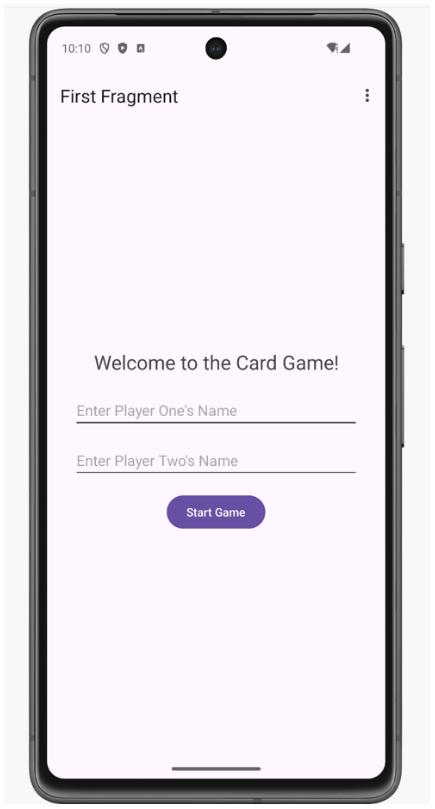
Project 4 – Poker Game App

Louis Chang (hungyic)

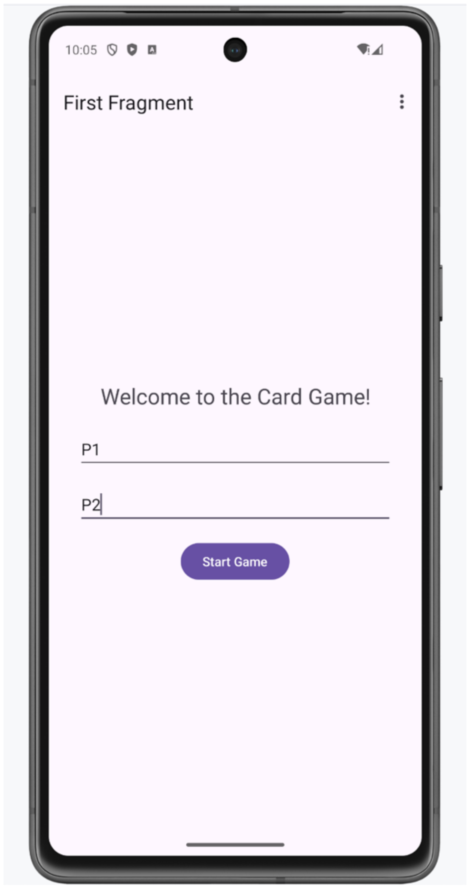
Description:

My application creates a poker game for two players. When the game starts, they will get two cards from a deck randomly, and the player with bigger value of card earn 1 point.

Here's how my application meets the task requirements:

1. Implement a native Android application
    a. Has at least three different kinds of views in the layout, including TextView, EditText, ImageView, and Button.
    Here's a screenshot of the layout before the new deck of card has been created
    

    b. Here's a screenshot of the user entering the name of the two players

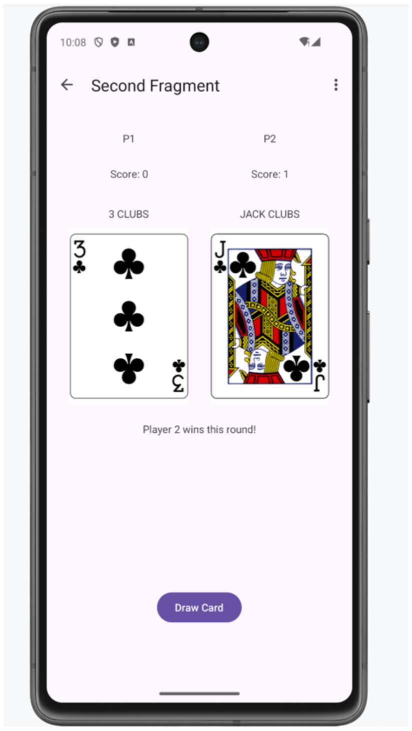


c. My application does an HTTP GET request and an HTTP POST request in CardServlet.java. The GET request is https://didactic-space-memory-657q67946j4crp9q-8080.app.github.dev/card?uuid=805d5f61-9aa8-427e-8a6b-bd1c6da47a63 , where uuid is the Universally Unique Identifier that each game session generates. The POST request is "", with no parameters but with uuid in request body.

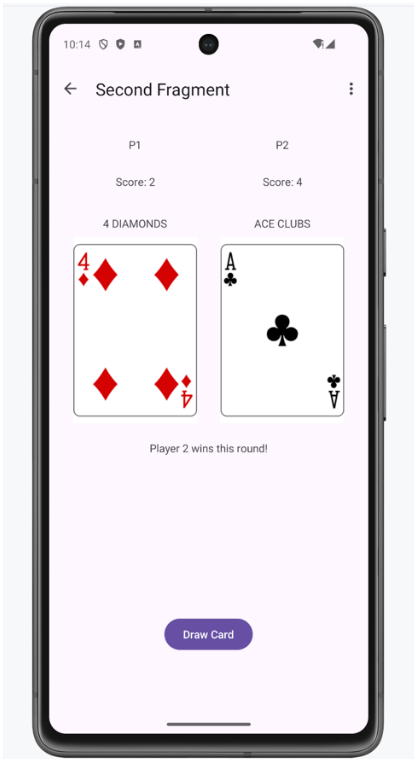
d. An example of the JSON response is:

```
{
    "success": true,
    "deck_id": "3v1kls55n9kq",
    "remaining": 52,
    "shuffled": true
}
```

e. Here's the screenshot after the card has been returned.



f. The players can keep drawing the card without restarting the application.



2. The URL of my web service deployed to CodeSpace is https://didactic-space-memory-657q67946j4crp9q-8080.app.github.dev
    a. In my web app project:
        - Model: CardServlet.java

   - View: index.jsp
   - Controller: DashboardServlet.java

b. CardServlet.java receives the HTTP GET request with no argument and with argument “deck_id”.

c. CardServlet.java makes HTTP requests to https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1 to create a new deck of card, to https://deckofcardsapi.com/api/deck/k8g9augvy69d/draw/?count=2 to draw two cards, and to https://deckofcardsapi.com/api/deck/iw30vc44rc7o/shuffle/ to shuffle cards

d. The responses:
   - Create a new deck

```
{
    "success": true,
    "deck_id": "10oee9fphy40",
    "remaining": 52,
    "shuffled": true
}
```

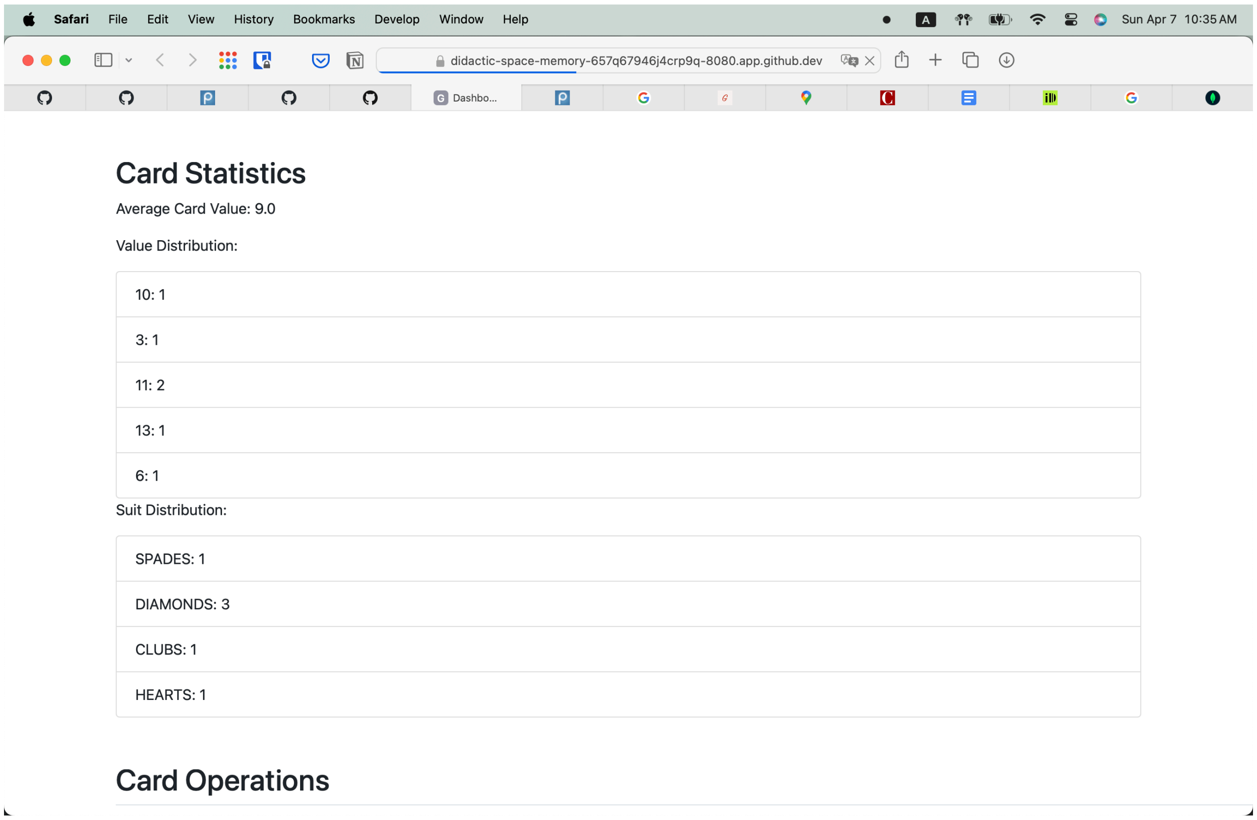
   - Draw 2 cards

```
{
  "success": true,
  "deck_id": "10oee9fphy40",
  "cards": [
    {
      "code": "KH",
      "image":
"https://deckofcardsapi.com/static/img/KH.png",
      "images": {
        "svg":
"https://deckofcardsapi.com/static/img/KH.svg",
        "png":
"https://deckofcardsapi.com/static/img/KH.png"
      },
      "value": "KING",
      "suit": "HEARTS"
    },
    {
      "code": "AC",
      "image":
"https://deckofcardsapi.com/static/img/AC.png",
      "images": {
        "svg":
"https://deckofcardsapi.com/static/img/AC.svg",
        "png":
"https://deckofcardsapi.com/static/img/AC.png"
      },
      "value": "ACE",
      "suit": "CLUBS"
    }
  ],
  "remaining": 48
}
```

- Shuffle the deck

```
{
  "success": true,
  "deck_id": "1Ooee9fphy40",
  "remaining": 52,
  "shuffled": true
}
```

3. Handle error conditions
4. Log useful information including User Agent, UUID, Timestamp, Method, IP Address, and Response Details
5. Connection String: mongodb+srv://hungyic:s17krRBnBmShobWE@cluster0.czpxrau.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0
6. Operations Analytics include average card value, value distribution, and suit distribution


Safari File Edit View History Bookmarks Develop Window Help
Sun Apr 7 10:35 AM
didactic-space-memory-657q67946j4crp9q-8080.app.github.dev
Dashbo...

Card Statistics

Average Card Value: 9.0

Value Distribution:

```
10: 1
3: 1
11: 2
13: 1
6: 1
```

Suit Distribution:

```
SPADES: 1
DIAMONDS: 3
CLUBS: 1
HEARTS: 1
```

Card Operations