Program Specification Documents: PensieveAI

General Description

PensieveAI is a web-based platform designed to streamline the analysis of qualitative textual data. By leveraging OpenAI's GPT-4o mini API, the platform automates identifying and categorizing themes within text sources such as interview transcripts, focus group discussion transcripts, open-ended survey responses, customer reviews, and other text-based data. This tool aims to provide researchers with a preliminary thematic analysis that includes descriptions of each identified theme, significantly reducing the time and effort traditionally required for manual qualitative data coding. The platform will accept input in PDF, DOCX, and TXT formats and deliver results in a user-friendly text format.

This platform is tailored for:

- Social Science Researchers: Academics, sociologists, anthropologists, and others who frequently conduct qualitative research.
- UX Researchers: Professionals involved in user experience research who need to analyze interview and focus group transcripts.
- Market Researchers: Individuals and teams analyzing consumer feedback, focus groups, or other qualitative market research data.
- Novice Qualitative Researchers: Students and beginners who are new to qualitative data analysis and need a starting point for their research.
- Businesses: Companies looking to analyze customer feedback, employee interviews, or other qualitative data for insights.

Problem Statement:
Identifying themes from the analysis of qualitative data is a time-consuming and cumbersome task. Researchers manually code transcripts, which take several weeks or months to identify recurring patterns and synthesize themes. This process is highly labor-intensive and liable to human error and bias. Novice researchers find this task daunting due to their lack of experience

in identifying and categorizing themes. There is an obvious need for a tool that, in addition to automating this process for faster insights, would provide a trustworthy starting point for researchers to conduct deeper analyses.

External Mechanisms and Technologies:

- OpenAI GPT-4o mini API: For analysis and theme extraction.
- File Processing Libraries: Libraries like PyPDF2 for PDFs and python-docx for DOCX files.
- Web Framework: The platform will ideally use Flask or Streamlit for a web-based GUI.
- Data Storage: If needed, a database system like SQLite or PostgreSQL will be used to store user data and results.

Note: While the primary interface is web-based for accessibility and ease of use, a minimal Command Line Interface (CLI) version will be implemented in version 1 for the initial development phases. The web front end will be undertaken in version 2.

Task Vignettes

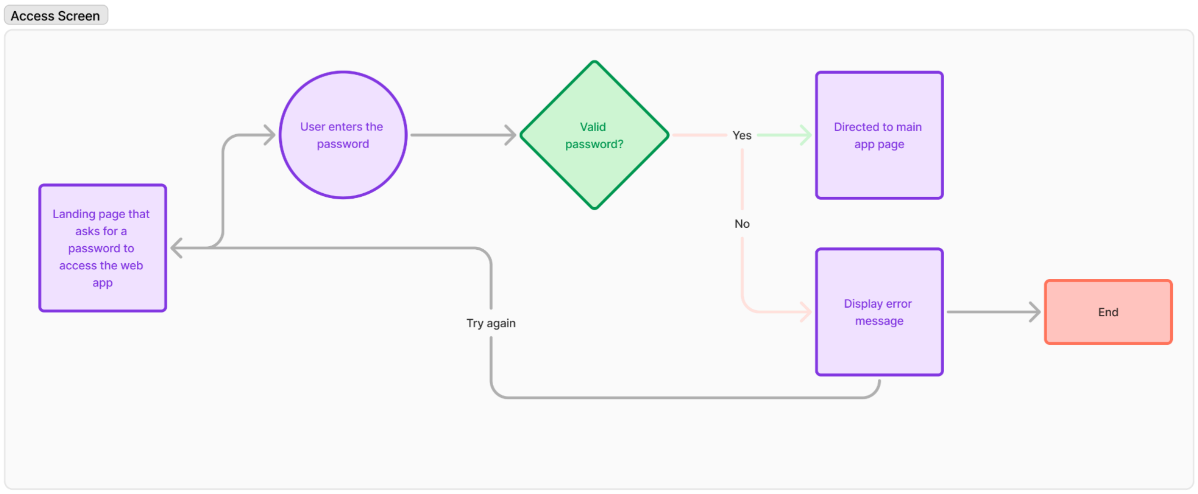
Vignette 1: Password-protected Access Screen

Scenario: Jasmine, a Math Education PhD student, visits the PensieveAI platform to analyze her interview transcripts for the first time.

When she navigates to the platform's URL, she is greeted with a passcode access screen instead of the typical user sign-up or login page. Jasmine enters the secret passcode that Shristi, the creator, shared with her earlier.

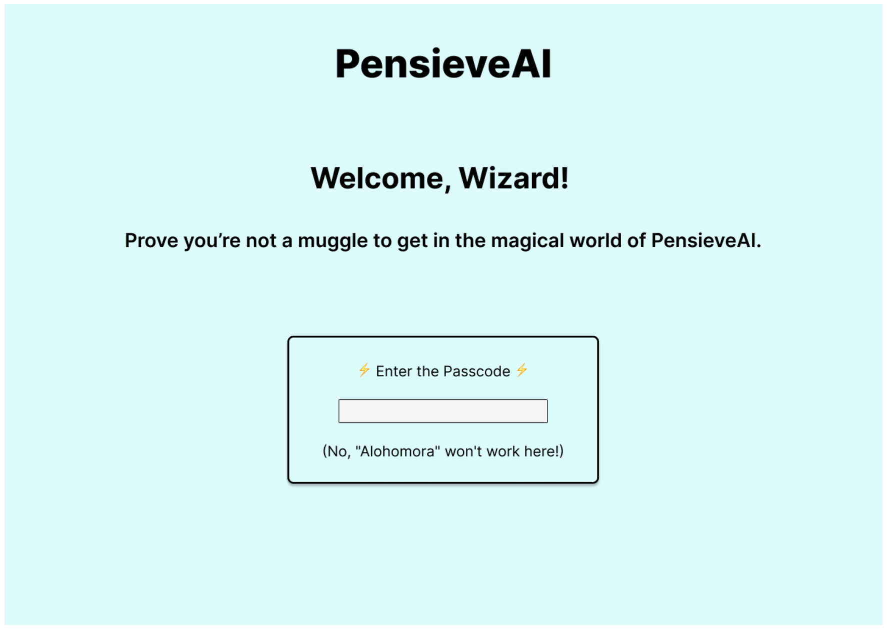
Once the correct passcode is entered, Jasmine is redirected to the main application interface, where she can begin uploading files for analysis. If she had entered an incorrect passcode, the platform would have displayed an error message, prompting her to try again.

Note: This simplified approach ensures that the platform is only accessible to authorized users without the complexity of creating user accounts and without exposing the GPT-4o mini API to spam or unauthorized usage.



Technical Details:

- Single Passcode Authentication: A hardcoded passcode in the backend allows all users access to the platform.
- Session Management: Flask tracks authentication status once the correct passcode is entered.
- HTML Form: A password input form collects the passcode and submits it for verification. If the passcode matches the predefined one, the user is redirected to the dashboard; otherwise, an error message is shown.
- Error Handling: If the passcode is incorrect, an error message is displayed, and the user remains on the login page.


PensieveAI

Welcome, Wizard!

Prove you're not a muggle to get in the magical world of PensieveAI.

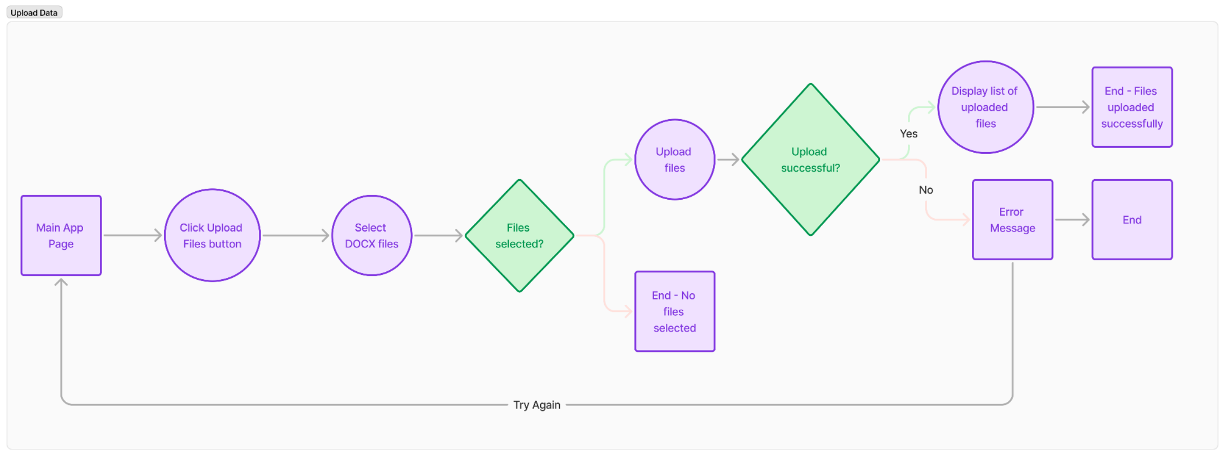
⚡ Enter the Passcode ⚡

(No, "Alohomora" won't work here!)

Vignette 2: Uploading Data

Scenario: Jasmine wants to upload her interview transcripts after logging in.

Jasmine clicks the "Upload Files" button on her main screen. She selects multiple DOCX files from her computer and uploads them. The platform displays the list of uploaded files, confirming successful uploads.



Technical Details:

- Multiple file upload capability supporting PDF, DOCX and TXT formats.
- Backend file handling. *[Note: Limited knowledge.]*
- Storage of uploaded files on the server. *[Note: Limited knowledge.]*

Vignette 3: Entering additional instructions (Revision)

Scenario: Jasmine needs to provide additional instructions to customize the analysis.

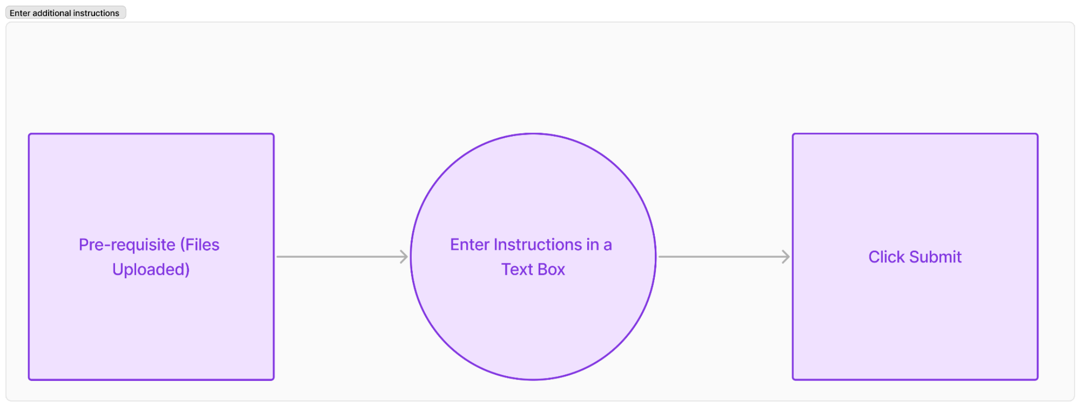
She is presented with a single input option: a text box where she can enter any instructions or details for the analysis. This might include predefined codes, thematic considerations, specific instructions on handling the data, or any other relevant notes she wants the system to follow.

Jasmine types her instructions into the text box. She details the themes she wants to focus on and includes additional notes on handling ambiguous data points.

Technical Details:

- A single text input box allows users to enter relevant instructions, including codes, themes, or analysis preferences.

- The system uses these instructions to guide the analysis process.
- This text input is appended to the prompt in the backend.
- Note: This input is optional.

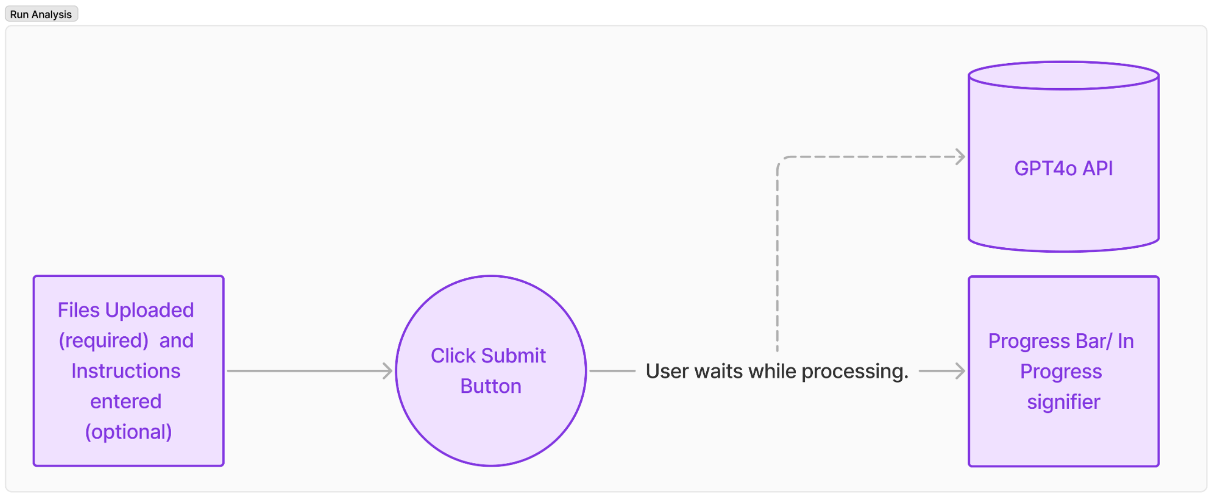

Enter additional instructions
Pre-requisite (Files Uploaded)
Enter Instructions in a Text Box
Click Submit

Vignette 4: Running the Analysis (Minor revision)

Scenario: Jasmine initiates the thematic analysis.

Satisfied with her uploaded files and instructions, she clicks the "Submit" button.

The platform shows a progress bar as it processes the data, sends it to the GPT-4o mini API, and waits for the response.

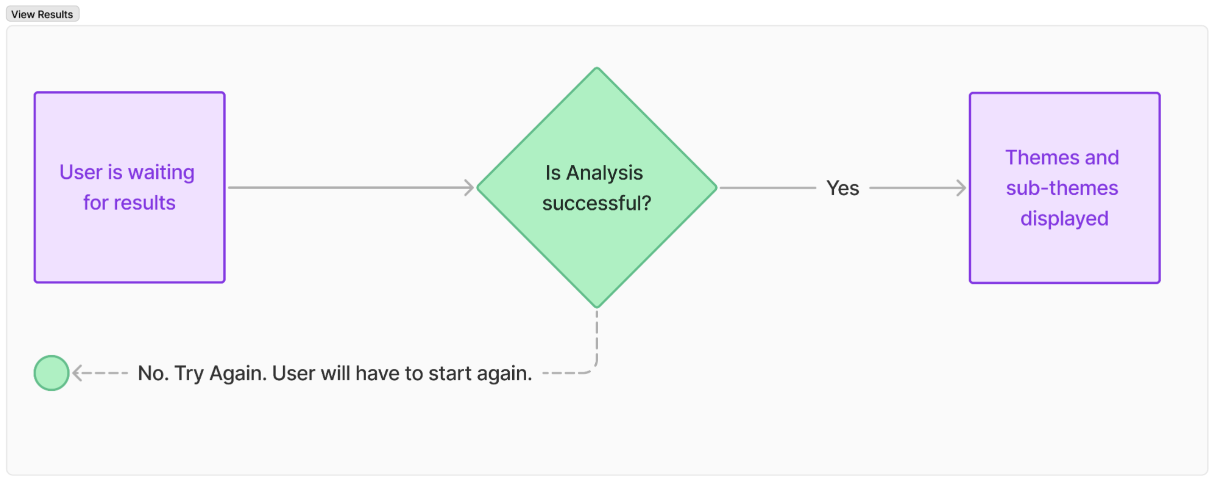

Run Analysis
Files Uploaded (required) and Instructions entered (optional)
Click Submit Button
User waits while processing.
GPT4o API
Progress Bar/ In Progress signifier

Technical Details:
- Backend integration with OpenAI's GPT-4o mini API.
- Error handling for API call failures or timeouts. *[Note: Limited knowledge.]*

Vignette 5: Viewing Results

Scenario: Jasmine reviews the identified themes and sub-themes.

Once the analysis is complete, the results page displays themes and sub-themes with descriptions. Jasmine can view associated text excerpts from her transcripts for each theme and sub-theme.
If the analysis is unsuccessful, there will be an error message, and the user will ask to try again. *[Note: Need to test whether getting the excerpts will be possible.]*


View Results
User is waiting for results
Is Analysis successful?
Yes
Themes and sub-themes displayed
No. Try Again. User will have to start again.

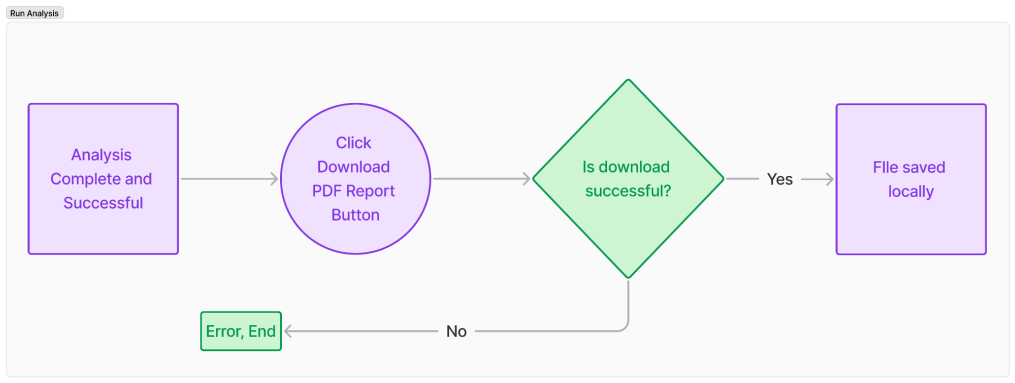
Technical Details:

- Dynamic rendering of results using HTML/CSS and possibly JavaScript for interactivity. *[Note: Limited knowledge of Js, so might use simple text box output format to make it less complicated.]*
- Data parsing from the GPT-4o mini API response into a readable format.

Vignette 6: Downloading the Report

Scenario: Jasmine wants a copy of the analysis for future reference

She clicks on the "Download Report" button. The platform generates a PDF report containing all the themes, descriptions, and relevant excerpts. She saves the report to her computer.



Technical Details:

- PDF generation using a library like ReportLab or WeasyPrint. *[Note: Limited knowledge.]*
- Providing the file as a downloadable attachment via the web interface. *[Note: Limited knowledge.]*

Technical Flow

The platform operates through interconnected modules that handle user interaction, data processing, and communication with external services.

Modules and Data Flow:

1. User Interface (UI):
   - Input: User credentials, file uploads, analysis options.
   - Output: Confirmation messages, analysis results, downloadable reports.
   - Technologies: Flask/Streamlit templates, HTML/CSS, JavaScript.
2. Authentication Module:
   - Handles: Password access, session management.
   - Data: User information stored securely in a database.
   - Technologies: Flask-Login
3. File Handling Module:
   - Processes: Uploaded PDF and DOCX files.
   - Converts: Files to plain text using PyPDF2 or python-docx.
   - Data: Raw text extracted from files.
4. Analysis Module:
   - Translates: Inputs into prompts for GPT-4o mini API based on analysis type.
   - Sends: Text and prompts to the API.

    - Receives: Themes and descriptions from the API.
    - Data: Themes and sub-themes.
5. Results Module:
    - Formats: API response for display and download.
    - Stores: Results in the database.
    - Data: User-specific analysis results.
6. Report Generation Module:
    - Creates: PDF reports from the analysis results.
    - Data: Final report file for download.
7. Database:
    - Stores: User data, uploaded files metadata, analysis results.
    - Technologies: SQLite/PostgreSQL with SQLAlchemy.
8. Error Handling:
    - Captures: Exceptions and errors throughout the application.
    - Notifies: Users of any problems encountered.

Data Types and Structures:

- User Data: Dictionaries containing user credentials and settings.
- Text Data: Strings of raw and processed text.
- Analysis Options: Boolean flags and lists of strings for codes and words to remove.
- API Prompts: Structured strings formatted per OpenAI's requirements.
- API Responses: JSON objects with themes and descriptions.
- Results Data: Lists and dictionaries organizing themes, sub-themes, and excerpts.
- Report Files: PDF documents generated from results.

User Interaction Points:

- Inputs:
    - Text fields for entering password.
    - File upload controls.
    - Optional tet box for additional instructions
- Outputs:
    - On-screen messages and alerts.
    - Displayed analysis results.
    - Download prompts for PDF reports.

Final (Self) Assessment

Biggest Change from Sketch: Incorporating detailed user interaction elements like account creation and data preprocessing options added complexity beyond the initial sketch.

Confidence in Implementation: I am fairly confident in basic web interface elements and prompting. I have a basic understanding of technical concepts and programming skills required for the project. So, I feel that I can use resources, guides, and forums for external support.

Biggest Potential Problem:
Managing the multiple file upload, API calls, and handling rate limits associated with the GPT-4o mni API is a concern. Also, PDF generation and report download is concerning due to my lack of experience. With extremely limited web application experience, I expect many oversights, confusion, and mistakes. This might lead to a loss of time and difficulty adding all the features.

Areas Needing Assistance:
- Multiple file upload
- API connection
- Database
- PDF Report Generation

Editable Google Doc File
Official Project GitHub Repo