Christine Chen, Naomi Kurian, Ethan Cheung, Owen Zeng
WheelOfFortune
SoftDev
P01 – ArRESTed Development
2025-12-02
TARGET SHIP DATE: 2025-12-23

Design Document

PROGRAM COMPONENTS

Flask Application

- Handles routing for all pages
- Manages user session data for login/logout and game states
- Connects to database through helper functions
- Renders HTML pages with Jinja templates

External API

- API calls, parsing, and fallbacks included in each game's Python file
- National Weather Service API used to select a background image based off of the current weather

Front-End Framework (Bootstrap)

- Provides responsive layout and styled components
- Used for navbar, grids, buttons, alerts, and off-canvas components
- Utility classes used for spacing and alignment

Custom CSS and JS

- Used for animated components, overlays, and interactive elements

Pages

Home page

- Only visible for logged in users, if not authenticated, send straight to login page
- *Navbar* will contain a logout, profile, and rewards button
- *Grid of buttons* that lead to different mini games

Login Page

- Asks for username/password and matches it to what is contained in the *user base*
    - Takes the username and compares it to existing *users*

- If username does not exist **OR** if username exists but the password does not match the one attached to the corresponding *User*, output ‘username or password incorrect’
- If username exists AND password matches, sends the user back to the *home page* and adds the username to the session

- Contains a button to create an account (sends to registration page)
    - Once account is created, adds the new *user* to the *user base* and signs them in (sending them to *home page* and adding info to the session)

Register Page

- Username, password, and confirm password form response box.
    - If submit is pressed and it does not pass the comparison check, the page will display a “passwords do not match” message
    - If submit is pressed and the username is the same as another one (case sensitive), the page will display a “username taken” message
    - If submission passes all checks, user will be sent back to the login page and the new user will be added to the user base
- Includes a button (back button) that sends the user back to the login page in navbar

Profile Page

- Displays username, level, XP
- Shows table including all animals that allows for activating a creature
- Creature Archive (Bootstrap offcanvas) which shows all creature species and levels with unowned creatures displayed at lower opacity

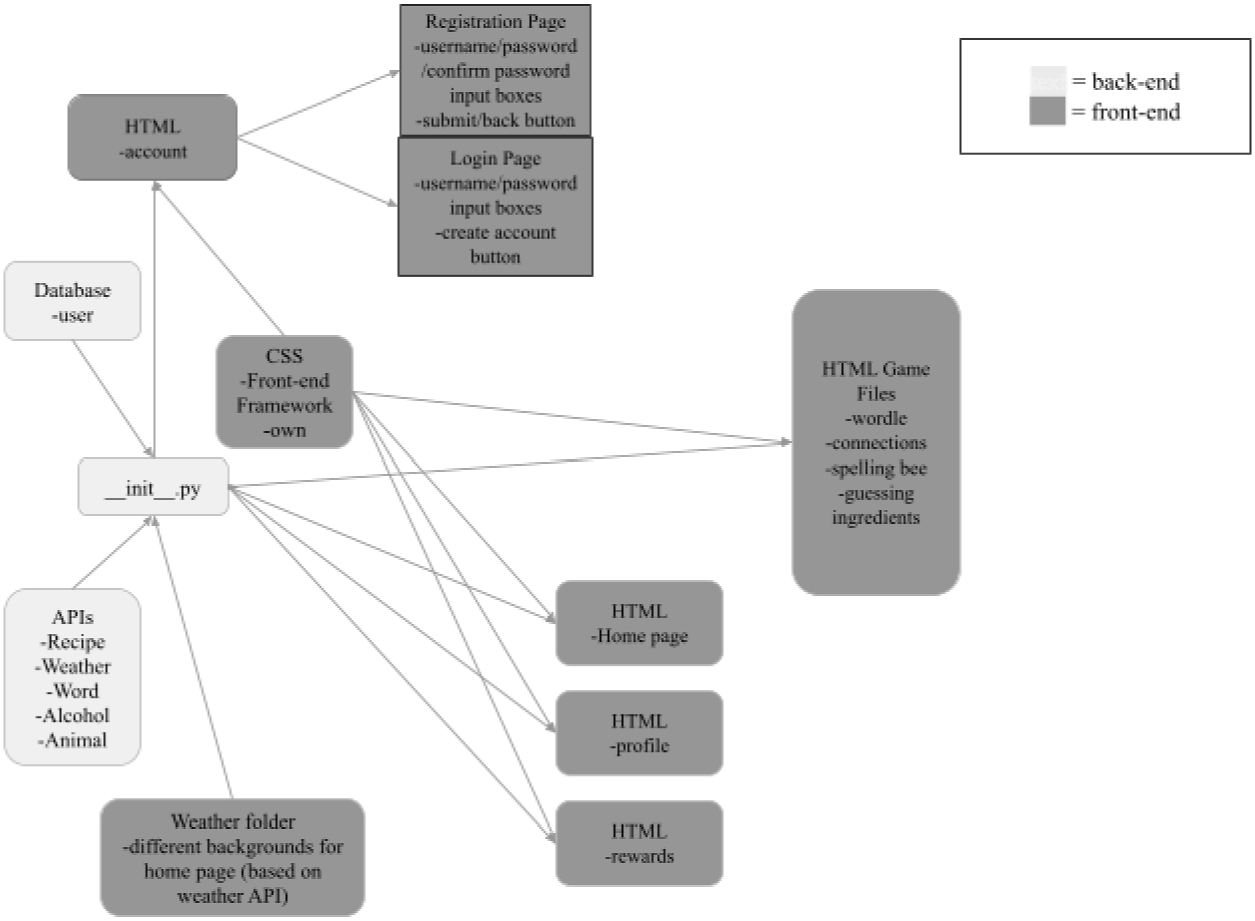
Game Pages

- Different HTML file, python file, and corresponding styling for each mini game
    - Games include: spelling bee, wordle, connections, and ingredients guesser
- Navbar
- Validates user input and awards experience based off result

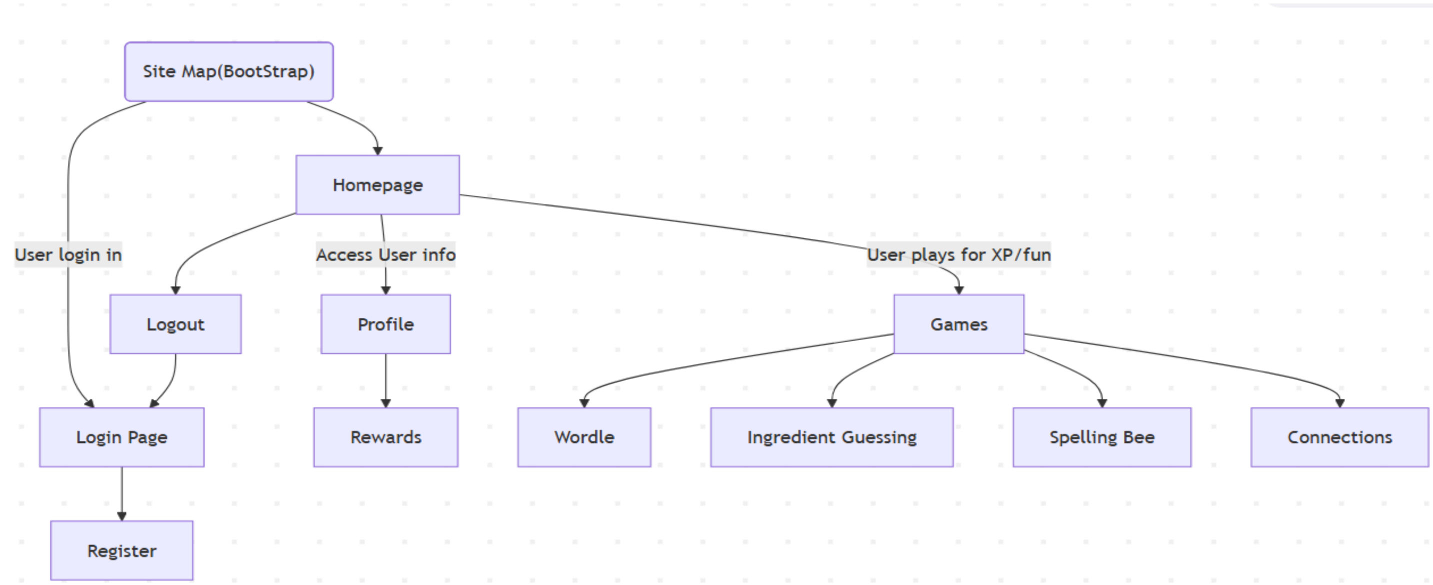
Rewards Page

- Button with pic of egg (new creature)
    - Can only hatch with enough XP (per level) (stored in database)
- Displayed XP; user gains eggs when leveling up

COMPONENT MAP



SITE MAP



DATABASE ORGANIZATION

users

INTEGER	user_id	PK
TEXT UNIQUE	username	
TEXT	password	
INTEGER	xp	
INTEGER	level	

creatures

INTEGER	creature_id	PK
INTEGER	user_id	FK
TEXT	name	
TEXT	rarity	
INTEGER	xp	
INTEGER	level	
TEXT	species	
TEXT	status	

TASK BREAKDOWN:

Task	Member	Deadline	Status
Implement SQLite database schema and database helpers	Christine, Naomi	12/10	Complete
Set up Flask backend: configure app, routes for all pages, handle GET/POST requests, and render templates	Naomi, Christine	12/10	Complete
Implement user registration, login, logout, and sessions	Ethan	12/10	Complete

Wordle Logic	Owen	12/20	Complete
Connections Logic	Christine	12/20	Complete
Spelling Bee Logic	Naomi	12/20	Complete
Ingredients-Guessing Logic	Ethan	12/20	Complete
Dynamic Weather Background	Ethan	12/20	Complete
Implement Reward System	Naomi, Christine	12/20	Complete
Deal with APIs and their outputs, as well as any logic that may come with using them.	All	12/20	Complete
Construct site layout and navigation using Bootstrap	All	12/20	Complete

APIs

- Weather API: used for backgrounds based off of weather
- Recipe API: used for ingredient guessing
- Random word API: used for wordle
- Merriam Webster API: used for spelling bee
- Datamuse API: used for connections

FRONT-END FRAMEWORK - BOOTSTRAP

- Grid system and card components will be implemented to present various mini games
- Grid system and card components will be implemented to present various creatures
- Navbar component for convenient navigation across pages
- Forms and buttons for registration/login or other user input
- Image and utility classes used to style images of creatures
- Alerts for feedback messages such as invalid login or level up notifications

Creature art from: https://www.irasutoya.com/p/nature.html?m=1