**Software Requirement 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 | Nov 10, 2022 | Document Created |
| 1.0 | Muhammad Eahtesham, Uzair Khan, Muhammad Hassaan Ahsan Qureshi | Nov 25, 2022 | Document Finalized |

Distribution List

| **Name** | **Role** | |
| --- | --- | --- |
| Miss Syeda Rubab Jaffar | | Supervisor |
|  | | Co- Supervisor |

Document Sign-Off

| **Version** | **Sign-off Authority** | **Sign-off Date** |
| --- | --- | --- |
| 1.0 | Miss Syeda Rubab Jaffar | Nov 25, 2022 |
|  |  |  |

**Table of Contents**

[1.](#_heading=h.30j0zll) Introduction 6

[1.1.](#_heading=h.3znysh7) Purpose of Document 6

[1.2.](#_heading=h.2et92p0) Intended Audience 6  
1.3 Abbreviations 6

[1.4.](#_heading=h.tyjcwt) Document Convention 6

[2.](#_heading=h.3dy6vkm) Overall System Description 7

[2.1.](#_heading=h.1t3h5sf) Project Background 7

[2.2.](#_heading=h.4d34og8) Project Scope 7

[2.3.](#_heading=h.2s8eyo1) Not In Scope 7

[2.4.](#_heading=h.17dp8vu) Project Objectives 7

[2.5.](#_heading=h.3rdcrjn) Stakeholders 8

[2.6.](#_heading=h.26in1rg) Operating Environment 8

[2.7.](#_heading=h.35nkun2) System Constraints 8

[2.8.](#_heading=h.1ksv4uv) Assumptions & Dependencies 8

[3.](#_heading=h.2jxsxqh) External Interface Requirements 9

[3.1.](#_heading=h.z337ya) Hardware Interfaces 9

[3.2.](#_heading=h.3j2qqm3) Software Interfaces 10

[3.3.](#_heading=h.1y810tw) Communications Interfaces 10

[4.](#_heading=h.4i7ojhp) Functional Requirements 11

[4.1.](#_heading=h.2xcytpi) Functional Hierarchy 11

[4.2.](#_heading=h.1ci93xb) Use Cases 12

[4.2.1.](#_heading=h.3whwml4) Car Registration 13

4.2.2 Complaint Registration 16

4.2.3 Car Recovery 18

[5.](#_heading=h.2bn6wsx) Non-functional Requirements 21

[5.1.](#_heading=h.qsh70q) Performance Requirements 21

[5.2.](#_heading=h.3as4poj) Safety Requirements 21

[5.3.](#_heading=h.1pxezwc) Security Requirements 22

[5.4.](#_heading=h.49x2ik5) User Documentation 22

[6.](#_heading=h.147n2zr) References 23

[7.](#_heading=h.3o7alnk) Appendices 23

1. **Introduction**

* 1. **Purpose of Document**

The SRS is the document that describes the nature of the software. It is a document that describes what the software will do and how it will be expected to perform. It is used to collect and analyze all sorted ideas that come up to define the software, its requirement against consumers. It is the complete specification and description of requirements of software.

* 1. **Intended Audience**

The document is intended for faculty members such as FYP coordinators, FYP supervisor, and FYP developers, to understand the idea of the product and all technical details behind it.

**1.3 Abbreviations**

RSU = Road Side Unit

CCTV = Closed Circuit Television

CNIC = Computerized National Identity Card

SCC = Secure Communication Channel

* 1. **Document Convention**

Document is written in Arial with font size = 10

1. **Overall System Description**
   1. **Project Background**

Street crimes, as one of the most common-seen malefactions, poses huge threats to the individual’s safety as well as the social stability and is increasing day by day in our society. To be more specific, there has been an increasing interest in the emerging trends of stealing cars which then being used in mugging incidents and can also use in bomb blasting or other serious crimes and wont be recover. Preventions were made but no proper solution regarding the recovery i.e., recovery rate has dropped significantly as these cars then move within an hour in other cities/territory and government officers are unable to track them down. In light of this issue, the team proposes a solution with the help of artificial intelligence and cybersecurity and desire to build a car recovery portal that will be used by the provincial governments’ security forces and officials to share the details and information regarding the unauthorized or stolen car with each other.

* 1. **Project Scope**

Team will analyze the tasks and produce minimal required documents, estimate the risks and milestone chart based on which team will agree on the final scope. A detailed description of deliverables are:

For FYP-1, the deliverables include,

1- Requirement Analysis

2- System design

3- Data training and data modeling

4- Registration setup

5- Car Recovery Portal

For FYP-2, the deliverables include,

1- Building Secure communication channel

2- Vehicle to everything setup

3- FYP final touches

* 1. **Not In Scope**

The software system does not aim to divert from the initial requirements because of the time complexity and workable load. There is no hardware implementation of the software system as the car recovery portal only helps users by generating alerts regarding the stolen cars and displaying the location to the user where it was last seen, not in the real time recovery of the car.

* 1. **Project Objectives**

Building a portal that will help authorities of the provinces to recover the stolen car through information sharing with cybersecurity. It helps in detecting and alerting the unauthorized car from the registered cars in the system and also for duplicate number plate cars.These will be features which help in providing the solution for stolen car recovery problem.

* 1. **Stakeholders**

First of all the customers such as employees of regulatory bodies, law enforcement agencies who engaged with the application and are the information providers of the stolen cars. The FYP members are responsible for timely software delivery and estimation.

* 1. **Operating Environment**

The client-side and server-side components of the software system must operate within common web browser environments using Secure Sockets Layer (SSL) / cryptographic protocols at a minimum encryption level of 256 bits / digital signatures for monitoring data integrity. Operating system will be Windows. For storing data, mysql will be used in the software system. The system is built on the platform such as React, Django, and Python.

* 1. **System Constraints**
* **Software constraints**

The application will use only the React framework of JavaScript and Django as the main web technologies. A basic general knowledge of computer skills is required to use this software system.

* **Hardware constraints**

It does not have fixed hardware requirements. It can be handle in most of the hardware infrastructure whether it is of laptop, mobile etc

* **Cultural constraints** (includes language etc.)

The application is not multilingual. It only caters with the English language.

* **Legal constraints**

Ensuring security (authentication, secure communication channel) like digital certificates, digital signatures, encryption algorithms for both the user and its data for safe environment and its the main purpose of our project for preventing attacks.

* **Environmental constraints** (e.g., the environment where the software will be installed, It could be a noisy environment, which may require that there is no sound event in the project).

Since it is a web-based application, the internet connection must be established as it functions via the internet.

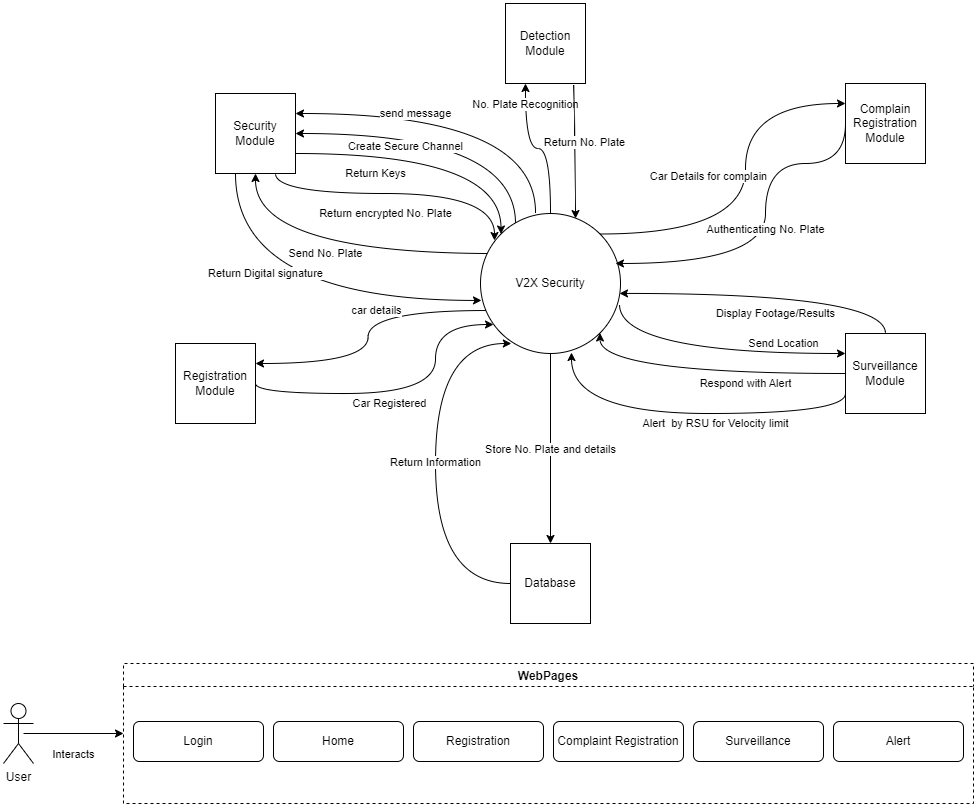
* **User constraints** (e.g., the project is developed for children, so it may be required that the project has more graphic controls rather than textual controls).

The application is developed for agencies like the police department therefore inexperienced users won't be able to use it. Each center/agency user has an account created and authenticated.

* 1. **Assumptions & Dependencies**

Each center admin has an account created and authenticated. The application will undertake all the complaints regarding cars if it is stored in the registered database or if stolen within the premises of the registered database. External or beyond premises stolen cars will not be able to register complaints. Stolen cars should have the number plate attached as it is the main attribute for functioning for stolen cars. Application does not undertake the real time streaming or detecting of cars through CCTV cameras, only extracting the data using the images already stored in the storage. Our dependencies will rely on the packages we use for the development of the application.

1. **External Interface Requirements**



* 1. **Hardware Interfaces**

As such, there is not really a place for hardware interfaces currently as all of the system lies entirely on the developer’s machine for now. Thus, as a result, the client will only need a web browser to view the pages on-demand. The only hardware in use is just the machine that the user is using to run the web application.

Moreover, there are two main hardware components that will interact with the software interfaces.

1. Road Side Unit
2. CCTV Camera

These two devices only be used in a real time environment but the project is purely based on simulation.

* 1. **Software Interfaces**

There are 4 primary flows of data between the system,

1. **Services** - Services provide the backend logic needed to connect with the database (MySQL) and offer modular services to the pages that need them
2. **API**- Calls the back-end resources on the front-end and to bring applications together in order to perform a designed function built around sharing data and executing pre-defined processes.
3. **Pages** - Pages are the views that the user actually sees.
4. **Components** - Components are the parts of a view that are developed individually and integrated with each of the pages to provide a complete view of the page.

Externally, we are depending on NPM packages to support our application. They help in reuse as in a library, but do not exactly act as a way for us to store the data. All the data is stored on the database. Since the library does not store or transfer data (to make a data flow), they are not well suited here in these diagrams.

The database in question is MySQL, connected with Django.

There are no commercial components as of such.

Each of the services either talks to other services OR the APIs to provide services/responses based on the requests given to it. The purpose of these data items is to provide business logic on how to deal with incoming requests and to manage the transfer of data between the database and the actual views.

The services needed are described in the diagram above:

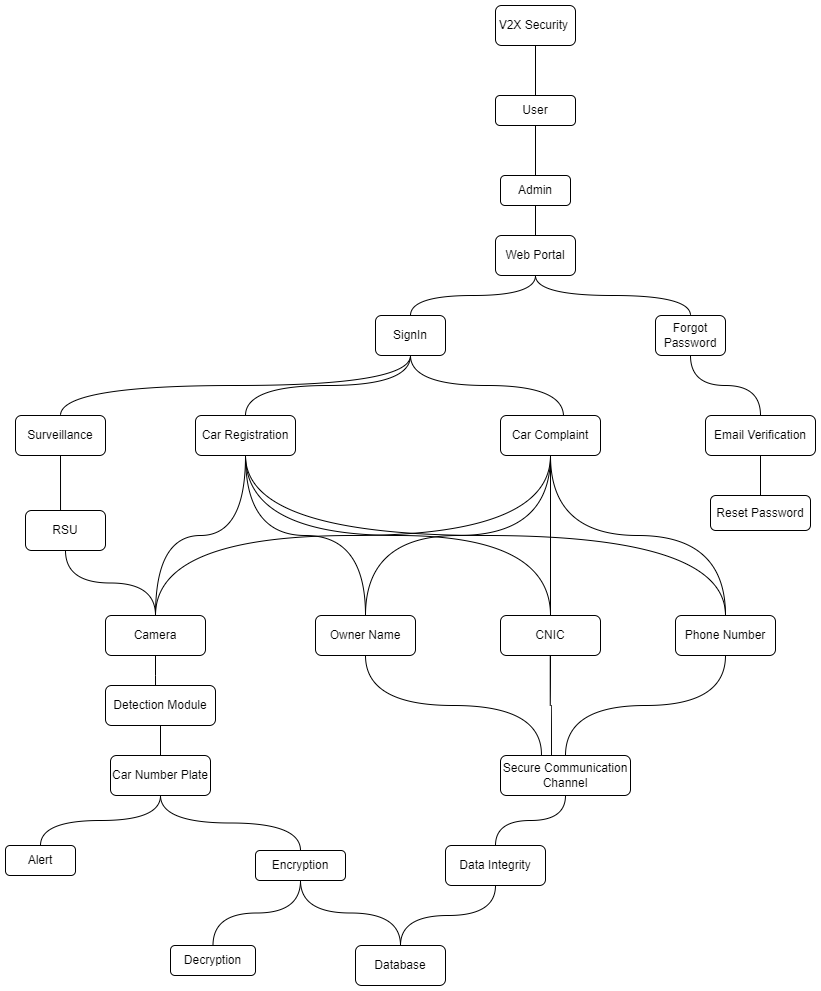
1. **Detection Module -** For all services related to detection of number plates, recognition of number plates, fetching number plates as string and send the data back to the registration module to register the vehicle in the system.
2. **Registration Module -** For all services regarding registration of vehicle including message passing to the detection module to export the number plate from vehicle and send data to the database.
3. **Complain Registration Module -** For all services concerning the registration of complaints from the user including services to send the data to the surveillance module.
4. **Surveillance Module -** For all the services under the module regarding capturing of data from the RSU and CCTV (in case of real time implementation of system) receiving location and respond with alert signals with location.
5. **Security Module -**  For all the services provided to all other modules including the website to provide secure communication channel, saving data to database in encrypted manner, digital signatures and self signed certificates.

All these services provide an end point that with the right information provide a response. It also has built-in integration of error handling for different error codes based on the request given.

* 1. **Communications Interfaces**

A secure communication channel used between the web portal and server will be used at first i.e. exchanging of keys between the web and server using public key encryption for exchanging, to be more specific using the Diffie-Hellman algorithm. Then for authentication between the user and web, self-signed certificates will be generated as for the authenticated web. Furthermore, the data exchange between the web and the server will be in the form of digital signatures therefore no result in the integrity of the data in the middleware i.e. message formatting in the electronic manner using digital signatures. The system will use the HTTP protocol for the data transfer.

1. **Functional Requirements**
   1. **Functional Hierarchy**



As per above, there is a functional hierarchy described by each of the levels of hierarchy. In terms of user and system interaction, these can be written as the following,

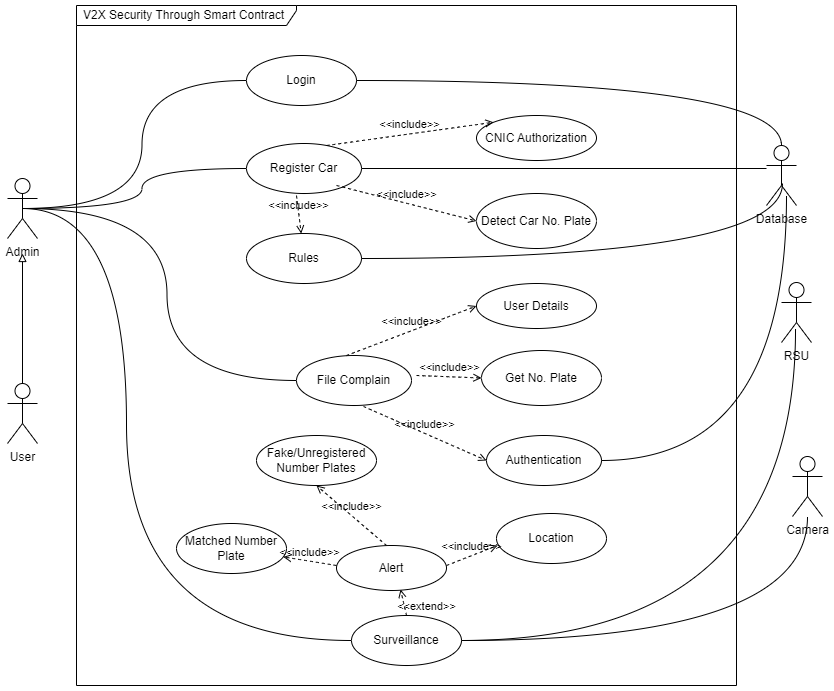
1. Admin should be able to login into the system.
2. Admin should be able to reset their passwords.
3. Admin should be able to register a car.
4. Registration of cars must be only done after following some set of rules.
5. Admin should be able to file a complaint for stolen/missing vehicles.
6. Admin should be able to check the complaint records.
7. Admin should be able to monitor the system that is to see footage (in case of real time implementation)
8. Admin should be able to enter details in complaint and registration of cars.
9. Admin should be able to monitor vehicle movement in particular locations.
10. Admin should be able to see all alerts and total complaints, pending complaints and solved complaints.
11. The Detection module should be able to fetch license plates from vehicles.
12. The Detection module should be able to trigger alerts in case of fake and unregistered license plates.
13. RSUs should be able to fetch and throw signals in terms of velocities to the detection system (in case of real time implementation).
14. Security module should be able to encrypt the license plate and send it to the database.
15. Security module should provide a tag to every car registered in the system.
16. Security module should provide a secure communication channel between services and web-portal..
17. Security module should maintain data integrity between processes.
    1. **Use Cases**

The primary uses of concerns right now falls into three categories:

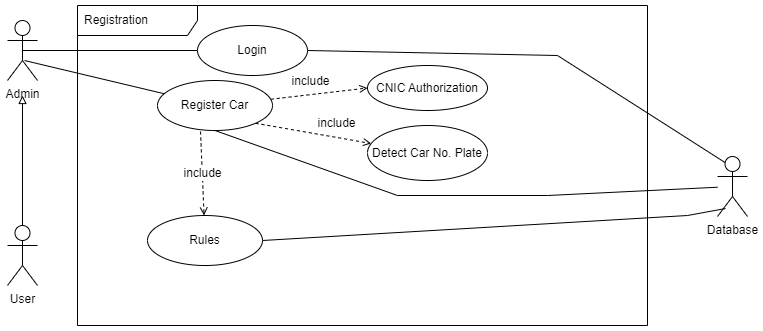
1. Car registration
2. Complaint registration
3. Car recovery

Each of the the following use cases are further categorized into sub-functions as following:

1. Car registration
2. Register
   1. CNIC Authorization
   2. Detect Car No. plate
3. Rules
   1. Attribute based rules
   2. Authentication rules
4. Complaint registration
   1. File complaint
      1. User Details
      2. Get Number plate
      3. Authentication
5. Car recovery
   1. Number plate
      1. Match plate
      2. Fake/Unregistered plate



* + 1. **Car Registration**



Admin, user and database will utilize the car registration system. Admin will login to the system and then the user will be telling the information to the admin when registering the user's car. This provided information will then be authenticated on the basis of attribute based rules.

| **Login: 1.1** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 1.1 | | |
| **Actors:**  User, Admin, Database | | | | |
| **Feature:** To access web portal | | | | |
| **Pre-condition:** | | Admin must have credentials | | |
| **Scenarios:** Admin opens a web browser and enters the web url then enters the credentials and click on login to successfully log in into the system. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Enter email and password | | | Home page is rendered |
| **Alternate Scenarios:** NA | | | | |
| **Post Conditions :** Successful login | | | | |
| **Step#** | **Description** | | | |
| **1.** | Successful login after providing correct credentials | | | |
| **Use Case Cross referenced** | | | NA | |

| **Register Car: 1.2** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 1.2 | | |
| **Actors:**  User, Admin, Database | | | | |
| **Feature:** To register number plate of car | | | | |
| **Pre-condition:** | | Admin must be signed in into the system. | | |
| **Scenarios:** User asked the admin to register the car by providing the details. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Admin enters user basic information | | | Backend validate the information on the basis of prescribed rules and then register number plate after validation |
| **Alternate Scenarios:** NA | | | | |
| **Post Conditions :** Car will register after fulfilling the registering criteria. | | | | |
| **Step#** | **Description** | | | |
| **1.** | Name, CNIC#, phone# , and email will be provided through the web form and image of number plate will be extracted by the camera | | | |
| **Use Case Cross referenced** | | | NA | |

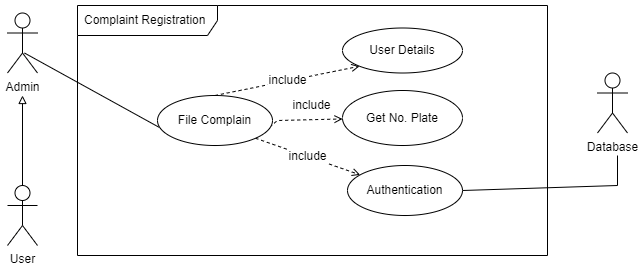
| **Rules: 1.3** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 1.3 | | |
| **Actors:**  User, Admin, Database | | | | |
| **Feature:** Attribute based rules to authenticate information provided through form | | | | |
| **Pre-condition:** | | Information must be provided to trigger the rules which will be checked in this use case. | | |
| **Scenarios:** Admin enters the data in the form and submit the form to the server, server triggers the rule to authenticate the vehicle. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Validates information provided on car registration form | | | If rules are satisfied then, database is updated with record |
| **Alternate Scenarios:** NA | | | | |
| **Step#** | **Description** | | | |
| **1.** | Hash will be placed in an excel file or database with respect to the car number plate. | | | |
| **Use Case Cross referenced** | | | Register car : 1.2 | |

| **CNIC Authorization: 1.4** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 1.4 | | |
| **Actors:**  User, Admin, Database | | | | |
| **Feature:** A unique CNIC# with a particular regex is needed to register a car | | | | |
| **Pre-condition:** | | CNIC must be unique for each user | | |
| **Scenarios:** While registering, CNIC of the car owner must be provided and it will be authorize via rules. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Validates information provided on car registration form | | | If rules are satisfied then, database is updated with record |
| **Alternate Scenarios:** NA | | | | |
| **Step#** | **Description** | | | |
| **1.** | CNIC number will be provided with respect to the car owner to register a car on a CNIC number. | | | |
| **Use Case Cross referenced** | | | Register car : 1.2 | |

| **Detect Car No. Plate : 1.5** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 1.5 | | |
| **Actors:**  User, Admin, Database | | | | |
| **Feature:** A unique car no. plate is needed to register a car | | | | |
| **Pre-condition:** | | Car no. plate must be unique for each user | | |
| **Scenarios:**  While in registration, a camera will detect the car’s number plate and send it to the database via rules. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Validates information provided on car registration form | | | If rules are satisfied then, database is updated with record |
| **Alternate Scenarios:** NA | | | | |
| **Post Conditions :** Camera successfully detected car’s number plate. | | | | |
| **Step#** | **Description** | | | |
| **1.** | Camera detects a number plate via AI model and it will extract string to store in the database. | | | |

| **Use Case Cross referenced** | | | Register car : 1.2 | |
| --- | --- | --- | --- | --- |

* + 1. **Complaint Registration**

****

In the complaint registration system,a complaint of stolen car is launched and before filing the complaint, the system will validate the information from the database and. The complaint will then launch on successful authentication.

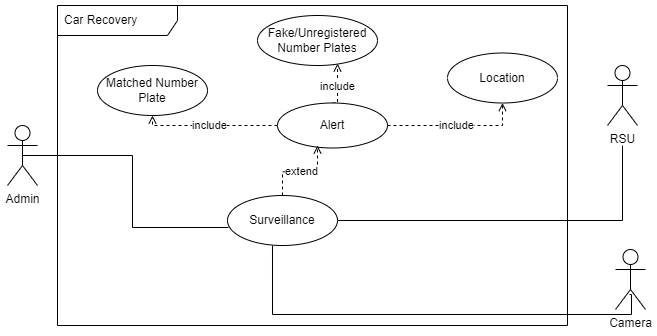
| **File Complain : 2.1** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 2.1 | | |
| **Actors:**  User, Admin, Database | | | | |
| **Feature:** User complain if his car is stolen | | | | |
| **Pre-condition:** | | Car no. plate must exist in the database | | |
| **Scenarios:** User comes to the admin to register a car stolen, theft, lost complaint and admin launches a complaint after filling a web form. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Complain information is extracted through form | | | Authenticates and show results i.e success or fail |
| **Alternate Scenarios:** NA | | | | |
| **Post Conditions :** User filed a complaint after fulfilling the criteria. | | | | |
| **Step#** | **Description** | | | |
| **1.** | Complain form is filled | | | |
| **2.** | Backend authentication | | | |
| **3.** | Updates on database or vice versa | | | |
| **Use Case Cross referenced** | | | NA | |

| **User Details: 2.2** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 2.2 | | |
| **Actors:**  User, Admin, Database | | | | |
| **Feature:** User details is necessary to file a complaint | | | | |
| **Pre-condition:** | | NA | | |
| **Scenarios:** Admin enters user details provided by user while filing a complaint for the car | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Credentials such as email, name, CNIC is provided | | | Validates from a set of rules |
| **Alternate Scenarios:** NA | | | | |
| **Post Conditions :** Admin successfully entered the details for a user to file a complaint. | | | | |
| **Step#** | **Description** | | | |
| **1.** | User details is entered in a form | | | |
| **2.** | Backend validates from database | | | |
| **Use Case Cross referenced** | | | Complaint Register:2.1 |  |

| **Get no. plate: 2.3** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 2.2 | | |
| **Actors:**  User, Admin, Database | | | | |
| **Feature:** Car no. plate is necessary to file a complaint | | | | |
| **Pre-condition:** | | Number plate must be registered in the database. | | |
| **Scenarios:** While launching a complaint for a car, the number late must be provided by user and it will validate via authentication. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Car no. plate is provided | | | Validates from database |
| **Alternate Scenarios:** NA | | | | |
| **Post Conditions :** Car number plate and details provided by the user and already present in the database. | | | | |
| **Step#** | **Description** | | | |
| **1.** | User details is entered in a form | | | |
| **2.** | Backend validates from database | | | |
| **Use Case Cross referenced** | | | Complaint Register:2.1 |  |

| **Authentication: 2.4** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 2.4 | | |
| **Actors:**  User, Admin, Database | | | | |
| **Feature:** All the details provided by admin is verified from a web form | | | | |
| **Pre-condition:** | | User Details must be provided by the admin via web form. | | |
| **Scenarios:** Admin launched the complaint, while successfully registering a complaint, authentication module validate and authenticate the details to complete the process. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Checks each field of information with rules | | | Satisfied(returns true) if all the rules matches and vice versa |
| **Alternate Scenarios:** NA | | | | |
| **Post Conditions :** User filed a complaint after fulfilling the criteria. | | | | |
| **Step#** | **Description** | | | |
| **1.** | Validating information via database. | | | |
| **2.** | Sending data to nodes. | | | |
| **3.** | Receiving response from Database | | | |
| **Use Case Cross referenced** | | | Complaint Register:2.1 | |

* + 1. **Car Recovery**

****

On car recovery, surveillance will be done through camera and RSU and alert will be launched if the number plate matches or the number plate is fake/unregistered. The alert feature will provide location to the admin by returning latitude and longitude of the car.

| **Surveillance : 3.1** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 3.1 | | |
| **Actors:**  Admin, RSU, Camera | | | | |
| **Feature:** Look for the stolen car and monitor vehicles. | | | | |
| **Pre-condition:** | | Cameras must be installed to monitor traffic. | | |
| **Scenarios:** Web Page displays option to monitor the vehicles. Admin clicks on the button, webpage displays traffic visuals. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Traverse through each camera | | | If car found then tells location(longitude, latitude) |
| **Alternate Scenarios:** NA | | | | |
| **Post Conditions :** Admin successfully able to monitor the traffic. | | | | |
| **Step#** | **Description** | | | |
| **1.** | Admin enter the number plate | | | |
| **2.** | a real time table is shown which is being update | | | |
| **Use Case Cross referenced** | | | NA | |

| **Alert : 3.2** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 3.2 | | |
| **Actors:**  Admin, RSU, Camera | | | | |
| **Feature:** System alerts when the location of stolen or fake/unregistered car is found | | | | |
| **Pre-condition:** | | There must be an unregistered vehicle or complaint vehicle for the alert. | | |
| **Scenarios:** System generate alerts for unregistered, stolen vehicles. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | This feature triggers as soon as location of car is found | | | notification keeps on generating until the user does not responds to it |
| **Alternate Scenarios:** NA | | | | |
| **Post Conditions :** System generate alerts successfully. | | | | |
| **Step#** | **Description** | | | |
| **1.** | System keeps on checking the car | | | |
| **2.** | Triggers alert feature once the car is found | | | |
| **Use Case Cross referenced** | | | Surveillance : 3.1 | |

| **Matched No. Plate : 3.3** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 3.3 | | |
| **Actors:**  Admin, RSU, Camera | | | | |
| **Feature:** Alert will trigger if number plate matches | | | | |
| **Pre-condition:** | | There should be a complaint vehicle to get matched from monitored vehicle. | | |
| **Scenarios:** From camera when a complaint vehicle matches, it triggers the alert module. Blockchain system will send and receive the data from nodes. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Keeps on searching the location of number plate | | | Alert will trigger if matched |
| **Alternate Scenarios:** NA | | | | |
| **Post Conditions :** A number plate matches from the monitored or complaint vehicle number plate successfully. | | | | |
| **Step#** | **Description** | | | |
| **1.** | Nodes send and receive number plates to get matched. | | | |
| **Use Case Cross referenced** | | | Alert: 3.2 | |

| **Location : 3.4** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 3.4 | | |
| **Actors:**  Admin, RSU, Camera | | | | |
| **Feature:** returns the location of the stolen car or is fake/unregistered | | | | |
| **Pre-condition:** | | RSU and camera should be functional | | |
| **Scenarios:** When a camera sends a signal via RSU to the system it will send the vehicle location. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Returns location of a car | | | Triggers alert function if stolen car is found or is fake/unregistered |
| **Alternate Scenarios:** NA | | | | |
| **Post Conditions :** Vehicle location send successfully accurate up to 80%. | | | | |
| **Step#** | **Description** | | | |
| **1.** | RSU and Camera work as a unit to send the data to the system for a particular vehicle. | | | |
| **Use Case Cross referenced** | | | Alert: 3.2 | |

| **Fake/Unregistered Number plate : 3.5** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 2.1 | | |
| **Actors:**  Admin, RSU, Camera | | | | |
| **Feature:** Alert will trigger if number plate is not in the systems database | | | | |
| **Pre-condition:** | | A camera should detect fake number plate, unregistered number plate or a car with no number plate. | | |
| **Scenarios:** RSU and Camera work as a unit to detect and send data for unregistered and fake number plates. | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Keeps on validating the number plate | | | Alert will trigger if number plate is not in database |
| **Alternate Scenarios:** NA | | | | |
| **Post Conditions :** Data for unregistered and fake number plates sent to the system successfully to trigger alerts. | | | | |
| **Step#** | **Description** | | | |
| **1.** | RSU and Camera work as a unit to send the data to the system for a particular vehicle. | | | |
| **Use Case Cross referenced** | | | Alert: 3.2 | |

1. **Non-functional Requirements**
   1. **Performance Requirements**

The system should be prepared to handle large amounts of incoming traffic and not falter. It should cater to increasing and decreasing demand, and provide a solution that does not prevent users from being unable to browse and use the features smoothly. The application should be able to perform requests and services as intended, providing a robust and correct set of inputs and outputs, and being able to cater to user mistakes. It should be fast in loading the correct pages and the data in a smooth fashion, and should not overwhelm the user’s machine with its software requirements.

The application should, thus, provide,

1. **Speed** **-** Fast for loading, reloading, content fetching, and not putting limitations on the client end and instead on the user’s end.
2. **Precision -** The software should be able to tolerate mistakes, user problems, and deliver the right results without compromise, and discrepancy in the output provided.
3. **Concurrency -** The software should be able to handle multiple requests and multiple “activated” functionalities concurrently, where needed.
4. **Capacity -** The software should be able to handle and deal with multiple concurrent users requesting for the same resources.
5. **Safety -** The complexity of the software is minimal, for both safety and security purposes, for administrative and normal user procedures. It does not pose harm to the end user. Security modes of software and system are addressed in the design of the software.
6. **Reliability -** The software should exhibit failure free functionality and work with correctness and reliability.
   1. **Safety Requirements**

The software does not include functionalities or operations that would result in possible loss, damage, or harm from the use of our system. It does not pose a physical, financial or otherwise cause of harm to the end user. It does not put interface in such a way that would affect the user negatively, or put any user in any risk or case of discrimination through poor interface design and color sensitivity.

* 1. **Security Requirements**

Obvious issues such as data privacy, integrity, confidentiality, protection of data generated by users through their content, protection of internal private information, and exposure to data to only what is needed when is the primary concern of the software project. The data modeling aspect concerns itself with what is populated by what, and who has access to what kind of information and at what level. We also define what resources are defined under what users and at what capacity to ensure un-authorized users and data are redirected accordingly. Unauthenticated requests for data are prevented from accessing any part of the system through the use of guards, proper digital signatures and self signed certificates assures the confidentiality, integrity of the requests and data is ensured by the proper management of sessions and through password encryption.

The system should focus on setting up its probable infrastructure as securely as possible, and making sure that communication between different services is done securely and data is trusted to be stored as expected. Authentication, authorization, are all important concerns and are implemented via security layered protocol.

* 1. **User Documentation**

Our requirements do not yet specify the development of user documentation to act as helping guides. We assume that the user interfaces will be simple and enough to be used at an MVP level. This is because the scope of an FYP is limited to development as a main priority. If time allows and if needed, user documentation can be developed after the development has been completed and has ceased.

1. **References**

Anshuman Dash, Satyajit Pal, and Chinmay Hegde (2018). Machine Learning-Based Ransomware Auto-Detection in IoT Devices.

R. Kumar, X. Zhang, W. Wang, R. U. Khan, J. Kumar, and A. Sharif, “A Multimodal Malware Detection Technique for Android IoT Devices Using Various Features,” IEEE Access, vol. 7, no. 6, pp. 64411-64430, 2019, Doi: 10.1109/ACCESS.2019.2916886.

1. **Appendices**

Not Applicable