

Team MAWD - Mohidul Abedin, William Cao, Devin Lin, Alex Olteanu

SoftDev1 pd 2

P #01: ArRESTed Development

2019-12-04

Roles:

1. Mohidul Abedin - Project Manager, Balance
2. William Cao - Pokemon Game
3. Devin Lin - Rick and Morty Game
4. Alex Olteanu - To21 game

General Usage: Surprise mechanics simulator

APIs Used: Currency Exchange API, Poke API, Card Deck API, and Rick and Morty API

Frontend: Bootstrap

Program Components:

- Accounts
 - Users are directed to a page where they can either login to their account or register a new account
 - Login- Leads to a new page where the user can input their credentials. If correct the user will be prompted to homepage with all their stories, else they would be redirected to the login page
 - Register- Leads to a new page where the user has to input their information. Upon registering, the user will be directed to their newly created homepage
- Profile
 - From the homepage the user can access their profile page
 - They can view
 1. their balance of MAWDollars (Test user starts with MAWD\$1,000)
 2. Account info
 3. Any purchases made (nice to have)
- Game general information:
 - Once the player enters the amount he wants to play and clicks submit, the amount played is removed from his balance. If the user leaves mid game, he loses what he enters.
- Pokemon (/pokemon)
 - Instruction to play the game:
 1. User will choose how much in-game currency he wants to spend on the game.
 - a. The only options are: 2, 10, and 20 MAWDollars.
 - b. If the user doesn't have enough money, he can add more funds (more details later)
 - c. There is a button to go back to the game page and a button that leads to instruction
 - d. Once the user decides on the amount he wants to spend, he clicks submit
 2. The submit leads to the game page:

- a. The computer will have four cards facing down (blank squares)
 - b. The user will have four cards facing up. There will be a picture of the pokemon, name, and the type. Once the user selects one, he will click the submit button.
 - 3. The submit leads to the end page:
 - a. Computer's cards will face up and the cards that were selected by the user and computer will be highlighted.
 - b. There will be a message of who won and how much MAWDollars was gained/lost
 - c. User can continue with another game by clicking "Play Again" or quit by clicking the "Give Up" button
- User is playing against the computer. The computer will not have any sophisticated AI, it will just choose a random card.
- To21
 - User challenges a computer in To21 and wagers a certain amount of MAWDollars
 - User is given 2 cards, and is able to request more
 - If the user wins, wins doubled what he spend on. If user loses, he loses all of what he played
 - Card values:
 - 1. Number: Worth the number it represents
 - 2. King / Queen / Joker: Worth 10
 - 3. Ace: Worth either 1 or 11. Value will work out in favor of the player
 - Player will go first then computer.
- Rick and Morty
 - Instructions to play the game
 - 1. User will choose how much money he wants to spend on the game.
 - a. The only options are: 2, 10, and 20 MAWDollars.
 - b. If the user doesn't have enough money, he can add more funds
 - c. There is a button to go back to the game page and a button that leads to instruction
 - d. Once the user decides on the amount he wants to spend, he clicks submit
 - 2. The submit leads to the game page:
 - a. There will be three pictures of characters from the iconic show Rick and Morty
 - b. 4 options will appear below each character one of which will be the name of the character—the others being randomly chosen character names from Rick and Morty other than the real name
 - c. User will fill out the questions and proceed to click "submit"
 - 3. The submit leads to the end page:
 - a. There will be three pictures of characters from the iconic show Rick and Morty
 - b. The actual name of each character will be displayed below each picture
 - c. Number correct will be displayed in the form of a fraction
 - d. The amount of MAWDollars gained or lost will be displayed next to the number correct
 - e. User can continue with another game by clicking "Play Again" or quit by clicking the "Give Up" button
 - If user scores 0/3→ all MAWDollars wagered will be lost

- If user scores $\frac{1}{3}$ → half of MAWDollars wagered will be lost
- If user scores $\frac{2}{3}$ → no MAWDollars will be lost
- If user scores $\frac{3}{3}$ → double up on MAWDollars wagered
- Redeem
 - The user can redeem their MAWDollars to any form of currency (1:1 USD ratio)
 - Upon redeeming the their balance is a convenience fee of 5%, along with any excess cents
 - The money is sent to their paypal account
- Pay
 - In order to acquire more MAWDollars to play games, the user can purchase it using their credit/debit card
 - Only whole numbers or else it will throw an error

Pokemon (More explanation)

- If a Pokemon only has 1 type, its primary and secondary type is the same
- Pokemon A (Bug and Ice) and Pokemon B (Fire and Fighting)
 - Bug will attack fire and fighting: Bug x Fire: $\frac{1}{2}$ __ Bug x Fighting: $\frac{1}{2}$, so total is 1
 - Fire will attack Bug and Ice: Fire x Bug: 2 __ Fire x Ice: 2, so the total is 4
 - Winner: Pokemon B
 - Prize Division: Gets Whole prize pool
- Prize Division:
 - If difference is: > 3 : 2x
 - If difference is < 2 , > 1 : 1.5x
 - If difference is < 1 : Tie, no money divided

Backend info

- Session:
 - current_game: Text
 - Keep track of what game the user is playing. This will prevent users from refreshing the page when they have a losing hand.
 - Possible values: “Pokemon”, “to21”, “rickmorty”
 - When the user first selects a game, the **session[“current_game”]** needs to be set
 - paid
 - **True** if user already paid, **False** otherwise. **This needs to be set to False after you check if the user paid.**
- Example:


```
if not session["paid"]:
    return redirect(url_for("bet"))
session["paid"] = False
```
- Login
 - The user’s login info

Database

User:

users	
username	text
password	text
balance	int

Rick and Morty Game:

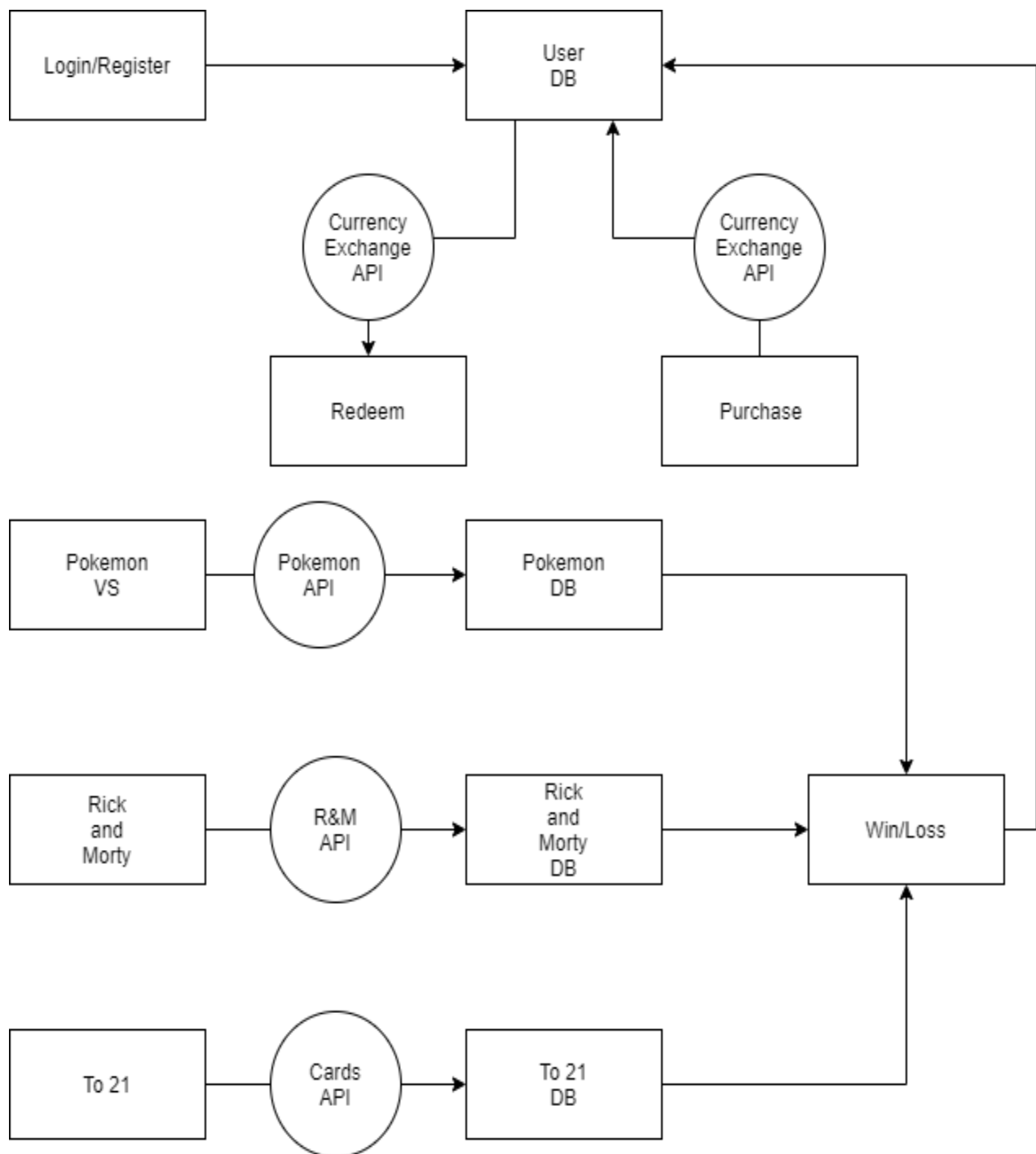
characters	
full_name	TEXT
imagelink	TEXT

Pokemon:

pokemon		type	
id	INTEGER	id	INTEGER
name	TEXT	name	TEXT
picture_link	TEXT	effective	TEXT
type1	INTEGER	weak	TEXT
type2	INTEGER		

To21:

cards	
suit (TEXT)	
suit	TEXT
value	TEXT
svg	TEXT



Component Map

