**Software Design Specifications**

***Car recovR***

**Version: 1.0**

| Project Code | F224601 |
| --- | --- |
| Supervisor | Miss Syeda Rubab Jaffar |
| Co Supervisor |  |
| Project Team | Muhammad Eahtesham 19K-0298  Uzair Khan 19K-1477  Muhammad Hassaan Ahsan Qureshi 19K-0272 |
| Submission Date | 25th November, 2022 |

| **Document History**   | Version | Name of Person | Date | Description of change | | --- | --- | --- | --- | | 0.1 | Muhammad Hassaan Ahsan Qureshi | 23rd Nov 2022 | Document Initialized | | 1.0 | Uzair Khan | 24th Nov 2022 | Software Level Architecture | | 1.0 | Muhammad Eahtesham | 25th Nov 2022 | Design Strategies | | 1.0 | Muhammad Hassaan Ahsan Qureshi, Uzair Khan, Muhammad Eahtesham | 25th Nov 2022 | Document Finalized | |  |  |  |  | |  |  |  |  | |  |  |  |  |   **Distribution List**   | **Name** | **Role** | | --- | --- | | Miss Syeda Rubab Jaffar | Supervisor | |  | Co Supervisor | |  |  |   **Document Sign-Off**   | **Version** | **Sign-off Authority** | **Project Role** | **Signature** | **Sign-off Date** | | --- | --- | --- | --- | --- | | 1.0 | Miss Syeda Rubab Jaffar | Supervisor | Rubab Jaffar | Nov 25, 2022 | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

**Document Information**

| **Category** | **Information** |
| --- | --- |
| Customer | FAST-NU |
| Project | Car recovR |
| Document | Software Design Specification |
| Document Version | 1.0 |
| Status | Draft |
| Author(s) | Muhammad Hassaan Ahsan Qureshi, Muhammad Eahtesham, Uzair Khan |
| Approver(s) | Miss Syeda Rubab Jaffar |
| Issue Date | 25th November, 2022 |
| Document Location | Cloud, Google Documents |
| Distribution | Advisor  Project Coordinator’s Office (through Advisor) |

**Definition of Terms, Acronyms and Abbreviations**

| **Term** | **Description** |
| --- | --- |
| ASP | Active Server Pages |
| DD | Design Specification |
| RSU | Road Side Unit |
| CNIC | Computerized National Identity Card |
| GUI | Graphical User Interface |
| CCTV | Closed-Circuit Television |
| SCC | Secure Communication Channel |
| V2X | Vehicle to every other vehicle (cars) |

**Table of Contents**

[**1**](#_heading=h.4f1mdlm) **Introduction *5***

[*1.1*](#_heading=h.2s8eyo1) *Purpose of Document 5*

[*1.2*](#_heading=h.2u6wntf) *Intended Audience 5*

[*1.3*](#_heading=h.19c6y18) *Document Convention 6*

[*1.4*](#_heading=h.3tbugp1) *Project Overview 6*

[*1.5*](#_heading=h.28h4qwu) *Scope 6*

[**2**](#_heading=h.nmf14n) **Design Considerations *7***

[*2.1*](#_heading=h.37m2jsg) *Assumptions and Dependencies 7*

[*2.2*](#_heading=h.1mrcu09) *Risks and Volatile Areas 8*

[**3**](#_heading=h.z337ya) **System Architecture *9***

[*3.1*](#_heading=h.46r0co2) *System Level Architecture 9*

[*3.2*](#_heading=h.1y810tw) *Software Architecture 10*

[**4**](#_heading=h.2lwamvv) **Design Strategy *11***

[**5**](#_heading=h.111kx3o) **Detailed System Design *12***

[*5.1*](#_heading=h.1ci93xb) *Database Design 19*

[5.1.1](#_heading=h.3whwml4) ER Diagram *19*

[5.1.2](#_heading=h.2bn6wsx) Data Dictionary *20*

[5.1.2.1 Admin](#_heading=h.qsh70q) *20*

[5.1.2.2 User](#_heading=h.3as4poj) *20*

[5.1.2.3 Car](#_heading=h.1pxezwc) *21*

5.1.2.4 Location *21*

5.1.2.5 Complaint *22*

[*5.2*](#_heading=h.49x2ik5) *Application Design 23*

[5.2.1](#_heading=h.2p2csry) Sequence Diagram *23*

[5.2.1.1](#_heading=h.147n2zr) Car Registration *23*

[5.2.1.2](#_heading=h.3o7alnk) Complain Registration *24*

[5.2.1.3](#_heading=h.23ckvvd) Surveillance *25*

[5.2.2](#_heading=h.ihv636) State Diagram *26*

[5.2.2.1](#_heading=h.32hioqz) Car Registration *26*

[5.2.2.2](#_heading=h.1hmsyys) Complain Registration *27*

[5.2.2.3](#_heading=h.41mghml) Surveillance *28*

[**6**](#_heading=h.vx1227) **References *28***

[**7**](#_heading=h.1v1yuxt) **Appendices *28***

# Introduction

## Purpose of Document

*The purpose of this document is to provide a comprehensive overview of the design decisions made within the software project, and showcase the depth in details in terms of data, architecture, interface, procedure related design. It is developed to provide a relationship between the functional requirements and the structure, implementation, data association and modeling, and the overall architecture of this software application. It aims to do this by providing different viewpoints of the system as per the representation in need in discussion to an audience.*

*This document is written keeping in mind the target audience of primarily the FYP team, the jury, faculty at FAST. By following technical and industrial standards, we wish to make this document available to any and all software developers that would be interested in learning more about the software project. Thus, it is our aim to write this document in such a presentational format that gives way to,*

1. *Clarity over the depths of the technical details of the project (intrigue)*
2. *Gives rise to questions that would allow us to refine this document (retain feedback).*

*It is complex to answer the exact questions regarding our “design methodology” - while our approach so far with the project requirements, design decisions and requirement gathering has been based around Agile, our work has been split into concrete modules because of the 60%-40% requirement of FYP I, FYP II respectively, which seem to align with a Spiral Design Model instead.*

*How are we doing Agile? A good contrast is by aligning with the Agile Manifesto,*

1. ***Individuals and interactions over processes and tools*** *- our team communication is intensive, almost everyday, with discussions about approaches/problems/flaws/fixes, next steps and so on. We have not rendered ourselves rigid to follow a very “strict” process or on the basis of a tool, instead letting tools help us achieve our goals.*
2. ***Working software over comprehensive documentation*** *- Our focus has been on development and incremental delivery with iterations on modules. We wish to have a working, in-order, demonstrable software over documentation that just says what we’re going to do (except this documentation, of-course).*
3. ***Customer collaboration over contract negotiation*** *- we take in and value additions, idea improvements, and discussions with our supervisor, instead of noting down strict “contracts” that do not detail what is left to be desired.*
4. ***Responding To Change*** *over following a plan - we make adjustments in the requirements as per request where it is needed*

## Intended Audience

*As mentioned above, the target audience of interest are,*

1. *The FYP team itself*
2. *The Jury*
3. *Our Supervisor*
4. *Any and all potential faculty at FAST (once submitted, it will be visible to all)*
5. *Software Developers and Solution Architects (review and analysis)*

## Document Convention

*The font is “****Arial****” with font size ”****10****”*

## Project Overview

*This software project aims to provide registration, complaint registration and surveillance of traffic which can help to provide such ecosystem in which all cars are registered in a system to provide security to all the cars and if someone steals a car from such a system so the car can be recoverable in just a matter of time after launching the complaint and right after they locate in surveillance system.*

*Car recovR attempts to provide multiple features together to reduce complexity levels, high intense communication needs, and remove redundancy in information sharing, alerts, events, updates, and so on.*

*On the high level, the project attempts to (for now) bring together 3 main components together as integration pieces, which are,*

1. *Registration System*
2. *Complaint System*
3. *Surveillance System*

*These are the main high level components of the system that define the functional requirements for the requirement process. All together, these components encapsulate entities logic for the detection system, registration process, complaint process and surveillance process (and more).*

*As stated before in “Purpose Of Document”, we would like to use the Agile approach for building the software*

## Scope

*Documentation is not in scope of the project. We do not intend to write code or functionalities that serve as “User Documentation” - that is, helping users navigate the application. Since there is the idea of an MVP, we wish to develop features and functionalities with greater efficiency and accuracy to fulfill our committed requirements and to perform operations smoothly and perfectly, rather than spending too much time on writing “comprehensive” documentation which is not the aim of an FYP anyways.*

*The system can be tested or analyzed in a simulation based environment.*

*The system will not integrate multiple login options, only working with a main email/password schema.*

*The system will not be based on real time surveillance as camera access and cameras are not available and there are no RSUs in the country.*

*The official recovery procedure is not the scope of the project.*

# Design Considerations

*The system should be in all cases reliable, robust, adaptive, and flexible for development and use purposes. For this intent, we spent a considerable amount of time trying to work towards a better design flow and structure for the project, in order to solve major problems that existed with the design of the system on a finer-grained level.*

*The issues related to design were,*

1. *What is the architecture we would like to develop our front-end and back-end on? What frameworks/libraries we believe align with the workflow we have and how would we like to break the project down into smaller components*
2. *How would we go about solving the remote code execution problem and what tools would we use to enable real-time collaboration? It is a niche and difficult problem to solve - what approaches could we use to counter these problems?*
3. *What would we use to build rapid UI, and how would we manage user states and sessions on the frontend?*
4. *How would we deal with authentication?*
5. *How would we talk to the database?*

*A few other affecting consideration can be about,*

1. ***Compatibility -*** *since it is a web application, we care about compatibility on different browsers enough so that it does not break for our users.*
2. ***Modularity -*** *modules are designed to be well-defined, independent components which would lead to better maintainability. Dependencies are clearly defined where needed and are well-intentioned and well-contained.*
3. ***Fault-tolerance -*** *the application is not yet planned to be beyond local development environments, so this consideration does not really apply.*
4. ***Maintainability -*** *the codebase can be broken down into modules, so bugs if any are module-contained, easy to trace and detect.*
5. ***Reliability -*** *because we are working on the local environment, it is easy to check for reliability for the software to do as it is expected within stated conditions for a specified period of time*
6. ***Reusability -*** *we are using already developed software and packages where we believe we can use the functionality that they provide us, reducing our time to develop from scratch.*
7. ***Robustness -*** *that the software operates under stress of users and inputs or tolerate unpredictable or invalid input.*
8. ***Security -*** *the software is able to withstand and resist hostile acts and influences.*
9. ***Usability -*** *the interface is designed to be simple and easy to use without the need to involve user documentation manuals.*
10. ***Performance -*** *the software will be usable across a number of platforms of mobile, desktop, and laptops.*
11. ***Portability -*** *since it is a web application, the software is perfectly portable.*
12. ***Scalability -*** *the software adapts well to increasing data or added features or number of users*

## Assumptions and Dependencies

*As per concern with the team’s idea of system and software design, we imagine design to be concerned with data modeling, entities, abstractions, architectural patterns, and strategies (design models being a strategy to achieve those goals). Our sense of assumptions encapsulates notions about the project’s timeline, deliverables, and the workflow that we assume will be in place, in order to help the project run but cannot be guaranteed.*

*It is a want to have those assumptions to never be proven wrong, otherwise there will be a direct impact on the project.*

*And so, what were our assumptions starting FYP I. They were as the following,*

1. *We would be able to finish all of the requirement engineering and analysis before Mid I (FYP defense) in order to not stumble into unforeseen territory and have a team consensus on what must be done and what should be left for now, as per a team timeline.*
2. *We would be able to finish our Registration module completely, all of the backend as well as the integration with the user flow before Mid 2.*
3. *We would be able to finish all of the Complaint System prior to the finals completely, all of the back-end, as well as the integration with the user flow before the finals.*
4. *We will not have any changes and will be fixated on a set of requirements at the start of the semester.*
5. *We would be able to manage personal time to handle this large project as per our expectations - we realize realistically and responsibly that this project is of large scope and cannot be implemented here in Pakistan but it’s importance is gigantic, and that missing deadlines would keep spiraling into other deadlines.*

*For each of the above assumptions, many were altered,*

1. *A change occurred after the FYP defense where we were asked to implement frontend in FYP-I deliverable and we have a plan to develop frontend after FYP-I as we have extensive work to do on backend. Moreover, we were also asked to remove the idea of blockchain from the project and add pure cybersecurity aspects for security concerns such as encryption, digital signatures and self signed certificates,*
2. *While most of our work related to the Registration module was completed before Mid 2 in terms of back-end, we could not work on integration and have been working on it since.*
3. *A bit of uncertainty with data modeling led us to exploring Remote Code Execution, Real Time Collaboration solutions, and we were generally unwilling to start implementation until our defense was approved. We were scared of “what if this does not get approved?”, or “what if we have to make changes at the core of requirements?”. While these fears were not of fruition, it led the team to realize document related deadlines/timelines and how much they can affect a project’s progress.*

*For dependencies, we were blocked by a few factors we were waiting on/for to pass,*

1. *FYP Defense - was the reason that we could not start work prior to the defense, and if we had the opportunity of getting our project approved prior to the fixed date, we might have taken that option.*
2. *Data modeling had to be altered at different times to cater to requirements*

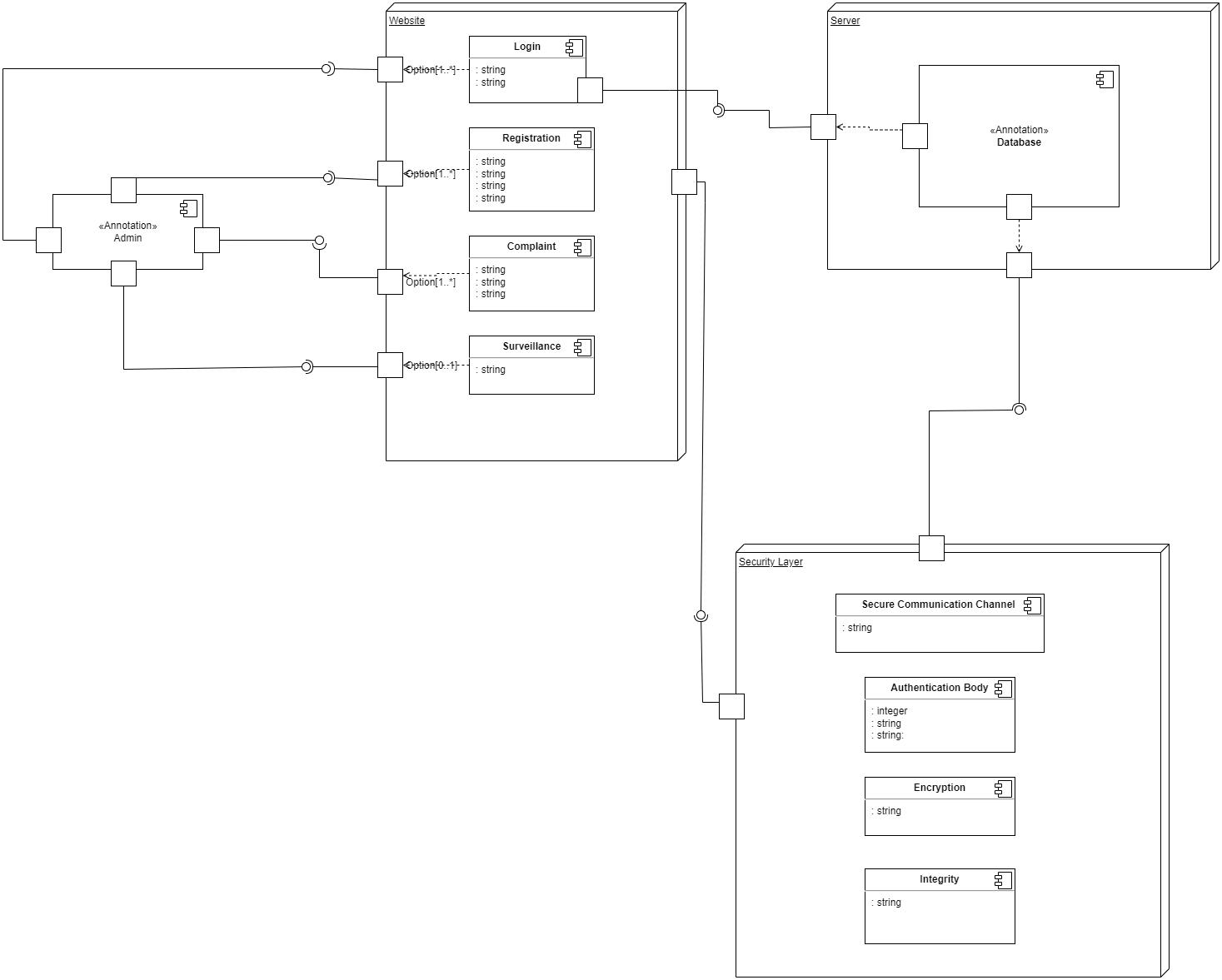
## Risks and Volatile Areas

*There is little to no time to consider, implement, and practice a contingency path. As such, we should not be facing any further changes in the requirements at the moment and our choice of technology is likely to remain consistent throughout the project - we have not faced any conditions where we would not use our current set of tools and implement functionalities further with those.*

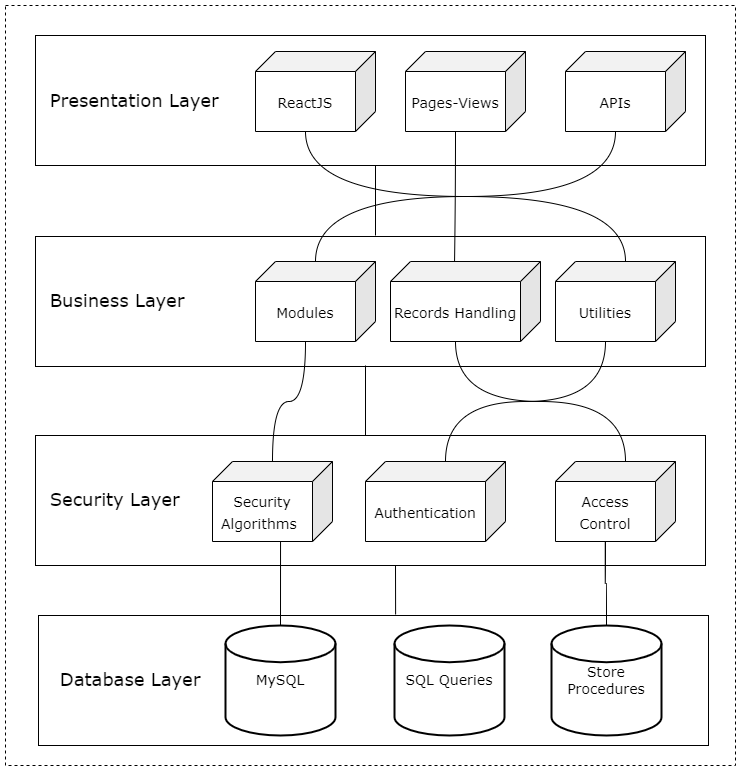
*It is also important to note that any further developments and requirements would further alter our timeline as altered it already is. If any developments do arise, our initial approach would be as always to understand if it’s a priority, and if so, when does it become a priority to get done first.*

# System Architecture

## System Level Architecture

**

## 3.2 Software Architecture

**

# Design Strategy

*The frontend of the website is done using the React framework which is a Javascript library as it allows developers to easily create fast user interfaces for websites and applications. It is a user-friendly and highly compelling framework that provides a number of possibilities to the developers to make them more creative.*

*Why React?*

* *React is a component-based framework that manages their own state, then composes them to make complex UIs.*
* *The JS library consists of several functions including one that converts the HTML components into required functions and transforms the entire project so that it is easy to understand.*
* *Provides reusability i.e complex and detailed components can further be divided into sub components. Also, one can develop new features in React without rewriting existing code.*
* *Provides simplicity through one-way data binding. This means that absolutely anyone can track all the changes made to any particular segment of the data.*

*The backend and all its functionalities are designed in Django REST framework Django REST framework is an open source, mature and well supported Python/Django library that aims at building sophisticated web APIs. In regards to functionality, we use the Python language to design the functions therefore to successfully enable these functionality we use django.*

*Reason for using Django REST;*

* *Simplicity, flexibility, quality, and test coverage of source code.*
* *Powerful serialization engine compatible with both ORM and non-ORM data sources.*
* *Generic classes for CRUD operations*
* *Clean, simple, views for Resources, using Djago’s new class based views.*
* *HTTP response handling, content-type are easily changed using HTTP Accept headers.*

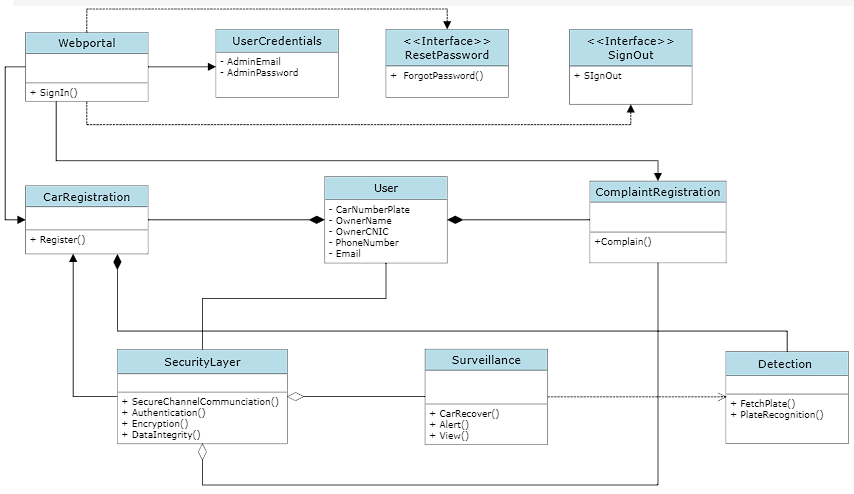
*For the database storage, the data will be stored in the platform named MySql as it securely stores the data and prevents data from data breach and the framework will be SQLite. SQLite is a C-language library that implements a small, fast, self-contained, high reliability, full featured SQL database engine. SQLite is the most used database engine in the world.*

*The software architecture used will be Layered Architecture which is the most common pattern otherwise known as n-tier architecture. Components within the layered architecture pattern are organized into horizontal layers, each layer performing a specific role within the application (e.g., presentation logic or business logic)*

*One of the powerful features of the layered architecture pattern is the separation of concerns among components. Components within a specific layer deal only with logic that pertains to that layer. For example, components in the presentation layer deal only with presentation logic, whereas components residing in the business layer deal only with business logic.*

# Detailed System Design

***Class Diagram***

**

*The attributes, functions or methods specifying interaction between different classes are as follows:*

*1- Web Portal*

* *SignIn() , this function allow users to log in to the system software. it contain user credentials like*

*- email address and password.*

*This function also include a*

*- ForgotPassword() method which helps in reset of password if user forget it.*

*- SignOut(), this function allows user to log out from the system software.*

*2- CarRegistration*

* *Register() , this function allow login user to register the cars by providing following attributes listed below,*

*- CarNumberPlate*

*- OwnerName*

*- OwnerCNIC*

*- PhoneNumber*

*- Email*

*3- ComplainRegistration*

* *Complain() , this function allows the user to file a complaint of the stolen car by providing attributes mentioned above in the Register () class.*

*4- Surveillance*

* *Car Recover() , this function helps in recovering stolen cars through inspection of the number plate.*
* *Alert() , this function makes an alert response to the user when a stolen or fake registered car is detected.*

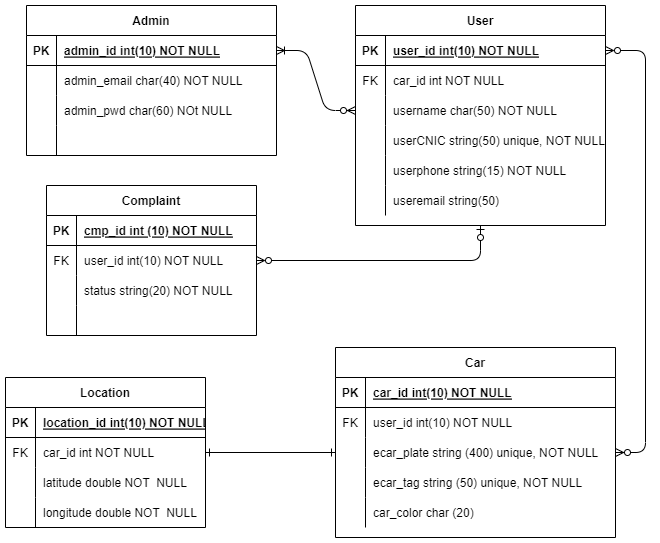
*5- Detection*

* *FetchPlate() , this function helps in extracting the number plate from the image being found as the stolen car image.*
* *PlateRecognition() , this function helps in detecting the number plate using a detection algorithm from the images given.*

*6- Security Layer*

* *SecureChannelCommunication() , this function helps in building the secure channel by transferring the keys securely using public key encryption.*
* *Authentication() , this function provides authentication of both the user and web application.*
* *Encryption() , this function is used to perform encryption on data been stored in the database.*
* *DataIntegrity() , this function helps in preventing altering the original data being sent from the web portal to the server.*

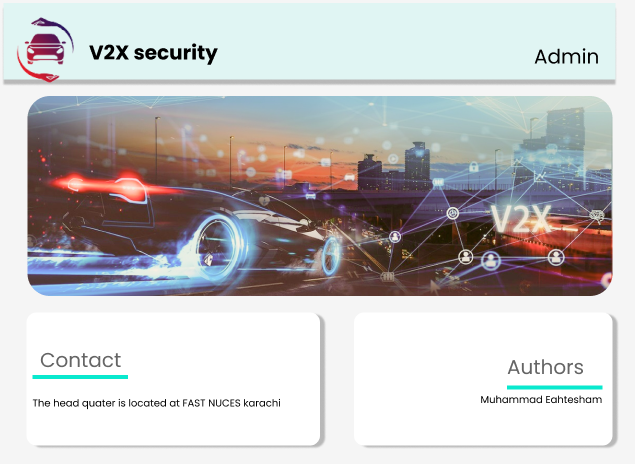
***ER Model***

**

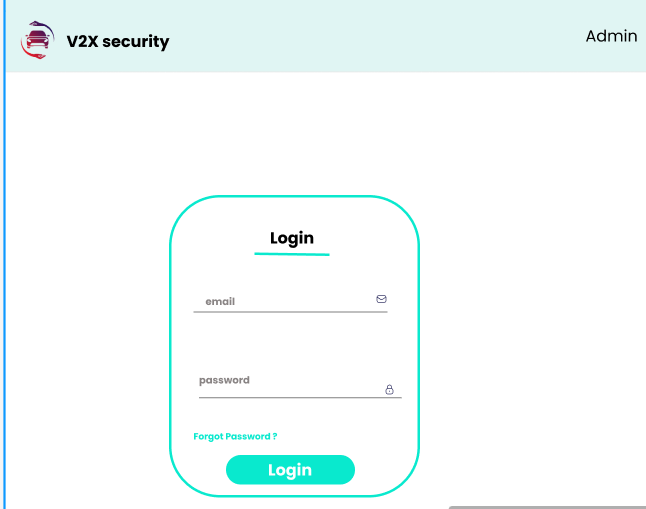
***Detailed Graphical User Interface***

*The GUI is volatile. The below wireframes are only “visual perspectives”, that is, how something may look like, instead of being exactly the same as the wireframes.*

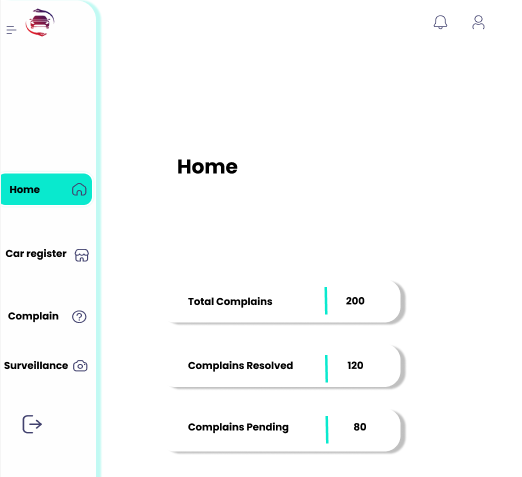
1. ***Dashboard***

**

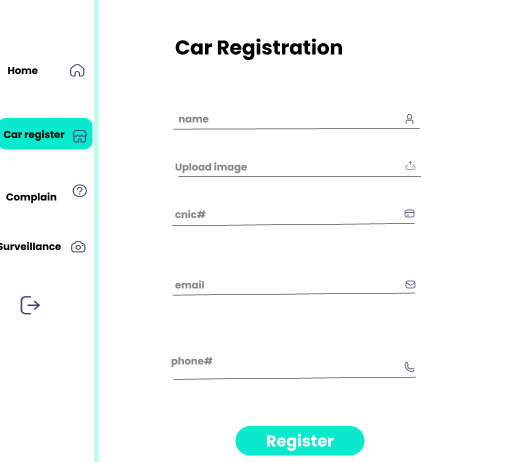
1. ***Admin login page***

**

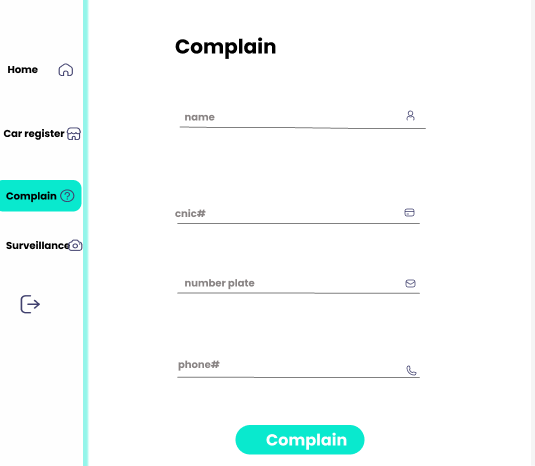
1. ***Admin Homepage***

**

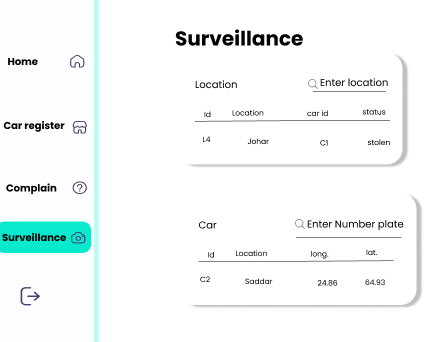
1. ***Car registration***

**

1. ***Complain***

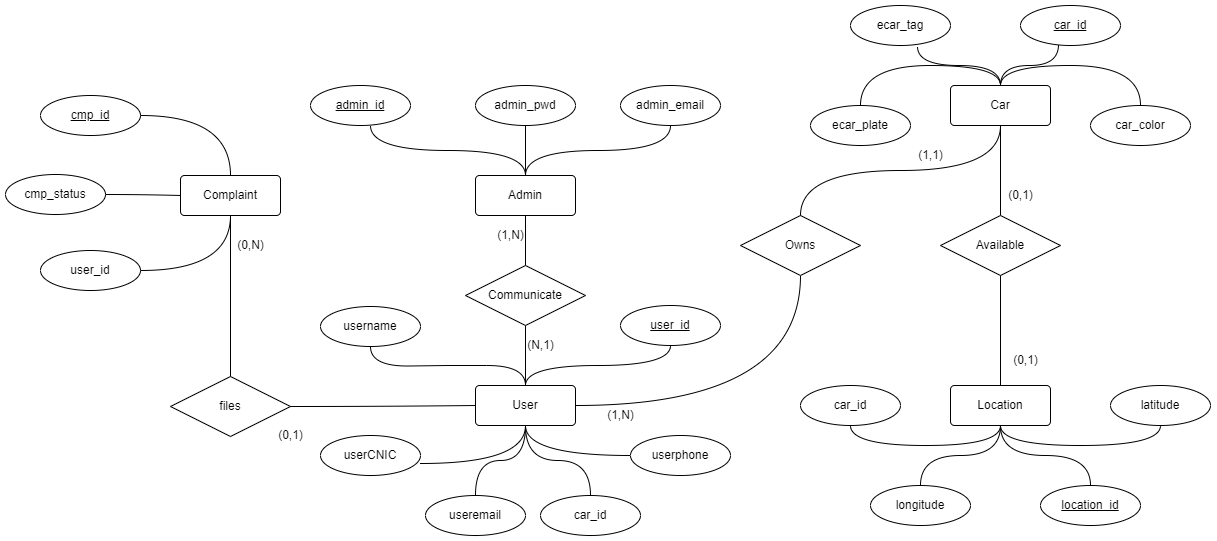
**

1. ***Surveillance***

**

## Database Design

### ER Diagram

**

*In the entity relationship diagram, it is been included that,*

* *User as an entity which contains the attributes like User\_id(PK), Username,UserCNIC, Userphone, Useremail, Car\_id(FK)*
* *Another entity named Admin which contains the attributes named admin\_id(PK), admin\_email, admin\_password.*
* *These two entities have an one to many relationship i.e. users are communicating to one admin at a time.*
* *Furthermore, entity named Car contains attributes like ecar\_plate, Ecar\_tag, car\_id(FK), car\_color.*
* *Entity named CAr and User has a relation i.e. user owns a car*
* *Locations as entities have location\_id(PK), longitude, car\_id(FK), latitude. Location entity help in detecting the car availability on the specific location i.e. car can be found or not on the location being monitored*
* *Complaint entities with cmp\_id(PK), cmp\_status, user\_id(FK), have a relationship with user for filing none or many complaints of the stolen car.*

### Data Dictionary

### **5.1.2.1** Admin

| **Admin** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | | Admin | | | | | |
| **Alias** | | N/A | | | | | |
| **Where-used/how-used** | | Used for performing activities like complain register, registration of cars | | | | | |
| **Content description** | | = | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| *admin\_id* | *the admin identity* | | *integer* | *10* | *NA* | *NA* | *Primary Key* |
| *admin\_email* | *the email of the admin* | | *character* | *40* | *NA* | *NA* |  |
| *admin\_pwd* | *the password to mail register to admin* | | *character* | *60* | *NA* | *NA* |  |

### 5.1.2.2 User

| **User** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | | User | | | | | |
| **Alias** | | N/A | | | | | |
| **Where-used/how-used** | | Used in response from the admin | | | | | |
| **Content description** | | = | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| *user\_id* | *the user identity* | | *integer* | *10* | *NA* | *NA* | *Primary Key* |
| *username* | *the username of the user* | | *character* | *50* | *NA* | *NA* |  |
| *user\_email* | *the email address of the mail* | | *string* | *50* |  | *NA* |  |
| *user\_phone\_number* | *the phone number of the user* | | *string* | *15* | *NA* | *NA* |  |
| *user\_CNIC* | *CNIC of the user* | | *string* | *50* | *NA* | *NA* |  |
| *car\_id* | *the car identity* | | *integer* | *10* | *NA* | *NA* | *Foreign Key* |

### 5.1.2.3 Car

| **Car** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | | Car | | | | | |
| **Alias** | | N/A | | | | | |
| **Where-used/how-used** | | Used in the recovery of the stolen car and register of cars | | | | | |
| **Content description** | | = | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| *car\_id* | *the car identity* | | *integer* | *10* | *NA* | *NA* | *Primary Key* |
| *e\_car\_plate* | *the number plate of the car* | | *string* | *400* | *NA* | *NA* |  |
| *e\_car\_tag* | *tag assigned to each car number plate* | | *string* | *50* | *NA* | *NA* |  |
| *car color* | *the color of the car* | | *character* | *20* |  | *NA* |  |
| *user\_id* | *the user identity* | | *integer* | *10* | *NA* | *NA* | *Foreign key* |

### 5.1.2.4 Location

| **Location** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | | Location | | | | | |
| **Alias** | | N/A | | | | | |
| **Where-used/how-used** | | Used in getting the location of the car been seen | | | | | |
| **Content description** | | = | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| *location\_id* | *the location identity* | | *integer* | *10* | *NA* | *NA* | *Primary Key* |
| *latitude* | *the latitude of the location* | | *double* | *50* | *NA* | *NA* |  |
| *longitude* | *the longitude of the location* | | *double* | *50* | *NA* | *NA* |  |
| *car\_id* | *the car identity* | | *integer* | *10* | *NA* | *NA* | *Foreign Key* |

### 5.1.2.5 Complaint

| **Complaint** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | | Complaint | | | | | |
| **Alias** | | N/A | | | | | |
| **Where-used/how-used** | | Used in filing complain of stolen car | | | | | |
| **Content description** | | = | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| *cmp\_id* | *the complain identity* | | *integer* | *10* | *NA* | *NA* | *Primary Key* |
| *user\_id* | *the user identity* | | *integer* | *10* | *NA* | *NA* | *Foreign Key* |
| *status* | *the complain status* | | *string* | *20* | *NA* | *NA* |  |

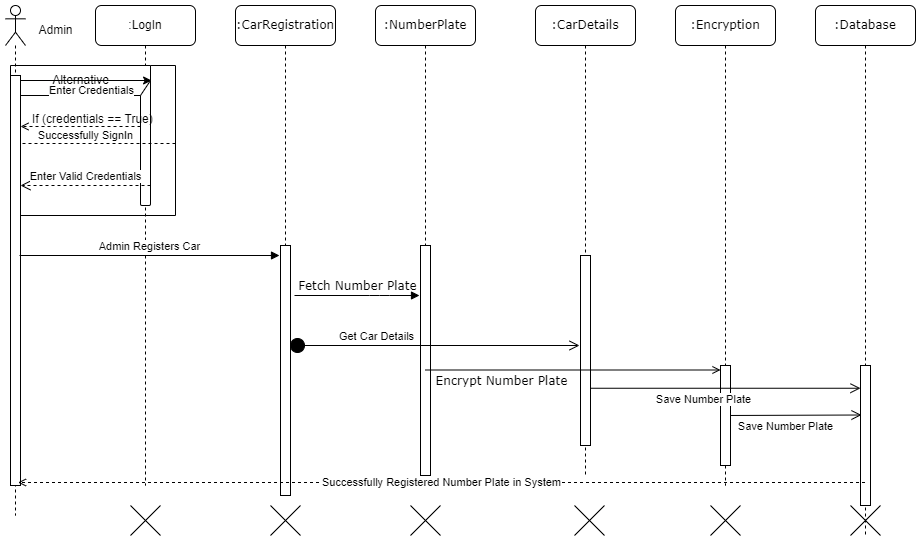
*The notation to develop content description is given below:*

| ***Data construct*** | ***Notation*** | ***Meaning*** |
| --- | --- | --- |
|  |  |  |
|  | *=* | *is composed of* |
| *Sequence* | *+* | *And* |
| *Selection* | *[|]* | *either-or* |
| *Repetition* | *{}n* | *n repetitions of* |
|  | *( )* | *optional data* |
|  | *\* … \** | *delimits comments* |
| ] |  |  |
|  |  |  |
|  |  |  |

## Application Design

### Sequence Diagram

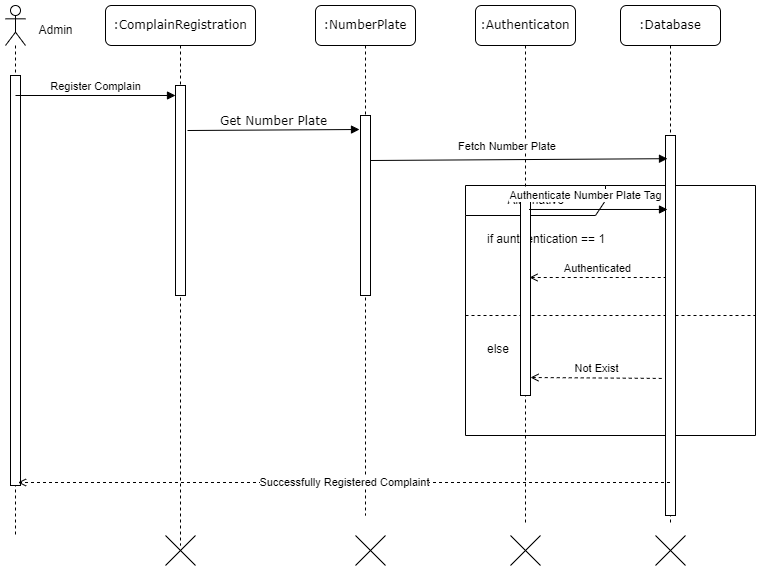
#### Car Registration

**

*This sequence diagram depicts the overall registration process.*

* *After login, the admin submits the registration form.*
* *The backend then fetch number plate , encrypt it*
* *The encrypted number plate and all the information is saved in the database*
* *The system also deals the wrong credentials i.e on incorrect credentials, it will request the user to enter a valid credentials*

#### Complain Registration



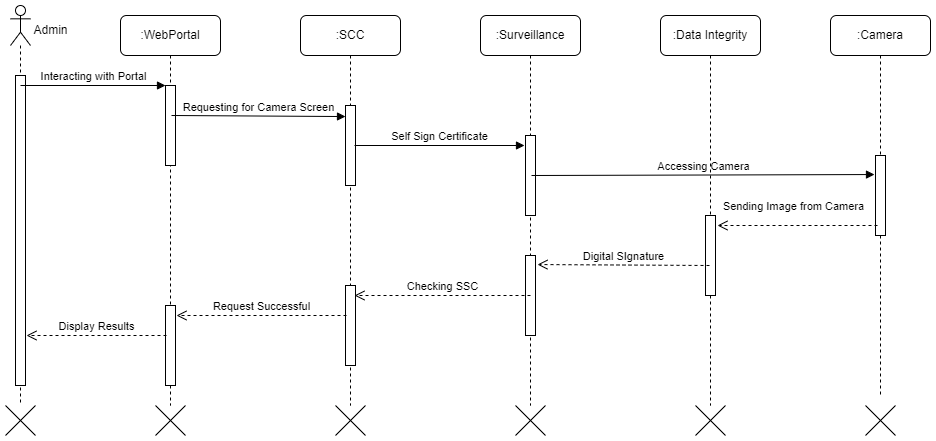
*This sequence diagram depicts the overall complaint registration process.*

* *Firstly, the admin registered the complaint in the complaint module.*
* *But for registering a complaint of the stolen car, admin has to have stolen car number plate against which action should be taken.*
* *Then, it is being verified that if the given number plate of stolen car belongs to the database of registered cars then*

*- if the car belongs to the database, successfully register a complaint of the stolen car.*

*- if not, unsuccessfully by giving a response as the car is not registered or exists.*

#### Surveillance



*This sequence diagram depicts the overall response/monitor from the camera to the web portal.*

* *Admin is interacting with the surveillance of a web portal to see the result of the camera.*
* *to communicate with the camera, web portal must perform some security infrastructure i.e.*

*- self-signed certificate to authenticate that its the registered web portal who is trying to*

*communicate with the camera.*

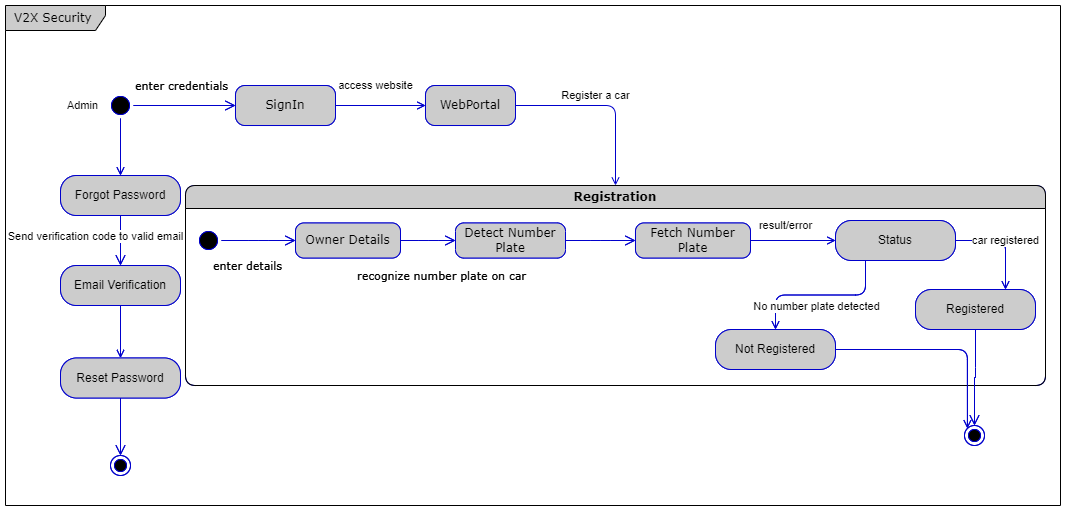
*- In response to the web portal,image were sent in the data format from the camera to the web*

*portal. In order to prevent integrity of data, data is assigned with the digital signatures.*

*- It shows the result in the web portal after successful transfer of data.*

### State Diagram

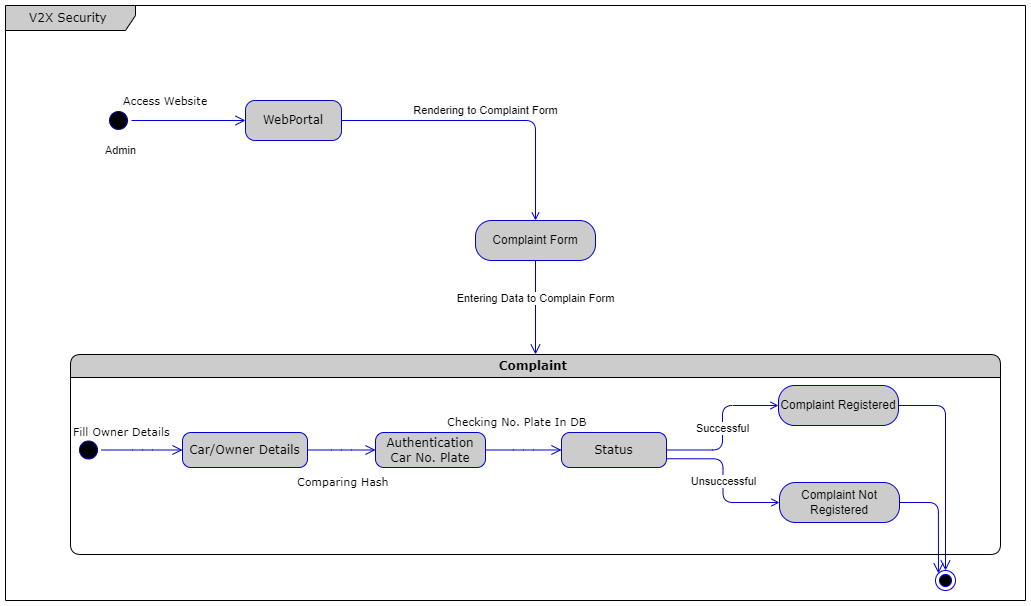
#### Car Registration

**

*The above state diagram illustrate all the states of a car registration;*

* *Admin enters the credentials in the login page .*
* *On successful login;*
  + *For registration, the owner details are provided through a web form.*
  + *The backend detects the textual number plate from an image and fetch it.*
  + *Then it decides whether to save all the information in the database or not. The decision depends upon the text extraction .*
  + *If the number plate is extracted, the data is then stored in the database*
  + *If the image is not clear or if the captured image does not contain number plate, then the data will not be stored in the database*
* *If the admin forget his/her password;*
  + *The system then ask for email to verify/authenticate and the system resets the password*

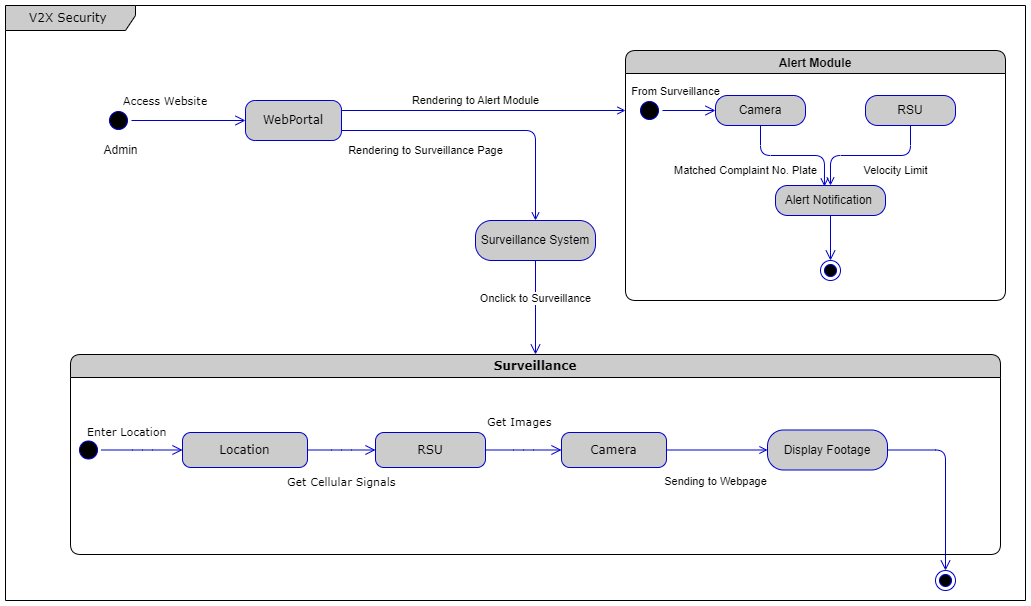
#### Complain Registration



*The above state diagram illustrate all the states of filing a complaint;*

* *The admin access the website through login process and goes to the complain section*
* *Data will be entered in the complaint form.*
* *Backend then authenticates the car number plate.*
* *If the number plate matches the number plate of the database, the complaint is then registered.*
* *On an unsuccessful match, a complaint will not be launched.*

#### Surveillance

**

*The above state diagram illustrate all the states of surveillance module;*

* *To reach a surveillance state, admin access the website through login process and goes to the surveillance section*
* *To view the information of a particular location, the admin enters the location name*
* *A table appears containing the information of all the location which the backend fetches from the images captured by the camera from RSU*
* *Alert module will be launched, i.e system will send the notification to the admin if the number plate is matched with the complaint number plate.*

***6*** **References**

*H. Alasmary and colleagues, “Analyzing and Detecting Emerging Internet of Things Malware: A Graph-Based Approach” IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8977-8988, October 2019, Doi: 10.1109/JIOT.2019.2925929.*

# 7 Appendices

*Not Applicable*