

SoftDev POI: ArRESTed Development

TNPG: Gluten Free Pizza

Roster: Michelle Zhu, Tahmim Hassan, Anastasia Lee, Ivan Gontchar

**TARGET SHIP DATE: 2024-12-16**

# DESIGN DOCUMENT

---

## I. Description

This project is a country guessing game where users must identify a country based on progressively more revealing hints. The website uses a database to store user information, APIs to generate the hints, and Bootstrap to provide a clean frontend view.

### A. Program Components

- Login/registration/logout:
  - Users stay logged in until they log out (Flask session)
  - Users' results are not stored unless they are logged in
  - Account information stored in database
- Game:
  - A country is randomly chosen, and the first hint is automatically given
  - Hints are generated by API calls
  - After each incorrect guess, a more helpful hint is given
  - If all hints are given and the user has not correctly guessed the country, they lose and the answer is revealed
- Leaderboard:
  - Displays players with the lowest average number of guesses per country
- HTML/CSS/FEF:
  - Neatly display all information
  - Provide coherent structure

## B. Component Map

Frontend (HTML/Bootstrap)

|

v

Backend (Flask)

|

+-- SQLite Database

|

- Stores countries, hints, and leaderboard data

|

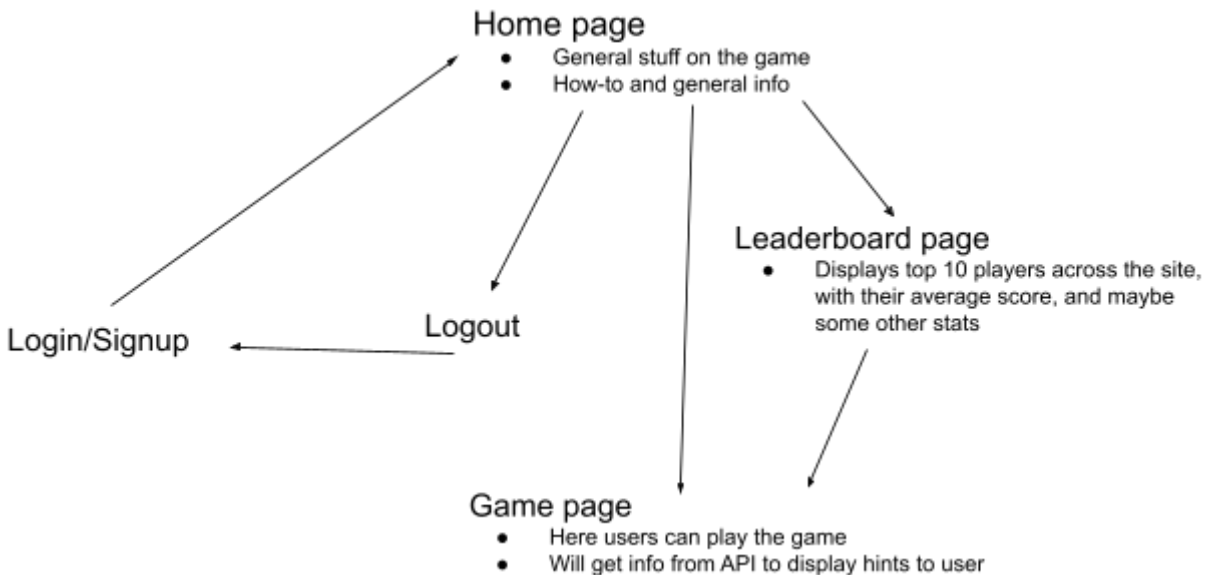
+-- APIs

- OpenWeatherMap (Fetches temperature)

- GeoDBCities (Provides country details)

- FlagsAPI (Fetches country flags)

## C. Site Map



## D. Database Organization

- User table
  - Username (must be unique)
  - Password
  - List of countries remaining
  - Total number of guesses used
  - Average number of guesses per country

## E. API Documentation

- OpenWeatherMap API
  - Endpoint: <https://api.openweathermap.org/data/2.5/weather>
  - Usage: Fetch the current temperature and weather data for the location.
- GeoDBCities API
  - Endpoint: <https://wft-geo-db.p.rapidapi.com/v1/geo/cities>
  - Usage: Provide country details like the capital city, population, and currency.
- FlagsAPI
  - Endpoint: <https://flagsapi.com/>
  - Usage: Retrieve the country flag as a visual hint.

## F. Front-end Framework: Bootstrap

- Top navbar to navigate the site
- Forms for login, registration, and submitting guesses for countries
- List groups to display hints
- Table to display leaderboard

## G. Task Assignments

1. Frontend Development - Anastasia
  - a. Design responsive HTML pages with Bootstrap styling.
  - b. Implement display of game hints and leaderboard.
2. Backend Development - Ivan
  - a. Set up Flask routes for handling game logic and interactions.
  - b. Implement API calls for OpenWeatherMap, GeoDBCities, and FlagsAPI.
3. Database Setup - Michelle
  - a. Use SQLite to design and populate tables for locations and game history.
4. Integration - Tahmim
  - a. Connect frontend with Flask backend.
  - b. Integrate SQLite database with Flask for storing and retrieving data.
5. Testing - Everyone!