

Project Proposal for Team 2

PROJECT NAME:

TheraBot, an Interactive Web Application

OBJECTIVE:

Develop an interactive website therapist-client application in which the user acts as a therapist and engages with an AI-driven client (bot). The goal is to simulate diagnostic dialogue and enhance understanding of mental health assessment processes. This project will provide psychology students and trainees with a safe, controlled environment to practice diagnostic reasoning and interviewing skills before applying them into real-world contexts.

BACKGROUND:

Current training tools for mental health diagnostics are limited. Most chatbot systems assume the human is the client seeking support, not the therapist practicing diagnostic skills. TheraBot addresses this gap. Current chatbot systems are often client-facing, where a user seeks help. Our project inverts this perspective: the website will position the human user as the therapist, and the bot will simulate a patient presenting symptoms or problems. By doing so, the application allows practice in diagnostic reasoning, empathy, and structured questioning. The website format was chosen for accessibility, cross-platform compatibility, and ease of classroom presentation. While existing chatbots support clients, few if any tools exist for therapists-in-training to practice their diagnostic techniques interactively. This gap limits experiential learning in mental health education, which TheraBot directly addresses.

PRODUCT ABSTRACT

TheraBot is a web-based application that simulates therapist-client interactions. Users log in as a therapist and engage with a conversational AI bot portraying a client. The bot will present a mix of realistic scenarios (stress, anxiety, academic issues, etc.), and the therapist-user must ask questions, interpret responses, and attempt a preliminary diagnosis. In the future, TheraBot may

serve as a foundation for integrating more advanced NLP techniques, making it scalable for research and clinical training modules.

Proposed System & Functionality

1. User Interface:

- A clean, web-based interface accessible on desktop and mobile browsers.
- Designed with simplicity for classroom use while supporting real-time interaction.
- Accessibility is considered (color contrasts, screen-reader friendly, and more).

2. Therapist Tools:

- Diagnostic categories, structured note-taking, and reasoning panel.
- Allows therapists to log their thought process throughout the session.

3. Client Simulation

- AI-driven patient responses, blending scripted dialogue with adaptive prompts.
- Patient responses are designed to ensure simulation of realistic hesitation, emotion, or ambiguity, reinforcing the user's need for careful questioning.
- Supports branching dialogue to create more naturalistic sessions.

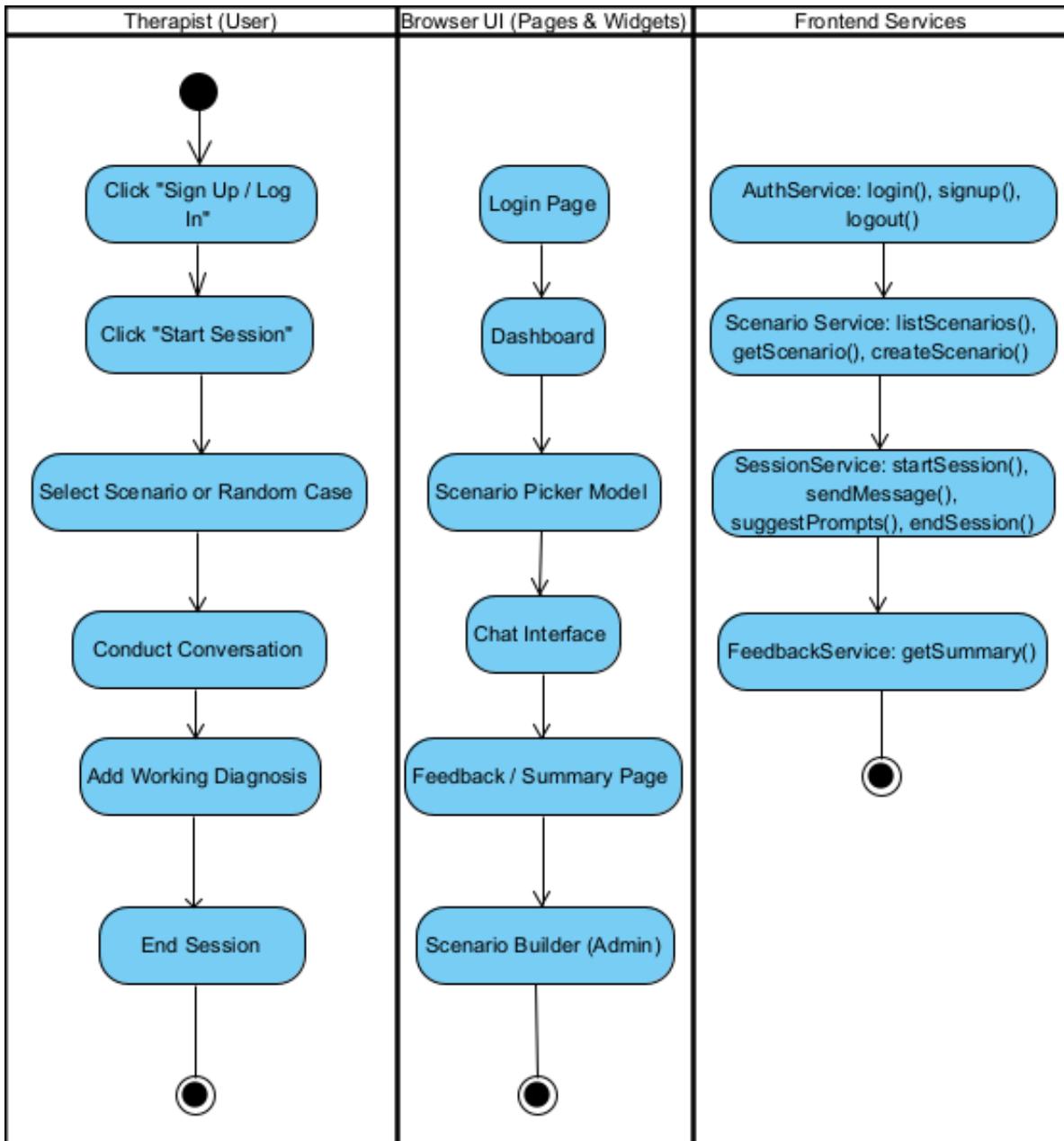
4. Scenario Creation & Library

- **Scenario Builder:** Admins can design new cases or import them via CSV.
- **Library:** Built-in scenarios (stress, anxiety, ADHD, etc.) for immediate use.

5. Feedback & Learning Support:

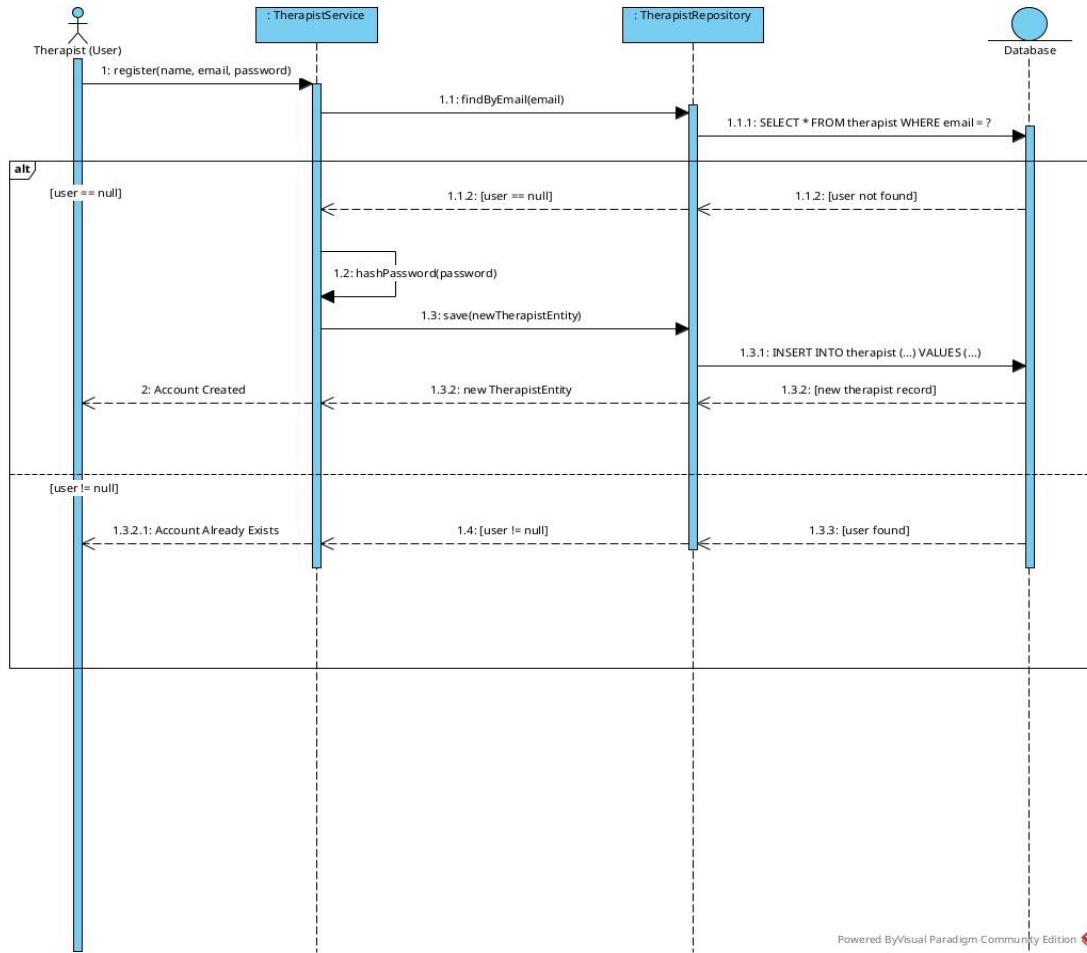
- After each session, the system will provide the following:
 1. Summary notes
 2. Suggested diagnostic considerations
 3. Option for instructors to add custom feedback

Frontend Model



This diagram illustrates the frontend functionality of TheraBot, showing how therapist-user actions map to specific UI screens and corresponding frontend services. It emphasizes the separation of concerns between user interaction, interface design, and service logic.

Authentication



Allows the user to create a secure account using our system that hashes their password with the SHA-256 hash function. Prior to account creation or the hashing of their password, a check is done to ensure that they don't already have an account in our system.

KEY FUNCTIONALITIES:

- SHA-256 password hashing
- Administrator and employee level accounts
- Account therapy session chat history

Although TheraBot is an educational tool, user authentication ensures privacy and data integrity. Session histories are stored securely for learning purposes, not clinical use.

Technical Stack

- **Frontend:** HTML, Tailwind CSS, JavaScript/TypeScript
- **Backend:** Node.js or Python Flask/Django
- **Database:** MySQL or Firebase for storing sessions and scenarios
- **Bot Engine:** Rule-based scripts + optional NLP integration

SOFTWARE & HARDWARE REQUIREMENTS

- Any modern web browser (Chrome, Edge, Brave, Safari, etc.)
- Server hosting environment (Heroku, AWS, local deployment)
- Open-source chatbot libraries or custom scripts

TEAM COMMUNICATION & DEVELOPMENT PROCESS

- Weekly progress meetings through Discord
- Project management utilizing Jira & Trello
- GitHub repo updates

DEVELOPMENT TEAM

Members

The development team for this project is composed of 5 students in the mixed course for CSCI-C308 (SYSTEM ANALYSIS AND DESIGN) and INFO-I 450 (DESIGN & DEV OF AN INFO SYSTEM) course offered in the Fall Term 2025 at IUSB. The main areas of work for this project were identified. Each member chooses a task for which he is assumed full responsibility including the drafting of a section about these topics in the related documentation for this project. However, the final product and documentation will be the result of group effort to edit, revise and perfect these drafts. At the same time, interfacing issues will require a mutual agreement coupled with joint development and testing.

- Andrew
 - Lead developer and project coordinator. Responsible for overseeing overall functionality and integration of the website. Focused on ensuring that the UI and backend work smoothly together while guiding the team's progress.
- Sammy
 - One of the UI developers, and the UX designer. Will be working on the looks, and functionality of the website. Focusing on the interconnectedness with the website and the database.
- Camryn
 - One of the backend developers and the main backend diagram maker. Will be working on the database, backend administrator UI, and the main backend system.
- Lorrin
 - One of the UI developers. Contributes to front-end design and implementation, ensuring that the interface is user-friendly, responsive, and visually consistent.
- Spencer
 - Focus on frontend-backend integration, as well as contributing to database design and queries.

PROJECT SCHEDULE

Week 1-2: Envisioning	Proposal finalization, tech stack selection
Week 3-4: Planning	Wireframes, database schema, UML diagrams
Week 5-6: Building	Frontend and backend core functionality
Week 7: Testing	Integration testing, bug fixes
Week 8: Deployment	Final polish, slides, and demo