Cross-Cultural Connect

Mason Wilson

Edward Berry

Adam Howard

Michael Gallagher

Table of Contents

System Overview

Deliverable:

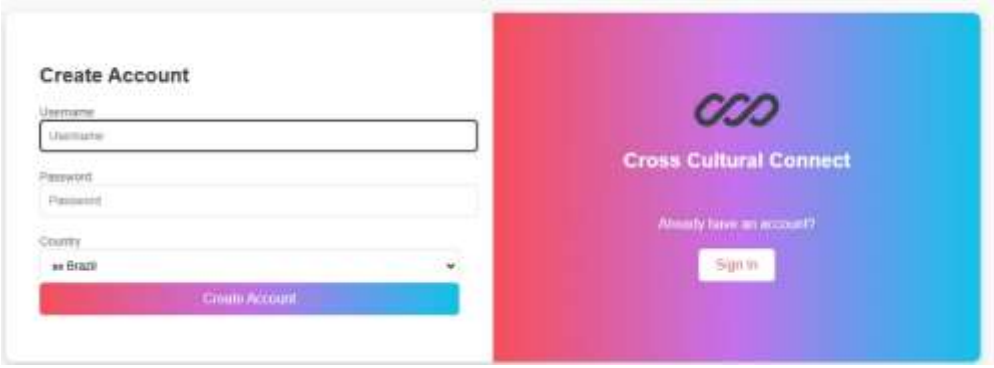
- A user-friendly web site enabling access to culturally relevant services, fostering diversity and inclusion, allowing users to connect to businesses and services in their area.

Description of the Problem:

- Immigrants in Orlando currently do not have a means to connect to businesses and services in their area, as well as other immigrants from their country of origin.
    - Access helpful resources and connect to other users via the message board incorporated in the website.
    - The app, however, will not allow previous user search history or other user input to be available to anyone.

Content to Include:

- **Title:** Cross-Cultural Connect
- **Team Members:** Mason Wilson, Edward Berry, Adam Howard, Michael Gallagher
- **Key Problem:** Immigrants lack accessible platforms to connect with businesses, services, and local communities from their home country in their new location.
- **Goal:** Build a secure, reliable website for connecting immigrants to businesses, resources, and local support networks.



System Highlights

Deliverable:

- Working Prototype / Proof of Concept demonstrating key features and requirements.

Content to Include:

- **Key Features:**
    - **Search Functionality:** Find businesses/services by zip code and country of origin.
    - **Community Board:** Message-based interaction.
    - **Localization:** Translator integration for multilingual support.
    - **Privacy-First Design:** No user location tracking; encrypted credentials.
- **Feedback Integration:**
    - Emphasize how the app incorporates "cultural compassion" (e.g., using user feedback loops to refine search results and content). (TBD)

System Requirements

Requirements Analysis

Deliverable:

- **Functional Requirements:**
    - Maintain a list of relevant cultural businesses and services in the database.
    - Allow business searches by origin country and zip code.
        - Results should show in a specified radius
    - Ensure secure login with usernames and passwords.
        - User information retained will be username, country of origin, and password
        - Usernames must be unique, and passwords must be of a minimum strength (Must be at least 8 characters, include at least 1 non-alphanumeric character)
        - Usernames should be encouraged not use identifiable information, for example, your full name.
        - Passwords should be encouraged not to include the username, or easily identifiable information.
        - Passwords should be stored in the database for verification in encrypted form
    - A community board feature will allow for reviews and posts. This will allow a community connection and allow users to post reviews and experiences.
        - Enable messaging with lifespan options (24-96 hours or indefinite).
        - Allow owning users to delete community board messages
        - Deleted messages data should be removed from the database
    - Inside the web application, users should have access to a translator that works with multiple supported languages.
- **Non-Functional Requirements:**
    - Intuitive UI and accessible, user-friendly navigation.
    - Strict privacy measures for user information and passwords.
    - Ensure cross-platform compatibility.
    - The software should be easily maintainable, and scalable, to allow for the integration of multiple countries in the future.
    - Developers should have no means possible to decrypt a user's password or data.

Additional Focus:

- Respond to feedback:
    - Define the types of businesses and services clearly.
    - Emphasize "how" the app will foster cultural diversity and compassion.

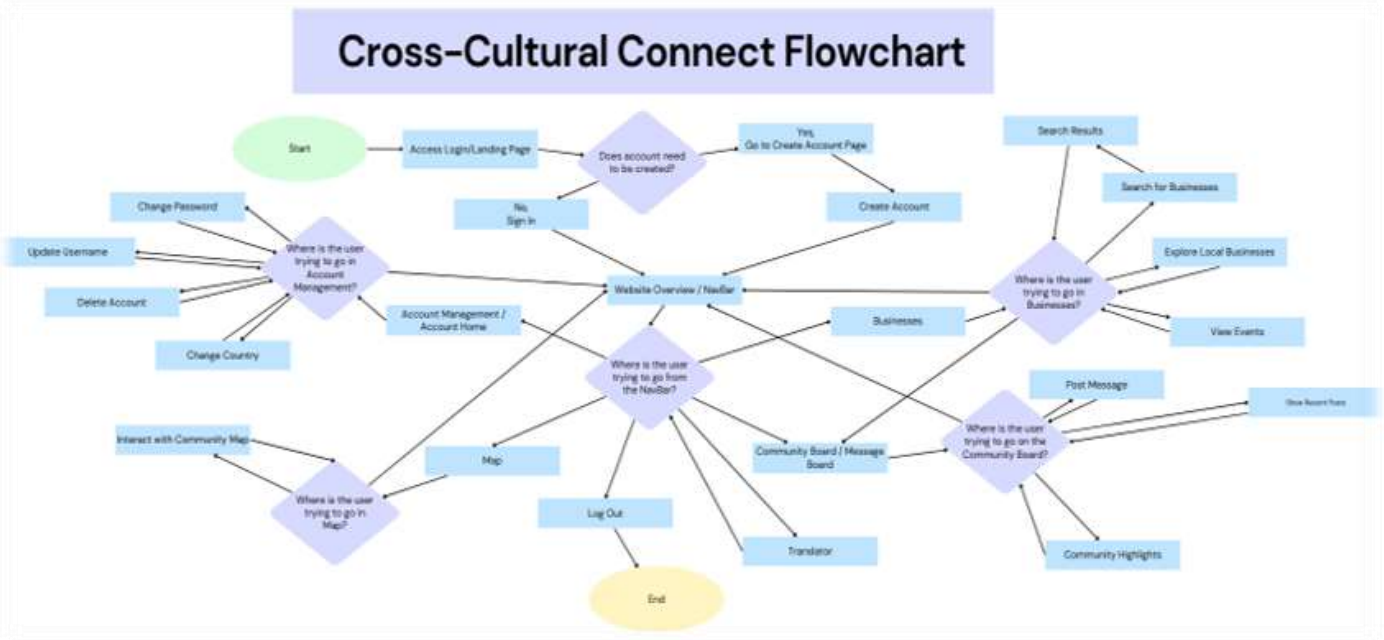
Development Process

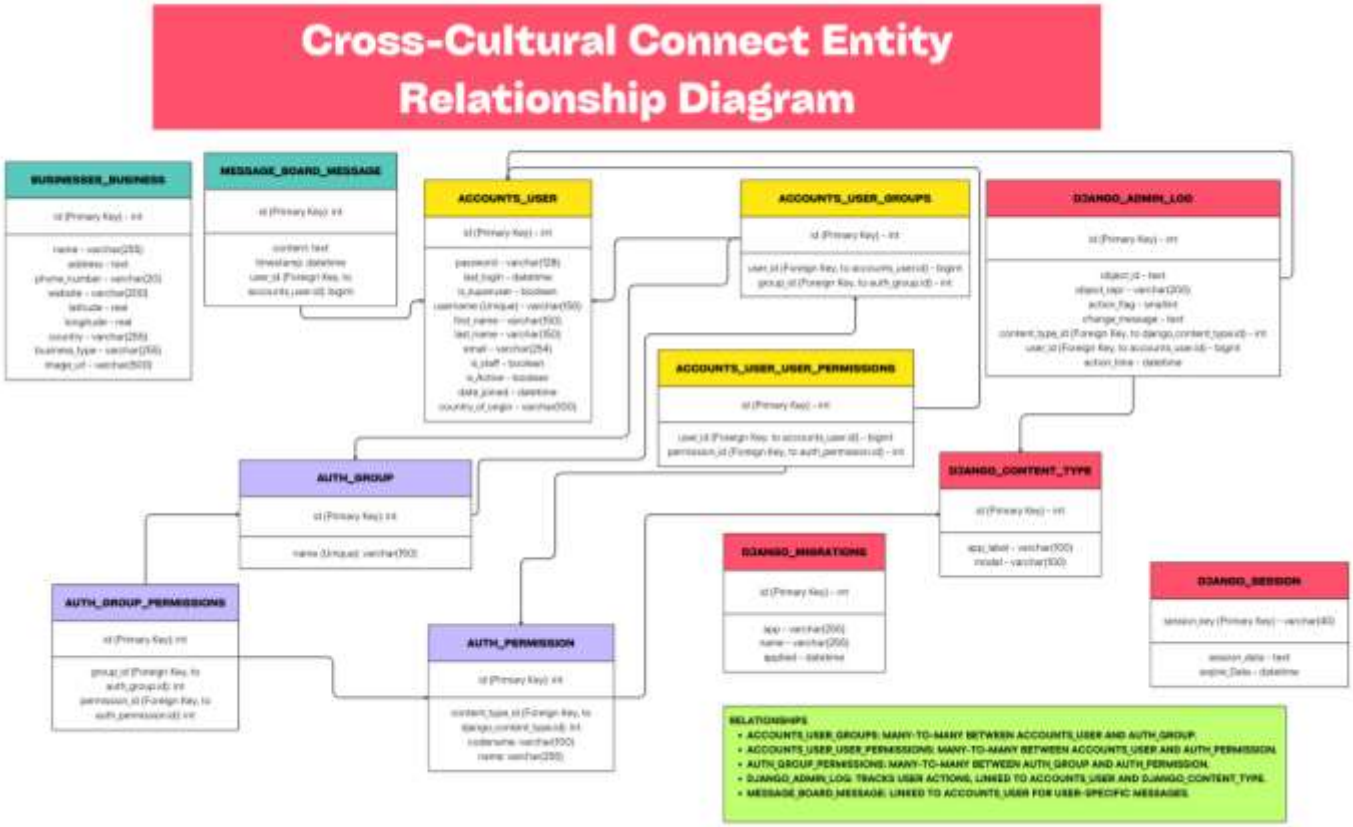
Planning

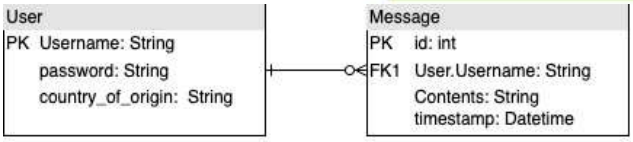
- Use a timeline visual showing phases:
    1. Problem identification and brainstorming.
    2. Functional and non-functional requirements gathering.
    3. Design and entity relationship models.
        - Use case model, Entity Relationship model, Flowchart model
    4. Prototyping and UI design.
    5. Development and testing.
    6. Deployment and feedback collection.

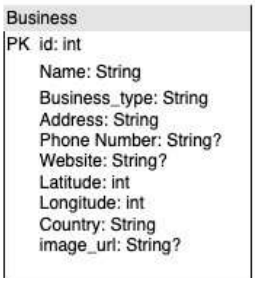
Design

- **Design Models:**
    - Provide user and design models
    - Illustrate a database model showing user profiles, businesses, and reviews.
    - Highlight privacy components for secure data handling.
- **Deliverable:**
    - Provide a prototype
    - Design models (Flowchart, Entity Relationship, Use Case)

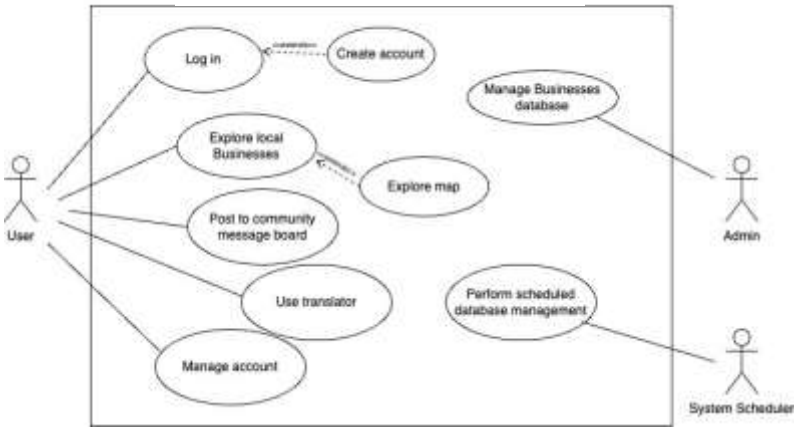








Cross-Cultural Connect Use Case Diagram



- **Integration:**
    - Website will be developed in Python using Django plugin for web application support, on the back end will be a database using SQLite.
- **Working Prototype:**
    - Home screen with search and categories.
    - Ability to create a new user, and to login.
    - Ability to navigate to searching businesses and services and to the community board.
    - Ability to post messages to community board, and interface with existing messages.
    - Ability to delete your own messages
    - Ability for admin to delete a user's account and posts as a forum administrator.
    - Business/service profile pages with community reviews.
- **Feedback Integration:**
    - Testing and feedback should be placed in the BugsAndIssues.docx document.

Coding

- The application will be developed using Python and Django plug-in. This decision is based on the experience of our team, and the standard format of Django website structure, for speed of development, and testing.
- Setting up the development environments (https://github.com/wedberry/cross_cultural_connect/blob/main/Developer%20workflow.txt)
- Download the Cross-Cultural Connect Django framework from this repository (https://github.com/wedberry/cross_cultural_connect.git)
- Team will implement features as based on system overview and system requirements, each working feature will be checked into Git, with comments and what has changed, and notify other team members of the update.
- Team members should keep each other informed of the features they are working on or testing, bugs should be listed in a separate BugsAndIssues.docx Word file in root, and updated accordingly, with comments and dates (note that change tracking is enabled in the document).

Testing

- All team members are responsible for testing once a prototype is available, and continue testing when updates are made, updating the BugsAndIssues.txt file, with new issues and removing fixed issues from the document, which allows us to keep live track of bugs found and bugs fixed.
- Testing should follow the activity diagram to ensure all paths are met.
    - Testing should include edge-case scenarios for user creation and password management.
    - Testing should also exercise admin flow and function.
    - Testing on the translator should ensure valid translations.
    - Testing should verify that business information is correct, and the search results are accurate.
    - Testing on the community board should ensure that other users can see posts.
    - Testing a deletion of an owner's post on the community board should ensure that other users can no longer see it.
    - Testing the navigation bar on every page should ensure the correct landing page.
    - Testing the map function works correctly, by verifying against another well-known map utility.
    - Testing user facilities and account management functions.
- All the above test cases should follow the website flowchart diagram, to ensure that all functionality has been achieved and functions as intended. Testing should follow that the system requirements have been met.

Feedback

- Highlight comments, such as scope or implementation and integration. These can also be added to the bugs and issues document.

Evolution

- The Cross-Cultural Connect software was developed in a group, as part of a collaborative project, and there is no intention of us to deploy it live, and therefore, there will be no requirements to maintain the software.
- If a group member decides to move forward with a live deployment, then that member will be responsible for the evolution of the software.

- **Platform Choice:**

    - Proof of Concept: The web site will be developed and available through a localhost port on the developers PC via Django web server and demonstrated to the class.

    - Implementation: After proof of concept, the site could be hosted in AWS, live with a front-end web server and backend SQLite server. The web servers can scale up on high demand via a load balancer which monitors traffic and CPU usage.

Lessons Learned

- **Challenges:**
    - The scope to provide users with relevant businesses and services in the Orlando area.
    - Addressing security concerns (e.g., handling user data responsibly).
    - Learning the skills needed to develop web applications in Django for creating robust backend systems, managing a database, server-side logic, and debugging practices, for example, loggers and JS console.
- **What Worked Well:**
    - Effective teamwork, problem-solving, collaboration, and feedback.
    - Focused effort on user-centered design and testing.
    - We are most proud that we were able to deliver some of the major functionalities we had hoped to achieve, such as:
        - The Map
        - The message board
        - The translator
    - These were all challenges in their own way, and delivering a robust working product in each of these areas was very rewarding.
- **Improvements:**
    - Future iterations could expand to include more features, like direct business booking or integration with local government services.
    - Integration with businesses in a wider area, not specifically Orlando, but could include Florida or even multiple states, and a wider range of zip code searches.

Demo

- **Deliverable:**
    - A prototype was demonstrated in class, on a laptop, and could be further developed based on improvements listed, before deployment would be considered.
- **Demo Flow:**
        - Was demonstrated with site navigation, as per the system flowchart, and included:
            - Introducing the app's purpose and the problem it solves.
            - Showcasing key functionalities:
                - Creating a new user and logging in.
                - Searching for businesses and services.
                - Posting and interacting on the community board.
                - Secure logging in and data protection features.
        - Highlight any feedback-driven changes, such as lifespan options for messages or privacy improvements.

- **Tools for Demo:**
        - Demo was conducted using a laptop