Authors: Sean Corcoan, Jack Wang, Greg Schultz, Jason Chen

IOS AppDev

2 December 2018

ChallengeMe Hack Challenge Final Submission

App Name: ChallengeMe

App Tagline: *We can challenge ourselves to do more, why not make it fun?*

Github links:

IOS: https://github.com/seancorc/ChallengeMe

Backend: https://github.com/gschultz49/challengeMe-backend

App Description:

ChallengeMe provides a daily challenge for the user which provides exposure to new experiences. According to research, people who try to complete daily challenges achieve a state of "flow", leading to greater happiness and a more fulfilling life. This app aims to help people get outside their comfort zone, grow, and make life simply more fulfilling by performing a daily novel challenge.

This app holds a wide array of experiences, from learning the Heimlich Maneuver to performing a perfect pushup. With a simple tap of a button, the app generates a random challenge and presents a timer which counts down as the user performs the task. When the challenge is complete, the user simply taps "Challenge Completed" and their experience is recorded. Users can sign in to keep a streak of challenges going, compete with friends to see who is the most adventurous, and see which challenges they have completed. With brilliant backend integration, a gorgeous user interface, and artistically chosen media gifs, this app yields a visually stunning, robust, and fun user experience.

In a fast-moving world, life can become monotonous and novel experiences can be hard to come by. With this app, people with busy schedules can enrich their lives and move out of their comfort zone, being challenged to learn, relax, exercise, experience, and grow daily.

Feature Highlights:

I. Sign in/ Login Options
II. Compete with friends in Users page
    a. See how many challenges they have completed
    b. Be surprised by their profile pictures
III. About Me Page
    a. Keep a streak
    b. See how many challenges you have completed
    c. Instead of your own face as a profile photo, enjoy a randomly generated face of an IOS instructor
IV. ChallengeMarket Page
    a. View all the possible challenges

   b. Enjoy small media gifs visually representing each challenge
   c. View the time allotted for each

V. Challenge Page
   a. Present a random challenge when you open the app
   b. Enjoy a luxurious media gif in the background
   c. Right swipe to reject a challenge (Tinder vibes)
   d. If press accept, timer will begin to count down and "Challenge Completed" button will appear
   e. If press Challenge Completed, a modally presented screen appears, congratulating you for making your life a little more awesome

IOS requirements:

1. AutoLayout
   - Requirements for Autolayout were addressed. We used NSLayoutConstraints to organize the placement of text and images in each view.
   - For challengeview, the NSLayoutConstraints were used to layout the challenge descriptions, times, gifs, etc
   - For the user views, the NSLayoutConstraints were used to layout the profile pictures, names, etc.
2. CollectionView/ TableView
   - Requirements for CollectionView were addressed. We used CollectionView to create the views for each of the challenges. Each challenge had an individual CollectionView cell which filled up the entire screen.
   - TableView was also used in the MarketPlace and Friends tab to showcase completed challenges and people that the user is competing against

- Navigation
   - Requirements for navigation were addressed. A UITabBarController was used to present each of the following tabs: Challenges, MarketPlace, Friends, and About Me.
   - In addition, UINavigationController was used to present the Congratulations page after the challenge is completed

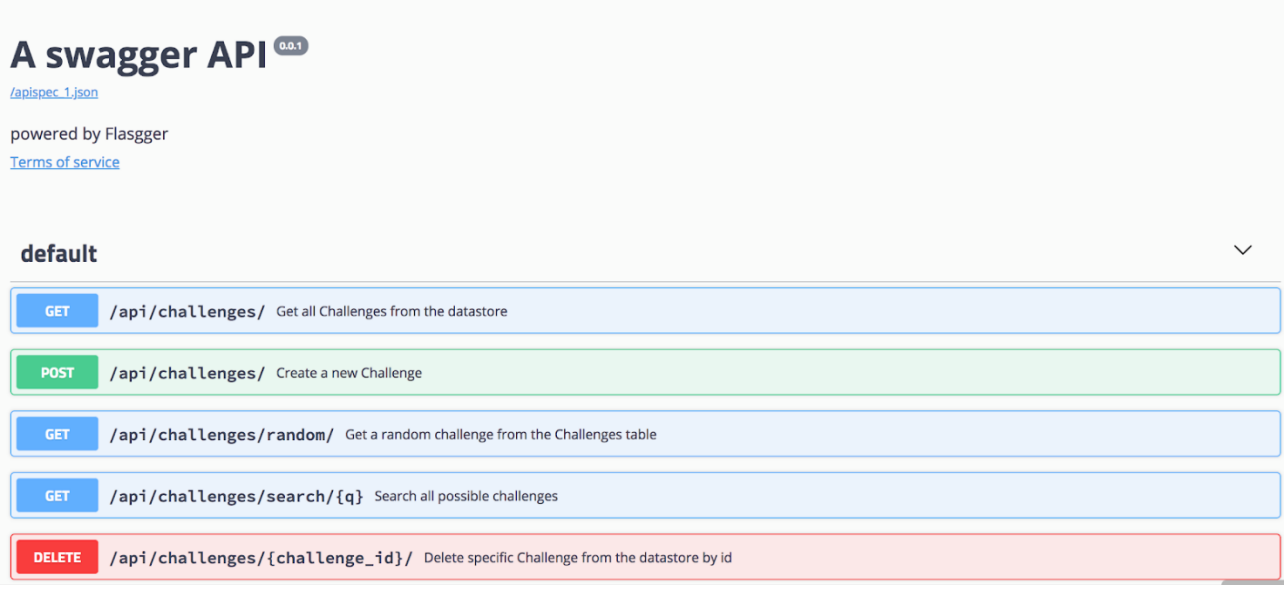
3. API
   - Requirements for API were addressed. A main API known as Swagger API was used in importing the challenges. This bodacious API included the challenge description, challenge time, challenge gif, usernames, user identities, user passwords, and much more.
   - For the challenge pages, this API was used for importing the challenge names, gifs, and times
   - For the sign in screen, the API was used to store and present the user information
   - For the user pages, the API was used to present the user photo, name, etc.

Backend Requirements:

The challengeMe backend api is original. It utilizes 3 main models: Challenges, Users, and Completions. The challenges are the actual challenges that users can complete, the users are the records of the signed in users, and the completions are the logs of which user is completing which challenge (through use of foreign keys). It also features the usage of the GIPHY api, as a means of generating fun gifs for the challenges. Finally, the app is fully deployed to Google Cloud.

To make the api as digestible as possible, the api uses the Swagger UI (see http://35.243.148.8/apidocs/). This makes it much easier to explain and interact with the api.

Backend Screenshot:



Frontend Screenshots: