1. INTRODUCTION

1.1: Preamble of the Project

The purpose of this project was to build a real time application which recognizes number plates from vehicles at a gate, for example at the entrance of a parking area. The system, based on regular PC with video camera, catches video frames which include a visible car number plate and processes them. Once a number plate is detected, its digits are recognized, displayed on the User Interface or checked against a database. The focus is on the design of algorithms used for extracting the number plate from a single image, isolating the characters of the plate and identifying the individual characters.

Number plate recognition is one form of ITS technology that not only recognizes and counts vehicles, but distinguishes each as unique. For some applications, such as electronic toll collection and red-light violation enforcement, NPR records license plates alphanumeric value so the vehicle owner can be assessed the appropriate toll or fine. In others, like commercial vehicle operations or secure-access control, a vehicle's license plate is checked against a database of acceptable ones to determine whether a truck can bypass a weigh station or a car can enter a gated community or parking lot.

NPR can be used to issue violations to speeders or simply to offer speeding drivers a reminder by displaying a plate number with the vehicle's speed on a variable messaging sign. It can facilitate emissions testing by recording a plate's alphanumeric sequence while automatically analyzing tailpipe effluents, or help identify and fine violators. NPR also can monitor the time it takes vehicles to travel from one point to another; keeping traffic management centre’s apprised of transit times along busy streets and highways.

At international border crossings, license plates-- the only universal vehicle identifier-- can be checked against a database of "hot" cars to locate stolen vehicles and plates or those registered to fugitives, criminals, or smuggling suspects.

1.2 Existing System

1.3 Proposed System

2. Project Scenario:

1. Name of the Project: Number Plate Recognition

2. Objective/Vision: The main objective of the proposed system is to build a real time application which recognizes number plates from vehicles at a gate, for example at the entrance of a parking area. Our project aimed to enable the installation of our solution by using a simple video camera and a standard PC in the college parking lot.

Our project is to develop Number Plate Recognition System and to implement it. In this system primarily we take a picture of the number plate and process the image using optical character recognition techniques to get the output in the form of text string. Number Plate Recognition system is comprised of a video image-acquisition subsystem, a CPU for image processing and control, a software-based character recognition engine, and a storage or transmission subsystem for electronically recording plate contents and data such as date, time, and location.

3. Users of the System: In Number plate recognition, the operations can be done by Administrator and User.

Administrator:

- Administrator creates the multiple users
- Administrator can create his own username and password
- Administrator updates the entry time and exit time of the vehicle.
- Administrator stores the details of the persons who are visiting
- Administrator can select a particular day and can see who visited on that day.
- Administrator can search for a particular Number plate in the database.
- Administrator can select a particular day and can see who visited on that day.

User:

- User logs in by the username and password which are provided by administrator.

- User captures the number plate of moving vehicles.
- User creates new vehicle registrations.
- User can search for a particular Number plate in the database.
- User can see the day report.

4. **Functional Requirements**

5. Non-Functional Requirements

6. Optional Features

7. User Interface Priorities

8. Reports

9. **Other Important Issues**

10. Team Size: 4

11. Technologies to be used: C#

12. Tools to be used: .NET 2008, SQL server 2005, M.S Office 2007

13. **Final Deliverable must include**

14. Advantages of using SQL Server management Studio:

- Redundancy can be avoided.
- Inconsistency can be eliminated.
- Data can be shared.
- Standards can be enforced.
- Security restrictions can be applied.
- Integrity can be maintained.
- Conflicting requirements can be balanced.

3. Detailed SRS

3.1 Purpose

The main objective of the proposed system is to build a real time application which recognizes number plates from vehicles at a gate, for example at the entrance of a parking area. Our project aimed to enable the installation of our solution by using a simple video camera and a standard PC in the college parking lot.

Our project is to develop Number Plate Recognition System and to implement it. In this system primarily we take a picture of the number plate and process the image using optical character recognition techniques to get the output in the form of text string. Number Plate Recognition system is comprised of a video image-acquisition subsystem, a CPU for image processing and control, a software-based character recognition engine, and a storage or transmission subsystem for electronically recording plate contents and data such as date, time, and location.

3.2 Scope

The project that we are implementing is a real time project, In India we have not used such kind of technology for maintaining log, such a technology is implemented only in European countries. Since this software has not implemented before we have a great scope for implementing this project.

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

- As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
- Because it is based on object-oriented design, any further changes can be easily adaptable.
- Based on the future security issues, security can be improved.
- By using sensors we can make complete automation of recognition.
- By use high definition camera we can solve some image clarity problems.
- We can implement our project through mobile Devices or hand held devices

The scope of this project is, we can use in any organization for storing log history and retrieving valuable information from it.

3.3 Definitions,Acronyms and Abbreviations

Previous Methods

Early NPR systems suffered from a low recognition rate, lower than required by practical systems. The external effects (sun and headlights, bad plates, wide number of plate's types) and the limited level of the recognition software and vision hardware yielded low quality systems.

However, recent improvements in the software and hardware have made the NPR systems much more reliable and wide spread. You can now find these systems in numerous installations and the numbers of systems are growing exponentially, efficiently automating more and more tasks in different market segments. In many cases the NPR unit is added as retrofit in addition to existing solutions, such as a magnetic card reader or ticket dispenser/reader, in order to add more functionality to the existing facility.

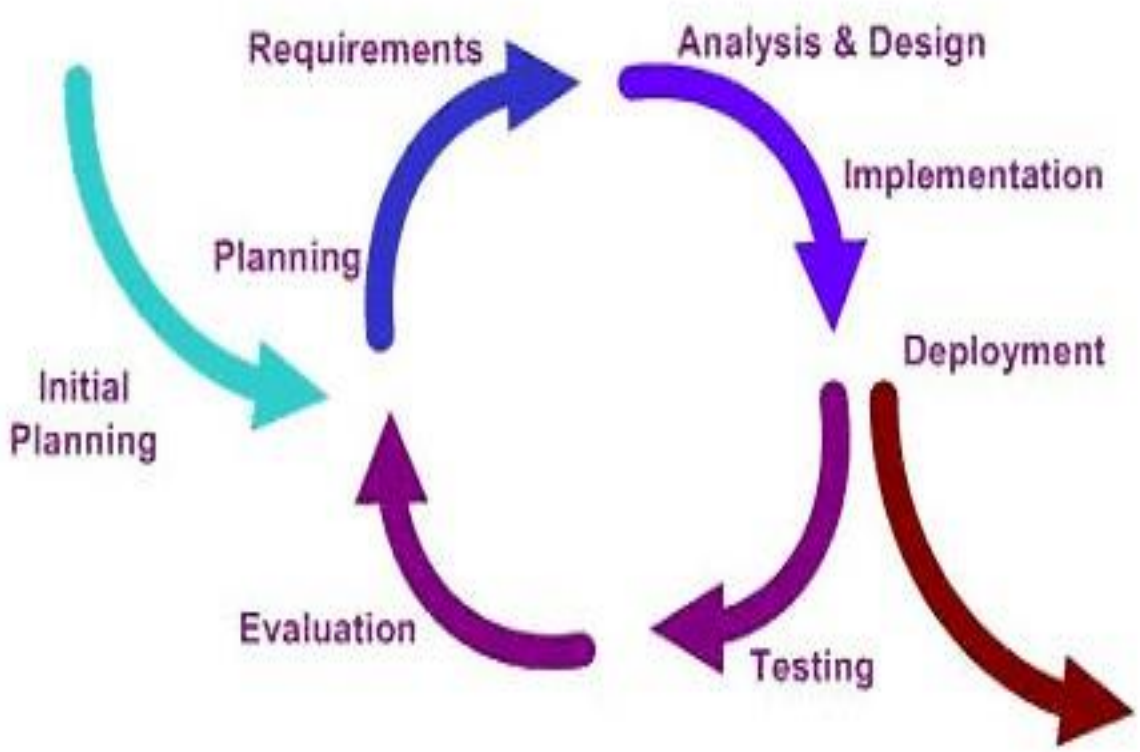
Even if the recognition is not absolute, the application that depends on the recognition results can compensate the errors and produce a virtually flawless system. For example, when comparing the recognition of the entry time of a car to the exit time in order to establish the parking time, the match (of entry verses exit) can allow some small degree of error without making a mistake. This intelligent integration can overcome some of the NPR flaws and yield dependable and fully automatic systems.

Project Approach and Motivation of the Project

Approach:

The basic idea behind iterative enhancement is to develop a software system incrementally, allowing the developer to take advantage of what was being learned during the development of earlier, incremental, deliverable versions of the system. Learning comes from both the development and use of the system, where possible key step in the process are to start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving sequence of versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added.

The procedure itself consists of the initialization step, the iteration step, and the project control list. The initialization step creates a base version of the system. The goal for this initial implementation is to create a product to which the user can react. It should offer a sampling of the key aspects of the problem and provide a solution that is simple enough to understand and implement easily. To guide the iteration, process a project control list is created that contains a record of all the tasks that are needed to be performed. It includes such items as new features to be implemented and areas of redesign of the existing solution. The control list is constantly being revised as result of the analysis phase.



The iteration involves the redesign and implementation of the task from the project control list and the analysis of the current version of the system. The goal for the design and the implementation of any iteration is to be simple, straight forward and modular, supporting redesign at that stage or as a task added to the project control list. The level of design detail is not

dictated by the interactive approach. In a light-weight iterative project the code may represent the major source of documentation of the system; however; in a mission-critical iterative project a formal software design document may be used. The analysis of iteration is based upon user feedback, and the program analysis facilities available. It involves analysis of the structure, modularity, usability, reliability, efficiency & achievement of goals. The project control list is modified in light of the analysis results.

Motivation:

Vehicle License plates were originally meant to be recognized by humans, especially government officials, such as policemen and transport personnel. If a car is involved with in robbery or stealing, for instance, then seeing its plate helps tracking down the criminals. If a stolen car is found, its plate helps to find the owner.

A typical NPI/R system is comprised of a video image-acquisition subsystem, a CPU for image processing and control, a hardware- or software-based character recognition engine, and storage or transmission subsystem for electronically recording plate contents and data such as date, time, and location.

At any NPR system's heart is its recognition engine and the embedded algorithm. It is important to understand that the system supplier decides how to code the recognition engine. The user, OEM, or integrator takes that algorithm along with the system. A basic understanding of how the recognition engine interprets image content is central to confirming that the overall "solution" can handle a given application.

The *correlation* or *template matching* approach to character recognition is straight-forward and can be reliable, provided the target is "cooperative" and the application remains invariant. As the name implies, once each character is isolated, the recognition engine attempts to match it against a set of predefined standards. Any condition-- lighting, viewing angle, obscuration, plate size, font-- that causes a character to vary from the standard is likely to confuse the engine and return a questionable result.

Structural analysis uses a decision-tree to assess the geometric features of each character's contour. The technique can be somewhat tolerant of variations in size, tilt, and perspective. As a simple example, the characters like B, D, 6, and 9. Features that might be used to distinguish

them are the number of loops in each character-- one or two-- and the vertical position of the loop-- top, central, or bottom. Two loops point to a B, and one loop leads to the next branch of the tree. A loop at the top indicates a 9; if the loop is central, it's a D, and a loop at the bottom means a 6. Characters without loops (E, M, N, hyphen, etc.) require additionally complex, time-consuming analysis.

3.4 Technologies to be used

The project "Number Plate Recognition" is designed using Visual Studio .NET framework version 2.0. The coding language used is C#.Net.

- **Microsoft.NET Framework:**

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet.

The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

Principal design features:

Interoperability:

Because interaction between new and older applications is commonly required, the .NET Framework provides means to access functionality that is implemented in programs that execute outside the .NET environment. Access to components is provided in the System.Runtime.InteropServices and System.EnterpriseService namespaces of the framework.

Architecture:

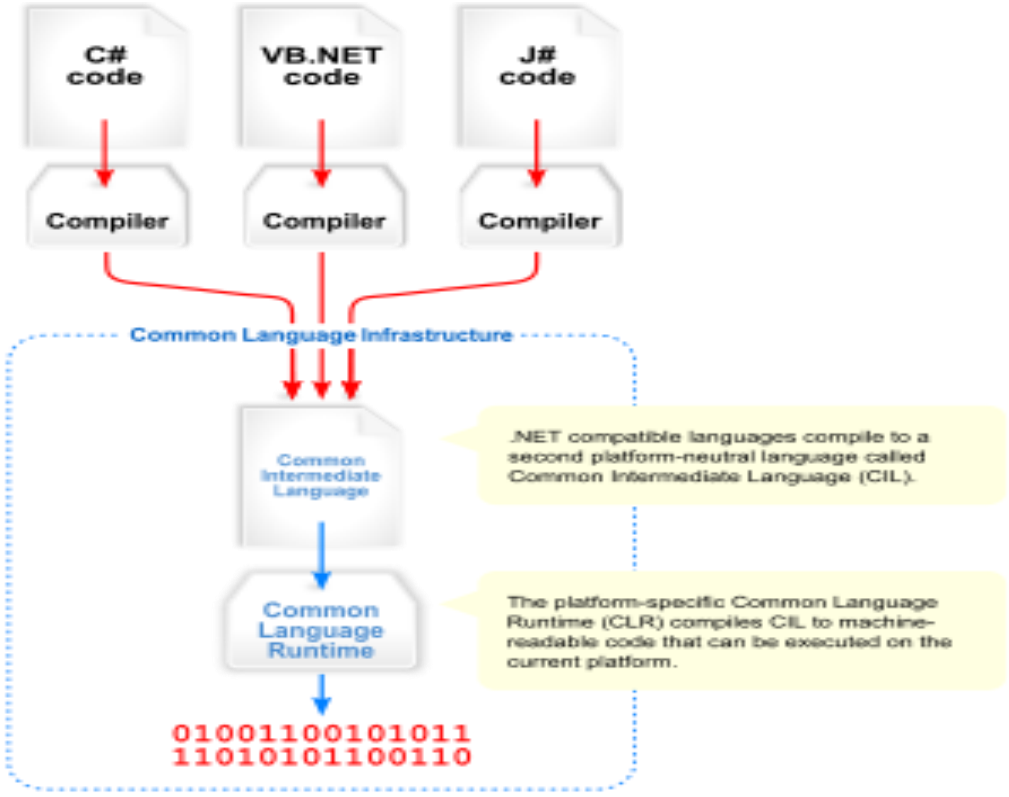


Fig: Visual overview of the Common Language Infrastructure (CLI)

Common Language Infrastructure:

The core aspects of the .NET framework lie within the Common Language Infrastructure, or CLI. The purpose of the CLI is to provide a language-neutral platform for application development and execution, including functions for exception handling, garbage

collection, security, and interoperability. Microsoft's implementation of the CLI is called the Class library

Microsoft .NET Framework includes a set of standard class libraries. The class library is organized in a hierarchy of namespaces. Most of the built in APIs are part of either System.* or Microsoft.* namespaces. It encapsulates a large number of common functions, such as file reading and writing, graphic rendering, database interaction, and XML document manipulation, among others. The .NET Framework class library is divided into two parts: the Base Class Library and the Framework Class Library.

Versions:

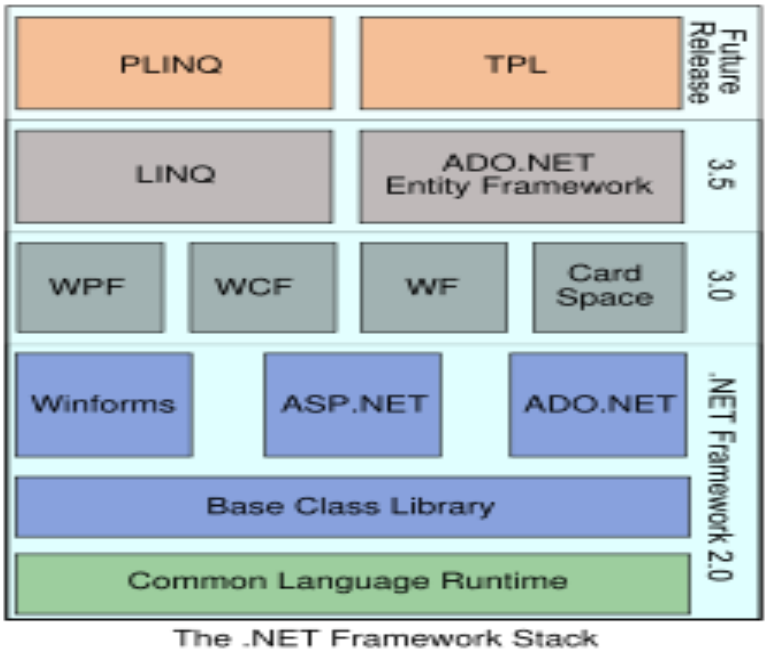


Fig : **The .Net Framework Stack**

The .NET Framework stacks...

Version	Version Number	Release Date
1.0	1.0.3705.0	2002-01-05

1.1	1.1.4322.573	2003-04-01
2.0	2.0.50727.42	2005-11-07
3.0	3.0.4506.30	2006-11-06
3.5	3.5.21022.8	2007-11-09

Table.Net Framework Versions

The Common Language Runtime:

Central to the .NET framework is its run-time execution environment, known as the **Common Language Runtime** (**CLR**) or the **.NET runtime**. Code running under the control of the CLR is often termed **managed** code.

However, before it can be executed by the CLR, any source code that we develop (in C# or some other language) needs to be compiled. Compilation occurs in two steps in .NET:

- Compilation of source code to **Microsoft Intermediate Language** (**MS-IL**)
- Compilation of IL to platform-specific code by the CLR

Language Interoperability:

How the use of IL enables platform independence, and how JIT compilation should improve performance. However, IL also facilitates **language interoperability**. Simply put, you can compile to IL from one language, and this compiled code should then be interoperable with code that has been compiled to IL from another language.

Intermediate Language:

From what we learned in the previous section, Intermediate Language obviously plays a fundamental role in the .NET Framework. As C# developers, we now understand that our C# code will be compiled into Intermediate Language before it is executed (indeed, the C# compiler *only* compiles to managed code). It makes sense, then, that we should now take a closer look at the main characteristics of IL, since any language that targets .NET would logically need to support the main characteristics of IL too.

Here are the important features of the Intermediate Language:

- Object-orientation and use of interfaces
- Strong distinction between value and reference types
- Strong data typing
- Error handling through the use of exceptions
- Use of attributes

Object Orientation and Language Interoperability:

Working with .NET means compiling to the Intermediate Language, and that in turn means that you will need to be programming using traditional object-oriented methodologies. That alone is not, however, sufficient to give us language interoperability.

An associated problem was that, when debugging, you would still have to independently debug components written in different languages. It was not possible to step between languages in the debugger. So what we *really* mean by language interoperability that classes written in one language should be able to talk directly to classes written in another language. In particular:

- A class written in one language can inherit from a class written in another language
- The class can contain an instance of another class, no matter what the languages of the two classes are:
    1. An object can directly call methods against another object written in another language.
    2. Objects (or references to objects) can be passed around between methods.

The Role of C# in .Net Enterprise Architecture:

C# requires the presence of the .NET runtime, and it will probably be a few years before most clients – particularly most home machines – have .NET installed. In the meantime, installing a C# application is likely to mean also installing the .NET redistributable components. Because of that, it is likely that the first place we will see many C# applications

is in the enterprise environment. Indeed, C# arguably presents an outstanding opportunity for organizations that are interested in building robust, n-tiered client-server applications.

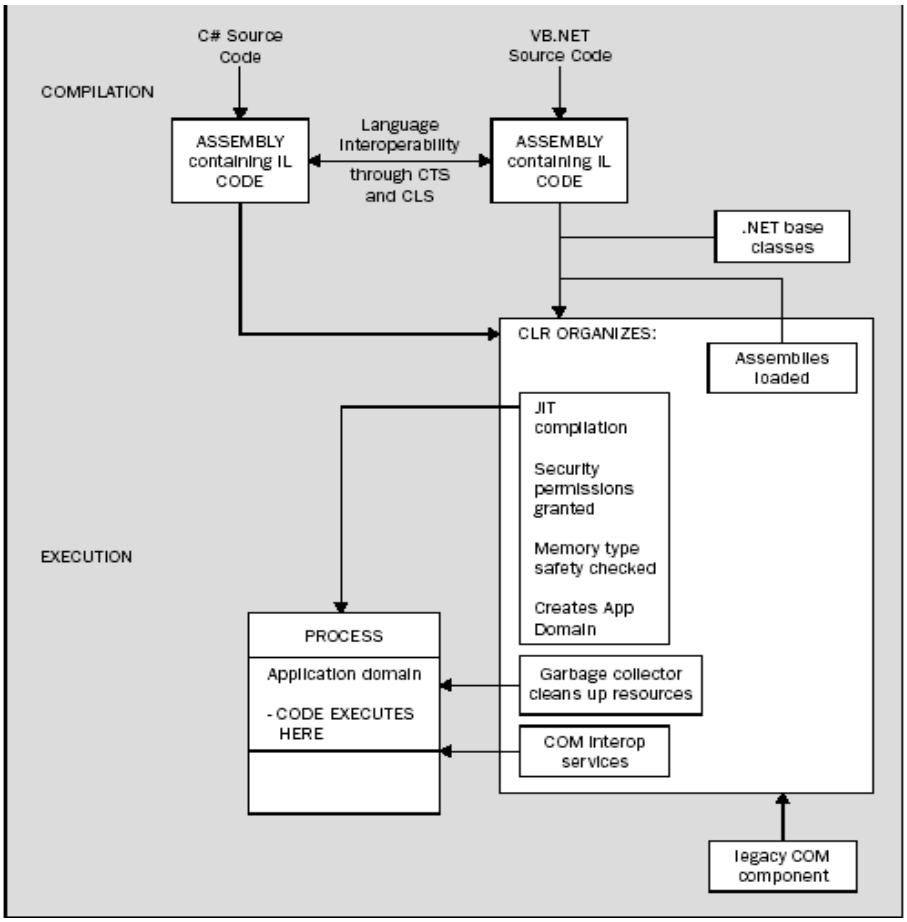


Fig 3.3.3 C# in .Net Enterprise Architecture

SQL SERVER

Data Base:

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

SQL Server Tables:

SQL Server stores records relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

Primary Key:

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

Relational Database

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the tables and enables you to define relationships between the tables.

Foreign Key

When a field is one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table.

Referential Integrity

Not only does SQL Server allow you to link multiple tables, it also maintains consistency between them. Ensuring that the data among related tables is correctly matched is referred to as maintaining referential integrity.

Data Abstraction

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained.

3.6 Overview

The main objective of the proposed system is to build a real time application which recognizes number plates from vehicles at a gate, for example at the entrance of a parking area. Our project aimed to enable the installation of our solution by using a simple video camera and a standard PC in the college parking lot.

Our project is to develop Number Plate Recognition System and to implement it. In this system primarily we take a picture of the number plate and process the image using optical character recognition techniques to get the output in the form of text string. Number Plate Recognition system is comprised of a video image-acquisition subsystem, a CPU for image processing and control, a software-based character recognition engine, and a storage or transmission subsystem for electronically recording plate contents and data such as date, time, and location.

3.7 Overall Description

3.7.1 product perpective

3.7.2 :Requirements:

A requirement is a feature that must be included in the actual design and implementation, getting to know the system to be implemented is of prime importance.

Software Requirements:

Operating System	:	Windows XP/2003
Programming Language	:	.Net Frame work 2.0

IDE	:	Microsoft Visual Studio .Net 2005
Database	:	SQL Server 2005.
Service	:	Internet Information Services
Application	:	MS Office 2003/2007

Hardware Requirements:

Processor	:	Pentium IV
Hard Disk	:	40GB
RAM	:	256MB
Camera	:	Closed Circuit Digital Camera (640x480)

3.7.4 Product Function

3.7.5 User Charcacteristics

- User logs in by the username and password which are provided by administrator.
- User captures the number plate of moving vehicles.
- User creates new vehicle registrations.
- User stores the details of the persons who are visiting.
- User updates the entry time and exit time of the vehicle.
- User can search for a particular Number plate in the database.
- User can see the day report.
- User can select a particular day and can see who visited on that day.

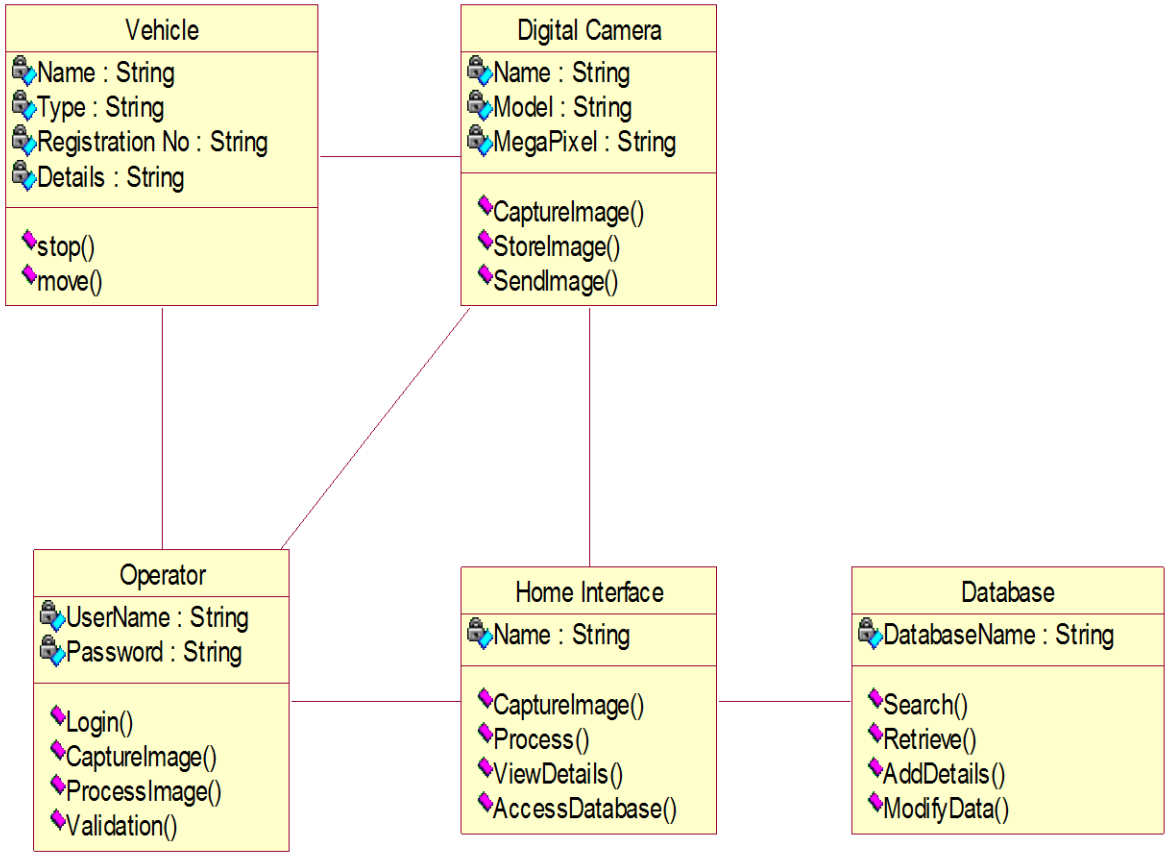
3.7.6 Constraints

3.8 Assumptions and Dependencies

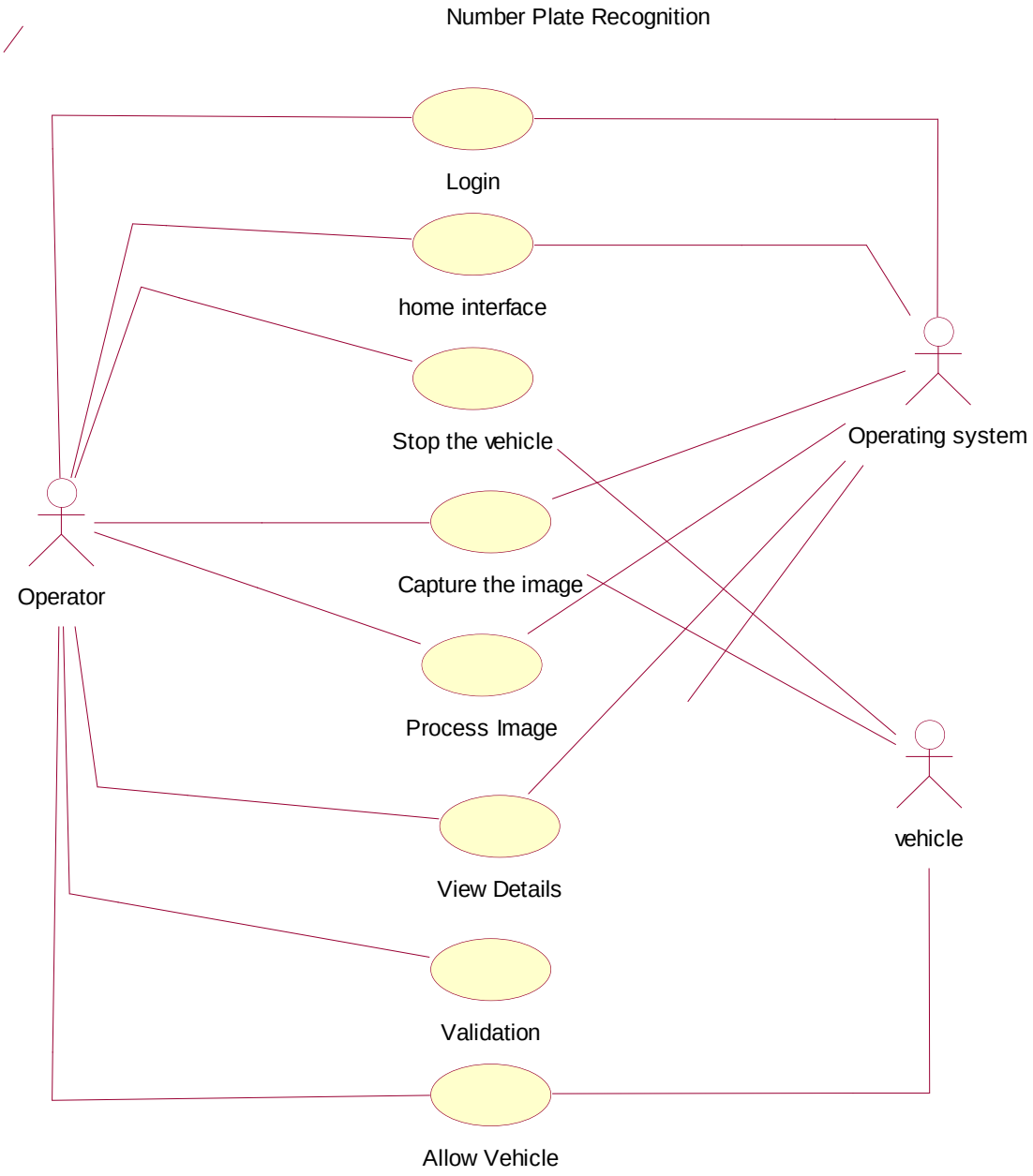
3.9 Supplementary Requirements

4. Design

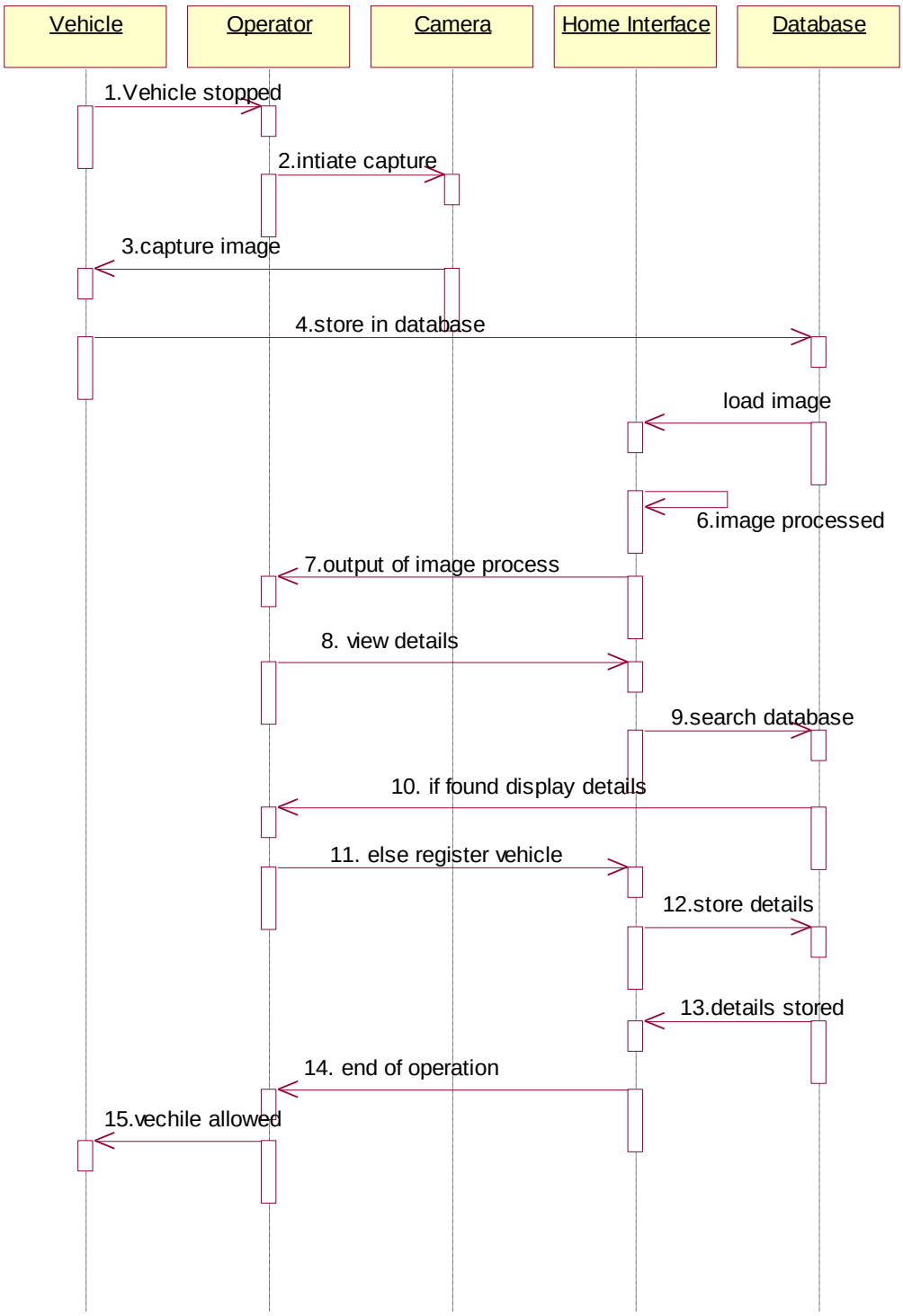
4.1 Class Diagram



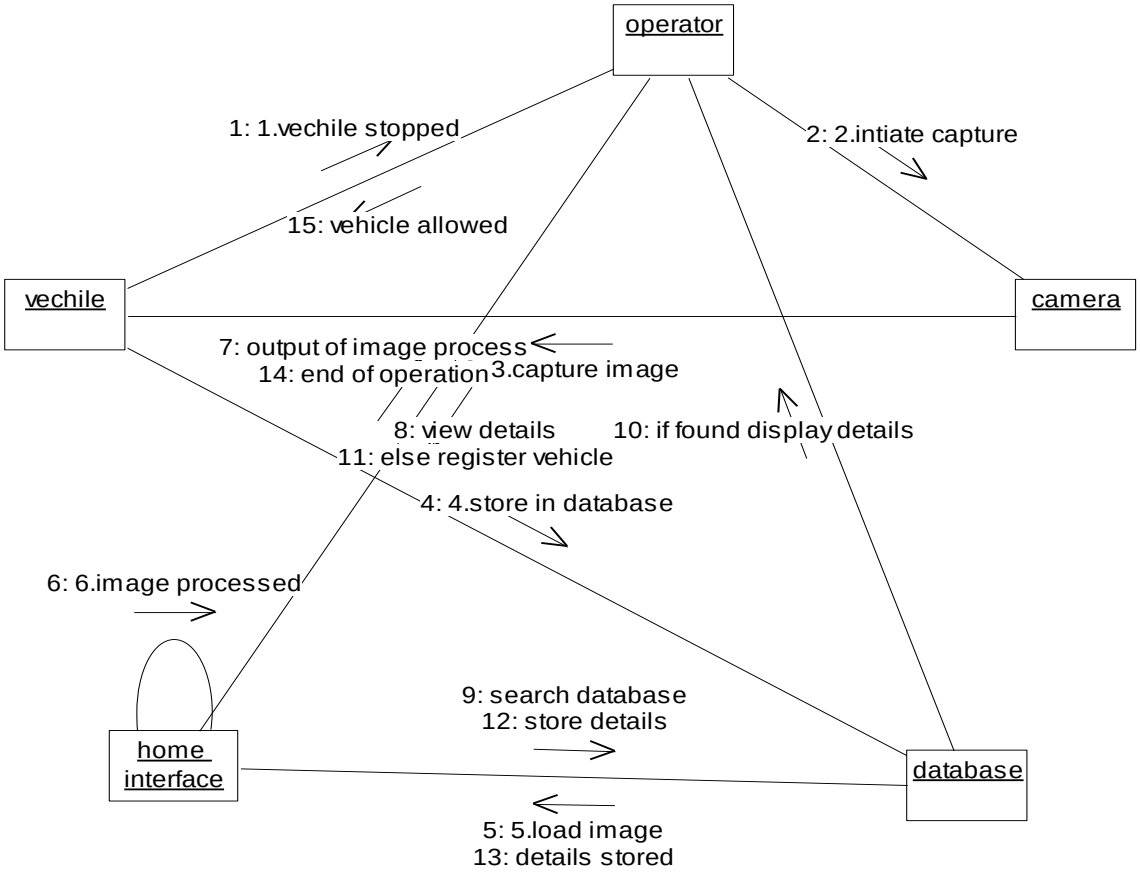
4.2 UseCase Diagram



4.3 Sequence Diagram



4.4 Collaboration Diagram



4.5 Activity Diagrams

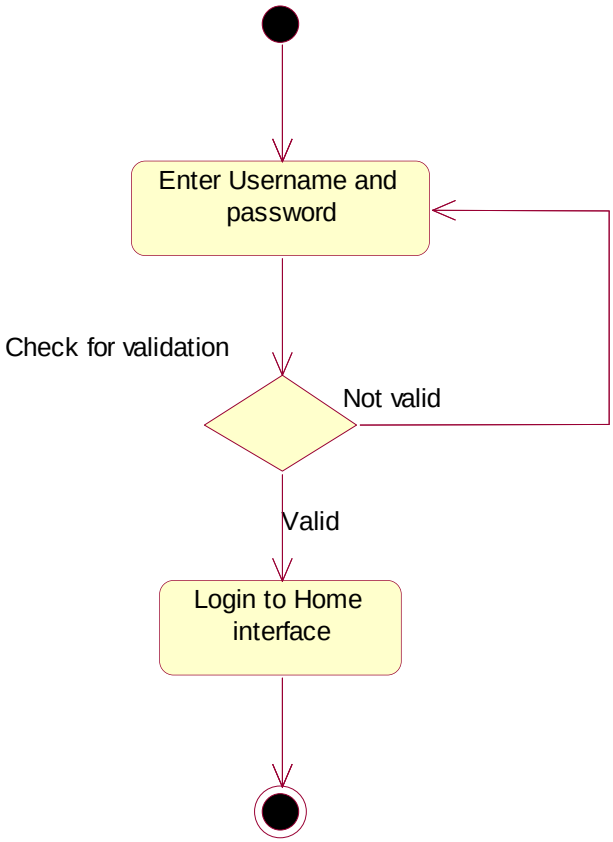


Fig: Activity Diagram for Login

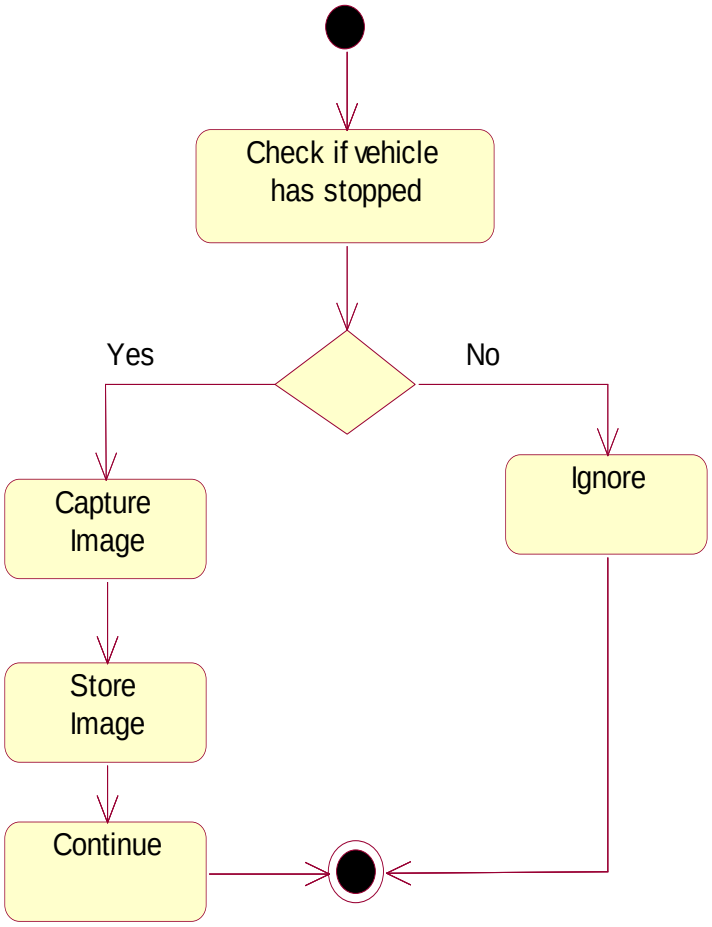


Fig: Activity Diagram for Home Interface

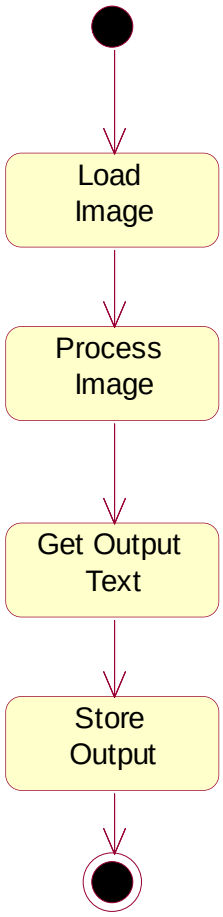


Fig: Image Processing

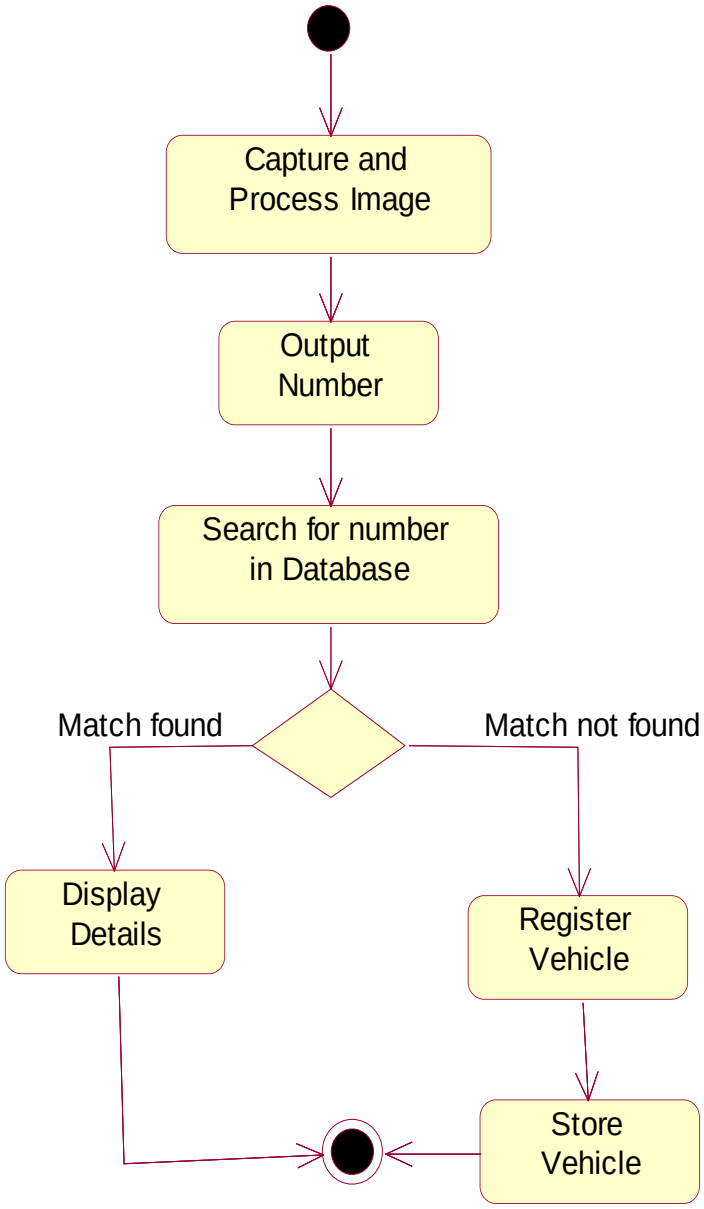


Fig: Activity Diagram to Display Details

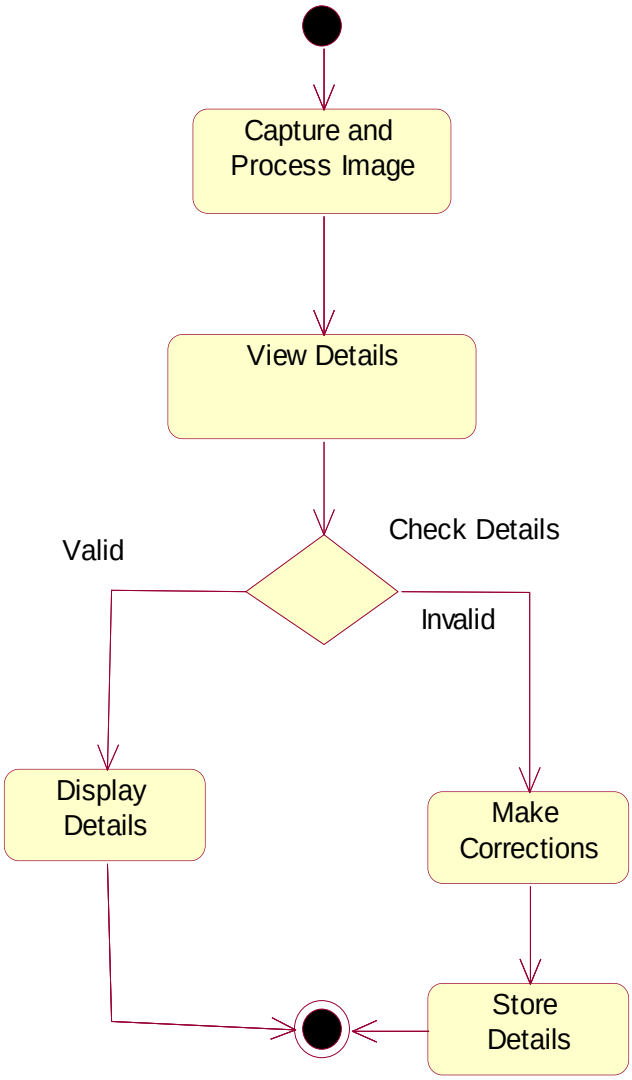


Fig: Activity Diagram For Validation

5. Database Schema

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

6.Sample Code

7. Screen Shots

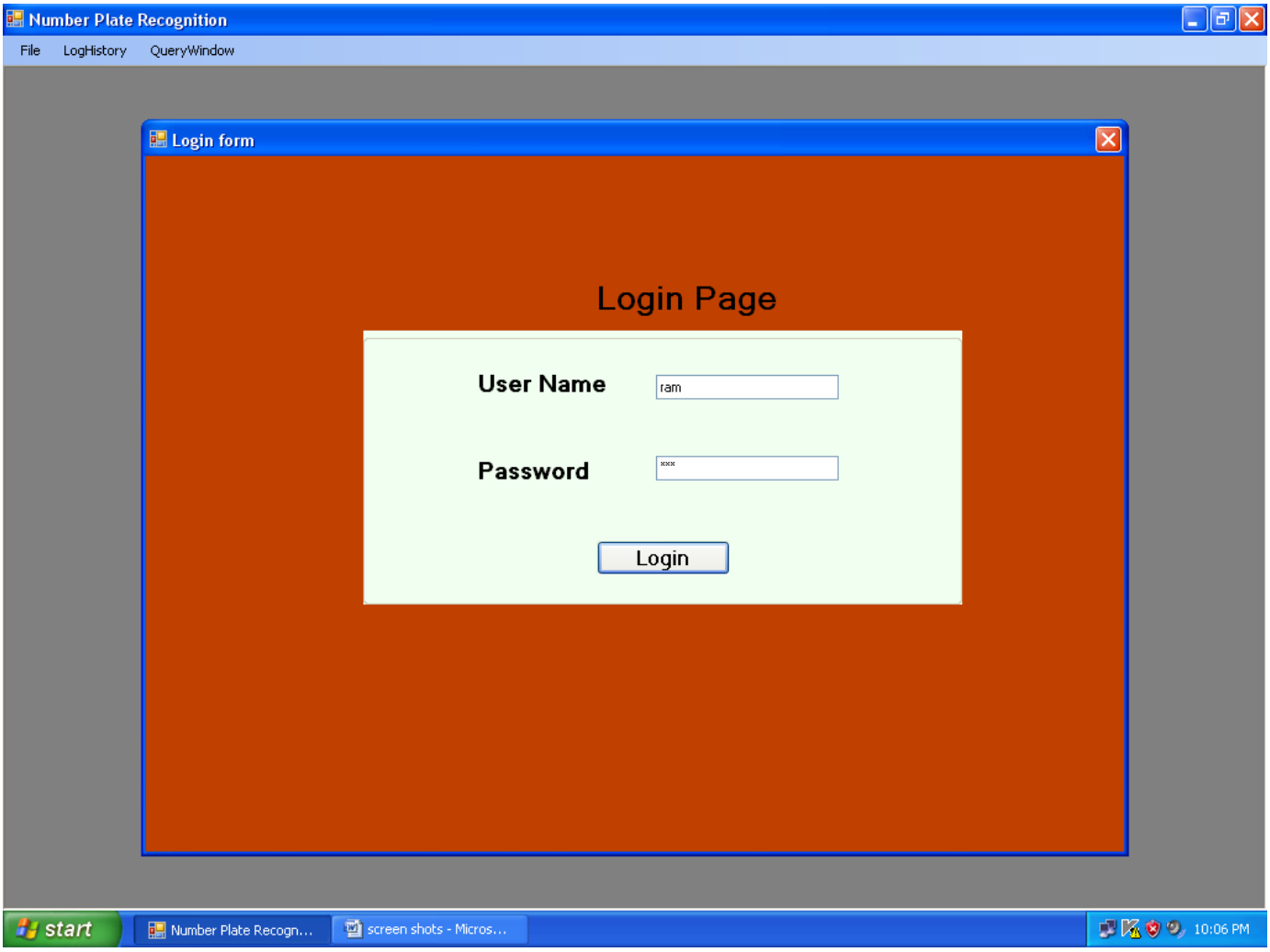


Fig: Login page

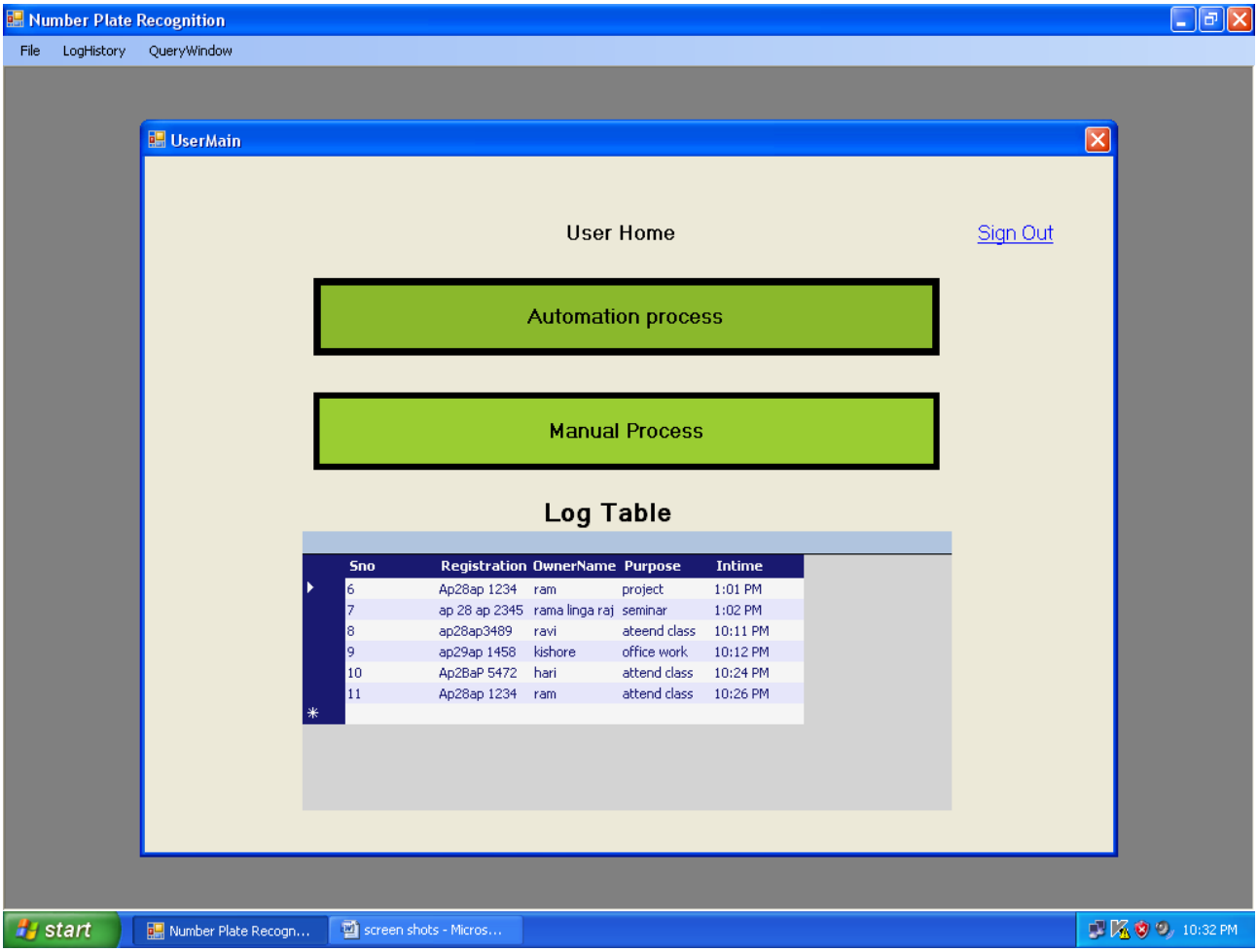


Fig: Main Page

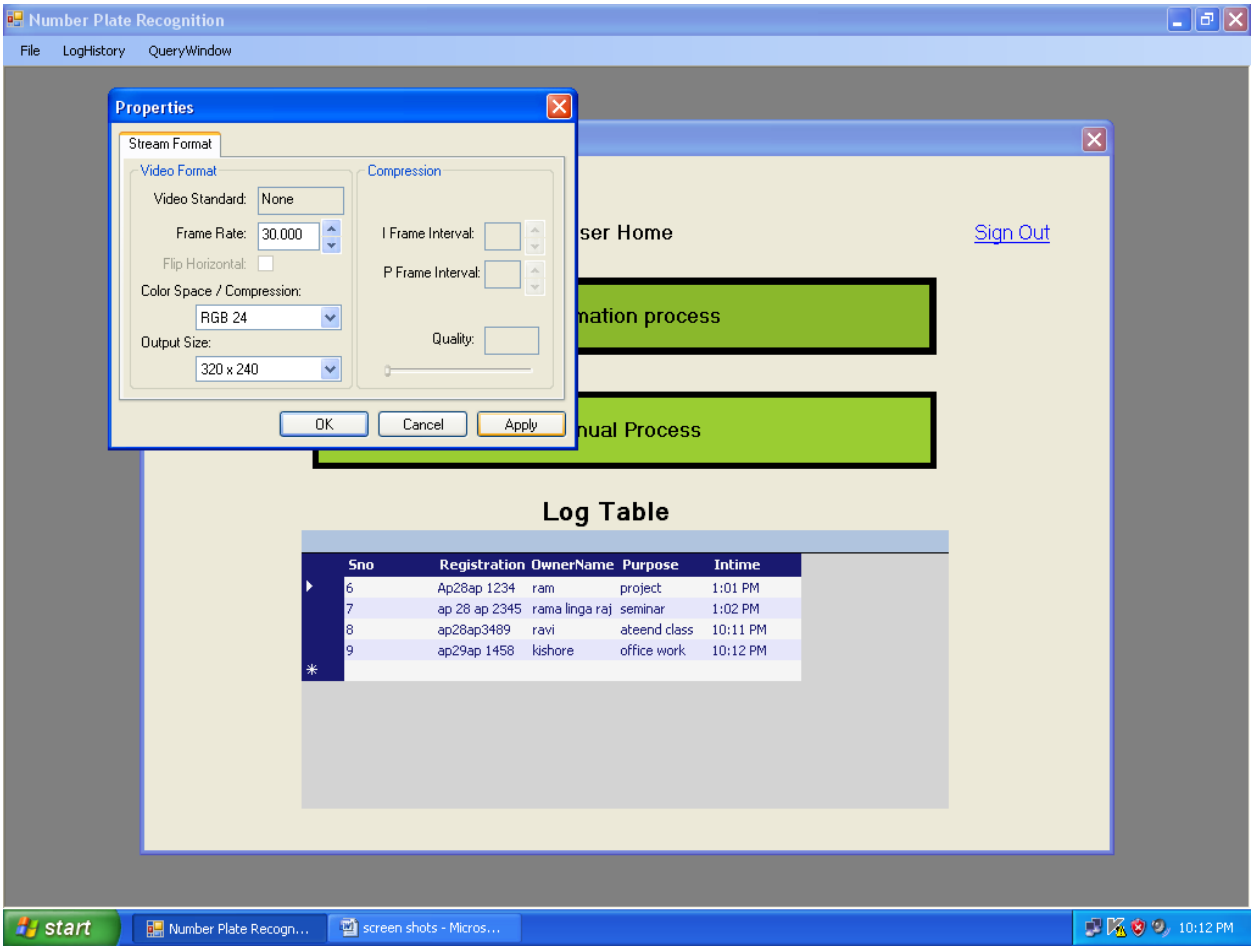


Fig: Selecting automatic process

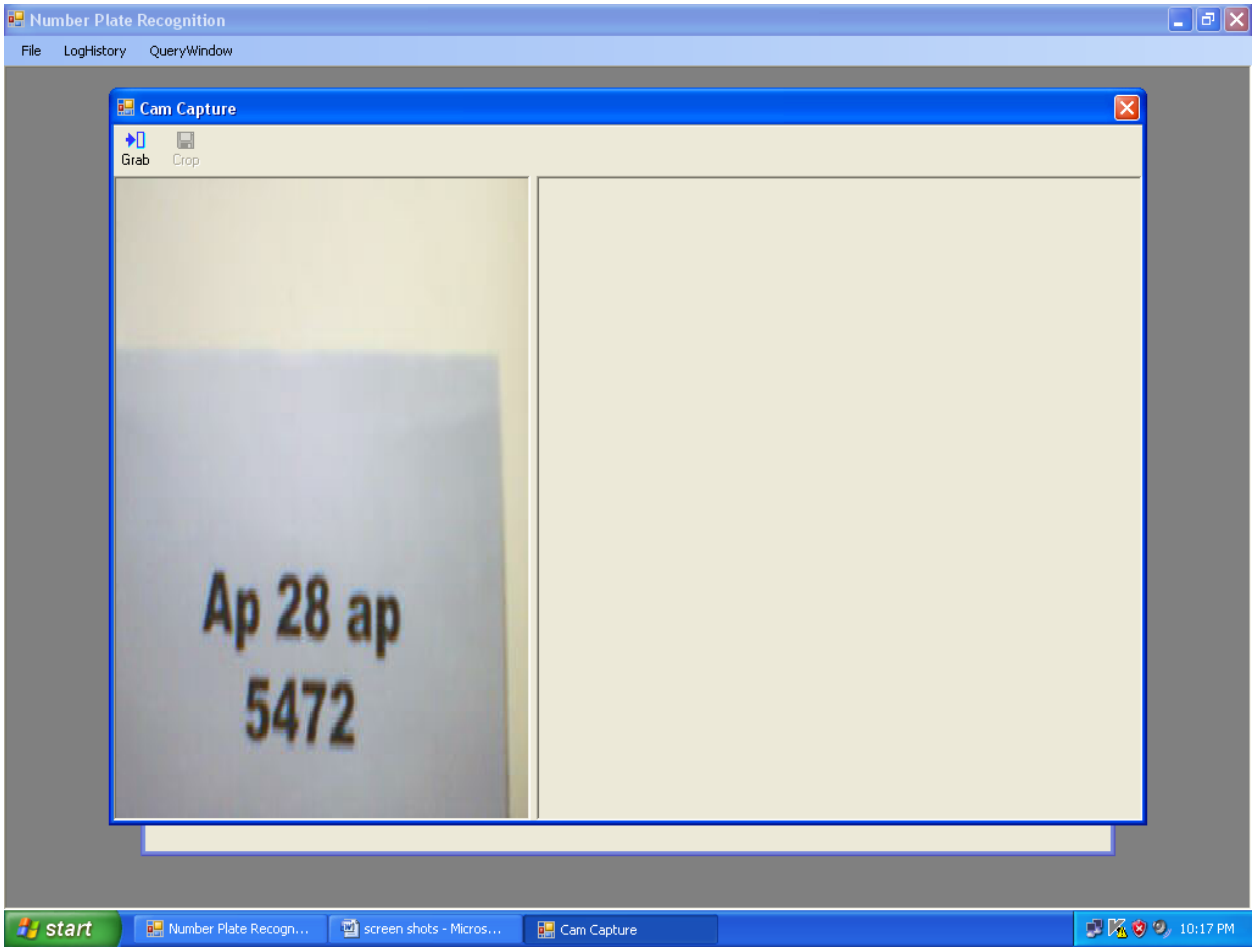


Fig: Capturing number plate

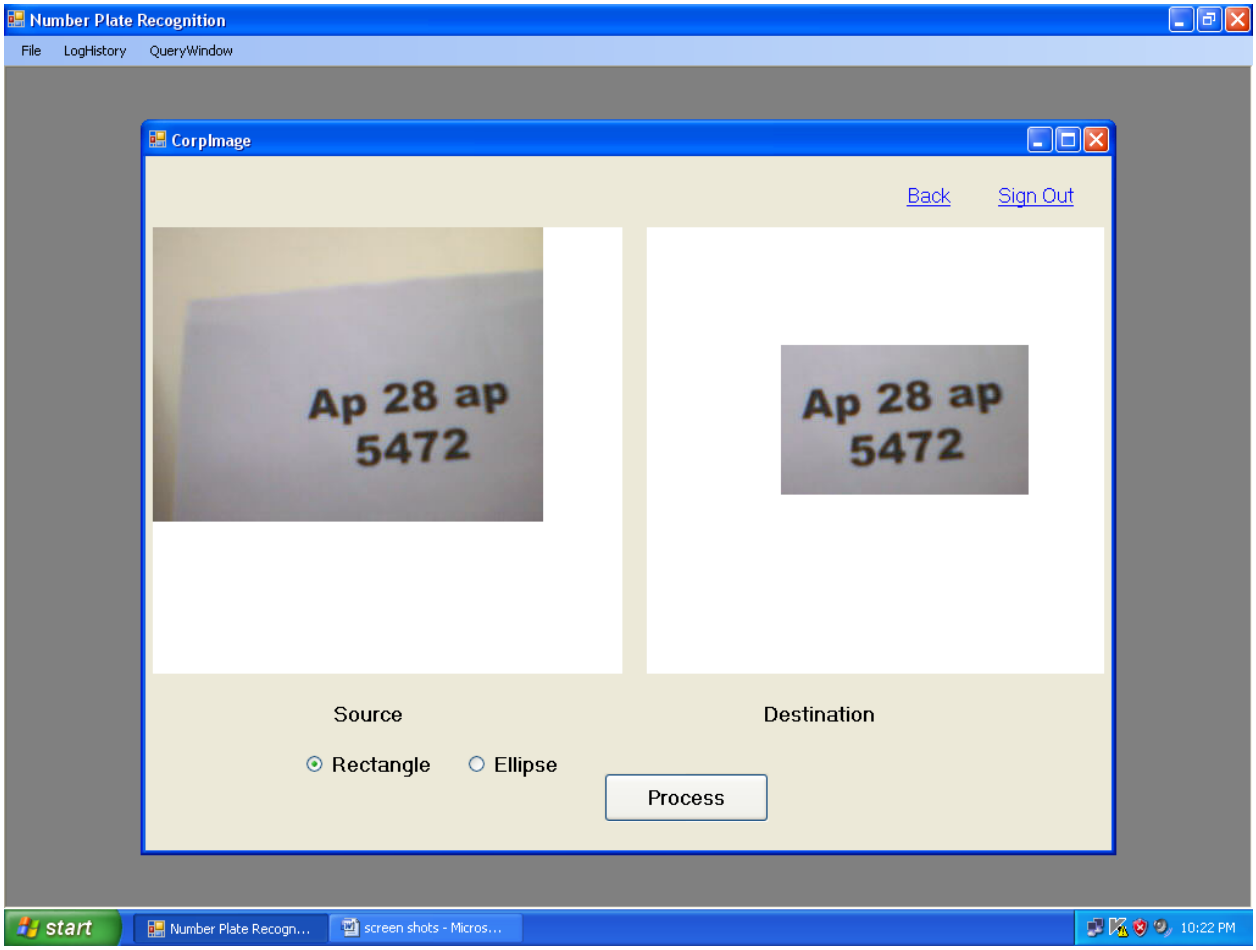


Fig: Cropping the image

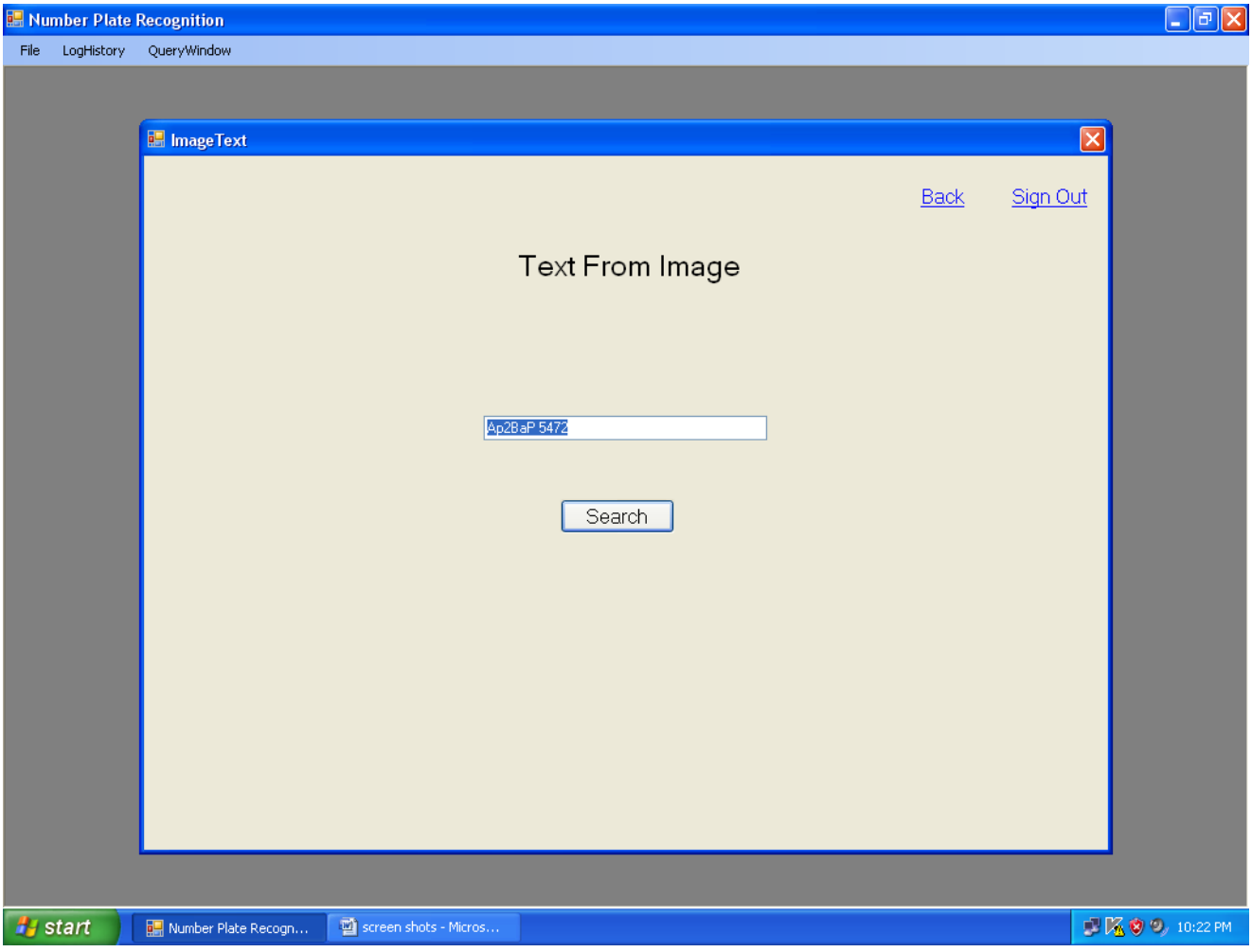


Fig: Text from the captured image

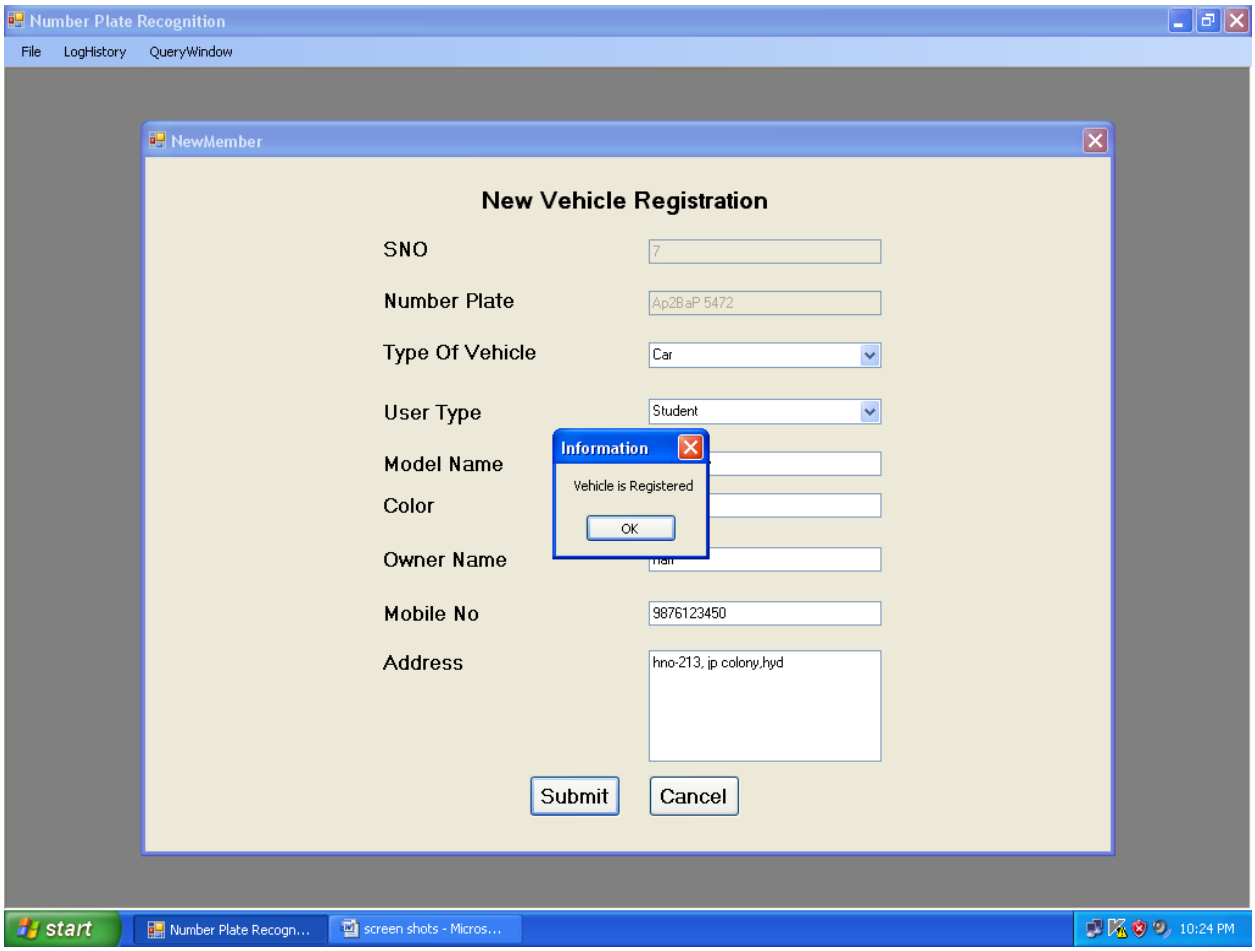


Fig: Registering the new vehicle

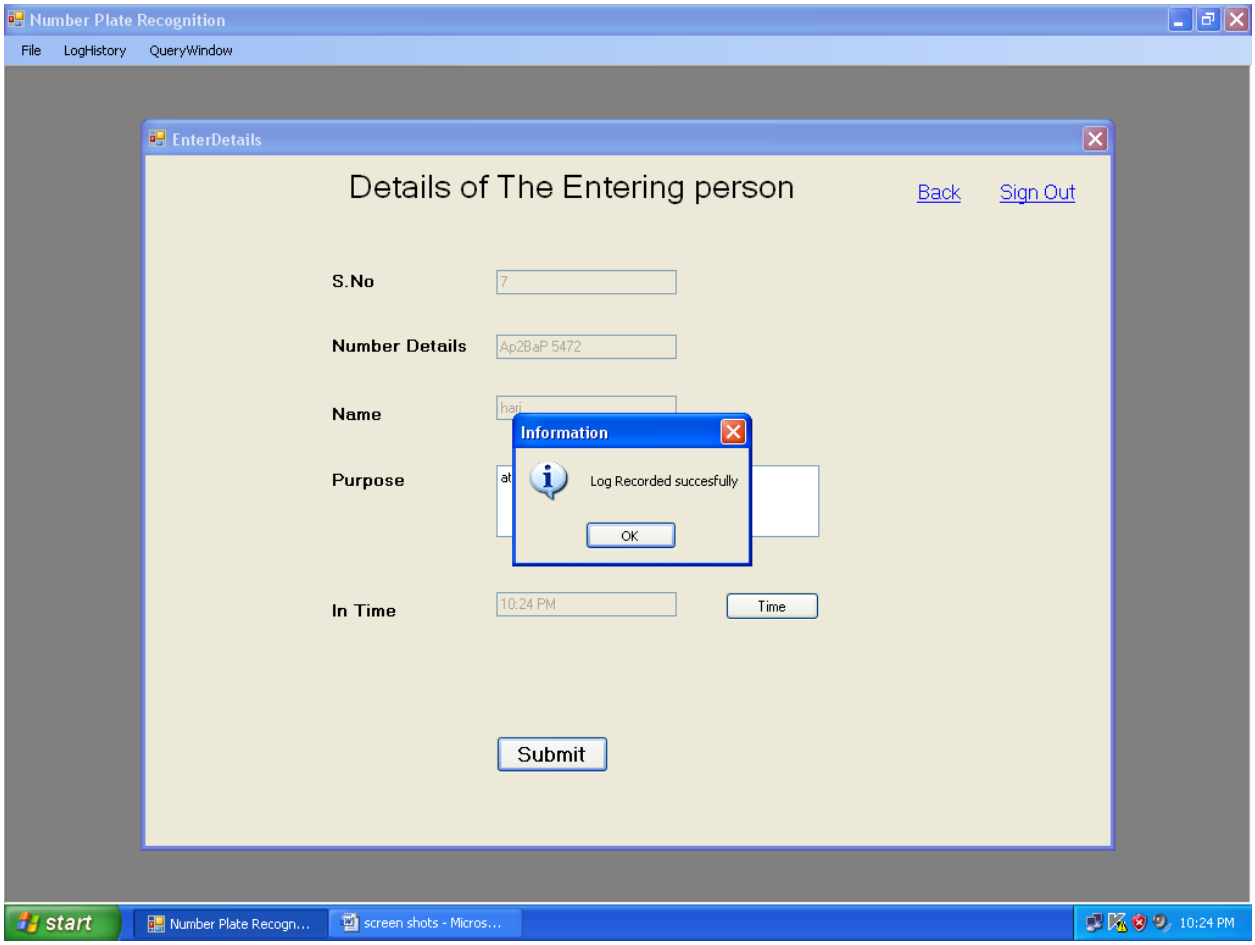

Number Plate Recognition
File
LogHistory
QueryWindow
EnterDetails
Details of The Entering person
Back
Sign Out
S.No
7
Number Details
Ap2BaP 5472
Name
Purpose
Information
Log Recorded succesfully
OK
In Time
10:24 PM
Time
Submit
start
Number Plate Recogn...
screen shots - Micros...
10:24 PM

Fig: Entering and storing the details

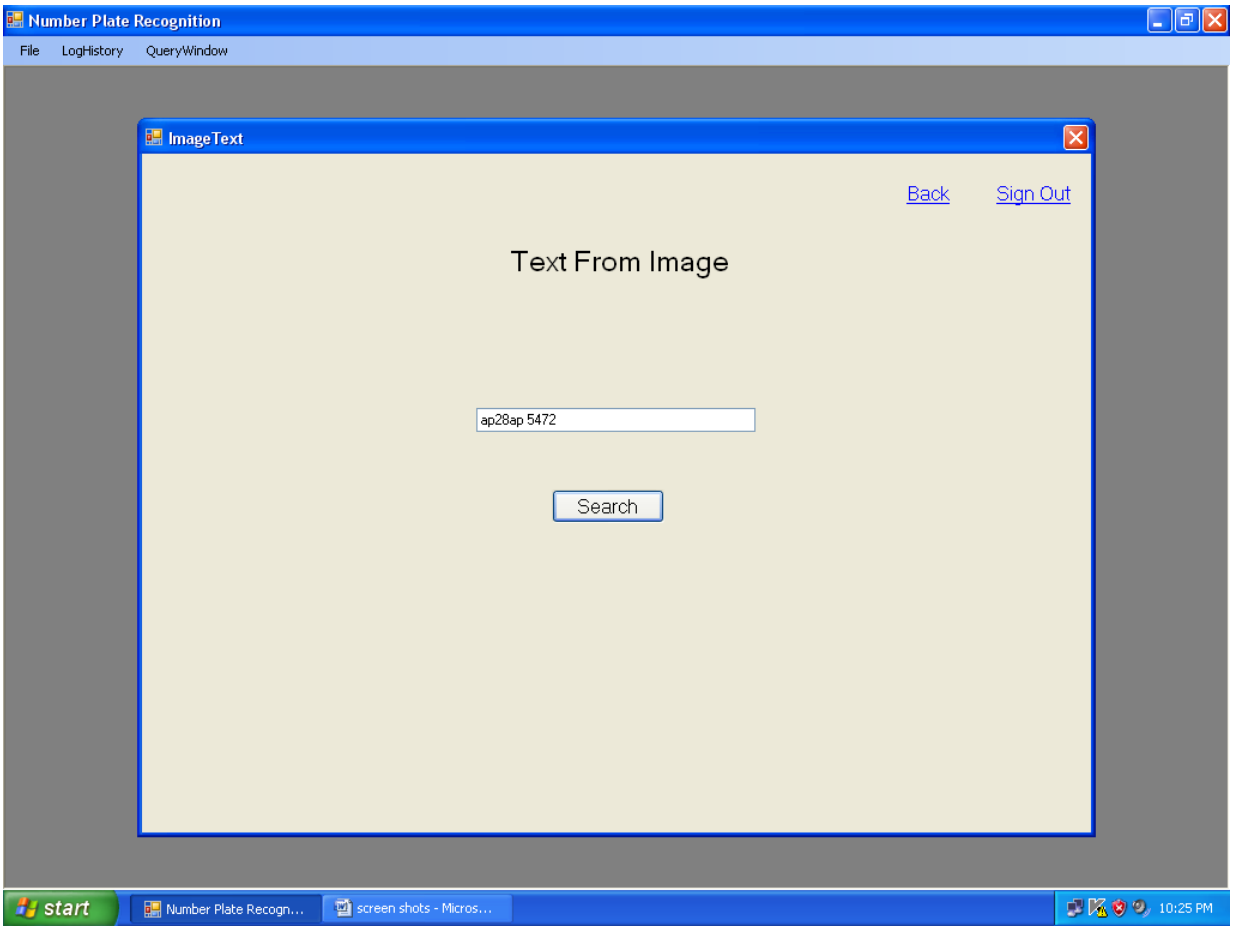


Fig: Manually searching the Number plate details

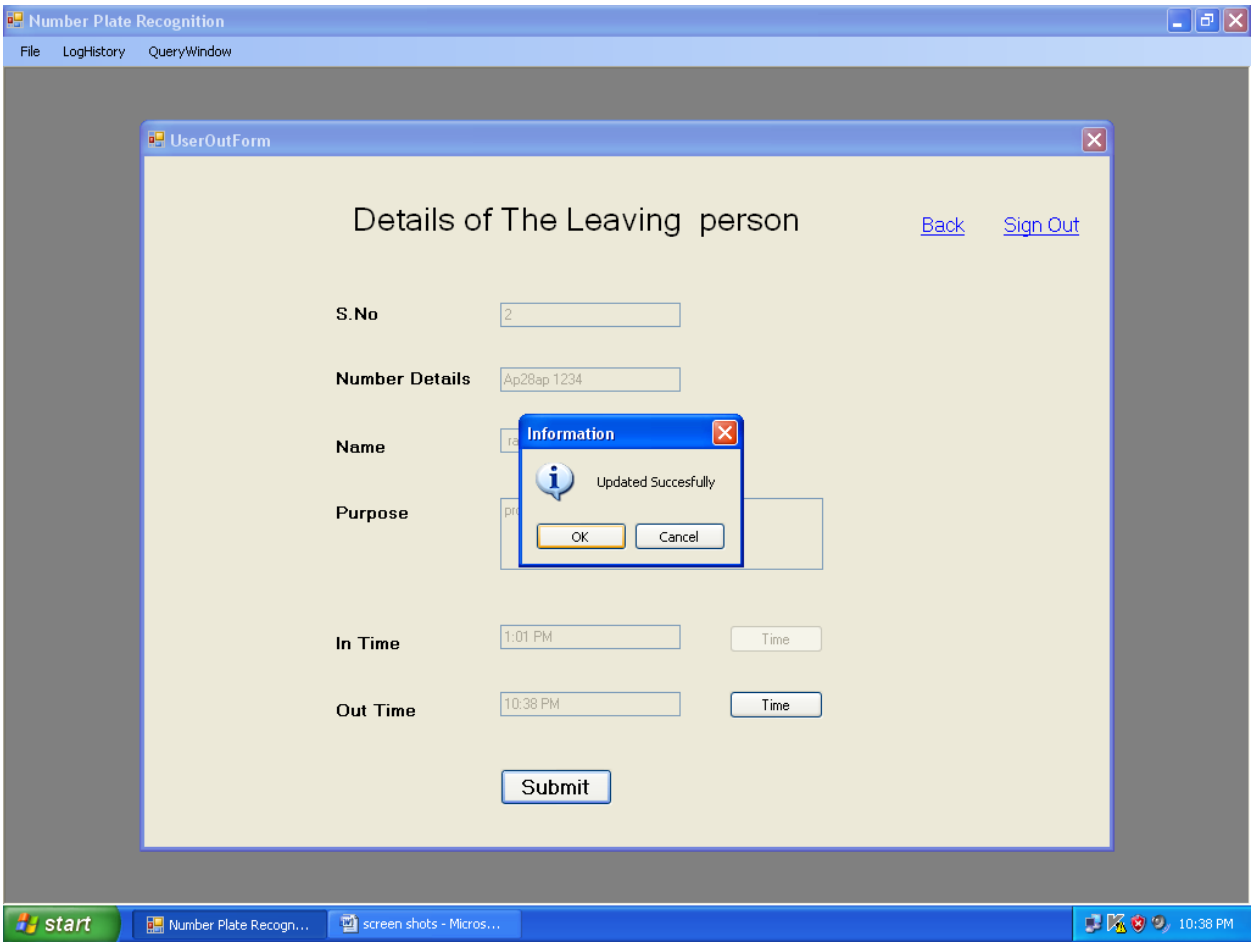

Number Plate Recognition
File
LogHistory
QueryWindow
UserOutForm
Details of The Leaving person
Back
Sign Out
S.No
2
Number Details
Ap28ap 1234
Name
Purpose
Information
Updated Succesfully
OK
Cancel
In Time
1:01 PM
Time
Out Time
10:38 PM
Time
Submit
start
Number Plate Recogn...
screen shots - Micros...
10:38 PM

Fig: Updating the out time details

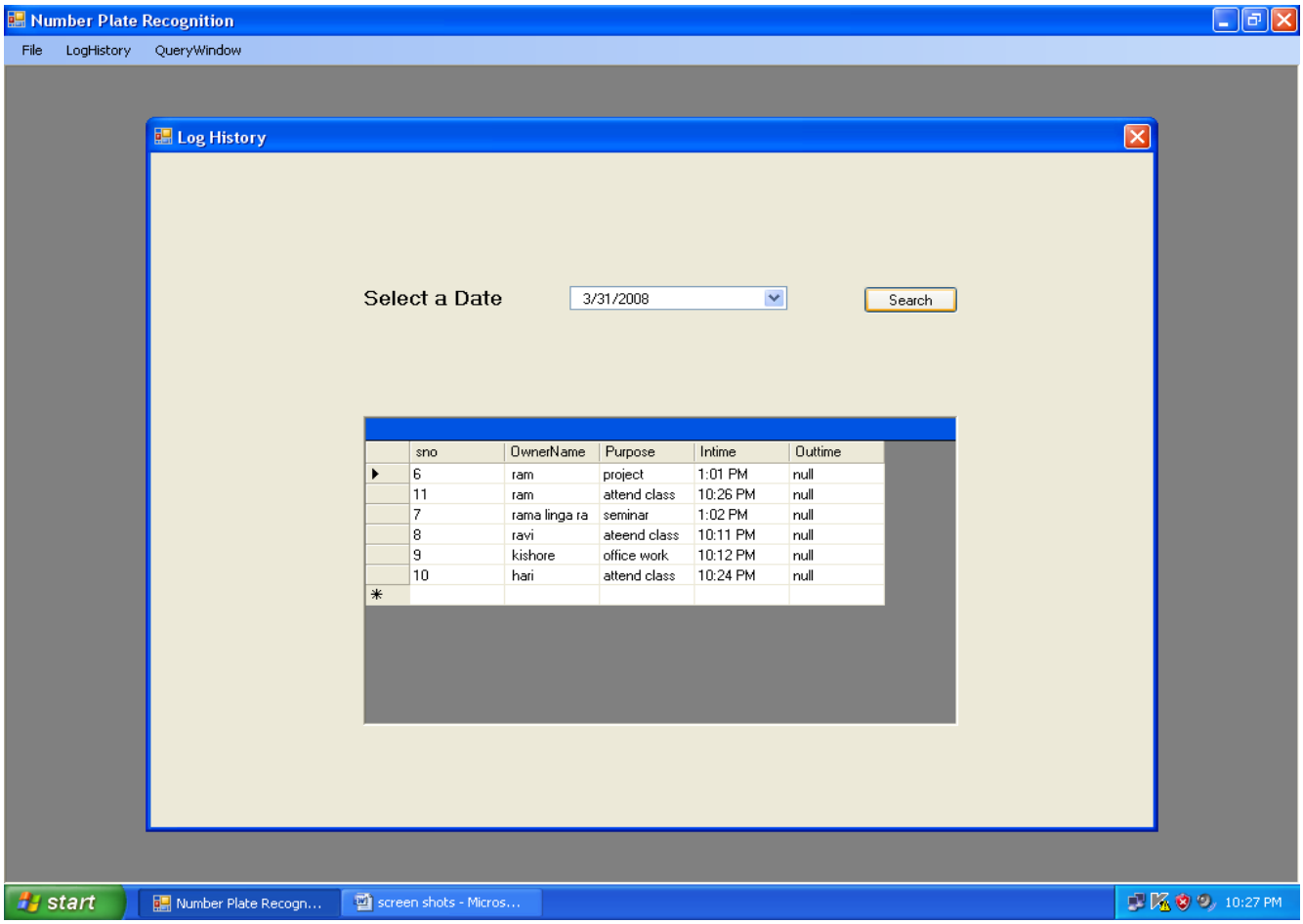


Fig: Viewing the log history

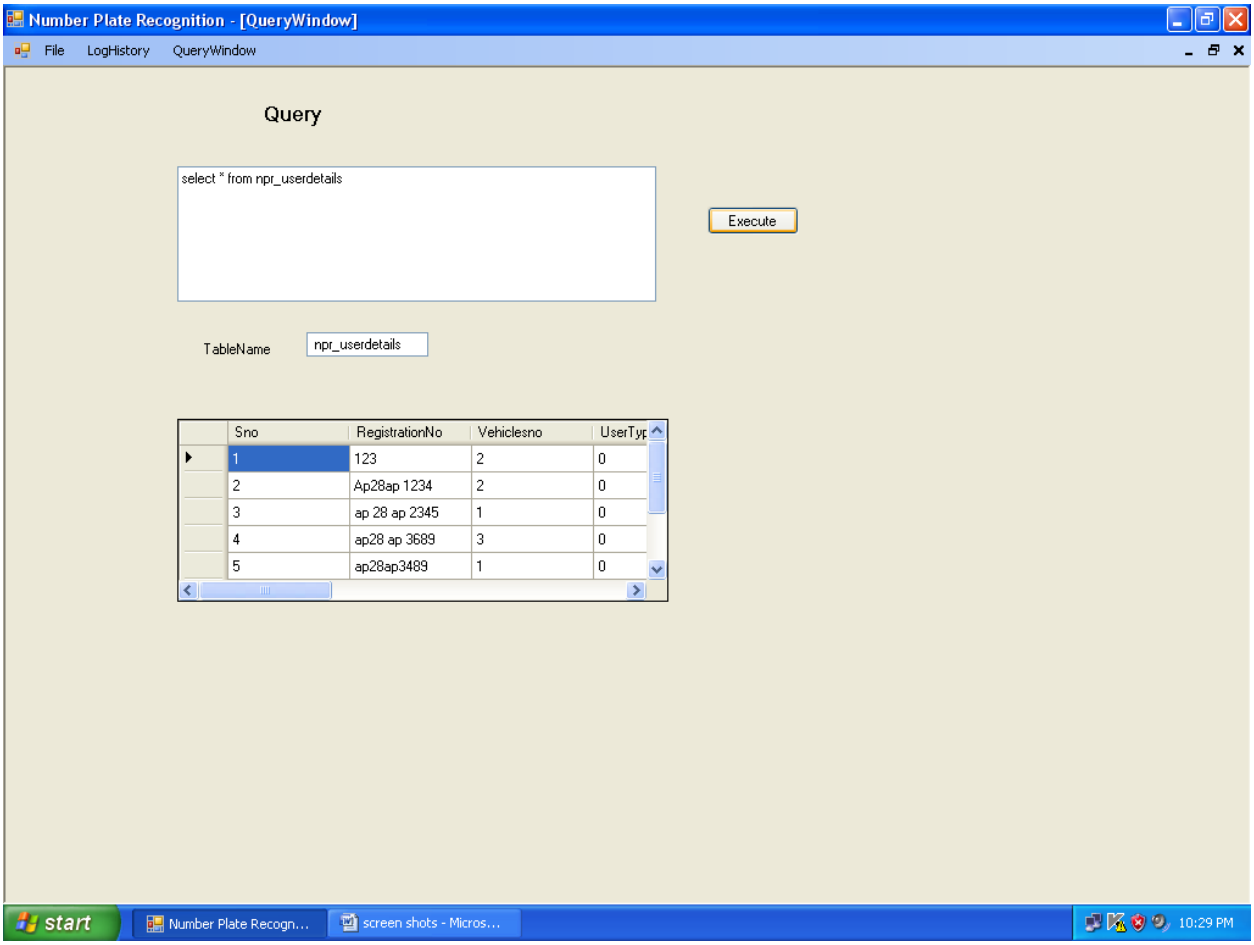


Fig: Viewing the stored data

8.Test Cases

TEST PLAN:

Software testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also be stated as the process of validating and verifying that a software program/application/product:

1. Meets the business and technical requirements that guided its design and development.
2. Works as expected.

3. Can be implemented with the same characteristics.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed.

Testing levels

Tests are frequently grouped by where they are added in the software development process, or by the level of specificity of the test.

- **Unit testing** refers to tests that verify the functionality of a specific section of code, usually at the function level. These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. Unit testing is also called *component testing*.

- **Integration testing** is any type of software testing that seeks to verify the interfaces between components against a software design. It works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

- **System testing** - Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.
- **System integration testing** verifies that a system is integrated to any external or third party systems defined in the system requirements.
- **Regression testing** focuses on finding defects after a major code change has occurred. Specifically, it seeks to uncover software regressions, or old bugs that have come back. Such regressions occur whenever software functionality that was previously working correctly stops working as intended.
- **Acceptance testing** can mean one of two things:
    1. A smoke test is used as an acceptance test prior to introducing a new build to the main testing process, i.e. before integration or regression.

2. Acceptance testing performed by the customer, often in their lab environment on their own hardware, is known as user acceptance testing (UAT). Acceptance testing may be performed as part of the hand-off process between any two phases of development.

- **Alpha testing** is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.
- **Beta testing** comes after alpha testing. Versions of the software, known as beta versions, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs.

Testing artifacts

Software testing process can produce several artifacts.

- **Test plan is** a document detailing a systematic approach to testing a system such as a machine or software. The plan typically contains a detailed understanding of what the eventual workflow will be. It documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements.

- **Traceability matrix** is a table that correlates requirements or design documents to test documents. It is used to change tests when the source documents are changed, or to verify that the test results are correct.

- **Test case** in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not. A test case is usually a single step, or occasionally a sequence of steps, to test the correct behavior/functionalities, features of an application. An expected result or expected outcome is usually given.

- **Test script** is the combination of a test case, test procedure, and test data. Test scripts can be manual, automated, or a combination of both.

- **Test suite is** the most common term for a collection of test cases. The test suite often also contains more detailed instructions or goals for each collection of test cases.
- **Test data** In most cases, multiple sets of values or data are used to test the same functionality of a particular feature. All the test values and changeable environmental components are collected in separate files and stored as test data.
- **Test harness** is the collection of software, tools, samples of data input and output, and configurations.

Test cases and Results

Test Strategy: Unit Testing **Testing Mechanism:** Manual

Date of Testing: 10th March, 2013

Module being tested: Administrator module

Test caseID	Test Description	Expected I/P	Expected O/P	Actual I/P	Actual O/P	Remarks
1	When Admin enters username and password it should be redirected to Admin form.	**User Name:** Admin **Password:** *****	Displays Admin form with admin functionalities	**a)** **User Name:** Admin **Password:** *****	Displays Admin form with admin functionalities.	Tested Successfully It displays Admin form with functionalities

Test strategy: Unit Testing **Testing Mechanism:** Manual

Date of testing: 12th march, 2013

Module being tested: User Module

Test caseID	Test Description	Expected I/P	Expected O/P	Actual I/P	Actual O/P	Remarks
2	When user enters userid and password, then user can perform operations like automatic process, manual process and search.	**User Name:** rohith **Password:** ******	Displays user home form with user functionalities	**a)** **User Name:** rohith **Password:** ******	Displays user home form with user functionalities	Tested Successfully It displays user home form with functionalities.
				b) **User Name:** rohith **Password:** ********	Displays user does not exist message.	Tested Successfully It displays error message until user enters correct userid and password

Test Strategy: Integration Testing **Testing Mechanism:** Manual

Date of Testing: 15th march, 2013

Modules being tested: User

Test caseID	Test Description	Expected I/P	Expected O/P	Actual I/P	Actual O/P	Remarks
1	After user selecting automation process, user has to take the image of number plate and process it.	User should capture the image which consist of number plate ex: AP36 V 4848	The output is text present in the image.	**a)** Image of the number plate consist of AP36 V 4848	Text is showed on the text box. i.e. AP36 V 4848	Tested Successfully
				b) If the capture image quality is less.	Displays incorrect output. i.e. AD86 4X4X	Tested Successfully .

Test Strategy: Integration Testing **Testing Mechanism:** Manual

Date of Testing: 21th march, 2013

Modules being tested: User module

Test caseID	Test Description	Expected I/P	Expected O/P	Actual I/P	Actual O/P	Remarks
2	User stores the details about the number plate after recognizing the characters.	User takes the number plate characters and searches the database.	User stores the details successfully.	**a)** User searches the database if found then enters the details.	User updates the details successfully.	Tested Successfully
				b) User searches the database if not found then registers the vehicle details.	User stores the newly registered vehicle details successfully.	Tested Successfully

Test Strategy: System Testing **Testing Mechanism:** Manual

Testing Approach: Bottom up Approach

Date of Testing: 25th March, 2013.

Test caseID	Test Description	Expected I/P	Expected O/P	Actual I/P	Actual O/P	Remarks
1	When user enters userid and password, then NPR system should provide access to home page after login validations	**User Name:** abhishek **Password:** ******	Displays user home form with user functionalities	**a)** **User Name:** abhishek **Password:** *******	Displays user home form with user functionalities	Tested Successfully It displays user home form with functionalities.
				b) **User Name:** Admin **Password:** *********	Displays error message i.e. user does not exist.	Tested Successfully It displays error message until user enters correct userid and password.

2	When User clicks on search button then Number plate recognition system should open the vehicle details form or user details form.	User should give the registration number.	Displays sno, owner name, purpose and intime and when user clicks on sno link then it will send to update the details of the leaving person, where outtime should be given.	**a)** Gives the registration number.	Displays sno, owner name, purpose and intime and when user clicks on sno link then it will send to update the details of the leaving person, where outtime should be given.	Tested Successfully
				b) User gives the registration number, if it is not present.	Displays No user found message.	Tested Successfully

3	When user clicks on Manual process then Number plate recognition system should open the image text form where a registration number should be searched.	User should enter the registration number	Displays the details of that registration number if present or else stores the details.	**a)** User should enter the registration number	Displays the details and asks us to update the purpose of visiting.	Tested Successfully
				b) User should enter the registration number if it is not stored.	Displays the vehicle registration page where all the details of the vehicle are stored	Tested Successfully
4	When user clicks on report then Number plate recognition system should open the form which shows the report for that day.	A click on report	Displays form which consists of Total no of vehicle entered and number of vehicles present and left	A click on report	Displays form which consists of Total no of vehicle entered and number of vehicles present and left.	Tested Successfully

5	When a user clicks on "signout" then Number plate recognition system will redirects user to login page.	A user clicking on "Signout"	Closing user's session and displaying login page	**a)** User clicks on "Signout"	Closes user's session and displays login page	Tested Successfully
				b) User directly closes the application	User session is closed directly when application is closed	Tested Successfully

9.Conclusion And Future Scope

Conclusion:

The project titled as " Number Plate Recognition" was deeply studied and analyzed to design the code and implement with various testing methods was done under the guidance of the experienced project guide.

The solution developed is free from all the bugs and executable with all different modules to the utmost satisfaction of the end user. All the current requirements and possibilities have been taken care during the project time.

The project is able to successfully incorporate all the requirements specified by the user. Proper care has been taken during database design to maintain data integrity and to avoid

redundancy. A client side validation has also been done with at most care by considering all the possibilities and requirements of different users to avoid data inconsistency.

The project is designed and coded in such a way that any further modification that are needed in future can be easily implemented without affecting the functionality of the system. The technical documentation provided in the project report helps the application developers to understand the internal architecture of the system and thus assist them in enhancing the system.

This project is purely user-friendly. It is very easy to implement or add many features to this tool. This plays a vital role in any big organization.

Future Scope:

The project that we are implementing is a real time project, In India we have not used such kind of technology for maintaining log, such a technology is implemented only in European countries. Since this software has not implemented before we have a great scope for implementing this project.

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

- As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
- Because it is based on object-oriented design, any further changes can be easily adaptable.
- Based on the future security issues, security can be improved.
- By using sensors we can make complete automation of recognition.
- By use high definition camera we can solve some image clarity problems.
- We can implement our project through mobile Devices or hand held devices

The scope of this project is, we can use in any organization for storing log history and retrieving valuable information from it.

References

- http//www. Microsoft .com
- http//www.planesource.com
- http//www.visualcoders.net
- http//www.C#.net

Books and Manuals:

- C#.NET by WROX Publication.
- C#.NET by Microsoft press.
- Black Book C# 2005 Programming by Matt telles.
- Software Engineering by Roger S. Pressmen.
- An Integrated Approach to Software Engineering by Pankaj Jalote.
- An Introduction to Database System by C.J. Date
- Database System Concepts by Henry F. Korth and Abraham Silberscatz
- System Analysis and Design by I.T.Hawryszkiewycz

Appendix

A:

ADO: Active Data Object

B:

BCL: Base Class Libraries

C:

CLR: Common Language Runtime
CTS: Common Type System
CMS: Content Management System

G: **GC:** Garbage Collector

I: **ITS:** Intelligent Transportation System

N: **NPR**: Number Plate Recognition.