Software Design Specifications Document

Project Title

By

Noor Ul Huda Ajmal 282685

Shaheer Ahmed 321953

Supervisor:

Dr. Hasan Ali Khattak

Co-Supervisor (if any):

Mr. Majid Maqbool

Mr. Bilal

Bachelor of Science in Computer Science (2019-2023)

Department of Computing

School of Electrical Engineering and Computer Science

National University of Sciences & Technology

Table of Contents

Project Title **1**

Table of Contents **2**

List of Tables 3

List of Sequence Diagrams 3

List of Activity Diagrams 3

1. Introduction **1**

2. Design Methodology and Software Process Model **2**

3. System Overview **3**

3.1 Architectural Design 3

4. Design Models **4**

5. Data Design **5**

5.1 Data Dictionary 5

6. User Interface Design **6**

6.1 Screen Images 6

6.2 Screen Objects and Actions 6

Appendix A **7**

Appendix B **9**

Appendix C **21**

List of Architecture Diagrams

Fig. ARD1 High level architecture diagram 3
Fig. ARD2 Website architecture 4

List of Activity Diagrams

Fig. AD1 System Activity Diagram 5

List of Design Diagrams

Fig. DF1 Edge Device Dataflow 6
Fig. DF2 Website Dataflow 7
Fig. BSD1 Drowsy Driver Event 8
Fig. BSD2 Accident Detection Event 8

List of Data Diagrams

Fig. ERD1 Website Schema 9
Fig. ERD2 Model schema 9

List of Tables

Fig. DDT1 Complaints Data Dictionary 10
Fig. DDT2 Models Data Dictionary 10
Fig. DDT3 User Data Dictionary 10

List of Screen Images

Fig.SI1 Authentication page 11
Fig.SI2 Download Models page 11
Fig.SI3 Complaints page 11

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) ***include the ones given at scope time both in doc and presentation**	**Action Taken**

Supervised by:

Signature_____________

1. Introduction

Driver Drowsiness Detection is a system that monitors driver behavior using a camera mounted on the dashboard and makes sure they are attentive to the road by detecting drowsy behavior. It consists of an edge device along with a camera and a neural network model utilizing computer vision that monitors the driver. Upon detection, a buzzer would be used to alert the drowsy driver.

The main modules and their development progress is as follows:

APPLICATION MODULE	PERCENTAGE PROGRESS
ML model	100%
Program logic	70%
Sensor Calibration	50%
User Dashboard	20%
Admin Panel	0%
Database	50%

2. Design Methodology and Software Process Model

The program logic has been developed using Procedural Design Methodology.

Procedural Design Methodology is a systematic and structured approach to designing and developing computer programs or systems. It involves defining a series of procedures to achieve a specific outcome, such as solving a problem or completing a task.

There is no specific "procedural design methodology" that is commonly used in the development of Machine Learning (ML) based real-time applications. However, as Neural Network models can exclusively use functions it can be said to be following the procedural programming paradigm.

The use of procedural design methodology offers several benefits such as:

1. Clarity and Consistency: Procedural design methodology provides clear and well-defined steps for developing the ML model, making the development process more organized and consistent.
2. Reusability: The modular nature of procedural design methodology allows the reuse of code, reducing development time and increasing efficiency.
3. Debugging: The well-defined steps and clear code structure provided by procedural design methodology makes it easier to debug the ML model in case of errors or unexpected results.
4. Flexibility: The procedural design methodology allows for modifications to be made easily, as the development process is divided into smaller, more manageable steps.
5. Improved documentation: The clear and organized development process provided by procedural design methodology makes it easier to document the design, implementation, and testing of the ML model, improving overall transparency and accountability.

3. System Overview

The system consists of an edge device along with a camera. A machine learning model would be deployed on the edge device that would keep check of the driver's behavior, making sure that the driver is attentive to the road. It will use a buzzer to either the vehicle's stereo system to alert the driver.

3.1 Architectural Design

The higher-level architecture diagram of the system is as follows:

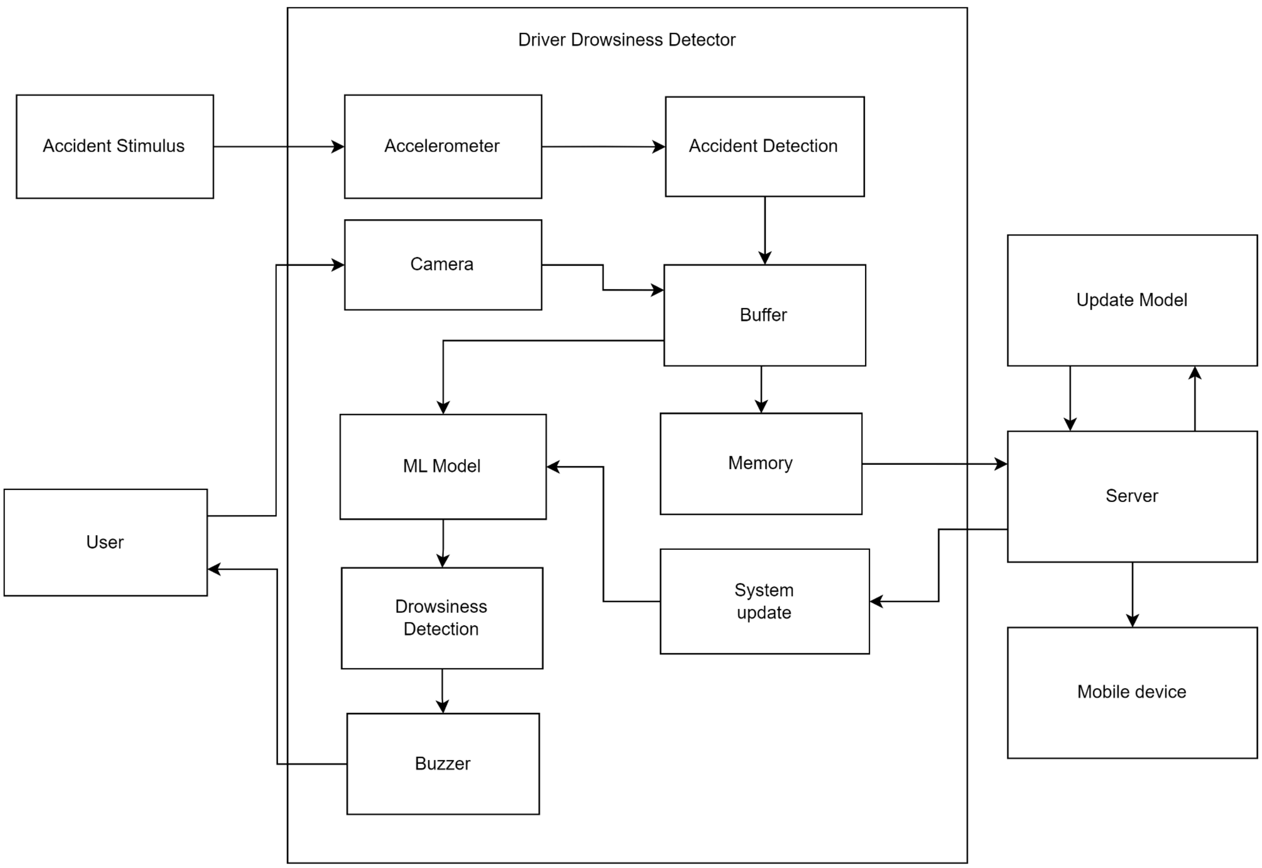

Fig. ARD1 High level architecture diagram

The camera records the driver's footage and sends it to the buffer after preprocessing. The model then classifies the image and the drowsiness detection module then determines whether the driver is drowsy or not. The user is alerted to wake up via a buzzer. The accelerometer can detect an accident via deceleration, in which case the system can store the last few seconds of the footage in the memory. The user can then upload that footage on the server. The developers can retrain the model and upload it to the server so the users can update their systems.

The program logic on the edge device follows procedural programming, hence it does not follow a specific architecture. The website on the other hand will use the model view controller architecture.

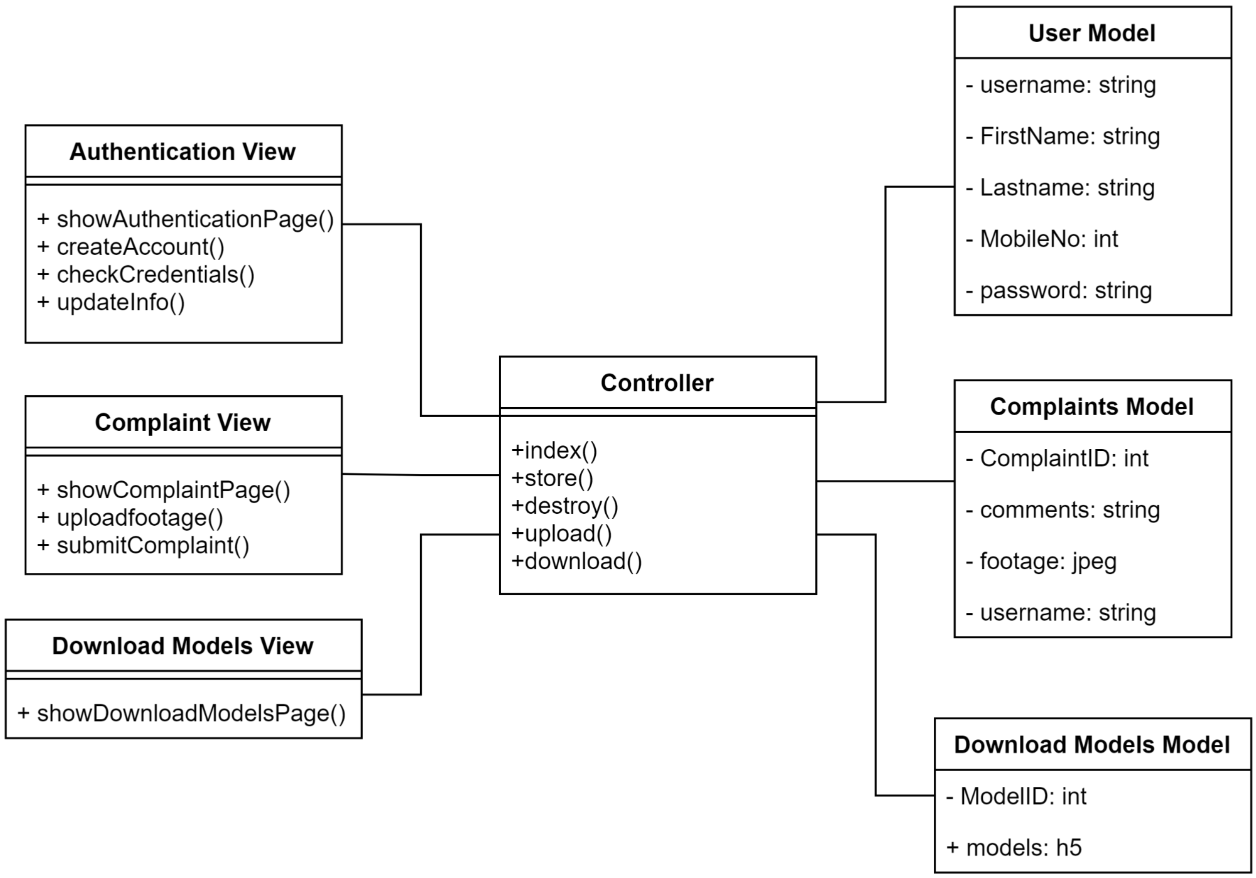

Fig. ARD2 Website architecture

The website will have 3 webpages, hence 3 views as well as 3 models corresponding to those views. The views are what the user sees and interacts with while the controller manages how the view affects the models. It then updates the views to reflect those changes.
The Authentication view corresponds with the User model, the Complaint view corresponds with the Complaints model while the Download Models View corresponds to the Download Models model.

4. Design Models

As the system uses the Procedural Approach, the design will be illustrated by the following diagrams.

Activity Diagram:

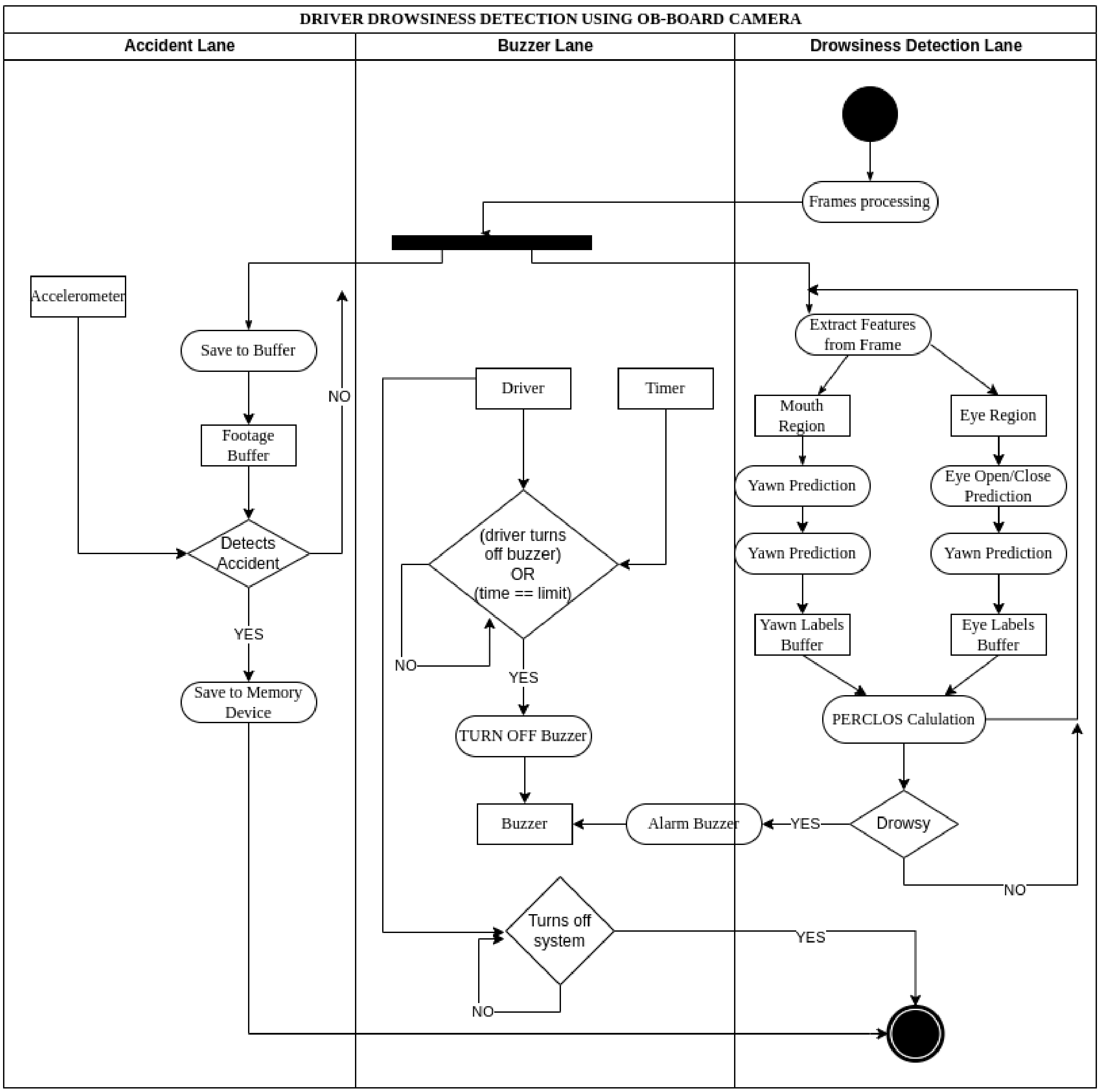

Fig. AD1 System Activity Diagram

The activity diagram shows the overall program logic. When the program begins, it starts capturing frames from the video using an IR camera. The frames are then further pre-processed to get three separate regions from a person's face; left eye, right eye and mouth. The extracted regions are cropped and then fed to corresponding models to get inference about the eyes being open or closed and whether there is yawning or not.
The frequency of person yawning and closed eyes is continuously calculated inside the buffer to get perclos measurements for the last 5 frames, if perclos value is exceeding the defined limit, then the alarm through buzzer will be generated to alert the drowsy driver till the buzzer is manually turned off by the driver.
Alongside, the frames are being saved to memory buffer and upon detecting the accident through accelerometer, the last 30 seconds footage will be saved to memory device.

Data Flow Diagrams:

The system consists of two modules, the edge device and the website.

Edge Device:

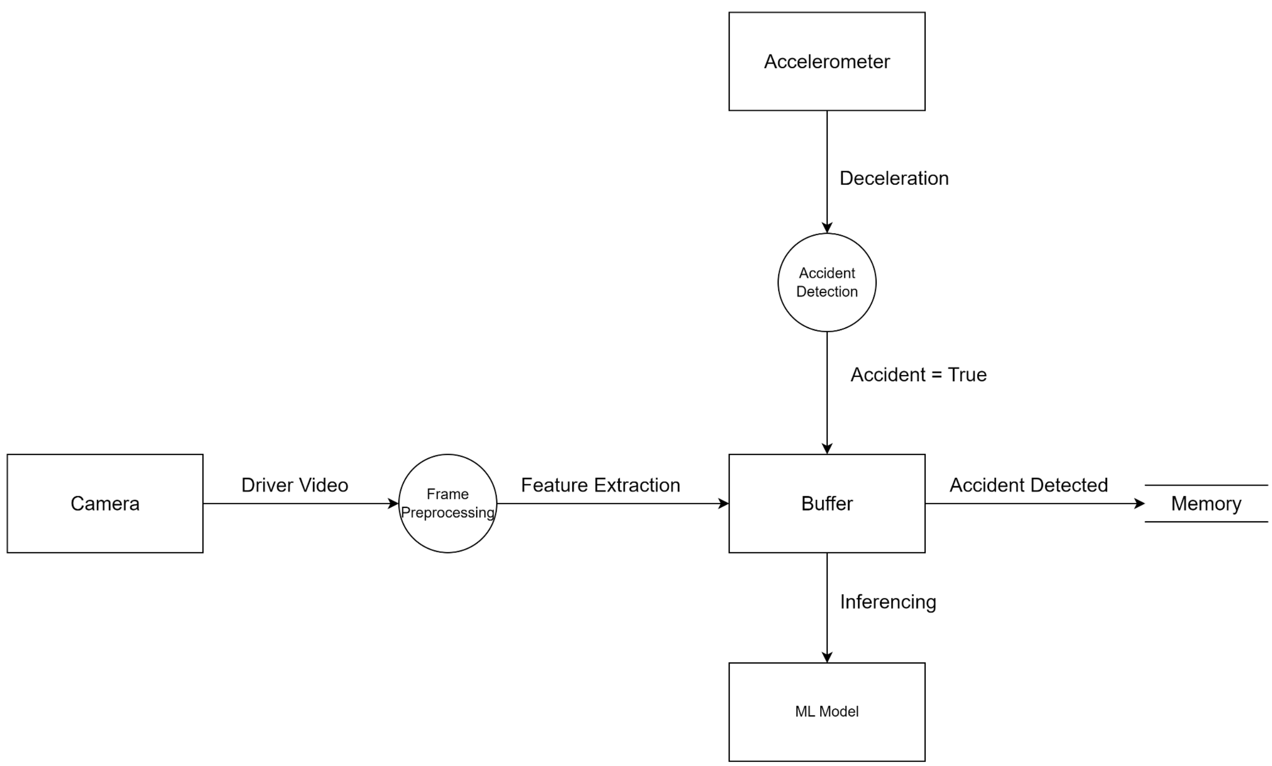

Fig. DF1 Edge Device Dataflow

The system takes video from the camera, preprocesses the frames into a buffer which is then fed into the CNN model for driver drowsiness detection. Additionally, in the event of an accident, the accelerometer can detect deceleration and hence instruct the system to write the buffer to the memory which can then be reviewed for inspecting system defects and ensuring quality control.

Website:

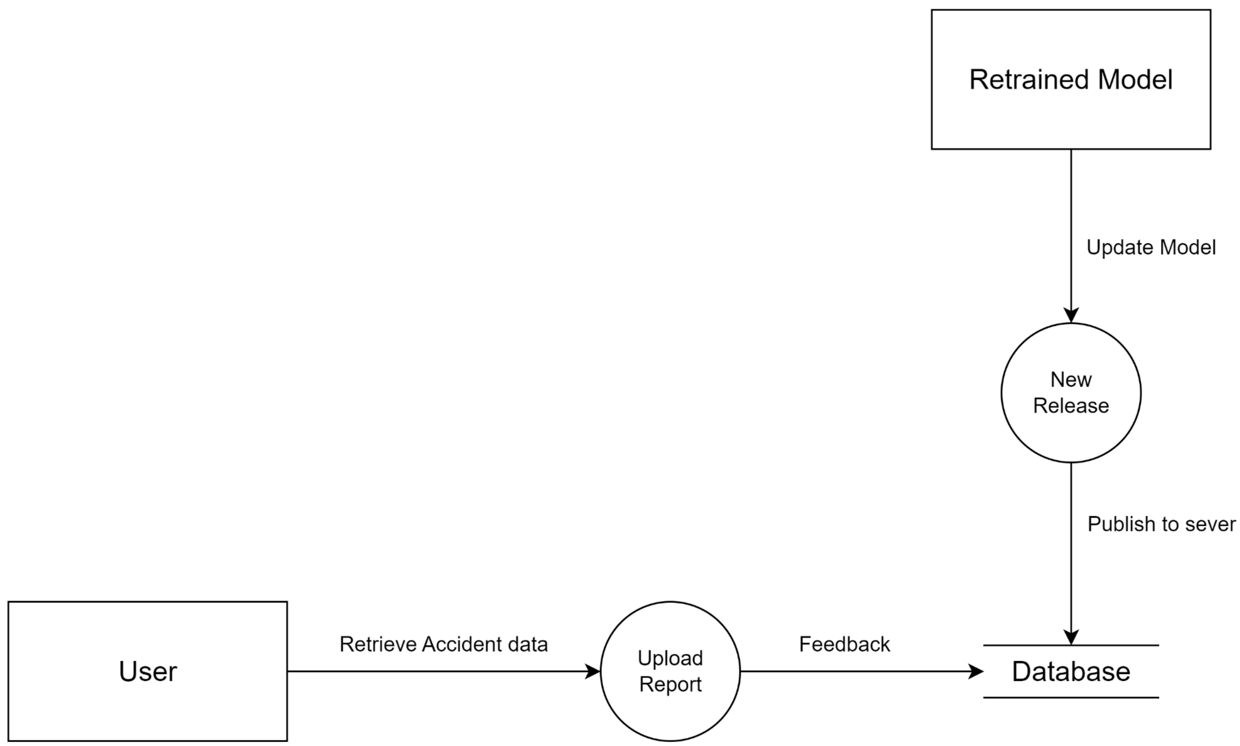

Fig. DF2 Website Dataflow

As the web ecosystem is based on a single page application, the data flow diagram for it is quite simple. The user can retrieve the accident data from the system and upload the complaint report for product review. If the complaint is appropriate, the developers can then retrain the model and release it on the server for the users to download and update their system.

Behavioural State Diagrams:

The system mainly handles two events, detecting a drowsy driver and detecting an accident.

Drowsy Driver

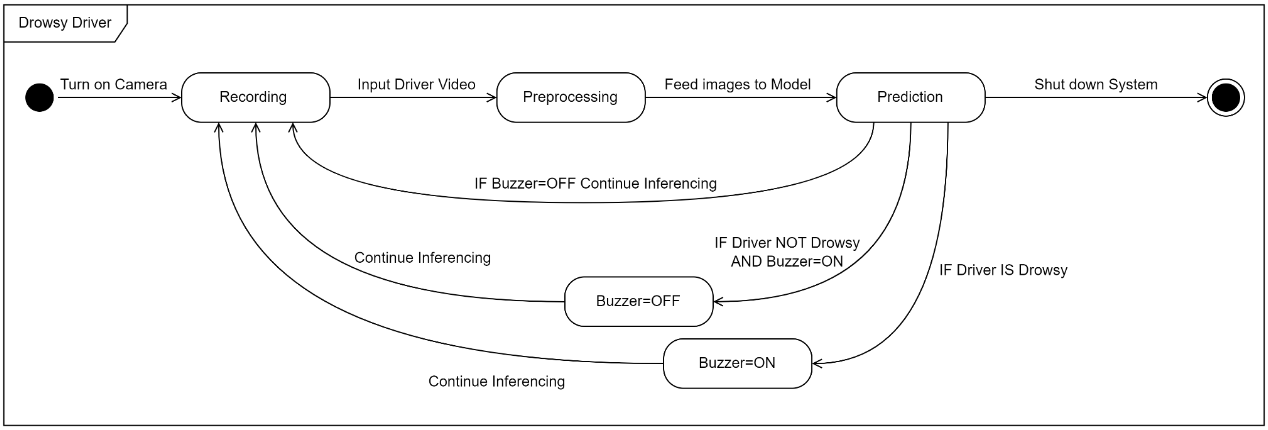

Fig. BSD1 Drowsy Driver Event

When the system is booted, the system is in the Recording state. The video is then preprocessed in the Preprocessing state. These frames are then fed to the ML model in the Prediction state. Now if the Driver is found to be drowsy, the system turns ON the buzzer, then continues inferencing from the Recording State. If the Driver is not drowsy and the buzzer is ON, then the system turns off the Buzzer and continues inferencing from the Recording state. Furthermore, if the buzzer is off and driver is not drowsy, the system will continue inferencing from the Recording State. Lastly, the system can be shut down to enter the terminal state.

Accident Detection

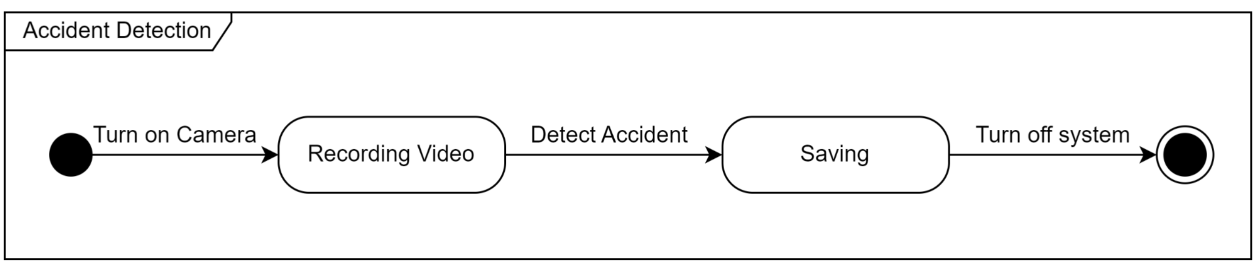

Fig. BSD2 Accident Detection Event

For accident detection, the system can detect an accident by using the accelerometer and enter a saving state where it can save the footage from the buffer into the memory. The system can then be turned off to enter the terminal state.

To view examples of the above mentioned models, see Appendix B

5. Data Design

The edge device does not store any images in normal operation. In an event of an accident, only the images in the buffer would be saved to the memory.

The single page website has a simple Database schema. It consists of two tables, the User table and the complaints table. The user table consists of Username (Primary Key), Password, First Name, Last Name and Mobile number fields. It is then associated with the complaints table which contains the Complaint ID (Primary Key), Footage, Comments and username (Foreign Key)

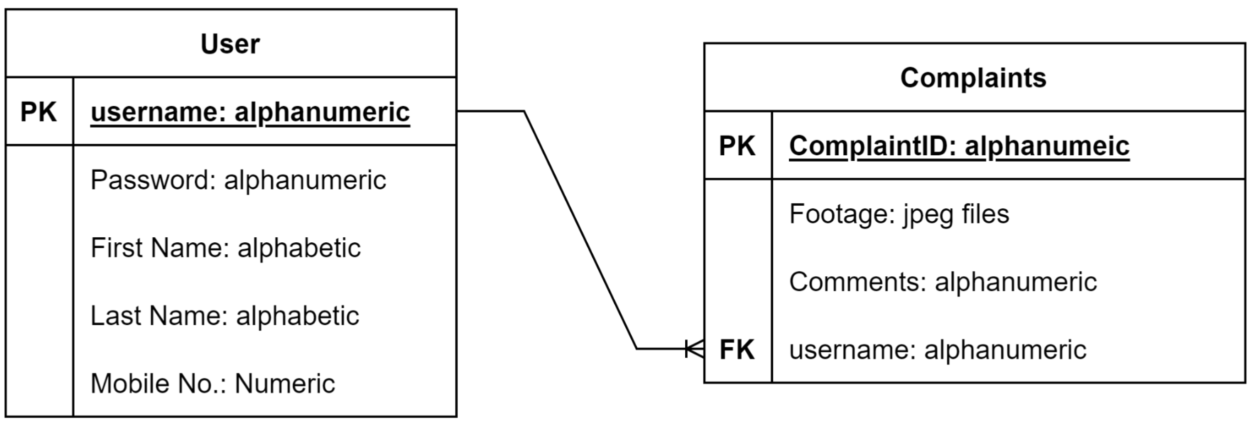

Fig. ERD1 Website Schema

The downloadable models are implemented in a separate database, as the users are downloading the models from another source.

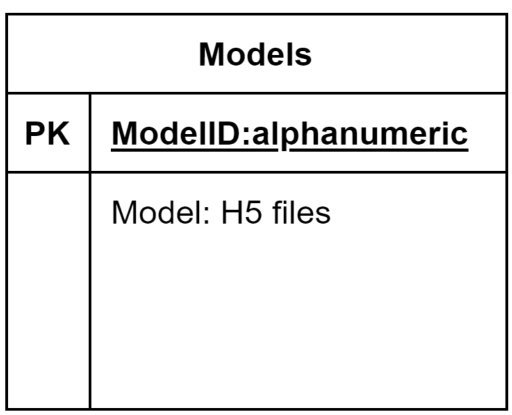

Fig. ERD2 Model schema

5.1 Data Dictionary

The data dictionary for the website database is shown below, arranged alphabetically:

For Complaints Table:

System Entity	Data Type	Data description
Comments	text	User comments for footage
ComplaintID	Autonumber	Primary key of table
Footage	image	image pertaining to accident
Username	text	Foreign key of table

Fig. DDT1 Complaints Data Dictionary

For Models Table:

System Entity	Data Type	Data description
ModelID	Autonumber	Primary key of table
Model	.h5 file	the weights and structure of the model

Fig. DDT2 Models Data Dictionary

For User Table:

System Entity	Data Type	Data description
First Name	text	First name of User
Last Name	text	Last name of User
Mobile Number	int	User mobile number
Password	text	User password
Username	text	Primary key of table

Fig. DDT3 User Data Dictionary

6. User Interface Design

As the edge device is an embedded system, it does not have a UI for the user to interact with. The system is going to enact a plug and play approach.

There are 3 main webpages. First is the authentication page, then the download models page and lastly the complaints page.

6.1 Screen Images

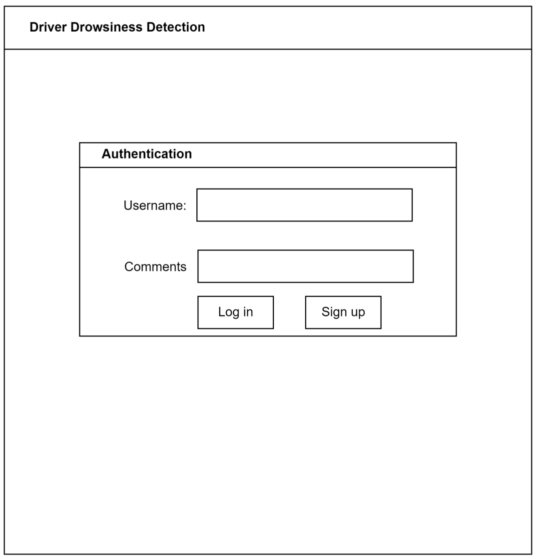

Fig.SI1 Authentication page

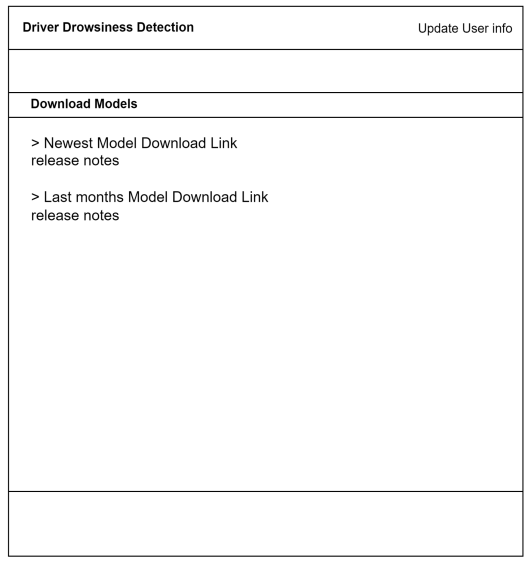

Fig.SI2 Download Models page

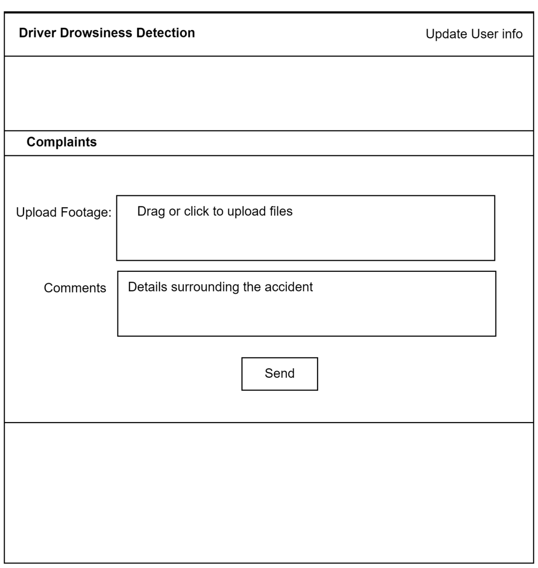

Fig.SI3 Complaints page

6.2 Screen Objects and Actions

The website consists of 3 webpages, they are listed below:

1. The authentication page:
   This page allows the user to log in or sign up for the service.
2. The Download Models page:
   This page contains the download links for the previous models as well as the release notes that denote the updates featured. The user can click on the link to download the new models.
3. The Complaints page:
   This page allows the user to upload the footage of the accident and provide details about the accident using the comments text box.

Appendix A

Appendix B

Appendix C