

Draft Design Document

for

Traffic Intensity Prediction and Recommendation

version 2.0

February 11, 2023

Undertaken by,

1. R. Shaikh,

Student,

Dr. Vishwanath Karad MIT WPU, Pune

**Table of Contents**

**Table of Contents ii**

**1.** **Introduction 1**

1.1 Problem Statement 3

1.2 Scope 3

1.3 List of Abbreviations 3

1.4 Overview 4

1.5 Modules 4

1.6 Design 4

1.7 Technology Stack 4

1.8 Project Plan 4

1.9 Timelines and Deliverables 4

**2.** **Design and Architecture 1**

2.1 Process Flow 1

2.2 User Interface 1

2.3 Validations

**3.** **Database Schema 1**

3.1 Database 1

3.2 Tables 1

3.3 ER Diagram

3.4 Data-Flow Diagram 1

# Introduction

## Problem Statement

Web application design and implementation of the Traffic Intensity Prediction and Recommendation (TIPR), a Machine Learning (ML) project was taken up and carried out by a graduate student from Dr. Vishwanath Karad MIT World Peace University, Pune, from 2nd January 2023, based on the assignments and requirements mentioned and proposed by authorities at Centre for Development of Advanced Computing (C-DAC).

The (TIP) web application is still undergoing development. The system has its roots in various technological fields. The TIPR is being designed and implemented for prediction purposes of the intensity of traffic at a

particular junction and recommendation to improve the conditions of the roads (widening) at that junction to overcome the problems of traffic congestion. The application is based on a to be trained Machine Learning model on which the provided data is predicted against and recommended.

Several algorithms are going to be reviewed and studied for the future implementation and development of the application

## Scope

Traffic prediction plays an essential role in intelligent transportation systems. It can help assist route planning, guide vehicle dispatching, and mitigate traffic congestion. The modern city is gradually developing into a smart city. The acceleration of urbanization and the rapid growth of population in India, bring great pressure to urban traffic management.

Traffic congestion has been increasing in India, and everything indicates that it will continue to get worse, representing an unquestionable menace to the quality of urban life. Its main expression is a progressive reduction in traffic speeds, resulting in increases in journey times, fuel consumption, other operating costs and environmental pollution, as compared with undisturbed traffic flow. There are plenty of factors behind traffic congestion and one of the important factors is road works (bottlenecks).

Urban transport supply tends to be categorized according to its capacity, i.e., the number of persons that can be transported in a given period of time. Just from the infrastructure standpoint, capacity is usually measured as the number of vehicles that can pass through in a junction at a given time; this parameter is useful when analyzing congestion.

Traffic prediction is essential to many real-world applications. For example, traffic flow predictions can help cities alleviate congestion, etc. The growing available traffic related datasets provide us potential new perspectives to explore this problem.

The goal of this project is to predict the intensity of traffic over given period of time and recommend if solution to overcome congestion is necessary or not. If solution is recommended then the count of that district increases and will later be shown in the form of map.

## List of Abbreviations

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Acronym** | **Naming Convention** |
| 1. | TIPR | Traffic Intensity Prediction and Recommendation |
| 2. | ER | Entity-Relationship |
| 3. | AI | Artificial Intelligence |
| 4. | ML | Machine Learning |
| 4. | MIT | Maharashtra Institute of Technology |
| 5. | ITS | Intelligent Transportation System |
| 6. | CLI | Command Line Interface |
| 7. | HTML | Hyper Text Markup Language |
| 8. | CSS | Cascading Style Sheet |
| 9. | JS | JavaScript |
| 10. | numpy | Numerical Python |
| 11. | pandas | Python Data Analysis Library |
| 12. | scipy | Scientific Python |
| 13. | npm | Node Package Manager |
| 14. | RxJS | Reactive Extensions for JavaScript |
| 15. | ORM | Object-Relational Mapping |
| 16. | JPA | Java Persistence API |
| 17. | SQL | Structured Query Language |
| 18. | JDBC | Java Database Connectivity |
| 19, | cors | Cross-Origin Resource Sharing |
| 20. | XML | Extensible Markup Language |
| 21. | HTTP | Hyper Text Transfer Protocol |
| 22. | JSON | JavaScript Object Notation |
| 23. | DBSCAN | Density-Based Spatial Clustering of Applications with Noise |
| 24. | UI | User Interface |
| 25. | API | Application Programming Interface |
| 26. | RDBMS | Relational Database Management System |
| 27. | DOM | Document Object Model |
| 28. | OS | Operating System |
| 29. | AJAX | Asynchronous JavaScript and XML |
| 30. | FFT | Fast Fourier Transform |
| 31. | ODE | Ordinary Differential Equation |
| 31 | C-DAC | Center for Development of Advanced Computing |

Table 2: Abbreviations

## Overview

Creation of a web application for a desktop/mobile device makes it very faster and easier to operate and look at the ground truth reality without having the burden of doing manual setup, installations and configurations for an inexperienced user.

Benefits:

* Get recommended for implementing a solution for traffic congestion
* Proper understanding of situation of traffic congestion at different situations
* Understand how many districts needed how many recommendations

## Modules

* + 1. **Machine Learning**

Machine Learning with python is the core part of this project. With help of machine learning we can visualize and build prediction models according to the data we are provided. It mainly allows us to choose or feed a computer algorithm or a model, an immense amount of data and have the computer analyze and make data-driven recommendations and decisions based on only the input data.

* + 1. **Web Programming**

Web programming refers to the writing, markup and coding involved in Web development. It is the process of creating web pages. We can shape the site according to the scope of the project, here, TIPR.

1. **Sub-module 1: User Management**
   1. Sign-up request by user
   2. Creation and registration of account by administrator
2. **TIPR User Interface**

## Design

---

## Technology Stack

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Component** | **Version** | **Description** |
| 1. | HTML | 5 | The Hyper Text Markup Language or HTML is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) for documents designed to be displayed in a [web browser](https://en.wikipedia.org/wiki/Web_browser). [Web browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and [render](https://en.wikipedia.org/wiki/Browser_engine) the documents into multimedia web pages. HTML describes the structure of a [web page](https://en.wikipedia.org/wiki/Web_page) [semantically](https://en.wikipedia.org/wiki/Semantic_Web) and originally included cues for the appearance of the document. [HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages. With HTML constructs, [images](element) and other objects such as [interactive forms](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page. |
| 2. | CSS | 3 | Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML. |
| 3. | JavaScript | V8 10.8.194.10 | JavaScript, often abbreviated as JS, is a [programming language](https://en.wikipedia.org/wiki/Programming_language) that is one of the core technologies of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/CSS). JavaScript is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), often [just-in-time compiled](https://en.wikipedia.org/wiki/Just-in-time_compilation) language that conforms to the [ECMAScript](https://en.wikipedia.org/wiki/ECMAScript) standard. It has [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing), [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) [object-orientation](https://en.wikipedia.org/wiki/Object-oriented_programming), and [first-class functions](https://en.wikipedia.org/wiki/First-class_function). It is [multi-paradigm](https://en.wikipedia.org/wiki/Programming_paradigm), supporting [event-driven](https://en.wikipedia.org/wiki/Event-driven_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming), and [imperative](https://en.wikipedia.org/wiki/Imperative_programming) [programming styles](https://en.wikipedia.org/wiki/Programming_paradigm). It has [application programming interfaces](https://en.wikipedia.org/wiki/Application_programming_interface) (APIs) for working with text, dates, [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), standard [data structures](https://en.wikipedia.org/wiki/Data_structure), and the [Document Object Model](https://en.wikipedia.org/wiki/Document_Object_Model) (DOM). |
| 4. | Bootstrap | 5.3 | Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. |
| 5. | Node.js | 18.12.0 | Node.js is a cross-platform, open-source server environment that can run on Windows, Linux, Unix, macOS, and more. Node.js is a back-end JavaScript runtime environment, runs on the V8 JavaScript Engine, and executes JavaScript code outside a web browser. |
| 6. | npm (Node package Manager) | 8.19.2 | npm is the world's largest software registry. Open source developers from every continent use npm to share and borrow packages, and many organizations use npm to manage private development as well. |
| 7. | Angular | 15.0.5 | Angular is a TypeScript-based, free and open-source web application framework. |
| 8. | TypeScript | 4.8.4 | TypeScript is a strongly typed programming language that builds on JavaScript. TypeScript code converts to JavaScript, which runs anywhere JavaScript runs |
| 9. | Karma | 6.4.0 | Karma is a JavaScript test runner that runs the unit test snippet in Angular. |
| 10. | Jasmine | 4.5.0 | Jasmine is a behavior-driven development framework for testing JavaScript code. |
| 11. | RxJS | 7.5.7 | RxJS (Reactive Extensions for JavaScript) is a library for reactive programming using observables that makes it easier to compose asynchronous or callback-based code. |
| 12. | Angular Material | 15.1.1 | Angular Material is a User Interface (UI) component library that developers can use in their Angular projects to speed up the development of elegant and consistent user interfaces. |
| 13. | Chart.js | 4.2.0 | Chart.js is a free, open-source JavaScript library for data visualization, which supports eight chart types: bar, line, area, pie, bubble, radar, polar, and scatter. |
| 14. | Java | 19.0.1 | Java is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [programming language](https://en.wikipedia.org/wiki/Programming_language) that is designed to have as few implementation [dependencies](https://en.wikipedia.org/wiki/Dependency_(computer_science)) as possible. |
| 15. | Apache Maven | 3.8.6 | Maven is a build automation tool used primarily for Java projects. |
| 16. | Spring Framework | 3.0.2 | The Spring Framework is an application framework and inversion of control container for the Java platform. |
| 17. | Hibernate ORM | 6.1.6 | Hibernate ORM (or simply Hibernate) is an [object–relational mapping](https://en.wikipedia.org/wiki/Object–relational_mapping) tool for the [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) programming language. It provides a [framework](https://en.wikipedia.org/wiki/Software_framework) for mapping an [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) domain model to a [relational database](https://en.wikipedia.org/wiki/Relational_database). Hibernate handles [object–relational impedance mismatch](https://en.wikipedia.org/wiki/Object–relational_impedance_mismatch) problems by replacing direct, [persistent](https://en.wikipedia.org/wiki/Persistence_(computer_science)) database accesses with high-level object handling functions. |
| 18. | Apache Tomcat | 10.1.4 | Apache Tomcat (called "Tomcat" for short) is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) implementation of the [Jakarta Servlet](https://en.wikipedia.org/wiki/Jakarta_Servlet), [Jakarta Expression Language](https://en.wikipedia.org/wiki/Jakarta_Expression_Language), and [WebSocket](https://en.wikipedia.org/wiki/WebSocket) technologies. It provides a "pure Java" [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) [web server](https://en.wikipedia.org/wiki/Web_server) environment in which [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) code can also run. Thus it is a Java web application server, although not a full JEE application server. |
| 19. | JPA |  | Jakarta Persistence (JPA; formerly Java Persistence API) is a [Jakarta EE](https://en.wikipedia.org/wiki/Jakarta_EE) [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface) specification that describes the management of [relational data](https://en.wikipedia.org/wiki/Relational_data_model) in enterprise [Java](https://en.wikipedia.org/wiki/Java_(software_platform)) applications. |
| 20. | MySQL | 8.0.31 | MySQL is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). A [relational database](https://en.wikipedia.org/wiki/Relational_database) organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an [operating system](https://en.wikipedia.org/wiki/Operating_system) to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups. |
| 21. | JDBC Driver for MySQL | 8.0.31 | Java Database Connectivity is an application programming interface for the Java programming language, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. |
| 22. | Python | 3.9.13 | Python is a high-level, general-purpose programming language. Python here is used for machine learning module. |
| 23. | Flask (python library) | 2.2.3 | Flask is a micro web framework written in Python. |
| 24. | flask-cors | 3.0.10 | A Flask extension for handling Cross Origin Resource Sharing (CORS), making cross-origin AJAX possible. |
| 25. | numpy | 1.24.1 | NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. |
| 26. | pandas | 1.5.3 | pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. |
| 27. | matplotlib | 3.6.3 | Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. |
| 28. | seaborn | 0.12.2 | Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions. |
| 29. | scipy | 1.10.0 | SciPy is a free and open-source Python library used for scientific computing and technical computing. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering. |
| 30. | scikit-learn | 1.2.0 | It features various [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) algorithms including [support-vector machines](https://en.wikipedia.org/wiki/Support_vector_machine), [random forests](https://en.wikipedia.org/wiki/Random_forests), [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting), [k-means](https://en.wikipedia.org/wiki/K-means_clustering) and [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN), and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy) and [SciPy](https://en.wikipedia.org/wiki/SciPy). |
| 31. | statsmodels | 0.13.5 | statsmodels is a Python library built specifically for statistics. |
| 32. | requests | 2.28.2 | Requests is a HTTP library for the Python programming language. The goal of the project is to make HTTP requests simpler and more human-friendly. |
| 33. | simplejson | 3.18.3 | simplejson is a simple, fast, complete, correct and extensible JSON encoder and decoder for Python. |

## **Project Plan**

**1.8.1 Timelines**

|  |  |  |
| --- | --- | --- |
| Sr. No | Task | Target Date |
| **1.** | **Work Conception and Initiation** | **2nd January 2023** |
| 1.1 | Finalize test project | 9th January 2023 |
| 1.2 | Understand project | 11th January 2023 |
| 1.3 | Finalize Technology Stack | 13th January 2023 |
| 1.4 | Requirement Gathering and learning technologies | 17th January 2023 |
| 1.5 | Complete test project | 25th January 2023 |
| **2.** | **Traffic Intensity Prediction and Recommendation - Main Project** | **15th March 2023** |
| 2.1 | Research in traffic intensity prediction and factors to recommend road-widening | 12th February |
| 2.2 | Programming prediction and recommendation logic in python | 23rd February |
| 2.3 | Building prediction and recommendation UI | 15th March |
| **3.** | **User Management (UI)** | **15th April 2023** |
| 3.1 | Signup based user registration | 20th March 2023 |
| 3.1.1 | * Registration UI | 29th March 2023 |
| 3.1.2 | * Approval by Admin UI | 5th April 2023 |
| 3.2 | Non-signup based user registration | 15th April 2023 |
| 3.2.1 | * Creation of user by Admin | 15th April 2023 |
| 3. | **Testing and Deployment** | **30th April 2023** |

Table 1. Project Plan

Week1: 2nd January 2023

Week17: 15th April 2023

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | W1 | W2 | W3 | W4 | W5 | W6 | W7 | W8 | W9 | W10 | W11 | W12 | W13 | W14 | W15 | W16 | W17 |
| **Work Conception and Initiation** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Finalize problem statement** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Understand project |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Finalize Technology Stack |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Requirement Gathering and learning technologies |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Complete test project |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Traffic Intensity Prediction and Recommendation – Main Project** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Research in traffic intensity prediction and factors to recommend road-widening |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Programming prediction and recommendation logic in python |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Building prediction and recommendation UI |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **User Management (UI)** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Signup based user registration |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Registration UI |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Approval by Admin UI |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Non-signup based user registration |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creation of user by Admin |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Testing and Deployment** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Deliverables

* + 1. Draft Design Document
    2. Source Code with comments
    3. Installation Manual

# Design and Architecture

## Process Flow

There are two ways to be registered for this application, In the first way, the user who wants to use the application, makes a request to the application admin for registration to the application. The admin then process the request and based on the decision the admin will either allow or disallow the user to be registered. If allowed, the admin creates a user account and asks the user for password to complete the registration and include the user in the application. In the second way, the admin itself creates the user account and lends it to the user.

After registration is done, the main application loads where user can see and visualize the traffic data, predict on it and get traffic reducing recommendations.

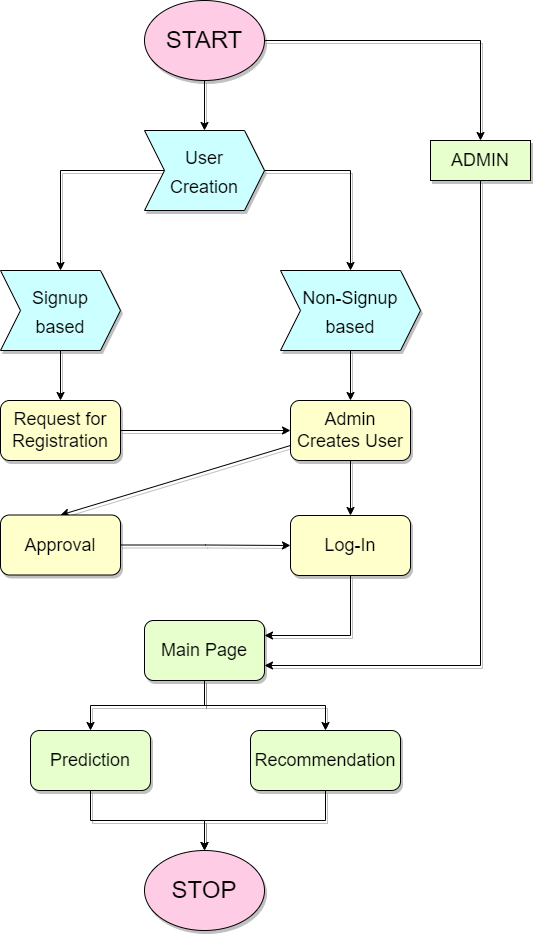


Fig 1: Process-flow diagram

## User Interface

Fig 2. Home page

Fig 3. Table View

Fig 4. Number of vehicles over time at a junction

Fig 5. Predicted values for junction 2 for next 5 days

Fig 6. Number of vehicles over time for predicted values

Fig 7. Shows accuracy of the Machine Learning Model and table for comparison of actual and predicted values

Fig 8. Comparison graph for actual and predicted values by the Machine Learning model

Fig 9. Option to download predicted results into a .csv file

* 1. **Validations**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Field / Control** | **Rule** |
| 1. | dataset\_path | The dataset\_path provided must not be empty. The user has to provide a dataset so that the application works |
| 2. | training\_ratio | The training ratio must not be empty |
| 3. | test\_ratio | The testing ratio must not be empty |
| 4. | val\_ratio | The validation ratio must not be empty |
| 5. | train\_ratio, test\_ratio, val\_ratio | All the ratios must add up to 1 |
| 6. | addOperation() | The function tries to add the data to the database and checks whether the query was successful on which the further actions proceed |
| 7. | dataset\_type | The dataset\_type is checked if it has .csv or a .xlsx or a .data extension |
| 8. | getPlot() | This function tries to get the line plot of the probability of number of vehicles at the given junction and checks if the request is completed upon which display is done |
| 9. | inputJunction | The inputJunction variable has only 5 values i.e. [‘Choose Junction’, 1, 2, 3, 4] from which only the last 4 numeric digits are allowed. |
| 10. | inputMonths | The inputMonths variable must not be empty. It is needed for prediction |
| 11. | predict() | The predict method tries to get the predicted plots and data from the backend and only when there is a response, the plots are shown |
| 12. | toggleError | If the validations on page 1 are incorrect then this variable shows the error message on the screen |
| 13. | plotReadyFor1 | shows probability plot of junction 1 when true |
| 14. | plotReadyFor2 | shows probability plot of junction 2 when true |
| 15. | plotReadyFor3 | shows probability plot of junction 3 when true |
| 16. | plotReadyFor4 | shows probability plot of junction 4 when true |
| 17. | predictionImageReady | shows the predicted plot if true |

Table 2. Validations

# Database Schema

## Database

A single database needs to be created which comprises of two tables i.e. users, which is used to store the login and signup information of the user and operation\_info which is used to store the dataset location on the test system and the training, testing, validation ratio for the continued process.

## Tables

Two tables are linked, stored and represented well in the dataset with the help of key constraints. The operation\_id table is linked to the users table with user\_id as a foreign key in operation\_id table to associate the choice of dataset the user prefers.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Extra | Remarks |
| operation\_id | int | NO | PRI | auto\_increment | Store the id of the current operation, a primary key |
| dataset\_path | varchar(255) | NO |  |  | stores the data set's path chosen |
| train\_ratio | int | NO |  |  | stores the training ratio provided by the user |
| test\_ratio | int | YES |  |  | stores the testing ratio provided by the user |
| val\_ratio | int | YES |  |  | stores the validation ratio provided by the user |
| user\_id | int | NO | MUL |  | Refers to **users(id)**, a foreign key |

(Table 3: operation\_info)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Extra | Remarks |
| id | int | NO | PRI | auto\_increment | stores the user id, a primary key, linked to operation\_info(user\_id) |
| email | varchar(255) | NO |  |  | stores the mail id of the user |
| password | varchar(255) | NO |  |  | stores the hash of the password provided by the user |

(Table 4: users)

## Entity-Relationship diagram

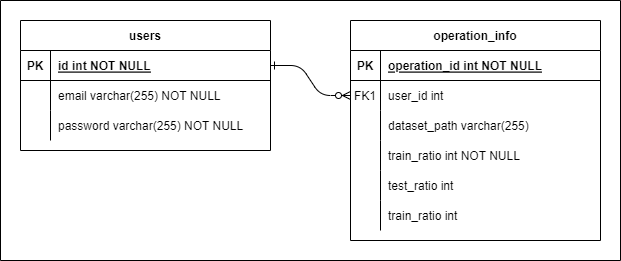


Fig 5. Entity-Relationship Diagram

## Data-Flow diagram

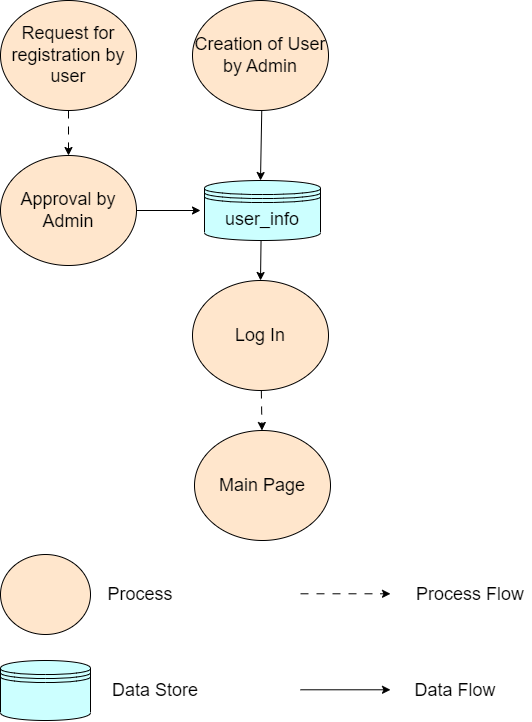


Fig 6. Data Flow Diagram