Software Project Proposal (COMP 680 ADV TOPICS SWE)

# Software Project Proposal Topic:

**Face Recognition Web Application**

**Team 2:** Karthika Dharmia Rajendran, Mansoureh Lord

# Proposal

**Business Objective:**

Recent technological developments have led to a wide proliferation of face recognition features. Face recognition can be used for a wide range of use cases including preventing crime, protecting law enforcement and even providing more convenient air travel.

In accordance with that, based on serverless computing our team has intended to create a face recognition web application that could help across various business sectors.

**Technology Stack:**

AWS services: S3, Lambda, DynamoDB, Rekognition, Cloudwatch, SES.

Languages: HTML, Javascript, Python

**Team Website:**

<https://sites.google.com/view/comp680group2>

**Sprint 1:**

**Business Goal/Objective:** To search if a person's identity is present in the database.

**User story:**

As an additional security measure for an event in the children's school, the school authorities would like to have a web application that detects peoples` faces and allows them to enter inside the school campus only if they are parents during school hours. Added to this, it makes it easier to identify parents in case the child is picked up earlier to school hours.

**Assumption** : The images of all the parents are already present in the school database.

**Story I**: The school authorities want a web application to upload images.

**Technical Tasks**:

* Creation of a webpage.

**Story II** : As a parent, the school wants my image to be uploaded for security verification purposes.

**Technical Tasks**:

* Loading of the webpage to AWS S3.

**Story III** : The school authorities want the uploaded image to be checked against the uploaded parent image to verify if it is a parent or a stranger.

**Technical Tasks**:

* Triggering AWS Lambda based on S3 upload of the image.
* Implementing the logic in AWS lambda to access the DynamoDB table items as per the uploaded image after detecting the face in the image.
* Setting up Identity and Access Management roles in lambda.

**Story IV** : If the uploaded image is an image of a parent who is already present in the school database, give them access to pick up the child earlier to school hours.

**Technical Tasks**:

* Give a confirmation message to the parent.

**Sprint 2:**

**Business Goal/Objective:** To allow parents to register their details using the registration page, and to trigger automatic email on daily basis to the parents about their children check-in and check-out details.

**Story I**: The registration page should take parent details such as first name, last name, email ID, child`s name and parent`s image as input.

**Technical Tasks**:

* Creation of a web page involving HTML, CSS and Javascript to take the user input.
* All the mandatory inputs should be checked if present and appropriate error messages to be triggered.

**Story II**: Once the parent clicks Register button, the input details should be passed on to the API gateway services for it to be updated to the database.

**Technical Tasks**:

* Configuration of API gateway for REST API services including the end points to send the POST parameters from the url to the associated lambda function, and to receive the return value from lambda using the ‘GET’ parameters.

**Story III**: The given input details of the parent should be saved in database for later verification and comparison for face recognition. The parents should be given a confirmation message that their details are stored along with the identification number that is generated automatically.

**Technical Tasks**:

* The REST API service should trigger lambda. The input parameters that are passed through ‘POST’ parameters are accessed in lambda.
* The image of the parent should be stored in a S3 bucket (Comparison bucket named ‘east1bucket1).
* The parent`s name, child`s name and the email ID should be stored in a table named ‘ParentDetails’ in Dynamodb.
* The current number of records in the table ‘ParentDetails’ is extracted and is sequentially incremented to get an identification number for the newly registered parent.

**Story IV:** The parent should receive an email on a daily basis regarding the child`s check in and check out time.

**Technical Tasks:**

* An input in the web page to be provided for the parent to give reason as ‘Checkin’ or ‘Checkout’ when he/she drops/picks up the child. Once the parent clicks submit, a database table ‘CheckinDetails’ has to get updated.
* The update ‘Checkout’ to the entry in the table should trigger a lambda function that should send email notification to the parent regarding the check-in and check out time of the child for that specific day.