

UncommonApp - Dennis Chen, T Fabiha, Addison Huang, Michelle Tang
SoftDev6
Project 2 – The End
2018-01-07

Project Name: WeLoveHue
Framework: BootStrap
APIs to be Used: Color Scheme, IP Identifier, Weather API

Functionality:

Using RESTful APIs we will mimic the game “i love hue.” People will be given a color scheme that broken is into pieces then randomized. Certain pieces will be locked, which will serve as a point of reference for people to complete the puzzle. They will then have to reorganize the pieces until the original color scheme is achieved. If the person likes the puzzle, they can choose to save it, which will require them to create an account. Creating an account will also allow users to create their own puzzles with customization such as the size of the puzzle or its color scheme. Another customization is getting a color scheme for the puzzle based off of a player’s location. The user will then have the ability to make the puzzle public so that other users can play it. As an extension, it can then play the puzzles that other users have made public. For public puzzles, there will be a tracker to count how many moves are made per puzzle, which is then compared to the ‘world’ average.

APIs:

COLOURlovers API: <http://www.colourlovers.com/api#palettes>

IP Identifier: <http://ip-api.com/>

Weather API: <https://openweathermap.org/api>

Components:

Front End:

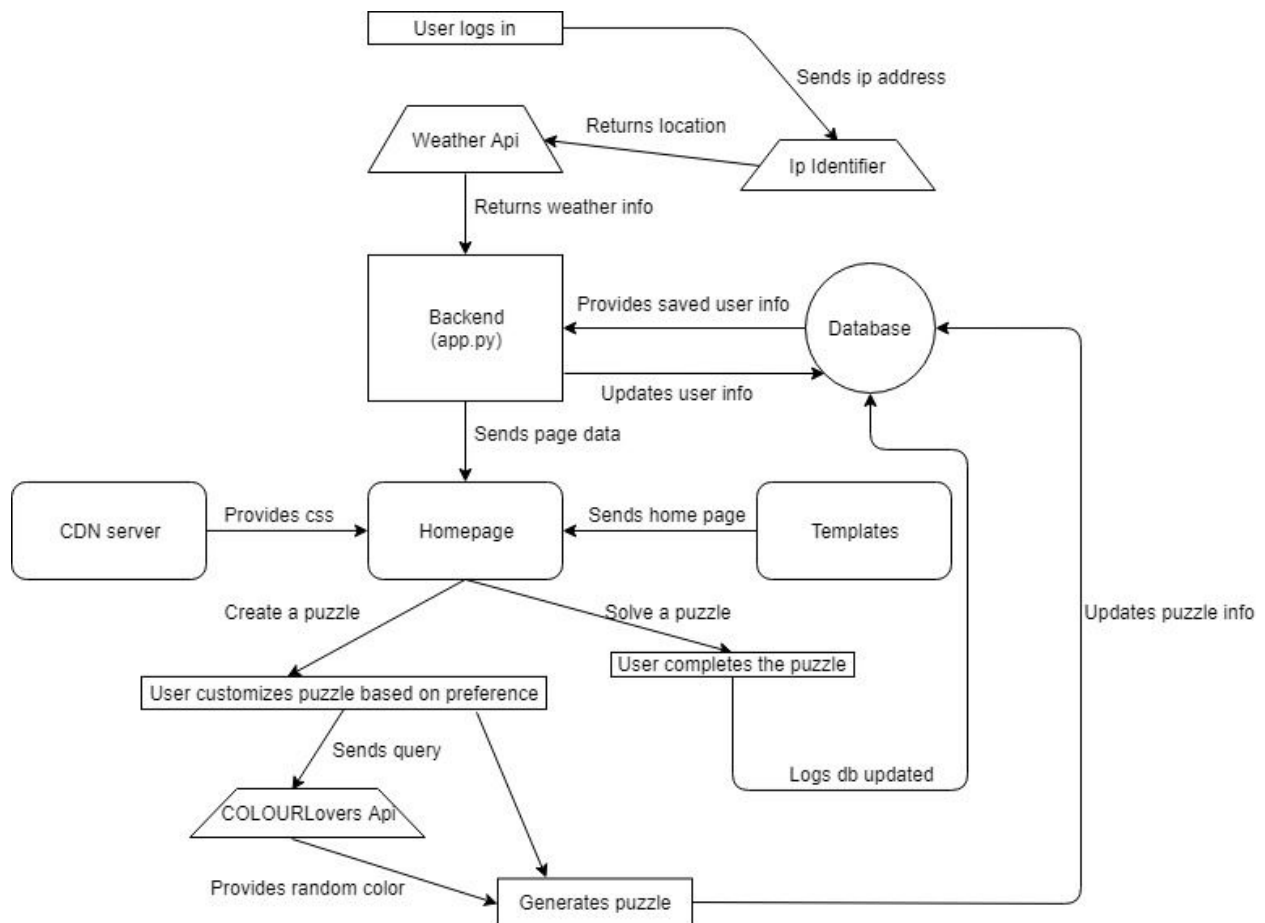
- Create an account, log in
- Create a puzzle
 - Generate color scheme from API, filter through any customization and process them through an HTML page
 - Jinja, html, bootstrap, java
 - Grant user the option to ‘save’ a puzzle which will be saved in their account
 - Grant user the option to ‘share’ a puzzle which will allow other users to view/play it
- Solve a puzzle (one that a player created or a preset one)

- Javascript
- Allow users to click two pieces and initiate a switch
- Tracks the number of moves it takes to complete
- Prevents users from making moves when the puzzle is solved
- HTML/CSS Templates
 - Jinja, html, bootstrap, javascript

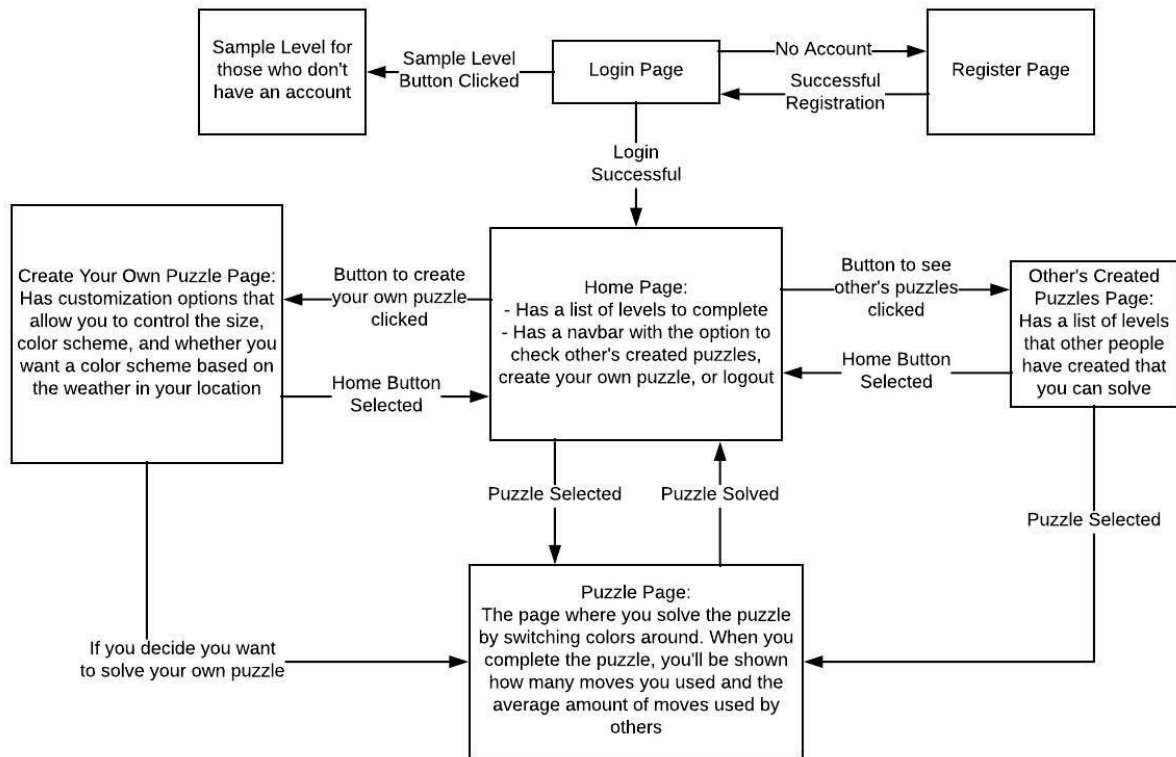
Back End:

- Flask route functions to handle the APIs, and rendering each page as appropriate.
- Work with the database to generate saved puzzles so that users can replay levels (will be displayed in a personal feed)
- Work with the database to generate public puzzles so that users can play alternate levels
- Create sessions
- Track moves of users to complete a puzzle to establish a 'world' average

Component Map:



Site map:



Database Schema:

- Facilitate logging in/out of users, as well as storing passwords(sessions) using dbs
- Maintain the storage of puzzles by assigning a puzzle an id, its content (colors), and how many moves, on average, it took to complete it
- Logs will track the history of a puzzle's completion including the username of those who completed the puzzle and the number of moves it took for them to complete it,

USERINFO

| | |
|--------------|------|
| username | TEXT |
| pass | TEXT |
| moves | INT |
| liked_puzzle | TEXT |

PUZZLES

| | |
|--------------------|------|
| Puzzle_id | INT |
| puzzle_description | TEXT |
| average_moves | INT |

LOGS

| | |
|-----------|------|
| user | TEXT |
| moves | INT |
| puzzle_id | INT |

Role Distribution:

- Dennis:
 - Backend: Javascript, Python and flask functions
- Addison:
 - Frontend: Javascript and HTML/CSS
- Fabiha:
 - Project Manager || Prime Minister || Pretty Marvelous || Perfectly Mindful
- Michelle:
 - Backend: Python functions, and database creation and code for modification, creation or retrieval of entries. (For both tables within the database)