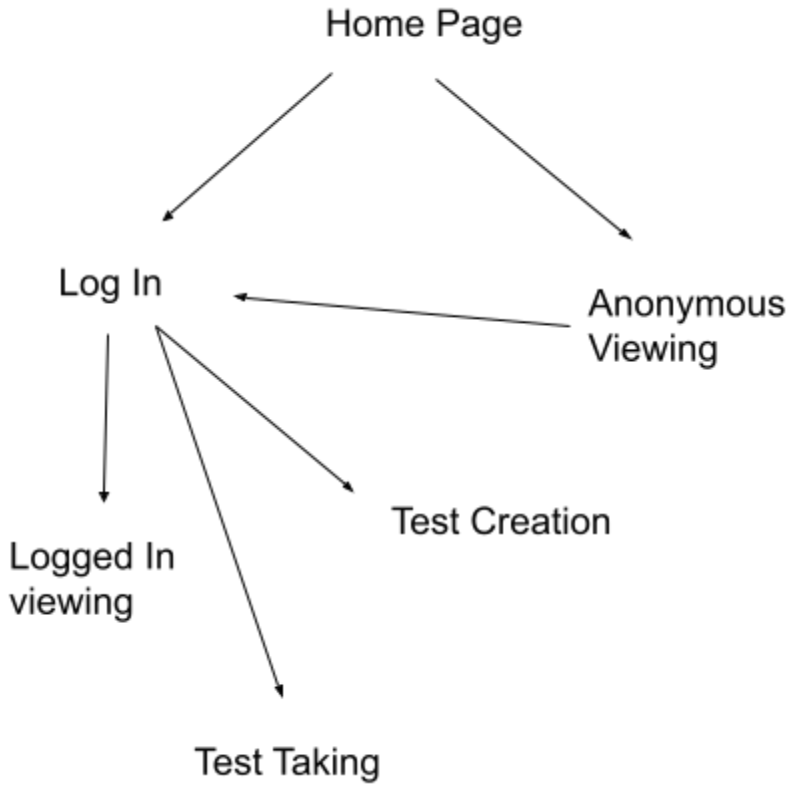
Team GoLang - Sean Ging (PM), Patrick Ging, David Chong, Aryaman Goenka
Design Doc:

Due Date: June 14th 2022

Concept: Using AI to create a quizlet-like scenario, using REACT for the user to take the quiz

- Limited amount that our "camera" can decipher
- Using Node.js, express.js, REACT predominantly

General Navigation:



- Uses Tesseract in the microservice perhaps Rabbit MQ for queue management

Current Program Components:

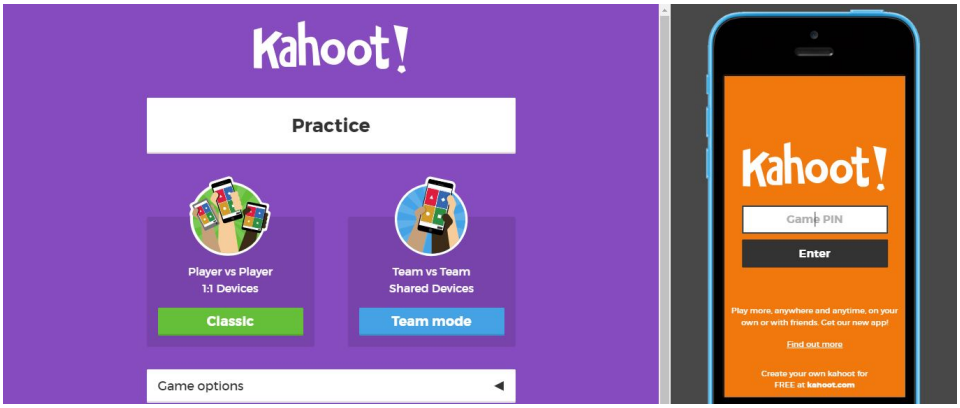
- **Home Page**
  - **User, login page**
  - **Why? So we can save different quizzes for users!**
- **Processing page**

    - **Form for uploading documents**
        - **Once uploaded, content will be sent to tesseract microservice**
            - **Users can correct and add-on to the created document**
    - **Alternatively one can manually input their information**
- **Quiz page**
    - **Using REACT, html, js**
    - **Allows for users to actively learn what they are reading**
- **Tesseract**
    - **OCR Engine that can transcribe images**
    - **Using python wrapper, possibly a fine-tuned engine**
    - **Going to be on a separate microservice than the flask instance**
- **Database**
    - **Upload information**
    - **Render it onto the quiz page using**
        - **Sqlite**
        - **Jinja Templating**
        - **React**
- **Flask, Apache**
    - **Host web service**
- **Question generator**
    - **Four answer choices per question**
    - **Saved on database**
    - **One choice saved as correct answer**
    - **API(s)?**
    - **AI(s)?**

<u>Stretch goals</u>

- **Open-ended question**
    - **We can use AI to see how "accurate" the answer is to the correct answer**

- **Kahoot-like interface**

- 

- **Timed-questions, points for correct answers!**
- **Multiplayer options**
  - **Allow many to answer questions at once, take the same quiz at once**
  - **Set up a server**