JEOPARDY! by Team Saturated Sidewalks

Roles:

Peihua Huang : Project Manager
Alice Ni : Frontend (Bootstrap)/APIs
Jacob Olin : Databases
Hilary Zen : Flask

Project Description:

We will recreate the famous Jeopardy game. After users login, they will be able to create their own Jeopardy boards with their own categories and questions. These boards will be saved and be open for all other users to play. They can also choose a randomly generated board with predetermined categories and random questions from the APIs. Users can also search for an existing board based on the name of the board. When a user decides on a board to play, they will first have to specify how many teams are playing and the team names. There will be a limit of 5 teams. Once the game begins, the teams and their scores will be displayed on the bottom of the page. After choosing a question, the answer will be revealed with a click of a button after the team answers. If the answer matches, the user can press the + button to add the points to that team. If the answer is wrong, the user can press the - button to deduct the points. The game ends when all of the categories have been chosen and the winner of the game will be recorded in the user's game history.

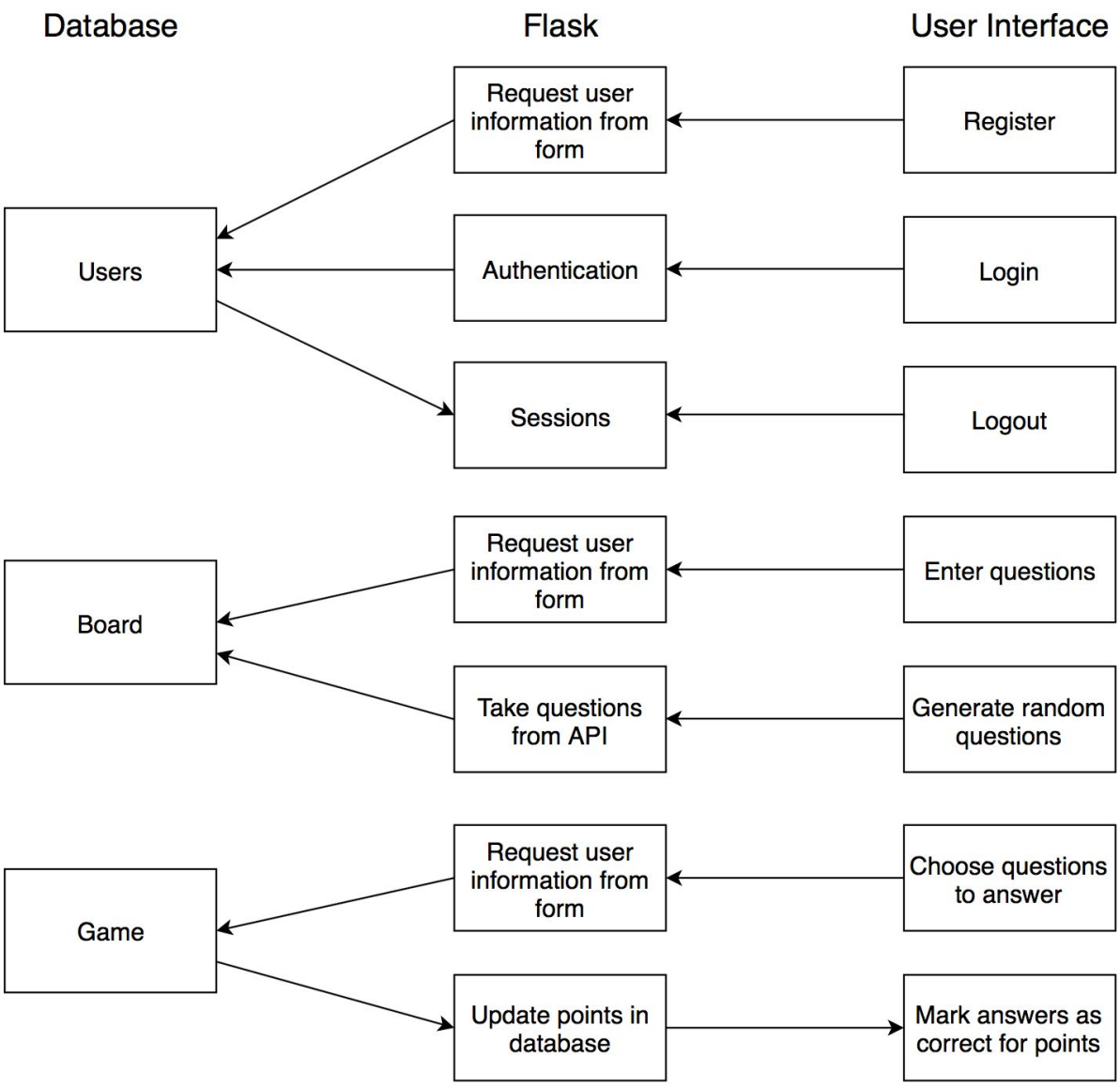
Core Components:

- Login system
- Users can start games, which will be recorded in their history
- Randomly generating questions
- Form to create their own board
- Keep track of number of rounds, which team is answering, and each team's points
- Determine a winner at the end of the game

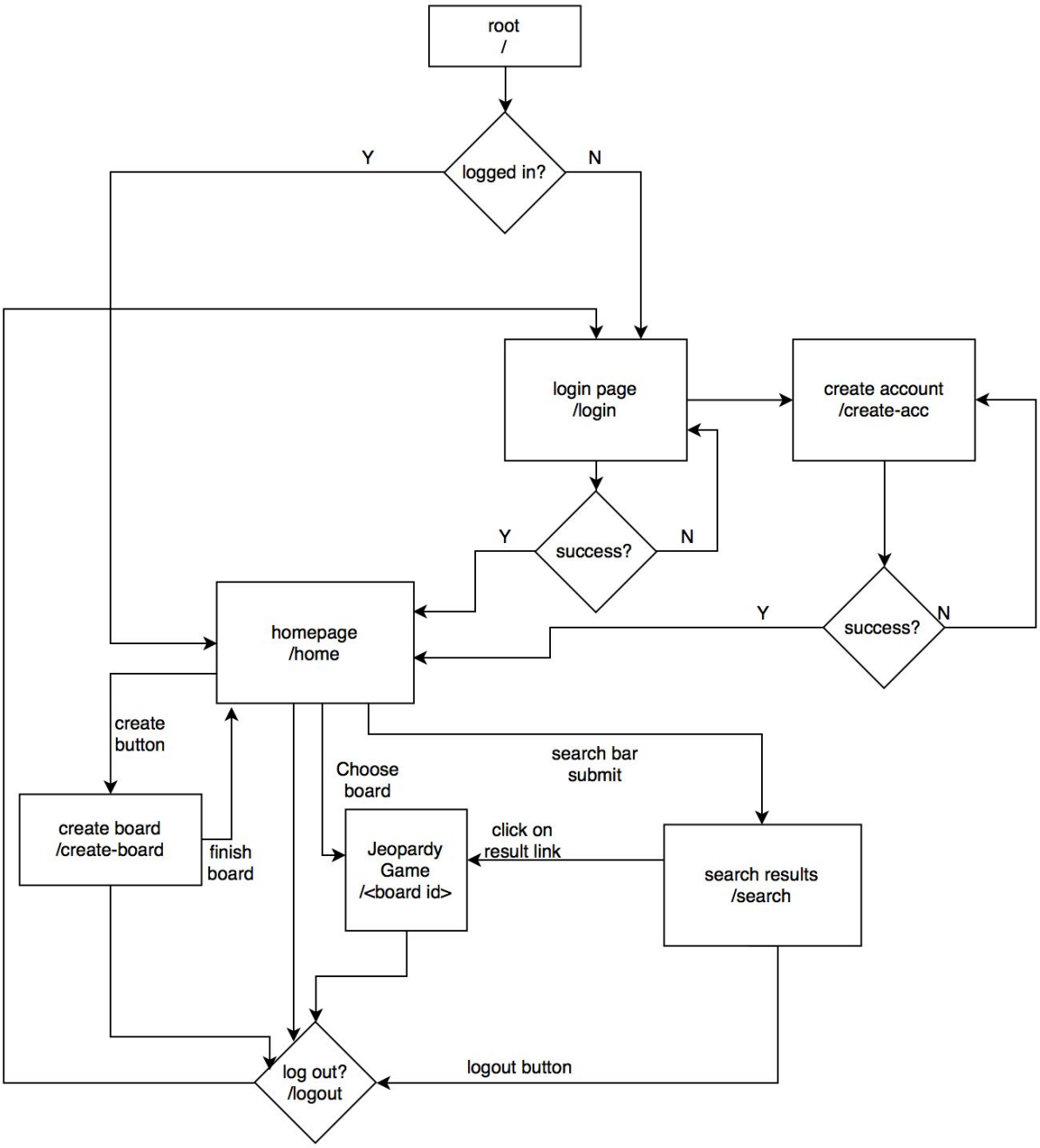
APIs Used:

Open Trivia - used to provide questions for the randomly generated boards
Rest Countries - used to get country flags and names for randomly generated questions
Rick and Morty - used to get character images and names for randomly generated questions
PokéAPI - used to get Pokemon images and names for randomly generated questions

Component Map:



Site Map:


root
/
logged in?
Y
N
login page
/login
create account
/create-acc
success?
Y
N
success?
Y
N
homepage
/home
create
button
Choose
board
search bar
submit
create board
/create-board
finish
board
Jeopardy
Game
/<board id>
click on
result link
search results
/search
log out?
/logout
logout button

Database:

Users

user_id	username	password
INTEGER	TEXT	TEXT
3892	bob	123

Board

User id	Board id	category	Q1	Q2	Q3	Q4	Q5
INT	INT	TEXT	INT	INT	INT	INT	INT
3892	4337	pokemon	0	1	2	3	4
...	(5 rows per board id)	...	...	...	...	...	...
3892	4337	countries	20	21	22	23	24

Teams

Game id	Team name	Score
INT	TEXT	INT
5238	Saturated Sidewalks	500

Board status

Board id	category	Q1	Q2	Q3	Q4	Q5
INT	TEXT	INT 0 or 1 (Y or N)	INT 0 or 1 (Y or N)	INT 0 or 1 (Y or N)	INT 0 or 1 (Y or N)	INT 0 or 1 (Y or N)
4337	pokemon	0	0	0	1	0

Questions

Question id	Category	Question	Answer
INT	TEXT	TEXT	TEXT
0	pokemon	What is Pikachu?	<image link>

Front End:

- **base.html**
    - Base template for all the pages
- **login.html**
    - Form for login
    - Redirects to “/login” if unsuccessful, else redirects to “/home”
- **register.html**
    - Form for registration
    - Redirects to “/login”
- **home.html**
    - Search bar for available games
    - Button to create game
        - Redirects to “/create-board”
    - Button to play games
        - Redirects to “/play-board”
- **create.html**
    - Form for creating a jeopardy game
    - Redirects to “/home”
- **play.html**
    - Game screen

Back End:

- **create_db.py**
    - SQL code to create the database tables needed
- **app.py**
    - **register():** Users can create a new account
    - **login():** Authenticates users who are logging in
    - **create():** Uses APIs to generate a board with questions
    - **customize():** Takes the 25 questions and answers that user enters, and creates a new board with them
    - **search():** Filters through all existing boards and returns matching results
    - **play():** Moves user into the game screen
    - **question():** When user chooses a box, the question comes up
    - **show_answer():** After user submits an answer, the correct one is shown
    - **update_points():** User can mark whether the answer is correct or not, and points will be added or subtracted
    - **finish():** After the last question has been answered, the game automatically shows a winner, and returns users to the homepage