virtual attendees to have more of a conference experience.

1.4. Product Scope

The IWAC Application is designed to allow attendees of the International Writing Across the Curriculum Conference a way in which they will be able to view and access information regarding events. Conference attendees, both in person and virtual, will be able to message and interact with other participants as well as the presenters of the events. Attendees will be able to create personal itineraries related to the events at the conference and will be able to access all materials used in the presentations. Presentation materials and recorded videos will be hosted on Google Drive and YouTube, respectively. The links to these materials will be stored in the IWAC Application database, where they will be mapped to their associated events. When a user selects a material link, the system will redirect them to the appropriate external platform.

Figure 1.4 – Top-Level Use Case and Context Diagram

Figure 1.4, modeled after the UML guidelines from Fowler (1997), illustrates the capabilities available to both users and application administrators within the IWAC Application. The large plane depicted in Figure 1.4 represents the IWAC Application and contains all features and systems that will be encapsulated within the application. The squares on the right side of the plane represents external applications that are not present within the IWAC Application but will

be integrated or interacted with. The lines extending from the attendee show the actions that can be performed by users.

2. Functional Requirements

This section defines the core functional requirements of the IWAC Conference application through UML use case diagrams made with *draw.io* (2025) and accompanying use case specifications based on the format from Alistair Cockburn. These functional requirements and use case specifications illustrate how users and the system interact to achieve key goals such as viewing event schedules, messaging other participants, and managing conference content. A corresponding set of system and acceptance tests follows the use cases to demonstrate how the functional requirements will be proven to be satisfied. Each test is paired with one or more use cases to ensure complete and traceable validation of system functionality.

2.1 Account Management Requirements

The section defines the use cases that enable users to create and access their accounts within the IWAC Conference App. Figure 2.1 provides a unified view of the “Create Account” and “Login” use cases, which are specified in Table 2.1.1 and Table 2.1.2, respectively.

2.1.1 Create Account

Table 2.1.1 – Create Account Use Case Specification

Number	FR-1
Name	Create Account
Summary	A registered conference attendee creates an account for the IWAC Conference application using their registered email and self-chosen password.
Priority	5
Preconditions	- The user must have externally registered for the IWAC Conference using a valid email address.

	- The email used must not already be associated with an account on the IWAC Application.	
Postconditions	- A new user account is created and associated with the verified conference registration. - The attendee can log in using the email and password provided.	
Primary Actor	Attendee	
Secondary Actors	External IWAC Registration System, Internal IWAC Authentication System	
Trigger	The attendee selects "Sign Up" on the application's welcome screen.	
Main Scenario	**Step**	**Action**
	1	The attendee selects "Sign Up" from the welcome screen.
	2	The system prompts the attendee to enter the email used during conference registration and a self-selected password.
	3	The user enters their conference email and a password.
	4	The user taps "Create Account."
	5	The system verifies the email against the external IWAC registration system.
	6	The system stores the new account's email and hashed password in the internal IWAC Authentication System.
	7	The system displays a message confirming the account creation.
Extensions	**Step**	**Branching Action**
	5a	The email is not found in the external IWAC Registration System. **:** Display an error message and prompt the attendee to re-enter their email or confirm conference registration.
	6a	The email is already associated with an account. : Display a message informing the user that the email is already associated with an account, and encourage them to log in.
Open Issues	It is unknown if IWAC allows users to make multiple conference registrations under a single email address. It is undetermined if the client wants two-factor authentication to be implemented for account creation.	

2.1.2 Login to Account

Table 2.1.2 – Login to Account Use Case Specification

Number	FR-2	
Name	Login to Account	
Summary	A registered conference attendee accesses their IWAC Application account by entering their verified email and password credentials.	
Priority	5	
Preconditions	- The attendee must have an existing account created through the IWAC Application welcome screen.	
Postconditions	- The attendee gains access to their personal account, which includes their profile, event feed, calendar, and messages.	
Primary Actor	Attendee	
Secondary Actors	Internal IWAC Authentication System	
Trigger	The attendee selects "Sign In" on the application's welcome screen.	
Main Scenario	**Step**	**Action**
	1	The attendee selects "Sign In" on the welcome screen.
	2	The system prompts the attendee to enter their registered email and password.
	3	The user enters their email and password.
	4	The user taps "Login."
	5	The system verifies the credentials against the internal IWAC Authentication System.
	6	Upon successful verification, the system grants access to the attendee's account.
	7	The system displays the application's home screen with personalized conference features.
Extensions	**Step**	**Branching Action**
	5a	The email is not in the IWAC Authentication System. **:** Display an error message telling the attendee to create an IWAC Application account before logging in.
	5b	The email is found but the password is incorrect. : Display an error message telling the user to re-enter their credentials.
Open Issues	It is undecided if password recovery will be handled internally or through the external IWAC registration system.	

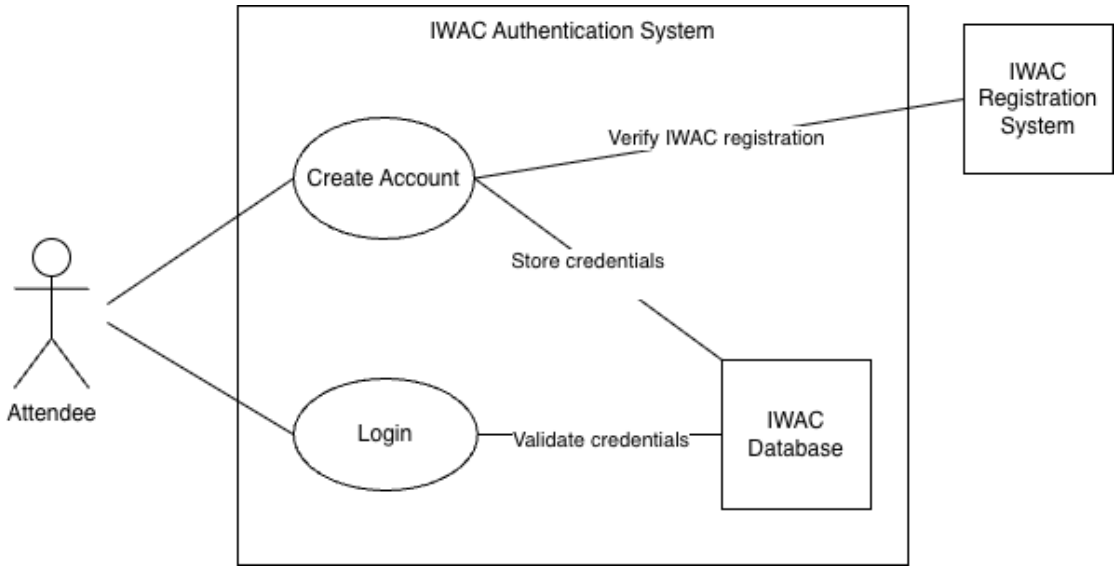

Figure 2.1 Use Case Diagram — Account Management

2.2 User Profile and Communication Requirements

The section defines the use cases that enable users to interact with others and manage personal settings within the IWAC Conference App. Figure 2.2 provides a unified view of the "Edit Profile Details", "View Other User Profiles", "Message User", and "Manage Push Notifications" which are specified in Tables 2.2.1 through 2.2.4, respectively.

2.2.1 Edit Profile Details

Table 2.2.1 – Edit Profile Details Use Case Specification

Number	FR-3
Name	Edit Profile Details
Summary	Registered users can personalize their profile by adding a profile picture and short bio. If the user does not upload a picture or enter a bio, the profile will display only their username.
Priority	3
Preconditions	- The user must have logged into their IWAC account. - The system has access to device files.
Postconditions	- The system displays the user's updated profile details.

	- If not added, the profile shows only the username.	
Primary Actor	Conference Attendee	
Secondary Actors	IWAC Database, User Device File System	
Trigger	The user clicks on their username from the home screen.	
Main Scenario	**Step**	**Action**
	1	The user selects “Edit Profile” from the profile screen..
	2	The system displays the editable fields for the profile picture upload and bio.
	3	The user may upload or update their profile picture from camera roll.
	4	The user may enter or update their profile bio.
	5	The user selects “Save” to save the changes.
	6	The system updates the user’s profile in the database.
	7	The system displays the updated profile details to the user.
Extensions	**Step**	**Branching Action**
	1a	If user cancels image upload**:** The default profile image remains.
Open Issues	There are no open issues with this use case.	

2.2.2 View Other User Profiles

Table 2.2.2 – View Other User Profiles Use Case Specification

Number	FR-4	
Name	View Other User Profiles	
Summary	This use case allows attendees and presenters to view other users’ profiles.	
Priority	2	
Preconditions	- The user must be logged in. - The application must be connected to the database.	
Postconditions	- The selected user’s profile information is displayed.	
Primary Actor	Attendee	
Secondary Actors	Database	
Trigger	The user clicks on another user’s profile.	
Main Scenario	**Step**	**Action**
	1	The user logs into the application.
	2	The user navigates to the list of attendees.

	3	The user selects an attendee's name.
	4	The system displays the selected user's profile information.
Extensions	**Step**	**Branching Action**
	1	The database is not connected. **:** The application displays an error message "Profile cannot be retrieved."
Open Issues	There are no open issues with this use case.	

2.2.3 Message User

Table 2.2.3 –Message User Use Case Specification

Number	FR-5	
Name	Message User	
Summary	This use case allows an attendee or presenter to send and receive messages within the app.	
Priority	1	
Preconditions	- The user must be logged in. - The application must be connected to the database.	
Postconditions	- The message will be stored in the database. - The recipient received a notification regarding the message. - The message is visible for both users.	
Primary Actor	Attendee	
Secondary Actors	Notifications system, Database	
Trigger	The user selects "message" while viewing another attendee's profile.	
Main Scenario	**Step**	**Action**
	1	The user opens the application.
	2	The user opens the list of other attendees.
	3	The user clicks on a specific attendee.
	4	The user clicks the "message" button on their profile.
	5	The user types a message in the text box.
	6	The user clicks "send."
	7	The system checks that the message is valid and that the recipient exists in the database.
	8	The system stores the message in the database.
	9	The system sends a push notification to the recipient, if enabled.

Extensions	Step	Branching Action
	6a	The text box is empty. **:** application displays an error message “Message cannot be empty.”
	8a	The message cannot reach the database. : The application displays an error message “Unable to send message.”
Open Issues	There are no open issues with this use case.	

2.2.4 Manage Push Notifications

Table 2.2.4 – Manage Push Notifications Use Case Specification

Number	FR-6	
Name	Manage Push Notifications	
Summary	A registered conference attendee toggles push notifications for conference updates and session reminders in the IWAC Application.	
Priority	2	
Preconditions	- The user must have logged into the IWAC Application. - The device must have notifications enabled for the application at the system level.	
Postconditions	- The user’s notification preferences are updated and stored in the database. - The application sends or suppresses notifications based on the user’s selected preferences.	
Primary Actor	Attendee	
Secondary Actors	Push Notification Service	
Trigger	The attendee selects "Notifications" from the settings screen.	
Main Scenario	**Step**	**Action**
	1	The attendee navigates to the Settings screen.
	2	The attendee selects Manage Notifications.
	3	The system displays toggle options for Conference Announcements and Direct Messages.
	4	The attendee adjusts the settings by toggling each category on or off.
	5	The attendee selects “Save” when satisfied with the settings.

	6	The system saves the updated preferences to the database and displays a message confirming the changes have been applied.
Extensions	**Step**	**Branching Action**
	3a	Device notifications are disabled at the system level. **:** The system displays a prompt directing the user to enable notifications in device settings.
Open Issues	The categories for notifications will be expanded if requested by the client.	

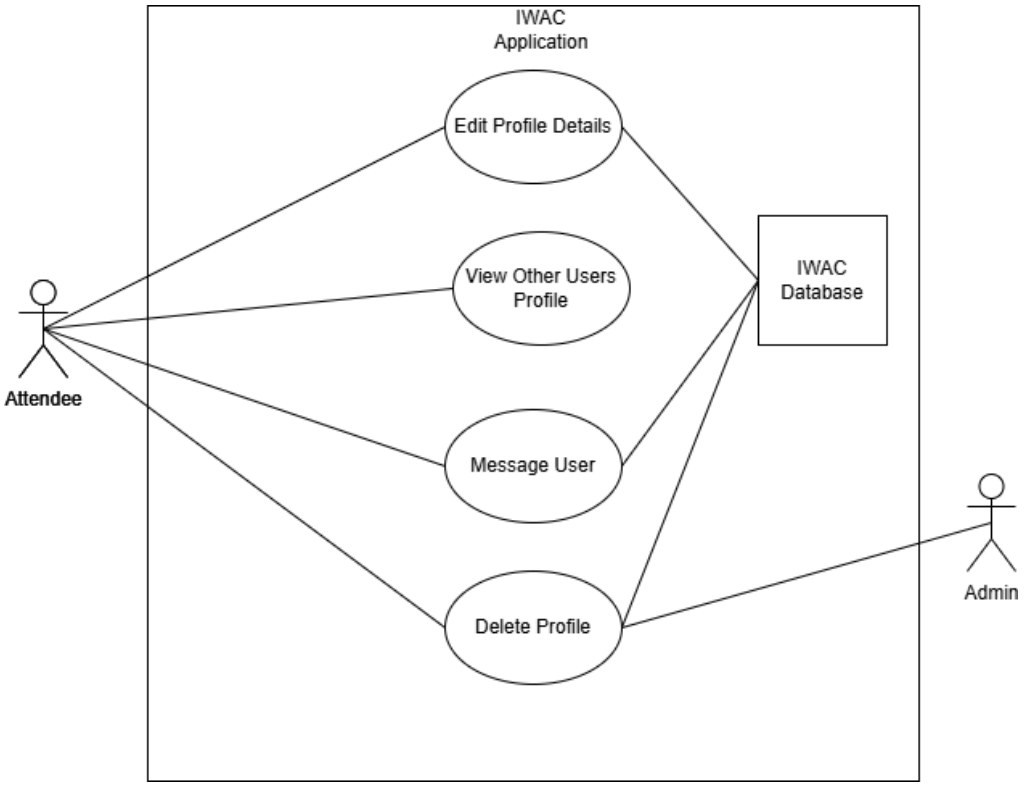

Figure 2.2 Use Case Diagram — User Profile and Communication

2.3 Presentation and Calendar Requirements

The section defines the use cases that enable users to filter and manage conference sessions within the IWAC Conference App. Figure 2.3 provides a unified view of the “Filter

Presentations", "Access Presentation Materials", "Add Presentation to Personal Calendar", "Access Personal Calendar", "Navigate to Event Location", "Comment on Event", and "Access Previous Years' Presentations" use cases, which are specified in Tables 2.3.1 through 2.3.7, respectively.

2.3.1 Filter Presentations

Table 2.3.1 – Filter Presentations Use Case Specification

Number	FR-7	
Name	Filter Presentations	
Summary	IWAC conference application users can filter the list of presentations by date and presenter	
Priority	5	
Preconditions	- The user must have finished creating their account on the application, and be logged in. - All presentations are inserted into the database correctly.	
Postconditions	- A narrower search result is returned including only presentations that match the selected filters.	
Primary Actor	Conference Attendee	
Secondary Actors	IWAC database	
Trigger	The user selects "search" after choosing their desired filters	
Main Scenario	**Step**	**Action**
	1	The user selects their desired filters
	2	The user selects "search"
	3	A narrower search result is returned with the appropriate presentations
	4	The user can return to the home screen, navigate to another screen, or make a new search.
Extensions	**Step**	**Branching Action**
	1a	The user changes screens after selecting filters. **:** The user goes to the new screen, and the filters are remembered for future searches.
Open Issues	We are unsure whether offline searches will be implemented.	

2.3.2 Access Presentation Materials

Table 2.3.2 – Access Presentation Materials Use Case Specification

Number	FR-8	
Name	Access Presentation Materials	
Summary	Users that are logged into an account will be able to view the slideshow of any given presentation. Users will also be able to be redirected to a previously recorded YouTube video of the presentation or a YouTube livestream of the presentation.	
Priority	5	
Preconditions	- The user must be logged into an account. - The user must navigate to the list of events.	
Postconditions	- The user will be redirected to a Google Drive folder that contains the slideshow of the presentation. - The user will be redirected to a previously recorded YouTube video of the presentation or a YouTube livestream of the presentation.	
Primary Actor	Conference Attendee	
Secondary Actors	IWAC Database	
Trigger	The user selects a listed event.	
Main Scenario	**Step**	**Action**
	1	The user selects “Events List” from the home screen.
	2	The user will then tap on one of the listed events.
	3	The user will then tap on the link to access presentation materials or the link of the Youtube video or livestream.
Extensions	**Step**	**Branching Action**
	3a	The user selects the button for the presentation materials. : The user is redirected to the Google Drive folder containing presentation slideshows via their web browser.
	3b	The user selects the button for the presentation video. : The user is redirected to YouTube for a prerecorded presentation video or a presentation livestream via their web browser.
Open Issues	We do not know which web browsers will support this feature.	

2.3.3 Add Presentation to Personal Calendar

Table 2.3.3 – Add Presentation to Personal Calendar Use Case Specification

Number	FR-9	
Name	Add Presentation to Personal Calendar	
Summary	Users that are logged into an account will be able to "favorite" events listed which will add them to their own personal itinerary.	
Priority	5	
Preconditions	- The user must be logged into an account. - The user must navigate to the list of events.	
Postconditions	The event will then be added to their personal calendar.	
Primary Actor	Conference Attendee	
Secondary Actors	IWAC Database	
Trigger	The user selects "Events list" from the application's home screen.	
Main Scenario	**Step**	**Action**
	1	The user selects "Events List" from the home screen.
	2	The user will then tap on one of the listed events.
	3	The user will then "favorite" that event.
Open Issues	We have not fully decided on what the user's personal calendar or itinerary will look like.	

2.3.4 Access Personal Calendar

Table 2.3.4 –Access Personal Calendar Use Case Specification

Number	FR-10	
Name	Access Personal Calendar	
Summary	Registered conference members can review their personal calendar including upcoming sessions and saved events. The user can filter their calendar by day or topic.	
Priority	5	
Preconditions	- The user must have logged into their IWAC account. - The personal calendar will initially be blank if the user has not saved any events.	
Postconditions	- The system displays the users personal calendar. - Filters are applied to the calendar if selected.	
Primary Actor	Conference Attendee	
Secondary Actors	IWAC Database	
Trigger	The user selects "My Calendar" from the application's home screen.	
Main Scenario	**Step**	**Action**
	1	The user selects "My Calendar" from the home screen.

	2	The system displays the user's personal calendar.
	3	The user may apply filters to view their events by a specific day or topic.
	4	The system updates the calendar view based on the applied filters.
	5	If events have been saved, the user can tap on an event to view details.
	6	The user can clear filters, return to the home screen, or navigate to another page.
Extensions	**Step**	**Branching Action**
	1a	No events have been saved to the user's personal calendar. **:** The system displays "No events saved yet".
	3a	No events match the selected filters. : The system displays "No events match selection".
Open Issues	There are no open issues with this use case.	

2.3.5 Navigate to Event Location

Table 2.3.5 – Navigate to Event Location Use Case Specification

Number	FR-11	
Name	Navigate to Event Location	
Summary	An attendee geographically locates a selected conference event by clicking a map icon, which redirects them to the location in an external navigation app.	
Priority	3	
Preconditions	- The user must be logged in to the IWAC Application. - The event must have an associated address in the database.	
Postconditions	- The user is redirected to an external mapping application displaying the selected event's geographic location.	
Primary Actor	Attendee	
Secondary Actors	External Map Service (e.g., Google Maps or Apple Maps)	
Trigger	The attendee selects a conference event from the Events or Calendar tabs, then clicks the map icon associated with it.	
Main Scenario	**Step**	**Action**
	1	The attendee selects a given event from the Event or Calendar tab to expand its details.

	2	The system shows the full details of the selected event, including the event's address and a map icon.
	3	The attendee clicks the map icon under the event.
	4	The system prompts the attendee to choose from a list of navigation applications on their device.
	5	The user selects their preferred navigation application from the provided options.
	6	The system redirects the user to the chosen navigation app, which displays the event's location as the destination.
Extensions	**Step**	**Branching Action**
	2a	The event is exclusively virtual. **:** The system displays "Virtual" in the address field and does not show a map icon. The use case ends.
	4a	An external map service is unavailable : The system displays an error message suggesting that the user manually search for the event using the provided address.
Open Issues	There are no open issues with this use case.	

2.3.6 Comment on Event

Table 2.3.6 –Comment on Event Use Case Specification

Number	FR-12
Name	Comment on Event
Summary	This use case allows attendees to post comments or questions under a specific conference event, creating a thread visible to all logged-in participants.
Priority	4
Preconditions	- The user must be logged in. - The application must be connected to the database.
Postconditions	- The comment is stored in the database and displayed under the event. - Other users can view and respond to the comment.
Primary Actor	Attendee
Secondary Actors	Notifications system, Database
Trigger	The attendee selects an event and clicks "Add Comment."

Main Scenario	Step	Action
	1	The user opens the application and finds a specific event.
	2	The attendee scrolls to the comment section.
	3	The user types a message in the text box.
	4	The user clicks “post.”
	5	The system checks that the message is valid and that the recipient exists.
	6	The system stores the message in the database.
	7	The system uploads the message to the event’s public thread.
Extensions	**Step**	**Branching Action**
	4a	The text box is empty **:** The application displays an error message “Message cannot be empty.”
	6a	The message cannot reach the database : The application displays an error message “Unable to send message.”
Open Issues	There are no open issues with this use case.	

2.3.7 Access Previous Years’ Presentations

Table 2.3.7 – Access Previous Years’ Presentations Use Case Specification

Number	FR-13
Name	Access Previous Years’ Presentations
Summary	This use case allows users to access presentation slides, recordings, or other materials from previous IWAC conferences.
Priority	3
Preconditions	- The user must be logged in. - The application must be connected to the database. - The database must contain previous presentation materials, grouped by conference.
Postconditions	- The user obtains access to the previous years’ presentation materials.
Primary Actor	Attendee
Secondary Actors	Database
Trigger	The attendee navigates to the “previous conferences” page and selects a conference year.

Main Scenario	Step	Action
	1	The user logs into the application.
	2	The user navigates to the “previous conferences” screen.
	3	The system displays a list of previous conference titles and their corresponding years.
	4	The user selects a conference year.
	5	The system displays a list of presentations with links to presentation materials in a Google Drive.
Extensions	**Step**	**Branching Action**
	3a	No materials are available. **:** The application displays an error message “No materials found.”
	5a	The external link is unavailable : The application displays an error message “Link unavailable.”
Open Issues	There are no open issues with this use case.	

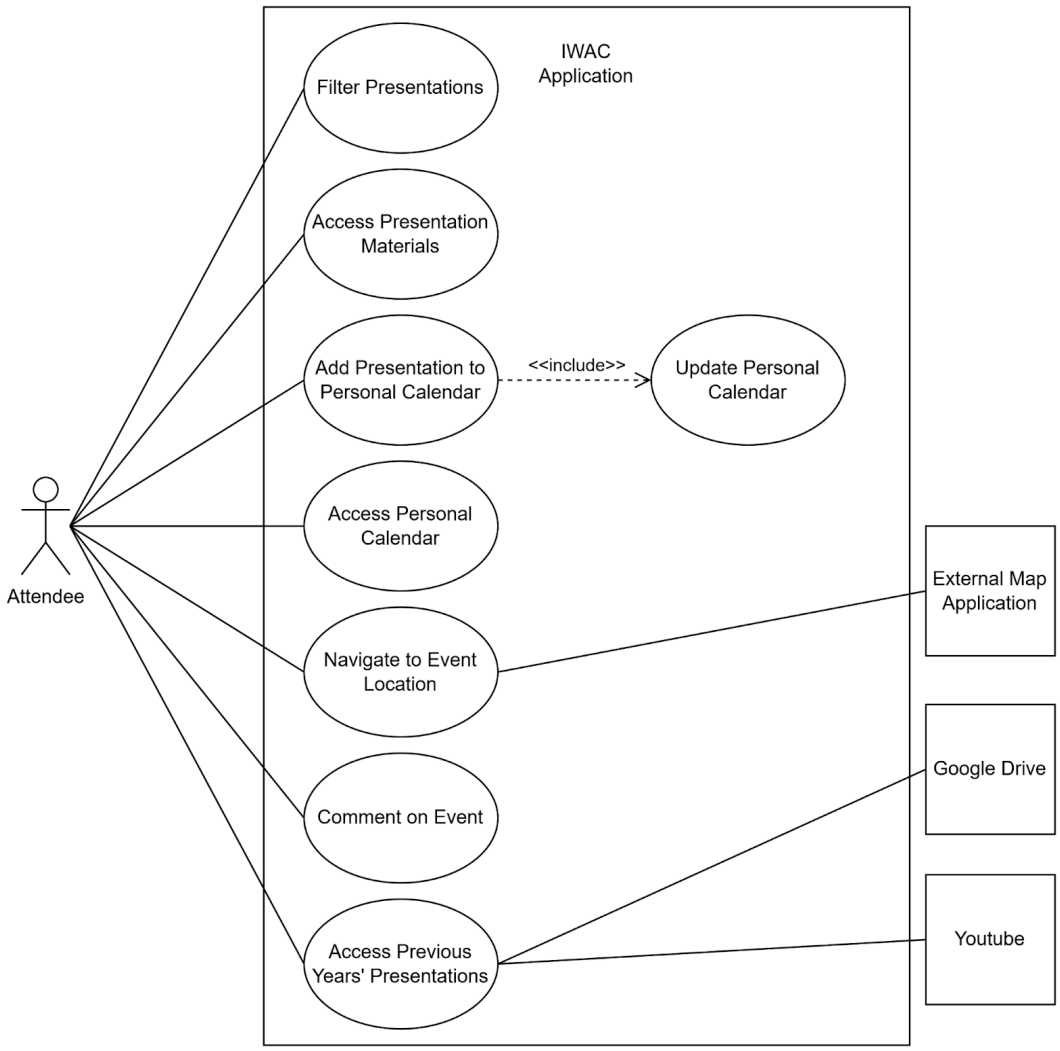

Figure 2.3 Use Case Diagram — Presentation and Personal Calendar

2.4 Administrator Requirements

The section defines the use cases that enable the administrator capabilities within the IWAC Conference App. Figure 2.4 provides a unified view of the “Create Presentation”, “Edit Presentation”, “Delete Presentation”, “Delete User”, and “Send Push Notifications” use cases, which are specified in Tables 2.4.1-2.4.5, respectively.

2.4.1 Create Presentation

Table 2.4.1 – Create Presentation Use Case Specification

Number	FR-14	
Name	Create Presentation	
Summary	This use case allows administrators to create new presentations and set all necessary information for them.	
Priority	5	
Preconditions	- The administrator must be logged in. - The application must be connected to the database. - The presentation details are filled out correctly.	
Postconditions	- The administrator successfully creates a new presentation	
Primary Actor	Admin	
Secondary Actors	Database	
Trigger	The administrator selects “Create new presentation” from the menu	
Main Scenario	**Step**	
	1	The administrator logs in to the application.
	2	The administrator selects “Create new presentation”.
	3	The administrator inputs the necessary information.
	4	The administrator clicks “submit”.
	5	The database and application both update to include the new presentation.
Extensions	**Step**	
	4a	The administrator attempts to submit a presentation with no information entered. **:** The system discards the presentation and notifies the administrator as such.
Open Issues	We are unsure whether there should be a built-in way to ensure that no presenter gets scheduled for multiple presentations at the same time.	

2.4.2 Edit Presentation

Table 2.4.2 – Edit Presentation Use Case Specification

Number	FR-15
Name	Edit Presentation

Summary	This use case allows administrators to edit their presentations and change the contents and details in it.	
Priority	4	
Preconditions	- The administrator must be logged in. - The application must be connected to the database. - The database must contain all presentations.	
Postconditions	- The administrator successfully makes changes to the presentation.	
Primary Actor	Admin	
Secondary Actors	Database	
Trigger	The administrator selects an event and clicks “Edit Presentation”	
Main Scenario	**Step**	
	1	The administrator logs in to the application.
	2	The administrator selects the presentation they wish to change.
	3	The administrator makes the changes they would like.
	4	The administrator clicks “submit changes”.
	5	The database and application both update the presentation with the new information.
Extensions	**Step**	
	3a	The administrator removes all information from a presentation. **:** The system warns that the presentation will be deleted if they select “submit changes”.
Open Issues	We are unsure whether previous versions of presentations should be stored in the database to allow administrators to revert changes, and users to access older iterations of the presentations.	

2.4.3 Delete Presentation

Table 2.4.3 – Delete Presentation Use Case Specification

Number	FR-16
Name	Delete Presentation
Summary	This use case allows administrators to delete presentations, such as if a presentation gets canceled.
Priority	5
Preconditions	- The administrator must be logged in. - The application must be connected to the database. - The database is functioning properly.

Postconditions	- The administrator successfully deletes a presentation	
Primary Actor	Admin	
Secondary Actors	Database	
Trigger	The administrator selects “Delete Presentation” from the menu	
Main Scenario	**Step**	
	1	The administrator logs in to the application.
	2	The administrator selects the presentation they wish to delete.
	3	The administrator clicks “delete presentation”
	4	The presentation is removed from the calendar and set to hidden, but not removed from the database.
Open Issues	We do not know if the client would like built-in method to prevent presentations from being deleted by accident.	

2.4.4 Delete User

Table 2.4.4 – Delete User Use Case Specification

Number	FR-17	
Name	Delete User	
Summary	Administrator users can permanently remove a registered user’s account from the conference system. After the administrator user deletes the user profile, it is no longer accessible through the application.	
Priority	4	
Preconditions	- Administrator user must be logged into administrator account. - Target user accounts exists in the database.	
Postconditions	- The selected user account is permanently removed from the database. - The deleted user can no longer login.	
Primary Actor	Administrator	
Secondary Actors	IWAC Database	
Trigger	Administrator selects user account and clicks “Delete User” from user management screen.	
Main Scenario	**Step**	**Action**
	1	Administrator logs in with an administrator account.
	2	The administrator navigates to user management screen.
	3	System displays a list of all registered users.
	4	Administrator selects user to delete.

	5	System displays pop up to confirm deletion.
	6	Administrator confirms deletion of account.
	7	System removes user from the database.
	8	System updates the registered user list.
Extensions	**Step**	**Branching Action**
	6a	The administrator cancels deletion from the confirmation prompt. **:** The user account is not deleted.
Open Issues		There are no open issues with this use case.

2.4.5 Send Push Notifications

Table 2.4.5 – Send Push Notifications Use Case Specification

Number	FR-18	
Name	Send Push Notifications	
Summary	Administrator can create and send push notifications to all registered users that have push notifications enabled. Notifications for IWAC may include schedule changes or important conference updates.	
Priority	4	
Preconditions	- The administrative user must be logged into their administrator account.	
Postconditions	- Users that enabled push notifications receive the notification from the application.	
Primary Actor	Administrator	
Secondary Actors	IWAC Database/Notification Center	
Trigger	Administrator selects “Send Announcement" from the administrator screen.	
Main Scenario	**Step**	**Action**
	1	Administrator logs in with an administrator account.
	2	Administrator navigates to "Announcements" section of the administrator screen.
	3	Administrator selects “New”.
	4	The system prompts administrative user with a field for new message.
	5	Administrator enters the title of the notification along with the announcement message.
	6	Administrator selects “Send”.

	7	System sends notification to the push notification center.
	8	System sends announcement to all registered users.
	9	Users with notifications enabled receive the announcement through the application.
Extensions	**Step**	**Branching Action**
	8a	The notification fails to be sent. **:** The system displays “Failed to send” message.
Open Issues		There are no open issues with this use case.

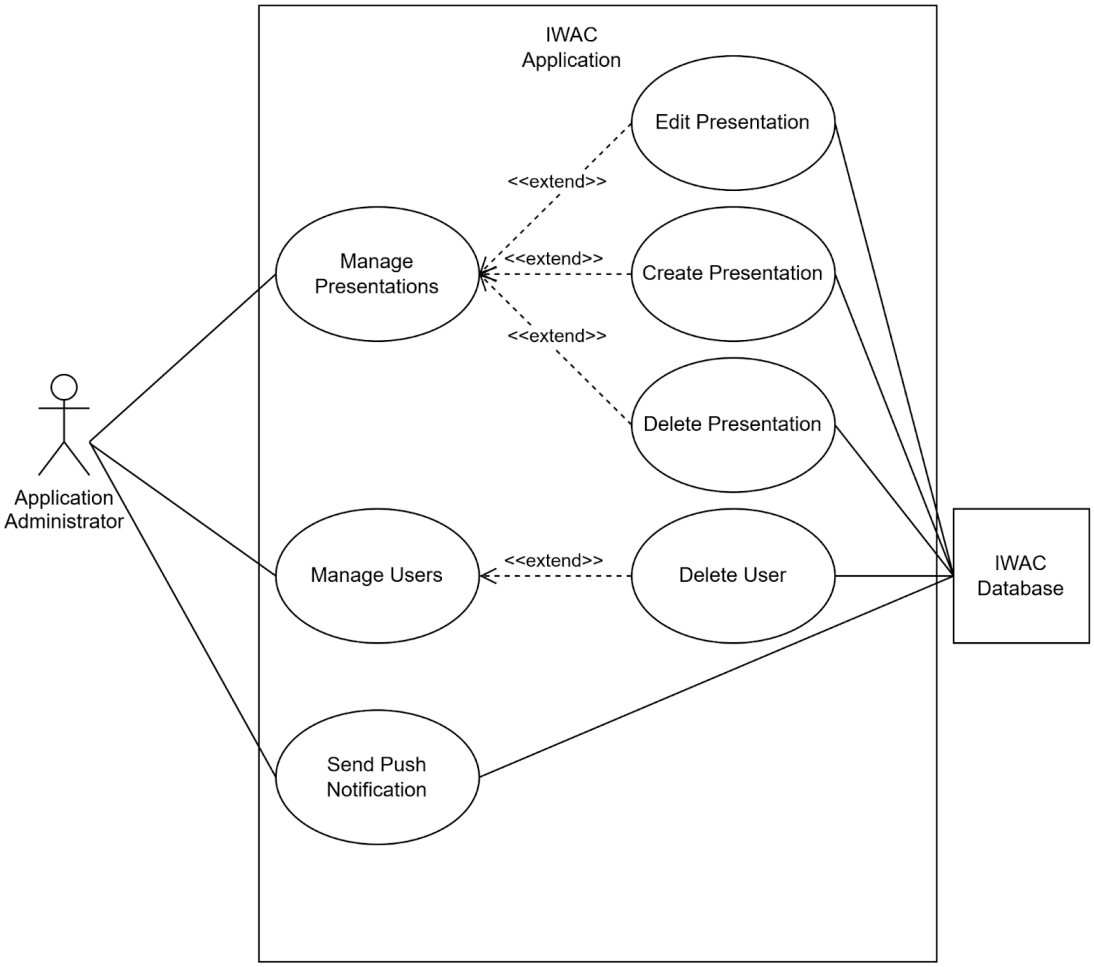

Figure 2.4 Use Case Diagram — Administrator

2.5 Functional Requirement Tests

All test cases verifying the functional requirements described in Sections 2.1 through 2.4 are presented in Table 2.5. Each test case is identified by a unique identifier, the FRs and specifies the corresponding functional requirements, preconditions, test description, and expected result. These tests will be executed using automated testing software to ensure consistency and efficiency.

Table 2.5 – Functional Requirement Tests

Test ID#	**Covered Functional Requirement(s)**	**Preconditions**	**Test**	**Expected Result**
1	FR-1 (Create Account)	- The email is registered for the conference. - No account in the application exists with the email.	Verify a successful account creation with valid credentials. 1) Open the welcome screen. 2) Select “Sign Up.” 3) Enter valid email and create a password. 4) Tap “Create Account.”	The user is redirected to the home screen in their personal account.
2	FR-1 (Create Account)	- The email is not registered for the conference.	Verify a failed account creation with an unregistered email. 1) Open the welcome screen. 2) Select “Sign Up.” 3) Enter valid email and create a password. 4) Tap “Create Account.”	An error message informs the user that no conference registration matches the email.
3	FR-1 (Create Account)	- An account in the application exists with the email.	Verify a failed account creation with a duplicate email. 1) Open the welcome screen. 2) Select “Sign Up.” 3) Enter an email associated with an account and create a password. 4) Tap “Create Account.”	An error message informs the user that the email is already associated with an account.
4	FR-2 (Login to Account)	- An account with the credentials exists in the	Verify a successful login with correct credentials. 1) Open the welcome screen. 2) Select “Sign In.”	The user is redirected to the home screen with their personalized

		database.	3) Enter a valid email and password. 4) Tap “Login.”	content.
5	FR-2 (Login to Account)	- An account with the credentials does not exist in the database.	Verify a failed login attempt with an incorrect email or password. 1) Open the welcome screen. 2) Select “Sign In.” 3) Enter an invalid email or password. 4) Tap “Login.”	An error message indicates that the email or password is incorrect, and the login fails.
6	FR-3 (Edit Profile Details)	- The user is logged in.	Verify user can update and save changes to profile. 1) Open profile screen. 2) Select “Edit Profile. 3) Modify bio or upload profile picture. 4) Click the “Save” button.	The system successfully saves changes and they are reflected on the profile.
7	FR-4 (View Other User Profiles)	- User is logged in. - The other user has an account.	Verify users can view other users' profiles. 1) The user selects another attendee’s profile.	The system displays the other user’s profile.
8	FR-5 (Message User)	- Database is connected. - User is logged in. - The recipient user has an account.	Verify messages can be sent and received. 1) The user selects another attendee’s profile. 2) The user sends a message.	The message is sent and the recipient user can view the message.
9	FR-6 (Manage Push Notifications)	-The user is logged in. - Device notifications are enabled.	Verify user availability to toggle notification preferences 1) Open the Settings screen. 2) Select “Manage Notifications.” 3) Toggle each category of notifications. 4) Tap “Save Changes.” 5) Query the database for the user’s notification preferences. 6) Confirm that the stored	The system updates the user’s notification preferences and confirms with a message.

			preferences match the user's selections.	
10	FR-7 (Filter Presentations)	- The user must be logged into their account. - All presentations are properly updated in the database.	1) Select different filters. 2) Click on "search". 3) Check that all the results returned match the filters selected. 4) Repeat for different filters and make sure they work as well.	The filtered list matching the selected filter tags is shown to the user.
11	FR-8 (Access Presentation Materials)	- The user must be logged into an account. - The user must navigate to the list of events. - The user must select an event.	Verify that the user is able to select the links associated with a presentation. 1) Select a presentation located on the events list tab. 2) Click on the link containing the presentation materials. 3) Verify the user is redirected to the Google Drive folder webpage.	The user will be redirected to the appropriate link using their web browser.
12	FR-9 (Add Presentation to Personal Calendar)	- The user must be logged into an account. - The user must navigate to the list of events.	Verify that the "favorited" presentation is added to the user's personal calendar. 1) Select a presentation on the events list tab. 2) "Favorite" the selected presentation. 3) Verify that the presentation is added to the user's personal calendar.	The selected presentation that is "favorited" will be added to the user's personal calendar.
13	FR-10 (Access Personal Calendar)	- The user is logged in.	Verify user can access their calendar and view event details. 1) Open the welcome screen. 2) Select "My Calendar". 3) Click on any saved event.	The system successfully displays the details of the saved event.
14	FR-11 (Navigate to Event	- The user is logged in.	Verify geographic navigation to an event location.	The external map application opens

	Location)	- Events are stored with associated addresses in the database.	1) Open the Events or Calendar screen. 2) Select an event. 3) Tap the map icon. 4) Choose a navigation application when prompted.	with the event location as the destination.
15	FR-12 (Comment on Event)	- User is logged in. - Database is connected. - Event exists in the database.	Verify that the comment is visible to other users. 1) The user selects an event. 2) The user types a comment and clicks “Post.” 3) Another user logs in to view the comment.	The system displays the comment to other users.
16	FR-13 (Access Previous Years’ Presentations)	- User is logged in. - Past presentations are linked.	1) User selects “previous conferences.” 2) User selects the conference year.	The system links to the Google Drive of the chosen year.
17	FR-14 (Create Presentation)	- Administrator is logged in - Application is connected to the database - Presentation details are filled out correctly	1) The administrator selects “create new presentation”. 2) The administrator fills out the necessary details. 3) The administrator clicks “save”.	The presentation is properly added to the database and the conference schedule.
18	FR-15 (Edit Presentation)	- Administrator is logged in - Application is connected to the database	1) The administrator selects a presentation. 2) The administrator clicks “edit”. 3) The administrator changes the details they wish to change. 4) The administrator clicks “save”.	The presentation is properly updated in the database and the conference schedule
19	FR-16 (Delete Presentation)	- Administrator is logged in - Application is connected to the	1) The administrator clicks a presentation. 2) The administrator clicks “delete”.	The presentation is properly removed from the conference

		database		schedule, but still exists within the database to be found later, as needed.
20	FR-17 (Delete User)	- The user is logged into an administrator account.	Verify administrator can delete user account. 1) Open the administrator screen. 2) Select user from list. 3) Click “Delete User”. 4) Click “Confirm”.	The system successfully removes user from the database.
21	FR-18 (Send Push Notifications)	- The user is logged into an administrator account.	Verify administrator can write notifications. 1) Open the administrator screen. 2) Click on the "Announcements" section. 3) Click “New”.	The system successfully displays a blank field for the administrator to write a message.
22	FR-18 (Send Push Notifications)	- Administrator has clicked “New” in announcements section of administrator screen	Verify administrators can send notifications to users. 1) Enter notification title and message. 2) Click “Send”.	The system successfully sends notification to all users that have notifications enabled.

3. Non-Functional Requirements

This section defines the non-functional requirements (NFRs) of the IWAC Conference App. In contrast to the functional requirements detailed in Section 2, this section outlines the quality attributes and operational standards that govern the app’s reliability, usability, and security. Table 3 uniquely identifies and prioritizes the NFRs to indicate each one’s relative importance to stakeholders. Clear descriptions and corresponding system and acceptance tests

are provided to ensure each requirement can be verified and traced through development. Altogether, these NFRs establish the performance expectations and quality benchmarks necessary for a stable, secure, and user-friendly conference application.

Table 3 – Nonfunctional Requirements (NFRs)

NFR ID#	**Priority (1 = lowest, 5 = highest)**	**Description**	**Test**
NFR-1	5	The system shall display the additional session information within 2 seconds of a user's selection, 95% of the time.	Measure the response time for 50 random session selections and verify 95% load within 2 seconds.
NFR-2	4	The system shall store user feed filters so that they persist across sessions without data loss.	Verify through testing that filtering settings remain consistent after logout and application restart for 50 sessions.
NFR-3	5	User passwords shall be hashed and salted on the backend server using a secure algorithm.	Generate 500 sample passwords and verify that each is stored only as a salted hash in the database.
NFR-4	4	Administrative tasks including uploading, editing, and deleting conference data shall be completable without training by a new administrator within 15 minutes.	Conduct usability testing with Heather Falconer to measure tasks completion times and error rates on her first use of the system.
NFR-5	5	Changes to events shall be correctly synchronized between the application and the database.	Verify through testing that changes to either the application or database automatically update each other.
NFR-6	5	The application shall maintain 98% uptime throughout the conference.	Conduct stress testing, as well as uptime monitoring.
NFR-7	3	The system shall display the filtered list of panels within 2 seconds of setting the filters, 95% of the time.	Test the response time for 10 different accounts, 5 different times, and test to see if 95% of them show results within 2 seconds (and whether sequential searches affect it).

NFR-8	4	The user shall be capable of using the application to its completion without extensive tutorials, within 20 minutes of downloading it.	Conduct user testing and gather information on user confidence in using the application after 20 minutes of usage.
NFR-9	5	The application shall run on both Apple iOS 16+ and Android 12+ cellphone operating systems.	The application will be tested on both of these operating systems to ensure all features function correctly.
NFR-10	5	Only users that have paid and registered for the IWAC will be able to create accounts.	We will create an account for a user that is registered for the IWAC and an account for a user that is not registered.

4. User Interface

See the User Interface Design Document for the IWAC Conference App.

5. Deliverables

All deliverable items to be produced over the course of the project are outlined in Table 5. Each row details the format, method of delivery, and expected delivery date of every deliverable. All electronic files will be uploaded and maintained through PenUltimate's GitHub repository.

Table 5 – Project Deliverables

Deliverable Item	**Format**	**Method of Delivery**	**Expected Delivery Date**
System Requirements Specification (SRS)	Hard Copy & Electronic File	Electronic file uploaded to GitHub Repository	October 29, 2025
System Design Document (SDD)	Hard Copy & Electronic File	Electronic file uploaded to GitHub Repository	November 17, 2025
User Interface Design	Hard Copy &	Electronic file uploaded to	December 3, 2025

Document (UIDD)	Electronic File	GitHub Repository	
Critical Design Review Document (CDRD)	Hard Copy & Electronic File	Electronic file uploaded to GitHub Repository	December 19, 2025
User Manual (UM)	Hard Copy & Electronic File	Electronic file uploaded to GitHub Repository	Spring 2026
Administrator Manual (AM)	Hard Copy & Electronic File	Electronic file uploaded to GitHub Repository	Spring 2026
Biweekly Status Reports	Hard Copy & Electronic File	Electronic file uploaded to GitHub Repository	Biweekly Through Project
Source Code	Electronic File	Uploaded to GitHub Repository	Spring 2026
Executable Program	Electronic File	Uploaded to GitHub Repository	Spring 2026

6. Open Issues

The following issues have been identified during the requirements and design phases but remain unresolved. These issues will require further risk analysis and decision-making later in the development process to ensure successful system implementation.

6.1 Scalability

It is not yet determined what level of user capacity the system is required to support as the IWAC Conference continues to expand. The specific performance and responsiveness requirements for peak usage periods remain undefined.

The projected resolution date for this issue is October 30, 2025.

6.2 Administrator Data Management

The requirements for how administrators will manage conference data have not been finalized. It is unclear what level of functionality must be provided for uploading, editing, and deleting conference data, or how ease of use will be defined.

The projected resolution date for this issue is October 30, 2025.

6.3 Data Privacy

Requirements for protecting user data have not yet been explicitly defined. Further research will illustrate what level of data protection, access control, and user consent mechanisms are required to comply with institutional and legal standards.

The projected resolution date for this issue is October 30, 2025.

Appendix A – Agreement Between Customer and Contractor

Our team PenUltimate will be responsible for developing and integrating the features outlined in this document for the IWAC Conference Application, primarily focusing on features of importance that have a priority rating of 4 or 5, implementing others should time allow. The development can be done using open source software to base our work off of, and future developers will have access to our code and the database to be able to update it in the future as needed, as our involvement with building this application will end after May 2026.

Should either party wish to make changes to this document, both parties must meet ahead of time and mutually agree on said change. If Heather Falconer is the party requesting these changes, they should reach out to our client liaison, Monica Agneta, with their requests. Similarly, should we request any changes to this document, Monica will reach out to Heather with our requests.

Signature: Brett Palmer Date: 10/29/2025

Signature: Monica Agneta Date: 10/29/2025

Signature: Rebecca [illegible] Date: 10/29/2025

Signature: [illegible] Date: 10/29/2025

Signature: George Pitt Date: 10/29/2025

Signature: H Falconer Date: 10/30/2025

Appendix B – Team Review Sign-off

All team members have signed to acknowledge they have reviewed this document and agreed on both its content and format. If team members have minor disagreements, they may state them in the comments area.

Rebecca Sonnemann
Signature: _____ Date: 10/29/2025
Comments: _____

Brett Palmer
Signature: _____ Date: 10/29/2025
Comments: _____

Monica Agneta
Signature: _____ Date: 10/29/2025
Comments: _____

Ben Caras
Signature: _____ Date: 10/29/2025
Comments: _____

George Pitt
Signature: _____ Date: 10/29/2025
Comments: _____

Appendix C – Document Contributions

This section outlines the individual contributions of each team member to the development of this document. Each subsection details the specific sections written, figures created, and revisions made by each contributor.

Brett

Brett wrote the introductions for sections 2 and 3 and wrote section 6. He wrote FR-1, FR-2, FR-6, FR-11, NFR-3, and NFR-4, including the corresponding tests. Also, he made Figure 2.1. Lastly, Brett revised the document to maintain a logical structure and consistent format for headers, tables, diagrams, and labels.

Brett estimates that he did 20% of the work for this document.

Monica

Monica wrote section 1.1 and completed section 5. She wrote FR-3, FR-10, FR-17, FR-18, NFR-1 and NFR-2, including the corresponding tests.

Monica estimates that she did 20% of the work for this document.

George

George wrote the introduction for section 1 and completed Appendix C. He wrote FR-7, FR-14, FR-15, FR-16, NFR-7 and NFR-8.

George estimates that he did 20% of the work for this document.

Rebecca

Rebecca wrote the introduction for section 1.3 and wrote 1.2 and Appendix B. She wrote FR-4, FR-5, FR-12, FR-13, NFR-5, and NFR-6. She also wrote the corresponding tests associated with the functional requirements. She made the profile management and messaging use case diagram.

Rebby estimates that she did 20% of the work for this document.

Ben

Ben wrote the entirety of section 1.4 as well as created the following figures: 1.4, 2.3, and 2.4. He also wrote FR-8, FR-9, NFR-9, and NFR-10 as well as wrote the tests associated with the functional requirements he wrote. He also created the revision history section of the document. Ben created all requirements or use cases initially and the entire team revised them and decided on which ones were necessary.

Ben estimates that he did 20% of the work for this document.

Appendix E - System Design Document

System Design Document

IWAC Conference Application for Heather Falconer

PenUltimate

Brett Palmer, Monica Agneta, George Pitt, Rebecca Sonnemann, Ben Caras

17 November 2025

Table of Contents

1 Introduction…….……...……2

1.1 Purpose of This Document……..…...…2

1.2 References……..……...….2

2 System Architecture……..…….....3

2.1 Architectural Design……..……..……..3

2.2 Decomposition Description…….……...…...5

3 Persistent Data Design……...7

3.1 Database Descriptions……....7

3.2 File Descriptions……....14

4 Requirements Matrix…….15

Appendix A – Agreement Between Customer and Contractor…….17

Appendix B – Team Review Sign-off……...18

Appendix C – Document Contributions……19

1 Introduction

This capstone project is being completed in partial fulfillment of the requirements for the B.S. in Computer Science degree for the University of Maine. The client for this project is part of the Association for Writing Across the Curriculum (AWAC), which organizes the International Writing Across the Curriculum Conference (IWAC), a multi-day event that hosts both educators and researchers. In 2027, UMaine is hosting this conference. The client is interested in having a mobile conference application that can be repurposed for future conferences. The client noted having issues with last year's application, such as having to scroll through too many events and being unable to filter through them (Whova, 2025). The client also mentioned that when a previous app was used on mobile devices, it opened up a web page that was not optimal (LineUpr, 2025). The problem that the client is trying to solve is finding an effective way to allow in-person attendees to navigate the session schedule and find the info they need, as well as creating a way for virtual attendees to have more of an interactive conference experience.

1.1 Purpose of This Document

The purpose of this System Design Document (SDD) is to detail the architectural and design specifications for the IWAC Conference Application being developed by the PenUltimate team. This document translates the functional and non-functional requirements defined in the System Requirements Specification (SRS) document into a framework that guides implementation of the product. The SDD is intended for the PenUltimate development team and organizers of the IWAC conference to ensure a shared understanding of the system's architecture. This document begins with an overview of the system architecture, followed by detailed descriptions of each function and the data flow. It then goes on to describe the user database and files intended to be used by the system. The document concludes with a table that depicts which system components satisfy which functional requirements from the SRS.

1.2 References

Association for Writing Across the Curriculum. (2025). *IWAC 2025*. https://iwac2025.lineupr.com/iwac-2025/

dbdiagram.io. (2025). Dbdiagram.io: Database relationship diagrams design tool. Holistics Software. https://dbdiagram.io/d

draw.io. (2025). *Draw.io: Free flowchart maker and diagrams online.* draw.io Ltd. https://app.diagrams.net/

Fowler, M. (1997). *UML Distilled: A brief guide to the standard object modeling language*. Addison-Wesley. http://ci.nii.ac.jp/ncid/BA65029497

LineUpr. (2025). *The Event application solution to boost your event communication – LineUpr*. LineUpr GmbH. https://lineupr.com/en

Whova. (2025). *Whova: Award-winning Event Apps and Event Management*. Whova, Inc. https://whova.com/

2 System Architecture

This section outlines the overall design of the IWAC Conference application, detailing the overall architecture and how the components interact to fulfill the functional requirements defined in the SRS. The architecture is first showcased by a layered architecture diagram created with draw.io (2025) that gives a high-level view of the data flow through the application. Complementing this, a UML class diagram created with dbdiagram.io (2025) shows the decomposition of the system into the major components, specifying variables, methods, and relationships. Together, these models illustrate both the system's logical and structural design and the user interaction with it.

2.1 Architectural Design

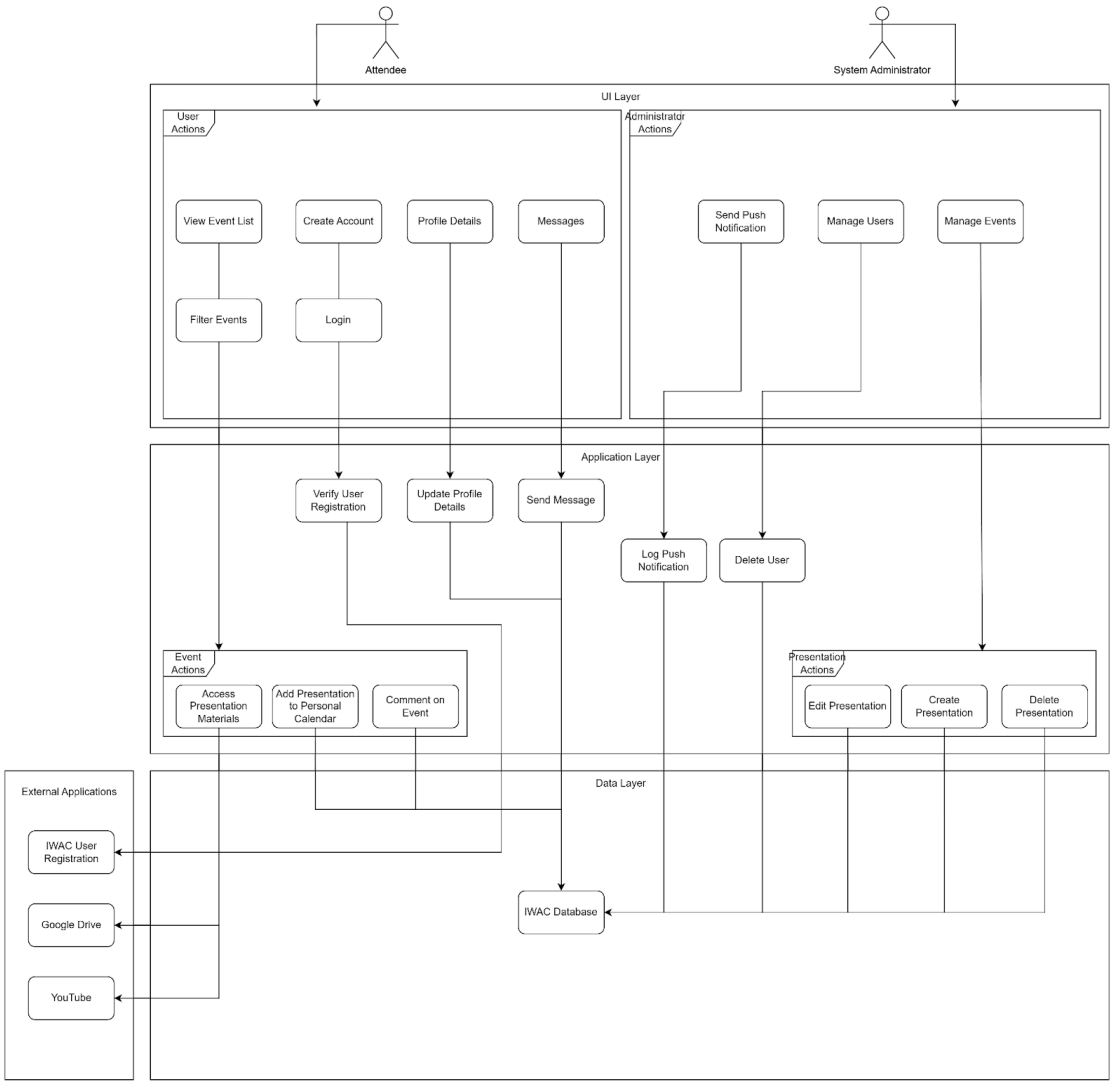

Figure 2.1.1 - Layered Architecture Diagram

Figure 2.1.1 is a layered architecture diagram with three distinct layers. The top layer is the user interface layer which contains all pages that the users and systems administrator are able to view. The application layer contains all of the different actions they are able to perform. These actions change the data within the database that is noted on the data layer of the diagram. Next to the data layer are external applications which are systems that the IWAC Application will use but are not present within the IWAC Application.

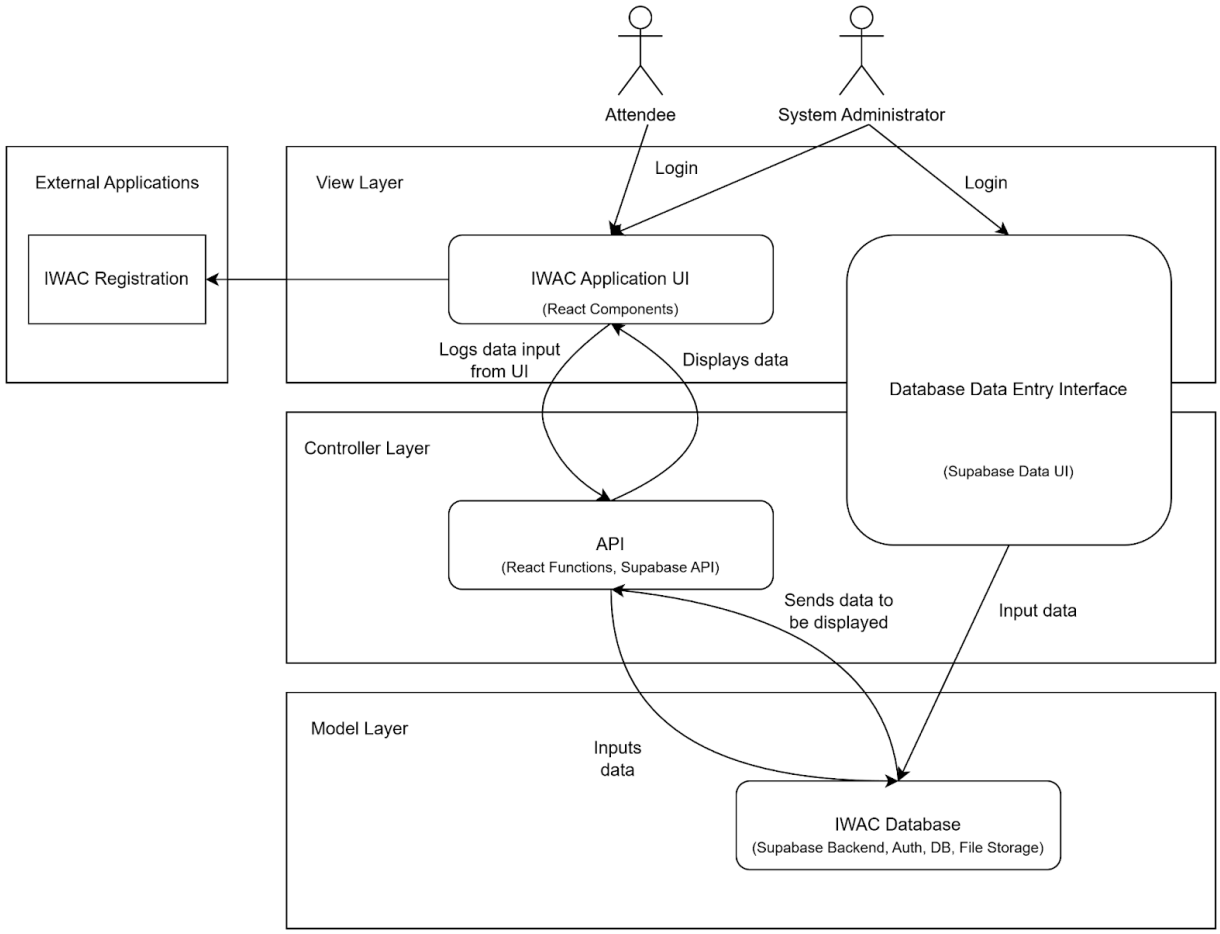

Figure 2.1.2 - MVC Logical Architecture Diagram

Figure 2.1.2 is a Model-View-Controller logical architecture diagram; it separates the system's components into three distinct layers. The View layer which contains the user interface of our application as well as the user interface used by the database which is how the client will input information into the database. The Controller layer which contains the API calls that will be made when information is entered into the user interface or when entered into the database. The Model layer contains the Supabase database that will be used by the IWAC Application.

The IWAC Application will be available on both Android and Apple operating systems, specifically on Android 12+ and Apple iOS 16+. The application will serve as a virtual environment where the attendees will be able to interact with conference events as well as conference attendees and presenters. The system will allow users to view a list of conference events and create their own personal itinerary. Users will be able to message other users, comment on presentations, and receive responses from the presenters. The information related to messaging, commenting, and creation of personal calendars as well as other key functions will be stored within our application's database. The application will also be linked to other external applications such as Google Drive which will store all presentation materials where users can be redirected to view this information, YouTube where users will be able to watch prerecorded presentations or livestreams of the presentations, and Google Maps which will help conference attendees gain a better understanding of where the events and presentations will take place.

The application will primarily be written in TypeScript using the Expo full-stack React Native framework. The system will be tested using Jest, a common testing framework for React Native. The application will use Supabase for a PostgreSQL database, an open source database software. The client will enter the information for each conference via Supabase's interface. Our team believes that TypeScript with the Expo full-stack React Native framework will work best for our application because of the fact that the application needs to be usable on both Android and iOS operating systems. The Jest framework for testing is the most understood amongst our team, and the database will allow our client to be able to input data into the application easily given the fact that Supabases UI and data entry aligns with the clients expertise.

2.2 Decomposition Description

The IWAC Application is an object-oriented system. The User represents anyone who uses the app. The Admin represents a subclass of users that can do everything a User can, but with additional privileges such as managing presentations and other user accounts. The Notification object handles push notifications sent to the user's device. The Presentation object represents a given panel being held at the conference. The Comment object is associated with the Presentation object, allowing users to leave comments. The Message object handles private communication between users, enabling one user to send a direct message to another user. Lastly, the Calendar object stores each user's favorited presentations, creating unique personalized calendars for each attendee. Note that some of these objects extend from other objects, such as Admin and User. The interconnectedness of these objects, the ability to individually alter or manage any one of them without affecting others, and the inheritance some objects gain when extending from others, explain why our application is

object-oriented. These motivations also justify our use of class diagrams to illustrate the implementation design of our system.

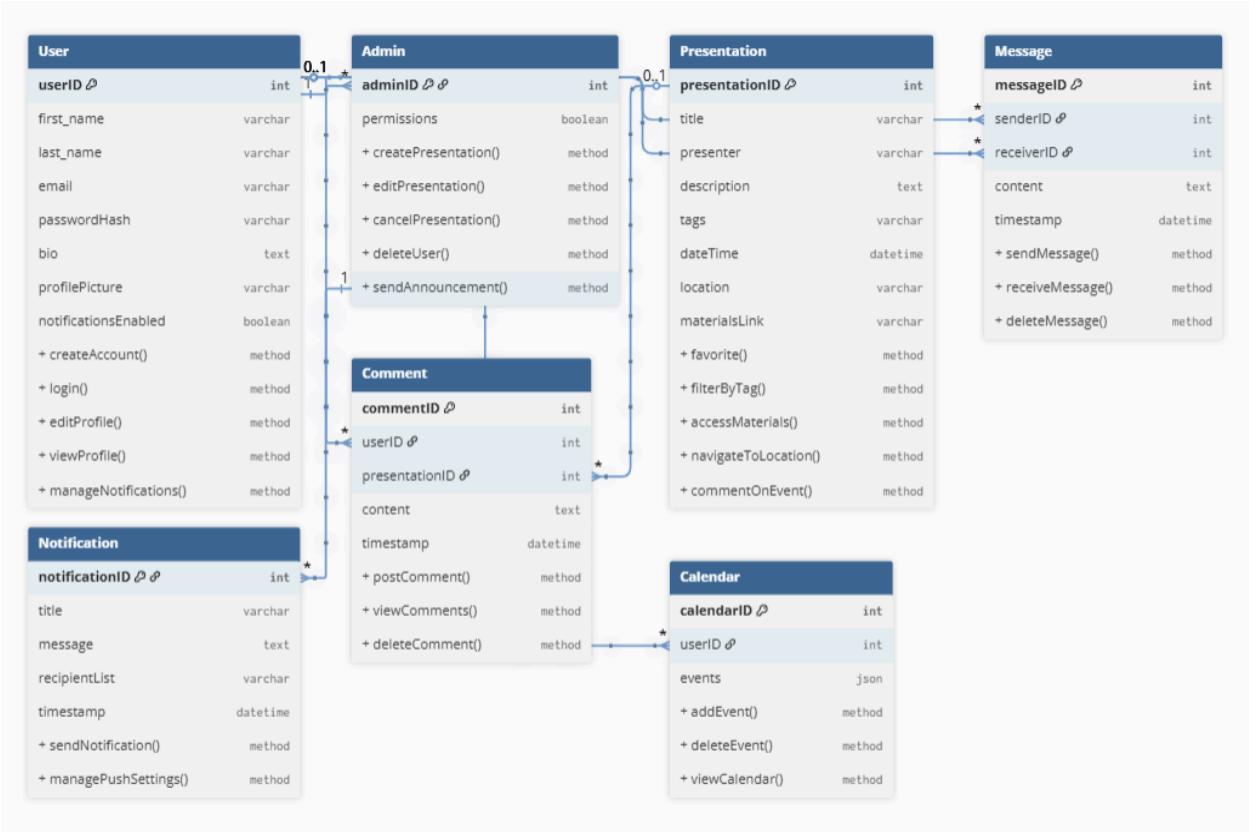

Figure 2.2 - IWAC Conference App Class Diagram

Figure 2.2 is a class diagram illustrating the various objects in our application. The labels “0”, “1”, and “*” on each side the lines connecting objects indicate how many instances of one object may be associated with another, with “*” being shorthand for “many.” For instance, many comments can be left under a single presentation, so the line between the objects shows a “*” on the comment side and a “1” on the presentation side. Furthermore, the diagram shows each object with its associated data and methods listed to the left, while the right side specifies the type of data held or whether it is a method. For example, the Comment object has a commentID to identify that specific comment, a userID to identify who made that comment, and a presentationID to indicate which presentation it was commented under, all stored as integers. It also contains the text contents of the message and the timestamp of when it was posted. Lastly, users can perform the actions of viewing, posting, and deleting comments, which are represented by the methods viewComment(), postComment(), and deleteComment(), respectively.

A Model View Controller design pattern was used for the system, as it allows us to separate the application's logic into distinct components. This separation makes the system easier to maintain and update in the future, and allows users to view and modify data in a controlled manner as they interact with the application's screens. First, the Model directly manages our application's data and the rules governing that data, ensuring that all stored information is consistent. The View is the interface that users see and interact with in the app, such as screens containing search bars, buttons, input fields, and other visual components. The Controller links the two, taking data in the Model and displaying it in the View, or receiving input from the View and updating data in the Model. Using the previous example of the Comment object, the commentID, userID, and presentationID values would all be stored in our database, which is the Model. Methods such as addEvent(), viewCalendar(), and viewEvent() would belong to the Controller, as they retrieve and update the appropriate information based on events. The Events screen would serve as the View, since this is what users will interact with in order to comment under a presentation.

3 Persistent Data Design

The persistent data design of the IWAC system defines how information is stored, organized, and accessed across the application. This design ensures consistency and scalability for managing events and user interactions. It includes the entity relationships and file structures that collectively create the system's functionality. The following sections describe the database descriptions and associated files used to maintain data persistence within the IWAC Application.

3.1 Database Descriptions

Figure 3.1 presents the entity-relationship diagram for the IWAC database used by the IWAC Application. It provides a high-level overview of the system's data structure and inner table relationships. Each table in the diagram represents an entity of our conference management system, such as users, conferences, and events. The lines between the tables indicate the relationships between the entities, such as one-to-one, one-to-many, and many-to-many. Within each table, a key icon designates the primary key that uniquely identifies each record of the table. Also, the link icon marks foreign keys that are used to link related tables. Together, these visual cues clarify how data flows between components of the system.

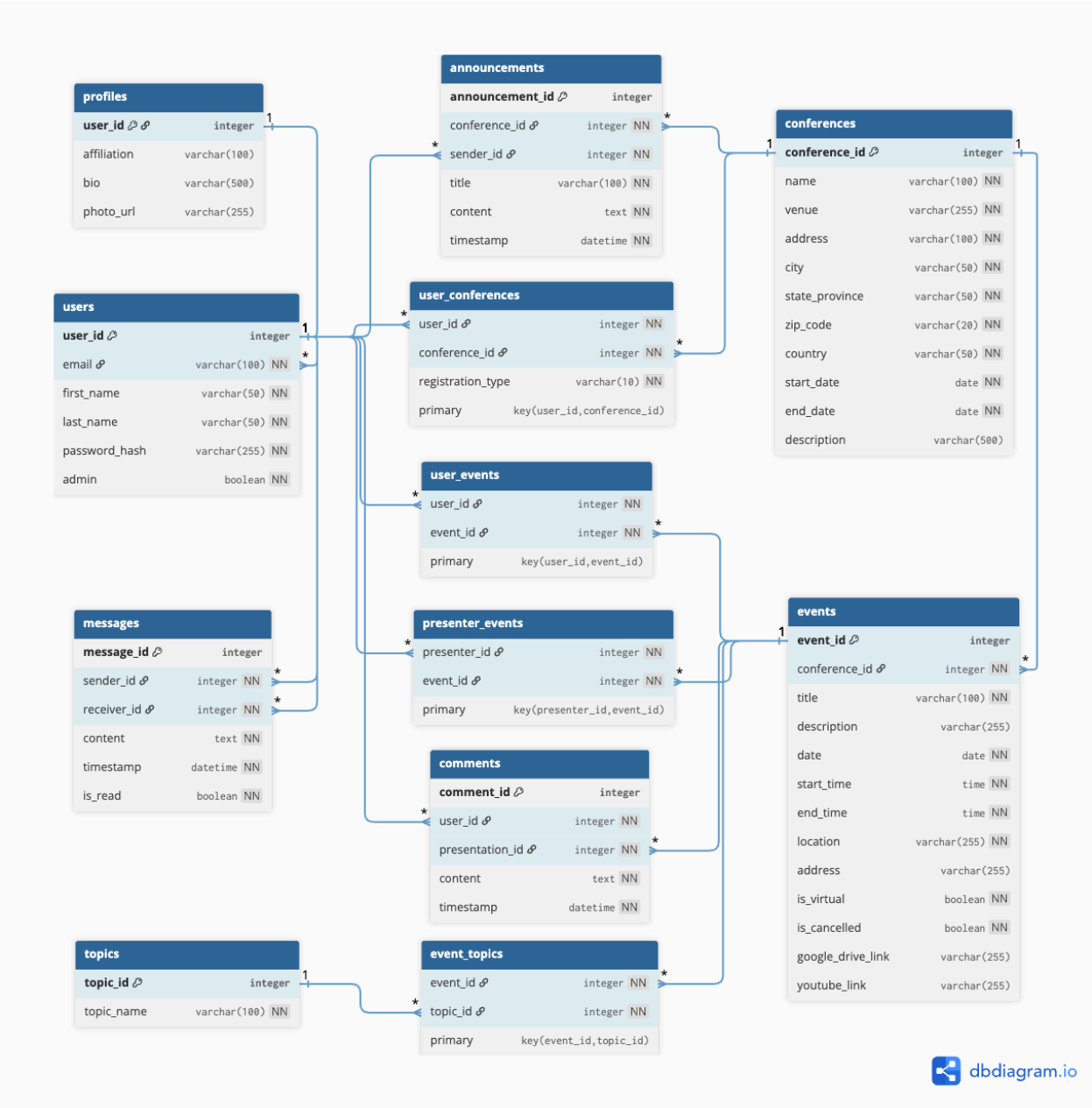

Figure 3.1 Entity-Relationship Diagram for the IWAC Database

The following tables (Table 3.1.1 through Table 3.1.12) provide a detailed breakdown of each entity from Figure 3.1, outlining the structure and attributes of the IWAC database. Each field is presented with its record type, size, description, and whether it is required. Primary keys are shown in bold, while foreign keys are shown in italics. Together, these tables describe how each database component stores and organizes information that is essential to manage the IWAC conferences.

Table 3.1.1 Users Table Definition

Field	Type	Size	Description	Required
user_id	INT	-	Auto-incremented unique identifier for each user.	Yes
email	VARCHAR	100	User's conference-registered email. Must be unique.	Yes
first_name	VARCHAR	50	User's first name.	Yes
last_name	VARCHAR	50	User's last name.	Yes
password_hash	VARCHAR	255	Hashed and salted password for login.	Yes
admin	BOOLEAN	-	True if the user is an admin, false otherwise.	Yes

Table 3.1.2 Profiles Table Definition

Field	Type	Size	Description	Required
user_id	INT	-	References Users(user_id). Unique identifier for each user.	Yes
affiliation	VARCHAR	100	User's institution or organizational affiliation.	No
bio	VARCHAR	500	Short biography for the user to describe themselves.	No
photo_url	VARCHAR	255	Link to the user's optional profile picture.	No

Table 3.1.3 Conferences Table Definition

Field	Type	Size	Description	Required
conference_id	INT	-	Auto-incremented unique identifier for each conference.	Yes
name	VARCHAR	100	Official conference title.	Yes
venue	VARCHAR	255	Name of the venue where the conference is held.	Yes

address	VARCHAR	100	Street address of the venue.	Yes
city	VARCHAR	50	City of the venue.	Yes
state_province	VARCHAR	50	State or province of the venue.	Yes
zip_code	VARCHAR	20	ZIP code of the venue.	Yes
country	VARCHAR	50	Country name of the venue	Yes
start_date	DATE	-	Start date of the conference	Yes
end_date	DATE	-	End date of the conference	Yes
description	VARCHAR	500	Brief overview of the conference and its theme	No

Table 3.1.4 UserConferences Table Definition

Field	Type	Size	Description	Required
user_id	INT	-	References Users(user_id). Unique identifier for each user.	Yes
conference_id	INT	-	References Conferences(conference_id). Unique identifier for each conference.	Yes
registration_type	ENUM (‘in_person’, ‘virtual’)	10	Indicates if the user has registered for this conference in person or online.	Yes

Table 3.1.5 Events Table Definition

Field	Type	Size	Description	Required
event_id	INT	-	Auto-incremented unique identifier for each event.	Yes
conference_id	INT	-	References Conferences(conference_id). Unique identifier for each conference.	Yes
title	VARCHAR	100	Title of the event.	Yes
description	VARCHAR	255	Brief overview of the event and its theme.	No

date	DATE	-	Scheduled date of the event.	Yes
start_time	TIME	-	Start time of the event.	Yes
end_time	TIME	-	End time of the event.	Yes
location	VARCHAR	255	Building name and room name (e.g., “Neville Hall, Room 116”). Is “Virtual” if held entirely online.	Yes
address	VARCHAR	255	Street address of the event, if in-person. The event’s other location attributes can be assumed to be the same as the conference.	No
is_virtual	BOOLEAN	-	True if users can participate in the event virtually, false by default.	Yes
is_cancelled	BOOLEAN	-	True if the event gets cancelled, false by default.	Yes
google_drive_link	VARCHAR	255	Link to slides and/or materials.	No
youtube_link	VARCHAR	255	Link to livestream or video.	No

Table 3.1.6 UserEvents Table Definition

Field	Type	Size	Description	Required
user_id	INT	-	References Users(user_id). Unique identifier for each user.	Yes
event_id	INT	-	References Events(event_id). Unique identifier for each event.	Yes

Table 3.1.7 PresenterEvents Table Definition

Field	Type	Size	Description	Required
presenter_id	INT	-	References Users(user_id). Unique identifier for each user, in this case a presenter.	Yes
event_id	INT	-	References Events(event_id). Unique identifier for each event.	Yes

Table 3.1.8 Comments Table Definition

Field	Type	Size	Description	Required
comment_id	INT	-	Auto-incremented unique identifier for each comment.	Yes
user_id	INT	-	References users(user_id). The user who sent the comment.	Yes
event_id	INT	-	References events(event_id). The event being commented on.	Yes
content	VARCHAR	1000	Text content of the comment.	Yes
timestamp	DATETIME	-	Date and time when the comment was posted.	Yes

Table 3.1.9 EventTopics Table Definition

Field	Type	Size	Description	Required
event_id	INT	-	References Events(event_id). Unique identifier for each event.	Yes
topic_id	INT	-	References Topics(topic_id). Unique identifier for each topic.	Yes

Table 3.1.10 Topics Table Definition

Field	Type	Size	Description	Required
topic_id	INT	-	Auto-incremented unique identifier for each topic.	Yes
topic_name	VARCHAR	100	Label or subject area describing the description of an event.	Yes

Table 3.1.11 Messages Table Definition

Field	Type	Size	Description	Required
message_id	INT	-	Auto-incremented unique identifier for each message.	Yes
sender_id	INT	-	References users(user_id). The user who sent the message.	Yes
reciever_id	INT	-	References users(user_id). The user who receives the message.	Yes
content	VARCHAR	1000	Text content of the message.	Yes
timestamp	DATETIME	-	Date and time when the message was sent.	Yes
is_read	BOOLEAN	-	True if the receiver has opened the message; otherwise false.	Yes

Table 3.1.12 Announcements Table Definition

Field	Type	Size	Description	Required
announcement_id	INT	-	Auto-incremented unique identifier for each announcement or push notification.	Yes
conference_id	INT	-	References Conferences(conference_id). Identifies which conference this announcement belongs to.	Yes
title	VARCHAR	150	Short title or headline of the announcement.	Yes
content	VARCHAR	1000	Main body text of the announcement message.	Yes
timestamp	DATETIME	-	Date and time when the announcement was created or sent.	Yes
sender_id	INT	-	References Users(user_id). Identifies the user (an admin) who posted the announcement.	Yes

3.2 File Descriptions

Table 3.2.1 and Table 3.2.2 describe the files used in the IWAC system, namely the banner images that are used for each conference and the profile pictures that users may optionally upload. Each file attribute is explained with its name, data type, size, description, and optionality.

Table 3.2.1 Banner Image File Description

Field	Type	Size	Description	Required
file_name	VARCHAR	255	Actual file name stored in the server directory.	Yes
file_path	VARCHAR	500	Full relative or absolute path to the banner file.	Yes
conference_id	INT	-	References Conferences(conference_id). Identifies which conference this banner belongs to.	Yes
upload_date	DATETIME	-	Date and time the banner was uploaded.	Yes
uploaded_by	INT	-	References Users(user_id). Identifies the admin who uploaded the banner.	Yes
file_type	VARCHAR	10	MIME type of the file.	Yes
file_size	INT	-	Size of the file in bytes.	Yes
width	INT	-	Image width in pixels.	Yes
height	INT	-	Image height in pixels.	Yes

Table 3.2.2 Profile Picture File Description

Field	Type	Size	Description	Required
file_name	VARCHAR	255	File name of the uploaded picture.	Yes
file_path	VARCHAR	500	Full relative or absolute path to the image.	Yes
user_id	INT	-	References Users(user_id). The user who owns the profile picture.	Yes

upload_date	DATETIME	-	Date and time the picture was uploaded.	Yes
file_type	VARCHAR	10	MIME type of the file.	Yes
file_size	INT	-	Size of the file in bytes.	Yes
width	INT	-	Image width in pixels.	Yes
height	INT	-	Image height in pixels.	Yes

4 Requirements Matrix

Table 4 is a Requirements Matrix Table that shows each of our functional requirements listed on the left, previously outlined in our SRS. The middle column holds the corresponding methods that address each of these issues, while the right column holds the class that uses the corresponding method in that row. Having each Functional Requirement be fulfilled by at least one method ensures that we satisfy all Functional Requirements in the development of the IWAC Application. These methods, and the objects that implement them are diagramed in Figure 2.2.

Table 4 Requirements Matrix Table

Functional Requirement Number and Name	**Corresponding Method(s)**	**Class(es) in Which the Method is Used.**
1- Create Account	createAccount()	User
2- Login to Account	login()	User
3- Edit Profile Details	editProfile()	User
4- View Other User Profiles	viewProfile()	User
5- Message User	sendMessage()	Message
6- Manage Push Notifications	managePushSettings()	Notification
7- Filter presentations	filterByTag()	Presentation
8- Access Presentation Materials	accessMaterials()	Presentation
9- Add Presentation to Personal Calendar	favorite()	Presentation
10- Access Personal Calendar	viewCalendar()	Calendar

11- Navigate to Event Location	navigateToLocation()	Presentation
12- Comment on Event	commentOnEvent()	Presentation
13- Access Previous Years' Presentations	filterByTag()	Presentation
14- Create Presentation	createPresentation()	Admin
15- Edit Presentation	editPresentation()	Admin
16- Delete Presentation	cancelPresentation()	Admin
17- Delete User	deleteUser()	Admin
18- Send Push Notifications	sendAnnouncement() / sendNotification()	Admin, Notification

Appendix A – Agreement Between Customer and Contractor

The PenUltimate team is responsible for developing the application as outlined within this document. The development includes application features, user interface requirements, user and administrator actions, data storage, and database management. The development can be done using open source software to base our work off of, and future developers will have access to our code and the database to be able to update it in the future as needed, as our involvement with building this application will end after May 2026.

Should either party wish to make changes to this document, both parties must meet ahead of time and mutually agree on said change. If Heather Falconer is the party requesting these changes, they should reach out to our client liaison, Monica Agneta, with their requests. Similarly, should we request any changes to this document, Monica will reach out to Heather with our requests.

Signature: _____ Date: 11/17/2025

Signature: _____ Date: 11/17/2025

Signature: _____ Date: 11/17/2025

Signature: _____ Date: 11/17/2025

Signature: _____ Date: 11/17/2025

Signature: _____ Date: 11/20/2025

Appendix B – Team Review Sign-off

All team members have signed to acknowledge they have reviewed this document and agreed on both its content and format. If team members have minor disagreements, they may state them in the comments area.

Rebecca Sonnemann

Signature: _____ Date: 11/17/2025

Comments: _____

_____

Brett Palmer

Signature: _____ Date: 11/17/2025

Comments: _____

_____

Monica Agneta

Signature: _____ Date: 11/17/2025

Comments: _____

_____

Ben Caras

Signature: _____ Date: 11/17/2025

Comments: _____

_____

George Pitt

Signature: _____ Date: 11/17/2025

Comments: _____

_____

Appendix C – Document Contributions

Brett

Brett designed the database structure and represented it in diagram 3.1, created tables 3.1.1 through 3.1.8, and wrote the text within section 3.1. He did general editing for consistent formatting and grammar within the document.

Brett estimates that he did 20% of the work for this document.

Monica

Monica wrote section 1, wrote the introduction for section 2, and created the decomposition description diagram, figure 2.2.

Monica estimates that she did 20% of the work for this document.

George

George wrote the diagram description for section 2.2, as well as section 4.

George estimates that he did 20% of the work for this document.

Rebecca

Rebecca wrote the introductions for section 3 and section 3.2, created tables 3.1.9 through 3.1.12, and did the tables for section 3.2.

Rebby estimates that she did 20% of the work for this document.

Ben

Ben wrote section 2.1 and the diagrams associated with this section. Ben also wrote Appendix A.

Ben estimates that he did 20% of the work for this document.

Appendix F - User Interface Design Document

User Interface Design Document

IWAC Conference Application for Heather Falconer

PenUltimate

Brett Palmer, Monica Agneta, George Pitt, Rebecca Sonnemann, Ben Caras

3 December 2025

Table of Contents

1 Introduction…….……...……...2

1.1 Purpose of This Document…….……...2

1.2 References…….……...…….…2

2 User Interface Standards…….……..3

2.1 General Layout and Common Components…….….…3

2.2 Navigation…….....4

2.3 General Error Handling…….....5

3 User Interface Walkthrough…….…….5

3.1 Create Account…..…….....5

3.2 Login……...…….…….6

3.3 Events……..……..7

3.4 Itinerary…….9

3.5 Profile & Edit Profile..……..……10

3.6 Messages……...12

3.7 Notifications & Manage Notifications.……13

4 Data Validation…….……...……..16

4.1 Administrator Data Entry……..16

4.2 User Data Entry……...…..16

Appendix A – Agreement Between Customer and Contractor……...…..18

Appendix B – Team Review Sign-off……...……19

Appendix C – Document Contributions…….……...20

1 Introduction

This capstone project is being completed in partial fulfillment of the requirements for the B.S. in Computer Science degree for the University of Maine (UMaine). The client for this project is part of the Association for Writing Across the Curriculum (AWAC), which organizes the International Writing Across the Curriculum Conference (IWAC), a multi-day event that hosts both educators and researchers. In 2027, UMaine is hosting this conference. The client is interested in having a mobile conference application that can be repurposed for future conferences. The client noted having issues with last year's application, such as having to scroll through too many events and being unable to filter through them (Whova, 2025). The client also mentioned that when a previous app was used on mobile devices, it opened up a web page that was not optimal (LineUpr, 2025). The problem that the client is trying to solve is finding an effective way to allow in-person attendees to navigate the session schedule and find the info they need, as well as creating a way for virtual attendees to have a more interactive conference experience.

1.1 Purpose of This Document

The purpose of the User Interface Design Document (UIDD) is to define and describe the user interface for the IWAC Conference Application, developed for the client, Heather Falconer, as part of the International Writing Across the Curriculum Conference (IWAC). This document specifies how attendees will interact with the application and provides a detailed overview of all visual components and interface behaviors. This document also provides all diagrams, interface standards and screen walkthroughs needed to guide implementation. The UIDD serves as the reference for the design and layout of the application, ensuring that there is a consistent understanding of the system's appearance and functionality.

1.2 References

Association for Writing Across the Curriculum. (2025). *IWAC 2025*. https://iwac2025.lineupr.com/iwac-2025/

LineUpr. (2025). *The Event application solution to boost your event communication – LineUpr*. LineUpr GmbH. https://lineupr.com/en

React Native. (2025). *React Native · A framework for building native apps using React.* Meta Platforms, Inc. https://reactnative.dev/

Supabase. (2025). *The Open Source Firebase Alternative*. Supabase Inc. https://supabase.com/

Whova. (2025). *Whova: Award-winning Event Apps and Event Management*. Whova, Inc. https://whova.com/

Wireframe.cc | The go-to wireframing tool. (2025). Wireframe.cc. https://wireframe.cc/

2 User Interface Standards

This section details the key user interface standards throughout the IWAC Conference Application. It includes a low-level screen layout of the homepage, the common UI components that will be used across screens, the overall navigation structure for users, and the general error-handling logic. These standards establish a clear and cohesive framework for how the UI will be developed.

2.1 General Layout and Common Components

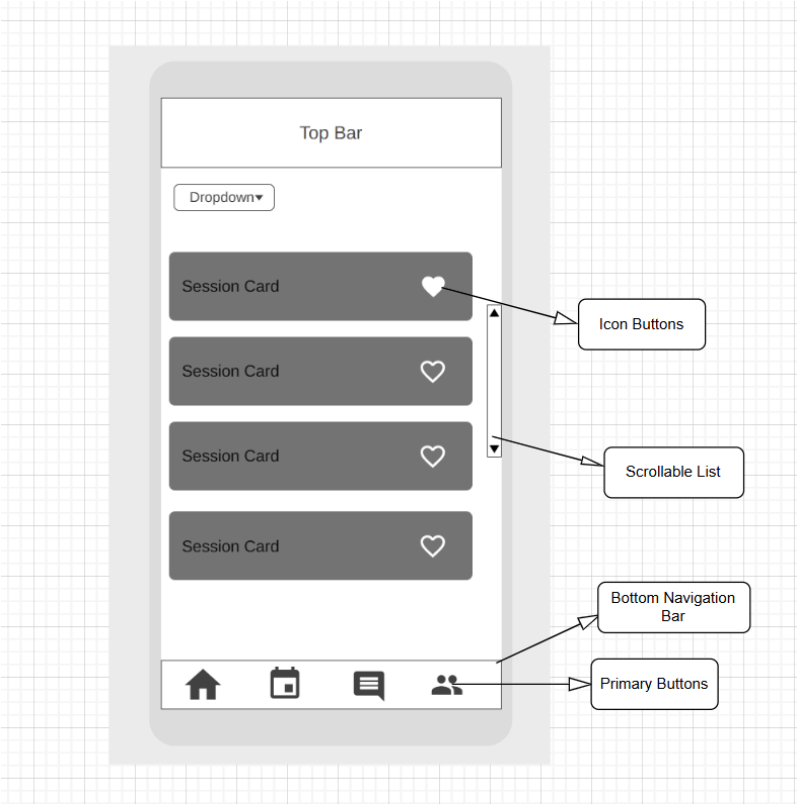

Figure 2.1.1 - Home Screen Layout (Wireframe.cc, 2025)

The home page layout of the IWAC Conference App has several key features that are listed in Table 2.1 and diagrammed in Figure 2.1. The very top of the page will consistently be the Top Application Bar which remains fixed and provides the page title along with a back button on secondary pages. Directly below that is the filter area that contains a dropdown menu for users to choose which conference sessions they would like to have shown on the home page. The center area of the screen is the vertically scrollable section that displays all the session cards. On the initial view of the session cards, the session name, time, and a favorite button is

displayed. The session details are available by clicking on the session cardsThe bottom of the screen includes the Bottom Navigation Bar, another fixed feature that holds the icons of Home, Schedule, Messages, and People pages all of which can be easily navigated to with a click. This screen layout keeps the main content of the conference sessions the highlight of the page and ensures simple navigation to the other application features.

Table 2.1.1 - Common Components

Component	**Description**
Bottom Navigation Bar	Persistent tab at bottom of page that allows user to navigate to the corresponding main page.
Top Navigation Bar	Persistent bar at top of page that displays page name and back button in left corner.
Session Cards	Cards that contain session title, time, and “favorite” button, located on the home page and schedule page.
Primary Buttons	Main buttons on pages for major actions such as sending messages; will be rounded rectangles.
Icon Buttons	Small buttons containing icons that will be shown on the bottom navigation bar.
Scrollable List	Main feature on home page and schedule page to allow users to scroll through vertical content.
Dropdown	Component that allows the user to choose multiple options from the list; used for filtering the home page.

Furthermore, Dr. Falconer provided an AWAC Social Media Kit that will be referenced to maintain a consistent style throughout the IWAC Application. Her document included the official AWAC logo, hex codes for preferred colors, and a set of fonts. These elements will be implemented into the IWAC Application to ensure uniformity within the mobile application and to uphold consistency with other AWAC content.

2.2 Navigation

Navigation through the application will be based on the bottom navigation tab for all primary pages as well as stack-based navigation to other features within the pages. For example, a user will be able to click on a session to view the session details and attend the session. When in attendance, the user will be able to leave comments under the session that get sent to the presenter. From the messages page, users will be able to see their chat thread and from there can view the profiles of the users they message. Users will always be able to return to the home page by using a back arrow that will be in the left side of the header. This will ensure that users can move through the application smoothly.

2.3 General Error Handling

The IWAC conference application will clearly communicate errors to the user through concise pop-up messages that explain what went wrong. For example, if a user attempts to create multiple accounts with the same email, the system will prevent the action and display a message that says “An account already exists with this email. Please log in or choose a different email.” The error messages will appear near the related input fields. For system related errors such as network failure, the app will display a banner with the necessary information. Errors will also be logged on the backend for developers to diagnose issues.

3 User Interface Walkthrough

The User Interface Walkthrough section displays user interface (UI) diagrams of all major screens and describes how the user would utilize them. All of the mockups were built with React Native (2025), so they serve as the first prototype of the IWAC Application’s UI. Each section covers a main feature such as login, account creation, and event viewing. The image of each screen is accompanied with an explanation of the screen’s purpose and how a user can interact with its buttons, text boxes, and other components to achieve their goal.

3.1 Create Account

When users first open the IWAC Application, they will need to create an account in order to access the system's features. Figure 3.1.1 shows the user interface for creating an account; users will be able to enter their name, email address, and password. The user's name will be shown to other users. Users will be able to navigate to the log in screen by clicking the “Already Have an Account? Go to Login Screen” text at the bottom of the screen.

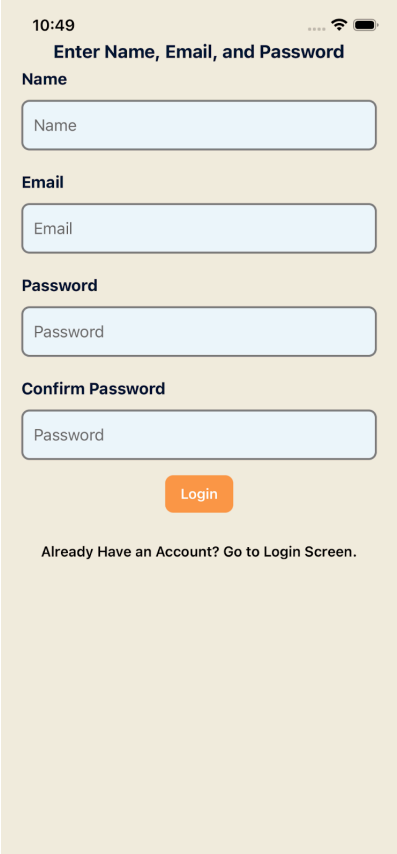

Figure 3.1.1 Create Account Screen

3.2 Login

When users open the IWAC Application after creating an account they will be able to subsequently login to their account. Figure 3.2.1 represents the login screen where users will be able to enter their email address and password. Users will be able to navigate to the create account screen by clicking the “Not Registered? Create Account” text at the bottom of the screen.

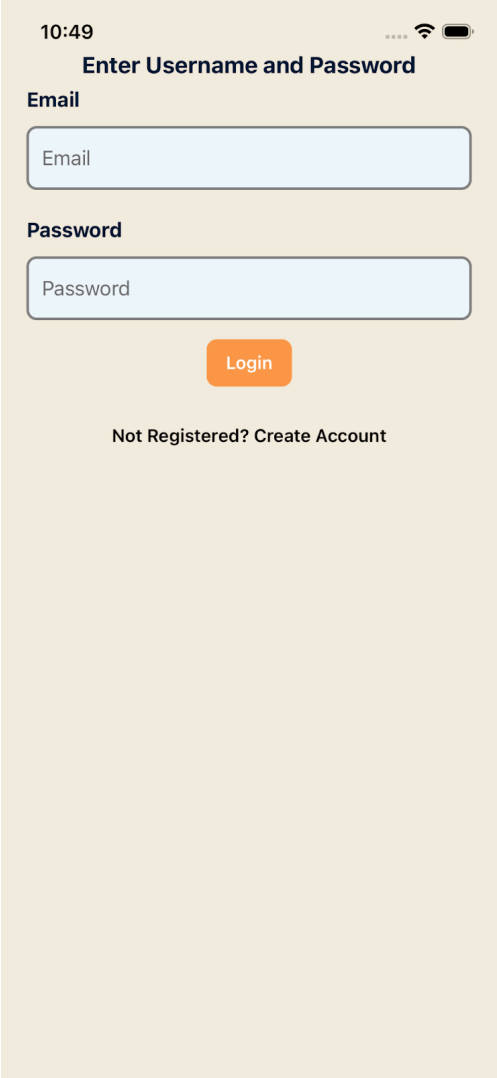

Figure 3.2.1 Login Screen

3.3 Events

The Events screen shown in Figure 3.3.1 provides an interface for users to browse and filter different conference sessions. The main Events screen includes a search bar at the top, allowing users to find sessions by title, topic, or speaker. A filter panel, shown in Figure 3.3.2, is accessible from below the search bar, which allows users to filter results by the theme. Each event has a heart icon on the top right to add the event to your itinerary.

Below the filter panel, the system displays a list of all matching events in a list. In Figure 3.3.1, it shows a few example sessions. Each event card includes the event's topic tag(s), session title, scheduled date and time, location, and the lecturer. Selecting an event card opens the full event details page, shown in Figure 3.3.3, where users can view more information about the

event and add the session to their itinerary using the heart in the top right corner. From the Events screen, users can also navigate to their Itinerary, Messages, Profile, and Notifications.

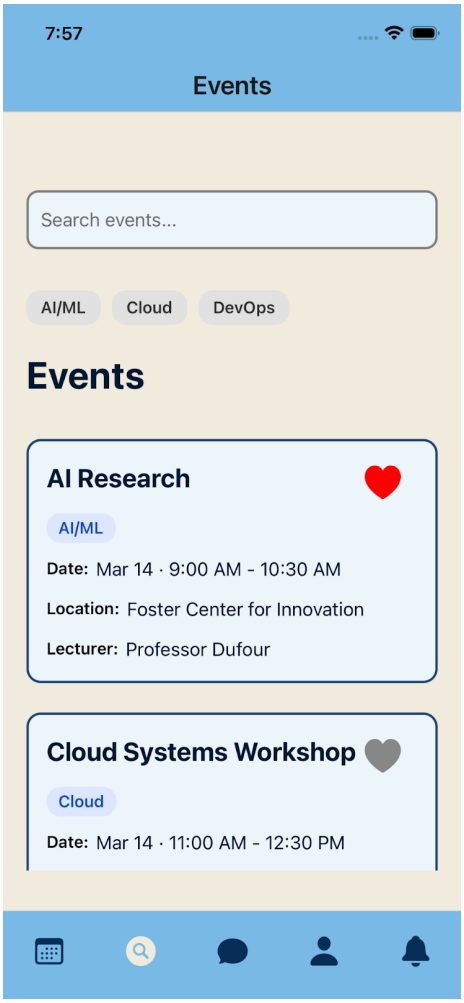

Figure 3.3.1 Events Page

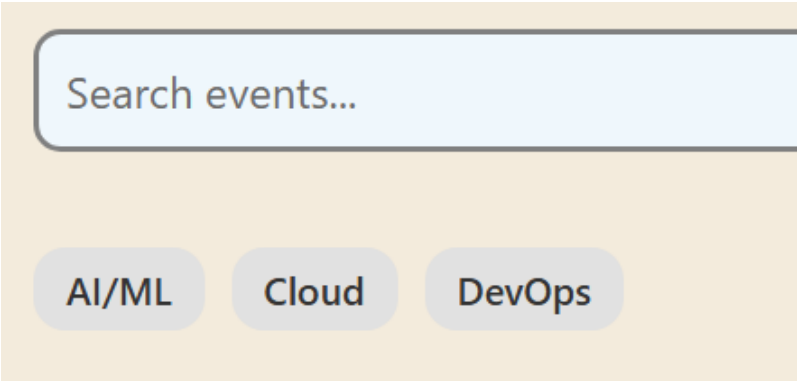

Figure 3.3.2 Filtering Panel

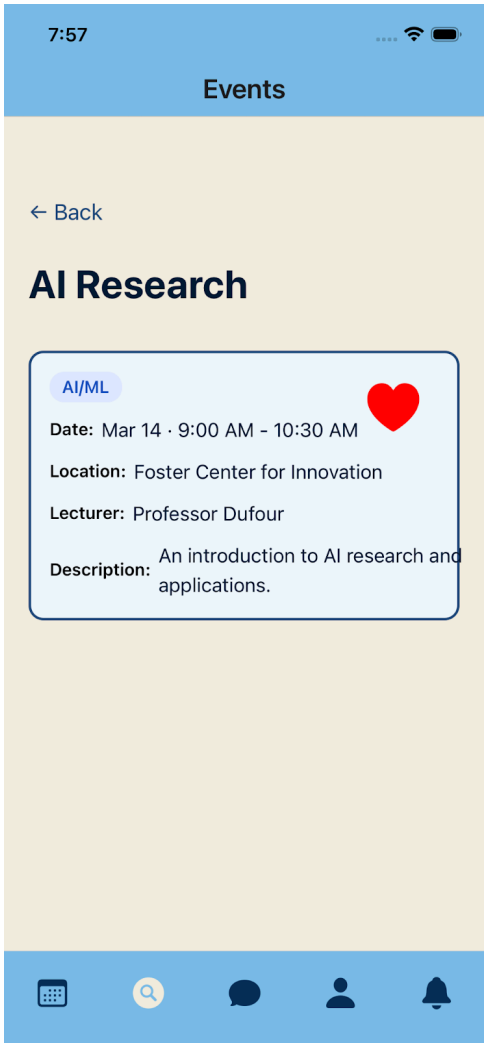

Figure 3.3.3 Event Details

3.4 Itinerary

After adding events to your itinerary, you can view the full list of events you have saved by clicking on the Itinerary button on the bottom navigation bar, represented by a calendar icon. The event cards saved to the user's Itinerary display the session time, location, and presenter. Users can easily remove events from their itinerary by clicking on the red "x" icon in the right hand corner of each event card, shown in Figure 3.4.1. Clicking on any event will open the full event details. In the top right corner of the Itinerary page, there is a "Browse More Events" button which takes the user to the Events page, where they can add events to their itinerary.

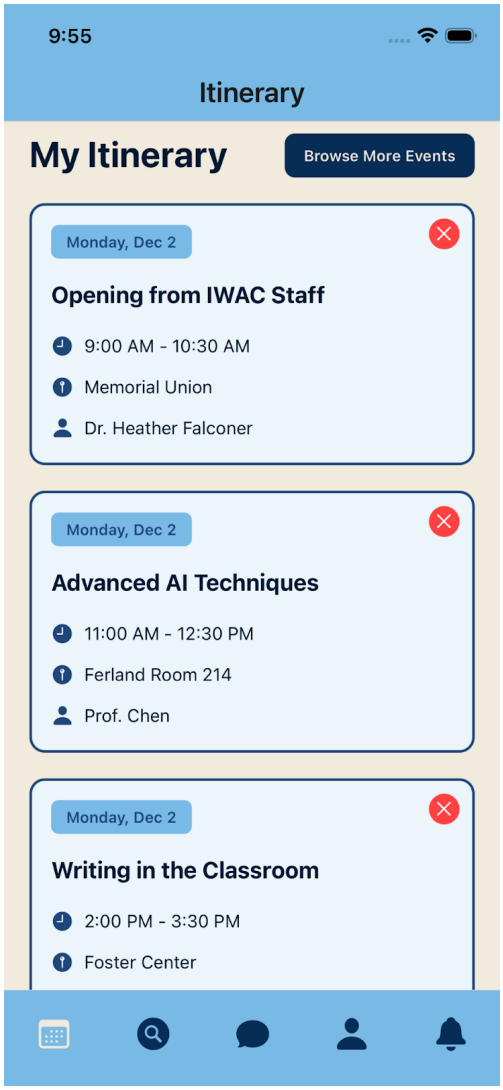

Figure 3.4.1 My Itinerary Screen

3.5 Profile & Edit Profile

Figure 3.5.1 shows the user's profile page, along with their information, including name, a short biography, interests, an "about me" section, and a profile picture. Edits to these details can be made by clicking on "Edit Profile" on the top right corner of the screen.

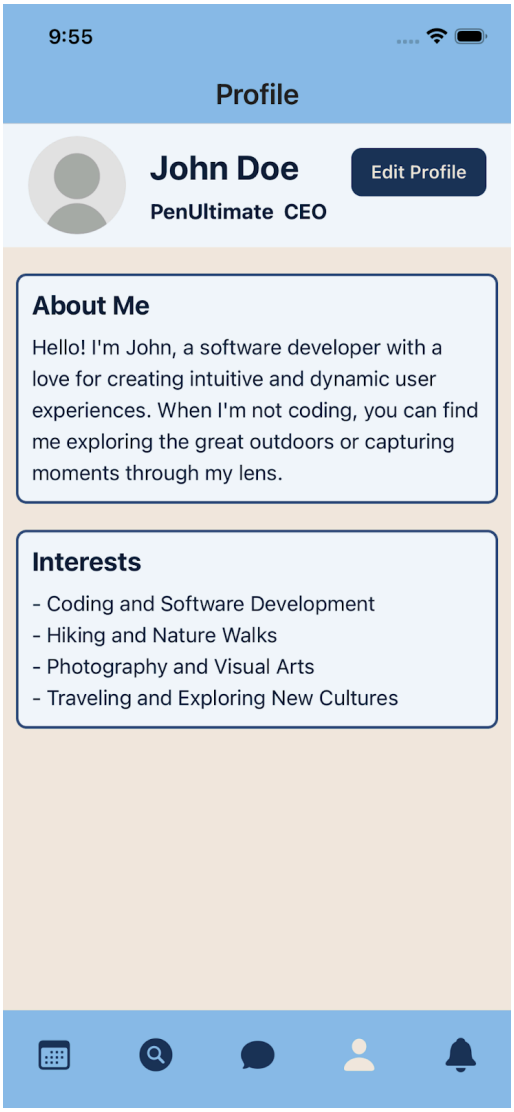

Figure 3.5.1 Profile Page

Selecting “Edit Profile” opens the Profile Settings screen, shown in Figure 3.5.2, where users can update and manage their personal information. The Profile Settings page includes fields for the user’s name, biography, interests, and an “about me” section.

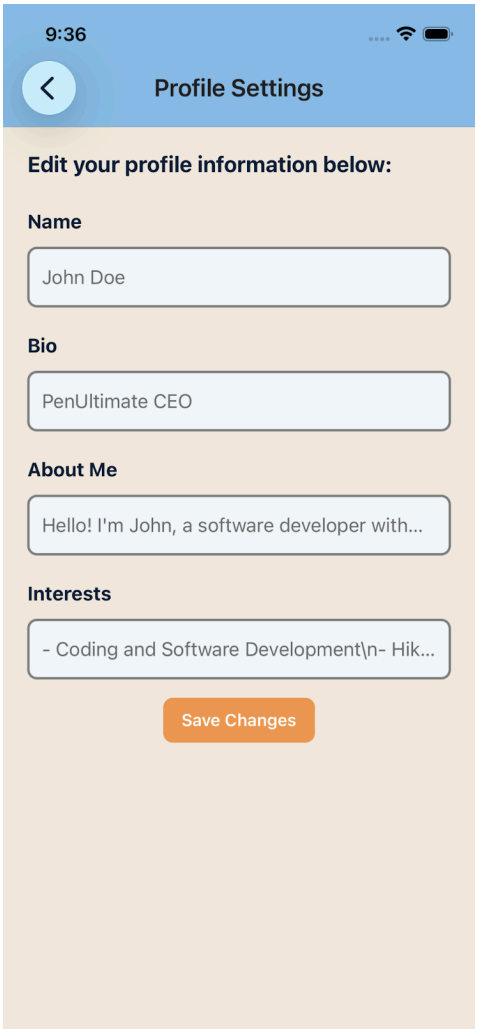

Figure 3.5.2 Profile Settings

3.6 Messages

The screens shown in Figure 3.6.1 provide a clear interface for users to view and manage direct conversations with other conference attendees. The main Messages screen displays a list of existing message threads, each showing the recipient's profile picture, full name, and a preview of the most recent message. Selecting any thread opens the corresponding Conversation screen.

At the top of the Conversation screen, a header shows the recipient's full name. The main area shows the conversation history in chronological order. Users can compose messages using the input field at the bottom of the screen and send them with the adjacent "Send" button.

From the main Messages screen, users can tap the plus icon to open the Search Users modal. They can enter a recipient's name or email into the search field. A list of matching users then appears below, with a profile picture and full name for each user. Choosing a given name launches a new Conversation screen for the recipient, allowing the user to begin a conversation.

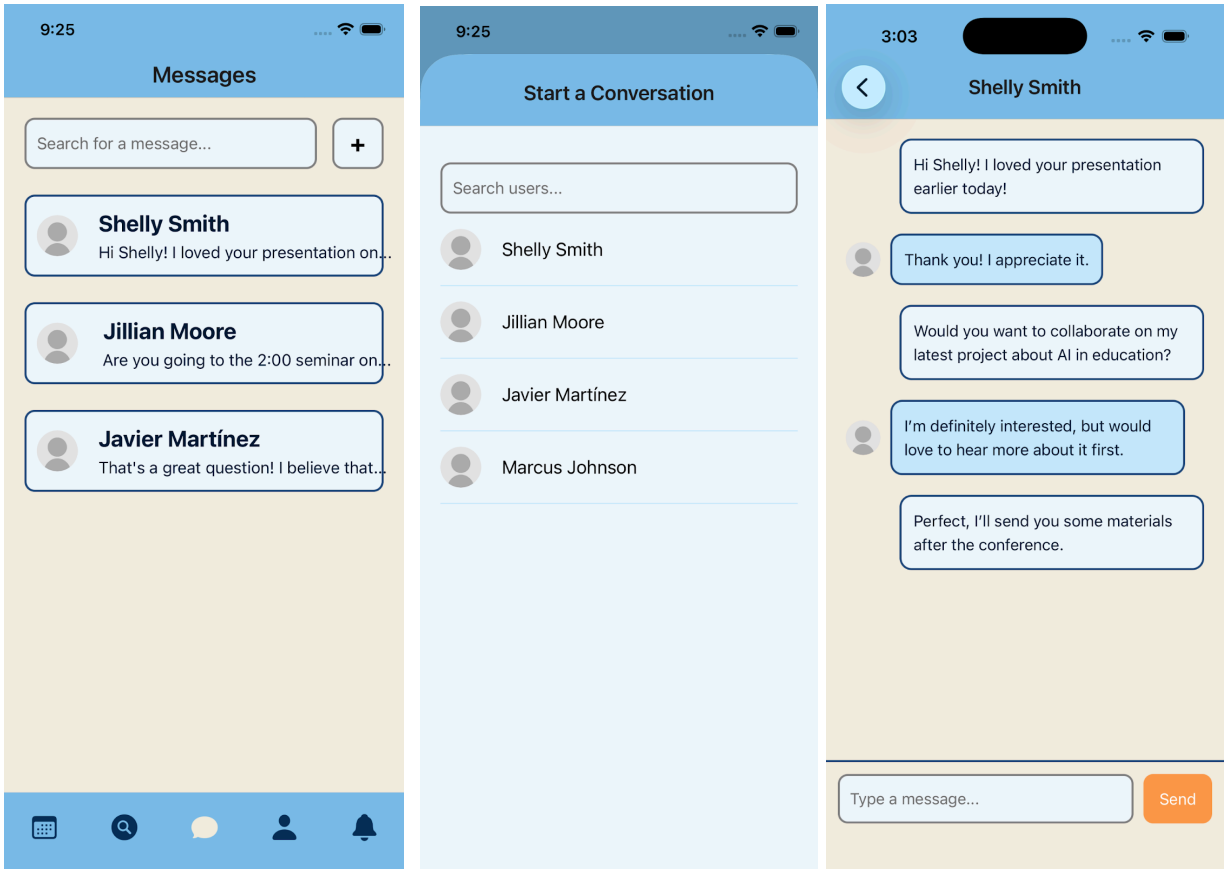

Figure 3.6.1 Messages, Conversation, and UserSearch Screens

3.7 Notifications & Manage Notifications

The Notifications screen shown in Figure 3.7.1 displays a chronological list of notifications that the conference administrator has sent to all participants. Notifications appear from most recent to oldest, and each entry shows the time elapsed since the notification was sent out. Unread notifications are emphasized with a bold text, and are marked as read once the user taps them. By default, each notification preview is limited to three lines, but users can tap any notification to expand it and view the full content.

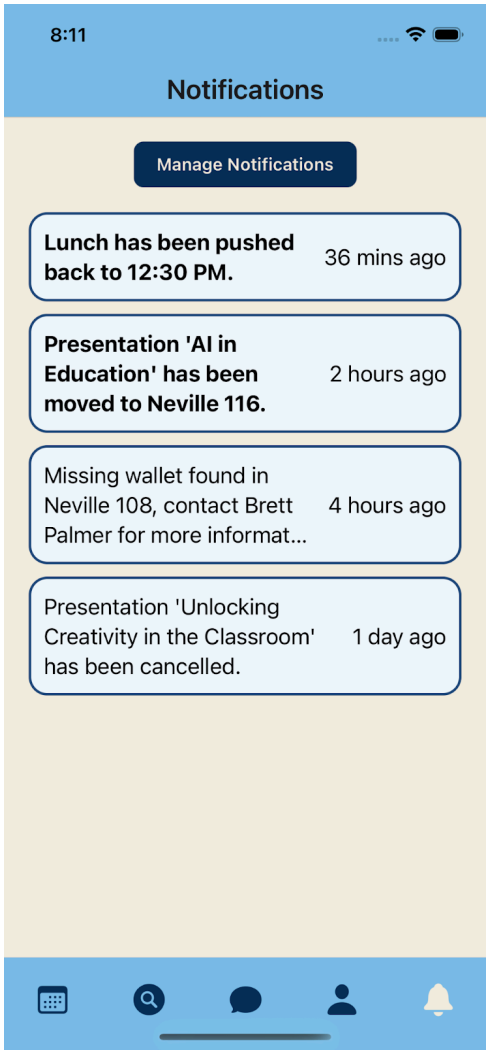

Figure 3.7.1 Notifications Screen

Selecting the “Manage Notifications” button opens the Manage Notifications screen, which allows users to adjust how they receive alerts about events, schedule changes, messages, and emails. As shown in Figure 3.7.2, the user is able to toggle whether they would like email notifications, push notifications, or SMS notifications, with checkboxes available to toggle each of these settings on or off.

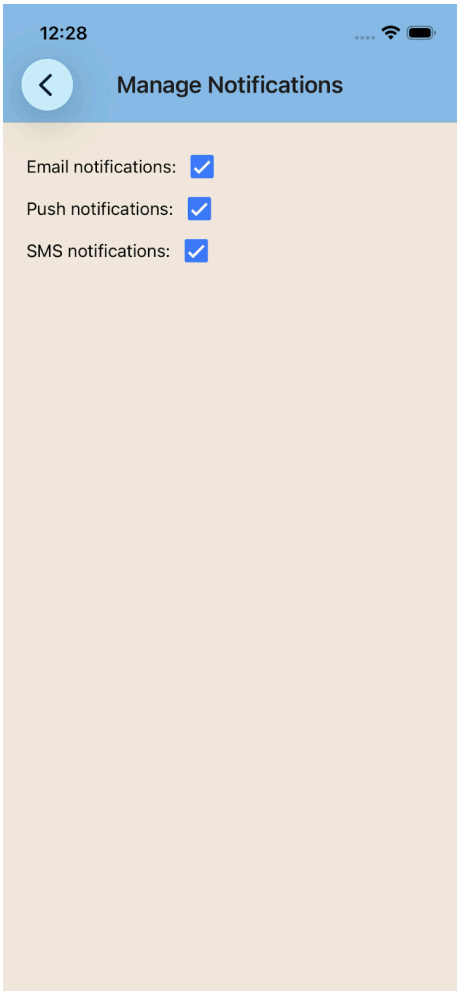

Figure 3.7.2 Manage Notifications Screen

4 Data Validation

This section describes how data is entered and validated in the IWAC Application to ensure consistency, accuracy, and security. Data validation rules are applied at both the database and application levels. This section distinguishes between data entered by administrators and data entered by end users.

4.1 Administrator Data Entry

For information entered by administrators, we will hold a training session with Dr. Falconer to demonstrate how to upload and manage data directly through Supabase, as well as to review supporting documentation for future reference. Dr. Falconer expressed a preference for a web-based spreadsheet interface rather than entering information directly through the mobile application. Supabase provides an intuitive interface for managing tables, which makes it a suitable choice for Dr. Falconer to fill in the Conference, Event, and Presentation tables as diagrammed in our SDD. All data constraints such as required fields, data types, and maximum lengths will be defined within the database schema in accordance to the rules established in the SDD. Figure 4.1 illustrates an example of the Supabase user interface using the Events table of the IWAC Application.

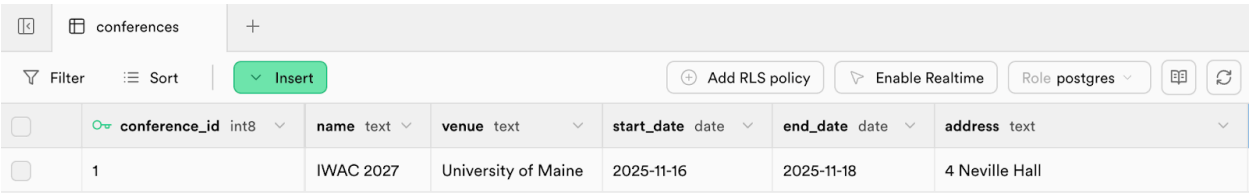

Figure 4.1 Supabase User Interface (Supabase, 2025)

4.2 User Data Entry

This section defines all data items that users can enter through the IWAC Application interface. Each item is mapped to its corresponding field in the database schema, specifying the data type, size, format requirements, and validation rules enforced by the system. Table 4.2 summarizes these inputs and constraints that ensure data integrity and consistency across the application.

Table 4.2 Data Validation

Data Item (Screen & Name)	**Database Item (Table & Field)**	**Data Type & Size**	**Format & Constraints**	**Validation Rules**
Registration email_input	Users email	VARCHAR(100)	Standard email format, e.g. “username@domain.tld”	Required; Unique; Exists in IWAC Registration;
Registration password_input	Users password	VARCHAR(255)	Length from 8 to 64 characters; At least one uppercase, lowercase, 1 digit	Required; Stored as salted hash of up to 255 characters
Profile first_name_input	Users first_name	VARCHAR(50)	Letters, hyphens, apostrophes, and spaces only	Required
Profile last_name_input	Users last_name	VARCHAR(50)	Letters, hyphens, apostrophes, and spaces only	Required
Profile affiliation_input	Profiles affiliation	VARCHAR(100)	Free text	Optional
Profile bio_input	Profiles bio	VARCHAR(500)	Free text	Optional
Comment comment_input	Comments content	VARCHAR(1000)	Free text	Optional
Messaging message_input	Messages content	VARCHAR(1000)	Free text	Optional

Appendix A – Agreement Between Customer and Contractor

The PenUltimate team is responsible for developing the application as outlined in this User Interface Design Document. The development includes application features, user interface requirements, user and administrator actions, data storage, and database management. The development can be done using open source software to base our work off of, and future developers will have access to our code and the database to be able to update it in the future as needed, as our involvement with building this application will end after May 2026.

Should either party wish to make changes to this document, both parties must meet ahead of time and mutually agree on said change. If Heather Falconer is the party requesting these changes, they should reach out to our client liaison, Monica Agneta, with their requests. Similarly, should we request any changes to this document, Monica will reach out to Heather with our requests.

Signature: _____ Date: 11/17/2025

Signature: _____ Date: 11/17/2025

Signature: _____ Date: 11/17/2025

Signature: _____ Date: 11/17/2025

Signature: _____ Date: 11/17/2025

Signature: _____ Date: 11/18/2025

Appendix B – Team Review Sign-off

All team members have signed to acknowledge they have reviewed this document and agreed on both its content and format. If team members have minor disagreements, they may state them in the comments area.

Rebecca Sonnemann

Signature: _____ Date: 12/03/2025

Comments: _____

_____

Brett Palmer

Signature: _____ Date: 12/03/2025

Comments: _____

_____

Monica Agneta

Signature: _____ Date: 12/03/2025

Comments: _____

_____

Ben Caras

Signature: _____ Date: 12/03/2025

Comments: _____

_____

George Pitt

Signature: _____ Date: 12/03/2025

Comments: _____

_____

Appendix C – Document Contributions

Brett

Brett wrote the section 3 introduction, section 3.6, and section 4, including Figure 4.1 and Table 4.2. He set up the React project for the UI and coded the Messages, Conversation, Search Users, and Notifications screens. Brett estimates that he did 20% of the work for this document.

Monica

Monica wrote the introduction paragraph, section 2, created figure 2, and Appendix A. She also coded the Itinerary screen and wrote 3.4 for it.

Monica estimates that she did 20% of the work for this document.

George

George wrote sections 3.5 and 3.6, and created the related pages for the prototype. George estimates that he did 20% of the work for this document.

Rebecca

Rebecca wrote section 3.3 and 3.4, as well as created the figures for those sections. Rebecca also wrote the purpose of the document. Rebecca estimates she did 20% of the work for this document.

Ben

Ben wrote section 3.1 and 3.2 as well as created the figures for these sections. Ben wrote the introduction paragraph for section 3. Ben estimates that he did 20% of the work for this document.