**Product Design**

|  |  |
| --- | --- |
| **Team** | 29 |
| **Project Name** | NeoEm |
| **Members** | Arushi Mittal, Meghna Mishra, Mihir Bani, Samay Kothari |

# **Design Overview**

## **Architectural design**

1. The website will serve as a portal which will connect the product created by **Ayushman Bhava** with the parties that intend to use it.
2. The device installed on emergency vehicles will detect traffic violations and upload videos as evidence for the same to be verified by the admins/authorities.  
   There will be two types of users that can use the website
   1. **Admin**: The admin will be able to access this data and check whether the considered video is a violation or not. If the admin marks it as a violation then it is reported to the traffic authorities.
   2. **Citizen**: Citizens can be able to access the portal to upload videos which they think is a potential traffic violation, the videos then uploaded are run through the machine learning model on the server side and then add it to the records.
3. The records will be stored as a video file along with some textual data, Time, Date, Location, Licence Plate of the Vehicle.

## **System interfaces**

### **User Interface**

* Registration: They will be able to register either as an admin(required to be verified by the super user) or a citizen. The user will be expected to enter, username(unique), unique verified email, mobile number, address, full name and password with optimal strength.
* Login: The user will be able to login either using username/email along with the password for their account.
* Dashboard: After login the users will be able to access the dashboard according to their role(citizen/admin).
  + For the admin dashboard:
    - The admin will be able view the records along with the accompanied details.
    - They will have the option to search, sort and filter according to the attributes which are specified for the records, ex: licence plate, date, location etc.
    - A record can be marked as a violation, shortlisted, or rejected by the admin.
  + For the citizen dashboard:
    - The citizen will be able to upload the videos of the speculated traffic violation.
    - There will also be an option to edit/delete the uploaded video.

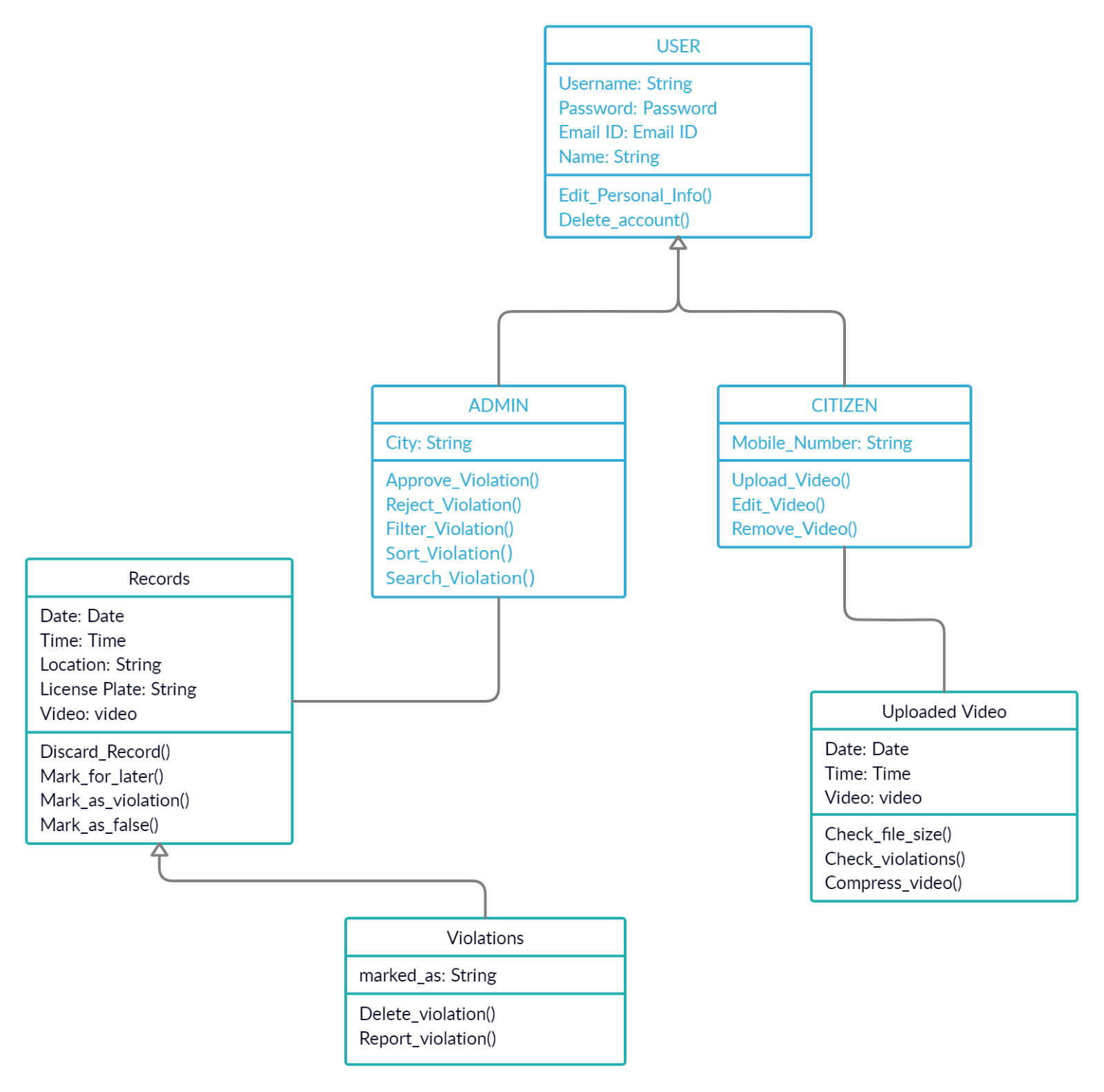
### **APIs**

1. User class:
   1. Login()
   2. Register()
   3. Email/username validation()
2. Records class:
   1. Retrieving()
   2. Marking()
   3. Shortlisting()
   4. Deleting()
   5. FuzzySearch()
3. Upload Video:
   1. Compress\_video()
   2. Check\_violations()
   3. Upload\_video()
   4. Delete\_video()

## 

## **Model**

UML Diagram:

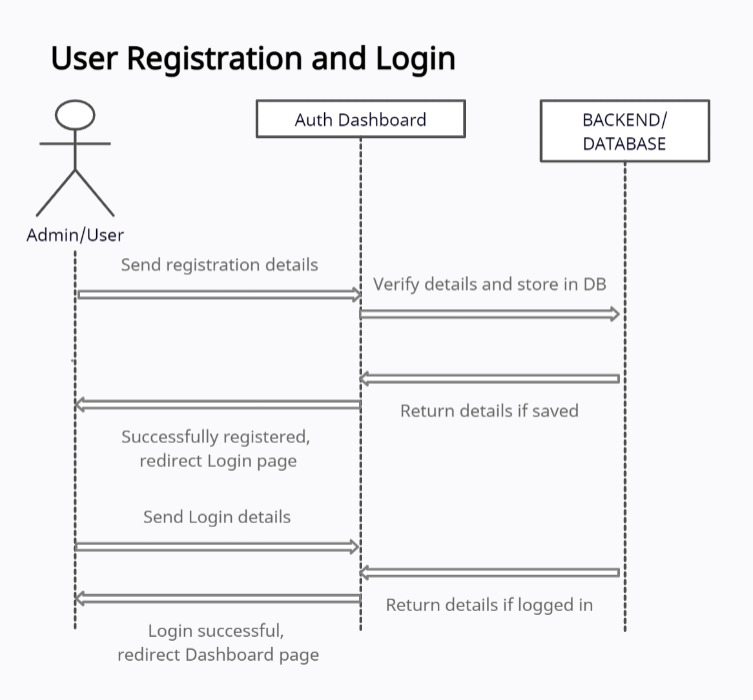


**Class:**

|  |  |
| --- | --- |
| Admin | Class State:  Stores the detail about the admin, such as city, username, email, password, email  Class Behaviour:   * Edit personal Info() * Delete Account() * ApproveViolation() * RejectViolation() * SearchViolation() |
| Citizen | Class State:  Stores the detail about the admin, such as mobile number, username, email, password, email  Class Behaviour:   * Edit personal Info() * Delete Account() * Upload Video() * Remove Video() |
| Records | Class State:  Videos of the violations along with the details such as: geo-tag, date, time, license plate and status.  Class Behaviour:   * DiscardRecord() * MarkforLater() * MarkasViolation() * ReportViolation() |
| Uploaded Videos | Class State:  Raw videos uploaded by the citizens which are to be ran through the ML model in the backend.  Class Behaviour:   * CheckfileSize() * Compress() * CheckViolation() |

.

# **Sequence Diagram(s)**



# 

# 

# 

# **Design Rationale**

Majority of our time till now was spent on annotation/ML related works instructed by the clients, upon further discussion they gave us this web app development related work.

Since we shifted to this relatively recently, we are currently sticking to the specification provided by the clients.

There haven’t been changes which are made till now to the idea and requirements to the project