

# COIN COUNTER

**TEAM: THE CEREAL KILLERS**

**GROUP MEMBERS :**

**BEKA ADMASU, AANVI GULIANI, JACOB REYNOLDS, JASKRIT SINGH,  
JACI STICKROD, VIENNA WONG**

# COINCOUNTER

**Goal:** To help users efficiently count their coins!

**Description:** Program takes in an image of coins laid out on a flat surface and the app will return the value of the coins!

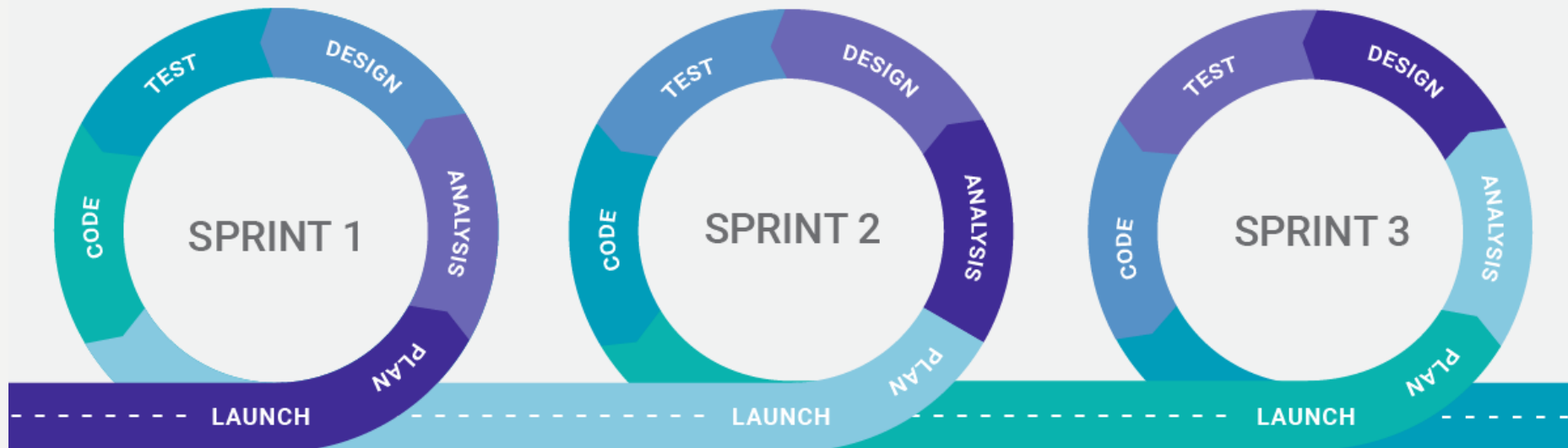
**Purpose:** Helping people who are unfamiliar with US coins count their currency.

## About Coin Counter



# METHODOLOGY

- Our team used Agile/ Scrum Methodology (Specifically Pair Programming)
  - **Rating:** ☆☆☆☆☆ (5/5 stars) because of organization of roles and efficiency of task completion
  - **Purpose:** Used Agile to increase project control throughout development. Agile also helps improve efficiency and make sure tasks are being completed in a timely manner. Agile ensures development of a high quality product!



# PROJECT MANAGEMENT & COMMUNICATION

- Management Tools:  
**Monday.com Rating:** ☆☆ (2/5 stars) – Initially helpful for group communication, but was much less useful in comparison to GroupMe. Our team eventually stopped using Monday.com
  - **Purpose:** Project Tracker
- **GroupMe Rating:** ☆☆☆☆☆ (5/5 stars) – Very helpful for instantly communicating with all group members
  - **Purpose:** Instant Group Communication
- Weekly meetings on Fridays from 1-3pm



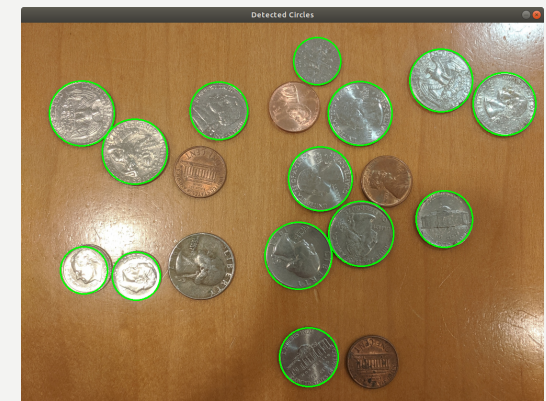
# VCS REPOSITORY

- Version Control Tool: GitHub- Project information added to three repositories:
  1. [Team Meeting Log Repo](#)
  2. [Milestone Submission Repo](#)
  3. [All project code/components Repo](#)
  - **Rating:** ☆☆☆☆☆ (5/5 stars)
  - **Purpose:** Tool for version control and organizing project materials.
  - Good organization of all project components and easy access to all group members.



# Our Python Backend

- We used OpenCV to detect the coins
  - **Rating:** ☆☆☆☆ (4/5 stars) because it was tough to learn, but had all the resources we needed
  - **Purpose:** It allowed us to filter images down to a simple black and white mask and use the mask to detect the circles of the coins



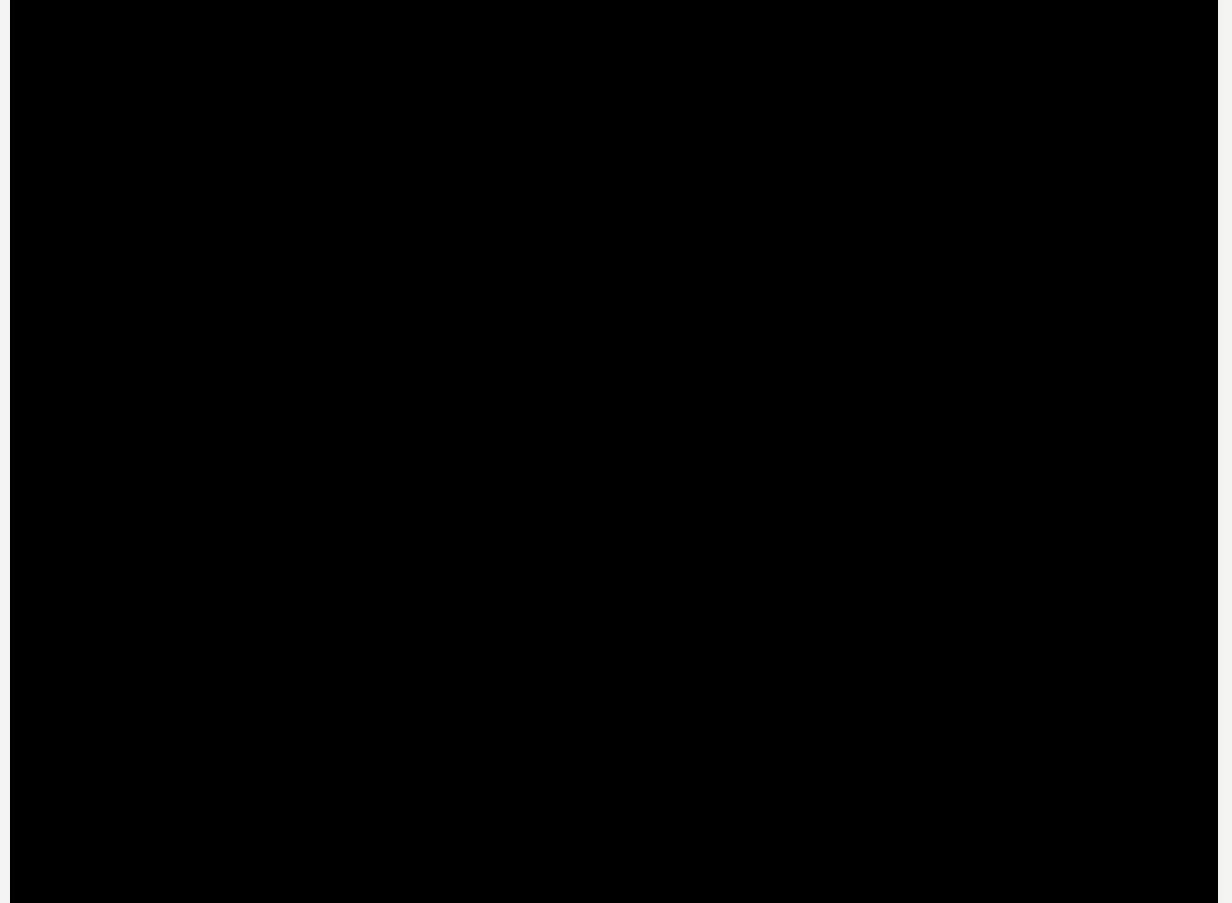
# DATABASE

- Tool: SQLite

- Rating: ★★★★★ (5/5 stars)

- Purpose: Store history of coin counts. Very easy to pick up with beginners knowledge and seamless integration with Swift!

Press play to see the database in action!



# TESTING TOOL

- Tool Used for Frontend: **Swift Simulator**

**Rating:** ★★☆☆ (4/5 stars)

**Purpose:** To test UI elements and button functionality. Sometimes the simulator behaved differently than the app on an iPhone.

- Tool used for Backend: Manual photo upload and test cases

**Rating:** ★★☆☆ (3/5 stars)

**Purpose:** Test our coin detection with different images to see how well it performed in different situations

- Tool used for endpoint testing: **Postman**

**Rating:** ★★★★★ (5/5 stars)

**Purpose:** Used to test endpoints when creating API with Flask





# DEPLOYMENT ENVIRONMENT

- Heroku

**Rating:** ★★★★★ (5/5 stars)- Easy and effective deployment tool

**Purpose:** Used Heroku to deploy our app!



# Frontend

- XCode/ Swift

**Rating:** ★★☆☆ (3/5 stars)

**Purpose:** Create a user interface that allows users to take a photo and presents the correct value of coins in the photo. Swift wasn't like any other coding language we had previously worked with.



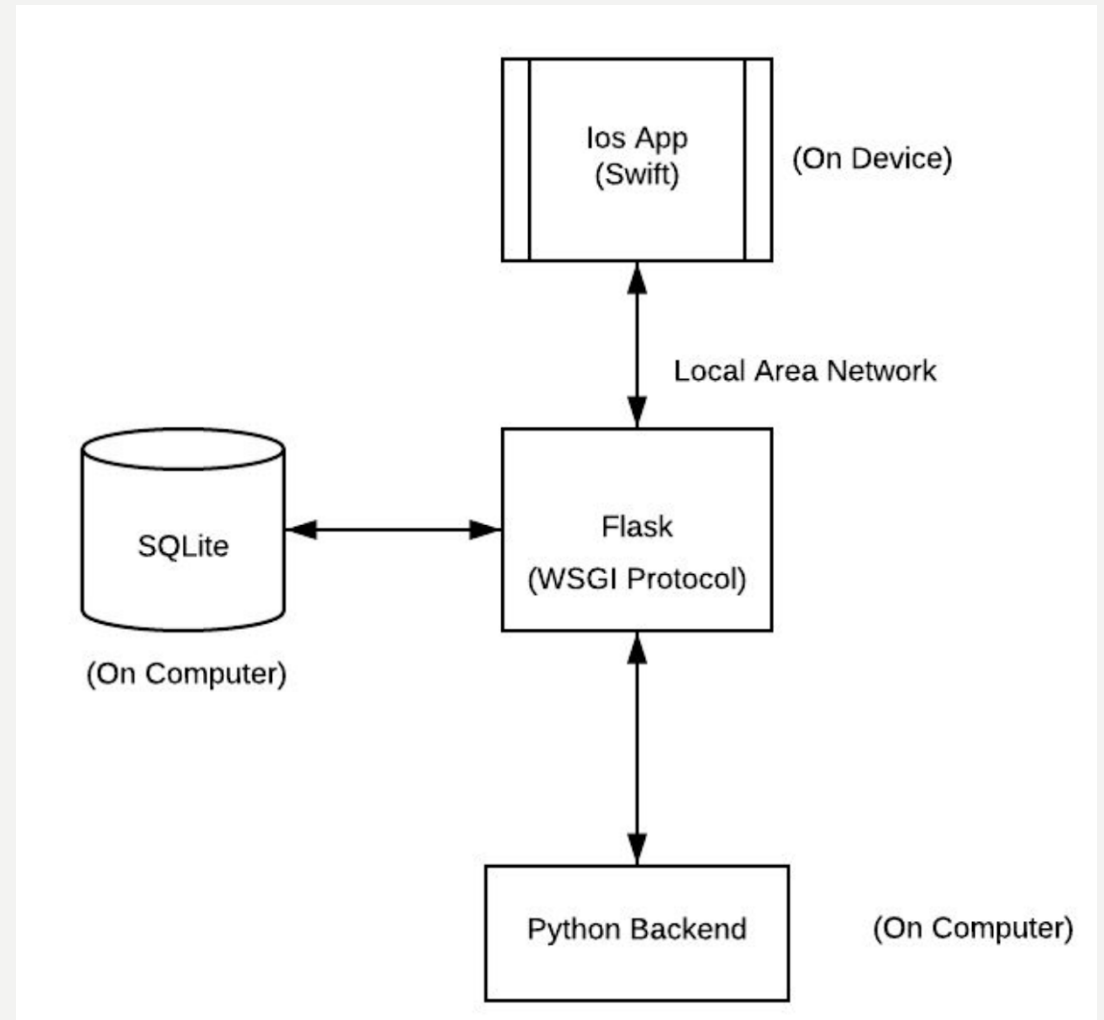
# WEB FRAMEWORK

## Flask Web Framework

**Purpose:** Used for communication between frontend and backend using endpoints (REST Requests)

**Rating:**☆☆☆☆☆ (5/5 stars) - because communicating the frontend and backend was really simple using POST and GET requests.

Endpoint testing one with **Postman**



# CHALLENGES

Challenges you encountered, and how you overcame them and how it may have affected your original project plans.

- Learning Swift was difficult to follow and took a lot of time
- Detecting coin circles accurately in different images. Had to try different techniques and see what worked the best for the most number of images. Also had to set guidelines for good and bad images.
- Finding a good error bound to use when calculating which coins are quarters vs. nickels vs. dimes. This took many test cases to figure out with each test displaying the coins at different sizes in order to get an error percentage that was big enough to capture each coin of the same relative size, but also small enough so that there was no overlap.
- Finding the right tool for the database proved to be a bit difficult. Having almost no knowledge about app development and only the foundations of SQL, we had to try a few different things before we landed on SQLite
- Finding a good way to communicate Swift frontend and Python backend. Challenge solved by learning how to create an API with Flask and Python. Connecting to TAs was a big help to understanding how to communicate frontend and backend.



**TIME FOR A DEMO!**

**Any Questions?**