SOFTWARE REQUIREMENTS SPECIFICATION

PulseCheck

Trinity Dhillon, Daniel Sirias, Michael Campos, Brandon Budhan, Peter Georgaklis, & Zaira Garcia

2025-02-07
v1.0.0

Table of Contents

Table of Contents

This Page Intentionally Left Blank

1. Introduction

This Software Requirements Specification (SRS) document provides a comprehensive overview of the requirements for the system called ***PulseCheck***, a real-time in-class polling web application. This system is designed for large lecture halls to facilitate quiz and exam-based assessments, enhancing student engagement and automating performance tracking. By providing instant feedback, it allows instructors to gauge student understanding and adjust their teaching accordingly.

This document adheres to the IEEE 830-1998 standard and serves as a foundational reference for instructors, developers, and administrators involved in its development, implementation, and use. It outlines the system's features, key stakeholders, technical requirements, ensuring a clear understanding of the application's capabilities and expectations.

1.1 Purpose

This document defines the functional and non-functional requirements for the Real-Time In-Class Polling Web Application, a system designed to facilitate real-time polling and multiple choice assessments in large lecture settings. The system provides a web-based interface for both instructors and students. Instructors can create and conduct quizzes and polls, while students participate and submit responses in real time. Responses are recorded, analyzed, and reported in real time, providing immediate feedback to both students and instructors.

The system is designed to achieve the following objectives:

- Enable real-time participation: Students can access quizzes and polls instantly through a responsive web interface.
- Automate result calculations and reporting. The system processes responses in real time, providing students with immediate feedback on their performance.

- Provide detail-oriented analytics. Instructors receive comprehensive reports and insights on student responses to assess comprehension and refine instructional strategies.

1.2 Scope

The Real-Time In-Class Polling Web Application (PulseCheck) is a web-based system designed to facilitate real-time multiple-choice quizzes and polls during large lecture sessions. The system aims to enhance student engagement, automate attendance tracking, and provide instructors with actionable insights into student comprehension.

1.2.1. Intended User and Roles

This web application is designed for use in lectures and supports two primary user roles:

Participant Capabilities

- Log in through a web interface to participate in real-time polls and quizzes
- Receive immediate feedback on their performance.
- View individual scores and incorrect answers after submission.

Host Capabilities

- Create, manage, and schedule multiple-choice questions for quizzes and polls, both manually and with the help of artificial intelligence.
- Monitor real-time student participation and attendance.
- Generate reports for performance, analysis, and instructional adjustments.

1.2.2 System Boundaries and Constraints

- The application requires an internet connection and runs on modern web browsers (e.g., Google Chrome, Mozilla Firefox, Microsoft Edge, Safari).
- User authentication is mandatory for both instructors and students.
- The system supports a maximum of 20+ concurrent users per session to ensure real-time performance.
- files uploaded for AI question generation, have to be in pdf format and have a maximum size of 20 MB

1.2.3 System Interactions

- Users authenticate using Firebase Authentication to access polls, quizzes, and their dashboard.
- The system displays poll or quiz results in real time using Firestore's real-time syncing feature. Updates reflect immediately across all connected devices.
- Multimedia elements (e.g. images) are uploaded to Firebase Cloud Storage and display dynamically during quiz and poll interactions.
- Student responses remain securely in a database and made available for further analysis and insights.

1.2.4 Out of Scope

- The system does not support subjective or open-ended questions (e.g., essays, short answers).
- It does not replace a full-fledged exam platform, but is intended for in-class engagement and formative assessments.
    - Does not support a wide variety of questions types (e.g. short answer, essay, coding exercises).
    - Does not support advanced proctoring features (e.g. webcam monitoring, screen recording, plagiarism detection).
    - Does not support strict exam security measures (e.g. lockdown browser, randomized questions).
    - Does not support automated grading and integration with external gradebooks.
- Eternal API integrations (e.g., third-party grading software) are not included in this version.

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 General Terms

- Multiple Choice Question (MCQ) - A question format where students elect the correct answer from multiple predefined options.
- Real-Time Polling - A system where students submit responses instantly and receive immediate feedback.
- Software Requirements Specification (SRS) - A document that defines the functional and non-functional requirements of a software system.

1.3.2 Technical Components

- Database Management System (DBMS) - Software that allows users to create, store, and manage structured data efficiently.
- Firebase - A cloud-based Backend-as-a-Service (BaaS) platform that provides authentication, real-time database, cloud storage, and hosting.
- User Interface (UI) - The graphical layout and interactive element of a software application that users interact with.
- Vertex AI - A cloud based machine learning/AI deployment and management platform. Allows for custom models to be stored and deployed via Google Cloud, and gives access to Google created models through APIs.

1.3.3 Development Tools

- ESLint - A static code analysis tool that identifies and enforces coding standards in JavaScript.

1.4 References

[1] "IEEE Recommended Practice for Software Requirements Specifications," in IEEE Std 830-1998 , vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286. keywords: {Software requirements and specifications;contract;customer;prototyping;software requirements specification;supplier;system requirements specifications},

[2] "ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes --Requirements engineering," in ISO/IEC/IEEE 29148:2011(E) , vol., no., pp.1-94, 1 Dec. 2011, doi: 10.1109/IEEESTD.2011.6146379. keywords: {IEEE standards;Software engineering;Product life cycle management;Software development;System analysis and design;buyer;characteristics;concept of operation;concepts of operations document;ConOps;contract;customer;operational concept;OpsCon;prototyping;requirement;software requirements specification;supplier;SyRS;system;system requirements specification},

[3] S. Jeffreys, Software Engineering Term Project - Spring 2025, Spring 2025. Hofstra University. [Online]. Available: CSC 190/289 Canvas. [Accessed Feb. 8, 2025].

1.5 Overview

The remainder of this document is structured as follows:

- Section 2: Overall Description - Provides a high-level overview of the system, including its purpose, intended users, constraints, and dependencies.
- Section 3: Specific Requirements - Defines the system's functional and non-functional requirements, covering external interfaces, system attributes, performance constraints, and security considerations.

This document ensures a comprehensive specification of the system's requirements, detailing its capabilities, constraints, and technical considerations.

2. Overall Description

2.1 Product Perspective

PulseCheck is a web application that allows hosts to display interactive quizzes and polls to complement lectures and presentations by encouraging active engagement from participants. The application runs on web browsers and uses Firebase to manage user data.

2.2 Product Functions

The system runs as a standalone application, complements lectures, notes, slide decks, etc. There are several key features that this application offers to enhance user engagement and interaction.

Real-Time Questionnaires

- During sessions, the host can create and display multiple-choice questions. Participants join sessions by entering a 6 character code composed of letters and numbers, in addition to their name. This allows them to respond instantly and simultaneously using any device with web access.

Automated Scoring

- The system automatically calculates quiz scores for individual participants using correct responses. Participants receive immediate feedback on their individual performance after each question, and also receive an overview of questions they've answered incorrectly at the end of each session.

Instructor Dashboard

- Instructors are able to monitor participants in real time. They can see participants' names, scores, and incorrect answers for analysis and feedback. These performance insights help instructors gauge participant understanding of the material.

Web-Based Access & Responsive UI

- PulseCheck being a web application works on desktop, tablets, and mobile devices that use modern web browsers. It is built using React, Material-UI, and Firebase to ensure an optimized user experience.

Attendance Tracking

- User participation in questionnaires can serve as an attendance check. Non-participating users can be flagged for review, and the host has the option to remove such participants.

Question Bank Management

- Instructors can create question banks and reuse them for future sessions. Questions can be randomized in order to prevent answer sharing between participants.

2.3 User Characteristics

The two main user classes are Logged-in Users and Guests. Logged-in Users who choose to host can create and manage new polls, as well as host live poll sessions. Logged-in Users and Guests are able to join sessions as participants through a unique six character code, and answer questions in real time.

Accessibility

- The software is intuitive and accessible, meaning that users do not need a particular level of technical proficiency.

Demographics

- The intended user base for this application includes students, teachers, and professors, but remains accessible to a broader audience. Polls can be centered around any topic, and are not limited to regard academic subjects.

Device Requirements

- This application is usable on any device that has internet access and can run modern web browsers.

2.4 Constraints

The following constraints apply to this application:

Slides

- PulseCheck is a complementary system for presentations allowing for polling at certain points throughout, rather than being an entirely encompassing system, including both polls and slides.

Participant Limit

- While the host is able to set the amount of participants taking part within a poll themselves there is a hard maximum participation limit of 300 people taking part in any given poll.

Firebase

- The system using Firebase brings a couple of constraints, namely;
    - Firebase Spark Plan: not having as many features when compared to the paid version such as storage, and a set amount of data that can be transferred per day. With the free version being able to transfer 360MB/day VS the paid version where data transfer costs $0.15/GB.
    - In the event that Firebase's servers are down, PulseCheck becomes inoperable for a period of time

Vertex AI

- Vertex AI, in relation to Google cloud, requires fees to be paid per prompt passed through the AI models. These fees are determined by the number of tokens in each prompt. However, this system is utilizing $300 trial credit given by Google Cloud until it is utilized in its totality. Once credit has been fully redeemed, charges will surge with each AI poll generation request made by users.

2.5 Assumptions and Dependencies

It is necessary for users to have a stable internet connection, on a supported device and web browser in order to connect to PulseCheck's website to interact with polls.

The major dependency for PulseCheck's operation is Firebase, as it is the database the entire system runs on, and where its data is stored. As mentioned previously in Section 2.4 PulseCheck is subject to any downtime that Firebase is experiencing, with the length of the downtime being out of PulseCheck's control. Another dependency is the availability of Vertex AI,

Google Cloud and the gemini-2.0-flash-001 multimodal model. If either of these platforms/components become unavailable, host users are not able to generate polls using the AI feature. In the case the gemini-2.0-flash-001 gets taken down or becomes outdated, the poll generation AI feature also becomes unavailable. When new versions of the multimodal become available, developers will update the system to use the most up to date model.

3. Specific Requirements

3.1 External Interfaces

3.1.1 User Interfaces (UI Design Considerations, Accessibility)

This user interface of the system focuses on simplicity, engagement, and accessibility, catering to both students and instructors in large lecture hall environments. The design process prioritizes creating an intuitive experience, ensuring that users can effortlessly navigate through the system's features, participate in quizzes and polls, and view results in real-time.

To achieve this, a comprehensive set of Figma prototypes illustrates every aspect of the user's journey. These designs reflect a user's first approach to the system, with clean layouts, responsive elements, and role-specific layouts tailored to the unique needs of students and instructors alike. ()

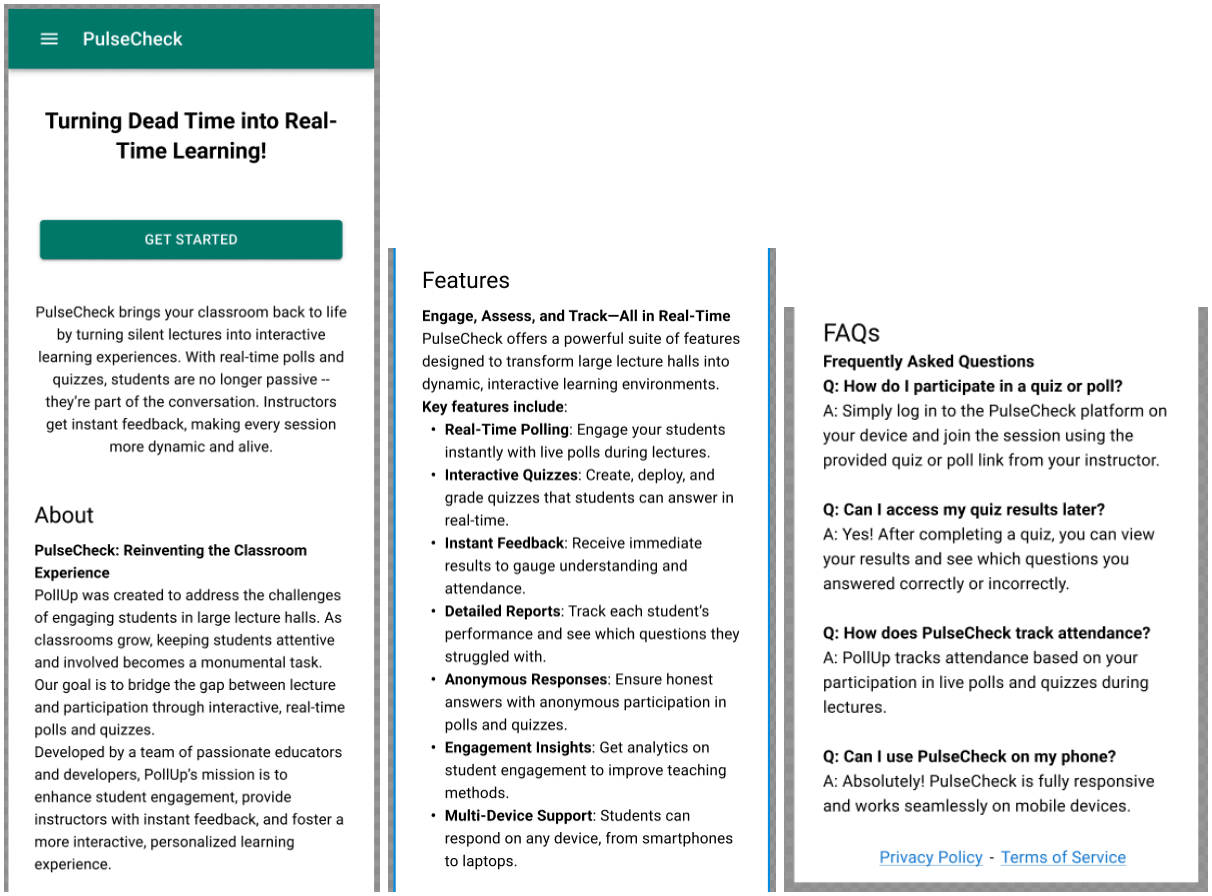


Figure 1P - Splash Screen

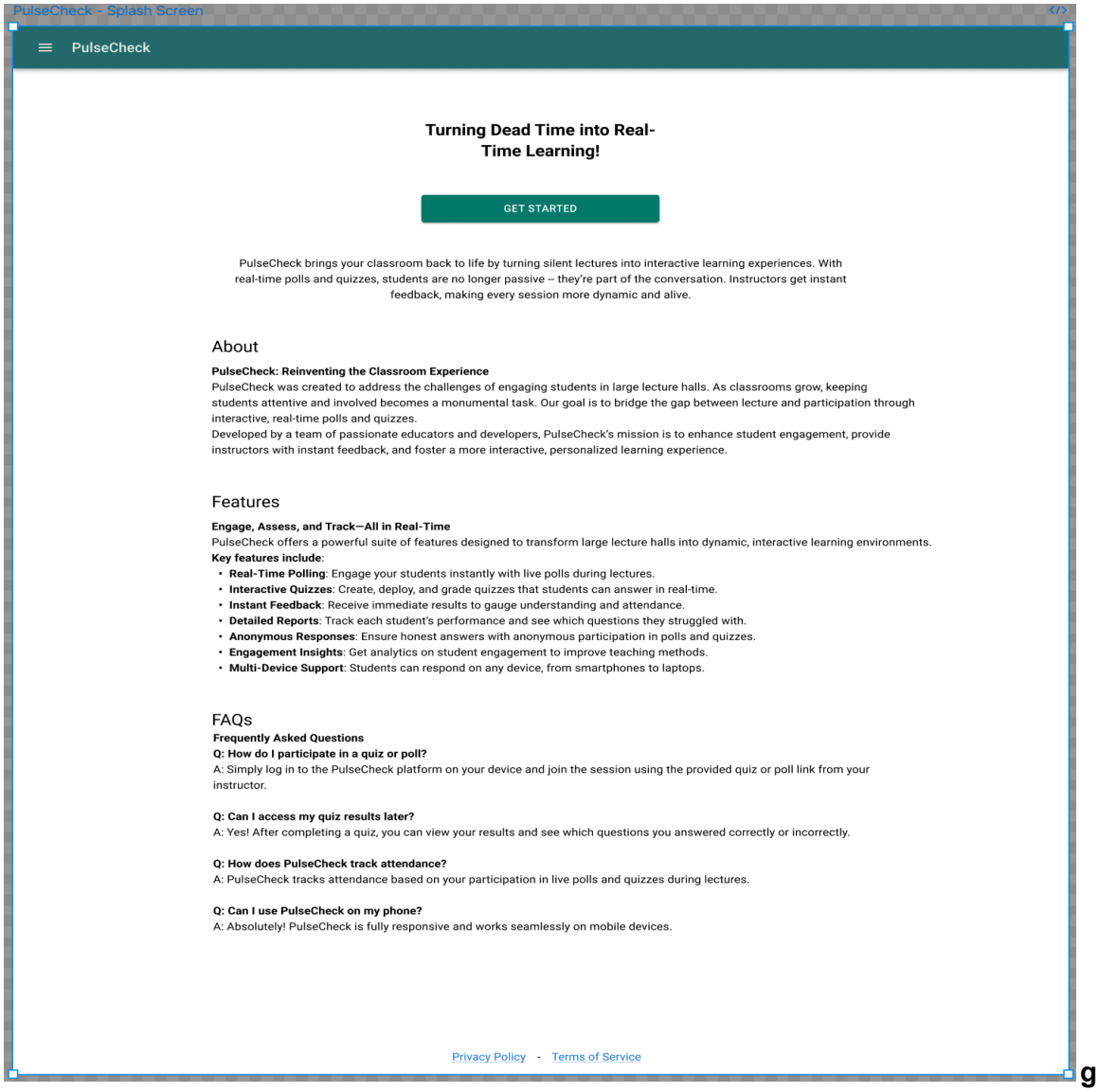


Figure 1D - Splash Screen

- The first screen the user is shown when visiting the site.
- This informs the user key features the system provides and possible questions that they may have throughout the system with answers to these questions.
- Clicking the ‘GET STARTED’ button, navigates the user to Figure 3P/D.

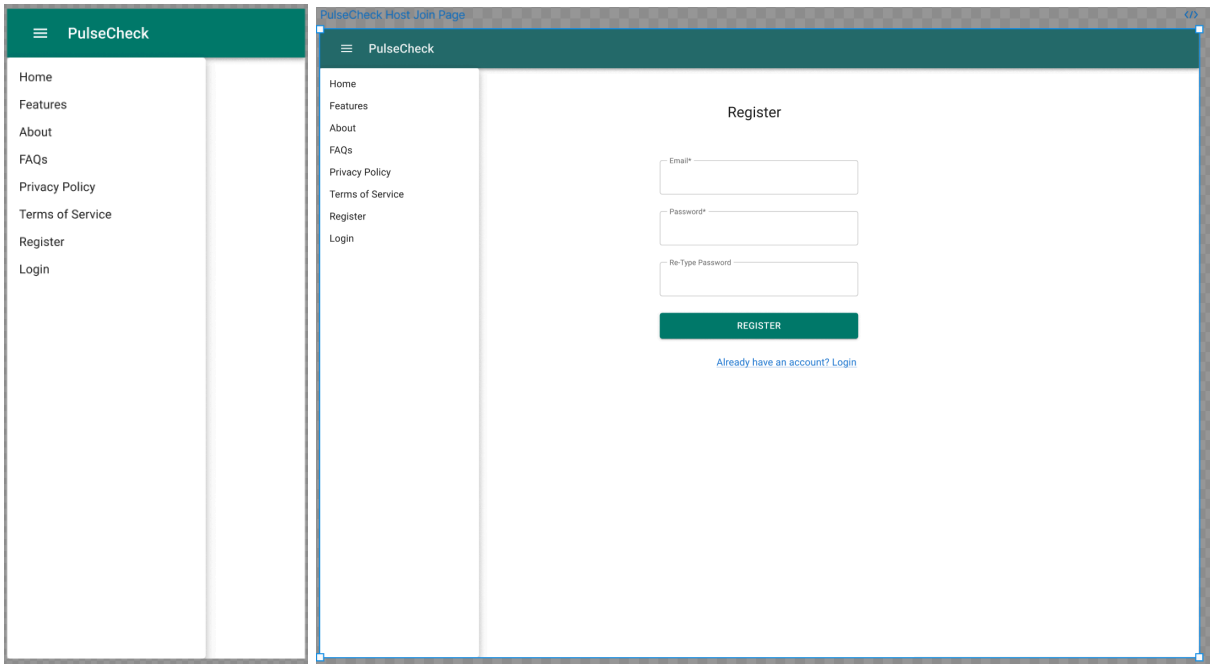


Figure 2P/D - Navigation Menu

- Only shown for new users (logged-out users) to navigate throughout the website.
- Home navigates to Figure 1P/D.
- Features navigates to Figure 1P/D to the features section.
- About navigates to Figure 1P/D to the about section.
- FAQs navigates to Figure 1P/D to the FAQs section.
- Privacy Policy navigates to Figure 6P/D.
- Terms of Service navigates to Figure 7P/D.
- Register navigates to Figure 5P/D.
- Login navigates to Figure 4P/D.

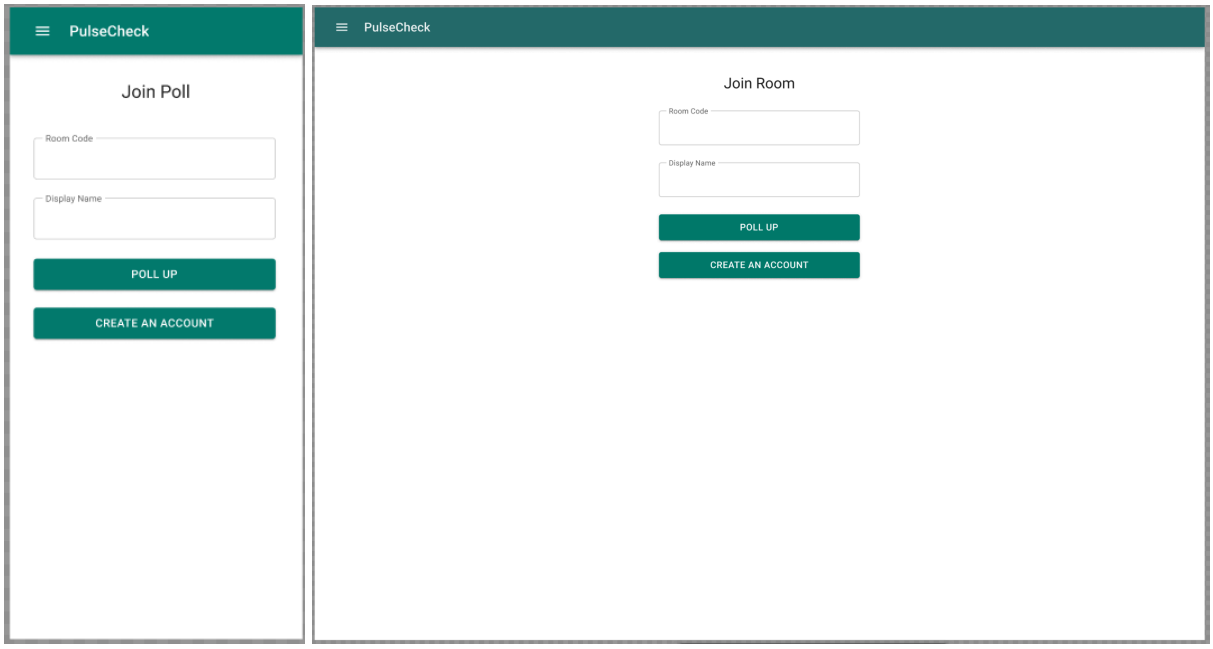


Figure 3P/D - Join Poll as Guest Screen

- Displays controls to join a poll. Enter a room code given by the host of the poll and enter the name you want to display. Joining a poll without an account is joining a poll as a Guest.
- Clicking the “POLL UP” button tries to join the poll.
- Clicking the “CREATE AN ACCOUNT” button redirects the user to Figure 5P/D.

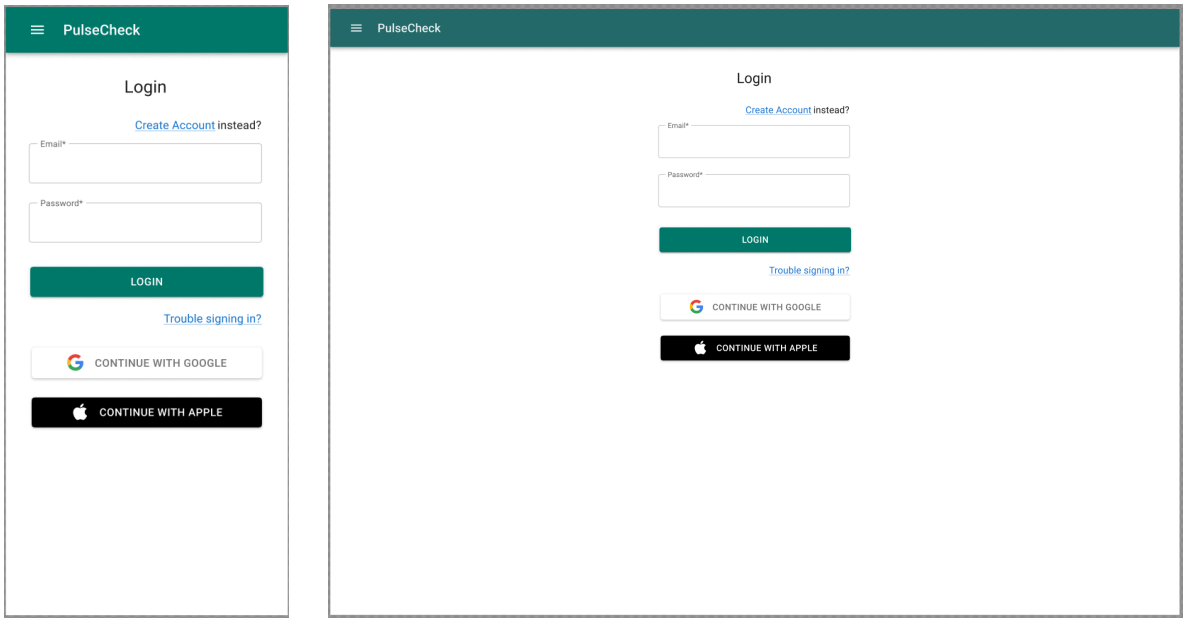


Figure 4P/D - Login Screen

- Allows the user to enter their credentials to login.
- The user has the choice to login with their Google account
- The user has the choice to login with their Apple account.***(REMOVED)***

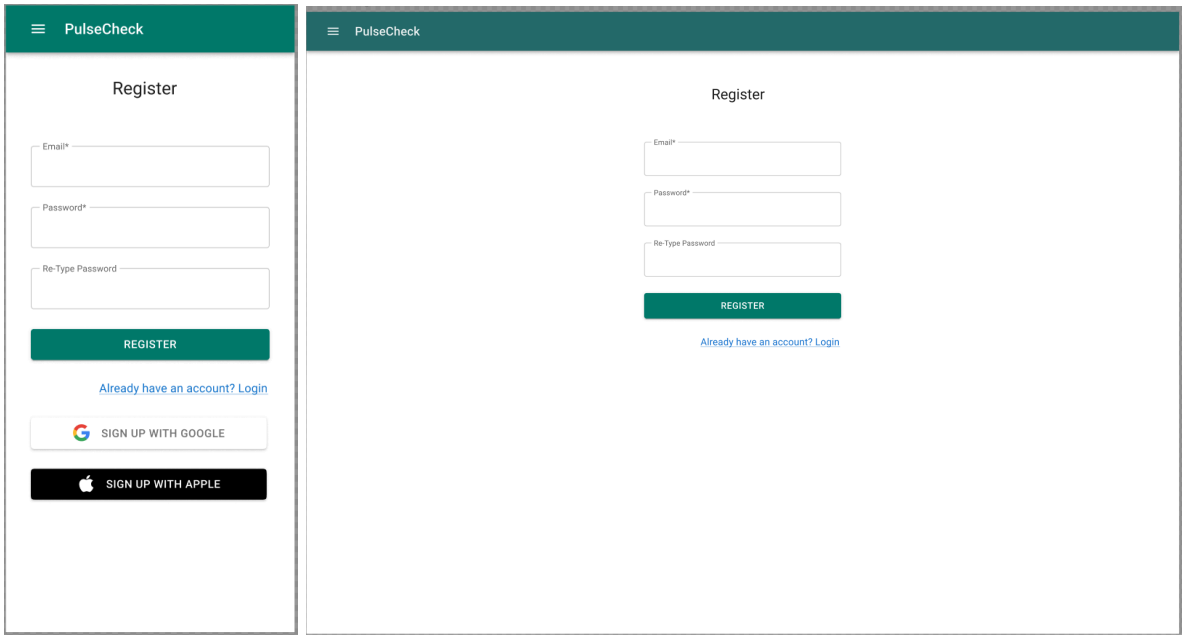


Figure 5P/D - Register Screen

- Allows the user to register their account.

- The user has the choice to sign up with their Google account.
- The user has the choice to sign up with their Apple account.***(REMOVED)***

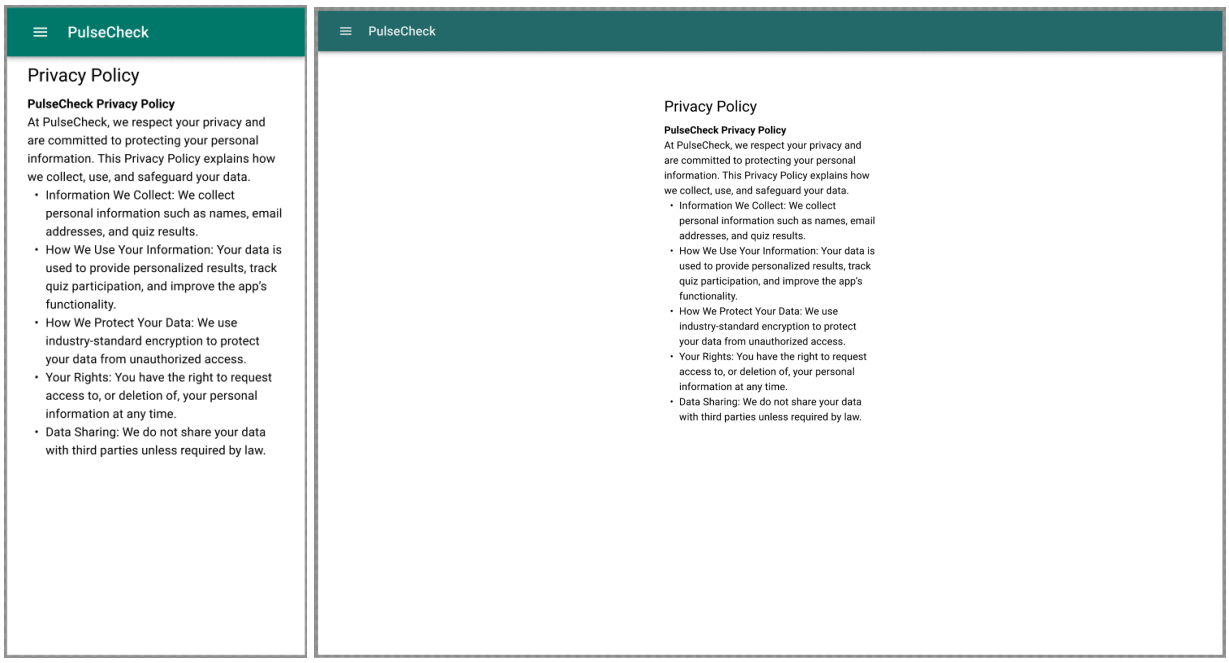


Figure 6P/D - Privacy PolicyScreen

- Displays the system's privacy policy.

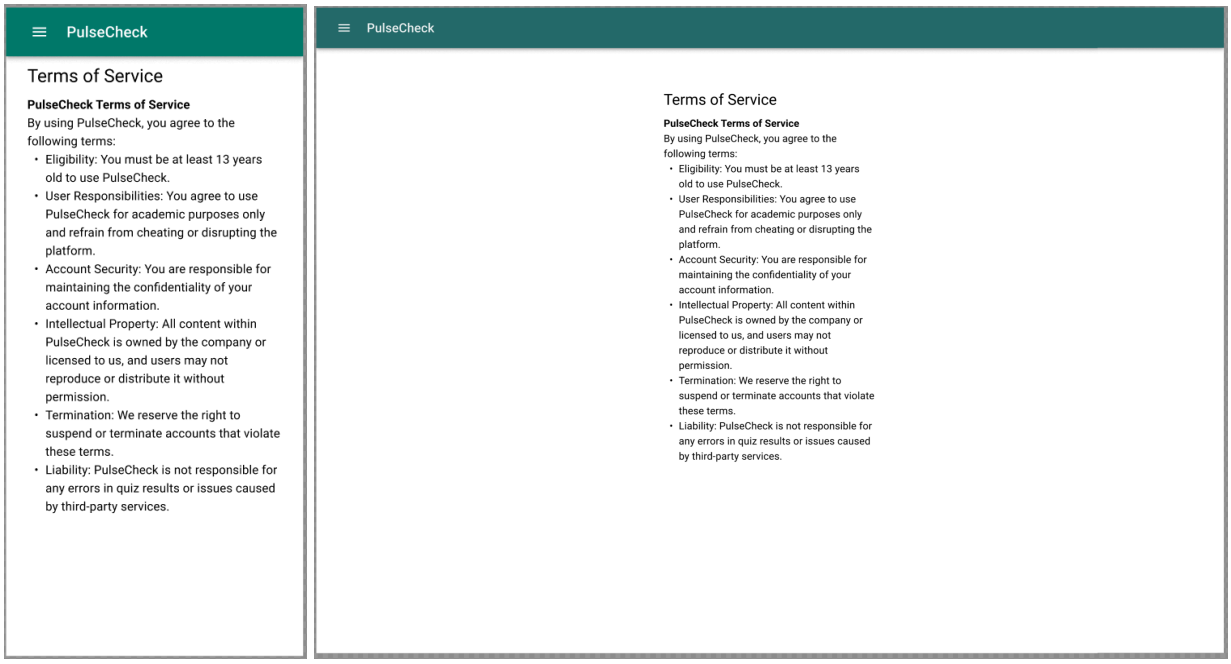


Figure 7P/D - Terms of Service Screen

- Displays the system's terms of service.

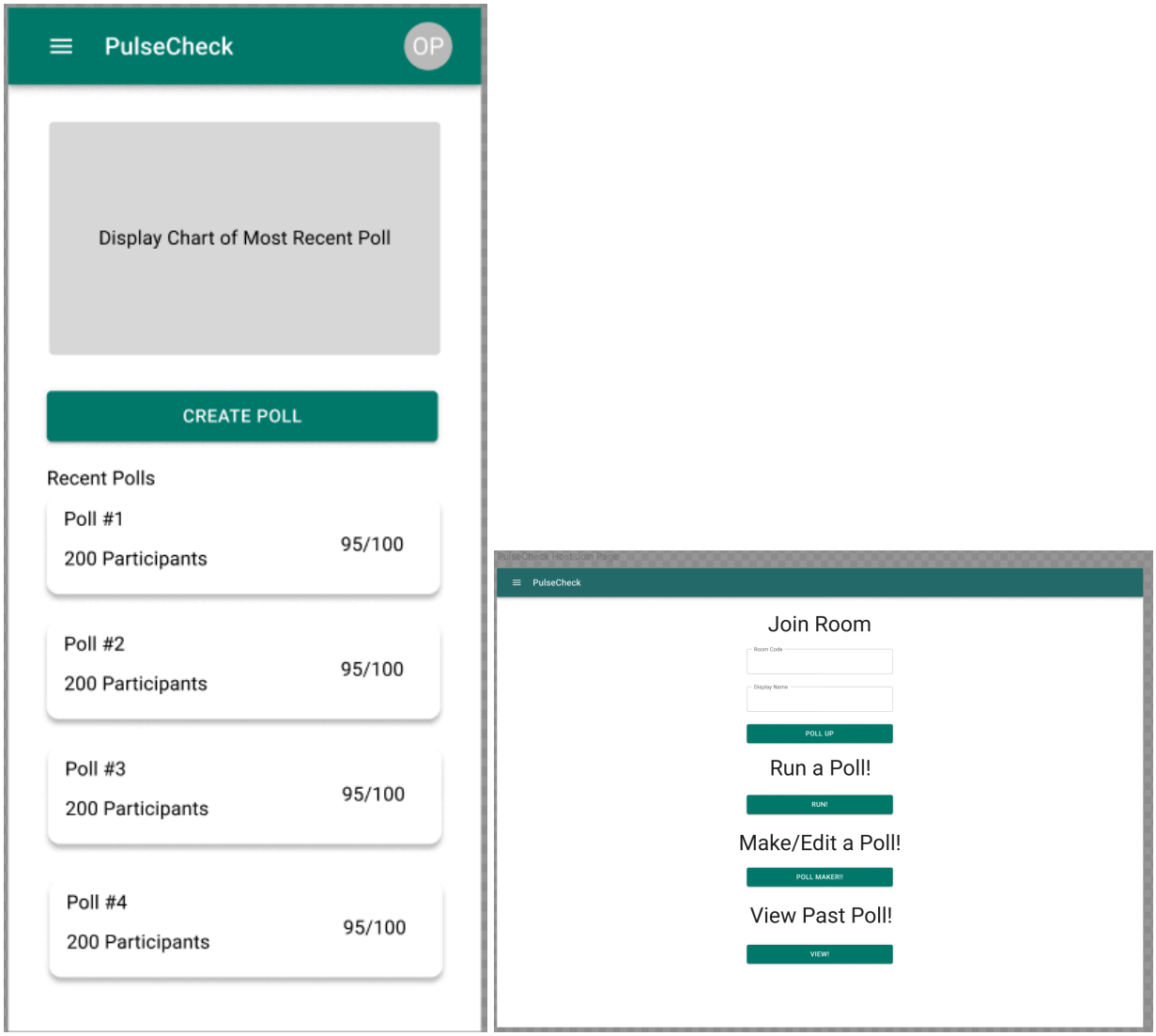


Figure 8P/D - User Dashboard

- *Displays the most recent poll the user participated in.*
- *Displays a button for the user to create a poll, navigates to Figure 10P*
- *Displays a list of recent polls the user participated in.*
- *Only displays for authenticated users.*

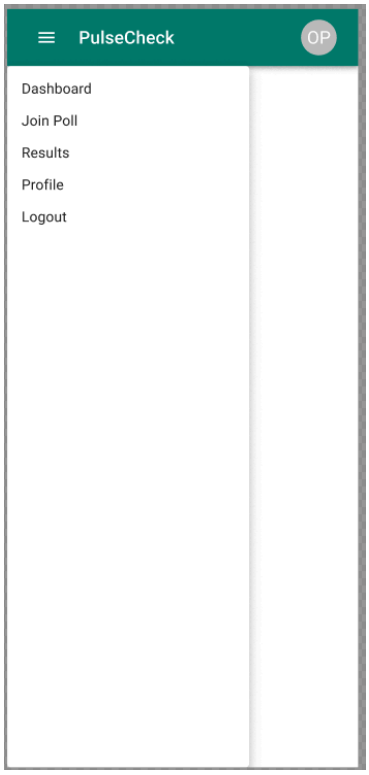


Figure 9P - Navigation Menu (Authenticated Users)

- Only displays for logged-in users.
- Dashboard navigates to Figure 8P/D.
- Join Poll navigates to Figure 10P/D.
- Results navigates to Figure 13P/D.
- Profile navigates to Figure 14P/D.
- Logout navigates to Figure 1P/D.

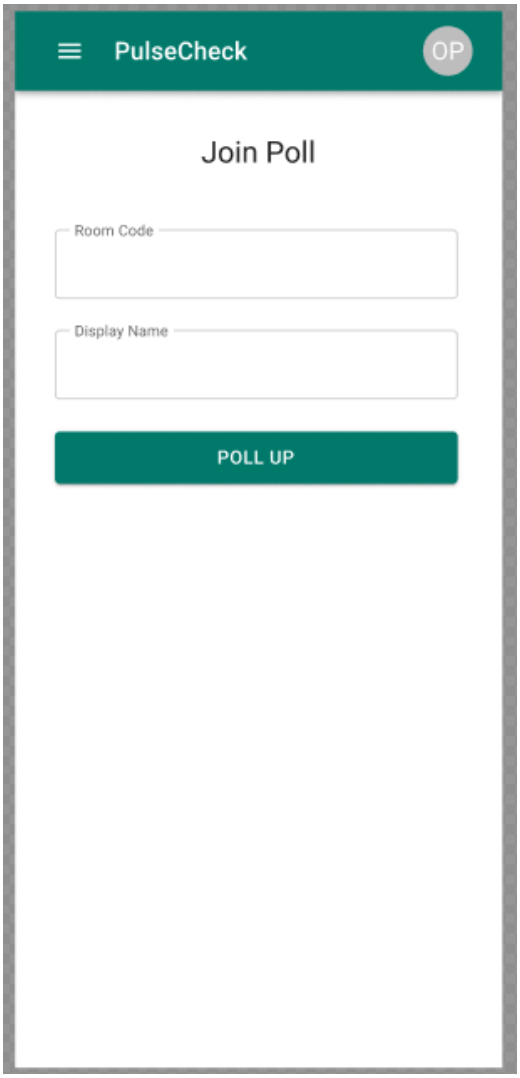


FIgure 10P - Join Poll (Authenticated User)

- Allows the user to enter the room code and their display name to join a poll.

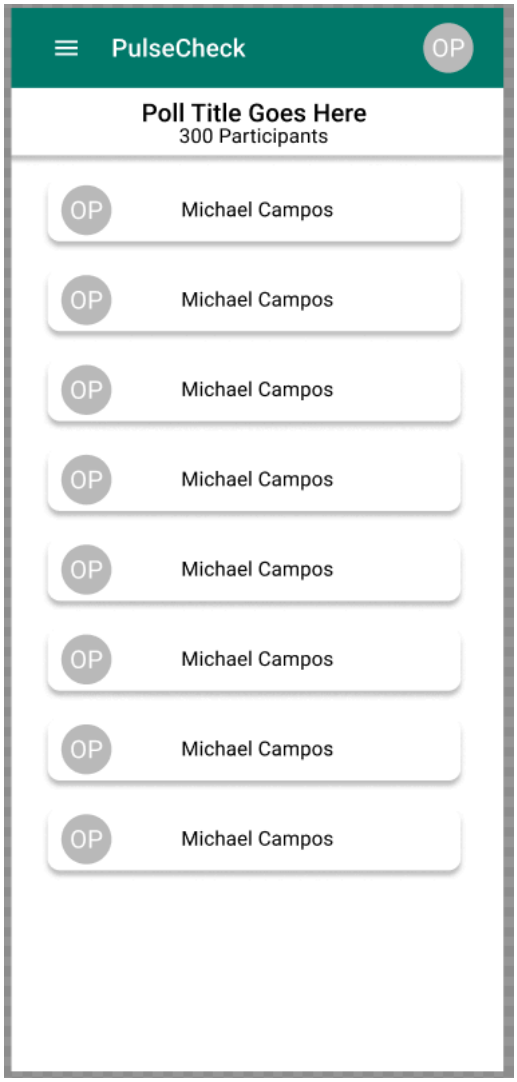


Figure 11P - Joining Poll (Participant Perspective)

- Displays after the user successfully joins a poll.

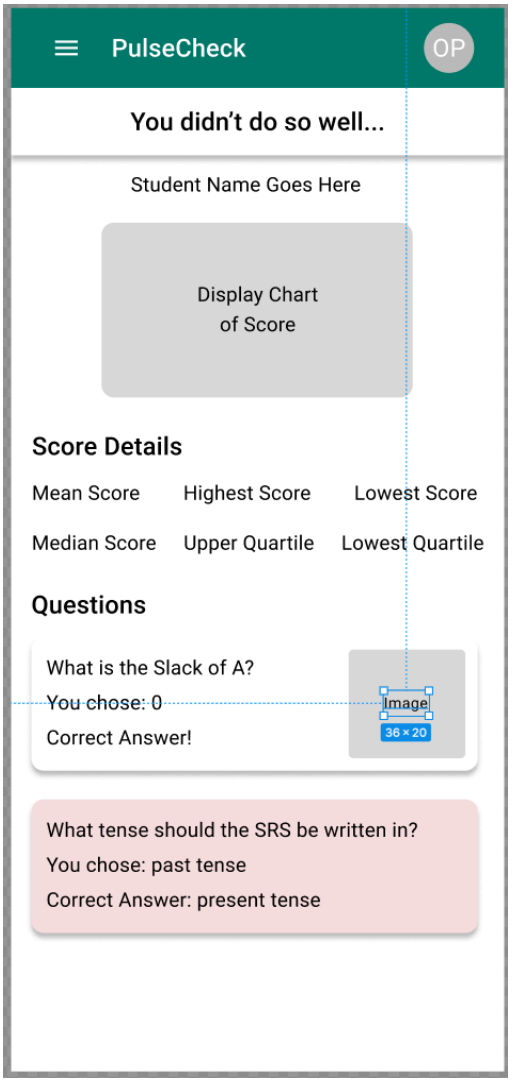


Figure 12P - Quiz (Poll) Results

- Shows the results after the user completes the quiz (poll)

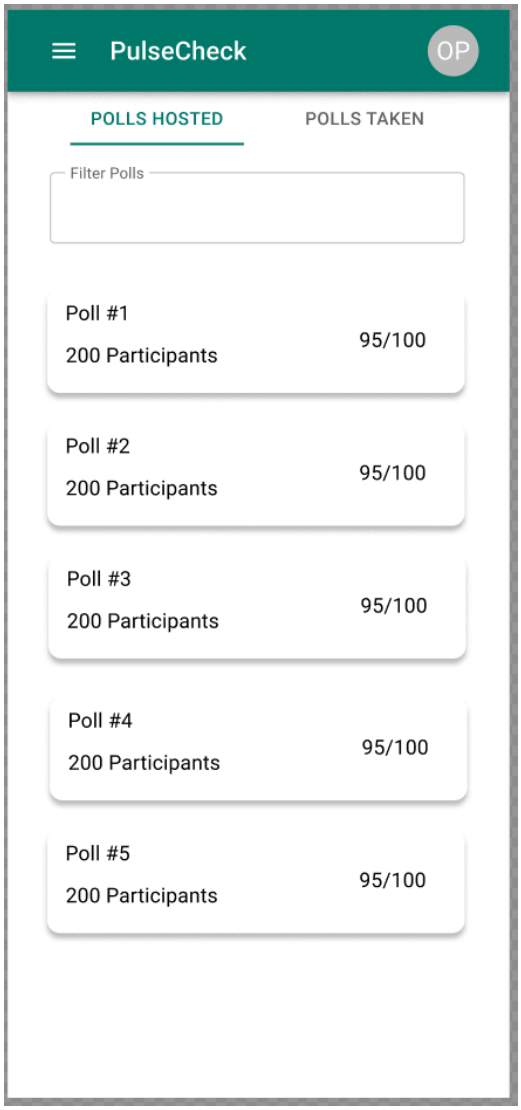


Figure 13P - Results

- Displays the results of polls hosted by the user.
- Displays the results of polls taken by the user.

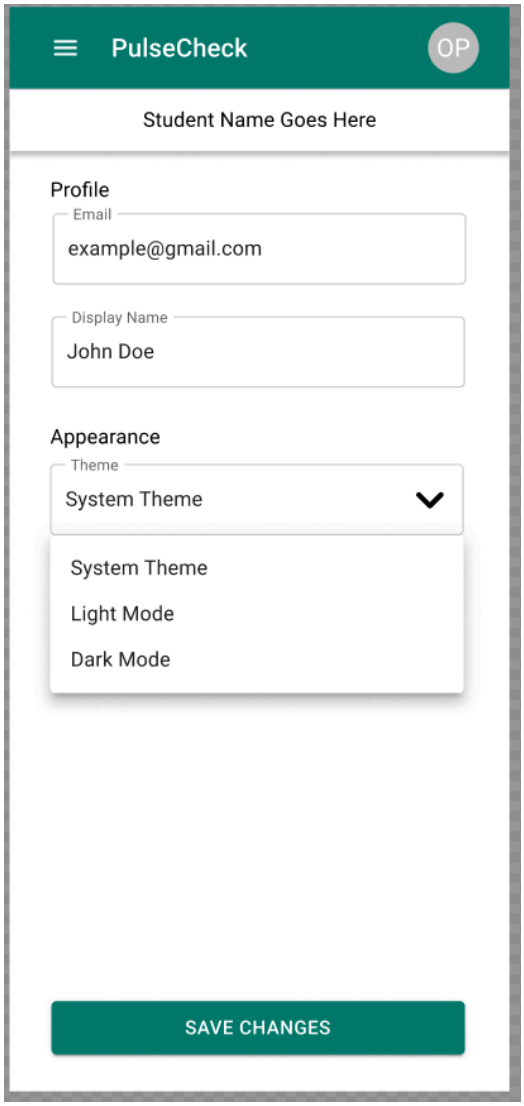


Figure 14P - User Profile

- Displays profile information about the user.
- Display theme chooser.
- Only displays for authenticated users.

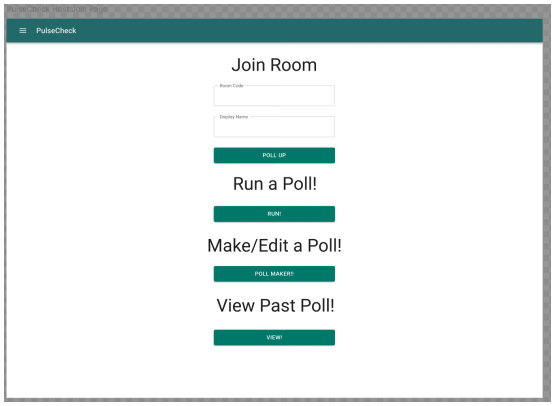


Figure 9D - Home View (Authenticated Users)

- Only displays for logged-in users.
- Allows the user to enter the room code and their display name to join a poll.
- Run! navigates to Figure 8D.
- Join Poll navigates to Figure 10D.
- POLL MAKER! navigates to Figure 13D.
- Profile navigates to Figure 14D.
- Logout navigates to Figure 1D.

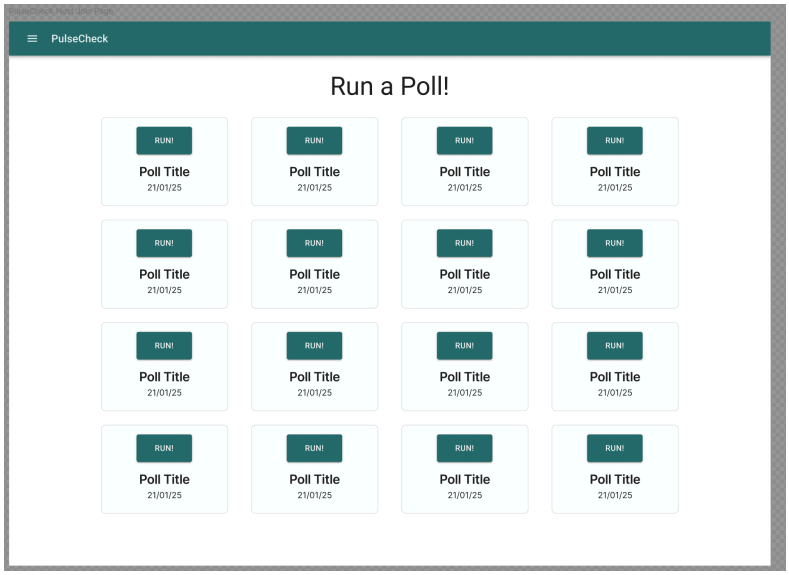
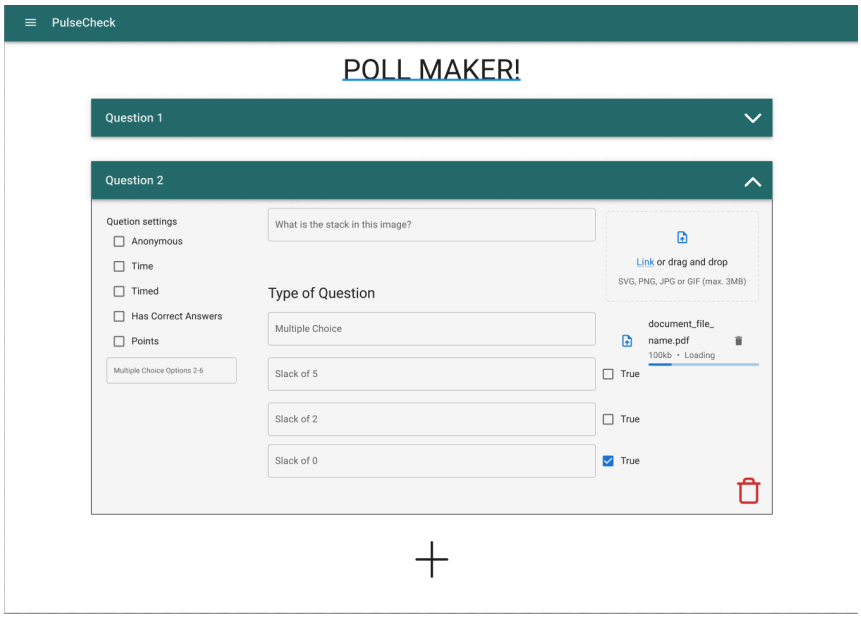


Figure 18D - Run A Poll

- Users who have created polls can host it here.

Figure 19D - Poll Maker

- Folder With a Plus allows users to make new poll navigates to Figure 20D.
- Poll Card navigates the user to a poll editor navigates to Figure 20D.



NOTE: Any resemblance to actual established products (POLL MAKER!), living or dead, or actual events is purely coincidental.*

Figure 20D - Poll Editor

- Plus symbol Adds another question.
- Allows the user to enable settings to the question.
- Allows the user to upload images
- Allows the user to choose what type of question is asked, shown in Figure 21D.
- Allows the user to choose how many questions are asked, shown in Figure 21D.
- Allows users to pick which questions are true, when the options are available.
- The Garbage Can symbol deletes the question.

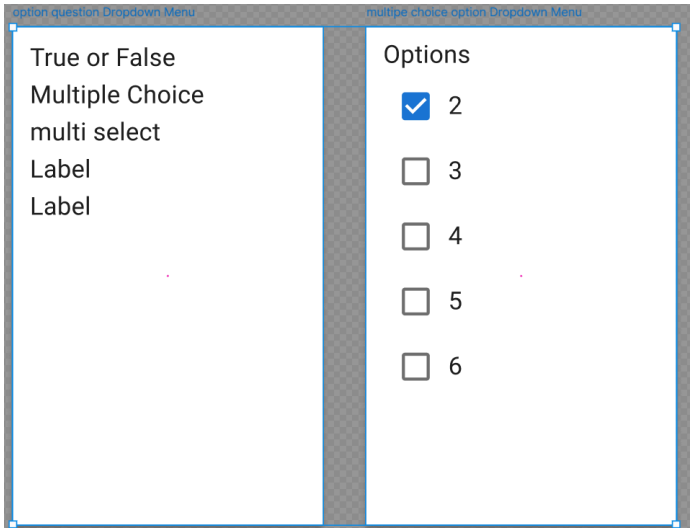


Figure 21D - Poll Maker

- Allows the user to choose what type of question is asked.
- Allows the user to choose how many questions are asked.

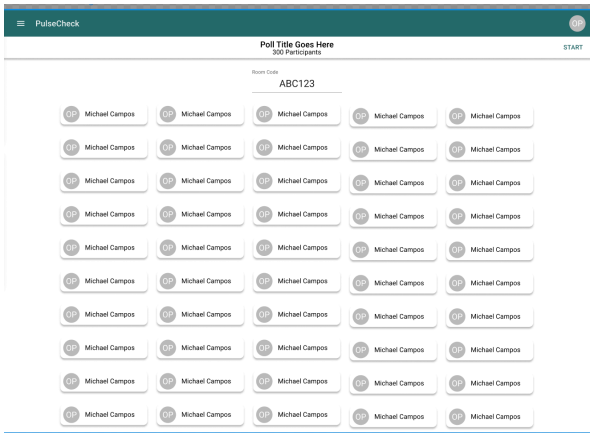


Figure 22D - Host Join View

- Allows the host to view all participants and how many are joined.
- Allows the host to kick participants, shown in *Figure 23D*.

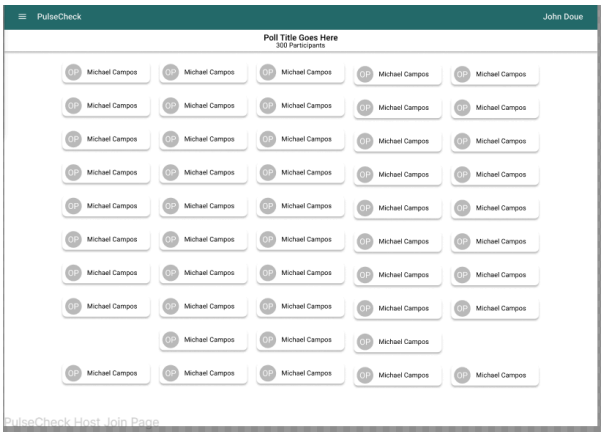


Figure 24D - Participants join view

- Allows the participants to view all other participants and how many joined.

- Allows hosts to share room code to allow participants to join.
- Allows the host to start the poll once everyone has joined

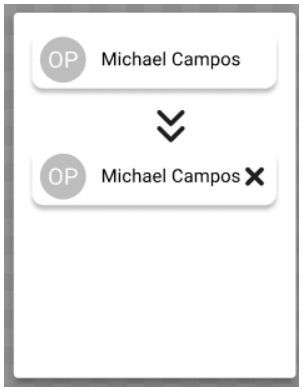


Figure 23D - Host Kick participants *card*

- Allows host to kick participants

Figure 23D - User In Poll dropdown

- Allows participants to share room code to allow other participants to join.
- Allows the Users to go to their settings.
- Allows the Users to leave the poll

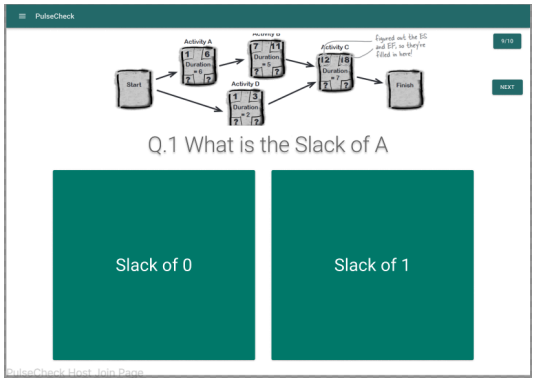


Figure 25D - Host Poll Question multiple choice (2)

- Shows the question number and the prompt.
- Shows the multiple choice answers.
- Shows IMG when provided.
- Allows the host to view all participants that have voted.
- Allows the host to view participants answers shown in *Figure 27D.*

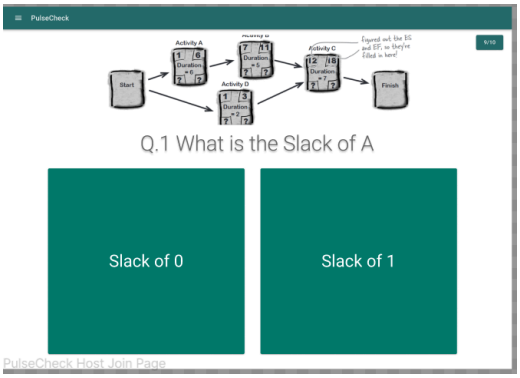
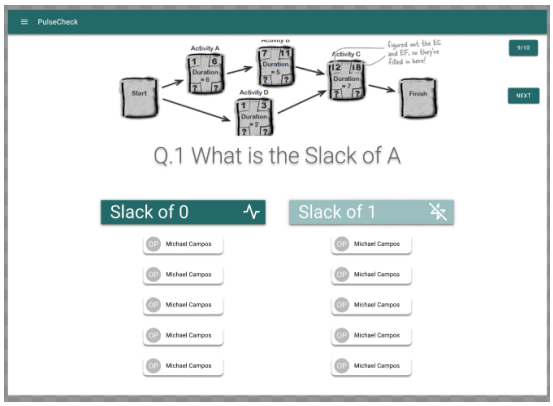


Figure 26D - Participants Poll Question multiple choice (2)

- Shows the question number and the prompt.
- Shows the multiple choice answers.
- Shows IMG when provided.
- Allows the participants to view the number of other participants that have voted.



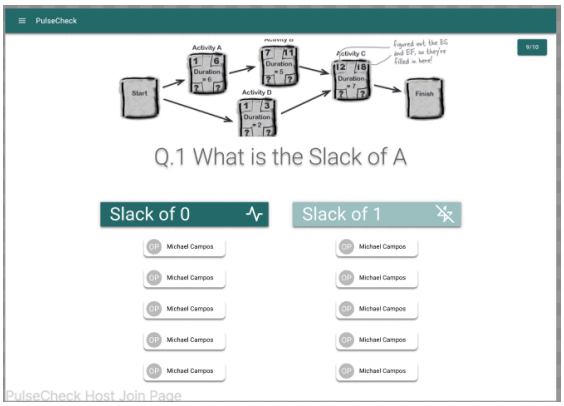


Figure 27D Host Poll Answers multiple choice (2)

- Shows the question number and the prompt.
- Shows the multiple choice answers.
- Shows IMG when provided
- Allows the host to view all participants that have voted
- Allows the host to go to the next question.
- When the question has an answer it shows the correct one.

Figure 28D - Participants Poll Answers multiple choice (2)

- Shows the question number and the prompt.
- Shows the multiple choice answers.
- Shows IMG when provided
- Allows the participants to view the number of other participants that have voted.
- When the question has an answer it shows the correct one.
-

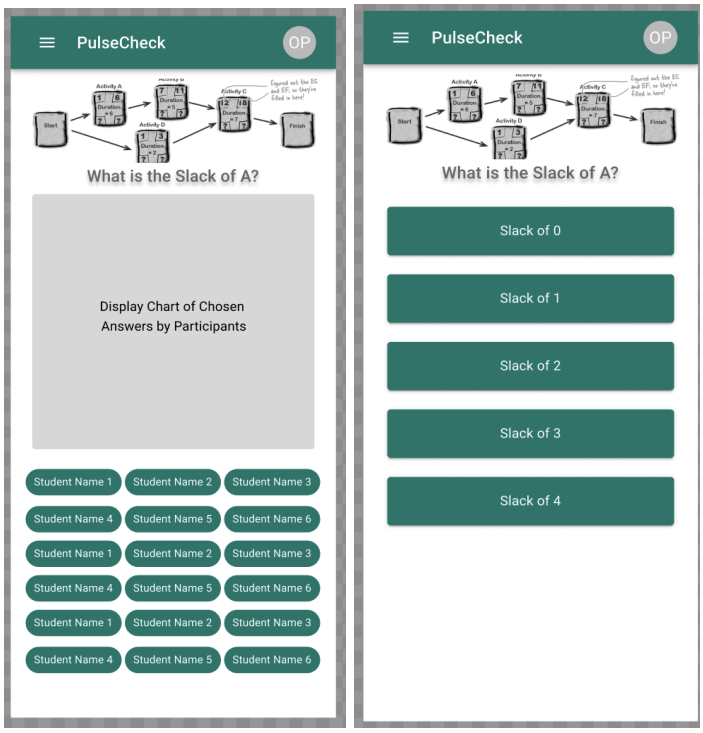


Figure 29P/D Host/Participants Poll Answers multiple choice (3-6)

- Same as *Figure 25D Figure 26D Figure 27D, and Figure 28D both for multiple choice of 3, 4, 5, 6 answers.*

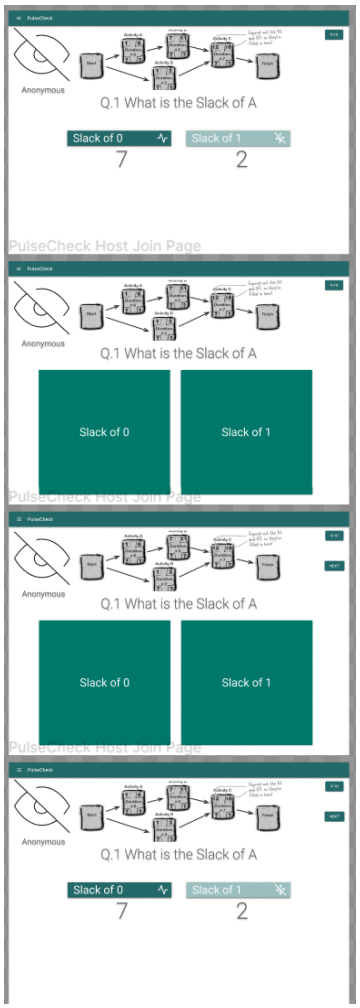


Figure 30D Host/Participants Anonymous Poll Answers multiple choice

- Same as *Figure 25D Figure 26D Figure 27D, and Figure 28D but with Anonymous polling activated.*

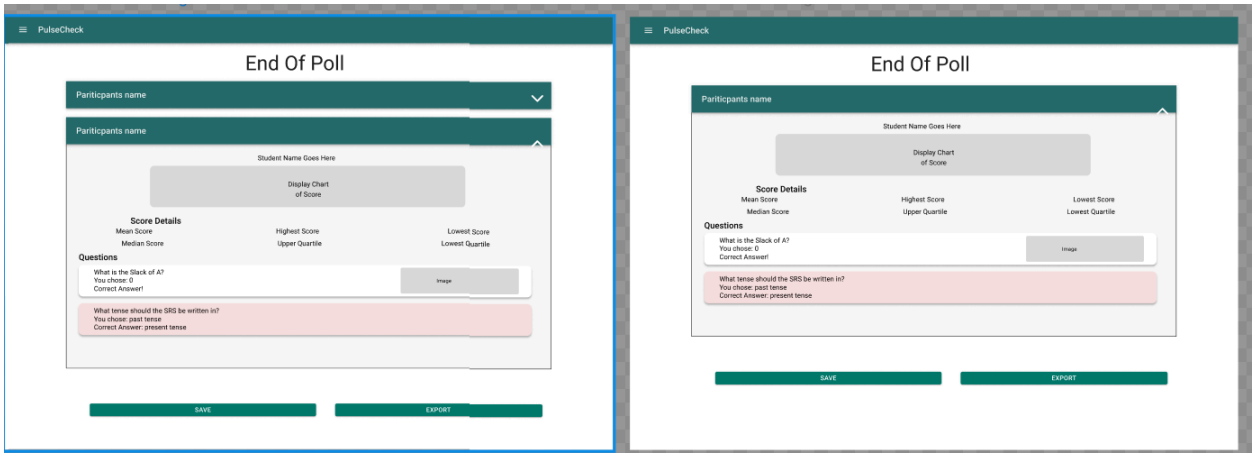


Figure 31D Host End of Poll

- Shows all participants and their scores/Answers at the end of poll
- Host can save or export all participants scores/Answers

Figure 32D - Participants End of Poll

multiple choice (2)

- Show participants scores/answers at the end of poll
- participants can save or export s scores/Answers

3.1.2 Hardware Interfaces

This section describes the hardware requirements and supported devices for the web application. The system is designed to run on standard web-enabled devices, including desktops, laptops, tablets, and mobile devices. It supports external display connections for in-class polling visualization. The application is for modern browsers and touchscreen interaction on mobile devices.

Supported Devices

- Desktops & Laptops: The system is compatible with modern web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, Safari.
- Tablets & Mobile Devices: The system features a fully responsive design to optimize for touchscreen interaction.
- Projectors & Large Screens : Supports external display connections to facilitate real-time polling visualization during lectures.

Minimum Device Requirements

- Processor: A mid-range dual-core processor (e.g. Intel i5 or equivalent) for smooth performance.
- Memory (RAM): 4GB RAM is a recommendation to handle real-time updates efficiently.
- Storage: No additional storage requirements beyond standard web browser caching.
- Network Connectivity: A stable WiFI (802.11n or higher) or Ethernet connection with at least 50 Mbps download speed is a recommendation for real-time polling functionality.

Compatibility Considerations

- Mobile OS Support: The application supports devices running Android 8.0+ and iOS 12+ for optimal performance.
- Screen Resolution: Minimum supported is 1280x720, but the UI dynamically adapts to different screen sizes dynamically.

- Peripheral Support: The system works with standard keyboard and mouse input, as well as touchscreen interfaces for mobile users.

3.1.3 Software Interfaces (API Interactions, Database Connections)

This section describes the software components and third-party services that the system interacts with. The web application relies on Firebase for backend services, including authentication, database management, and media storage. The frontend uses React with Material-UI and UI components. The system does not use REST APIs or Webhooks, but relies on Firebase's built-in SDKs and CLI for integration.

Firebase Services & Interactions

- Firebase Authentication: Manages user sign-in and authentication.
- Firestore: Stores structured data for user interactions, polls, and results.
- Firebase Cloud Storage: Stores media data such as images or files uploaded by users.
- Firebase CLI and APIs: Provides commands to set up Firebase for the development environment and direct access to Firebase services from the frontend.

Front-end Integration

- Firebase SDK: Provides built-in authentication and database functions.
- MUI (Material-UI): Supplies pre-built UI components and styling.
- React & React Router: Handles state management and navigation.

Development & Deployment

- IDE & Extensions: Visual Studio Code with relevant extensions for Firebase, ESLint, and Prettier.
- Github Workflows: Automates deployment to Firebase Hosting.
- Webpack: Bundles frontend assets into an optimized web app.
- TypeScript: Ensures type safety; transpiles to JavaScript before handling.
- Node.js & npm: Manages dependencies and development tools.
- ESLint & Prettier: Maintain code consistency and enforce style guidelines.

- Git & GitHub: Handle version control and collaboration.

3.1.4 Communication Interfaces (Real-Time Data Transmission, Web Sockets)

This section describes how the system handles communication between the frontend, backend, and external services. The web application relies on Firebase built-in real-time communication features to ensure seamless data synchronization and user interactions.

Real-Time Communication

- Firestore Real-Time Updates: Firestore automatically synchronizes data changes between the frontend and backend, ensuring the students and instructors receive live updates without manually refreshing the page.
- Firebase Authentication Events: The system listens for authentication state changes to manage user sessions dynamically.
- Cloud Storage Triggers: Firebase Cloud Storage notifies the application when users upload media files, ensuring that updated content is immediately available.

Client-Server Communication

- Firebase SDK: The frontend communicates with Firebase services directly using the Firebase SDK, eliminating the need for REST APIs or Webhooks.
- Firebase Security Rules: Firestore enforces data access permissions at the database level, preventing unauthorized access and ensuring secure data transactions.

External System Interactions

- GitHub Workflows: The system integrates with GitHub Actions to automate deployment processes and maintain code quality.

3.2 Functions

This section describes an overview of the essential functions and operations for the Real-Time In-Class Polling Web Application. These functions ensure seamless interaction between students and instructors, leveraging React, TypeScript, MaterialUI, and Firebase for optimal

performance and scalability. Key functional areas include user management, poll creation and management, real-time data updates, and results reporting.

Style Requirements

- SR-1
    - The system defines a custom theme using Material UI's built-in color palette. The primary color is set to teal/700, a built-in color in MUI's UI library. This color is in use consistently across buttons, links, headers, and other interactive elements.

Logged-Out Users Functional Requirements

Splash Page (refer to figure 1P/D)

- LU-1 Splash Page Access
    - The system displays the splash page for users who visit the site for the first time.
- LU-2 Brief Explanation
    - The splash page has a brief explanation of the web app and its purpose.
- LU-3 Get Started Button
    - The splash page has a button saying "GET STARTED" that redirects the user to the guest join poll page.
- LU-4 About Section
    - The splash page has an about section that goes into further detail of the web app.
- LU-5 Features Section
    - The home page has a features section that lists out the web apps key features.
- LU-6 FAQs Section
    - The home page has a FAQs section for frequently asked questions.

Navigation Menu (refer to figure 2P/D)

- NM-1 App Bar

    - Each page includes an app bar for simple navigation to the home page. The system always displays the app bar both for logged-in and logged-out users.
    - The app bar has a hamburger icon aligned to the left side of the app bar. The hamburger icon has an event handler to show a navigation menu to allow the user to navigate throughout the website.
    - On the right side of the hamburger icon is the name of the system, ***PulseCheck.***
- NM-2 Home Menu Item
    - The app bar has a “Home” menu item that navigates to the splash page.
    - This menu item only shows for logged-out users.
- NM-3 Features Menu Item
    - The app bar has a “Features” menu item that navigates to the features section on the home page that provides a brief overview of the web app’s features
    - This menu item only shows for logged-out users.
- NM-4 About Menu Item
    - The app bar has a “About” menu item that navigates to the about section on the home page.
    - This menu item only shows for logged-out users.
- NM-5 FAQs Menu Item
    - The app bar has a “FAQs” menu item that navigates to the frequently asked questions section on the home page.
    - This menu item only shows for logged-out users.
- NM-6 Privacy Policy Menu Item
    - The app bar has a “Privacy Policy” menu item that navigates to the privacy policy page.
    - This menu item only shows for logged-out users.
- NM-7 Terms of Service Menu Item

    - The app bar has a “Terms of Service” menu item that navigates to the terms of service page.
    - This menu item only shows for logged-out users.
- NM-8 Register Menu Item
    - The app bar has a “Register” menu item that navigates to the registration page.
    - This menu item only shows for logged-out users.
- NM-9 Login Menu Item
    - The app bar has a “Login” menu item that navigates to the login page.
    - This menu item only shows for logged-out users.
- NM-10 Response Design
    - The home page and all other pages are responsive and accessible on both mobile and desktop devices.
- NM-11 Consistent Styling
    - The app bar and navigation items have consistent styling across all pages using MaterialUI.
- NM-12 Access Control
    - Logged-out users are not able to access restricted pages for authenticated users.
- NM-13 Error Handling
    - The system redirects the logged-out user if they try to access restricted content. Restricted content is content that can only be accessed by a logged-in user.
- NM-14 Footer Links
    - Splash and Dashboard page has footer links to the privacy policy and terms of service pages.
- NM-15 Dashboard Menu Item
    - The navigation menu has a “Dashboard” menu item that navigates to the dashboard page.

    - This menu item only shows for logged-in users.
- NM-16 Join Poll Menu Item
    - The navigation menu has a “Join Poll” menu item that navigates to the join poll page.
    - This menu item only shows for logged-in users.
- NM-17 Results Menu Item
    - The navigation menu has a “Results” menu item that navigates to the results page.
    - This menu item only shows for logged-in users.
- NM-18 Profile Menu Item
    - The navigation menu has a “Profile” menu item that navigates to the profile page.
    - This menu item only shows for logged-in users.
- NM-19 Logout Menu Item
    - The navigation menu has a “Logout” menu item that logs the user out.
    - This menu item only shows for logged-in users.
- NM-20 Avatar Icon
    - Display an avatar icon aligning on the right side of the app bar.
    - Has an event handler to redirect the user to the profile page
    - This only shows for logged-in users.

Guest Join Poll Requirements

GJP-1: Guest Join Poll Form (refer to figure 3P/D)

- The system has a form to allow guest users to join a poll without creating an account. The form has a title (“Join Poll”)to indicate this where guest users may join a poll. The form has fields that say “Room Code” and “Display Name” where guest users can enter their room code to join a poll and enter their name they want to display when joining the poll.

GJP-1.1: Join Poll Button

- The form has a button that says “POLL UP”. On click, the guest user tries to join the poll using their given room code.

GJP-1.2: Create an Account Button

- *The form has a button that says “CREATE AN ACCOUNT” to allow the user to create an account to gain access to authorized features.*

User Management Functional Requirements

User Registration (refer to figure 5P/D)

UR-1: User Registration Form (refer to figure 5P/D)

- The system has a registration form where users can enter their email and password.
- The registration form layout is in the center of the screen for both desktop and mobile screen. All fields and controls are displayed in a column direction top to bottom.
- The form starts with a title that says “Register” to notify the user they are in the register page.
- The form has three fields in order, email to enter the user’s email, password to enter the user’s password, and retype password to retype the user’s password. Each field has their respective label to indicate the purpose of each field.
- The form has a button that says “REGISTER” that has an event handler to register an account via Firebase’s Authentication services.

UR-2: Password Validation

- The user enters their email and password, then retypes their password to ensure the user is typing in their intended password. If their password and retype password do not match, then display an error message using MUI's Snackbar to display the message (i.e. Passwords do not match).

UR-3: Error Handling

- The application displays appropriate error messages if registration fails. For example:
    - If the user enters an email that is already in use, then it displays an error message saying "Email already in use" using MUI's snackbar.

UR-4: Register with Google Button

- The user registers their account using their Google Account.
- The user proceeds through Google's authentication system to register their account.

UR-5: Register with Apple Button ***(REMOVED)***

- The user registers their account using their Apple account.
- The user proceeds through Apple's authentication system to register their account.

User Login (refer to figure 4P/D)

UL-1: User Login Form

- The system has a login form where users can enter their email and password.
- The login form layout is in the center of the screen for both desktop and mobile screen. All fields and controls are displayed in a column direction top to bottom.
- The form has text fields for the user to enter their email and password. The style of the fields are in MUI's default style. The fields are entered

UL-1.1 User Login with Google Button

- The login form provides a button to login the user with their Google account.

- The button has Google's 'G' icon and text displaying "Continue with Google" to indicate the user can login with their Google account. The button's event handler prompts the user to login with their Google account via Google's authentication service.

UL-1.2 User Login with Apple Button ***(REMOVED)***

- The login form provides a button to login the user with their Apple account.
- The button has Apple's icon and text displaying "Continue with Apple" to indicate the user can login with their Apple account. The button's event handler prompts the user to login with the Apple account via Apple's authentication service.

UL-2 Error Handling

- The system displays appropriate error messages if login fails (e.g. incorrect email or password) using MUI's snackbar.

Privacy Policy (refer to figure 6P/D)

PP-1

- Displays the system's privacy policy in respect to figure 6P/D.

Terms of Service (refer to figure 7P/D)

TOS-1

- Displays the system's terms of service in respect to figure 7P/D.

User Dashboard (refer to figure 8P/D)

UD-1: Most Recent Poll Chart

- Displays the most recent poll in a graphical chart. On click, displays the results of the poll.

UD-2 Create Poll Button

- Display a button that says "CREATE POLL". On click, navigate the user to the poll editor page.

UD-3: Join Poll Button

- Display a button that says "JOIN POLL". On click, navigate the user to the join poll page.

UD-4: Most Recent Polls

- Display a title that says “Recent Polls” to tell the user what the purpose of the content below the title is.
- Display a list of cards where each card displays a recent poll the user has taken, displays the poll name, number of participants in the poll. And if applicable, displays the user’s score if the poll is a quiz.

Join Poll (Authenticated User) (refer to figure 10P)

JP-1: Join Poll Form

- Displays a form to allow the user to join a poll. The form contains two fields for the user to enter the room code of the poll and their display name.

JP-2 Join Poll Button

- Display a button that says “POLL UP” to allow the user to try and join with their room code and display name.

JP-3 Error Handling

- The system displays appropriate error messages (i.e. invalid room code) using MUI’s snackbar.

Poll Results Manager (refer to figure 13P)

PRM-1: Result Tabs

- The system displays filter options to filter polls that the user owns and polls that the user participated in. The current filter the user chooses has focus.

PRM-2: Filter Polls Field

- The system displays a textfield to allow the user to filter out polls by the name of the poll.

PRM-3 Poll List

- The system displays a list of polls depending on the filter option, is either polls the user owns or polls the user participated in. Each poll displays using a card that displays the name of the poll, number of participants, and if applicable, the user’s score for that poll.

- On click of a poll card, the user navigates to a view of the poll's results.

User Profile Management (refer to figure 14P)

UPM-1 Profile Page

- The application provides a profile page where users can view and update their profile information. Information includes the user's email and display name.

UPM-2: Profile Email Field

- The system displays the user's current email in use for their account.
- Users may edit the field to update their profile.

UPM-3: Profile Display Name Field

- The system displays the user's current display name in use for polls they join.
- Users may edit the field to update their profile.

UPM-4 Theme Option

- The user has the option to set the theme of the system. Options are light, dark, and system themes.

UPM-5: Save Changes Button

- The system displays the "SAVE CHANGES" button to allow the user to save any changes on their profile.

User Post-Quiz Statistics Storage (refer to figure 32D)

UQS-1 Post Quiz Statistic Display

- After the user has completed the quiz, the system displays the user's final score, the questions and its answers relative to the user's answers.
- The system displays a "SAVE" button for guest users.
- Logged in users do not see the "SAVE" button, however their results are automatically saved on the database, and are directly linked to their accounts

UQS-2 Save Button

- The system redirects the user to the log-in page, where they have the option to log into their accounts so that their results are saved, or create a new account.
- If a Guest user opts to not log in or create an account, their work is saved in the database, but is not linked to their id; meaning the user is not able to access them after they leave the page, but the host can still view their results.

UQS-3 Export Button

- The system takes a snapshot of their results and uses the user's system file manager to save their results.

UQS-4 Viewing Past Quizzes

- Displays all previously hosted and completed quizzes in the form of individual files. Data is retrieved from Firebase Database
- The system also displays a drop menu option for filtering and sorting saved quizzes.

UQS-5 Saved Quizzes Drop Menu items

- Items on the drop menu include:
    - Filter: Created by you
    - Filter: Participated in
    - Sort Ascending (A-Z)
    - Sort Descending (Z-A)
    - Sort Ascending (Newest - Oldest)
    - Sort Descending (Oldest - Newest)

User Poll Creation (refer to figure 20D)

UQC-1 Question Settings

- The application provides users the following options
    - Whether the participants are anonymous
    - Whether participants will get points for correct answers

    - Whether the questions are timed
    - Whether the chat is enabled
    - Whether questions are being scored for correctness
    - The number of answer options for each question

UQC-1.1 Question Input

- The application provides a box where the user can input the full question they intend to ask
- The application provides spaces for which the user can input as many multiple choice answer options as they've specified.
- The application provides check boxes in order to select the correct multiple choice answers.

UQC-1.2 Upload Image Button

- The application provides a button that allows the user to upload an image to complement the question content
- The user is able to upload SVG, PNG, JPG, or GIF files.

UQC-1.3 Generate with AI button

- Within the poll editor, the "Generate with AI" button opens up an upload image/pdf pop up.
- Uploaded image or pdf gets sent to Vertex AI model for text extraction.
- Uploading a pdf or image with "Generate with AI" button, triggers the entirety of the AI pipeline.

UQC - 1.4 Vertex AI (AI Pipeline)

- The AI model used and its specifications are as follows:
    - Model: gemini-2.0-flash-001
    - Input type: Multimodal (this system inputs image (pdfs are classified as images) and text.

    - Maximum file size: 20 MB
    - Accessed with api calls through Vertex AI within firebase
- The Vertex AI api is accessed once user uploads image through the “Generate with AI “ button from UQC-1.3
- Once the model receives an image or pdf file, it is prompted to extract text from files.
- When text is extracted, the model gets prompted to generate 5 questions based on the extracted text, with 4 multiple choice options for each question; of which 3 answers are incorrect and 1 is correct. The output is returned in json format, specifying what the correct answer is for each respective question.
- Once model returns the json, it is used to autofill the following question components within the poll editor:
    - Question
    - The multiple choice options
    - Selects correct answer for each question

Quiz Completion for Non-Hosts

This section pertains to the completion of the quiz for Non-Hosts. Non-hosts are users that are answering a quiz of which they do not have admin controls.

UCNH-1 Waiting Room

- The waiting room is the page in which non-host users are redirected automatically after entering quiz code
- In this page, users see the number of people in the waiting room, their names and the quiz title. The system also displays the main menu at the top right corner.
- The number of participants and their names update in realtime as users join.
- The system redirects users to the first question only when the Host starts the quiz or the maximum number of participants is reached.

UCNH-2 Viewing and Answering Questions

- Once the quiz starts, the system begins to display the questions and its possible answers in a multiple choice format.
- Users are able to select their answers and change them as many times are they need, up until the hosts dictates the system to move onto the next question
- There is no back button for users to return to a previous question.
- If the user leaves the quiz or disconnects, the system pulls their past answers from cache. However, if the host has moved to later questions, the user automatically receives 0 points for the missed answers.
- Once the host has terminated the quiz, the system automatically redirects users to their score page.

Quiz Controls for Hosts

UQCH-1 Accessing Previously Created Quiz

- Hosts are able to access a previously created poll by clicking the “run” button in the host join page.
- The “run” button redirects hosts to the “Run a Poll” page, where all the polls created by that host are displayed; along with its respective name, creation date and “RUN!” button

UQCH-2 Initiating Quiz and Waiting Room (refer to figure 22D/24D)

- When the user selects the “RUN!” button on the “Run a Poll” page, the system automatically redirects them to the quiz waiting room and opens the waiting room to non-host users.
- The waiting room displays the access code for the quiz, along with the quantity and names of the users that are joining in real time.
- The waiting room displays a “Start” button, which allows the host to initiate the quiz, or the system starts it automatically if the waiting room reaches the maximum number of participants.

UQCH-3 In-Quiz View (refer to figures 29P/D)

- For each question, the system displays the question, the possible answers and who is selecting which answers in real time
- The system also displays a ratio of how many people have answered to how many people are in the quiz.
- On the upper right side of the page, there is a “next” button where they can move onto the next question, which automatically updates the page of non-host users as well.

UQCH-4 In-Quiz View (Anonymous)

- For quizzes that are preset to have anonymous responders, the view remains the same as regular quizzes, with the exception that names are not displayed for answers selected by users.
- Alternatively, the host sees the number of people that have chosen the given categories, but not their names.

UQCH-5 Quiz Termination

- Quizzes are terminated when Host selects “End Quiz” button on the last question
- The system redirects all users to the quiz statistics page.

LBG-1 Lobby Game

- The system displays an interactive memory game to users while they wait in the lobby before a session begins.
- A 4x4 grid of facedown cards is presented to the user. Each card conceals an animal icon.
- Users interact with the grid by clicking on cards to reveal the underlying animal icon.
- The objective of the game is to match all animal pairs.

This Page Intentionally Left Blank

3.3 Performance Requirements

This section defines the system's performance accomplishments under more specific, but common environmental conditions that can arise while the system is in use. Several categories apply to this.

PR-1: Response Management

- UI interactions, such as updates between themes or colors, have a latency reflection within 100ms
- Authentication for users completes in under 3 seconds under stable network conditions.

PR-2: Latency

- The system relays real-time polling results of user answers with latency no more than 500ms after new response submissions.

PR-3: Scalability

- The system handles a minimum of 100 new users created per hour for each IP address without losing performance
- PulseCheck Handles real time data syncing to provide smooth scaling

3.4 Design Constraints

This section outlines the limitations and restrictions the system endorses into the product's design features. More specifically, these limitations adhere to the restrictions being put upon the user interface design and accessibility:

- No slides included
- Maximum limit of 100 questions
- Character limit on questions

- No written responses
- Max of 300 users can participate in one poll at a time
- Can only use certain languages: (English)
- UI does not support a maxWidth of 'sm' where 'sm' is the smallest possible max width that is supported in MUI.

3.5 Software System Attributes

3.5.1 Reliability

The system sustains a high performance of dependability to ensure pre-time polling and quizzes to function without disruptions. System specifications include:

- The System has 99.5% uptime compared to its downtime
- All data is backed up on the cloud with recovery access for users with created logins
- Accessibility to the system's web services is 24/7 under high traffic and dynamically adjusts resources to accommodate increased number of users.

3.5.2 Availability

The system's accessibility to users is critical to ensuring a smooth user experience. This section outlines key attributes such as uptime, maintenance, and monitorization of the system that define the system's robustness and efficiency throughout the duration of use. The upkeep in maintenance provides a secure platform use that maintains data protection as well accessibility even under high traffic or unexpected disruptions. The system services include:

- Accessibility to system web services 24/7 under high traffic and dynamically adjusts resources to accommodate increased number of users.
- When web services are temporarily down, the system provides cached results and redirects traffic to a backup system.
- Ensurance of a backup database is set to prevent data loss.

- Latency reduction to deploy data across regions to improve response time.
- Users are notified about maintenance to system services.

3.5.3 Security

The system ensures a secure authentication mechanism to prevent unauthorized access. All user data, including responses in transit, shall be encrypted using secure protocols such as HTTPS to ensure data integrity. Personal data shall be securely stored and managed in compliance with relevant data protection regulations. Role based access control with appropriate permissions depending on the user (host or participant) is upheld. Data that is stored for an extended period of time are to be considered up for deletion after the date of expiration is up (after 14 days).

3.5.4 Maintainability

The code and documentation for this system utilize consistent conventions for naming and syntax throughout the product. For the portion of the system that is written in TypeScript, the external tool ESLint is used to help enforce the convention standards. Files, functions, methods, and additional components include in-file documentation describing the functionality of each portion. To facilitate handoff to future contributors, the documentation of the frameworks used, is found in Section 1.4: References of this document. In order to reduce redundancy in the system, functions and methods are written in a generalized manner. New functions/methods are implemented when existing ones cannot fulfill the task at hand. Additionally, in order to protect the integrity of the system, new features and all updates are tested several times before being implemented on the main system.

3.5.5 Portability

The frontend of the system is built with React, which uses JavaScript as one of its main coding languages. JavaScript is a highly portable language, therefore aiding in future environment transfers. Additional libraries within React, such as Material-UI, allow the user interface to self-adjust to different screen sizes. Furthermore, since this system is accessed via web browser, operating system portability on the user end does not pose constraints. On the database portions of the system, Firebase is very portable within its different tiers but less portable to external database companies. However, due to the small scale size of the system, database portability does not pose serious immediate concerns.

This Page Intentionally Left Blank