

The **Mistakes**: Jesse Xie (PM), Alif Abdullah, Daniel Sooknanan, Raymond Yeung  
SoftDev

P01 -- ArRESTed Development Design Doc

2021-12-09

---

### **General Description:**

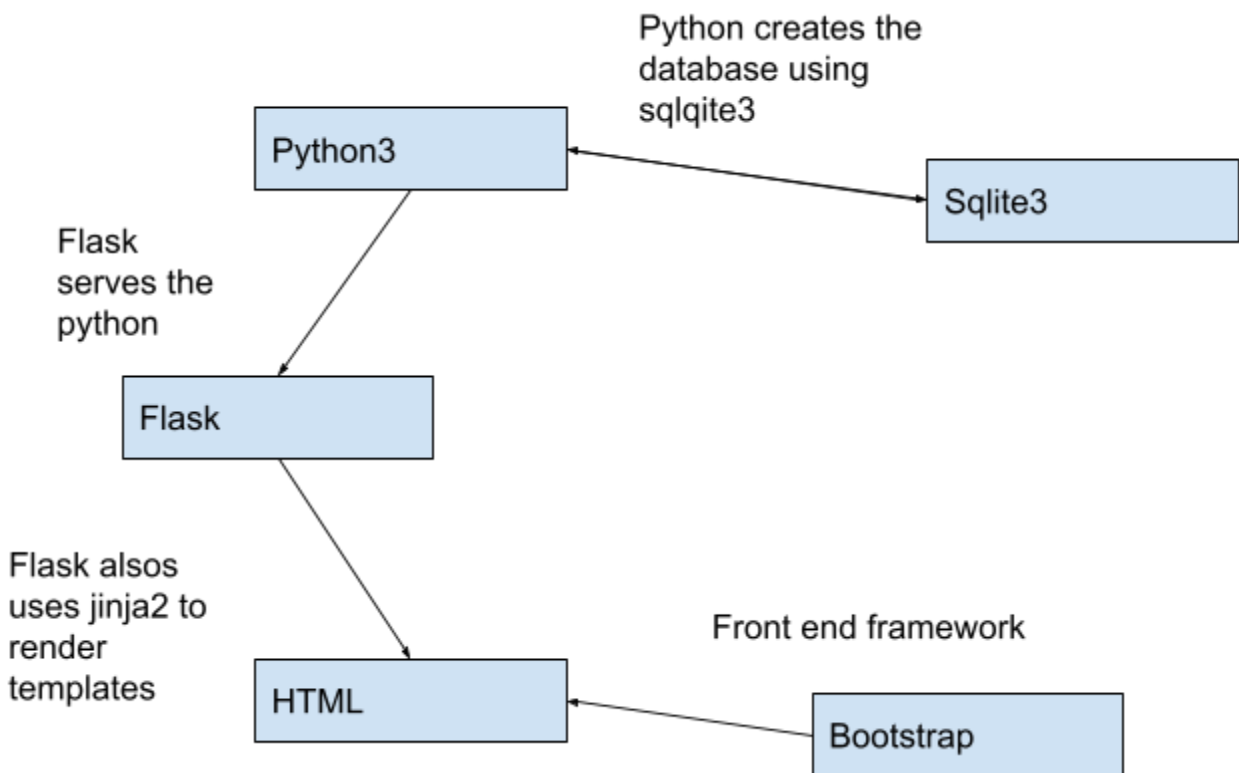
Our web application, called *APHigh Mi-stakes*, allows users to create an account and earn points / items through playing blackjack with a number of CPUs. Points are the generalized currency used for buying items from the market that is available to the user. These items are solely cosmetics that will decorate their profile and ranking on the leaderboard, including name color, profile picture (PNG / JPG or GIF). After winning a match, the player will be randomly awarded such an item. The aforementioned leaderboard will rank and display all players based on the number of wins they have.

### **Program Components:**

- Flask
  - We will use flask to host our webpage, save sessions, and facilitate onscreen interactions with the user.
  - Data inputted into flask forms will be saved into tables in our sqlite3 database.
- Python3
  - We will use python to facilitate interactions with the database and the other components.
  - Python will create and fill in databases, as well as work with flask and html to make our pages
- Sqlite3
  - We will use sqlite3 as our backend data storage system, and save our relevant data in tables.
- Bootstrap

- We will use bootstrap as our framework in creating a visually pleasing and user friendly interface.
- Bootstrap will work with HTML to create pages with all the necessary features.
- HTML
  - We will use HTML to create the basic version of our webpage which will be fleshed out by bootstrap.

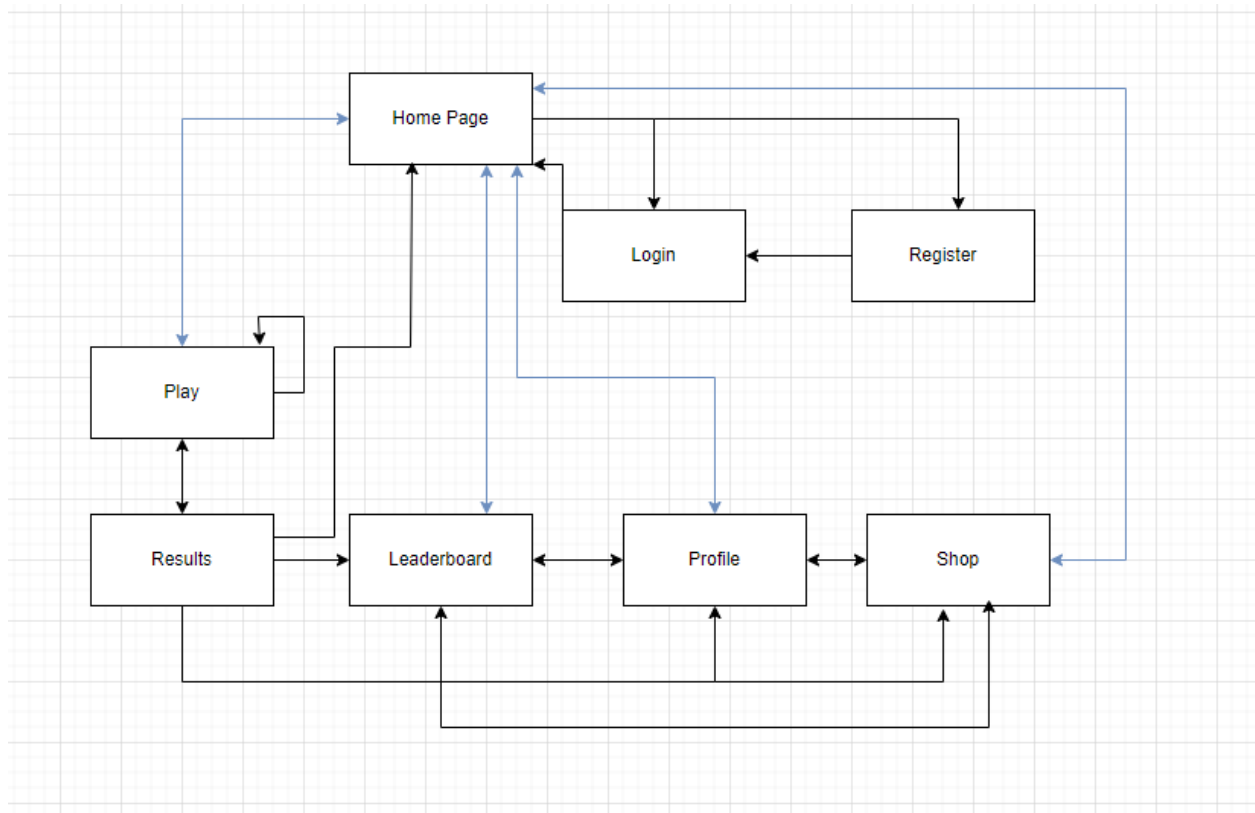
### Component Map:



### Database Organization:

- Users table: username TEXT, password TEXT, points INTEGER, wins INTEGER, losses INTEGER
- Items table: name TEXT, imageUrl TEXT, owner TEXT
- Market table: name TEXT, imageUrl TEXT, price INTEGER

### Site Map:



### Page Descriptions:

Login / Register (*Not Logged In*, Route: /) - Prompts the user to enter their login credentials, or sign up for an account if they do not currently have one

Home (*Logged In, Route: /*) - Central hub to allow users access to Play, Profile, Shop, and Leaderboard.

Play (*Logged In, Route: /play*) - Allows the user to play against a set number N CPUs in the card game Blackjack.

Results (*Logged In, Route: /results*) - Awards the user points based on whether they won the game or lost. Gives random rewards (new colors, profile pictures).

Leaderboard (*Logged In, Route: /leaderboard*) - Displays a list of users ranked on how many wins they have. Also, displays their current amount of total points.

Shop / Market (*Logged In, Route: /leaderboard*) - Displays a grid of purchasable items (colors, profile pictures). Users can use points (earned from winning games) to buy such items and use them on their profiles. Available items cycle on a weekly basis or a daily basis (depending on the type of item).

Profile - Allows the user to change their profile picture, color, & display name.

#### **APIs:**

- Deck of Cards
  - The Deck of Cards API will help us in dealing, shuffling, and returning cards for our blackjack game.
- Color
  - The Color API will generate a random color that we will use for cosmetic purposes (ex. name color, table background, card color, etc).
- Picsum (may change based on needs)
  - Generates a random picture with a specified size.

#### **Framework(why/how):**

- Bootstrap: Personally, we prefer the structure and keywords utilized in Bootstrap as opposed to Foundation. More specifically, we'll make use of features like the navbar to create a dynamic website that is also aesthetically pleasing.

### Roles:

- Jesse (PM): Manage backend development of database, facilitate interactions with python
- Alif: Manage development of python code, establish interactions between flask and python
- Daniel: Manage front end development, create easy to use interface
- Raymond: Manage API data retrieval, ensure proper utilization of APIs

**Target Ship Date:**

December 23, 2021

### Template Representations:

Home page:

The diagram illustrates a web application layout. At the top, there is a navigation bar with six buttons, each with a label and a condition in parentheses:

- Sign Up (Appears if you are not logged in)
- Login (Appears if you are not logged in)
- Profile (Appears if are logged in)
- Leaderboard (Appears if you are logged in)
- Shop (Appears if you are logged in)
- Play (Appears if you are logged in)

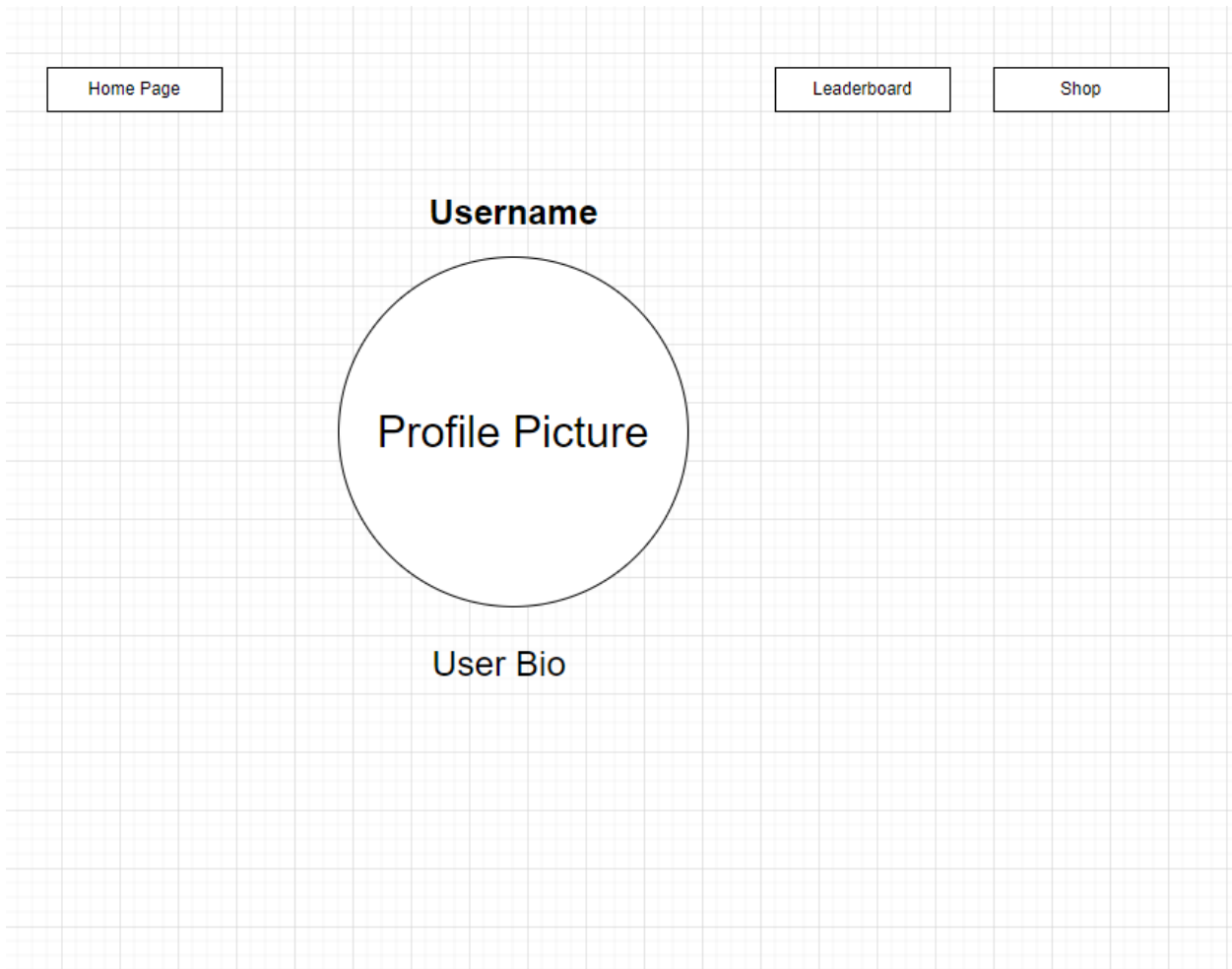
Below the navigation bar is a large grid area. The first row of the grid contains the text "Greeting to site". The second row of the grid contains the text "Descriptions of what to do".

Sign up/ Login page:

Username:

Password:

Profile page:



Leaderboard:

NAV BAR

USER 

#n  User

k wins   p points

#n  User

k wins   p points

#n  User

k wins   p points

#n  User

k wins   p points

#n  User

k wins   p points

#n  User


k wins   p points


#n  User

k wins   p points


Shop:

NAV BAR


USER 




Profile Picture  
n points




Profile Picture  
n points




Profile Picture  
n points




Profile Picture  
n points



n points  
#FFFFFF  
Color



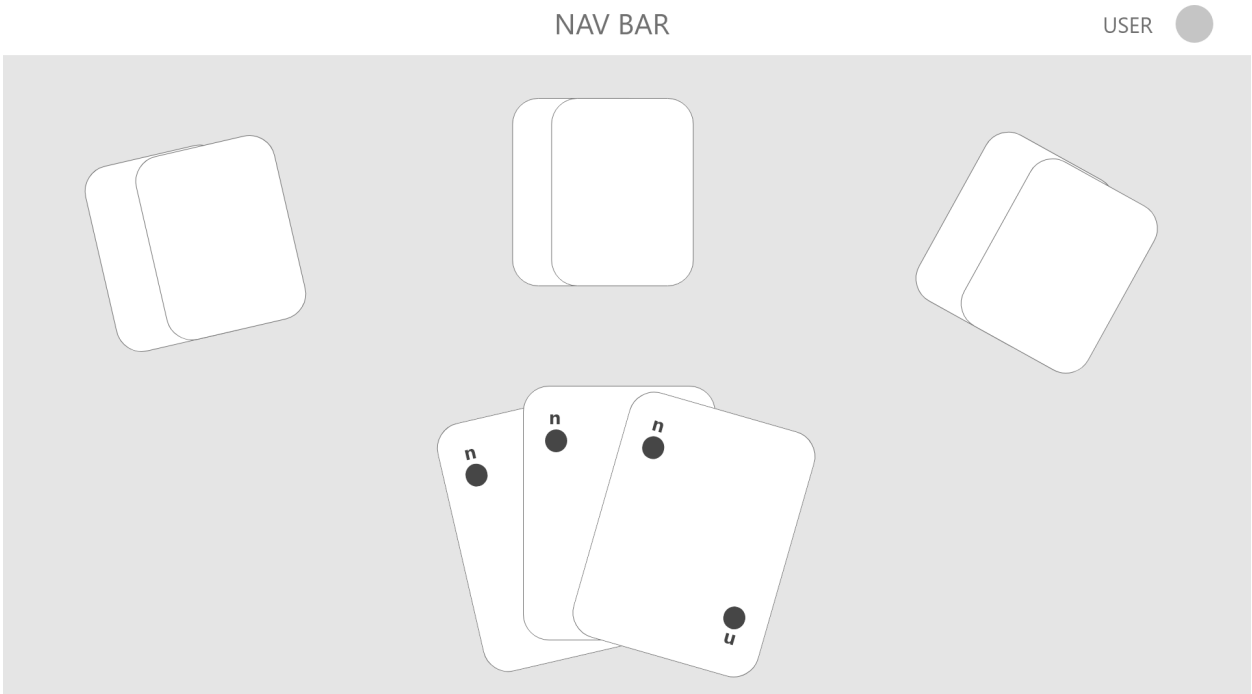
n points  
#FFFFFF  
Color



n points  
#FFFFFF  
Color



Play:



Results:

