**Is My Family Okay?**

Ioannis Kakadiaris, Pavan Garidipuri

*1Computational Biomedicine Lab, University of Houston, 4800 Calhoun Road, 77004 Houston, Texas*

**Abstract**

This document discusses the implementation of the “Is My Family Okay?” App, a concept proposed by Dr. Kakadiaris. The app provides a platform for users to semi-automatically notify family and friends of their safety during natural disasters.

**Introduction**

Have you ever been in a situation during a natural disaster where you wanted to check if your family or friends were safe, but you were unable to reach them? This scenario occurs around the world during any natural disaster. In 2017 natural disasters displaced some 30.6 million people, with more than four-fifths of those being in Asia, according to a new report [1]. Families and friends lose contact with each other during evacuation and migration.

Currently, the American Red Cross maintains Safe and Well, a web-based system that helps reunite friends and family displaced by a disaster. This system, however, requires users to report themselves after they reach safety during a disaster, which can be unfeasible given the circumstances of major natural destruction, such as weak internet connectivity or more important priorities. Also, the registry has no authentication proof, meaning other people can enter someone's name which is not confirmed to be safe and tamper with the registry. [2]

FEMA’s National Emergency Family Registry and Locator System (NEFRLS) helps reunite families separated during a disaster. It allows displaced individuals to register and provide information about their current location and situation. The system provides a secure web-based environment where survivors and up to seven individuals (their loved ones and/or others) specifically identified can communicate their location as well as provide a personalized message. A drawback of this system is that it only activated at the request of a State to support Presidentially-declared disasters. And it also faces the same constraint of reporting safety during a disaster. [2]

Therefore, there is a critical need for developing a semi-automated service to support incident response by harnessing the advantages of recent advances in decentralization, biometric identification, and facial recognition in a way that respects individuals’ privacy. CBL has developed a prototype service that provides awareness of individuals’ location to both their family and to emergency personnel who need to make decisions, all in a person-centered privacy-preserving manner. This service will be mobile-focused and expand into a website in later versions. This research will advance the state-of-the-art in facial recognition, privacy-preserving techniques, and blockchains, and its development will be informed by extensive interviews with hurricane victims, emergency personnel, and decision makers

The “Is My Family Okay” app provides functionality to locate registered family and friends in non-governmental organization (NGO) shelters during natural disasters. A user can register pictures of his or her face onto the app. The app then stores the data into the system. In the case of a natural disaster, The NGO shelters have cameras which detect if a person in the shelter is registered on the app. If the person has registered followers (i.e. family and friends) on the app, they can be notified of the person’s safety on the shelter.

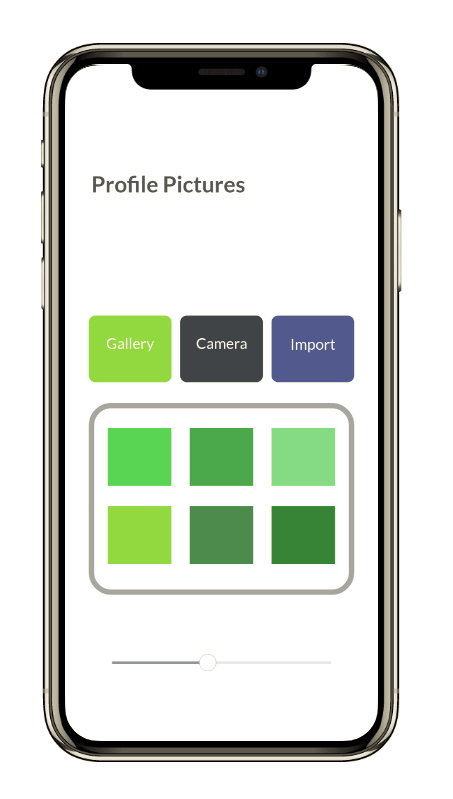
This write-up provides an overview of the application’s features, front-end technologies, and backend protocols. The format is

1. User App Front End
2. NGO Front End
3. Backend Cloud Services

**1. User Front End**

1. User authentication and Registration: We want to make sure a user’s profile does not get into the wrong hands. If that were to happen, other user’s private location data would be in the hands of an unauthorized person. Thus, the most vulnerable part of the application is user authentication. If you share your data with a follower, we want to make sure no one else is accessing the follower’s application.

* We will use two-step authentication with email and face-verification.
* During the registration, the user sets up email and face data. The registration begins with a simple signup page.

The user has two options: sign up with email or connect with Facebook.

1. Once the user clicks on either option, the second set up screen uses the email or Facebook account to fill in data about a name, date of birth, and residence.
2. If this information is not provided in the account, the user can add the information manually on the page.
3. The next screen (Figure 1) allows users to set up their face profiles: an array of pictures for the app to use to locate him or her. The user can import these pictures from their gallery, camera, or import from Facebook.
4. After clicking “Finish Set Up,” the user is greeted with the Homepage (Figure 2), which has three buttons: “Register Followers,” “Check on Followings,” and “Mark Yourself Safe.” Also, at the top right, the “profile” button gives access to her or her profile. Each button is explained in “2. App Features.”

Figure 1: When registering, users can upload pictures of themselves through Facebook, phone gallery, or selfies through the app. These pictures are used to train the app to recognize the user in other pictures.

* The user’s email and face data are stored on the server. After 1 week of inactivity, the app logs the user’s account out. During re-login for the app, the user needs to input email, password, and take a selfie to verify that it is the user.
* For a more secure authentication process, we should use legally issues ID numbers to verify users. This would entail further review about legal procedures regarding ID collection and storage

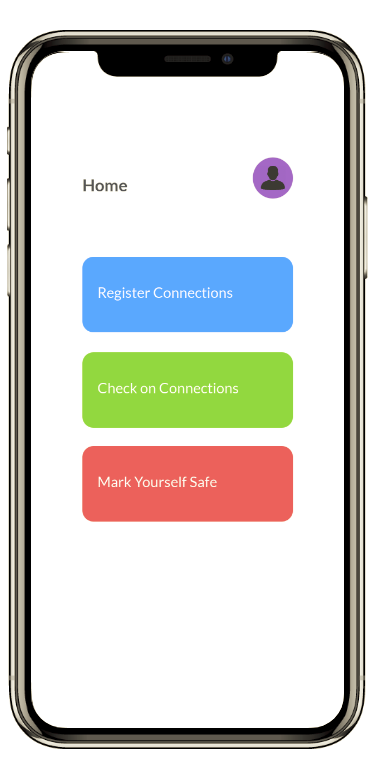
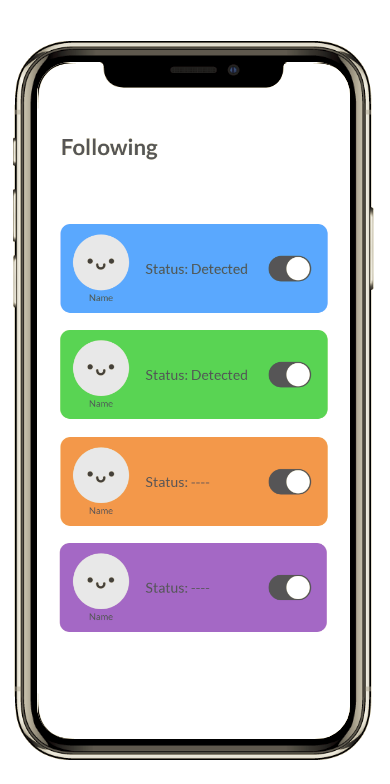
1. App Features: The app has three key features 1) Register Followers, 2) Check on Followings, and 3) Mark Yourself Safe. These features are described in detail below
2. “Register Followers”: To give friends or family access to the user’s data, the user needs to register the friends and family as a “Follower.” To do this, the user will follow these steps:
3. Click on the “Register Followers” button.
4. The user then clicks “Allow to Import Contacts,” which allows the app to match the user’s contacts with existing profiles on the app. All the contacts are displayed on the screen, and the contacts that are on the page are highlighted.
5. The user can click on the highlighted contacts he or she wants to be able to locate him or her in the case of an emergency.
6. If the contacts he or she wants do not use the app at that moment, the user can still select those contacts, and they will receive a text notifying them about the app and how they can join the app. The Amazon Simple Notification Service (SNS) will send an automated text message to that specific contact.
7. The user can select what data can and cannot be shared to each follower. The options include
   * + Mark Yourself Safe update
       - Described in “C. Mark Yourself Safe”

Figure 2: The app has an intuitive homepage with three buttons: “Register Connections”, “Check on Followings”, and “Mark Yourself Safe.”

* + - Face Detection and Image display update
      * Shows the image in which the user was located to the followers
    - Image Location update
      * Provided the metadata location of the image to the flowers

[In version 2, we will implement a “Groups” feature to categorize followers into custom groups and allow for easy control of data shared with a group.]

1. “Check on Followings”: A list of other users whom the user is following.

* In the connections tab (Figure 3) the user has a full list of other users whom he or she is following with statuses. These statuses indicate whether the following profile has been detected at an NGO shelter.
  + If the user is detected at the shelter, the status says “detected.”
  + If not, the status says, “not detected.”
    - * If a family member is shown as “not detected,” it means the sources have not located them, and he or she is likely away from the area of danger.
* **** The user can select the following user’s profile to see comprehensive details about data received about the following user (Figure 4) when they click on a profile. This includes
  + Mark Yourself Safe update
    - * Described in “C. Mark Yourself Safe”
  + Face Detection and Image display update
    - * Shows the image in which the user was located to the followers
  + Image Location update
    - * Provided the metadata location of the image to the flowers

1. “Mark Yourself Safe”: If the user is in the midst of a disaster and safely evacuated, he or she can go to the mark yourself safe tab.

* Here, he or she can select your status to say “Safe” and can share life-location data to your followers.

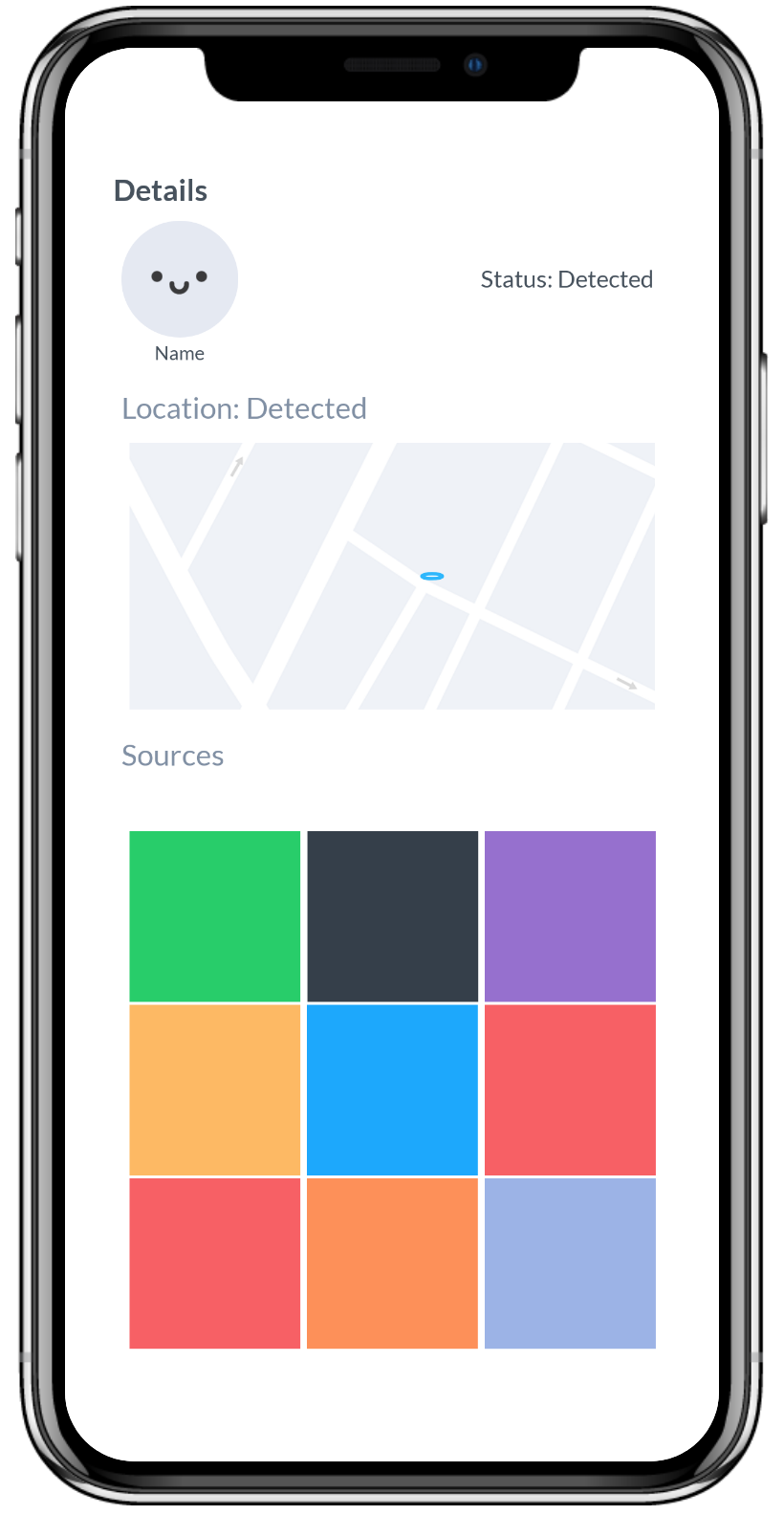
Figure 3: The “Check on Following” page shows a list of users who have shared data access with you. A short summary of the users’ statuses is provided.

* + This allows the followers to get live updates on your location based on the user’s phone location.
* The “Safe” status is temporary for up to 12 hours, meaning after that time, the status will revert to “unknown.”

**2. NGO Front End**

The detection of registered app users in shelter is the crux of the app. Therefore, we created a streamlined solution for NGOs to feed data to the backend cloud system

* NGOs will be provided with four webcam-enabled cameras which are wirelessly connected to a central computer or laptop. This computer will have access to the app’s Amazon Web Services (AWS) server.
* The webcams are placed in shelter common areas.
* Steps for data uploading to system’s AWS server

1. The webcams can then live stream to the computer.
2. The computer will separate the videos into frame images and upload the images to the app’s AWS server through the NGO’s channel.
3. The AWS server will attempt to detect a person in the frame images using the Amazon Kinesis program, used for real-time video and image analysis such as face recognition.
4. The app will notify the person’s followers on the app of his or her safety in the shelter, along with location and contact information of the shelter.

**3. Backend**

* Authentication: The app has a secure 2-layer process for user authentication

For account registration and authentication, we will be using Email/Facebook and Facial Authentication [Formal IDs in the future]

Figure 4: Each user you are following has a profile page with information about their safety.

* 1. Email - For email login, we will use the Django-registration-redux framework in python, which is compatible with React Native
  2. Facebook - For Facebook log-in option, we will use the Facebook Log-in framework, which allows simple user login along with user data such as name and birthday for the app.
  3. Facial Authentication - After using email or Facebook to log in, the user will be prompted to take a selfie in the app to verify the user. We will be using the same FaceCollection verification process as described in “2. Face Detection,” but we will only be attempting to match an image with one FaceCollection. This method will be efficient, fast, and increases security for users on the app.
  4. Formal IDs - In the future, we can implement a third verification process to ensure security
* Face detection: The app’s backend is able to detect and recognize people in a photo from the NGO
  + When a user enters an array of images of themselves, the application generates a “FaceCollection.”
  + The FaceCollection is an array of face images used when matching unknown faces to a person’s identity.
    - These FaceCollections are stored into the App’s distributed data storage system (as described in “Blockchain Data Storage” below).
    - The AWS system is configured to store the FaceCollection data from the distributed storage system during times of disasters when matching is needed. Otherwise, the data is not on the Amazon’s servers.
  + We also create a photo-uploading channel for the NGO webcams in shelters to submit pictures to the App server to identify the unknown faces against the FaceCollection.
    - Once a user who is registered on the app is matched within Amazon Rekognition, AWS’s surveillance and facial recognition software, the app notifies the user’s followers adds the picture to the user’s app profile in the “Check on Followings” tab.
    - All the detected pictures appear on the profile details page for followers to see.
    - If a picture has been mistakenly tagged, followers can flag as “inaccurate.” This feedback will be sent to the face-recognition software to help it become more accurate.
* Data storage: Users might be hesitant to upload their data onto a centralized server due to lack of privacy and potential for misuse of surveillance. Thus, our top priority is to make sure user data is stored securely and within a tamper-proof system.
  + There are also several legal concerns storing personal data (face images) on a centralized app server:

1. Electronic discovery - Police requiring personal data to be given to them
2. Privacy - Protecting the confidentiality of private information
3. Evidence Management and Forensics - Administration, management, and control of information and data related to an event so that it can be used to prove the circumstance of an event
4. Addressing accountability and responsibility issues
5. Factoring in the geopolitical boundaries and the jurisdiction implications
   * Solution: Blockchain will provide strong security of user data by
     1. Breaking up data into chunks called “data blocks”.
     2. Encrypting the data so that the user is the only one with access to it.
     3. Distributing files across a network in a way that means all of the user data is available, even if part of the network is down.
   * Distributed user data cannot be stolen by anyone and does not provide the aforementioned legal issues with data storage on a central app server.
   * We will store FaceCollection data and NGO photo data on a distributed database.
     + - Storij.io and Sia.tech are two decentralized data storage companies available.
   * The database would be instructed to send NGO photo data to the AWS Cloud to test against FaceCollection data.
   * By using blockchain, we get more security, privacy, along with efficiency due to its ability to scale
   * Storage to Server Protocol
     1. When a user registers their FaceCollection, the FaceCollection is sent to the blockchain storage system.
     2. Each FaceCollection is paired with an ID number, which is then sent to Amazon servers.
     3. When there is a match between an NGO photo a FaceCollection, the ID number is sent to the blockchain storage as a PUSH request, which allows the storage system to find the user profile and edit the profile status.
   * Uploading to a decentralized storage system like Storij.io is very similar to the PUSH and FETCH requests from Web Servers, so we will be using existing PUSH and FETCH protocols to use the storage system.

We are using AWS over services like Microsoft Azure because of Amazon Rekognition, which can run real-time analysis on video from Amazon Kinesis Video Streams or analyze images as they are uploaded to Amazon S3. Real-time analysis can prove beneficial in circumstances where the NGO image data comes from a live-surveillance camera instead of manual photographs.

**Proposed Timeline**

Week (July 23-27) – Completed Registration and Log In portion of the app. Completed the UI for “Check on Followings” page and their respective profiles for the users in the page. Hosted user data on a centralized json-server

Week (July 30-August 3) -

Week (August 6-10) -

Week (August 13-17) -

Week (August 20-24) -

**References**

[1] “IDMC | Global Report on Internal Displacement 2018.” *IDMC*,

internal-displacement.org/global-report/grid2018/.

[2] “Relief Organizations for Natural Disasters.” NAIS - National Association of Independent

Schools, School & Student Services, www.nais.org/articles/pages/relief-organizations-for-natural-disasters-150850.aspx.

(numbers inside “[ ]” represent days required for each Task