

# Clear Orbit

## Design Document

Edwin Pellot, Daniel Vega, Tyler Fung,  
Andrew Weatherby, Shafin Siraj, Vince Williams

10/8/19

Senior Project Fall 2019

---

## Table of Contents

<b>1. Overview</b>	
1.1 Landing Page/Home page	2
<b>2. Screens - Web Application</b>	3
2.1 Landing Page/Home page	3
2.2 Register	4
2.3 Login	5
2.4 Profile	6
2.5 Information View	6
2.6 Flow Diagram	7
<b>3. Screens - Mobile Application</b>	8
3.1 Login	8
3.2 Camera View	9
3.3 Connections View	9
3.4 Business Card View	10
3.5 Navigation	11
<b>4. Screens - Glass Application</b>	12
4.1 Opening the App	12
4.2 Picture Taking Screen	12
4.3 Information Screen	12
4.4 Glass User Flow Diagram	13
<b>5. Backend Systems</b>	
5.1 Amazon Web Services	14
5.2 Database	15
5.3 Restful Endpoints	16
<b>6. Team Contribution</b>	17
<b>7. Mid-Semester Status</b>	18

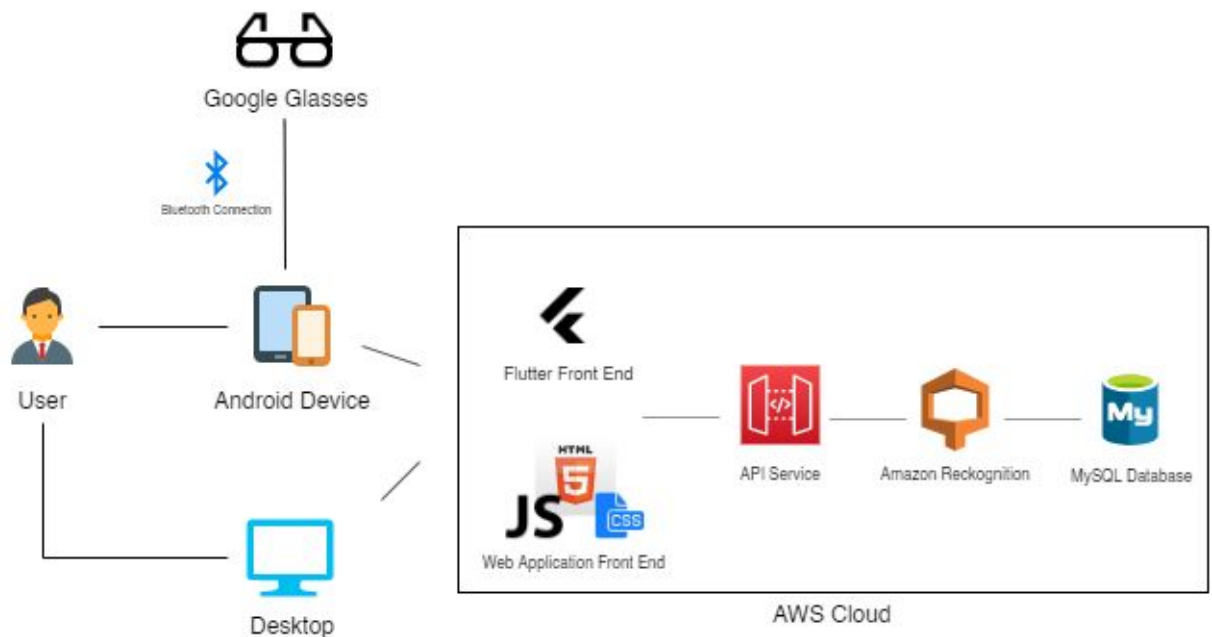
# 1. Overview

The purpose of this application is to provide people in a business setting with an easier way to pass contact information along to new connections. By using facial recognition technology Clear Orbit uses the user's face as a digital business card.

To be allowed to access the Clear Orbit network and to be recognized by other users, candidates must successfully create a user profile on the web application. The web interface allows users to provide/update information, and determine which pieces of information they would like to display. Once a profile is created, users can download the mobile application and login using their credentials.

When a user logs into the app, the mobile application will connect to Google Glass to take snapshots of the world around them. When a snapshot has been taken, the application will identify anyone in the picture who is a registered user. These registered users will be added to the users connections. Users can refer back to this list to retrieve contact information, making this information easily accessible.

## 1.1 Infrastructure Diagram

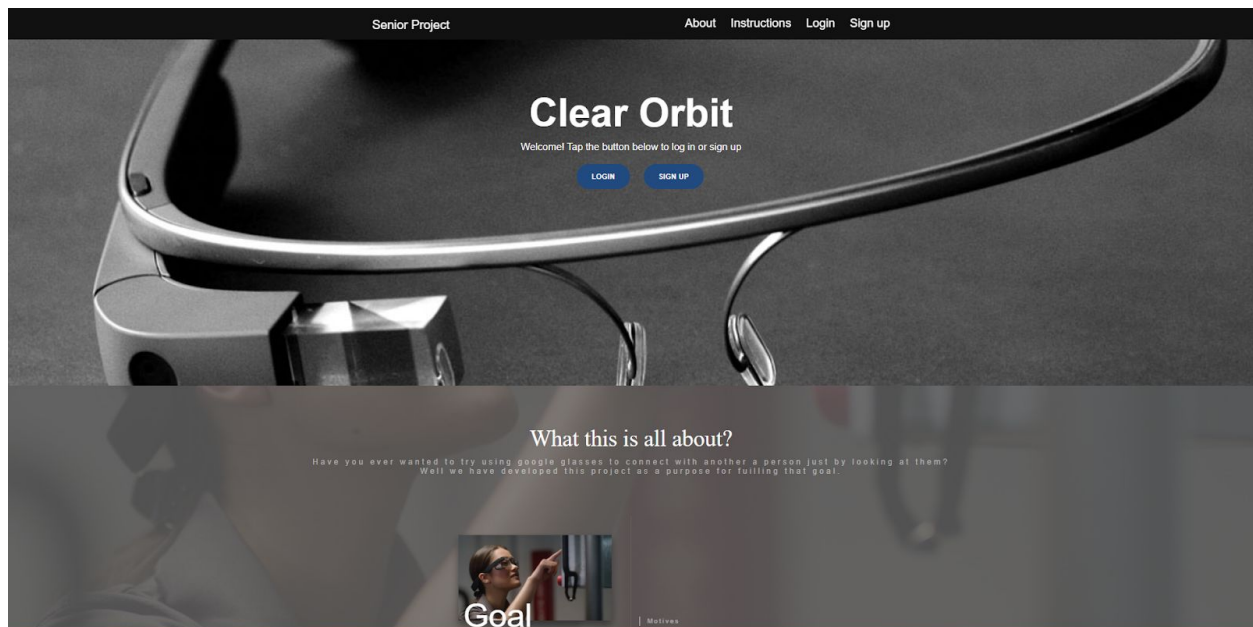


## 2. Screens - Web Application

The web application is where the user will create a Clear Orbit account, and become a registered user. The web application will only allow users to create, review, or edit a profile. The web application is composed of 4 pages: a landing page, register page, login page, and a profile page to add/edit personal information.

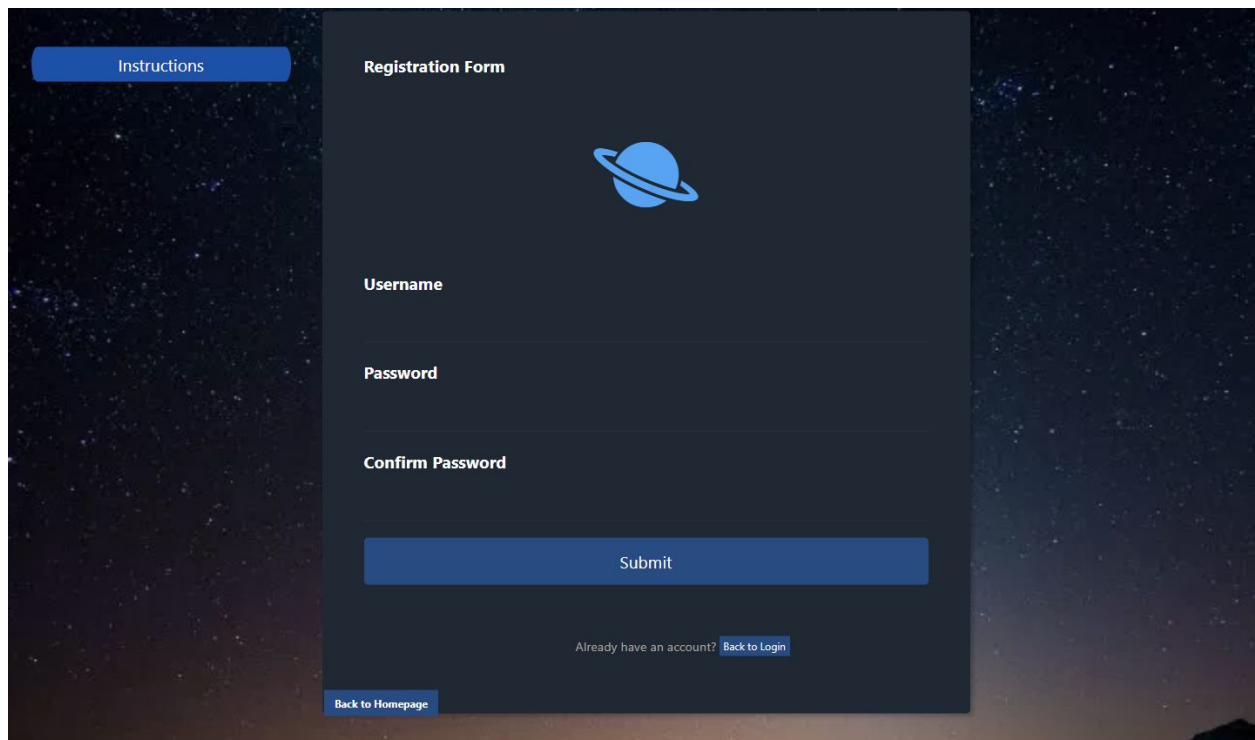
### 2.1 Landing/Home Page

The home page will provide the user the opportunity to learn about our project and it's status. It will also contain a navigation bar that links to the register and login page.



## 2.2 Register

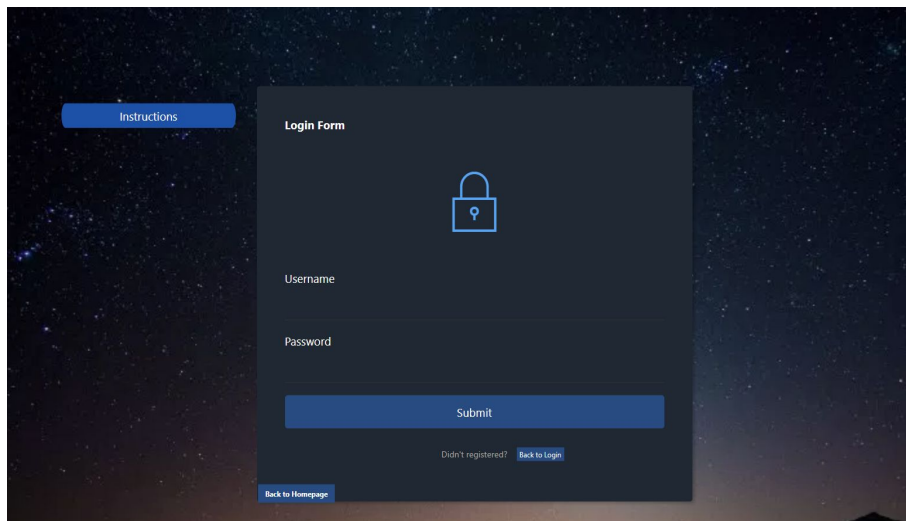
The register page will be responsible for creating the user's account. Creating the account requires a username and password. Duplicated usernames cannot be submitted for account creation and the user will be notified of this. When the user submission is successful, the user's data will be sent to a SQL database. Each user will have their own unique id in the database that will be used for the identification process within the mobile application.



The image shows a mobile application registration form. The background is a dark, starry space scene. On the left, there is a blue button labeled "Instructions". The main form area is dark gray and contains the following elements: a blue icon of a planet with rings (Saturn) at the top; the title "Registration Form" in white; three input fields labeled "Username", "Password", and "Confirm Password" in white; a large blue "Submit" button; a link "Already have an account? Back to Login" in white; and a blue button labeled "Back to Homepage" at the bottom left.

## 2.3 Login

The login page will be responsible for connecting the user to the profiling session. The login page is required to get to the profiling page. The requirements are the username and password which were already established in the register page. Incorrect entry will not allow access to the user's profile page. When inputted information is valid, the page will direct you to the profile page. The page also contains buttons that will lead you to navigate easily to the home and register page.

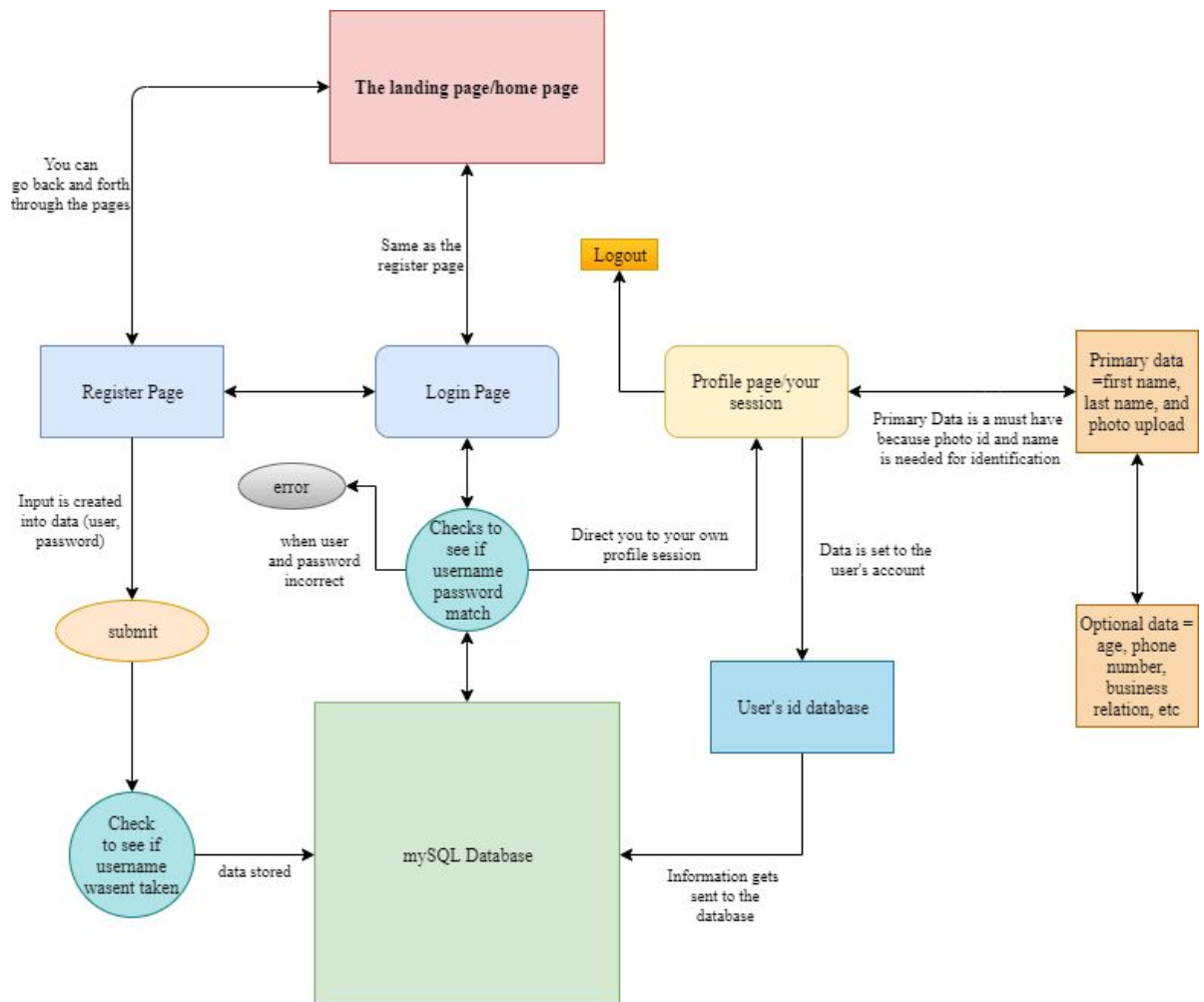


## 2.4 Profile

After logging in the user will be redirected to this page. This page is only accessible when a user is successful on logging in. The information inputted will be sent to the user's id section which is stored on the database. The profile must require a photo in order for submission to work. Although a name is optional it is also required for identification purposes. This is required in order for the google glass to work on identifying the user from matching the photo with their name. The rest of the data which is age, phone number, business relations, etc. are optional data; meaning it is dependent upon the user's choice. With all the info set, all you do is press submit and it will redirect back to the homepage indicating it was successful. If not an error will come up.

The screenshot displays a user profile interface. At the top left, there is a blue button labeled 'Instructions'. The main header area says 'Welcome: DD' next to a circular profile picture placeholder. Below this, there are several input fields arranged in two columns: 'first name', 'last name', 'email', 'education', 'job', 'position', '12/31/99' (likely a date field), 'phone', 'state', 'city', 'zip', and 'age'. There is also a text area labeled 'about'. Below the 'about' field is a 'Resume' section with a file upload button labeled 'Choose File' and a status message 'No file chosen'. At the bottom of the form is a large blue 'Submit' button. A small 'Log Out' button is visible in the bottom left corner.

## 2.5 Flow Diagram





### 3. Screens - Mobile Application

The mobile application is where the user will spend most of their time. Currently the mobile application runs off the Flutter Framework. The mobile application will be composed of 4 main screens: Login/Landing view, Connections view, Camera view, and a Business Card view.

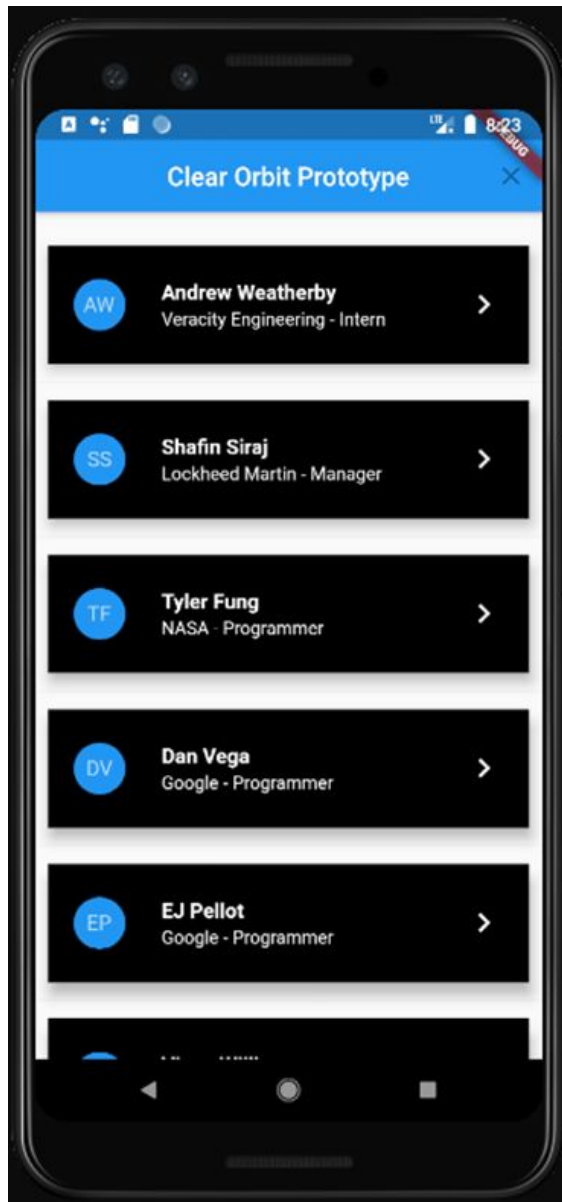
#### 3.1. Login

The login/landing page is where the user will arrive when the application first loads on the mobile device. This will also be the screen when a user signs off. When prompted for a username and password, the user will enter their credentials. Credentials can be received on the web application. If a user enters the wrong credentials, instructions will be prompted to create an account on our website. When a user enters the correct credentials, the app will redirect to the Camera view.



## 3.2. Connections View

The Connections view is a list containing anyone who has previously been recognized by the facial recognition software. Each member in the list can be expanded upon with their personal details. When the user has finished reviewing the information, the card can be collapsed so they can continue browsing connections, return to the Camera view, or review their own information on the Business Card view.



### 3.3. Business Card View

The Business Card view contains information about the logged in user. It details what information is currently entered into the system and what will be displayed to other users.



### 3.4. Navigation

When opening the mobile Clear Orbit application, the first screen that will appear is the login screen. Users will need to supply their credentials and enter them into the appropriate fields. If a user does not supply correct credentials, they will not be allowed to advance to the rest of the application and a message will be displayed stating their information was incorrect. When a user does supply the correct information they will then be allowed to progress into the application.

After a user has logged in, the first screen that will appear is the Camera view. From here the user can swipe either left or right to navigate to additional pages. If a user swipes left from the Camera view, they will be directed to the Connections view; If a user swipes right from the Camera view, they will be directed to the Business Card view. Swiping in the opposite direction from either view will direct users back to the Camera view. To log out of the application, a button located in the toolbar menu labeled “log out” will allow user to log out of their accounts. Upon log out, users will be returned to the original login page of the application.

## 4. Screens - Glass Application

The Glass Application for this project will run using Java. It will have two main screens. One of which will be used to take a picture of someone. The second, will serve the purpose of showing where the application is in the process of sending an image to the phone.

### 4.1. Opening the App

In order to open an application on Google Glass, you must say “Ok Glass” in the home screen. You will then be prompted to choose from a list of applications installed on the device through either touch or voice controls.

### 4.2. Picture Taking Screen

The picture taking screen is extremely straightforward, simple and self explanatory. This screen will be what shows up when you open the Glass Application. It will showed detailed instructions on how to properly use the application with the mobile application. Once the user taps the side of the Glass, they will be taken to the bluetooth screen, which will show the user how far along the process of sending a photo they are in.

## 5. Backend Systems

Amazon Web Services (AWS) will be the backend processing for the facial recognition portion of the Clear Orbit application. The *Amazon Rekognition* API supplies a vast selection of tools to use in analysing pictures, videos, and streams. The Clear Orbit application communicates with the AWS API directly through the mobile application when a snapshot is taken. When a match has been made between a registered user and a person in the submitted picture, the Clear Orbit database is updated along with the mobile application.

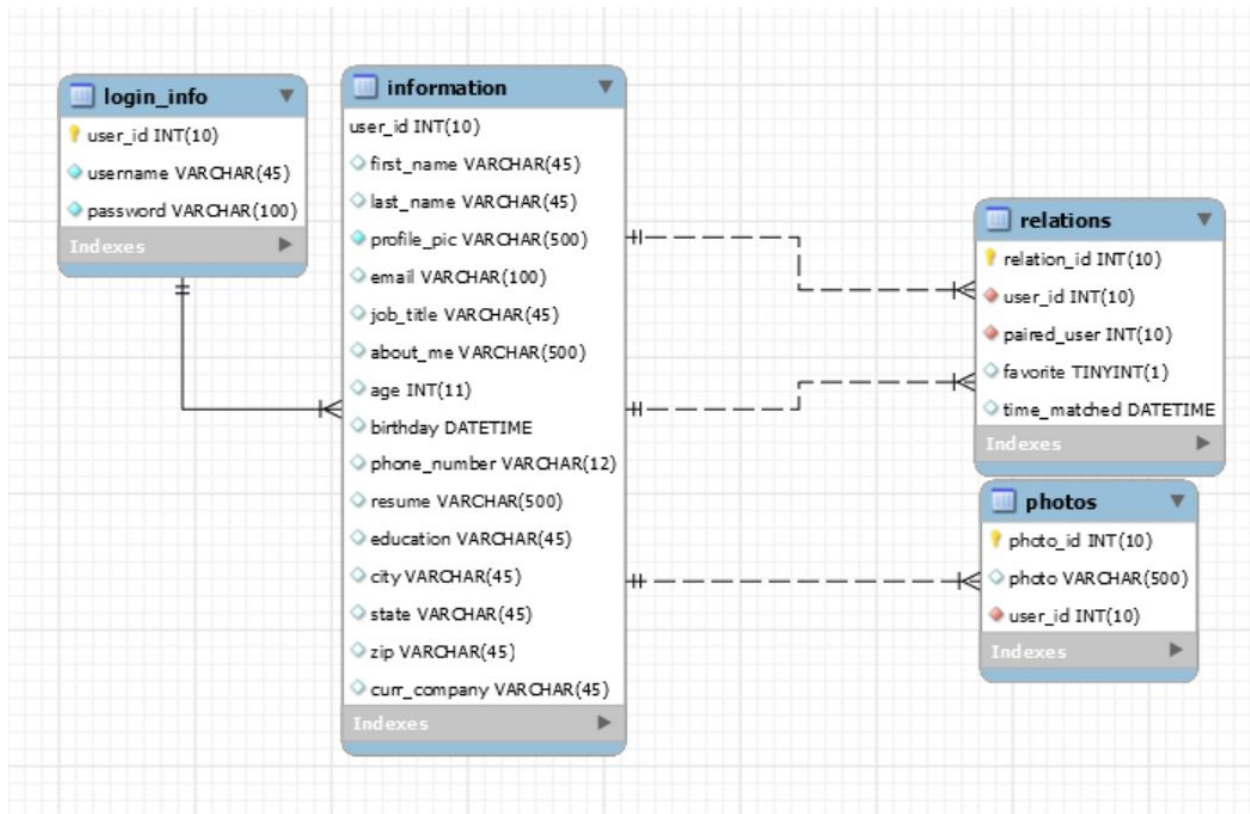
### 5.1. Amazon Web Services

Amazon Web Services (AWS) is the backend processing for the facial recognition portion of the Clear Orbit application. The *Amazon Rekognition* API supplies a vast selection of tools to use in analysing pictures, videos, and streams. The Clear Orbit application communicates with the AWS API directly through the mobile application when a snapshot is taken. When a match has been made between a registered user and a person in the submitted picture, the Clear Orbit database is updated along with the mobile application.

## 5.2. Database

The database contains all the information of registered users. Currently, the database runs off of MYSQL. When a user creates a profile on the web application, the users information will be stored in the login\_info table with an automated user\_id, username, and hashed password. Information can later be retrieved by both the web application and the mobile application to be reviewed. Currently only the web application will be able to alter the information located in the information table. The Mobile application will be able to login by accessing information from the login\_info table; and also retrieve information using the information, photos, and relations tables to grab relevant information.

There are two major relations between the database. These include Login\_info and Information. They both share a one-to-one relationship with user\_id. A user can only have one username/password and an Information variable for their user\_id. The photos and relations table have a one-to-many relationship with the Information table and the Primary Key (user\_id). This is due to the fact that the user is expected to have many different photos that correlate with the user\_id.



## 5.3. Restful Endpoints

Resource	Get (Read)	Post (Create)	Put (Update)	Delete (Delete)
/id	Returns specific users information	Method not allowed	Method not allowed	Deletes specific user
/photocomparision/{user_id}	Returns a status of face match	Receives picture from mobile	Method not allowed	Method not allowed
/connections/{user_id}	Returns a user's connection list	Method not allowed	Method not allowed	Method not allowed
/login	Returns to login template	Receives User's login information	Method not allowed	Method not allowed
/register	Returns registers template	Creates a user account	Method not allowed	Method not allowed
/logout	Method not allowed	Clears Session	Method not allowed	Method not allowed
/initialize	Return account info page and info	Creates info for account	Method not allowed	Method not allowed
/homeOrb	Returns landing page template	Method not allowed	Method not allowed	Method not allowed
/mlogin	Returns login status	Receives username and password from mobile	Method not allowed	Method not allowed
/profileinfo	Returns users information from session	Method not allowed	Method not allowed	Method not allowed



## 6. Team Contribution

### Edwin Pellot - **Backend Developer** (AWS/ Mobile Focus)

Creating endpoints/routes for mobile services which include login, connections/relations, photo uploads, and user information. With that I'll be helping Andrew and Vince with creating the functionality between the Flutter front-end and the Flask application. I will also be assisting Shafin in implementing the Facial Rekognition with our mySQL database.

### Daniel Vega - **Glass Developer** (Glass Focus)

Creating the Google Glass application. This includes picture taking with the Glass through the application, navigation using the touch and speech controls, displaying data to the user and using the Android bluetooth api to send and receive photos. I will be working with the mobile team to help integrate bluetooth capabilities into the mobile application. This is to allow for sending and receiving information between the phone and the Glass.

### Shafin Siraj - **Backend Developer** (Database/ Website Focus)

Responsible for implementing our mySQL database and Facial Rekognician API along with EJ. Creating endpoints/routes for the website which include login, registration, photo/ information uploads, and user information retrieval. Will be working extensively with Tyler to implement backend for the website. Will also manage the search algorithm for the Facial Rekognition Will also be assisting E.J. with mobile backend needs with sending information to the mobile app

### Tyler Fung - **Front-End Developer** (Website Focus)

Responsible for creating the front end of our website. By implementing the visual elements within our website, the user will be able to create an account and input their information. After successfully creating the bases of our website my future role will be monitoring the front end side of the website in case of adding more content or fixing up any styling issues. I will also be accounted for working with Shafin on any new merges in case a part of the project changed the whole site and be fixed immediately.

### Vince Williams - **Front-End Developer** (Mobile Application Focus)

I am responsible for creating the front end of the mobile application with Andrew. By interacting with the mobile application the user will be able to see a list of previous users they have interacted with, take photos of a user to see their profile, and also view their personal profile. Going forward I will be working closely with Dan and Andrew to incorporate the connection with the Google Glasses and mobile app through the use of bluetooth. Andrew and I will also be fine tuning all features currently listed in the mobile application. This will include style changes, server interactions, and debugging any possible errors in our code.

---

Andrew Weatherby - **Front-End Developer** (Mobile Application Focus)

Responsible for creating the front end mobile application with Vince. While Vince will be working with Dan to connect the Google Glass, I will be focusing on making connections between the mobile application and the RESTful endpoints EJ. Connecting the points includes verifying login credentials, retrieving user information, and displaying the user information in the mobile application. This is a critical part of the project as without the connections, the application will be unable to function. Vince and I will be working closely to merge our code into the final product.

## 7. Mid-Semester Status

By mid-semester evaluation, the mobile application is planned to be communicating with the backend system, which will grant us the ability to access user information through their credentials, display user information and send photo data to the backend. Along with connecting the mobile application to the backend, we plan to have a method to connect the Google Glass to our mobile application. We will also have the full implementation of the Website and its backend with registration, login information, and initializing and accessing user information.