People Movement

Introduction

People Movement is an Android Application on the front end that uses Firebase and Google Cloud Platform in the backend for its computations and queries. The front end work will be done by Shaked. She will be creating the interface for the Android app. The back end work will be done by Kate. She will be creating the interface that communicates with Firebase and Google Cloud Platform.

Backend

Firebase Integration

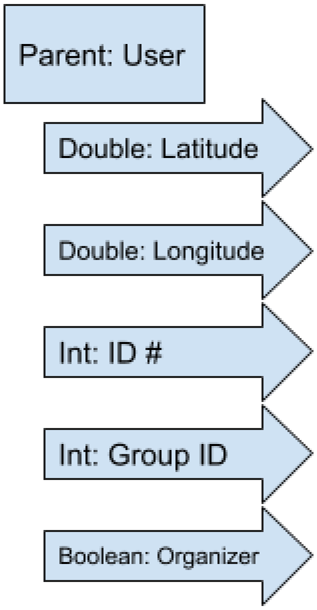
Using Firebase

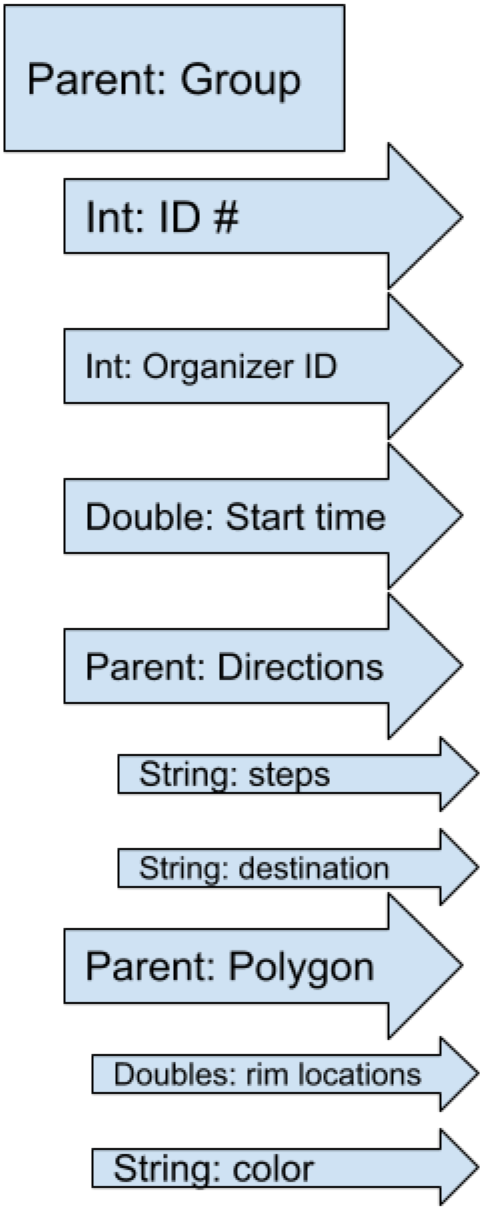
Obtain permissions and login - contact ktmcnally3@gmail.com to get added to the project
Installing SDK to write code in local command line and for writing Cloud Functions - https://youtu.be/EvV9Vk9iOCQ

Authentication

Sign Up -
Through Facebook
Through Email
Through Phone

Database





Cloud Functions

Function Name: getGroupInfo()
Operation: Query the database for group information
Return: Start time, directions
Parameters: group ID #

Function Name: setGroup()
Operation: Query the database for groups and add user to that group. Sets the user’s group ID # to group ID # of corresponding group and add user to list of users within specific group. This function will also allow users to view group information.

Return: Message to confirm group join
Parameters: group ID # and user ID #

Function Name: updateGroupInfo()
Operation: When organizer wants to update any aspect of the group, function sees which data is being updated, and updates that data in database. Pushes that data to the group user's to update them of changes.
Return: Message to confirm updates
Parameters: Group ID #, new information

Function Name: startJourney()
Operation: Sets journey start time to current time and pushes that information to all group members
Return: Message to notify users in group that journey has begun
Parameters: group ID #

Function Name: leaveGroup()
Operation: Used for when a user wants to leave the group. Removes the group ID # from the user's data and then sets it to null. Removes the user's ID # from the list of users in the group
Return: Message to user that they have left the group
Parameters: User ID #, Group ID #

Function Name: updateDirections()
Operation: Used for when the group needs to be rerouted due to a list of circumstances (user feedback, traffic). Sends group information to App Engine, waits for computation, receives new directions, and then pushes directions to group and updates organizer.
Return: New directions
Parameters: Group ID #, Organizer ID

Google Cloud Platform Integration

App Engine

Installing SDK and getting Google App Engine set up - https://cloud.google.com/appengine/docs/

- Choose a language
- Go through step-by-step guide to set up SDK
- Choose us-east4 for location

Update Directions Algorithm

Overview: This algorithm takes in group information and reason for redirection and using that information to give the group a better set of directions to get from point A to point B. If the reason is road traffic, the algorithm tries to reroute to roads that experience less car traffic. If the reason is violent outbreak, overcrowding, or anything else related to issues in the movement of people, then the algorithm tries to do a general reroute without many requirements.

Functions:

Name: Reason()
Operation: Look at reason that directions need to be rerouted
Return: Reason 1 (car traffic), Reason 2 (people crowding)
Parameters: n/a

Name: locateLowTrafficArea()
Operation: Use Google Maps API to find general low traffic areas
Return: Radius of low traffic areas
Parameters: Radius of desired locations

Name: potentialReroutes()
Operation: Create and populate a list of potential sets of directions based on locateLowTrafficAreas() return
Return: List of directions
Parameters: n/a

Name: rankingReroutes()
Operation: Loop through list of potential reroutes and set a rank for each. Then order the reroutes from best to worst based on crowd number, destination, and predicted traffic levels for the area.
Return: Ordered list of reroutes
Parameters: List of directions from potentialReroutes() function

Name: decideOnReroute()
Operation: Look at reroute list and save if need to recalculate
Return: Number 1 reroute option
Parameters: Ordered list of reroutes from rankingReroutes() function

Front-End App Development

Object Classes

LocationObject Class

Create a location object with the following fields: latitude and longitude
Class includes following methods:

- Name: LocationObject(double latitude, double longitude)
  Operation: constructor to set latitude and longitude of the location object
  Return: N/A
  Parameters: double latitude, double longitude

- Name: getLat()
  Operation: get latitude
  Return: double latitude
  Parameters: N/A

- Name: getLon()
  Operation: get longitude
  Return: double logitude
  Parameters: N/A

- Name: setLat(double lat)
  Operation: set latitude
  Return: N/A
  Parameters: double lat

- Name: setLon(double lon)
  Operation: set longitude
  Return: N/A
  Parameters: double lon

Event Class
Create an event object with the following fields:

- String name, String time, LatLng startLocation, LatLng endLocation

Class includes following methods:

- Name: setName(String name)
  Operation: set the group's name
  Return: N/A
  Parameters:String name

- Name: setTime(String time)
  Operation: set the event's time
  Return: N/A
  Parameters:String time

- Name: setStartLocation(LatLng location)
  Operation: set the event's start location
  Return: N/A
  Parameters:LatLng location

- Name: setEndLocation(LatLng location)
  Operation: set the event's end location
  Return: N/A
  Parameters:LatLng location

- Name: addMember(User user)
  Operation: add a user to a group

Return: N/A
Parameters:User user

- Name: getName()
  Operation: get the group's name
  Return: String name
  Parameters:N/A

- Name: getTime()
  Operation: get the event's time
  Return: String time
  Parameters:N/A

- Name: getStartLocation()
  Operation: get the event's start location
  Return: LatLng location
  Parameters:N/A

- Name: getEndLocation()
  Operation: get the event's end location
  Return:LatLng location
  Parameters:N/A

- Name: getMembers()
  Operation: get the members of the group
  Return: ArrayList<User> members
  Parameters:N/A

User Class

Create user object with the following fields:

- String name, ArrayList<Event> events, LocationObject location
  Class includes following methods:

- Name: setName(String name)
  Operation: set the users's name
  Return: N/A
  Parameters:String name

- Name: addEvents(Event event)
  Operation: add an event to the user's events list
  Return: N/A
  Parameters:Event event

- Name: setLocation(LatLng location)
  Operation: set the user's location
  Return: N/A
  Parameters:LatLng location

- Name: getName()
  Operation: get the users's name
  Return: String name
  Parameters:N/A

- Name: getEvents()
  Operation: add an event to the user's events list
  Return: ArrayList<Event> events
  Parameters:N/A

- Name: getLocation(LatLng location)
  Operation: get the user's location
  Return: LatLng location
- Parameters:N/A

Activities

MainActivity Class

- Home page of the app, set up of UI to show a burger menu with the options of: Map View, Create Group, Join a Group, Group Chat

MapsActivity Class

The activity showing the map includes the following methods:

- Name: onMapReady(GoogleMap mapReady)
  Operation: send a request for location updates when map is ready
  Return: N/A
  Parameters: GoogleMap mapReady

- Name: setUpMap()
  Operation: Set map type and UI
  Return: N/A
  Parameters: N/A

- Name: startLocationUpdates(**final** GoogleMap mapReady)
  Operation: Request location updates at a chosen interval
  Return: N/A
  Parameters: GoogleMap mapReady

- Name: onLocationChanged(Location location, GoogleMap mapReady)

Operation: Detect when location is changed
Return: N/A
Parameters: Location location, GoogleMap mapReady

- Name: getDirectionsUrl(LatLng origin,LatLng dest)
  Operation: Send a http request to Google Maps API fort a walking route between two locations
  Return: String url
  Parameters: LatLng origin,LatLng dest

- Name: downloadUrl(String strUrl)
  Operation: Download JSON data from URL
  Return: String data
  Parameters: String strUrl

- Name: getGroupLocations()
  Operation: Get an array of locations of group members
  Return: ArrayList<LatLng> locations
  Parameters:N/A

- Name: showLocationsOnMap(ArrayList<LatLng> locations)
  Operation: Present the group's locations on map
  Return: N/A
  Parameters:ArrayList<LatLng> locations

- Name: showCrowds()
  Operation: Calculate crowds and present them on map using polygons
  Return: N/A
  Parameters:N/A

DirectionsJSONParser Class

- Receives a JSONObject of directions between two locations and returns a list of lists containing latitude and longitude

CreateGroupActivity Class

An activity with a UI to create a group

- Name: createGroup()
  Operation: allow users to input a group name, time, start and end location in order to create a group
  Return:Event event
  Parameters:N/A

- Name: addToDatabase(Event event)
  Operation: add event to database
  Return:LatLng location
  Parameters:Event event

SearchGroupActivity Class

An activity with a UI to search for groups in the group database

- Name: searchGroup(String event)
  Operation: search for a group event in database
  Return:Event event
  Parameters:String event

APIs

Google Maps API

- API Key: *Obtain google maps API key* - contact shakedrad2@gwmail.gwu.edu to receive the API key
- Show a map fragment

  -
    ```
    MapFragment mapFragment = (MapFragment) getFragmentManager()
      .findFragmentById(R.id.map);
    ```

- Send HTTP request for directions

  -
    ```
    String url =
    "https://maps.googleapis.com/maps/api/directions/json?str_origin+"&"+str_dest+"&"+sensor;
    String url = getDirectionsUrl(origin, dest);
    DownloadTask downloadTask = new DownloadTask();
    downloadTask.execute(url);
    ```

- Draw a route using polylines

  -
    ```
    lineOptions = new PolylineOptions();
    map.addPolyline(lineOptions);
    ```

- Show crowds using polygons

  -
    ```
    Polygon polygon = map.addPolygon(new PolygonOptions())
    ```