NUMBER PLATE RECOGNITION SYSTEM

SOFTWARE REQUIREMENTS AND SPECIFICATIONS

DARSHIT PANDYA [IU1641050034]
NISARG PETHANI [IU1641050045]
KRUNAL MEHTA [IU1641050027]

INDEX

Introduction:

With increasing number of vehicles on roads, it is getting difficult to manually enforce laws and traffic rules for smooth traffic flow. Tollbooths are constructed on freeways and parking structures, where the car has to stop to pay the toll or parking fees. Also, traffic Management systems are installed on freeways to check for vehicles moving at speeds not permitted by law. All these processes have a scope of improvement. In the center of all these systems lies a vehicle. In order to automate these processes and make them more effective, a system is required to easily identify a vehicle. The important question here is how to identify a particular vehicle? The obvious answer to this question is by using the vehicle's number plate.

Vehicles in each country have a unique license number, which is written on its license plate. This number distinguishes one vehicle from the other, which is useful especially when both are of same make and model. An automated system can be implemented to identify the license plate of a vehicle and extract the characters from the region containing a license plate. The license plate Number can be used to retrieve more information about the vehicle and its owner, which can be used for further processing. Such an automated system should be small, and portable.

Various license plate detection algorithms have been developed in past few years. Each of these Algorithms has their own advantages and disadvantages. This project described the method in which license plate is detected using confidence related predictions. As multiple detections are available
For single license plate, post-processing methods are applied to merge all detected regions. In addition, trackers are used to limit the search region to certain areas inane image. It suggests a different approach of detection using banalization and elimination of unnecessary regions from an
Image. In this approach, initial image processing and banalization of an image is carried out based on the contrast between characters and background in license plate. Afterbinarizing the image, it is divided into different black and white regions. These regions are passed through elimination stage to get the final region having most probability of containing a number plate.

Purpose of this Project:

The main purpose of this project is to detect a license plate from an image provided by a camera. An efficient algorithm is developed to detect a license

plate in various luminance conditions. This algorithm extracts the license plate data from an image and provides it as an input to the stage of Number-Plate Recognition System. The image of a vehicle is given as an Input from the camera. Extracted image can be seen on screen for verification.

Tools used:

> Windows 8.1 (operating system)
> OpenCV
> Python

Requirements:

- Functional Requirements:

1. Software should provide a way to load scanned image from the traffic camera for recognition purpose.
2. Software should provide the processing tools for cropping the number plate image from basic captured image.
3. Captured basic number plate will contain any property from below.
    - Blur
    - Night light
    - Low light
    - High/low brightness
    - High/Low contrast
    - Different background (White-personal / Yellow-commercial)
4. Traffic camera :
    - Captured image quality: HD min/FHD/EDHD
5. Software should process extraction and recognize characters of number plate with processing & recognition tools of computer via image processing.
6. There should be a database for getting information of owner by the recognized characters
7. Database must contain,
    - Number plate characters(Vehicle no)
    - Name of owner
    - Model no
    - Address
    - License no
    - Year of passing
    - Contact details of owner
    - No of foul committed
    - Type of foul
8. Considerable amount of training data set
9. The input image in computer must in RGB format

10. In case of image captured by traffic camera must have high resolution
11. Software must support all type of image like jpeg , png , bmp etc.
12. Software building time requirement:6 months
13. Programming language to be used: Open CV , Visual C++

- Non-Functional Requirements:

A non-functional requirement is a system must behave or how is the system's behaviour . This also specifies how the system's quality characteristics or quality attributes . In order to put this constraint upon the specific system behaviour , the qualities goals of the designed system should goes in these:

Execution qualities:

- Functionality
- Security
- Usability
- Effectiveness & Efficiency

Evolution qualities:

- Availability
- Reliability
- Manageability

- Reliability: The system is high reliable.
- Accessibility: It can be easily accessible i.e. click & run.
- Efficiency: Resource consumption for given load is quite low.
- Fault tolerance: Our system is not fault tolerant due to insufficient hardware.
- Robustness: Our system is not capable to cope with errors during execution.
- Scalability: Our project is scalable i.e. we can add more resources to our project without disturbing the current scenario.

- Software requirements:

Main Programming Language:

- Open CV

- Python
- Visual C++

Main Database Developer:

- SQL Server 2005

- Hardware requirements:

  - 1.6 GHz or faster processor.
  - 1024 MB RAM (1.5 GB if running on a virtual machine).
  - 3 GB of available hard-disk space.
  - 5400-RPM hard-disk drive.
  - DirectX 9-capable video card running at 1024 x 768 or higher display resolution.
  - DVD-ROM drive.

Data-dictionary:

Vehicle No.	Owner Name	Model Name	Model No.	Passing Year
GJ01AB1234	Raj	Activa-3G	3G	2008
GJ01AC4567	Mahesh	Access-250	250	2011
GJ01YA2553	Ankit	Jeep-Compass	Limited	2018
GJ18BH621	Ramesh	Activa-5G	5G	2017
GJ01JL9999	Karan	Swift	Vxi	2014
GJ18GH2534	Mosam	i-10	Grand	2016
GJ18CJ6898	Varun	Activa-3G	3G	2010
GJ01GH9090	Harsh	i-20	Vdi	2014
GJ01JH6219	Jayesh	Access-250	250	2007
GJ01OP6743	Bharat	Jupiter	300	2017

Vehicle No.	Owner Name	Licence No.
GJ01AB1234	Raj	GJ123456
GJ01AC4567	Mahesh	GJ145678
GJ01YA2553	Ankit	GJ198765
GJ18BH621	Ramesh	GJ786954
GJ01JL9999	Karan	GJ879656
GJ18GH2534	Mosam	GJ789652
GJ18CJ6898	Varun	GJ326542
GJ01GH9090	Harsh	GJ546214
GJ01JH6219	Jayesh	GJ124565
GJ01OP6743	Bharat	GJ453621

Vehicle No.	Owner Name	Licence No.	Fine
GJ01AB1234	Raj	GJ123456	1000
GJ01AC4567	Mahesh	GJ145678	500
GJ01YA2553	Ankit	GJ198765	1500
GJ18BH621	Ramesh	GJ786954	700
GJ01JL9999	Karan	GJ879656	200
GJ18GH2534	Mosam	GJ789652	100
GJ18CJ6898	Varun	GJ326542	1000
GJ01GH9090	Harsh	GJ546214	500
GJ01JH6219	Jayesh	GJ124565	1500
GJ01OP6743	Bharat	GJ453621	100

Licence No.	Owner Name	Owner Address	Contact Details	Vehicle No.
GJ123456	Raj	Gurukul,Ahmedabad	9898989898	GJ01AB1234
GJ145678	Mahesh	Thaltej,Ahmedabad	8989898989	GJ01AC4567
GJ198765	Ankit	Naranpura,Ahmedabad	7070707070	GJ01YA2553
GJ786954	Ramesh	Sector18,Gandhinagar	9845632145	GJ18BH621
GJ879656	Karan	Nikol,Ahmedabad	9632587412	GJ01JL9999
GJ789652	Mosam	Sector16,Gandhinagar	9786543216	GJ18GH2534
GJ326542	Varun	Sector 21,Gandhinagar	8965473674	GJ18CJ6898
GJ546214	Harsh	Vastral,Ahmedabad	8754693214	GJ01GH9090
GJ124565	Jayesh	Paldi,Ahmedabad	7898654732	GJ01JH6219
GJ453621	Bharat	Gatlodia,Ahmedabad	7598698412	GJ01OP6743

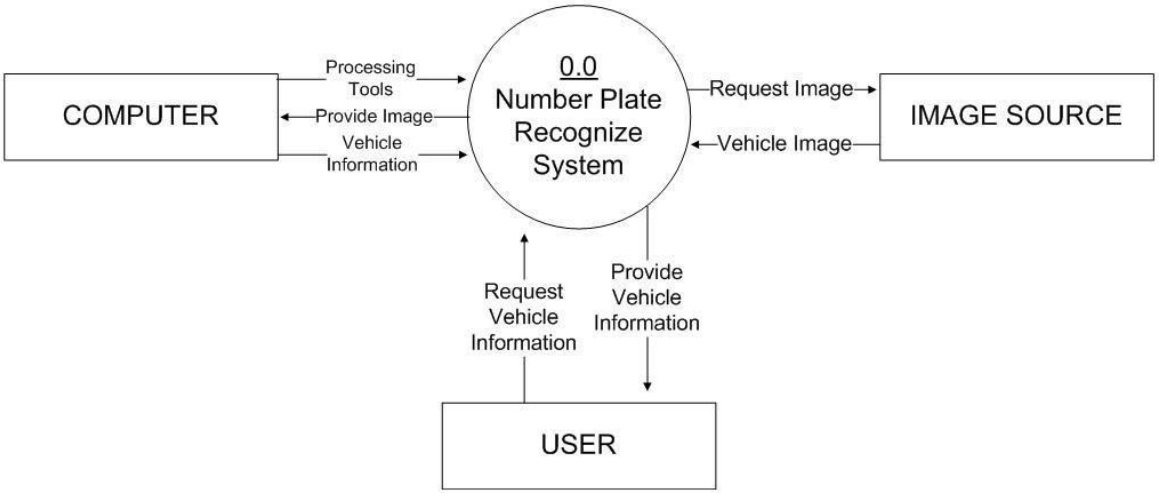
Licence No.	Issue Date	Expiry Date
GJ123456	01-02-2001	01-01-2036
GJ145678	03-04-1990	03-03-2025
GJ198765	04-05-1998	04-04-2034
GJ786954	05-06-2000	05-06-2036
GJ879656	28-04-2001	28-04-2037
GJ789652	04-09-1996	04-09-2032
GJ326542	02-04-1990	02-04-2025
GJ546214	03-05-1997	03-05-2033
GJ124565	15-06-1991	15-06-2026
GJ453621	06-12-1995	06-12-2031

Licence No.	Fouls Committed
GJ123456	2
GJ145678	3
GJ198765	3
GJ786954	2
GJ879656	1
GJ789652	2
GJ326542	1
GJ546214	1
GJ124565	2
GJ453621	3

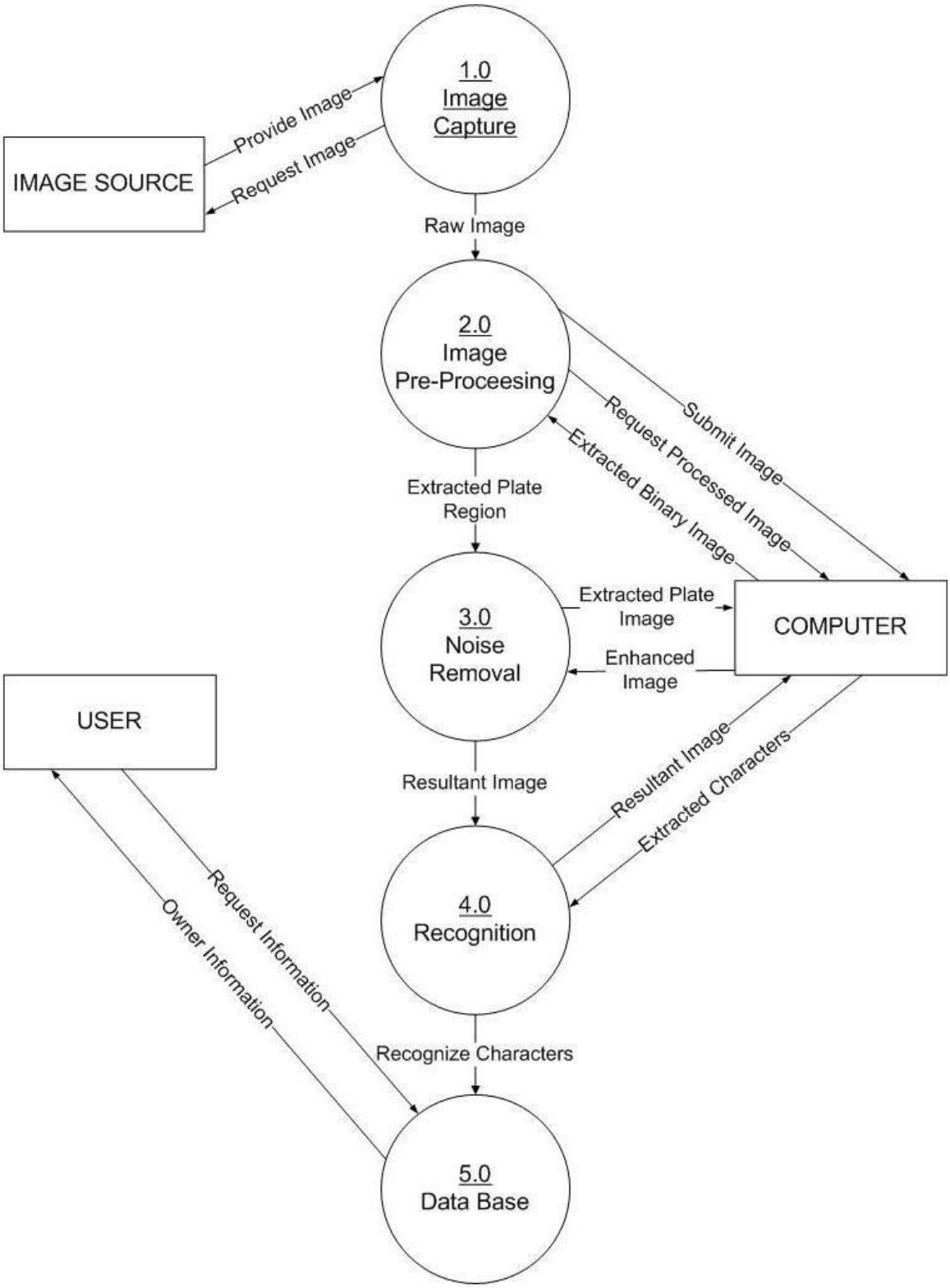
Licence No.	Type Of Foul	No. of Foul
GJ123456	Helmet	2
GJ145678	Signal	3
GJ198765	Seat-Belt	3
GJ786954	Signal	2
GJ879656	Seat-Belt	1
GJ789652	Signal	2
GJ326542	Helmet	1
GJ546214	Signal	1
GJ124565	Helmet	2
GJ453621	Helmet	3

Data FlowDiagram:

DFD Level0:

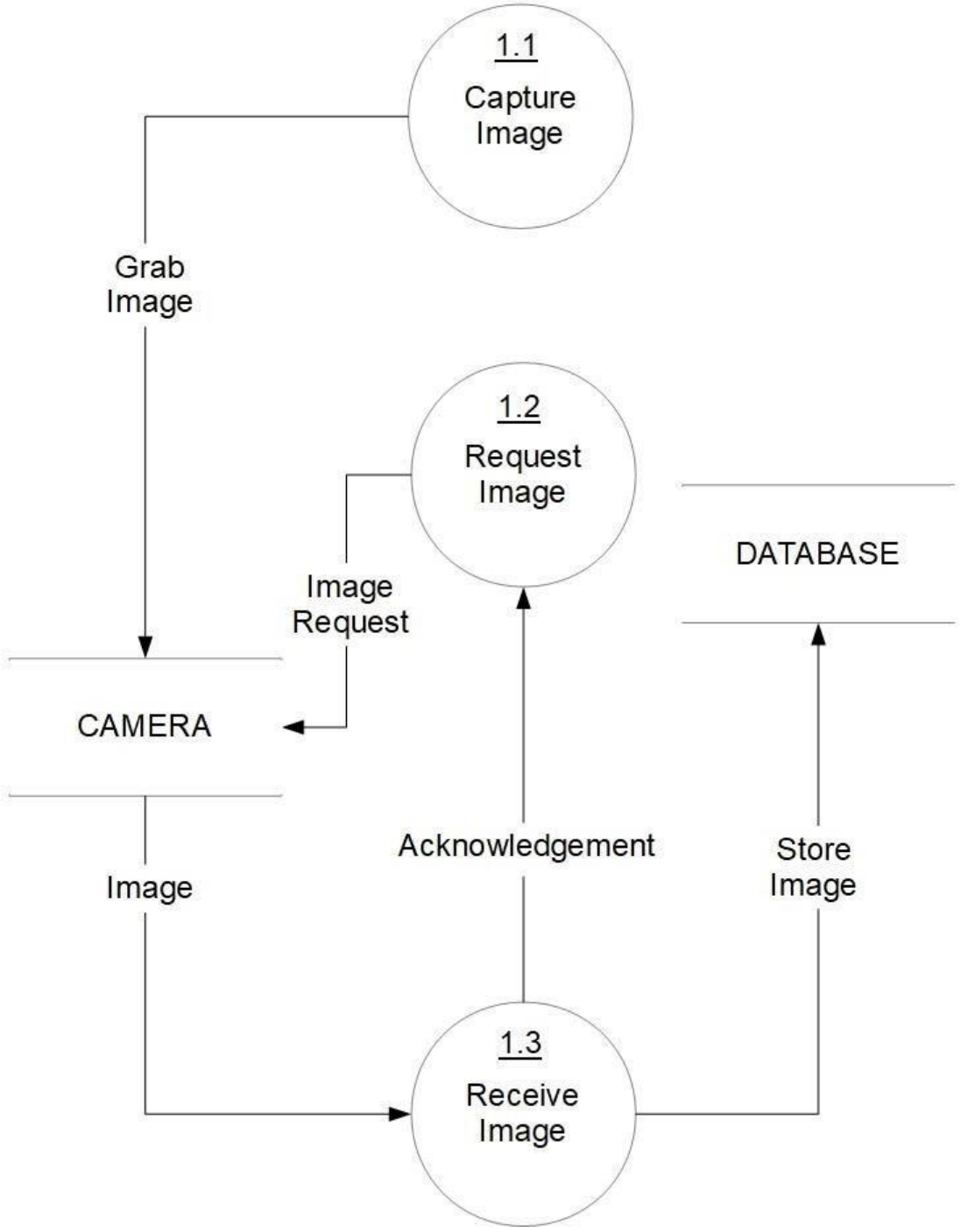


DFD Level1:

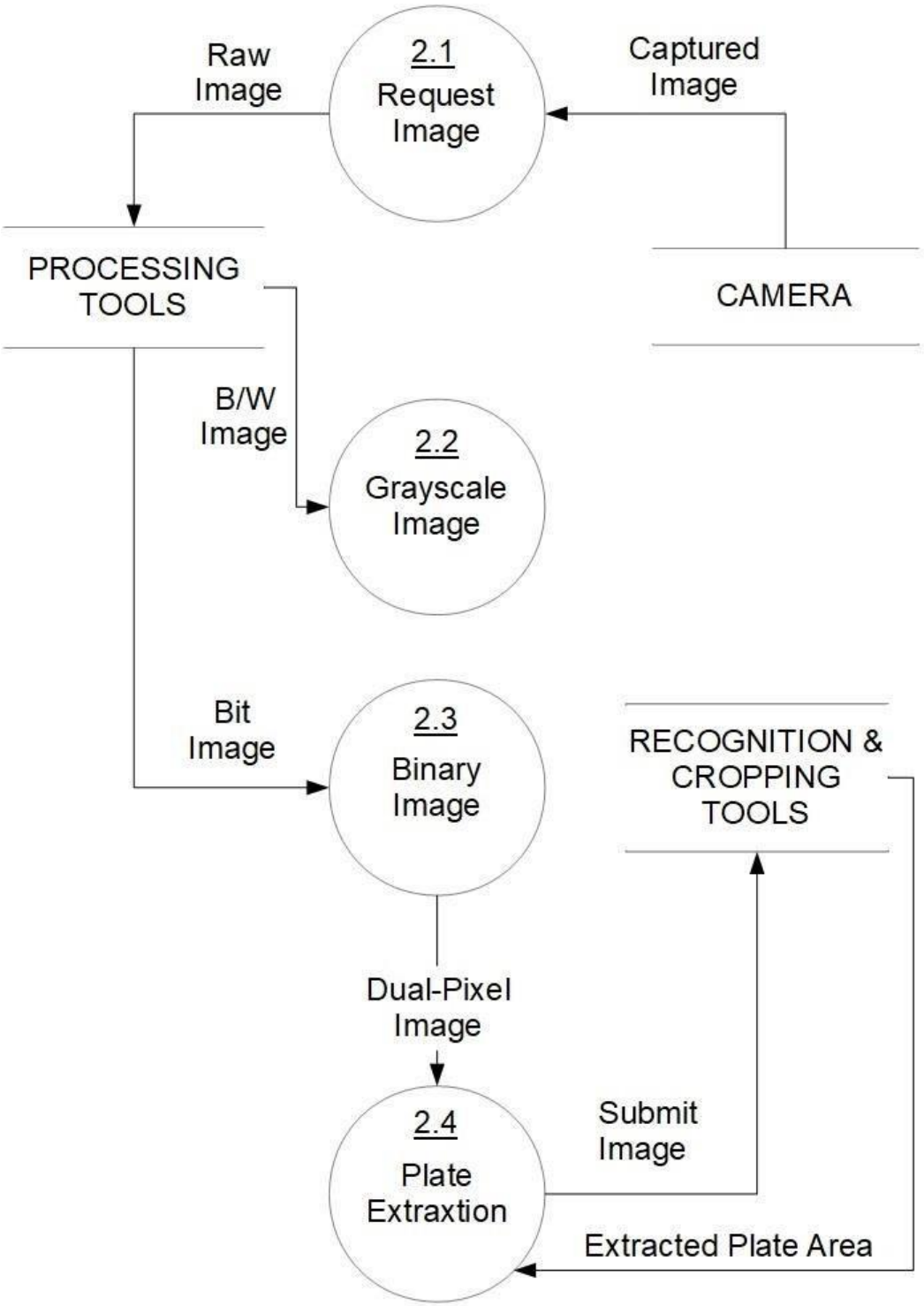

1.0
Image
Capture
IMAGE SOURCE
Provide Image
Request Image
Raw Image
2.0
Image
Pre-Proceesing
Submit Image
Request Processed Image
Extracted Binary Image
Extracted Plate
Region
3.0
Noise
Removal
Extracted Plate
Image
Enhanced
Image
COMPUTER
USER
Resultant Image
Resultant Image
Extracted Characters
4.0
Recognition
Request Information
Owner Information
Recognize Characters
5.0
Data Base

DFD Level2:

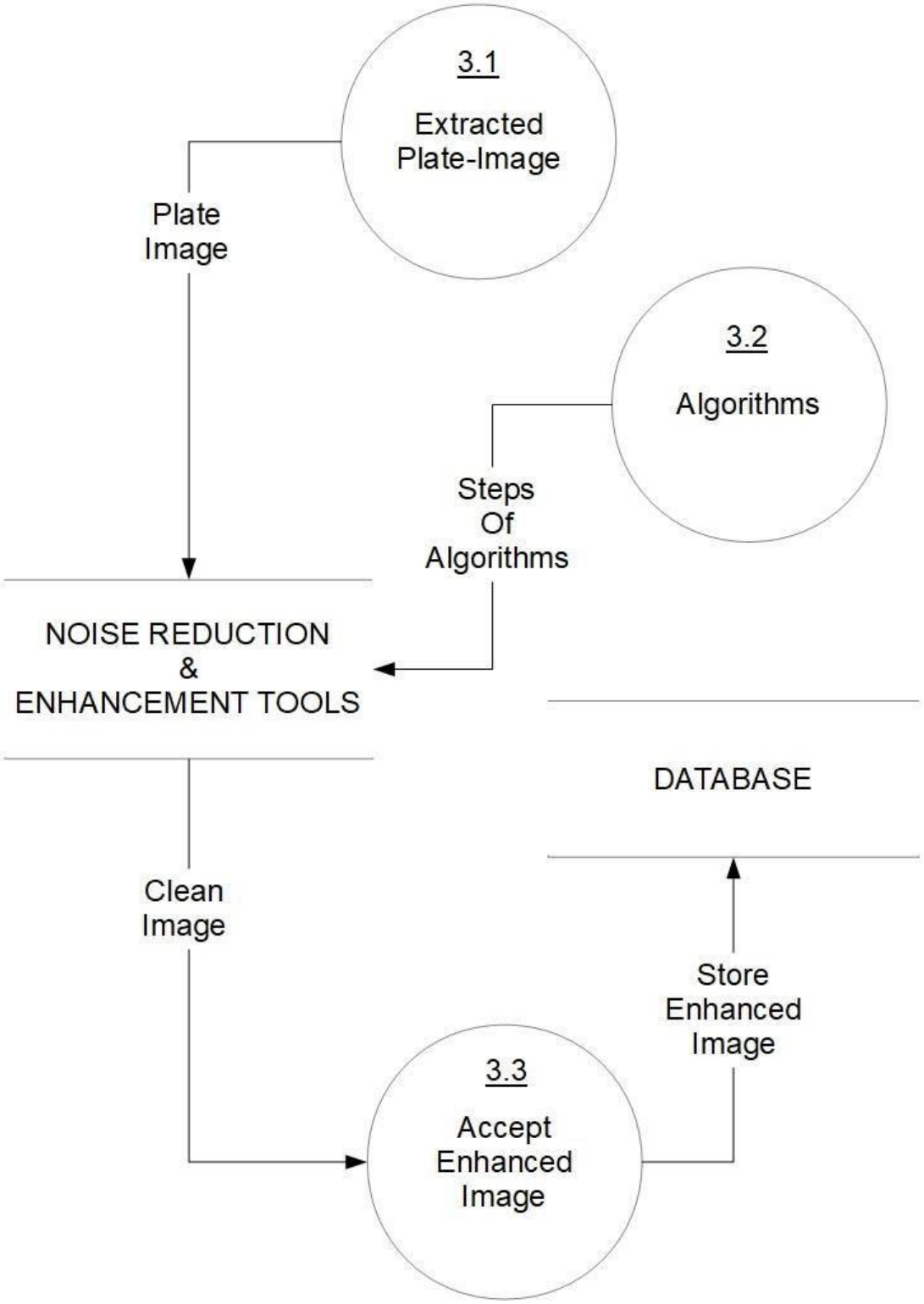
DFD Level2.1:



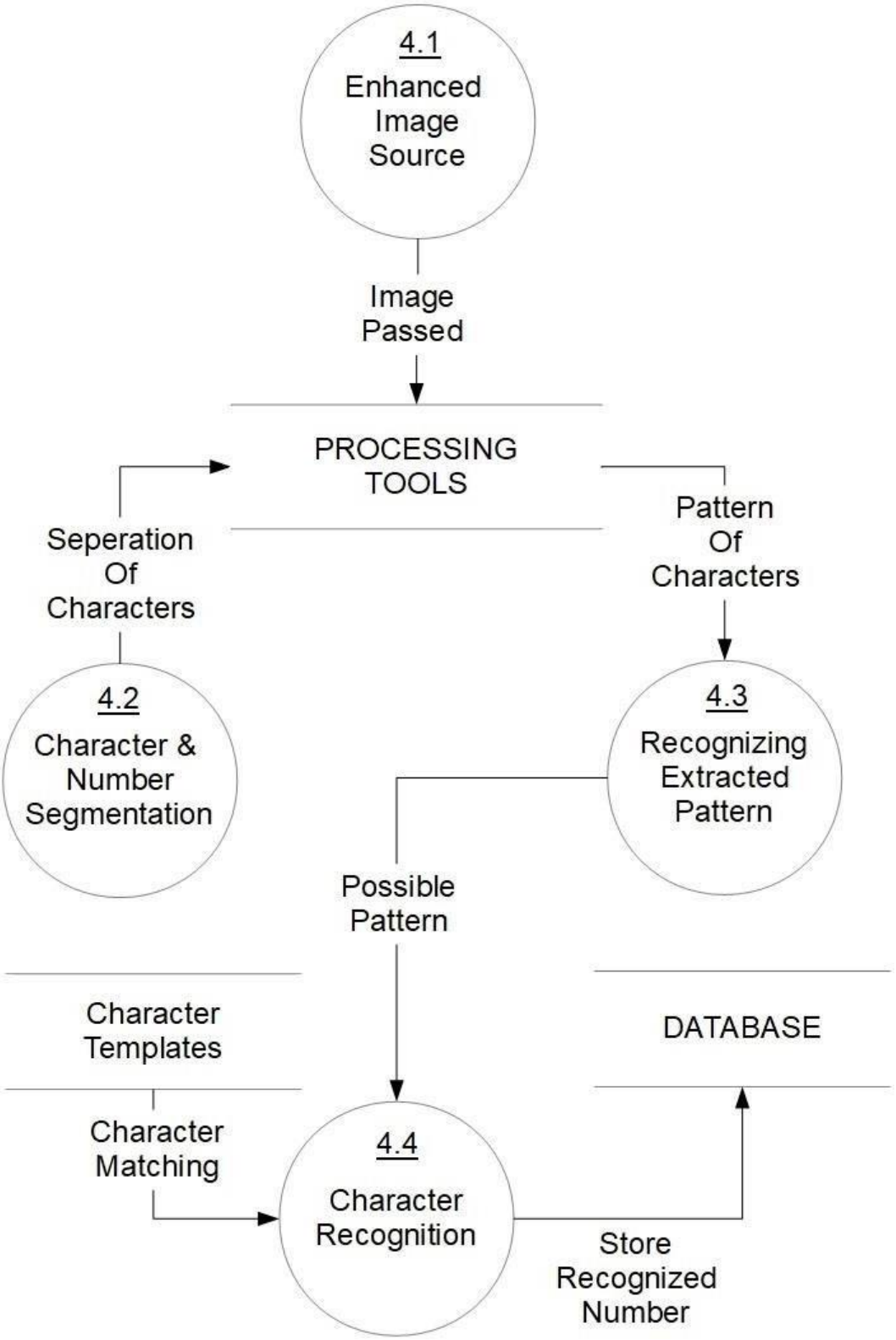
DFD Level2.2:


2.1
Request
Image
Raw
Image
Captured
Image
PROCESSING
TOOLS
CAMERA
B/W
Image
2.2
Grayscale
Image
Bit
Image
2.3
Binary
Image
RECOGNITION &
CROPPING
TOOLS
Dual-Pixel
Image
2.4
Plate
Extraxtion
Submit
Image
Extracted Plate Area

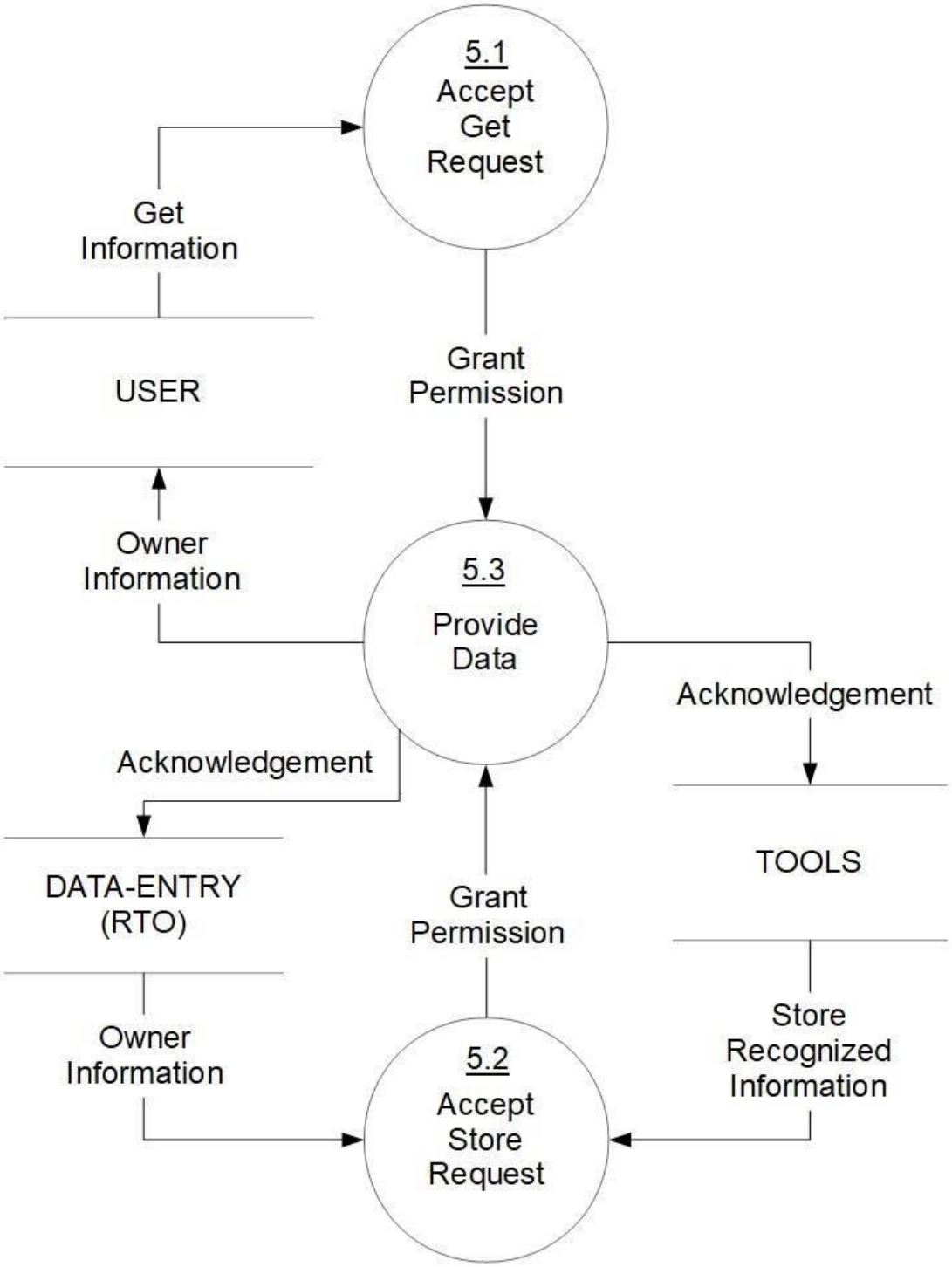
DFD Level2.3:


3.1
Extracted Plate-Image
Plate Image
3.2
Algorithms
Steps Of Algorithms
NOISE REDUCTION & ENHANCEMENT TOOLS
DATABASE
Clean Image
Store Enhanced Image
3.3
Accept Enhanced Image

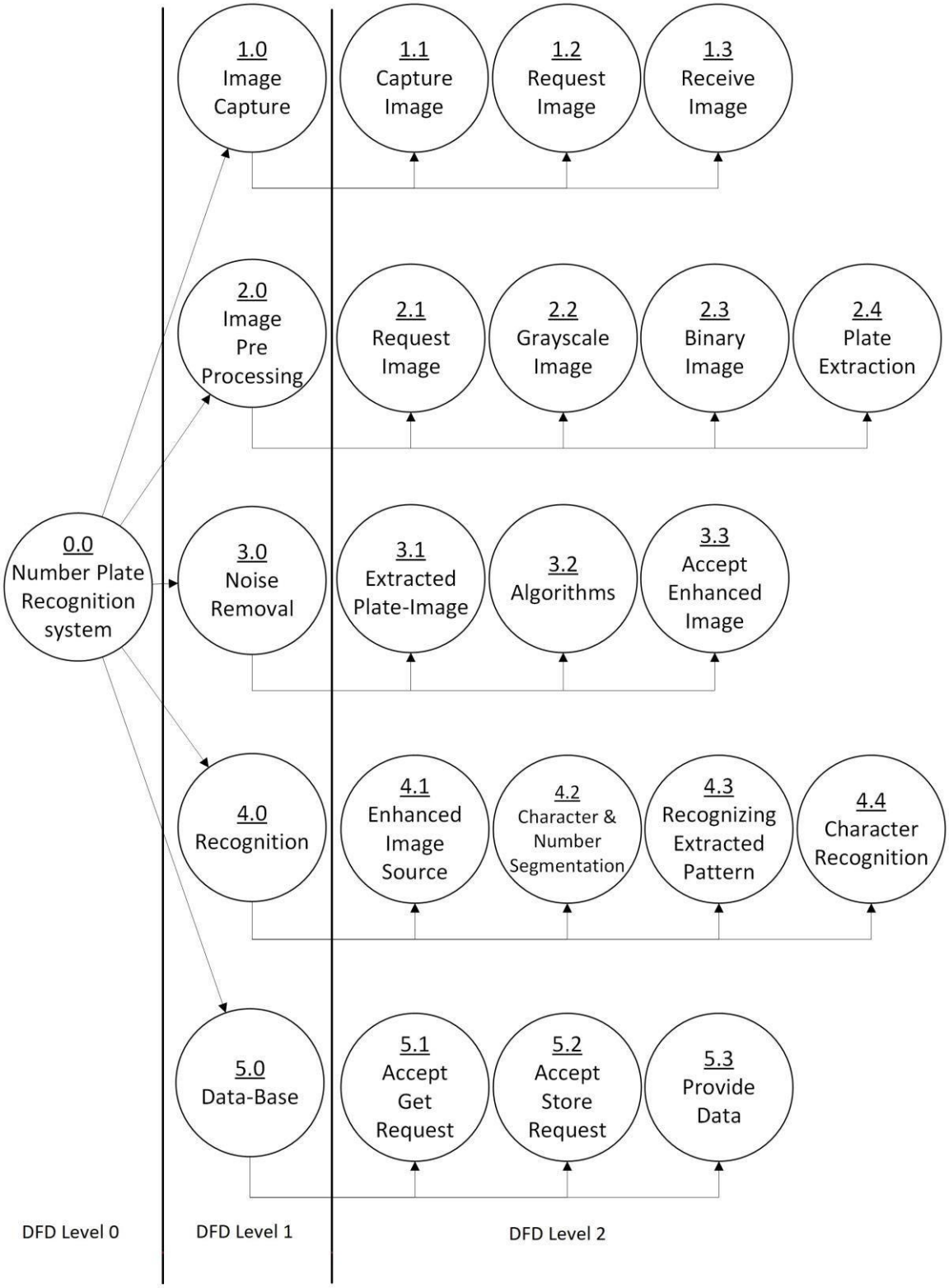
DFD Level2.4:


4.1
Enhanced Image Source
Image Passed
PROCESSING TOOLS
Seperation Of Characters
Pattern Of Characters
4.2
Character & Number Segmentation
4.3
Recognizing Extracted Pattern
Possible Pattern
Character Templates
DATABASE
Character Matching
4.4
Character Recognition
Store Recognized Number

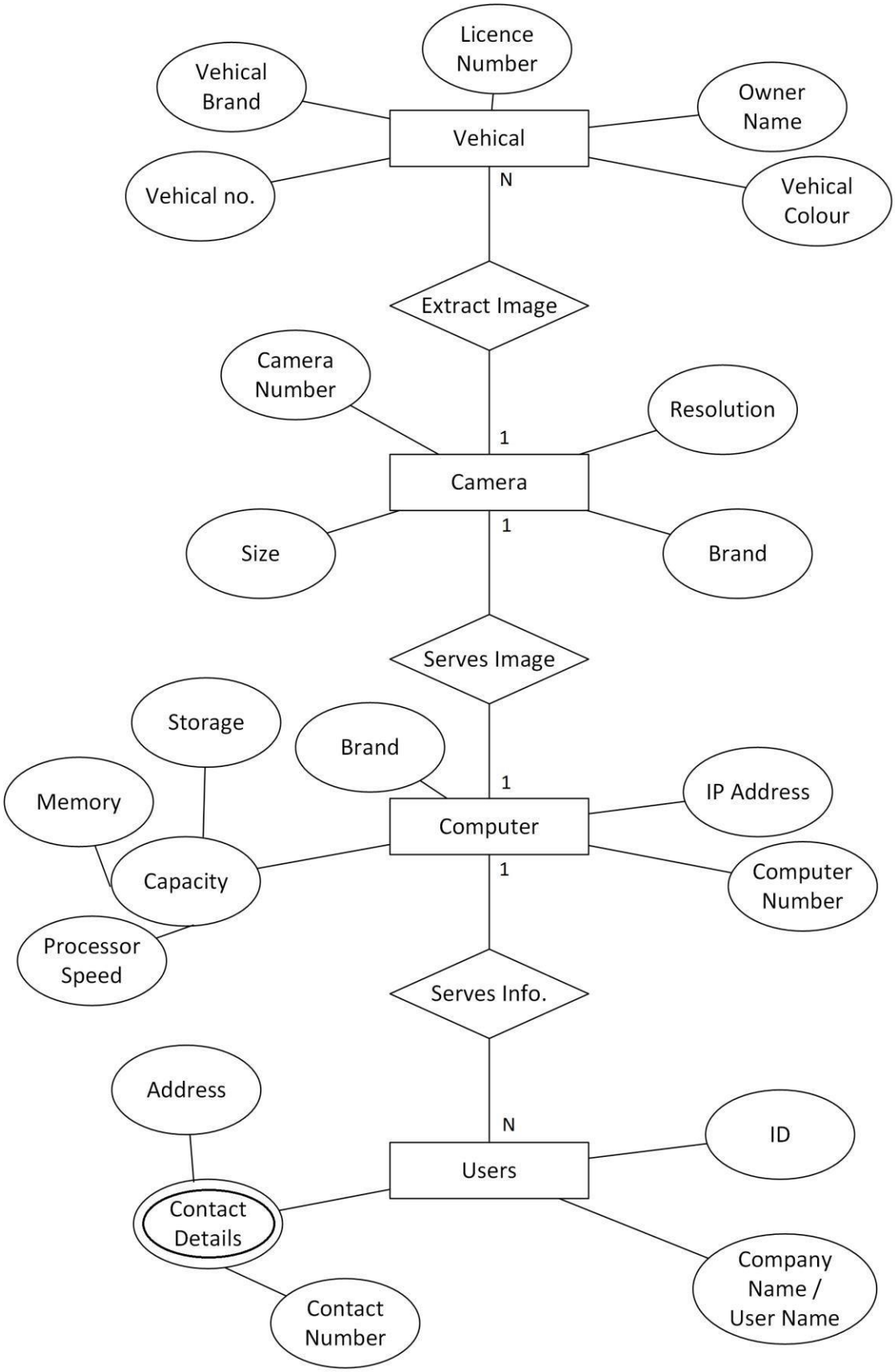
DFD Level2.5:


5.1
Accept
Get
Request
Get
Information
USER
Grant
Permission
Owner
Information
5.3
Provide
Data
Acknowledgement
TOOLS
Acknowledgement
DATA-ENTRY
(RTO)
Grant
Permission
Owner
Information
5.2
Accept
Store
Request
Store
Recognized
Information

DFD Decomposition:

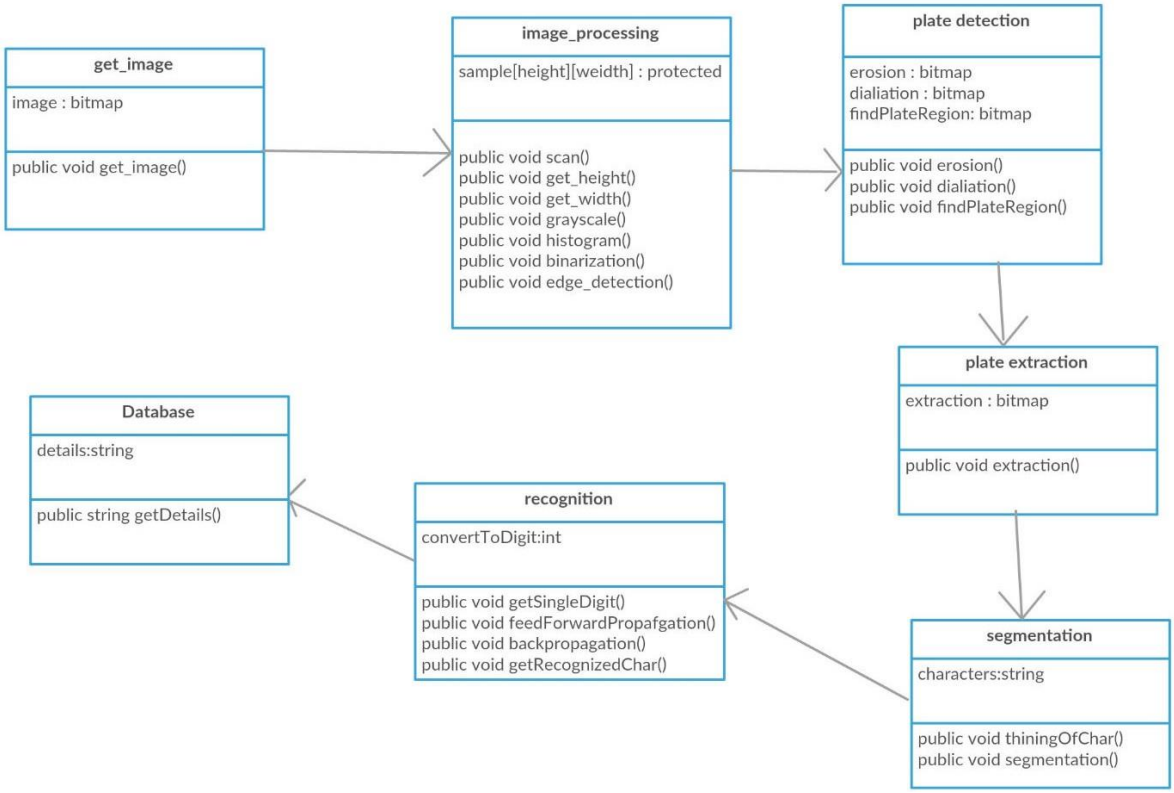


Entity-Relationship Diagram:

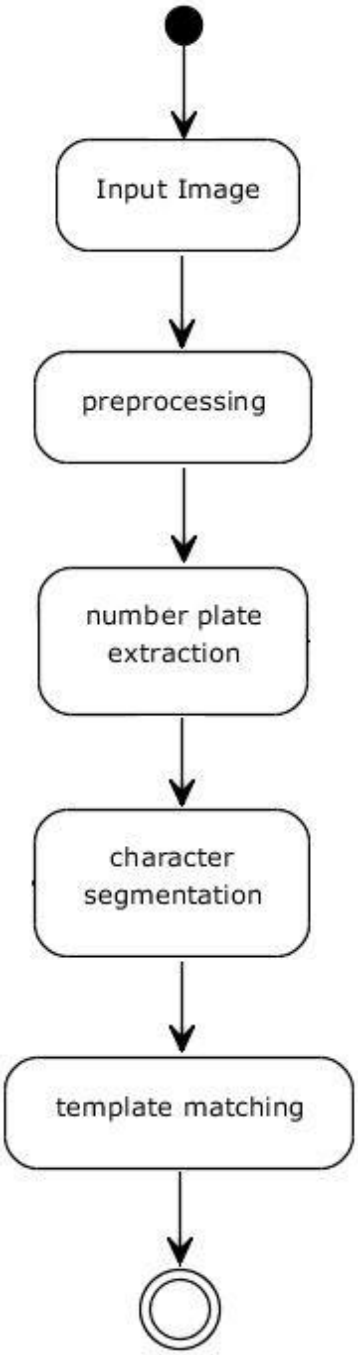


UML Diagrams:

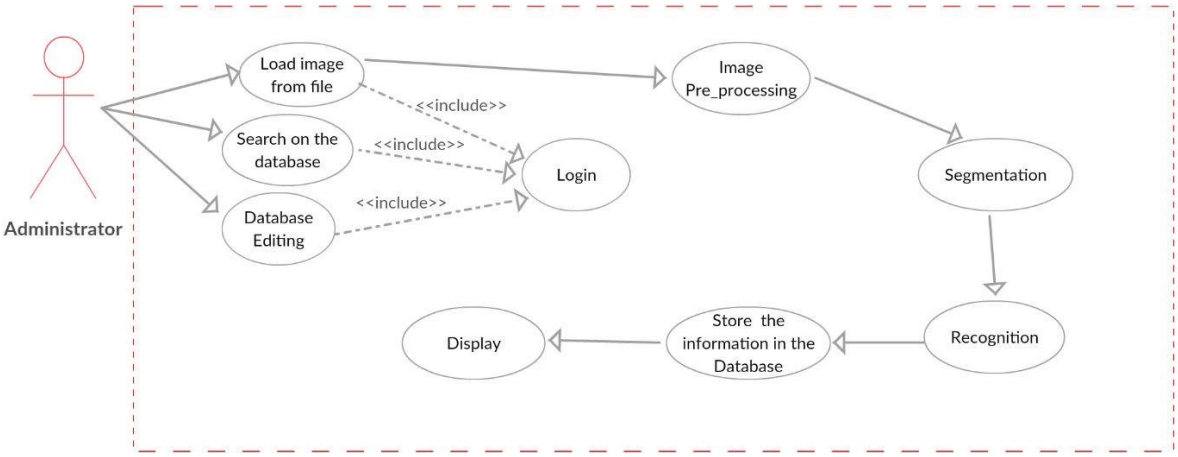
Class Diagram:



Activity Diagram:



Use-Case Diagram:



Sequence Diagram: