

# InstaAudit - ADD



**California State University | Long Beach**

Members:

Greyson Cabrera

Kevin Reed

Dustin Martin

Tylar Simone

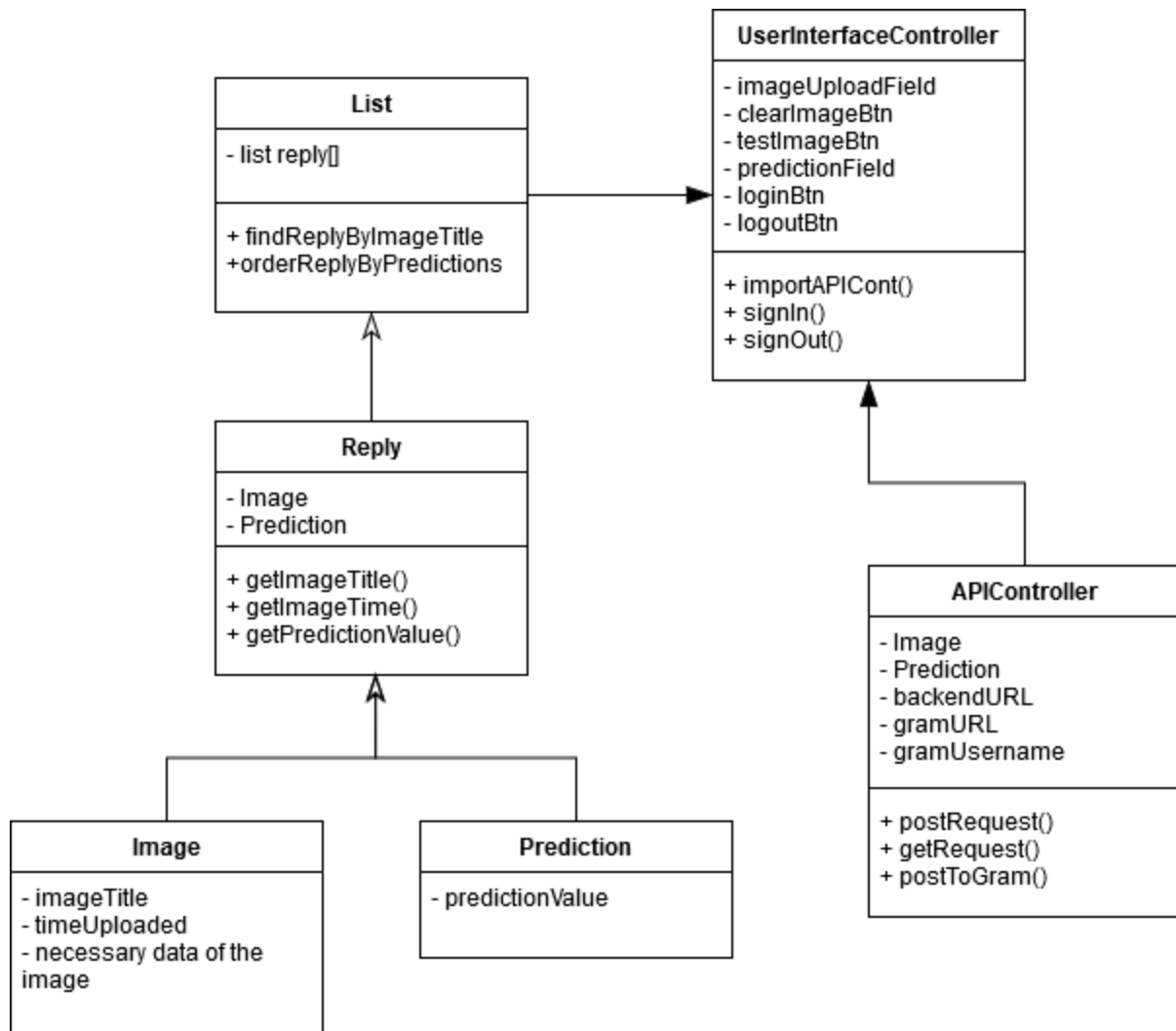
# Contents

<b>Contents</b>	<b>2</b>
<b>Diagrams</b>	<b>4</b>
Class Diagram	4
Deployment Diagram	5
Interfacing with the Database/ ML Algorithm:	5
Use Case Diagram	6
“Text Based” Use Cases	7
Data Modeling Diagram	15
Sequence Diagrams	16
Activity Diagrams	17
<b>Tradeoff Analysis</b>	<b>18</b>
MERN Stack vs. Python-Django Stack	18
Connect Machine Learning Model through Flask vs. Tensorflow.js	19
NoSQL (i.e. Mongo, Firebase) vs. SQL Database (i.e. MySQL)	20
REST API	21
Cloud Decisions	22
Instagram Scraper	23
Describe our dataset and how we procure it.	23
<b>Machine Learning Write-up</b>	<b>24</b>

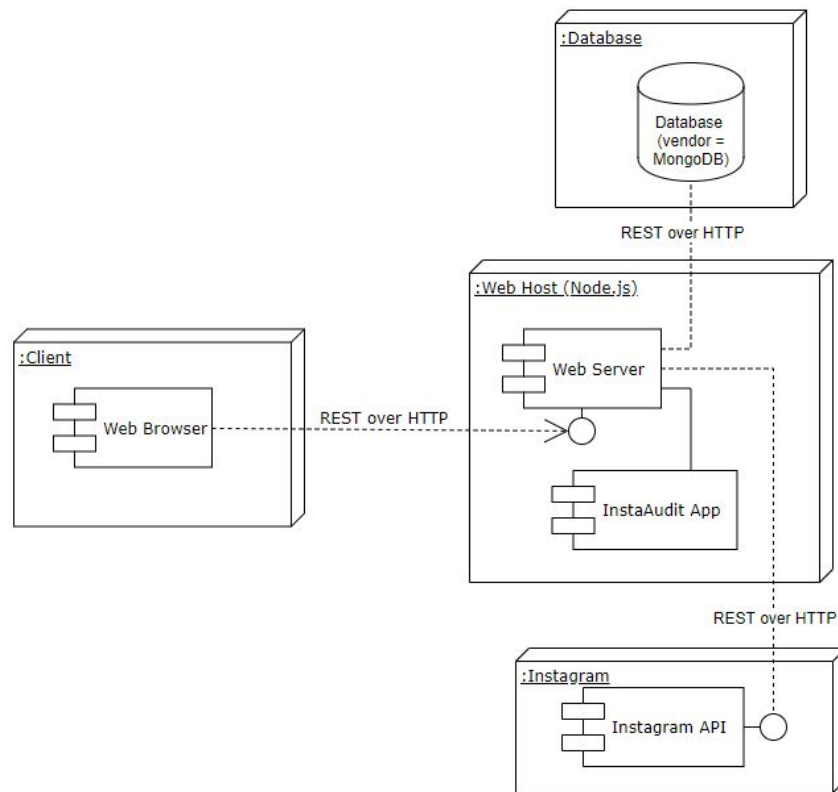


# Diagrams

## Class Diagram



## Deployment Diagram



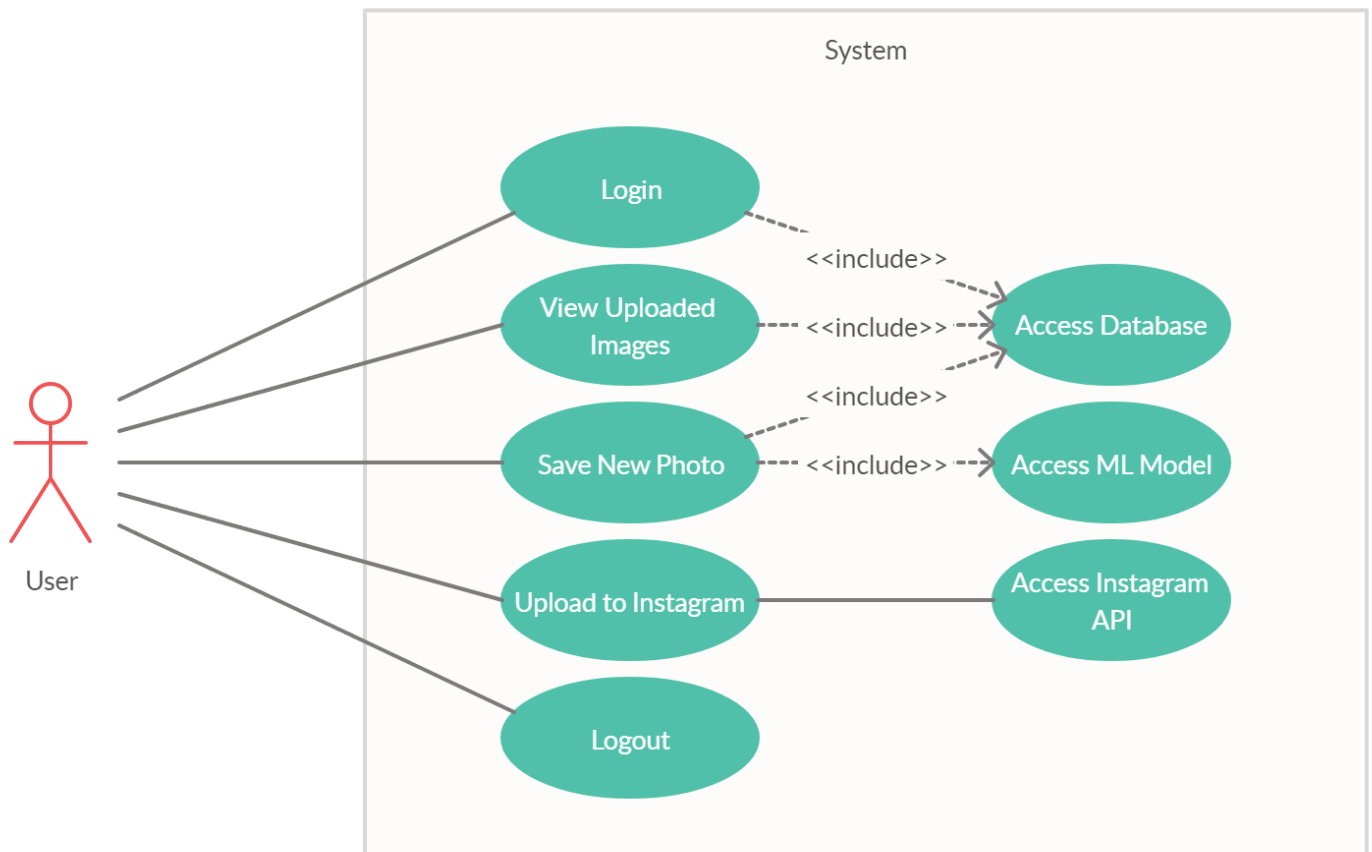
### Interfacing with the Database/ ML Algorithm:

We will be running a webapp where the user can interact with our machine learning algorithm after creating an account and logging into our system. Any API attempts to interact with our machine learning algorithm will be denied if the requests do not have a valid session. A valid session would only be able to be created if the user provides valid login information, where that information would be validated by our database. A webapp instance would be communicating with the server through REST on HTTP.

### Languages/ Frameworks:

Standard webpage built upon React/Express.js. Python will be used for much of the project backend including web-scraping, the machine learning algorithm, and some of the server/client functionalities. We will be using MERN to connect Node.js and Python. MongoDB used to store user login info and potentially previous session data (previous images and their ranks).

# Use Case Diagram



## “Text Based” Use Cases

Use-case Field	UC-01
Use-case Name	Login
Subject Area	User Activities
Business Event	User opens application
Actors	User
Use-case overview	Login into application by validating entered credentials
Preconditions	User must have account in system
Termination Outcome	Access into the system is granted.
	No access is granted.
	Prompted to create an account.
Condition Affecting Termination Outcome	Credentials are valid
	Credentials are invalid
	No account exists
Use-case Description	User is prompted with username and password fields. User enters correct credentials and is granted access into the system.
Use-case associations	Access Database
Traceability	N/A
Input Summary	Username and password
Output Summary	Error message for invalid or non existent credentials
Usability Index	9
Use-case Notes	Credentials are stored in the system’s database.

Use-case Field	UC-02
Use-case Name	View Uploaded Images
Subject Area	User Activities
Business Event	User has successfully logged in
	User selects view images menu option
Actors	User
Use-case overview	Users are able to quickly glance at all uploaded images with predicted ranking easily visible on image.
Preconditions	User has successfully logged into account
Termination Outcome	Application displays a different menu option
Condition Affecting Termination Outcome	User selects a menu option that is not “view images”
Use-case Description	User successfully logs in and is directed to the view images menu option by default or User selects view images menu option.
Use-case associations	Access Database
Traceability	N/A
Input Summary	None
Output Summary	Displays images and associated rankings for that account
Usability Index	8
Use-case Notes	Images and rankings are stored in system database



Use-case Field	UC-03
Use-case Name	Save New Photo
Subject Area	User Activities
Business Event	User selects menu option to upload new image
Actors	User
Use-case overview	User is prompted with an upload image page that allows a user to upload a new image to be ranked by the application.
Preconditions	User has successfully logged into an account
Termination Outcome	Image ranking is shown
Condition Affecting Termination Outcome	User uploads image
Use-case Description	User uploads a new image into the system and the system assigns a ranking to that image.
Use-case associations	Access Database, Access ML Model
Traceability	N/A
Input Summary	Image
Output Summary	Ranking
Usability Index	10
Use-case Notes	After ranking, the image and ranking are uploaded into the database

Use-case Field	UC-04
Use-case Name	Access Database
Subject Area	User Activities
Business Event	Attempted login
	View images selected
	Upload new image selected
Actors	User
Use-case overview	System is prompted to retrieve information from the database and the database either returns the specified information or an error if the information cannot be located.
Preconditions	User has successfully logged into an account
Termination Outcome	Information is returned
	Error is returned
Condition Affecting Termination Outcome	Query is valid and results are found
	Query is invalid and/or no results are found
Use-case Description	System requests information from the database and the database returns the requested information
Use-case associations	Login, View Uploaded Images, Upload New Image
Traceability	N/A
Input Summary	No user interaction
Output Summary	Requested data or an error
Usability Index	10
Use-case Notes	Use cases will request different data.

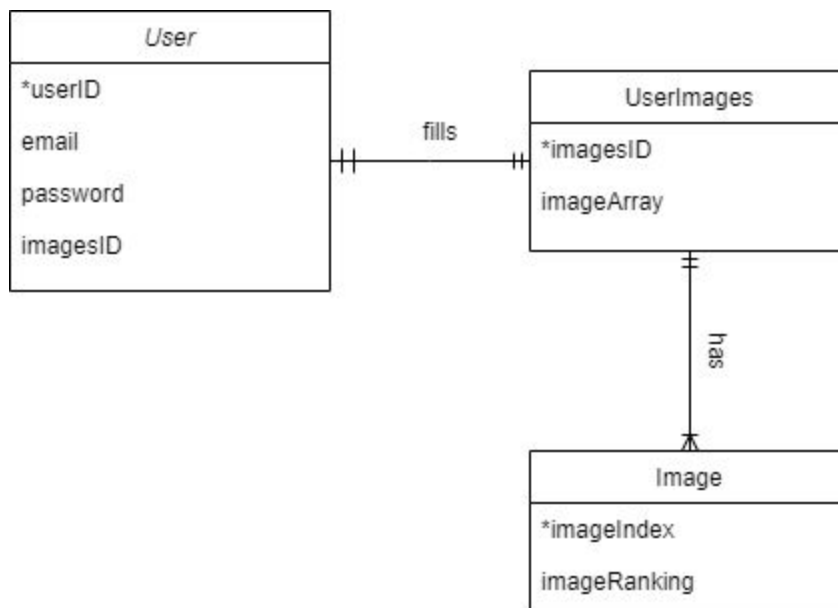
Use-case Field	UC-05
Use-case Name	Access ML Model
Subject Area	User Activities
Business Event	User has submitted their uploaded image
Actors	User
Use-case overview	Send a request to the ML Model to get a prediction
Preconditions	User must be signed-in and upload an image
Termination Outcome	A response comes back from the ML Model
	Connection to ML Model fails
	Image uploaded is not acceptable
Condition Affecting Termination Outcome	User request was a success
	The connection to the ML Model was disrupted
	Image given is not of valid type
Use-case Description	User has uploaded an image and clicked the submit button, a request is formed, then sent to the ML Model
Use-case associations	Upload New Image
Traceability	N/A
Input Summary	Image
Output Summary	Response that has the prediction or an error message for an invalid image
Usability Index	9
Use-case Notes	Predictions will be displayed to the user

Use-case Field	UC-06
Use-case Name	Upload to Instagram
Subject Area	User Activities
Business Event	User selects the option to upload the image to instagram.
Actors	User
Use-case overview	A user selects the option to upload the photo to instagram, a confirmation box appears and if confirmed, the image is uploaded to the associated instagram account.
Preconditions	User has selected a picture from the view images page
Termination Outcome	Return to view images page
Condition Affecting Termination Outcome	User confirms upload
Use-case Description	User selects the button to upload the current photo to instagram and confirms the action.
Use-case associations	Access Instagram API
Traceability	N/A
Input Summary	N/A
Output Summary	N/A
Usability Index	6
Use-case Notes	Requires connection with an existing instagram account

Use-case Field	UC-07
Use-case Name	Logout
Subject Area	User Activities
Business Event	User successfully logs out of the system
Actors	User
Use-case overview	Logout of the application and User's connection to the backend is severed
Preconditions	User must be signed in to the system
Termination Outcome	Users access to the system is severed
Condition Affecting Termination Outcome	User selects logout button
Use-case Description	User is shown a prompt displaying to the user that they have been disconnected from their session
Use-case associations	N/A
Traceability	N/A
Input Summary	N/A
Output Summary	Message that signifies to the user that they have disconnected
Usability Index	9
Use-case Notes	Previous information tied to an account that was being displayed must be removed

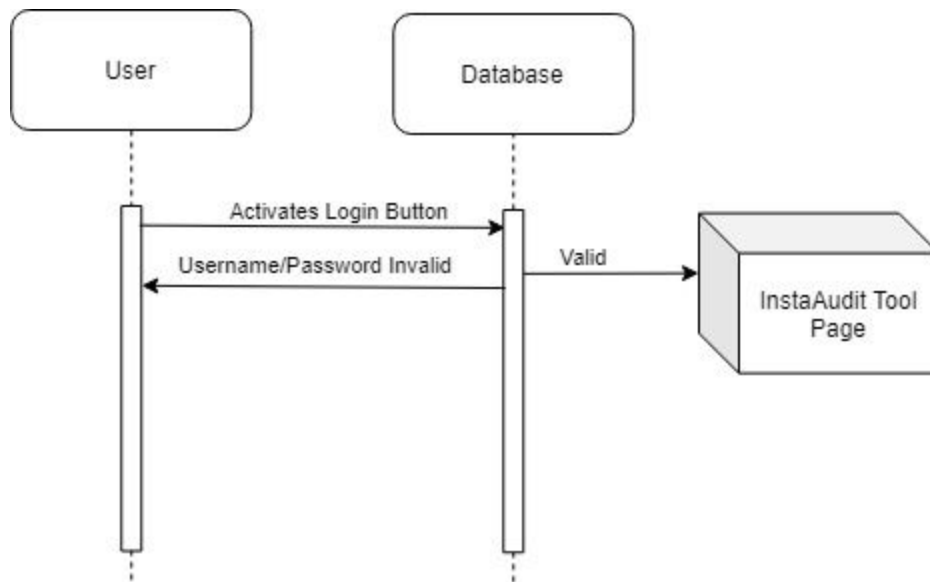
Use-case Field	UC-08
Use-case Name	Access Instagram API
Subject Area	User Activities
Business Event	User confirms upload to instagram
Actors	User
Use-case overview	Image is uploaded to instagram via the instagram API
Preconditions	User confirms image to be uploaded to instagram
Termination Outcome	Return to View images page
Condition Affecting Termination Outcome	Upload successful
Use-case Description	Image is successfully uploaded to the associated instagram account
Use-case associations	Upload new image
Traceability	N/A
Input Summary	N/A
Output Summary	Success or failure message
Usability Index	6
Use-case Notes	Requires connection to an existing instagram account

## Data Modeling Diagram

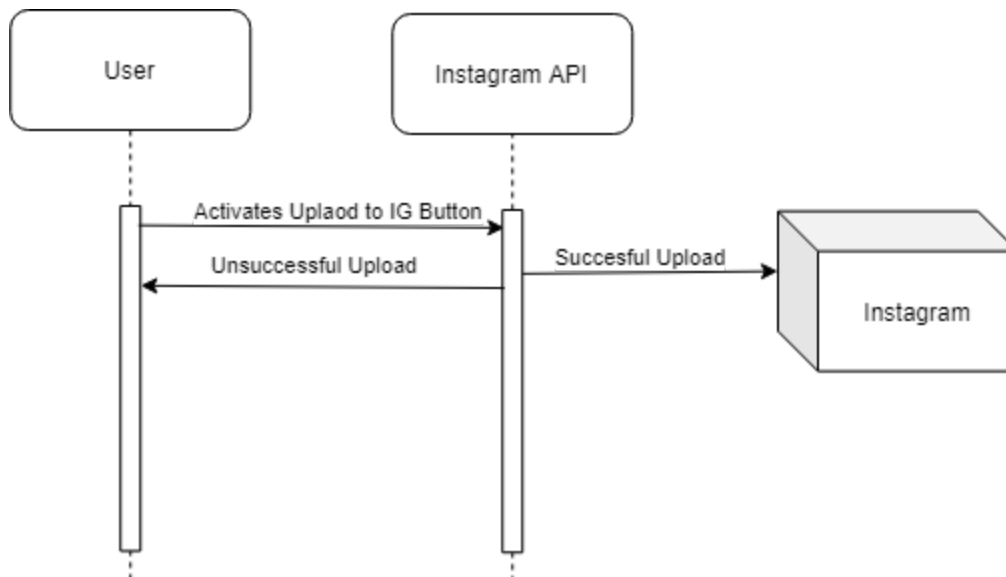


# Sequence Diagrams

## Login Sequence Diagram

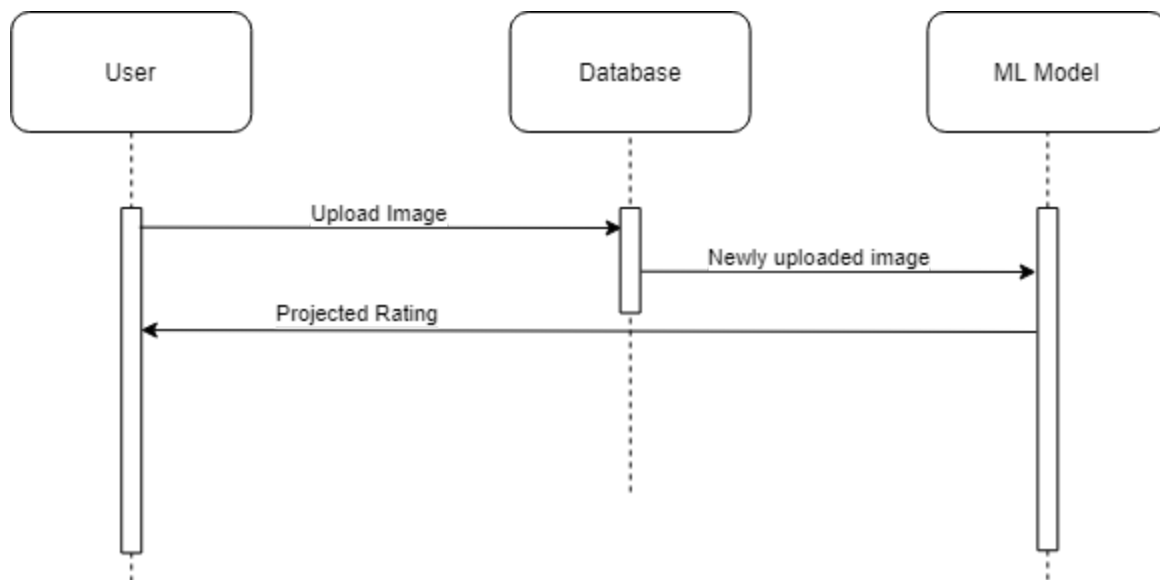


## Upload to Instagram Sequence Diagram

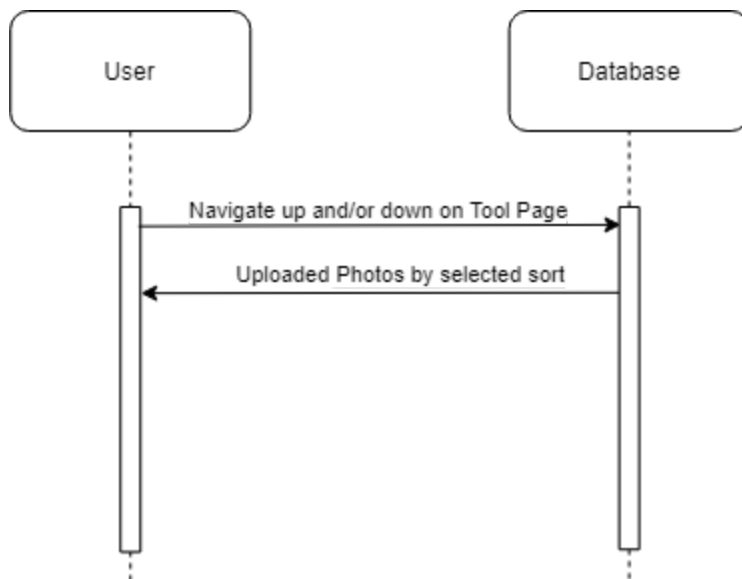




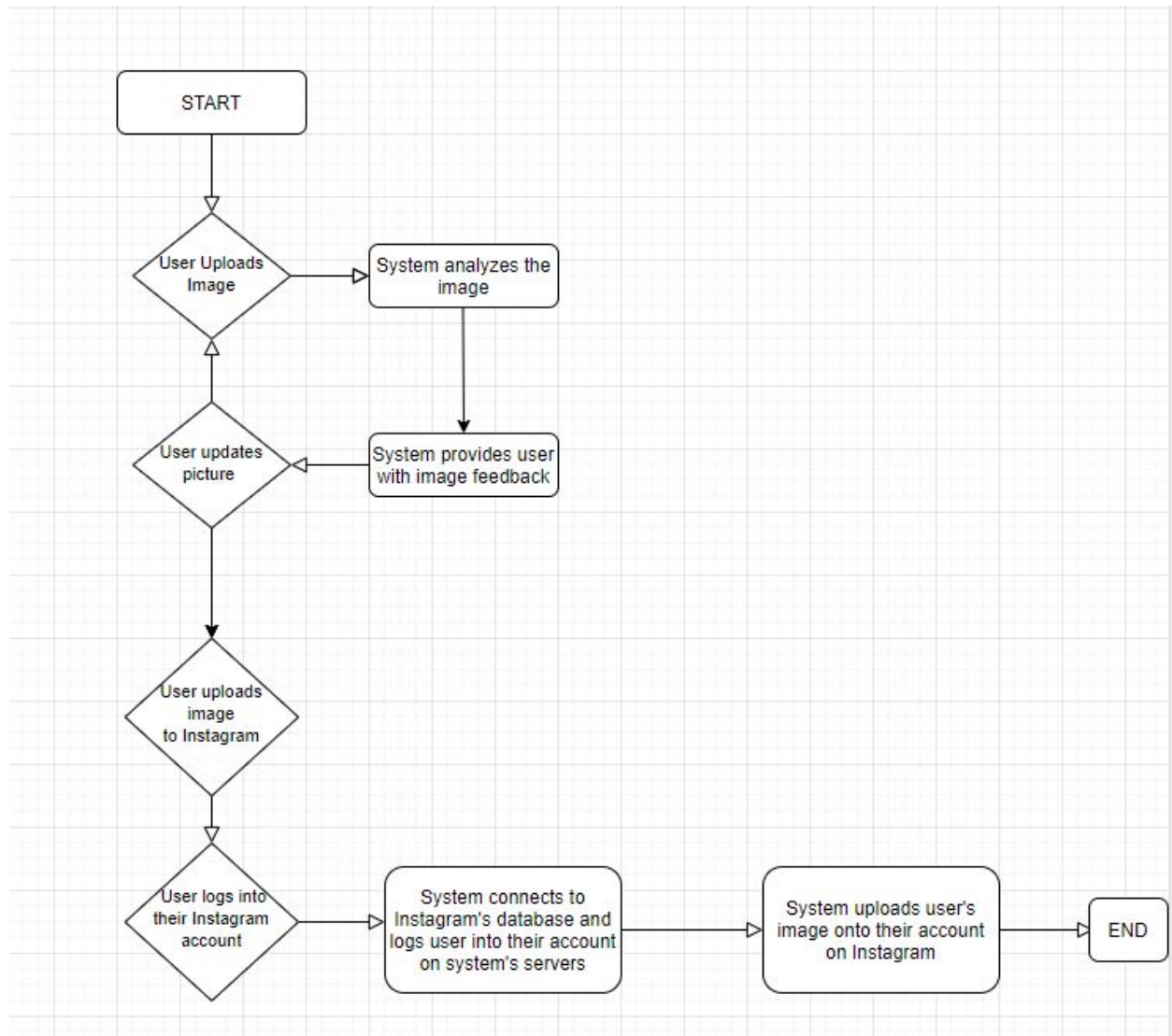
### Save New Photo Sequence Diagram



### View Uploaded Images Sequence Diagram



# Activity Diagram



# Tradeoff Analysis

## MERN Stack vs. Python-Django Stack

Criteria	MERN	Python-Django
Team Preference	5	5
Previous Experience	6	3
Quick Setup	9	9
Scalable	5	5
Available Tutorials	6	2
Security	5	5
Older	6	9
Extend the Stack	8	4
Trending	8	7
Score	58	49
Rank	1	2

Our team chose the MERN stack since it has many tutorials available and with it being reliant on javascript it has a lot of extra libraries we can include to extend our functionality.

## Connect Machine Learning Model through Flask vs. Tensorflow.js

Selection Criteria	Flask	Tensorflow.js
Team Preference	7	5
Previous Experience	6	4
Popularity	8	8
Sustainability	6	9
Support	8	9
Scalability	9	8
Score	44	43
Rank	1	2

In the end, our team chose Flask due to its lightweight and customizable nature. Flask allows our team to utilize only the extensions that are necessary, rather than a number of features that our application will never use.

## NoSQL (i.e. Mongo, Firebase) vs. SQL Database (i.e. MySQL)

Criteria /5	NoSQL		SQL
	MongoDB	FireBase	MySQL
Team Preference	8	4	1
Team Experience	2	0	2
Popularity in the field.	7	6	5
Performance	7	5	4
Operability	5	6	2
Ease of Implementation	5	5	3
Scalability	5	5	5
Security	7	5	5
<b>Total</b>	46	36	27
<b>Rank</b>	<b>1</b>	2	3

Our Team settled on MongoDB for saving user login and session info, as well as previous-use information (uploaded images, previous rankings) if such use-case is to be implemented.

## REST API

Our REST API was written in Node.js and serves as the backend component to our server.

- The client side component is served on the same server and sends requests to our REST API.
- The requests are sent to the corresponding route for the requested resource. The route is the code that handles an HTTP request (GET, POST, PATCH, DELETE) passes through any middleware required for the HTTP request.
- The route connects to the endpoints to access our database. Endpoints refer to the resources we access in a response to our HTTP request.
- The response is received by our client side and presented to the user in a useful manner.

The Middleware attached to most of our HTTP requests is an authorization function that checks the received JSON Web Token attached to the HTTP request header. For the creation of user products there is another middleware that collects any images attached to this request and stores them on the server.

## Cloud Decisions

1. Our database exists on the cloud through MongoDB's free Atlas tier. The database is provided by Amazon Web Services. Specifically our database is a cluster that exists in Oregon in an Amazon datacenter.
2. Heroku is a cloud platform where we can deploy our full stack web app to. Heroku uses Dynos which are isolated and virtualized containers that execute user specified code. Web Dynos handle the requests being sent to them, manages many requests to a queue, then executes those queued requests on the correct Dyno.
3. Google Colab is a free environment that allows us access to Google Cloud Servers. In terms of Machine Learning Google Colab gives us access to GPU's and TPU's to hasten our data processing and the training of our models. Since we have a convolutional neural network the added processing power is extremely useful for training a model with many layers.

## Instagram Scraper

Our Instagram Scraper collects image and engagement data from Instagram including the likes, comments, and upload time for the post. Our dataset should be organized by the industry the posts are from (i.e. fashion, tech, fitness, etc.) Some techniques for processing engagement equally across each post would be to create ratios from the likes and comments attached to each post.



# Machine Learning Write-up

## Business objective for using machine learning

- Provide Enterprises with accurate predictions on levels of engagement for uploaded Instagram posts before posting to the main account.

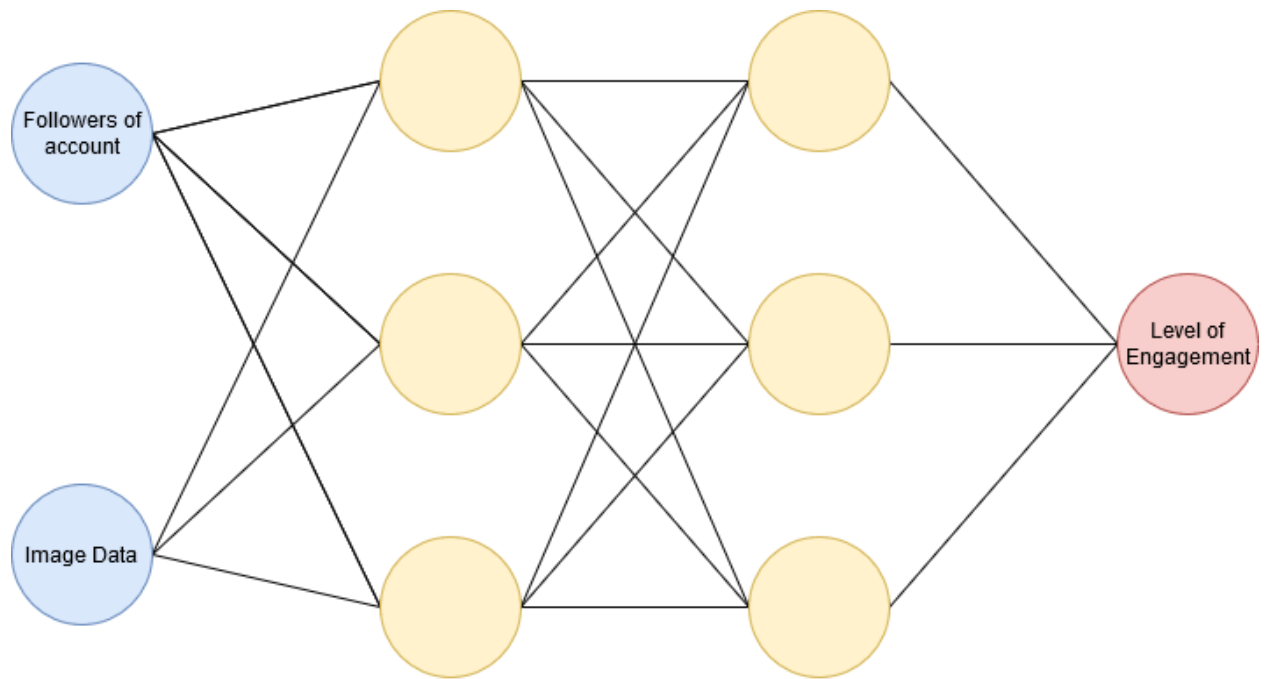
## Data Sources

- Instagram web-scraper (through REST API calls) will grab post data from business marketing accounts and classify posts based on their engagement and overall ranking among other, similar posts.
  - Example sources:
    - <https://www.instagram.com/walmart/>
    - <https://www.instagram.com/target/>
  - Data will be used to train the ML model on what constitutes a high-engagement post.
- Such data is necessary to provide accurate estimates of post engagement on similar types of Instagram posts.
  - Not all instagram posts will work. A business marketing ML model needs business marketing data.

## Machine Learning 10 step documentation

1. The problem is determining whether or not a post will provide the necessary engagement needed to run a marketing account on instagram.
2. Utilizing a web-scraping script sending API requests to Instagram, we can gather mass amounts of posts from thousands of business marketing related accounts.
  - a. Image Data
  - b. Follower count.
  - c. Post Engagement (Likes, comments)
3. Data will be formatted so each post know's the account where posted, and the accompanying engagement it received.
4. Before training the model, the data needs to be classified.
  - a. Posts will be given a ranking, decided by its relative engagement compared to other posts by the same account.
  - b. This ranking is the inevitable output our model will be tasked to estimate.
5. A subset of this data will be split to use as a test set. The rest will be fed into the model to train the ML model.
6. Selecting a number of models and choosing the one with the most accurate result is crucial for effectiveness
7. Model is trained on the training set
8. From the results, we can compare the ML's ranking with the test post's actual ranking.
9. From here, we can predict the eventual effectiveness of the ML model and plan accordingly
10. Deploy model, monitor, and reuse

## Model Design



# Risk Management

Description	Severity	Solution	Status
Instagram Availability	10	Instagram Scraper	Solved
Parsing Test Data	10	Determine what data we can use from the Instagram Scraper	Unsolved
Creating a CNN	10	Study tutorials that are similar to what we want to accomplish	Unsolved
Connecting webapp to CNN	10	Using flask.js	Unsolved
Create Frontend	10	Using Node, React, Express	Solved
Create Backend	10	MongoDB	Solved
User login	5	Javascript tutorial	Solved
Connect Frontend to Backend	9	Middleware Proxy	Solved
User uploads image	8	Javascript tutorial	Solved
User receives a prediction	9	Connect through Flask	Unsolved
Webapp stores all the predictions made in a session	3	Setup a list	Unsolved
Webapp API implementation	5	Implement with axios library	Solved