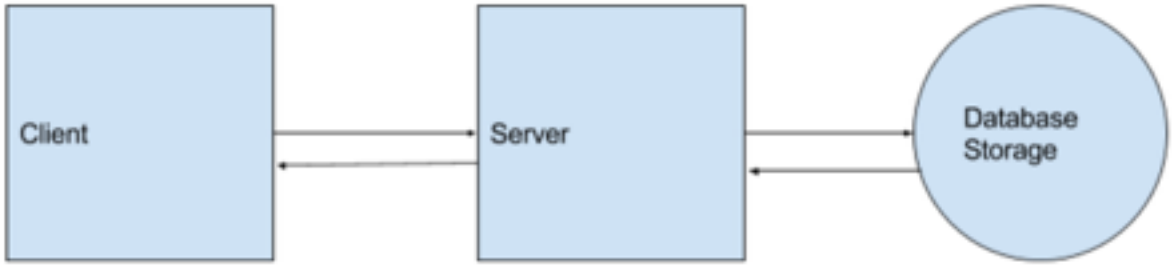
ECS160
Waikiki Design Document

Team BHE
Bowen Lin
Dong Wook Kim
Hao Chen

1.Brief summary of 5 use cases

1.Architecture Style:
We used the Three-Tier Systems Architecture, since there are constantly data transfer and request from client and server.



2. Object-oriented design pattern:
We using the Decorator Pattern for comment and ratting, using the observer pattern for the walking_finish_detect. and using the composite pattern for page design.

Use case:
1.CREATE A WALK AND TAKE IT.

We will query Google maps and use GPS service to get map and user's current location for map-involving features. During the walk, the client can update map with taking path.
We will use IOS location service for set current location (it's in real time during walking)
We will use IOS camera service.
We will use IOS system clock to record the time for the walk.

2.Post/Share a walk
We will use facebook service to post and share walking information including pictures, comment, rating user has inputted at the end of walking.

We will use own post server to sent the walk to the database.

3.Take a walk by searching for it (keyword search).
We will use own search server to get walks based on the keyword.
We will use google map to take the walk in real time.

4.Search for nearby walks.
We will use ios location(GPS) service to locate user's current location, then use google maps to search and show nearby walks.

5. Upload media content to a walk
We will allow user to use their built-in camera to take picture and upload during / after walk.
We will allow user to upload picture later on walk_history menu by using built-in photo gallery

External API USE

IOS GPS:

Document: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/LocationAwarenessPG/CoreLocation/CoreLocation.html

```
- (void)locationManager:(CLLocationManager *)manager
      didUpdateLocations:(NSArray *)locations {
   // Add the new locations to the hike
   [self.hike addLocations:locations];

   // Defer updates until the user hikes a certain distance
   // or when a certain amount of time has passed.
   if (!self.deferringUpdates) {
      CLLocationDistance distance = self.hike.goal - self.hike.distance;
      NSTimeInterval time = [self.nextAudible timeIntervalSinceNow];
      [locationManager allowDeferredLocationUpdatesUntilTraveled:distance
                                                         timeout:time];
   self.deferringUpdates = YES;
   }
}
```

Google Map:

https://developers.google.com/maps/documentation/ios-sdk/

```
- (void)viewDidLoad {
  [super viewDidLoad];
```

```
  GMSCameraPosition *camera = [GMSCameraPosition cameraWithLatitude:-33.868
                                                          longitude:151.2086
                                                               zoom:6];
  GMSMapView *mapView = [GMSMapView mapWithFrame:CGRectZero
camera:camera];

  GMSMarker *marker = [[GMSMarker alloc] init];
```

FACEBOOK:

https://developers.facebook.com/docs/sharing/ios

Share Button
With the Share Button you will allow people to share content to their Facebook timeline, to a friend's timline or in a group. The Share button will call a Share dialog. To add a Share button to your view add the following code snippet to your view:

```
FBSDKShareButton *button = [[FBSDKShareButton alloc] init];
button.shareContent = content;
[self.view addSubview:button];
```

2. Detailed Use case: Create Walk

API Description:

We will query Google Maps and use GPS service to get map and user's current location for map-involving features.
We will use Facebook service to post and share walking information including pictures, comment, rating user inputted at the end of walking.
We will use Facebook service to get list of user's friends.
GMSMapView Class Reference: https://developers.google.com/maps/documentation/ios-sdk/reference/interface_g_m_s_map_view

Detail of classes is in UMLdiagram.png
Create Walk: Create a walk and saving it

1. Clicking Create Walk from main screen will lead user to Walk_create_page.

2. This page contains map with two options. We will query google maps and GPS service to get map. One is set current point and the other one is set destination. Setting current point won't be able to click if GPS locates user's current location, if GPS couldn't locate where user is, user needs to manually set his/her current location. The other option is setting the start node destination node which user needs to set. The reason we chose Google map is we believe most people have experience using google map before; hence it will be user friendly and easy to use.

3. Once user has set two points and click Start Button, this will lead to Walk_taking_page. This screen is also using Google map and GPS. While user is walking, user is able to upload picture by clicking upload picture button. Clicking Upload_button will redirect user to PicturePost screen. User has two options. One is take picture with built-in camera and upload, and the other one is select from album.

4. Once user has uploaded a picture, it will redirect them back to walking screen, and the picture will be put into the walk to the data_manager.

5. Once user has finished walking and click Done button, it will lead user to Walk_finish_page.

6.This Walk_finish_page screen has rating, comment, post, share features. Rating that user has rated will saved into database, show what user inputted on his/her walking history, and average of all users' rating on search walk/route feature.
- Once user clicks post, it opens Confirmation screen to verify user and show options user can chose. In this screen, we will query Facebook so that user can post their finished_walk information on their Facebook. We believe that Facebook would be most appropriate place where user can upload and many people around him/her can see the status. Google+ circle and twitter might be alternative.
- Once user clicks share, it opens Confirmation screen to verify user and show options user can chose. In this screen, we will query Facebook so that user can share their finished_walk information on their Facebook. We believe that Facebook would be most appropriate place where user can upload and many people around him/her can see the status. Google+ circle and twitter might be alternative.
- Once user clicks done, it redirects user to main screen and save walk to walking history screen with information they inputted during and after walk (picture, rating, comment).

3. Potential security issues, performance, failure risk

security issues

Major issue : How are we going to store data into database
-> potential security issue : what if cracker cracks into database and see user account information? (user id, password)
- solution : We never save plain text of password into database, nor reversible encryption password. We rather use hash and nonce(salt) to store user information into database securely.

Performance

Search route performance will be fast enough since we are going to user google map and gps, which are one of most famous tool in this case.
Getting access to friends list will be fast enough since we are going to use well-known external api (facebook).
We are planning to use hash table for search features / elements over other data structure because with this data structure, it will have fastest big-o time (O(1)) compare to other data structure.

We will create tables for each functions and use relational database on server-side in order to connect them to make functions work and manage data from thousands of users. Because, if thousands of users are accessing a single server to request various actions, it will slow down performance for those actions critically. Separating data and functions into different servers will allow users to request actions to different database which locates in different server. In this way, performance would be much faster than putting all tables and functions into one huge server.

There are different types of data. Some data will be data that user inputs and don't really use often, but some data will be used by users frequently. For this frequently used data, if we can implement cache in proper way, performance will be much faster because we can store the frequently using data into that page and use whenever user wants to access it rather than user request - go to server, database, get data, and do full cycle of function.

Failure risk

What if users do something that's not available in app?
> We will create a page that says "You have tried something that's not available in our system. Please go back and try something else" and redirect users to there whenever page returns thing that we don't have in our system.

We also have a feedback system, when the users find some bugs/problems, they can contact us by the feedback system.

4.Test plan for how to test your application.

We would like to test our application with 12 test cases which based on the 12 user cases.

Test Case:

1 Authenticating a user.

1.1 From the “log in page”, after input the account and password information, then push the “login” button. You should go to the “after_login page” if the account and the password info are correct. Otherwise, you should see an “error”.
1.2 If you don’t have an account, then just push the “Create an Account” button, and you should be in the “create_account” page.
1.3 When you are in the “create_account” page, you can input the “Account id” and set the “Password” (you need to re-type password in order to check it). Then, push the “create an account” button, you should then go back to the “log in page” and repeat the step 1.1.

2 Create a walk -> the end state would be starting the walk.
2.1 From the “after_login page”, push the “create walk” button, then you should go to the “create_walk page”.
2.2 From the “create_walk page”, push the “set current location” button to set current location and “set destination” button to set destination, after that, push the “start” button on the top right corner, then you should be in the “take walk page” which means that the walk you take is started.

3 Search for walks -> the end state would be starting the walk.
3.1 From the “after_login page”, there are two ways to search for walks
3.1.1 You can just type the key word in the “search bar” on the top of the page and press enter, and then you should go to the “search route” page and the result should be listed on the bottom of the scene.
3.1.2 You can also push the “search route” button and go to the “search route” page. Then you can type the route name or key words or click the tags on the scene. The results should be listed on the bottom of the scene.
3.2 While you see the result of the search, you can click it and then you should go to the “view route” page and see the walk information, such as the pictures, the rate, is this walk active, the points of interest and the comments.
3.3 From the “view route page”, push the “path detail” button, you should see the walk on the map, and then decide to take it or go back.
3.4 Push the “take” button on the top right corner to start the walk, and you should go to the “take walk page”.
3.5 From the “view route page”, if you push the “take walk” button, you should be the same as 3.4 directly.

4 View a specific walk (view the path, points of interest, comments, ratings, pictures).
4.1 From the “view route page”, you should see the rating, the route pictures, the comments and the points of interest
4.2 Push the “path detail” button, you should view the path.

5 Take the walk.

5.1 (Take Walk) from “take walk page”, you should see where you are on the map. If you push the “upload picture” button on the right of the scene, you should go to the “picture post page” and able to upload the picture you take.
5.2 (Take Walk Group) from “take walk group page”, you can upload pictures as 5.1, and if you push the “locate group member” button, you should see where all the group members are on the map.

6 Upload pictures for a walk (either during the walk or later from walk history).
6.1 (During the walk) from the “picture post page”, if you click “take picture”, then the application will open the camera to take picture. Then click “post” button to post/upload pictures.
6.2 (Later from walk history) from the “walk_history_detail page”, if you push the “upload picture” at the bottom, you should able to upload pictures.
6.3 After the pictures are posted, you can go to the walk history, and should see the pictures you just posted.

7 Rate a walk.
7.1 From the “finished_walking page”, you can rate the walk here.
7.2 After finished rating, push “done” button, then go to the walk history and find the walk you just rated, you should see the rating.

8 Post/Share a walk.
8.1 From the “finished_walking page”, push the “post” button / “share” button to post/ share the walk, then go back to search walk page, you should find the walk you just post/ share.

9 Verify uploaded media content before it is posted.
9.1 From the “Picture posting page”, you can take picture, then you can verify the picture.
9.2 If you don’t like the picture, push the “delete” button to delete and retake the picture, you should able to see the original picture is deleted.
9.3 Otherwise, just push the “post” button to post.

10 Tag walks (so they can be searched using these tags).
10.1 From “finished_walking” page, you can choose tags for the walk.
10.2 After that, go to “search route page”, click the tag and search, you should able to see the walk with tags that you just chosen.

11 Invite people to a walk/ join a particular walk.
Similar with 12.

12 Create an event so multiple people can join a walk.
12.1 From the “create event page”, first select route by “create route” button or “search route” button.

12.2 Then, click the friends whom you want to invite.
12.3 Set the event time.
12.4 Set privacy(everyone, friend only, self only)
12.5 Push the “create” button on top right corner to create the event.
12.6 You should able to see the created event in the event list.

13 Locate group members during a group walk.

Look at 5.2.