Automatic Ticketing System

theDoers

The Current Scenario

- The way ticketing works currently is either the passenger buys the ticket before getting onboard or the ticket is issued by a human after boarding.

- This makes it very inconvenient as there is a long waiting time at the counters, problems related to cash or there is a need to remember to carry a card always.

What is the project about?

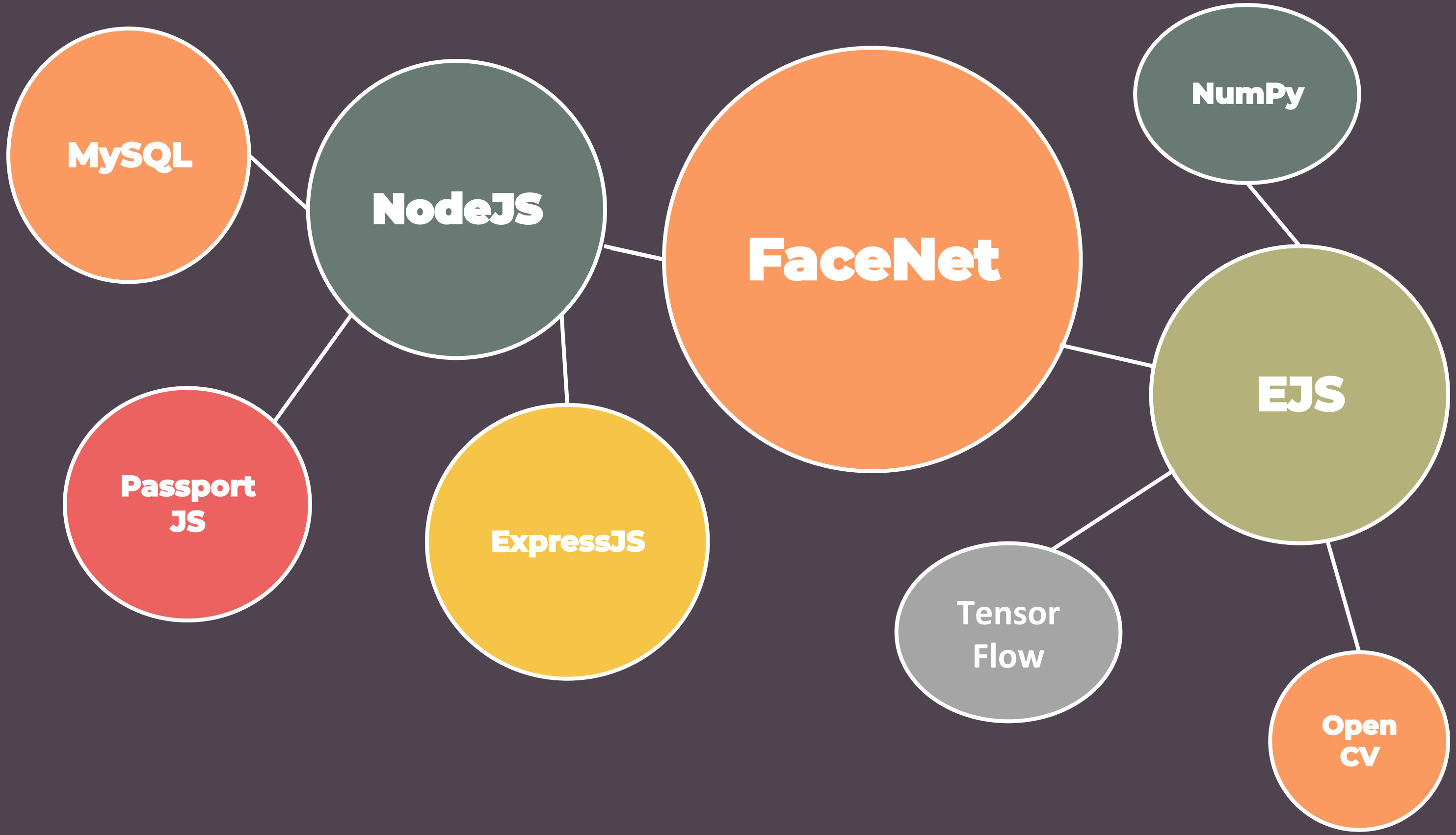
The Project aims to automate the current token system in Metros and other public modes of transport. Facial recognition is being used to identify and validate passengers. There is no ticket being issued, and the entire travel will be made very convenient as there is no waiting in ques to buy tickets or cash/card involved.

The Implementation

- The passenger has to first register at one of the centers, wherein a few of his photos are captured and details like name, mobile, email are fed to the system.
- The passenger is allowed to choose between wallet or bank account which is then going to be linked with a unique key generated for the passenger.
- When the passenger enters the premises, at the security check, his face is scanned and is then identified by facial recognition model.

- The model returns the unique ID to the server which adds the starting point to the passengers record in the database.
- After the travel is complete, the passenger's face is scanned again and the ID is fed to the server.
- The Server then adds the terminal point in the passengers record after which the fare is calculated and is deducted from the passengers linked account.
- The travel is then updated in the database for the sake of record keeping.

Technologies Used



- Google’s Facenet model is used to generate the embeddings to train the Artificial Neural Network to recognize the commuter’s face.
- MySQL, a relational database is being used for all the storage needs.
- The Front end will be designed using ReactJS.
- We will be using MaterialIO for styling the Front end.
- JavaScript (NodeJS) is used for the backend.
- Express, MySQL, body-parser, basic auth, and other node modules are used for the web app.
- We will be using Python for facial recognition and fare calculation.
- Various python libraries like Tensorflow, NumPy, OpenCV, etc.

The Three Tier Architecture

Front End

For adding and accessing passenger details, also issuing tokens to have a re-scan in rare cases.

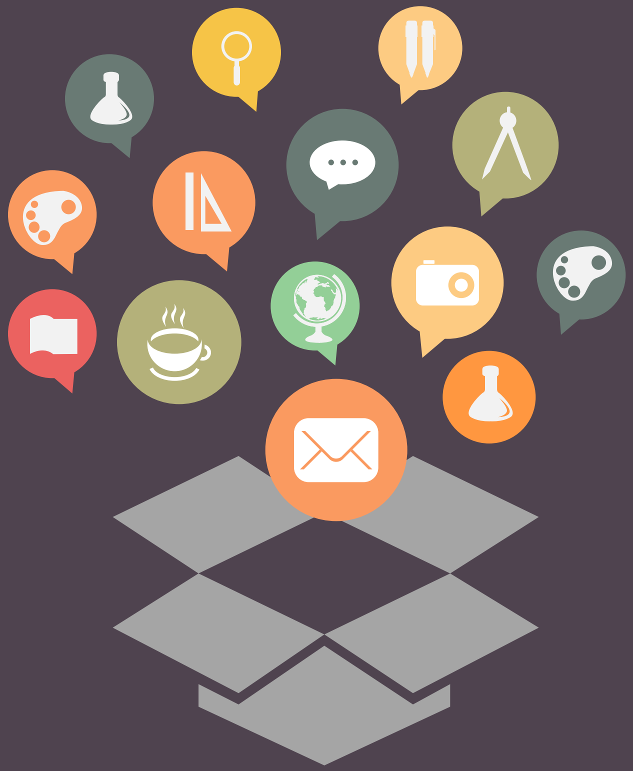
Database

The Database is used to store all passenger details like images which are required for faceNet model to ID passengers

Backend

The Server adds travel records, calculates fare, issues warnings and is used to access the database.

Why is our Project important?



- People need to start using public transport more often; it is a more environmentally friendly option, but long waiting lines discourage commuters.
- Our project aims at changing the ticket checking system. We will use facial recognition to identify a passenger so that the process gets short, hence there will be less physical contact, lesser crowds, and shorter waiting lines.
- It will also be helpful for the civic authorities to catch hold of criminals and defaulters.

Time for the questions!

Thank You