



CPRO2902  
Capstone II  
Project Proposal – Mental Health Research Aid Application

Ryan McGrandle - 000299562  
Kaden de Frece - 000364780  
Bergen Cunningham - 000288719

## Table of Contents

I.	<a href="#">Table of Figures</a>	– Page III
II.	<a href="#">Introduction</a>	– Page IV
III.	<a href="#">Purpose</a>	– Page VI
IV.	<a href="#">Client’s Needs</a>	– Page VII
V.	<a href="#">Security Requirements</a>	– Page IX
VI.	<a href="#">Technical Requirements</a>	– Page XI
VII.	<a href="#">Timeline</a>	– Page XIII
VIII.	<a href="#">Responsive Design</a>	– Page XV
IX.	<a href="#">User Interface</a>	– Page XVI
X.	<a href="#">Relational Database</a>	– Page XVIII
XI.	<a href="#">Application Use Case</a>	– Page XIX
XII.	<a href="#">Diagrams</a>	- Page XX
	A. <a href="#">Site Map</a>	– Page XX
	B. <a href="#">Use Case Diagram</a>	– Page XXI
	C. <a href="#">Conceptual ERD</a>	– Page XXII
XIII.	<a href="#">Conclusion</a>	– Page XXIII
XIV.	<a href="#">Appendix</a>	– Page XXIV
XV.	<a href="#">Sources</a>	– PageXXV

## Table of Figures

- I. [\*Figure 1 – Mental Health Research Aid Site Map\*](#) – Page XX
- II. [\*Figure 2 – Researcher Use Case Diagram\*](#) – Page XXI
- III. [\*Figure 3 – Conceptual ERD\*](#) – Page XXII

## Introduction

Studies are an extremely important aspect of modern academia. To the point that they're almost exclusively what many students do, especially towards the end of their education in the sciences. Understanding this, we ended up interviewing the current Student Body President, Devyn Shannon, who is actively setting up to do his own studies on mental health and the effects of tabletop role-playing games as a form of therapy. In our interview, we came to understand exactly what he needed for his specific program, and how we could create a general health study aid and create modules for it to perform his exact needs. After acquiring a list of needs from our client, we got to work on the exact path we wanted to take this project on. Settling eventually on the aforementioned Mental Health Research Aid Application (title pending).

This application will be composed of a few core pieces to branch out and create a robust piece of software. The first of the core pieces is the database. We decided to go with SQL [1] instead of a NoSQL [2] approach such as MongoDB [3] as the number one priority from our client was participant security. This is extremely sensitive data we're working with and with a priority of security over speed, it seemed to us and the client that it was the optimal choice. SQL is also very nicely embedded within the C# ASP.NET Core environment, which is what we will be building this application in.

The second core aspect to this software is going to be form integration, interpretation, and general research tools. We want to give researchers the ability to add any forms they need to fill out and attach to a participant's profile, and have the program store and interpret that data for ease of viewing. This will be in the forms of graphs, tables, and potentially even outputting results depending on the form. Forms will be interpreted by the program as a Portable Document Format file (PDF) and generated as a webform inside the application with appropriate fields to store in the database. Researchers will also be able to add notes to a given participant based on the session, as well as audio files with playback.

Our third cornerstone of this project are the access security levels. As mentioned in the database section, this project has personal information that absolutely cannot be compromised. If the data is prompted to be deleted, the one initializing the delete must have proper clearance, must verify their intention, and the data will be scrubbed from both the database and its backup. If a researcher wants to view specific data or graphs, they must have the proper level of clearance, or they will be denied and rerouted. And much of the data will only be able to be interpreted by the program itself, hashing sensitive information such as names and addresses.

By focusing on these core pieces of our application, we truly believe that we will be able to present our client with a complete and powerful full stack application that is both secure and user-friendly. Creating our application in such a modular way also opens up its use for studies beyond what just our client needs, and can deploy fully for educators and students to use as a powerful tool in their arsenal. In this report, we hope to show the value of the work we are doing, and that you, the reader, will be as excited as we are to get this project underway.

## Purpose

The purpose of this project is to develop a secure and flexible web-based application to assist researchers in managing, storing, and analyzing sensitive mental health study data. Specifically, the application is designed to support research initiatives that involve participant studies, form submissions, session tracking, and data visualization with a primary focus on ensuring data security and anonymity.

The Mental Health Research Aid Application will expedite the research process by providing tools for form integration, session logging, and participant management. It will enable researchers to conduct studies with full confidence that personal information is protected through secure authentication methods, encryption, and role-based access control. Additionally, the platform will be modular, allowing it to be adapted for different research needs beyond the initial project, making it a versatile tool for other studies and institutions.

By focusing on security, usability, and future scalability, this application will not only meet the specific needs of the client in his study on the therapeutic benefits of tabletop role-playing games but also serve as a robust framework for researchers conducting similar studies in mental health and beyond.

## Client's Needs

In our meeting with our client, he requested many great features and design aspects of our program. The following are the most important client needs as expressed in said meeting, a description of the requirement, and our current implementation strategy.

### Requirement:

- Complete data destruction.

### Description:

- In the event of a participant leaving the study, measures must be taken to ensure that the entirety of the participant's data is removed from the database.

### Implementation:

- Triggers within the database to completely purge requested data.
- Checks that stretch across the entire program to ensure data consistency and reliability.

### Requirement:

- Session tracking

### Description:

- Every session will have multiple details that are automatically logged and stored by the program based off of the input of the researchers.

### Implementation:

- The program front end, back end and SQL database will all output logs to be stored with complete tracking of all activities in a session.
- Logs will be categorized for easier navigation, and accessible to the highest tiers of clearance.

### Requirement:

- Form and document integration

### Description:

- Researchers will be able to upload and store forms, notes, their data, and audio with playback attached to each session of a study.

### Implementation:

- Files will be stored securely within C#'s file storing system, access to file structure will require proper clearance.

Requirement:

- Two-factor authentication

Description:

- Researchers must register with two-factor authentication in order to sign in to their account for security purposes.

Implementation:

- A secure two-factor authentication solution will be applied to the program, the specifics are TBD, but likely using Google's API or text-message based 2FA.

Requirement:

- Heavily secured information

Description:

- Client has stated multiple times that the security of the participant's information is the number one priority.

Implementation:

- Password, name, address, etc. personal information hashing.
- Tightly locked access levels.
- Thorough testing against SQL injection, compromised accounts and role based delivery of views.



## Security Requirements

Given the sensitive data handled by the Mental Health Research Aid Application, security is paramount. These measures ensure participant data, study information, and user credentials are thoroughly protected:

- Anonymity & Data Protection:
  - Data Anonymization: Participant data (e.g., names, emails) will be anonymized using hashing or encryption to maintain confidentiality.
  - Controlled Access: Only researchers with proper authorization can access sensitive information. Others will only view anonymized or aggregated data.
- Complete Data Destruction:
  - Cascading Triggers: When a participant's data is deleted, it will be automatically removed from all related tables and backups to ensure complete erasure.
  - Verified Deletion: Multiple verification steps will confirm that all traces of data are eliminated from the system.
- Personal Information Hashing:
  - Password Encryption: Passwords will be securely hashed using bcrypt or Argon2, combined with unique salts to ensure security even in the event of a data breach.
  - Sensitive Data Hashing: Other personal data, such as names and contact details, will also be hashed to protect against unauthorized access.
- Role-Based Access Control (RBAC):
  - Tiered Access Levels: Users are assigned roles (e.g., admin, researcher) that dictate their access to specific data and functions. This ensures only authorized personnel can modify or view sensitive information.
  - Authorization Policies: Different policies will govern what users at each access level can do, with admin-level users having full control, while researchers have limited access based on their role in a study.
  - Admin-Only Account Creation: Only administrators can create new accounts by sending expirable email tokens to new researchers, ensuring no unauthorized accounts are created.
- Two-Factor Authentication (2FA):
  - Mandatory 2FA: All users must enable two-factor authentication to access their accounts. This may be implemented using third-party solutions like Google Authenticator or SMS-based 2FA.
  - Expirable Tokens: Tokens used for account creation or password resets will have a limited lifespan (e.g., 24 hours) to prevent reuse or exploitation.
- Data Encryption:
  - Data at Rest: All sensitive data, including participant information, will be encrypted using AES-256 to ensure that even if the database is compromised, the data remains secure.

- Data in Transit: All data transfers between the server and client will be encrypted using TLS to protect against interception or eavesdropping.
- Protection Against SQL Injection:
  - Parameterized Queries: Entity Framework Core will enforce parameterized queries, preventing SQL injection attacks and ensuring that malicious SQL cannot be executed.
  - Input Validation: All user inputs will be sanitized and validated before processing, reducing the risk of injection or exploitation.
- Account Lockout Mechanism:
  - Brute-Force Protection: After a set number of failed login attempts, the account will be temporarily locked, and the user will be notified to prevent brute-force attacks.
  - Password Reset: Secure email-based password reset will be in place, with confirmation required to regain access, ensuring only the account owner can reset the password.
- Audit Logging & Monitoring:
  - Comprehensive Logging: All key activities (logins, data modifications, account changes) will be logged securely for review. Only authorized admins can access these logs.
  - Monitoring for Suspicious Activity: The system will monitor for abnormal activities, such as repeated failed login attempts, unusual data access, or unauthorized role changes, and notify admins for investigation.

These security measures ensure the application protects sensitive data, ensures integrity, and safeguards user privacy. By following best security practices, risks of data breaches, unauthorized access, or cyber threats are significantly minimized.

## Technical Requirements

The following tools and technologies are required for the development and deployment of the Mental Health Research Aid Application:

- Development Environment:
  - Visual Studio: Primary IDE for C# ASP.NET Core development.
  - .NET Core SDK: Required for building and running the ASP.NET Core application.
  - Entity Framework Core: ORM for database operations and secure data handling.
- Database:
  - SQL Server: Relational database for securely storing participant data, session logs, and study information.
  - SQL Management Studio: Tool for managing the database schema, running queries, and performing administrative tasks.
- Version Control:
  - Git: For tracking code changes and managing development collaboration.
  - GitHub: Remote repositories for storing code and enabling team collaboration.
- Web Technologies:
  - HTML5, CSS3, JavaScript: Core technologies for front-end development to ensure a responsive and interactive user interface.
  - Tailwind CSS: Framework for creating mobile-friendly, responsive design.
  - Razor Pages: Server-side UI framework for dynamic, secure web pages.
- Security:
  - ASP.NET Identity: For managing user authentication, role-based access, and password management.
  - Two-Factor Authentication (2FA): Additional layer of security using Google Authenticator or SMS-based 2FA.
- Testing:
  - xUnit: For unit testing to ensure the application's functionality.
- Cross-Browser Testing:
  - Google Chrome, Firefox, Microsoft Edge: Ensuring compatibility and consistent performance across modern web browsers.
- Development Machines:
  - Processor: Intel Core i5 or AMD equivalent or higher, to handle development workloads and multiple processes.
  - RAM: Minimum 8GB
  - Storage: SSD with at least 50GB of free space for faster read/write operations, IDEs, SDKs, databases, and project files.
  - Operating System: Windows 10/11 (64-bit) for Visual Studio and SQL Server compatibility.

➤ Server

- Processor: Quad core Intel Xeon or AMD equivalent
- RAM: Minimum 16 GB
- Storage: 500 GB SSD
- Network: High-speed internet connection
- Operating System: Windows Server 2019

➤ Client Machines

- Processor: Minimum Intel Core i3 or AMD equivalent for standard web browsing and interaction with the application.
- RAM: Minimum 4GB for basic functionality
- Operating System: Windows 10/11, macOS, or Linux for compatibility with modern web browsers and security updates.
- Browser Compatibility: Latest versions of Google Chrome, Mozilla Firefox, or Microsoft Edge to ensure full compatibility with the web application.
- Internet Connection: Stable broadband connection with at least 5Mbps download/upload speed for smooth access to web-based resources.

## Timeline

### Phase 1: Proposal & Planning

- Sept 16, 2024: Project Proposal (10%) submitted.
  - Sept 16 – 22, 2024: Define scope, set up environments, finalize milestones.
  - Sept 23 – 25, 2024: Project Requirements Document submitted.
- 

### Phase 2: Project Plan Presentation

- Sept 26 – Oct 1, 2024: Finalize database schema and UI wireframes.
  - Oct 2 & 7, 2024: Project Plan Mini-Presentation.
- 

### Phase 3: Database & Core Setup

- Oct 8 – 15, 2024: Develop SQL database and basic CRUD functions.
  - Oct 16 – 27, 2024: Implement user authentication and access control.
- 

### Phase 4: Core Features Development

- Oct 28 – Nov 5, 2024: Build form submissions, session tracking, and file uploads.
  - Nov 6 – 12, 2024: Implement data visualization and encryption.
- 

### Phase 5: Design Presentation

- Nov 13 & 18, 2024: Analysis and Design Presentation.
- 

### Phase 6: Testing & Finalization

- Nov 19 – 24, 2024: Conduct security and final UI testing.
- Nov 25 – 27, 2024: Implementation and Testing.

---

## Phase 7: Final Report & Demo

- Dec 2, 2024: Project Report submitted.
- Dec 4, 2024: Technical Poster/Demo presented.

## Responsive Design

As this is a web-based application, responsive design is an extremely important aspect for client usability. Our client's main desire is to have the program be used on standard laptop and desktop aspect ratios, however in the future there may be researchers that prefer to use tablets, smartphones, or even non-standard aspect ratio PCs. To accommodate this, and future-proof our site, we're building in responsive design from the ground up. The following section will detail key components of our responsive design and how they will affect the overall application.

Component:

- Multi-device support

Description:

- The web application will work seamlessly across all devices and without any display issues.

Implementation:

- Thorough dynamic coding of all views and pages to ensure proper stacking of content as well as resizing.

Component:

- Seamless transition

Description:

- When moving between devices, items will be spaced appropriately and without confusing the user.

Implementation:

- Careful planning of page layout, ensuring consistency across devices and appropriate migration of elements in a way that makes sense.

Component:

- Content prioritization

Description:

- Ensures that the most relevant content for the application is displayed first based on the screen size.

Implementation:

- By setting triggers within the code (either CSS or within C#'s razor views) we can prioritize what is being displayed and how.

## User Interface

The artistic design of the site is often overlooked by computer programmers. However, to a user it's the first thing they notice and leaves a heavy impression of the product. If our design is ugly, outdated, and unappealing, then it doesn't matter how good our product is. No one will want to use it. For this reason, we are dedicating a substantial amount of resources to making sure the site looks good, feels good, and is laid out in a way that wows the user. In this section, we will outline some important design tenants that we'll be following.

### Design Tenant:

- Site Clarity

### Description:

- How easy is it for a user with no prior knowledge to use the site?

### Considerations:

- Appropriate icon location and sizing.
- User-friendly navigation with clear messaging.
- Proper use of layout to achieve a cohesive page.

### Design Tenant:

- Aesthetics

### Description:

- How nice does the site look?

### Considerations:

- Following modern design practices and habits.
- Proper use of colour theory to create an engaging and easy to look at UI.
- Pleasing item spacing where relevant.
- Appropriate use of icons, text and images.

### Design Tenant:

- Consistency

### Description:

- How does the site maintain a similar look across the entire application?

### Considerations:

- Maintaining a design scheme across the site.
- Ensuring icon use is consistent and related.
- Following best practices regarding colour palettes within the site.



Design Tenant:

- Accessibility

Description:

- Ensuring those who may have certain disabilities can still effectively navigate the site.

Considerations:

- Colour blindness in button, warning, and navigation.
- Ensuring alt text is always accessible in site images.
- All elements should be navigable via mouse or keyboard.

Design Tenant:

- Error Handling

Description:

- The site notifies the user of any exceptions or issues clearly and understandably.

Considerations:

- Effective use of pop up messages, hidden fields, and alerts to make sure the user is never confused or frustrated by not knowing why something isn't working.
- Offer users suggestions for the user's next steps (eg. password was incorrect, please try re-entering).

## Relational Database

For this web application, we have chosen to use the SQL database built in to C# ASP.NET Core. This choice offers several advantages aside from its superior security that align well with the needs of modern web development.

- Seamless Integration:
  - ASP.Net Core's built-in SQL is highly integrated with the framework, allowing for seamless interaction between the application and the database. This reduces development complexity by providing a unified environment for managing data access, which can accelerate development.
- Efficient Development with Entity Framework Core:
  - The built-in database leverages Entity Framework Core (EF Core), an object-relational mapper (ORM), which simplifies data manipulation and reduces the amount of boilerplate code needed to interact with the database. EF Core supports powerful querying capabilities and data integrity enforcement, which contributes to an efficient development process.
- Scalability:
  - The Scalability of SQL databases makes them a strong choice for growing applications. The built-in SQL support within ASP.NET Core ensures that the database can handle increasing traffic and data loads while maintaining performance.
- Data Consistency and Robustness:
  - SQL databases are known for their reliable data transactions and strong support for relational data management, ensuring data integrity and consistency in the web application.

## Application Use Case

### Use Case 1:

#### ➤ Researcher Joining a Study:

- Researcher receives an email with an invitation link
- Researcher clicks the link and is redirected to the Account Creation Page
- Researcher fills out the required details and submits the form to create an account
- Researcher is redirected to the login page and uses new credentials to log in
- Researcher has access to the study they were invited to join.

### Use Case 2:

#### ➤ Applying to Use the Software for a Study:

- Applicant navigates to the Home Page
- Applicant selects Apply for Study
- Applicant fills out and submits the application form with necessary details
- The generates a receipt of the application and notifies the Applicant that it is awaiting admin approval.
- Applicant receives invite link via email and may begin their study

### Use Case 3:

#### ➤ Admin Approving a Study:

- Admin navigates to the Home Page
- Admin logs in using their credentials
- Admin is directed to the Navigation Panel
- Admin selects Admin Panel
- Admin selects Study Approval Page
- Admin reviews pending study requests
- Admin approves or denies the study request
- If approved, the system changes the status to Approved and emails Applicant

### Use Case 4:

#### ➤ Researcher Reviewing Participant Data:

- Researcher is logged in with authorization access to study data
- Researcher navigates to the Data Review/Analysis page
- Researcher selects the specific study they are working on
- Researcher reviews the submitted data
- The system logs any interactions with the data for auditing purposes

## Site Map

The site map displays the website's structure and how users navigate between pages. Key Pages in our site will be the following:

### Home Page:

- The entry point for all users
- Accessible routes:
  - Log In
  - Apply for Study
  - About Us

### Navigation Panel:

- The primary page logged-in users will see
- Accessible routes:
  - Admin Panel
  - Participant Registration
  - New Form Submission
  - Participant Drop-Out/Deletion
  - Participant Form Submission
  - Researcher Permissions
  - View Submissions
  - Data Review and Analysis
  - Invite Researcher

This will be a large multipage site that we hope becomes a secure, versatile and useful research tool for those who need it.

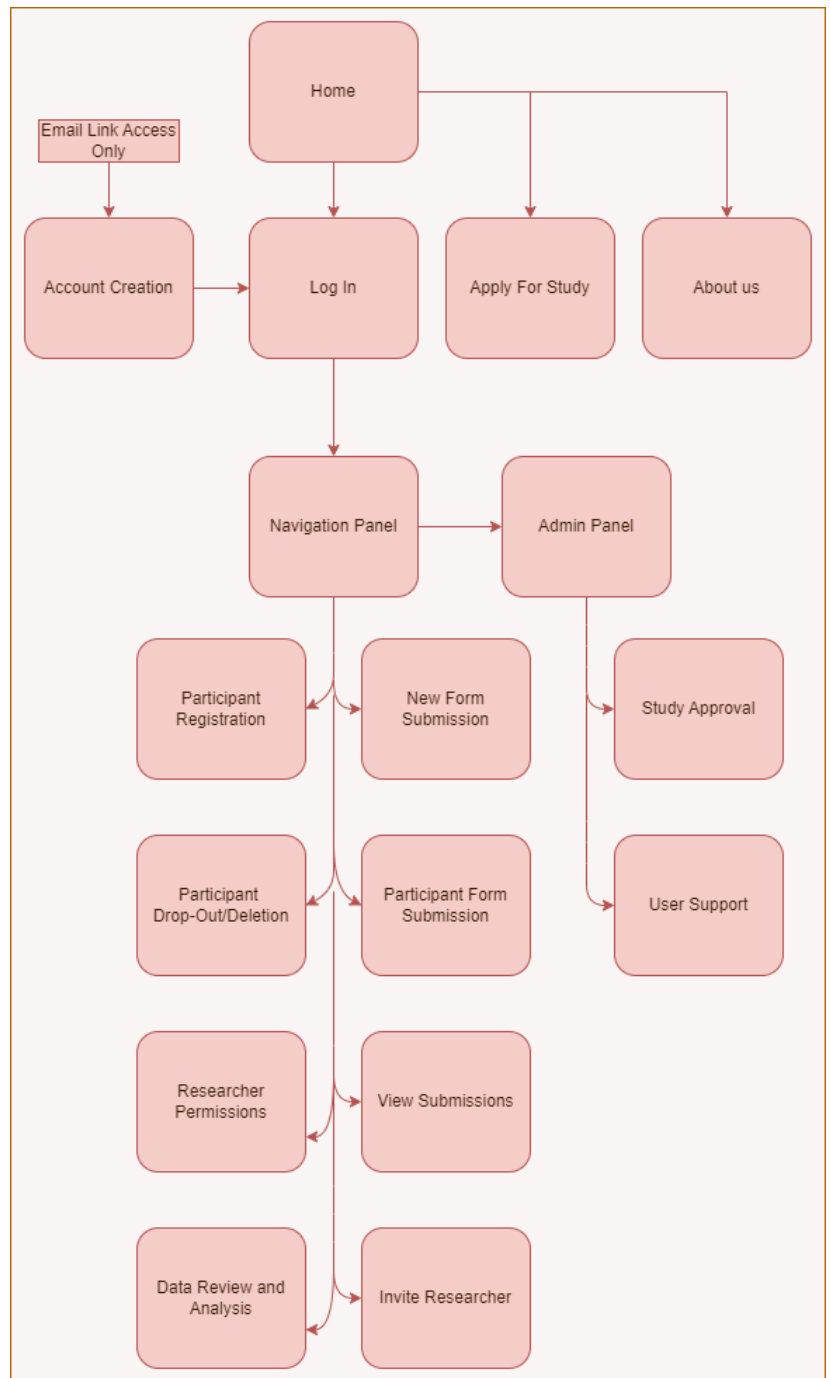


Figure 1 – Mental Health Research Aid Site Map

## Use Case Diagram

The use case diagram outlines how users interact with the system, showing user roles and the tasks they can perform. Additionally, the diagram aids in displaying the multi level authorization we plan to provide researchers with their study so that they may strictly control the data access of those they invite to the study.

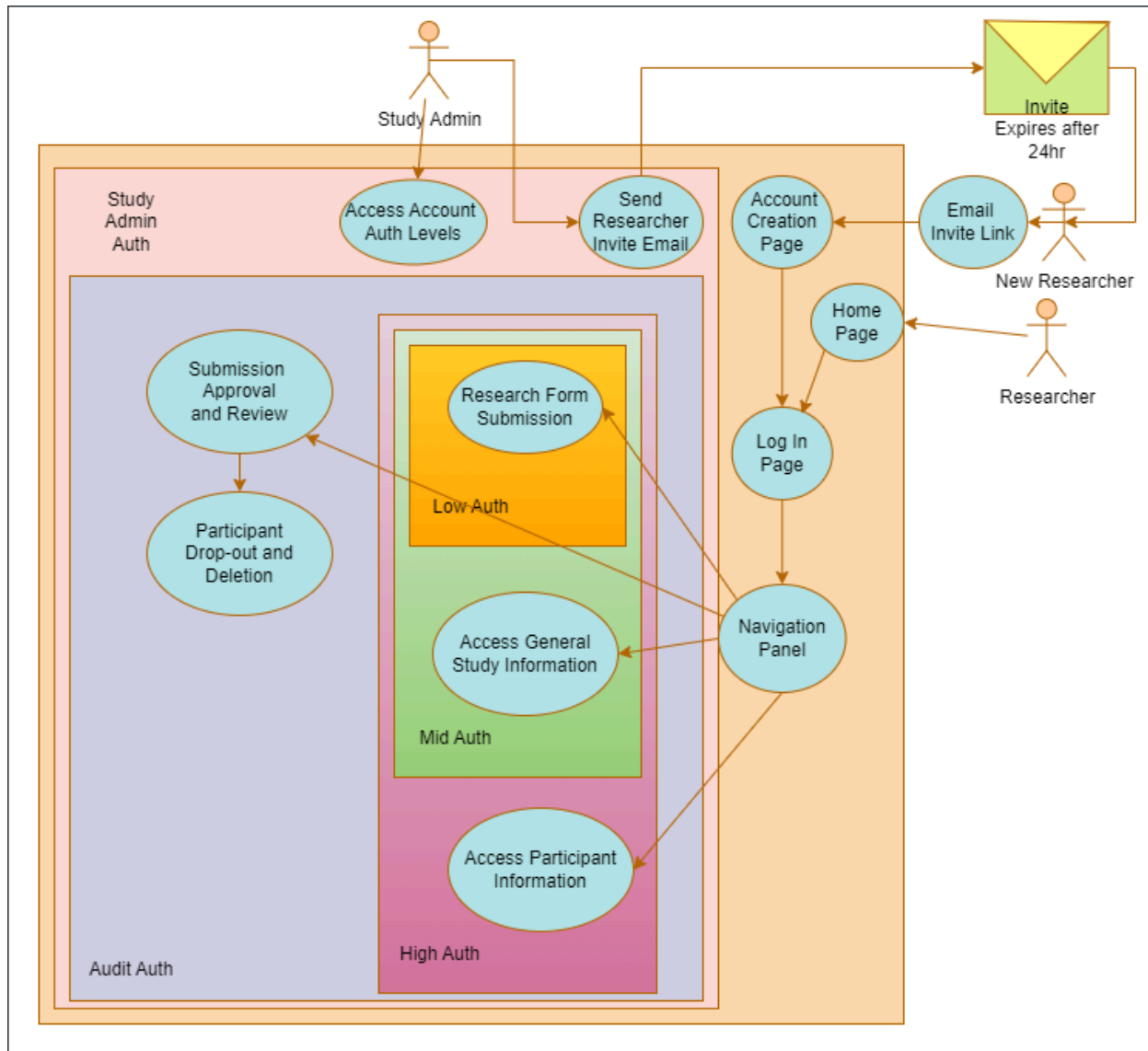


Figure 2 – Researcher Use Case Diagram

## Conceptual ERD

The conceptual ERD models the system's main entities and their relationships, providing a high-level view of the database structure.

Primary Entities:

➤ Forms

- Forms will be stored as rows, using multiple tables we will avoid null values for varying forms.

➤ Roles

- Roles will be the authorization levels that can be assigned to each researcher.

➤ Researchers

- Researchers will be the users utilizing the site and its functionalities.

➤ Participants

- Participants will not have access to the site, however their data will be filed by researchers.

➤ Sessions

- Sessions will be conducted by researchers and attended by participants.

➤ Studies

- Studies will be created by a researcher after being approved by the site, the researcher may submit new forms they wish to use in the study.

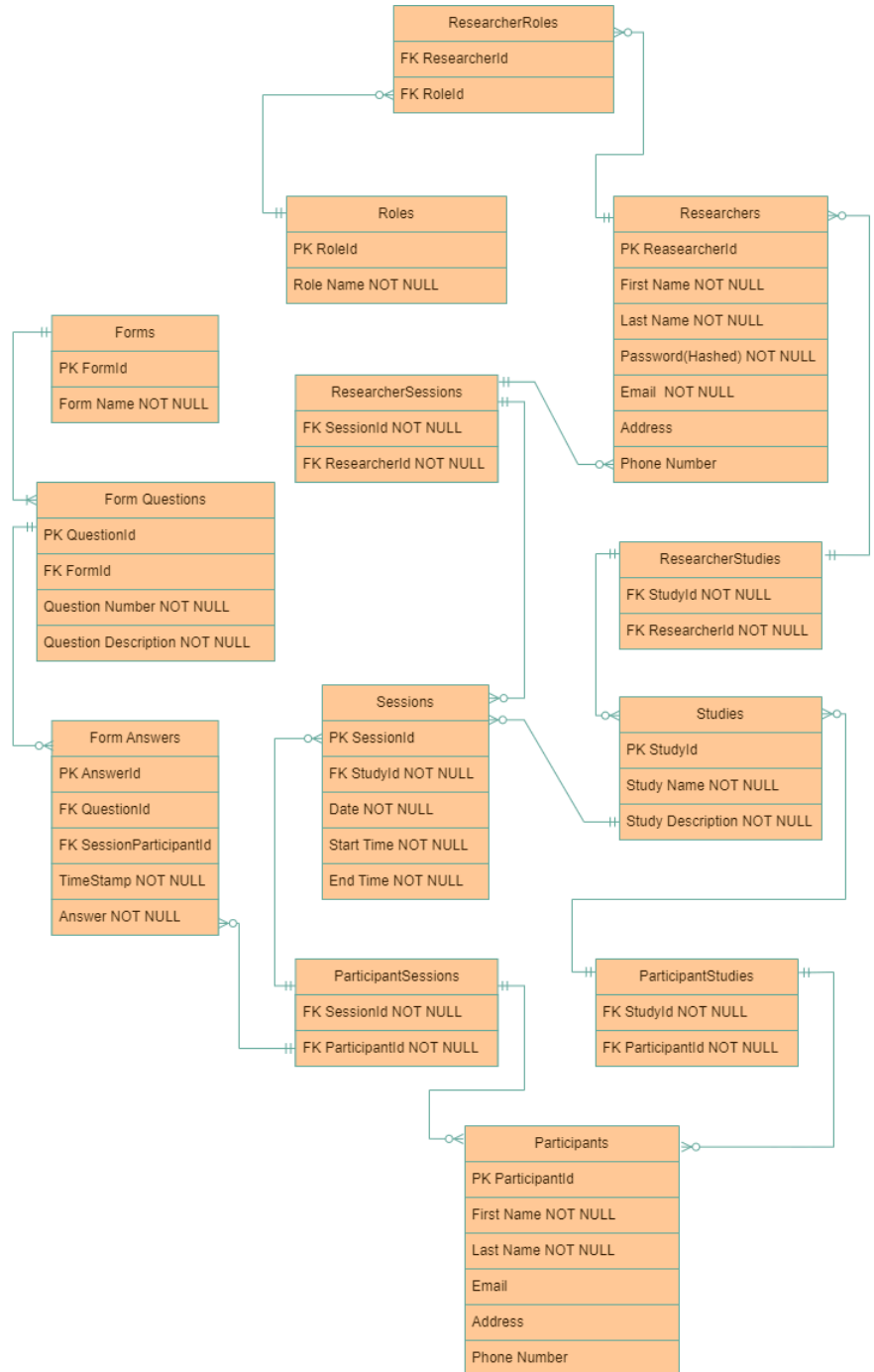


Figure 3 – Conceptual ERD

## Conclusion

In conclusion, we're very excited to bring forward this program to improve the lives of any researchers that wish to use it. We are also looking forward to developing a relationship with our client and delivering a tool to him that will exceed his every expectation. From the ground up, we are going to build a robust, fully featured program that researchers like our client will be using for, hopefully, years to come. While we aren't expecting to encounter zero hiccups along the way, we're confident in our skills and teamwork. We will deliver a product at the end of this Capstone and showcase our many computer science talents.

In this proposal, we covered a variety of topics with the hopes that it will shed some light on exactly what our intentions with this program are. We also hope that we have defined our timeline clearly and intentionally, we are optimistic about maintaining said timeline and having regular updates to report. The technologies we are implementing in this Capstone are ones with which our team has honed considerable skill over the course of our program. By showcasing this aspect of our talents, we believe each member of our team will shine.

From our front end, to the back end, to our database, each aspect of this program will be given our utmost care and respect. We understand the weight of working in a real-world setting such as the one we've established with our client. We will not let him down, nor will we put forth any less than our best into the Mental Health Research Aid Program. Thank you for your time, and we cannot wait to show you what we have in store.

Thank you.

– The Mental Health Research Aid Application Team

## Appendix

### [1] SQL

Stands for: Structured Query Language. A strongly typed querying language designed by IBM in the 1970s to handle the development of secure relational databases.

### [2] NoSQL

A non-relational database. Often stored as a JSON or other such file type. Useful for fast querying but less secure than SQL.

### [3] MongoDB

A NoSQL database product. <https://www.mongodb.com/> (Accessed 09-16-2024)



## Sources

ASP.NET Core Identity (Accessed 09-14-2024):

<https://learn.microsoft.com/en-us/training/modules/secure-aspnet-core-identity/>

Role-Based Authorization (Accessed 09-14-2024):

<https://learn.microsoft.com/en-us/aspnet/core/security/authorization/roles>

Entity Framework Core with Identity (Accessed 09-14-2024):

<https://dotnettutorials.net/course/asp-net-core-identity-tutorials/>