

CIS4813 Web Applications Programming: Semester Project | System Design Document

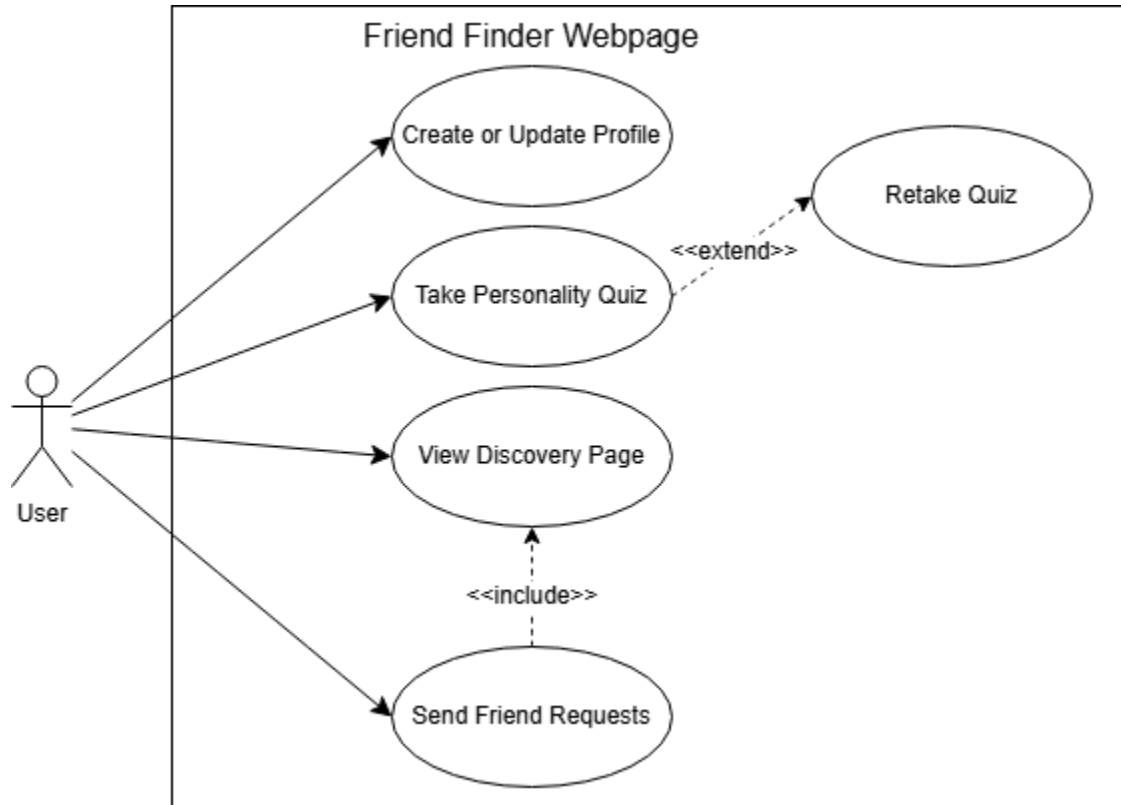
1. Student's Registration Information

- **Full Names & IDs:**
 - Carter Susi: 200649966, cts24d
 - Albert Velazques: 200198435, amv24h
 - Anika Guevarra: 200725744, asg23d
- **Course Name:** COP4813
- **Instructor Name:** Dr Ahsan Abdullah
- **Date of Submission:** 06-13-2025

1. Title and Team Members

- **Friend Finder** (Clifton Strengths Assessment)
- **Team Members:**
 - Carter Susi
 - Albert Velazques
 - Anika Guevarra

2. Use Case



Use Case 1: Create/Update Profile

Actor: User

Description: Users sign up for a public profile or can edit existing profile information (ex. name, bio, preferences)

Basic Flow: Users will sign-in to the platform using email and password, then navigate to the profile page. If new, users will fill out profile fields. If returning, users can update profile fields. Users must save changes to finalize any modifications.

Postcondition: Profile data is stored and can be viewed by other users on platform

Relationship: Standalone use case

Use Case 2: Take Personality Test

Actor: User

Description: User completes the Clifton Strengths Assessment to determine key personality traits used in compatibility matchmaking.

Basic Flow: Users will initiate the test and answer a series of multiple choice questions. Once completed, users will submit their test. The system will calculate and store the results to users' profiles.

Postcondition: Test results are used to determine friend compatibility

Exception Flow: If the user attempts to retake the test before 30 days, the system will block access and display an error message.

Relationship: Extends to "Retake test"

Use Case 3: View Discovery Page

Actor: User

Description: Users can explore other people, groups, or clubs based on test similarities and common interests listed on their profiles.

Precondition: Users must have completed the personality test and completed their profile.

Basic Flow: Users will open a Discovery Page where the system will retrieve a list of recommended matches using personality test compatibility and common interests/preferences in profile. Results will include individual users and relevant group/club recommendations. Users have the ability to sort/filter through options and can click on profiles and clubs/groups for more details.

Postcondition: The user will be presented with a customized list of matches and may take further action like view profiles, send requests, or join groups/clubs

Exception Flow: If there are no matches found, the system should display a message to the user (e.g., "No strong matches found. Try updating your profile or expanding interests")

Relationship: Includes "Send Friend Request"

Use Case 4: Send Friend Request

Actor: User

Description: Users can send friend requests to other matched/compatible users.

Basic Flow: Users browse through their discovery page results and can click on "Send Request" on a selected user's profile. The system will create a pending friend request in the database.

Postcondition: The request is tracked and awaits acceptance or rejection

Relationship: Included in "View Discovery Page"

Use Case 5: Retake Test

Actor: User

Description: Users are allowed to retake the personality test after a 30-day cooldown period.

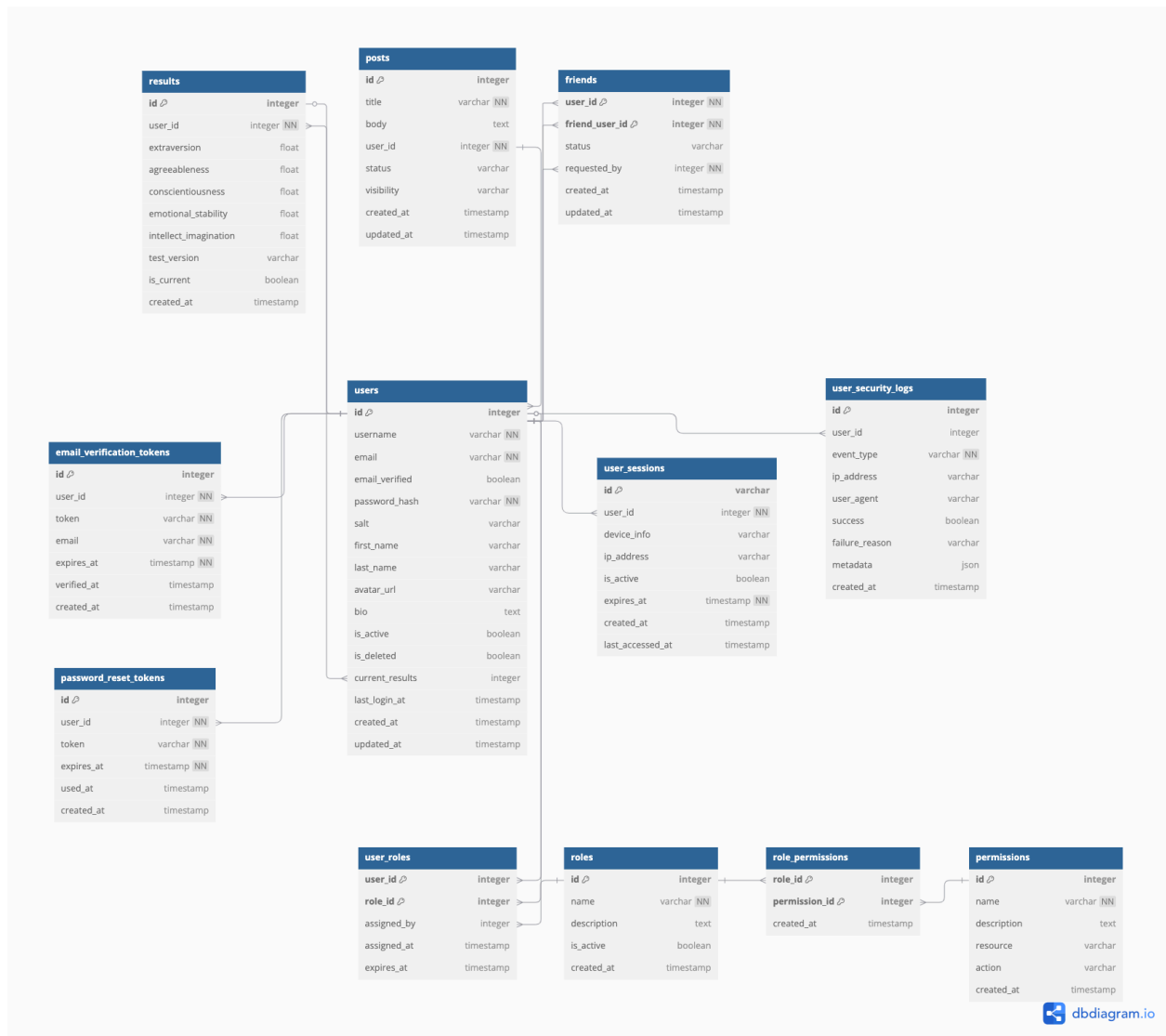
Basic Flow: Users will initiate the test retake and the system will check the timestamp of users' previous attempt. If eligible, the system will allow the user to proceed with retake; otherwise, error messages will display onto screen.

Postcondition: Users' new results will overwrite the previous results or are versioned alongside old results

Relationship: Extended from "Take Personality Test"

3. ER Diagram

PostgreSQL Schema



Core Tables Overview

users: Core user profiles with authentication credentials, personal information, and account status. Links to current personality test results.

user_sessions: Active user sessions for authentication management with expiration tracking.

password_reset_tokens: Temporary tokens for password reset functionality with expiration dates.

email_verification_tokens: Email verification tokens to confirm user email addresses during registration.

user_security_logs: Audit trail of security events including login attempts, password changes, and other authentication activities.

roles: User role definitions (admin, user, moderator, etc.)

permissions: Granular permissions for different actions and resources

role_permissions: Links roles to their allowed permissions

user_roles: Assigns roles to users with optional expiration dates

friends: Friend relationships between users with status tracking (pending, accepted, blocked) and request history.

results: Personality test results storing the Big Five personality traits with versioning support.

posts: User-generated content with visibility controls and status management.

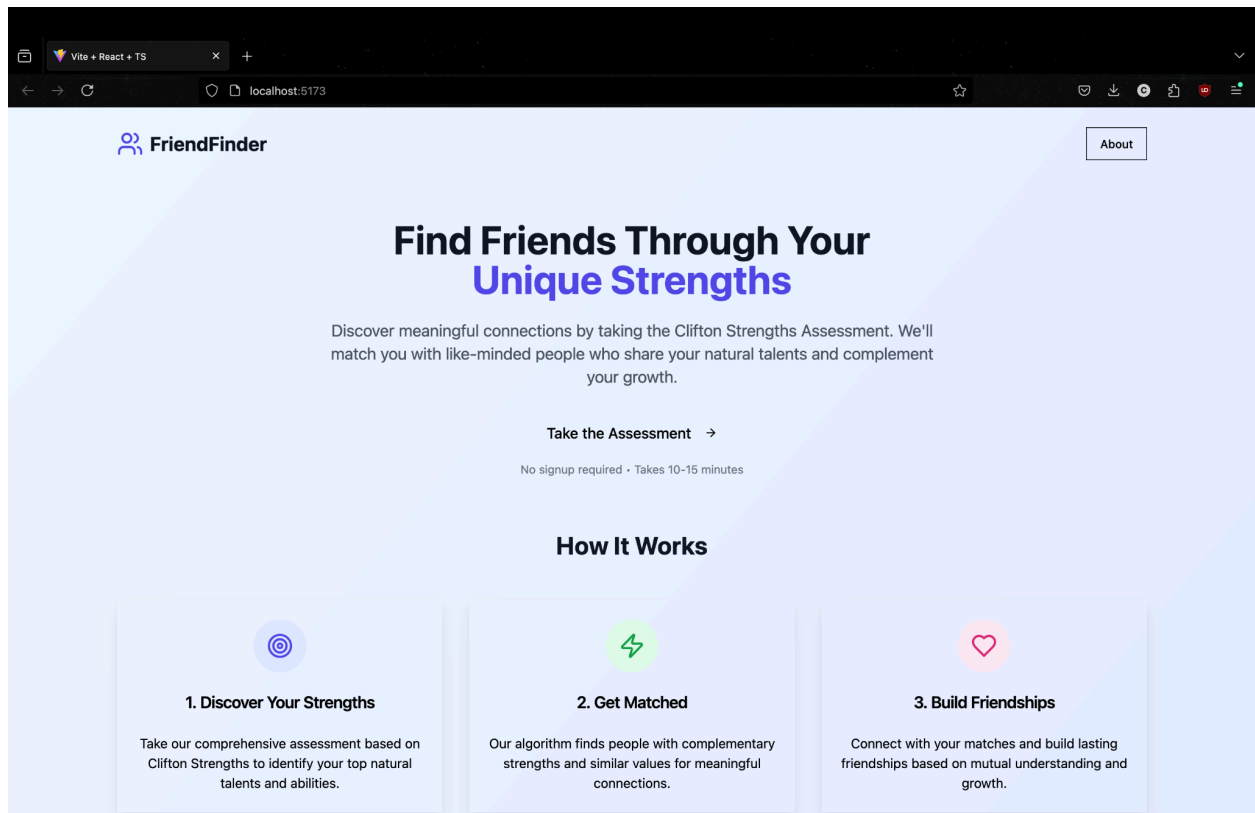
Key Relationships

- Users have one current active personality test result
- Users can have multiple historical test results
- Friend relationships are bidirectional with status tracking
- Users can have multiple roles with different permissions
- All user activities are logged for security auditing
- Posts belong to users with configurable visibility settings

4. Basic Wireframes / UI Mockups

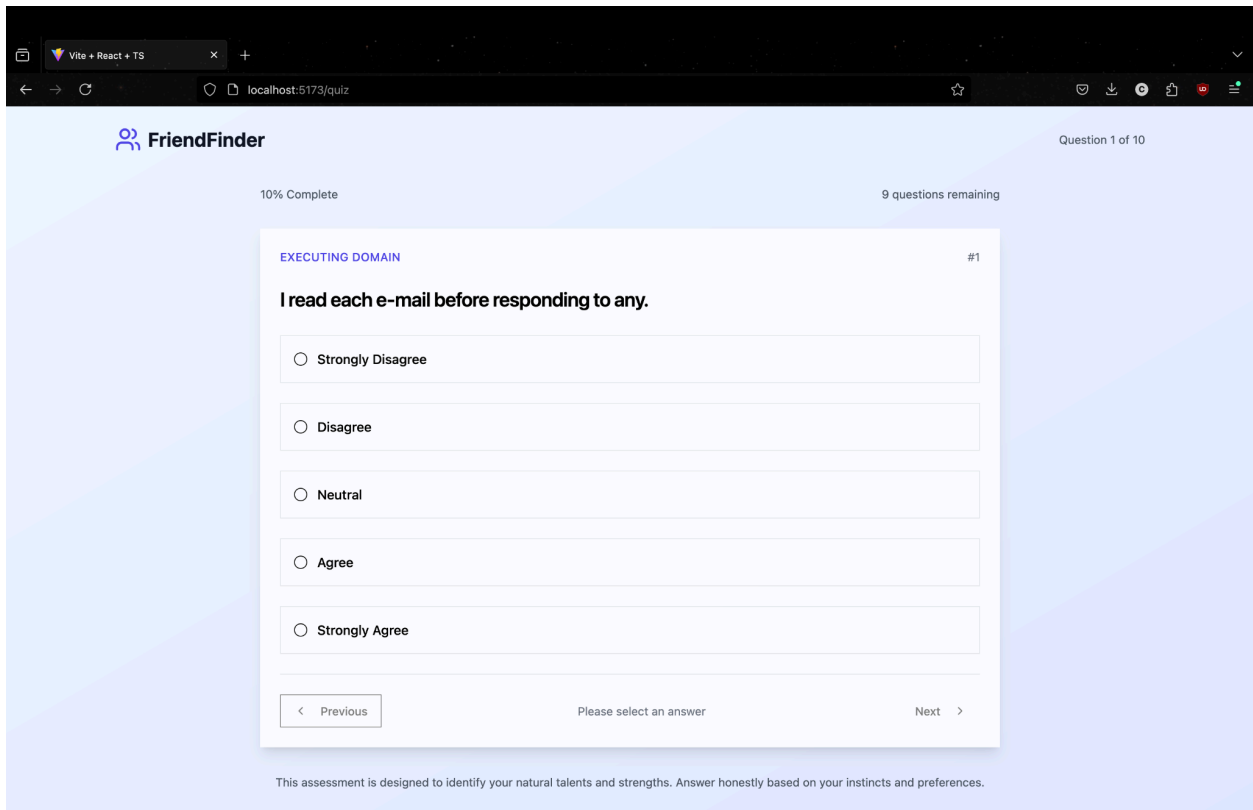
Home/Landing Page

Serves as an “advertisement” for the service. Provides multiple links to take the assessment/quiz. Assures the visitors that the quiz is low commitment.



Quiz/Assessment Page

Allows all visitors to take the quiz without signing up. After completing the quiz, visitors will be allowed to share results and be prompted to create an account if they would like to continue with the friend finder service.



The screenshot shows a web browser window with the address bar displaying 'localhost:5173/quiz'. The page features the 'FriendFinder' logo in the top left and 'Question 1 of 10' in the top right. A progress indicator shows '10% Complete' and '9 questions remaining'. The main content area contains a question titled 'EXECUTING DOMAIN' with the instruction 'I read each e-mail before responding to any.' Below the instruction are five radio button options: 'Strongly Disagree', 'Disagree', 'Neutral', 'Agree', and 'Strongly Agree'. At the bottom of the question card, there is a 'Previous' button, a prompt 'Please select an answer', and a 'Next' button. A footer note states: 'This assessment is designed to identify your natural talents and strengths. Answer honestly based on your instincts and preferences.'

FriendFinder

Question 1 of 10

10% Complete 9 questions remaining

EXECUTING DOMAIN #1

I read each e-mail before responding to any.

☐ Strongly Disagree

☐ Disagree

☐ Neutral

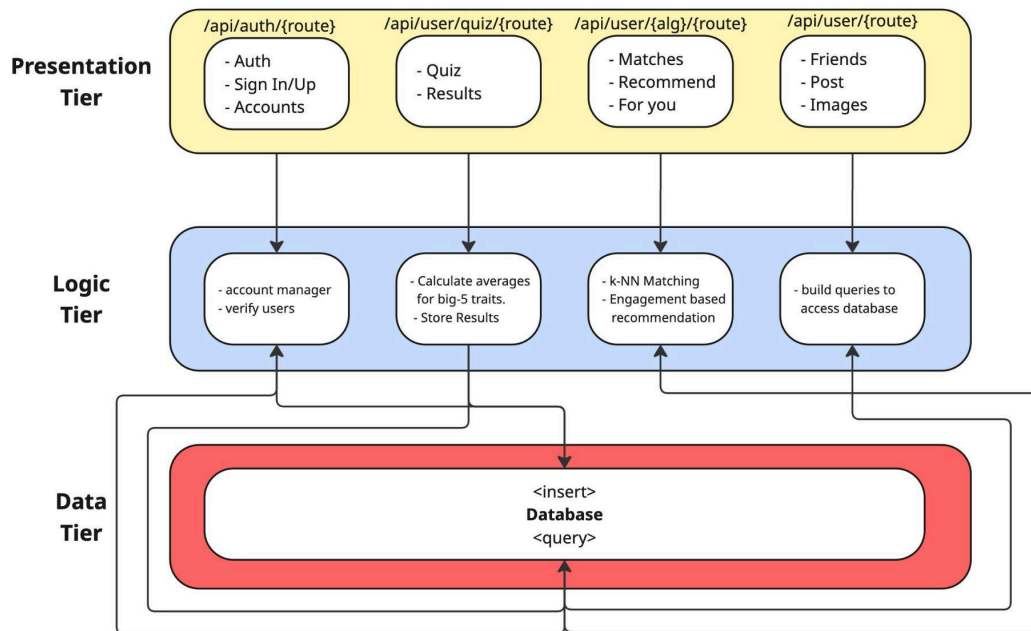
☐ Agree

☐ Strongly Agree

< Previous Please select an answer Next >

This assessment is designed to identify your natural talents and strengths. Answer honestly based on your instincts and preferences.

5. 3-Tier Architecture Diagram



Presentation Tier

Technologies: React, Javascript

Implementation: Using react router, the app will function as a traditional website with minimal highly dynamic pages. Visitors will have access to the static Home, Quiz, About, etc... pages which do not interact with the logic tier. Users will have access to the Profile, Recommended, Matches, and Friends pages which all interact with the logic tier.

Logic Tier

Technologies: Python, FastAPI, Numpy, Pandas, Scikit

Implementation: Backend server with router groups for Auth and User. The User group will have sub-groups for Stats, Quiz Results, and Matches/Recommendations. The auth groups will have read and write access to the database. The user sub-groups will have varying read and write access depending on the sub-groups target functionality.

Data Tier

Technologies: PostgreSQL

Implementation: The data tier will be a PostgreSQL database managed by a service in the project root outside the scope of the backend and frontend server.