Major Project

“KAYO - The Traffic Manager”

Submitted in partial fulfillment for the award of the degree of

Bachelor of Engineering in

Computer Engineering

By:

Amitoj Singh (2K1/COE/010)
Amritpal Singh Dhillon (2K1/COE/011)
Gaurav Sharma (2K1/COE/022)
Vaneet Chadha (2K1/COE/062)

Under the Skillful Guidance of
Dr. Goldie Gabrani

Department of Computer Engineering
Delhi College of Engineering
Delhi University
Delhi -110042
2001 – 2005

These days, even traffic lights are Keeping An eYe On you.

KAYO

The Traffic Manager

Table of Contents

Acknowledgement

We feel honored in expressing our profound sense of gratitude and indebtedness to Dr. Goldie Gabrani department of computer engineering, Delhi College of Engineering, Delhi for giving us the opportunity to work on such a practical problem, under her expert guidance. She constantly guided and helped us throughout the project. Words cannot express the support and motivation provided by her.

Our sincere thanks to Dr. D Roy Choudhury, HOD computer engineering department, Delhi College of Engineering, Delhi and other faculty for their cooperation and help during the project.

Amitoj Singh	2K1/COE/010
Amritpal Singh Dhillon	2K1/COE/011
Gaurav Sharma	2K1/COE/022
Vaneet Chadha	2K1/COE/062

Certificate

This is to certify that the project entitled "**KAYO - The Traffic Manager**" is a bonafide work of the following student of Delhi College of Engineering

Amitoj Singh	2K1/COE/010
Amritpal Singh Dhillon	2K1/COE/011
Gaurav Sharma	2K1/COE/022
Vaneet Chadha	2K1/COE/062

This project was completed under my direct supervision and guidance and forms a part of their bachelor of engineering (B.E.) course curriculum.

They have completed their work with utmost sincerity and diligence.

I wish them all the best for their future endeavors.

Dr. Goldie Gabrani
Department of Computer Engineering
Delhi College of Engineering
Delhi – 110042

1. Introduction

1.1 Project Origin

Drivers and passengers lose two billion hours each year to nerve-wracking traffic jams caused by an estimated annual increase of 20 percent to 30 percent in the number of vehicles clogging the highways. Congestion is one of the world's biggest transport problems. Over the last 5 years, congestion has increased by 56%. Every year, according to the General Accounting Office (GAO), it costs the nation approximately $100 billion in traffic delays, and an additional $70 billion per year is spent on traffic accidents.

With ever increasing pressure on transport infrastructure like highways due to increasing number of vehicles, the need to find the efficient and feasible solution is inevitable. The three most likely solutions are:

- To develop an efficient public transport system.
- Increasing the number of roads/highways proportionately to increase in vehicles.
- Or the third one proposed is KAYO which uses the already existing infrastructure by a better management of traffic using the technology developed. Aim of this project is to make world a better place to live in by relieving commuters of their ever growing woes.

1.2 Project Description

- Project explains a device named **KAYO (Keep An eYe On)** that will help in monitoring, regulating and at the same time help in decongesting of traffic.
- The idea is to monitor the traffic flow on the major routes, analyze it and give the useful information to the enquiring customer.
- The average speed on a given stretch of road is determined. The obtained data can be then passed on to a Windows CE based data server in a controlled manner from each speed detecting device.
- This data will then be processed and sent onto an Online Database using an application. Online Database will have lot of functionality in terms of average speed on a stretch, alternative paths and other add-ons.
- The user can access the data using an application on his PDA/Pocket PC.

1.3 KAYO Hardware/Software Specifications

The hardware part of the project consists of a camera that has following specifications:-

Programmable features

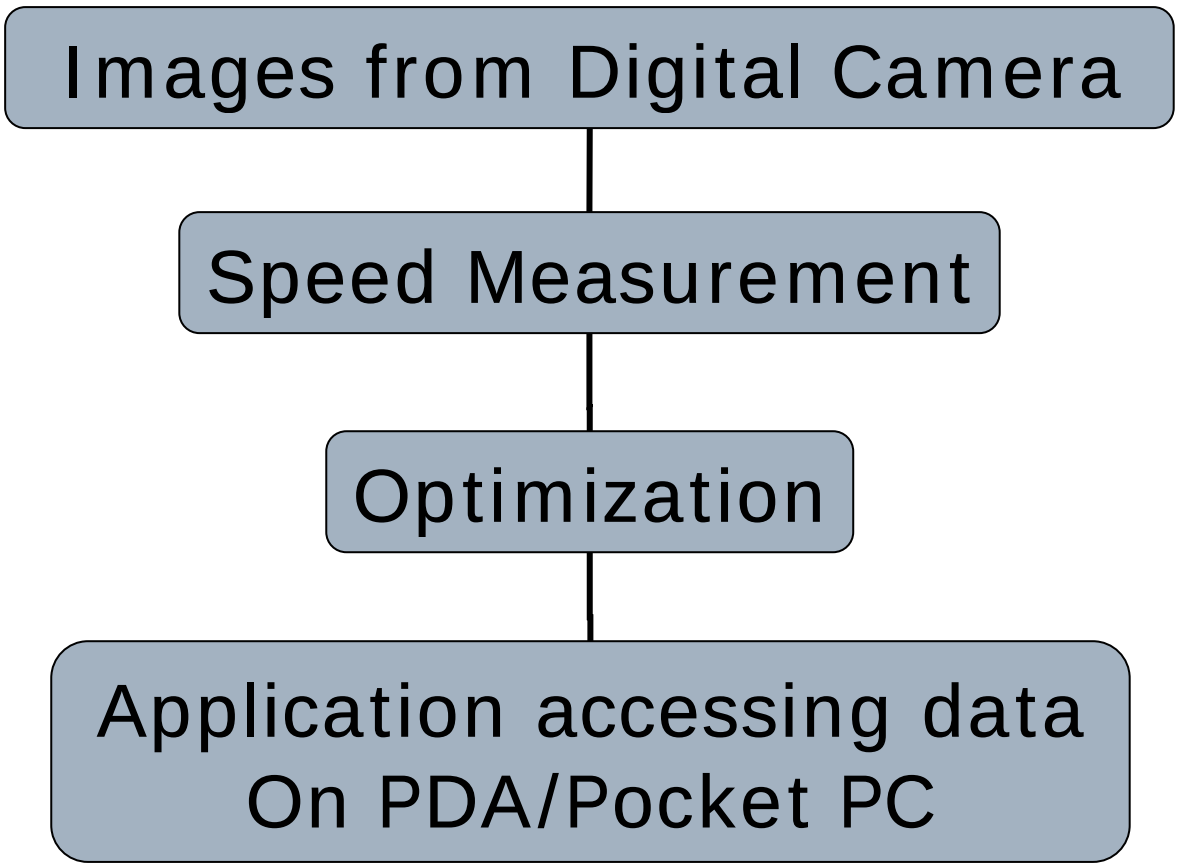
Frame rate of 50 ~ 0.5 frames per second (fps)
Exposure setting: 1 frame ~ 1/100 frame

Image size: 4.2 mm x 3.2 mm
Pixel elements - 384 x 288
Pixel dimension of 11 um x 11 um
Single 5-volt supply operation for analog and 5/3.3 volts for digital
Power consumption
Active: less than 100 mW at 50 fps
Standby: less than 100 mA
48-pin package

It helps in gathering the images from road at a fixed fps. The images are in the bmp formats that are being fed into the parallel port of the EBOX attached to each camera.

The software is meant to get the images from the parallel port of the EBOX and should calculate the speed from it using the procedures as shown below. Then this speed is converted to the time taken for the vehicle to travel that stretch of road. Now this time for each road stretch is known and it is gathered at a central server at which by using the modified all pair shortest path algorithm we find out the value of minimum time taken and the path followed to move from a given source to destination. After calculating this time and path for each node for the whole city the database is uploaded on the internet for the user to access the path that they must follow.

1.4 KAYO Flow Chart



2. Speed Measurement

There are many ways of speed measurement of vehicles on the road. Few of them which are feasible and implementable are:

2.1 Laser Technology

Laser technology works on the simple principle of striking the object whose speed has to be calculated with laser with a transmitter and catching the reflected wave with the receiver. The frequency change in the transmitted and the reflected wave can be used to calculate the speed of the moving object using the concept of Echo Effect.

The echo occurs because some of the sound waves reflect off of a surface and travel back. The length of time between the moment the sound wave leave the transmitter and the moment the receiver detects it that the echo is heard determines the distance between you and the far surface that creates the echo.

A laser speed gun measures the round-trip time for light to reach a car and reflect back. Light from a laser speed gun moves a lot faster than sound -- about 984,000,000 feet per second (300,000,000 meters) or roughly 1 foot (30 cm) per nanosecond. A laser speed gun shoots a very short burst of infrared laser light and then waits for it to reflect off the vehicle. The gun counts the number of nanoseconds it takes for the round trip, and by dividing by 2 it can calculate the distance to the car. If the gun takes 1,000 samples per second, it can compare the change in distance between samples and calculate the speed of the car. By taking several hundred samples over the course of a third of a second or so, the accuracy can be very high.

2.2 Radar Technology

The concept of measuring vehicle speed with radar is very simple. A basic speed gun is just a radio transmitter and receiver combined into one unit. The working principle of Radar uses simple effect discovered by Doppler called as Doppler shift.

A Radio transmitter is a device that oscillates an electrical current so the voltage goes up and down at a certain frequency. This electricity generates electromagnetic energy, and when the current is oscillated, the energy travels through the air as an electromagnetic wave. A transmitter also has an amplifier that increases the intensity of the electromagnetic energy and an antenna that broadcasts it into the air.

Radio receiver is just the reverse of the transmitter: It picks up electromagnetic waves with an antenna and converts them back into an electrical current. At its heart, this is all radio is -- the transmission of electromagnetic waves through space.

Radar is the use of radio waves to detect and monitor various objects. The simplest function of radar is to tell about how far away an object is. To do this, the radar device emits a concentrated radio wave and listens for any echo. If there is an object in the path of the radio wave, it will reflect some of the electromagnetic energy, and the radio wave will bounce back to the radar device. Radio waves move through the air at a constant speed (the speed of light), so the radar device can calculate how far away the object is based on how long it takes the radio signal to return.

Doppler shift occurs when sound is generated by, or reflected off of, a moving object. Doppler shift in the extreme creates a sonic boom which occurs in cases where the speed of the vehicle is comparable to the speed of wave used in detecting it.

Like sound waves, radio waves have a certain frequency, the number of oscillations per unit of time. When the radar gun and the car are both standing still, the echo will have the same wave frequency as the original signal. Each part of the signal is reflected when it reaches the car, mirroring the original signal exactly.

But when the car is moving, each part of the radio signal is reflected at a different point in space, which changes the wave pattern. When the car is moving away from the radar gun, the second segment of the signal has to travel a greater distance to reach the car than the first segment of the signal. This has the effect of "stretching out" the wave, or lowering its frequency. If the car is moving toward the radar gun, the second segment of the wave travels a shorter distance than the first segment before being reflected. As a result, the peaks and valleys of the wave get squeezed together and the frequency increases. Based on how much the frequency changes, a radar gun can calculate how quickly a car is moving toward it or away from it.

Features of Laser & Radar speed calculation system are:

- **Accurate and reliable.**
- **Impossible to intercept.**
- **Safe.**
- **Practical.**
- **Easy to use.**
- **Versatile.**
- **Light weight & compact construction.**
- **Ready for interface.**

Versatility: Thanks to its reduced dimensions, Laser speed calculation system is an extremely versatile instrument, suitable for installation in a patrol vehicle, or directly on the road surface by means of a tripod, or in a fixed cabin.

Reliability: Measurement reliability is guaranteed by a special automatic control system. In particular, Laser speed calculation system duplicates speed measurement

on each vehicle, thus eliminating readings which do not match exactly. Moreover, this technique makes it possible to ascribe, without doubt, the speed reading to the vehicle which has prompted the measurement, as the latter may take place only on one vehicle at a time.

Accurate: The control electronic allows only error free readings to be accepted as evidence.

Impossible to Intercept: There is no way of detecting or disturbing the laser beams.

Safe: There is no need to install cables or other devices on the road.

Practical: No need to install permanent instrumentation in patrol vehicles. No modifications adaptations or periodic calibrations of the measuring instruments are necessary.

Easy To Use: Once installed it requires no intervention.

Versatile: Suitable for in-vehicle or road surface installation or fixed site.

Light Weight and Compact Construction: The basic version is housed in one portable case only.

Ready for Interface: Connection to a computer allows to store the speed calculated that cane forwarded to the required device.

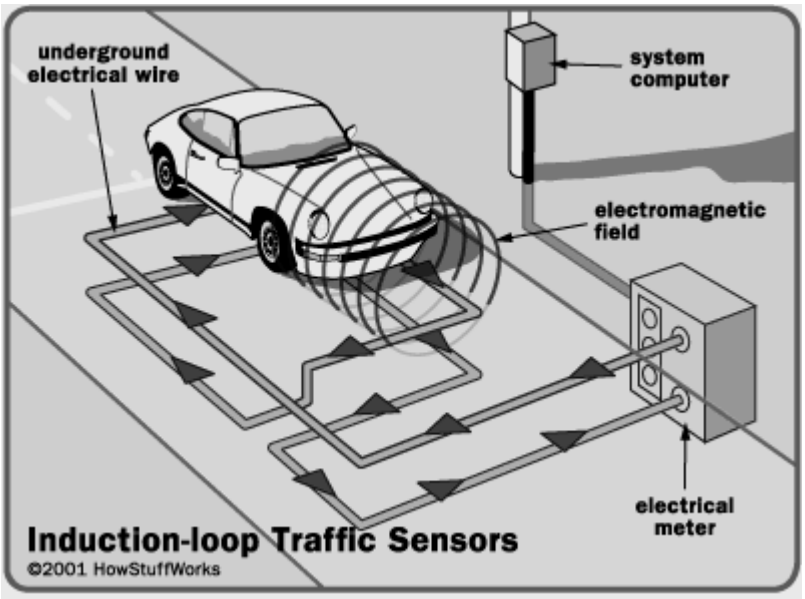
2.3 Trigger Technology

This technology uses sensors or ground loops that are cut into road surface. When a vehicle passes over them the speed is calculated, multiple inductive loops under the road monitor the vehicles movement over the target area (one target area for each lane and one camera for each lane). The Piezo strips can also be used by keeping them set apart from each other. The vehicle speed is calculated from the time taken to cross both strips.

Rubber strips in the road calculate the speed as vehicle passes over them via the small telltale grey post which sits along side. The system then requires a mobile camera usually in a van to plug into this grey post in order to photograph vehicle.

The main trigger technology used in red-light systems is the induction loop. An induction-loop trigger is a length of electrical wire buried just under the asphalt. Usually, the wire is laid out in a couple of rectangular loops resting on top of each other (see diagram below).

This wire is hooked up to an electrical power source and a meter. When electrical current passes through a wire, it generates a magnetic field. Positioning the wire in concentric loops, as in any electromagnet, amplifies this field. When a car drives over an induction loop, it disturbs the loop's electromagnetic field. This changes the total inductance of the loop circuit. This sort of field affects not only objects around the loop, but also the loop itself. The magnetic field induces an electrical voltage in the wire that is counter to the voltage of the circuit as a whole. This significantly alters the flow of current through the circuit.



The intensity of this induction depends on the structure and composition of the loop; changing the layout of the wires or using a different conductive material (metal) will change the loop's inductance. You can also change the inductance by introducing additional conductive materials into the loop's magnetic field. This is what happens when a car pulls up to the intersection. The huge mass of metal that makes up the car alters the magnetic field around the loop, changing its inductance. The meter in the system constantly monitors the total inductance level of the circuit. When the inductance changes significantly, the computer recognizes this shift and knows that a car has passed over the loop.

One emerging trigger mechanism is the video loop. In this system, a computer analyzes a video feed from the intersection. As the computer receives each new video frame, it checks for substantial changes at specific points in the image. The computer is programmed to recognize the particular changes that indicate a car moving through the intersection. If the light is red and the computer recognizes this sort of change, it activates the still cameras. The main advantage of this system is you don't have to dig up the road to install it, and you can adjust the trigger areas at any time. Essentially, it is a virtual inductive-loop trigger.

2.4 RFID

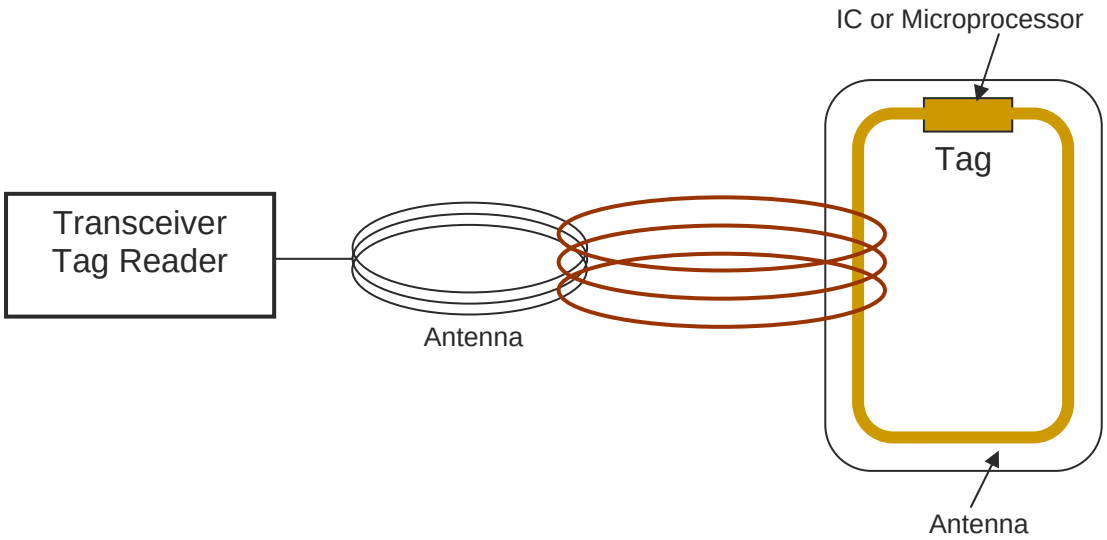
It uses Radio Frequency Identification (RFID) technology to identify various objects. Each object is embedded with and RFID tag which uniquely identifies it.

The implementation of the RFID technique involves three Components

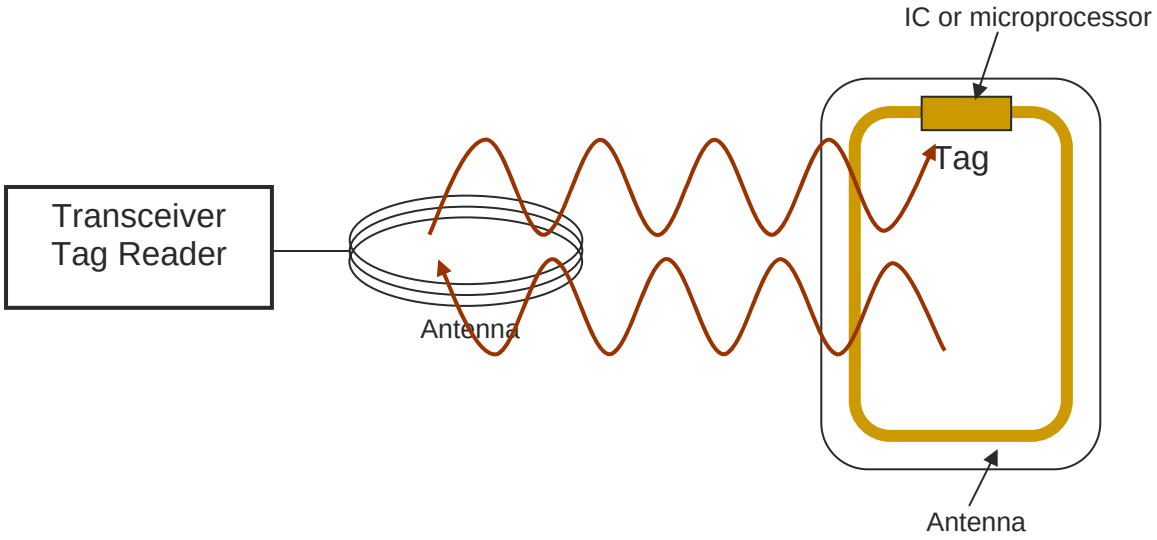
- Transceiver – Tag Reader
- Transponder – RFID tag
- Antenna

RFID Hardware

Magnetic Coupling



Inductive Coupling



RFID tags can be used in speed measurement by installing two transceivers at a fixed distance on the road. These two transceivers will identify the vehicle and thus calculate the time taken by the vehicle to traverse that distance and hence the speed of vehicle.

RFID's Disadvantages: -

- Cost.
- Lack of standards.
- Short range.
- Breaches the privacy of the vehicle owner.

2.5 KAYO

KAYO is the one of new technologies developed recently. The images of the vehicle are taken at predetermine frame rate and are processed to calculate the speed of the vehicle. The processing of the images can be done in two ways as discussed below: -

- Mapping screen-space coordinates into the road space coordinates: - in this method of processing the images to find out the speed, 2 mapping techniques are used to get the distance the vehicle has moved between two consecutive images. In the first mapping whole of the screen-space coordinates are mapped to the road space coordinates using a transformation matrix of order (3X3). In the second mapping certain areas of the road are marked by a distinguishable color. Now using these markings the position of the vehicle is determined on the road (hence the distance moved) and the speed is calculated.
- Linear Approximate Mapping (LAM): - This technique overcomes the shortcomings of the previous technique discussed. Like roads are to be marked again each time the markers fade out increasing the maintenance costs and complex calculation involved in determining the transformation matrix which requires high proficiency in Geometry and Trigonometry. LAM introduces a new concept of linear approximation. A region of interest is chosen and the region is so small that it can be approximated by linear mapping between the Screen space and the Road space. The choice of region is an important decision to be made. It can also be produced in matrix format such that matrix is reduced to a scalar matrix.

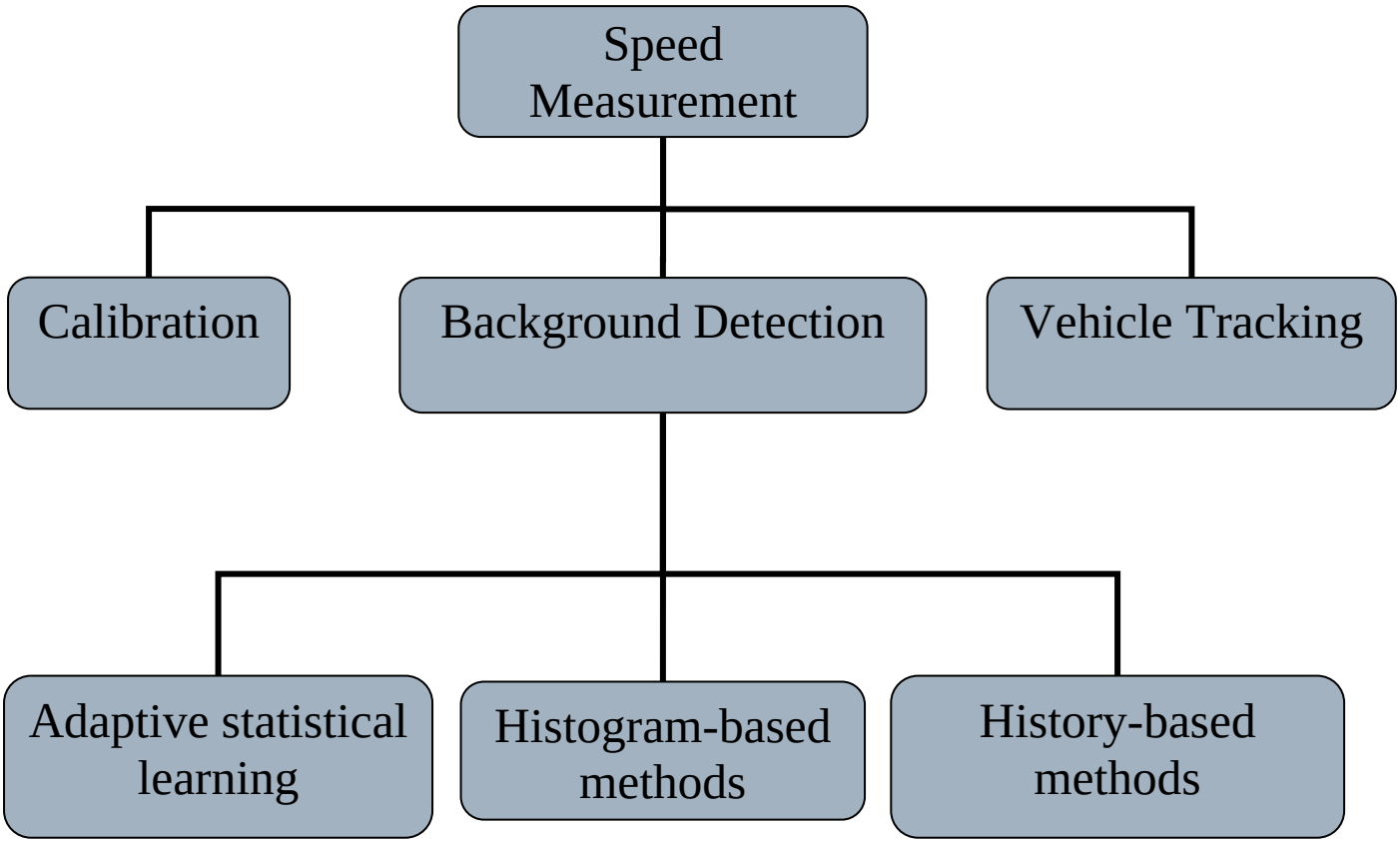
For the implementation of LAM the vehicle needs to be represented as a point object. For the conversion of vehicle into point object it is first converted to a rectangular object which is then converted to a point object using the following Bilinear interpolation.

$$\mathbf{F(Q) = [F(Q_1)/d'_5 + F(Q_2)/d'_6]/[\ 1/d'_5 + 1/d'_6] = [F(Q_1)d'_6 + F(Q_2)d'_5]/[\ d'_5 + d'_6]}$$
$$\mathbf{F(Q_1) = [F(P_1)d'_4 + F(P_4)d'_1]/[\ d'_1 + d'_4]}$$
$$\mathbf{F(Q_2) = [F(P_2)d'_3 + F(P_3)d'_2]/[\ d'_2 + d'_3]}$$

Here P_1, P_2, P_3, P_4 are the coordinates of the edges of the rectangular object.

3. Detailed Working of KAYO

For speed measurement the methods of calibration, background detection and vehicle tracking are used which are explained below to give holistic idea why KAYO uses these methods.



3.1 Calibration

The input to the tracking algorithm is an image acquired from the traffic camera. In order to use the image pixels to compute real world distances and speeds, camera must be calibrated.

The following assumptions are made on the camera and the road.

1. The road is flat (all points lie along a single plane).
2. The road is straight (traffic motion is parallel to some axis).
3. Occlusions are minimized (background images separate individual cars).

Now procedure for transformation from the image space into "road space" is sought. A 2-dimensional coordinate system that is aligned with the plane of the road is taken.

The x-axis of road space corresponds to the direction perpendicular to traffic flow, and the y-axis of road space is parallel to traffic flow. Once a car has been placed in road space, it is trivial to determine its velocity as the change Euclidean distance between points over time.

The transformation from screen space to road space is a projective transformation. The projective transformation is can be characterized by selecting 4 points in screen space, and providing their correspondences in road space. The transformation from screen space to road space is a projective transformation which can be performed by representing points in screen space as 2d homogenous coordinates (3 individual components), where (ivy) maps to (u', v', 1) in homogenous representation, and (u', v', w) maps to (u'/w v'/w) in non-homogenous representation. This allows the use of a 3x3 matrix to perform both translation and projection.

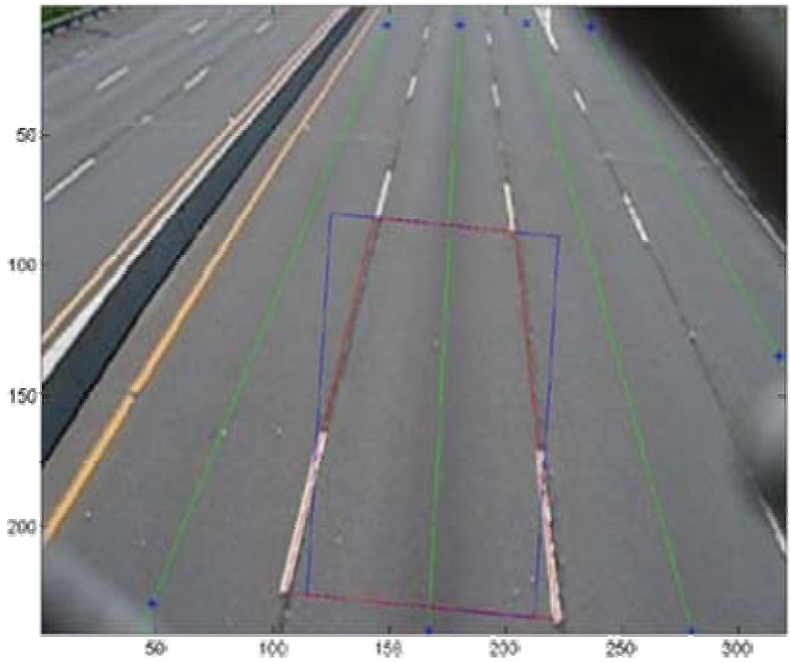
Let us refer to the four screen-space points selected by the user as a, b, c, d. When selected in counter-clockwise order around the rectangle with width w and height h, these correspond to the relative road space points (0, 0), (w, 0), (w, h), (0, h). Therefore, the problem is reduced to finding the best transformation matrix that maps the screen-space coordinates to their corresponding road-space coordinates.

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a}_x & \mathbf{b}_x & \mathbf{c}_x & \mathbf{d}_x \\ \mathbf{a}_y & \mathbf{b}_y & \mathbf{c}_y & \mathbf{d}_y \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & w & w & 0 \\ 0 & 0 & h & h \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

The affine transformation can be used as the initial quest for an unconstrained minimization, which is the best projective matrix **that transforms the screen-space points into the road space points**. This can be expressed using the following objective function: -

$$\sum_{i=1}^{4} \left(\frac{Ax_i + By_i + C}{Gx_i + Hy_i + 1} - u_i\right)^2 + \left(\frac{Dx_i + Ey_i + F}{Gx_i + Hy_i + 1} - v_i\right)^2$$

The results of the calibration can be visualized by transforming the points in road space back into screen-space using the inverse transformation, and plotting them to the screen. As can be seen in the figure, the affine transformation (blue quadrilateral) produces a reasonable approximation, but is limited to a parallelogram shaped region, as the parallelism of the rectangle is preserved by the affine transformation.



As can be seen in the figure, the affine transformation (blue quadrilateral) produces a reasonable approximation, but is limited to a parallelogram shaped region, as the parallelism of the rectangle is preserved by the affine transformation.

The projective transformation is able to accurately reproduce the points, as shown in the red quad. This maintains a rhomboid shape, as required by the projective transformation, which keeps two lines parallel (the front and back), but does not maintain the parallelism of the sides. This transformation will correctly account for the effects of distance.

Once the camera has been calibrated, it will be useful to identify the locations of the lanes in road space and screen space. The user is prompted to click on the center of each lane to be tracked. For each point (x_s, y_s), the system maps this to the road space points (x_r and y_r), and represents the lane in road space as the line $x=x_r$. In the figure, the line mapped back into screen space is shown and drawn over the background image

3.2 Background Detection

Cars in a frame can be located by looking for pixels that differ by some minimum amount from a standard background image. Therefore the first step was to find an efficient and accurate way to generate a background image from the given frames.

3.2.1 Histogram-based Background Detection

The techniques in this class involve maintaining a color histogram for each pixel over a given set of frames. Assuming a pixel will most frequently have its background RGB value, its color in the final background image corresponds to the most popular value in its histogram.

Two versions of the histogram-based background generator

1. The first maintains a separate histogram for each of the color channels for each pixel, and carries out median filtering on the final background image.
2. The second associates with each pixel a single histogram of all possible RGB combinations; also, a fraction of histogram values from adjacent pixels is added to the current pixel's histogram values.

The former method takes less memory - O (n) instead of O (n^3) - but latter is expected to be more accurate.

3.2.2 Adaptive statistical learning Based Background Detection

1. The algorithm stores the mean m and standard deviation s for the hue, saturation and intensity distributions of each pixel, over all frames up to the current one.
2. When a new frame is read, each pixel is marked as occluding or background depending on whether its value falls within two standard deviations of the current means.
3. The background model for the pixel is then updated according to the following equations:

   $m = (1 - a)\, m + ax$

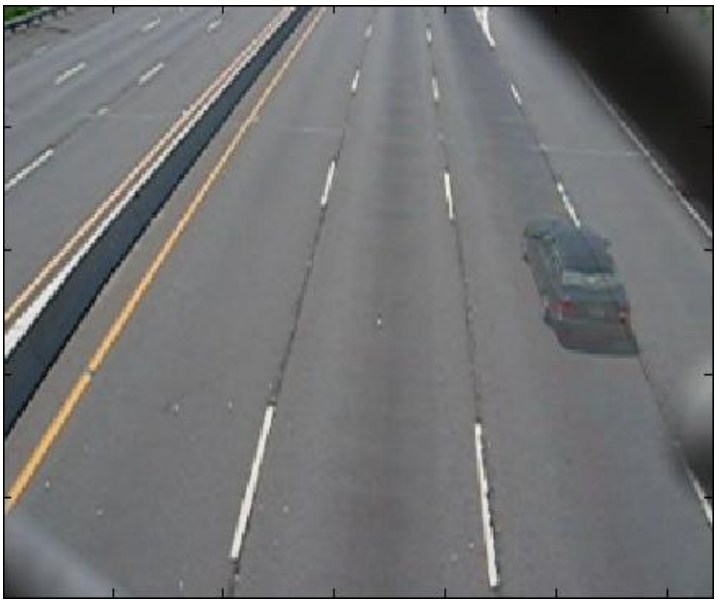
   $s_2 = \max (s_{min}2, (1 - a)\, s_2 + a(x - m)\, 2)$

   s_{min} refers to a minimum standard deviation

   x is the observed value of the pixel,

   and a is the learning rate of the model.

This algorithm as described runs much faster than the histogram-based methods. With a low learning rate of 0.02, it takes many frames for an accurate background image to be generated. The image below is the result after 52 frames.



With a high learning rate of 0.2, the background is generated in fewer frames, but some noisy pixels are added as seen below. Also, the resulting background image is unstable, as it rapidly adapts to incorporate newly seen cars into the background before it erases them.

3.2.3 History-based Background Detection

This Algorithm maintains a mean for each pixel. However, unlike the previous method it maintains the mean over a fixed-size history instead of over all previously seen frames. This method converges to an accurate background image more rapidly than the low learning rate RGB statistical learning method described above. At the same time, it is much more stable than the RGB statistical learning method with a high learning rate.

3.3 Vehicle Tracking

The tracking system used will monitor the fastest lane on every stretch of highway under consideration. A couple of things are assumed about the input footage: -

- First, the camera should be fixed in location. A camera strictly dedicated to analyzing traffic of a road would probably be fixed on a pole or overpass in one direction, making this a reasonable requirement.
- Second, an overhead view of the traffic is required.

With an overhead view, a system can simply analyze one area of the frame and know that this area belongs to only one lane of traffic. With side views, cars may be occluded by other cars or foreign objects.

With these two restrictions on input, it becomes feasible to analyze the traffic with minimal calculation. In short, the background from an input frame is deducted and a mask is created based on some error threshold. Then this mask is analyzed to track moving objects along the road. The key observation is made that almost all of the movement along the road is done in the same direction. With this in mind, the two-dimensional motion problem can be reduced to the one-dimensional problem of tracking a vehicle within its lane.

The implementation of the above system is straightforward. For each region of interest, its coordinates are stored in pixel space, the number of cars it has seen, the average speed of the lane, and an array of car structures with an entry for each car currently in the lane. In an initialization step, the background image of the line is extracted by pulling corresponding entries out of the input background image.

The pixels on the line of interest for the lane are extracted and background information is deducted. Then the sum of squared differences is compared with a threshold, and a mask is generated based on this comparison.

Once the background is deducted and mask is generated a line of 0’s and 1’s is left, where 0’s indicate background and 1’s indicate the presence of a car.

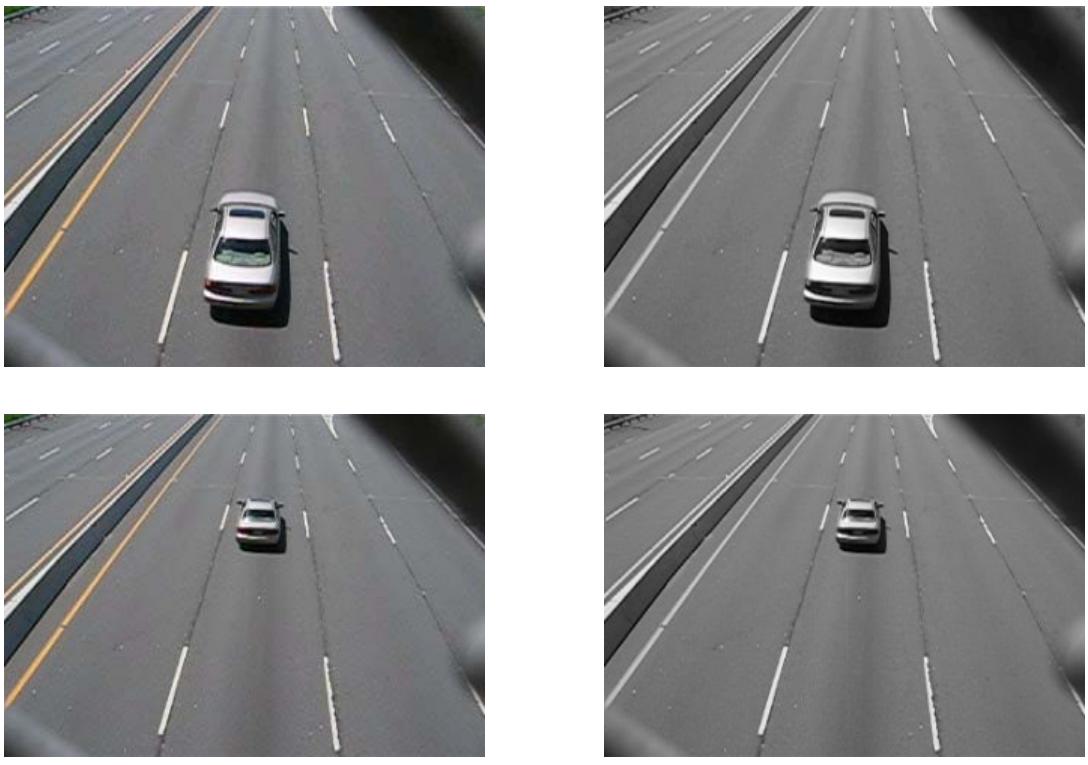
Using the calibration data, these pixels are mapped to a virtual road space and the distance in feet is determined. Simple calculation leads to a speed for a car in miles per hour, and in the aggregate case, an average speed for the lane.

The images below show the tracking process for 3 frames. White pixels indicate the presence of a car, and black pixels indicate the background.

3.4 Conversion of Original images to Gray Scale images.

This is of prime importance in speed measurement as the original images are colored. Even though the images here been taken from the same camera and when the camera is a t same position, still due to noise and other conditions the two images taken at a gap of 1 second may also never be same. So the first requirement is to convert the images into Gray Scale format so as to reduce the differences in the images that could have arisen due to color difference being there. Once the images are there in the Gray Scale format two options are available. Either the Gray Scale images can be converted to quantized images and then filters applied onto it to obtain final images that can be compared with background image, which has also passed through the same procedure. Or what can be done is to take the same Gray Scale image and pass it to the image comparison procedure

Conversion of colored background image to gray scale background

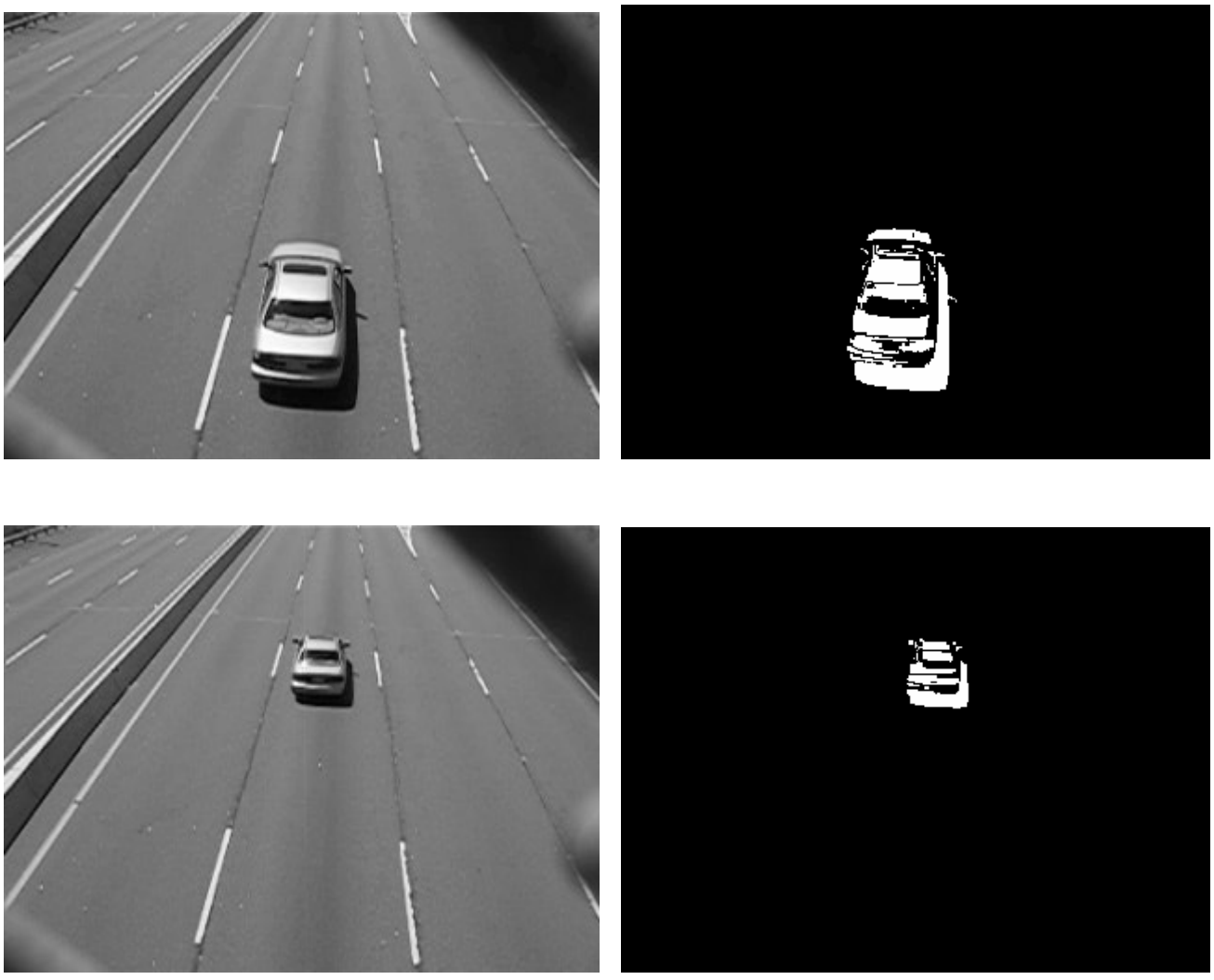


Conversion of various RGB images into corresponding grayscale images.

3.5 Identification of moving vehicle

In this method image comparison of the background image is done with current frame pixel by pixel. In each pixel the individual values of RGB (Red, Green, and Blue) are compared, if they lie in a specific range (i.e. 40 in this case) the two pixels are declared equal otherwise they are different. A new image is created that has each pixel as either black or white depending upon whether two pixels are same or not respectively in the parent images. Hence only the image of a vehicle remains in the resultant image.

The results of these figures are shown below:-



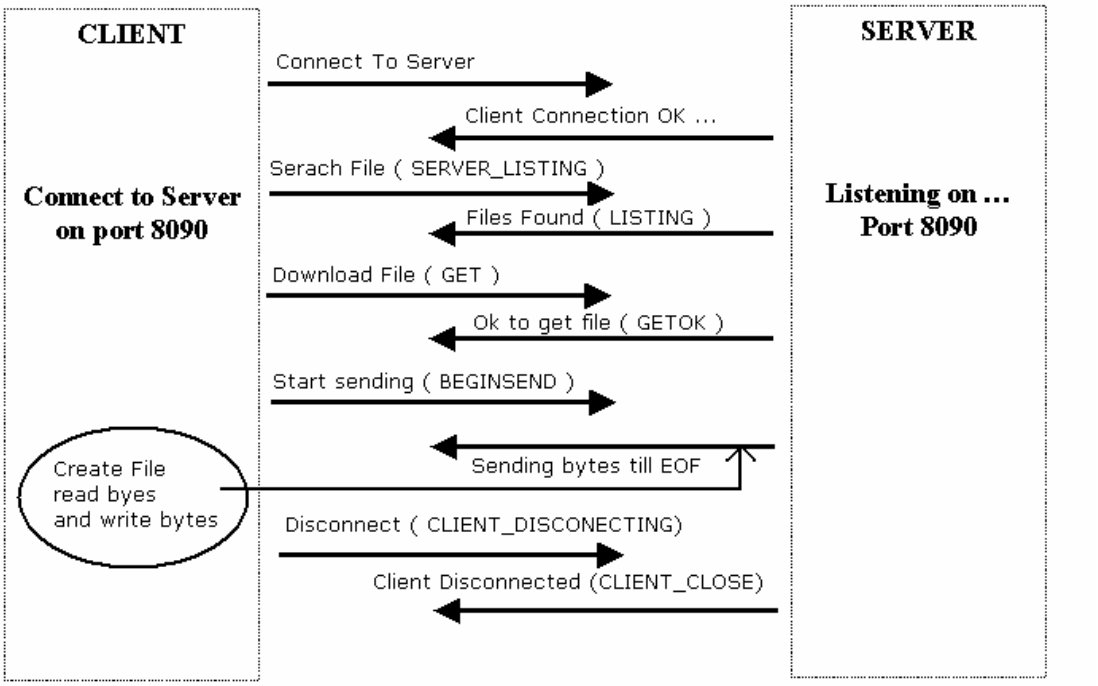
Isolation of moving vehicle for coordinates calculation.

4. Information Flow and Processing

- Gathering data from all the WIN-CE Nodes.
- Processing data to form a matrix representing the graphical view of the city.

4.1 Gathering data from all the WIN-CE Nodes

Each node i.e. the KAYO equipment at each installed position acts as a client. Client maintains a file having details of average time offset on stretch connecting the source and destination point monitored at that point. Client also updates the file at a fixed time interval t_c And the most important advantage of this client-server setup is that every client shares its file with a server. Thus server can communicate with all clients (nodes) and gather the time offset between all pairs of nodes by downloading the respective files.



Client-server setup: file sharing technique

File sharing technique implies the capability of a client to search and download a file shared by the server and a server to search and download a file shared by the client

4.2 Processing data to obtain graphical view of the city

The file on the server consists of N X N matrix having time offset between each pair of nodes gathered from all the clients. This file is updated at fixed interval of time t_s where $t_s = k \times t_c$; k is a natural number; k>1;

Thus graphical view of the city is generated in form of this matrix called as M. In this matrix M the nodes between which there is no direct connection are having the time

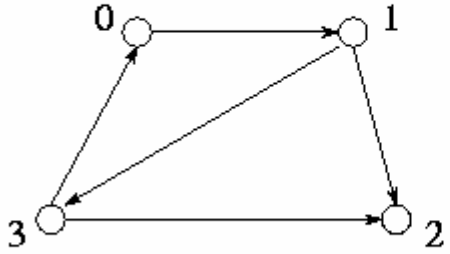
offset value as a fixed large value which is greater than any of the average time offset possible.

4.3 Finding the Route

- M is modified using "Modified Shortest path algorithm" to obtain M_N . The Time Complexity of the algorithm is O (N^3).
- M_N gives not only the shortest time offset possible between any nodes but also helps in giving the path to be followed as the output using another value stored with each member of matrix. This new value signifies the next node to be taken from the source node towards the destination node.
- Once the route is determined it is ready to be displayed along with the expected time to be taken on that route in response to the user query.

4.3.1 All Pairs shortest path

The all-pairs shortest-path problem involves finding the shortest path between all pairs of vertices in a graph. A graph **G = (V, E)** comprises a set **V** of **N** vertices, $\{V_i\}$, and a set $\mathbf{E} \subseteq \mathbf{V X V}$ of edges connecting vertices in **V**. In a directed graph, each edge also has a direction, so edges ($\mathbf{V_i}$, $\mathbf{V_j}$) and ($\mathbf{V_j}$, $\mathbf{V_i}$), $\mathbf{i \neq j}$, are distinct. A graph can be represented as an adjacency matrix A in which each element (**i**, **j**) represents the edge between element i and j. $\mathbf{A_{ij}}$=1 if there is an edge ($\mathbf{V_i}$, $\mathbf{V_j}$); otherwise, $\mathbf{A_{ij}=0}$. A path from vertex $\mathbf{V_i}$ to vertex $\mathbf{V_j}$ is a sequence of edges $\mathbf{(V_i, V_k), (V_k, V_l)\ldots (V_m, Vj)}$ from **E** in which no vertex appears more than once. The shortest path between two vertices $\mathbf{V_i}$ and $\mathbf{V_j}$ in a graph is the path that has the least edge cost. The all pairs shortest path problem requires finding the shortest path between all pairs of vertices in a graph.



G

	0	1	2	3
0	0	1	0	0
1	0	0	1	1
2	0	0	0	0
3	1	0	1	0

A

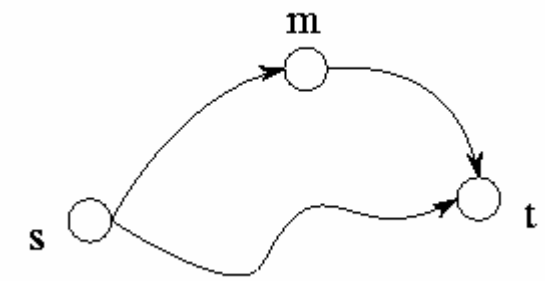
*A simple directed graph, **G**, and its adjacency matrix, **A***

To solve the above problem there are two sequential shortest-path algorithms due to Dijkstra and Floyd. Both algorithms take as input an **N X N** adjacency matrix **A** and compute an **N X N** matrix **S**, with $\mathbf{S_{ij}}$ the length of the shortest path from $\mathbf{V_i}$ to $\mathbf{V_j}$, or a distinguished value (∞) if there is no path.

4.3.2 Dijkstra's Algorithm

Dijkstra's single-source shortest-path algorithm computes all shortest paths from a single vertex, $\mathbf{V_s}$. It can also be used for the all-pairs shortest-path problem, by the simple expedient of applying it **N** times---once to each vertex $\mathbf{V_0, V_1 \ldots, V_{n-1}}$.

Dijkstra's sequential single-source algorithm maintains as **T** the set of vertices for which shortest paths have not been found, and as d_i the shortest known path from $\mathbf{V_s}$ to vertex $\mathbf{V_t}$. Initially, **T=V** and all $\mathbf{d_i}$. At each step of the algorithm, the vertex $\mathbf{V_m}$ in **T** with the smallest **d** value is removed from **T**. Each neighbor of $\mathbf{V_m}$ in **T** is examined to see whether a path through $\mathbf{V_m}$ would be shorter than the currently best-known path as shown in the figure.



The comparison operation performed in Dijkstra's single-source shortest-path algorithm. The best-known path from the source vertex $\mathbf{V_s}$ to vertex $\mathbf{V_t}$ is compared with the path that leads from $\mathbf{V_s}$ to $\mathbf{V_m}$ and then to $\mathbf{V_t}$.

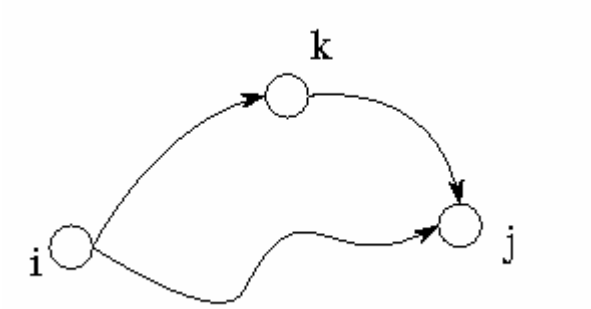
An all-pairs algorithm executes Algorithm **N** times, once for each vertex. This involves $\mathbf{O(N^3)}$ comparisons and takes time $\mathbf{N^3 t_c F}$, where t_c is the cost of a single comparison in Floyd's algorithm and **F** is a constant. Empirical studies show that $\boldsymbol{F \cong 1.6}$, i.e., Dijkstra's algorithm is slightly more expensive than Floyd's algorithm (discussed below).

The algorithm is listed as shown below

```
dist Dijkstra( graph G, vertex V0 )
// V0 is the source vertex of graph G
{
  S = {v0};
  for (i=1; i< n; i++)
    D[i] = C[v0,i];        // C[i,j]: cost matrix
  for (i=0; i< (n-1); i++)
  {
   choose a vertex w in V-S such that
    D[w] is minimum;
   add w to S;
   for each vertex v in V-S
     D[v] = min(D[v], D[w]+C[w,v]);   // update distant array
  }
}
```

4.3.3 Floyd's Algorithm

Floyd's Algorithm derives the matrix **S** in **N** steps, constructing at each step **k** an intermediate matrix **I (k)** containing the best-known shortest distance between each pair of nodes. Initially, each $\mathbf{I_{ij}\ (0)}$ is set to the length of the edge ($\mathbf{V_i}$, $\mathbf{V_j}$), if the edge exists, and to ∞ otherwise. The $\mathbf{k^{th}}$ step of the algorithm considers each $\mathbf{I_{ij}}$ in turn and determines whether the best-known path from $\mathbf{V_i}$ to $\mathbf{V_j}$ is longer than the combined lengths of the best-known paths from $\mathbf{V_i}$ to $\mathbf{V_k}$ and from $\mathbf{V_k}$ to $\mathbf{V_j}$. If so, the entry $\mathbf{I_{ij}}$ is updated to reflect the shorter path as shown in the figure. This comparison operation is performed a total of $\mathbf{N^3}$ times. Hence, the sequential cost of this algorithm can be approximated as $\mathbf{t_c N^3}$, where $\mathbf{t_c}$ is the cost of a single comparison operation. As already shown Floyd's algorithm is better than Dijkstra's algorithm by a factor of **F**.



The fundamental operation in Floyd's sequential shortest-path algorithm: Determine whether a path going from $\mathbf{V_i}$ to $\mathbf{V_j}$ via $\mathbf{V_k}$ is shorter than the best-known path from $\mathbf{V_i}$ to $\mathbf{V_j}$.

The algorithm is listed as shown below:

```
shortest( double A[n][n], double C[n][n], int P[n][n] )
// shortest taks an nxn matrix C of edge costs and produces
// an nxn matrix A of lengths of shortest paths, and an nxn
// matrix P giving a point in the middle of each shortest path
{
    int i,j,k;
    for (i=0; i<n; i++)
      for (j=0; j<n; j++)
      {
        A[i][j] = C[i][j];
        P[i][j] = -1;
      }
    for (i=0; i<n; i++)
      A[i][i] = 0;              // no self cycle
    for (k=0; k<n; k++)
      for (i=0; i<n; i++)
        for (j=0; j<n; j++)
          if (A[i][k]+A[k][j] < A[i][j])
          {
            A[i][j] = A[i][k] + A[k][j];
            P[i][j] = k;        // k is included in the shortest path
```

```
        }
   }

path ( int i, int j )
// path prints out the actual path, using P array
{
     int k;
       k = P[i][j];
     if (k != -1)
     {
       path(i,k);
       printf("%d ",k);
       path(k,j);
     }
}
```

4.3.4 Algorithm used in KAYO

The algorithm used in KAYO to find the route is a modified version of Floyd's Algorithm. The modification done here is that the route is also stored along with the cost of the path. The path is to be calculated for each node separately using a small code that is executed only when the user access the website to get the information. The return type of the function that finds the all pair shortest path using modified Floyd's algorithm is an array of a structure that consists two integer values, first (Cost) is used for storing the cost of the edge for going from node A to node B and the other (Next) holds the value of next node that is to be encountered while traveling. Hence, the total path can be evaluated by using the second integer value in the following manner.

1. Add the source into the path.
2. Compare the value of node.Next with the destination.
3. If not same then search the path from node.Next to destination using the same algorithm.
4. If same then just add destination to the path.

This small algorithm is recursive in nature and easily implementable. The path contains the name of the nodes that are to be traversed for reaching the destination. The path thus achieved is the shortest path between the source and destination. The code can be easily modified to achieve better efficiency for path calculation.

5. Delivering Information to the User

Till this point raw information is available. To present this information to the user it is converted to a standard format of XML and then the XML file is uploaded to web server. User can now access this information using PDA/Pocket PC.

5.1 Transformation from File format to XML

File consisting information about cost and route between the all pair of nodes, in other words file storing M_N is converted to XML format so that application on PDA or Pocket PC gets interface to provide user with useful information .

5.1.1 Why XML?

XML applications provide many advantages.

- XML's strongest point is its ability to do data interchange. Because different users use rarely standardize on a single set of tools and applications for querying, it takes a significant amount of work for two groups to communicate. XML makes it easy to send structured data across the web so that nothing gets lost in translation.
- When using XML, XML-tagged data can be received from any server, and server can receive XML tagged data from any client. Neither of two knows how the other's system is organized. If another partner or supplier teams up with existing service provider, no code has to be written for exchanging the data within their systems.
- One major problem with today's web is that search engines can't process HTML intelligently. But XML has DTD for name and address records, making searching more specific, accurate, useful and efficient.

5.1.2 Overview of XML

XML stands for **eXtensible Markup Language**, and it is a standard for structured text documents developed by the World Wide Web Consortium. XML can be used to structure text in such a way that it is readable by both humans and machines, and it presents a simple format for the exchange of information across the internet between computers. As such, electronic commerce is the principal application area for XML.

XML document: -An XML document contains a prolog and a body. The prolog consists of an XML declaration, possibly followed by a document type declaration. The body is made up of a single root element, possibly with some comments and/or processing instructions. An XML document is typically a computer file whose contents meet the requirements laid out in the XML specification.

XML Declaration: -The first few characters of an XML document must make up an XML declaration. The declaration is used by the processing software to work out how

to deal with the subsequent XML content. A typical XML declaration is shown below. The encoding of a document is particularly important, as XML processors will default to UTF-8 when reading an 8-bit-per-character document. This will cause characters to be rendered incorrectly if the document uses Latin encoding (iso-8859-1). XML processing applications are required to handle 16-bit-per-character documents in the Unicode encoding, which makes XML a truly international format, able to handle most modern languages.

<embeddeddtd>

<?xml version="1.0" encoding="iso-8859-1"?>

</embeddeddtd>

Document Type Declaration: -A document author can use an optional document type declaration after the XML declaration to indicate what the root element of the XML document will be and possibly to point to a document type definition. A typical document type declaration for a CellML document is shown below. Note that the document type declaration facility defined in the XML specification provides a lot more functionality than what is discussed or shown here.

<embedded>

<!DOCTYPE model SYSTEM "http://www.cellml.org/cellml/cellml_1_1.dtd">

</embedded>

Start / End Tag: -The simplest way of encoding the meaning of a piece of text in XML is to enclose it inside start and end tags. A start tag consists of the tag-name in between less-than and greater-than signs, and the matching end tag has a slash preceding the tag-name, as shown below. A *well-formed* XML document has an end-tag that matches every start-tag.

<my_tag>

the text data

</my_tag>

Element: - The combination of start-tag, data and end-tag is known as an element. The data may be plain text (as in the example above), further elements (sub-elements), or a combination of text and sub-elements. A document is usually made up of a tree of elements with a single root element as shown below.

<root_element>

<sub_element_1> data for sub-element 1 **</sub_element_1>**

<sub_element_2> data for sub-element 2 **</sub_element_2>**

</root_element>

Attribute: - Another way of putting data into an XML document is by adding attributes to start tags. The value of the attribute is usually intended to be data relevant

to the content of the current element. Whitespace is used to separate attributes from the tag-name and each other. Each attribute has a name followed by an equal's sign and the value of the attribute. The value of the attribute is enclosed in single or double quotes. In the example below, **<my_tag>** has two attributes: **att_1** and **att_2**.

<my_tag att_1="1**" att_2="**2**">** the text data **</my_tag>**

Empty Element: - If an element has no content, the end-tag can be left out. In this case, a slash is added to the end of the start-tag to indicate that this is an empty element. Element content is anything that the XML specification allows to appear between a start-tag and an end-tag, such as text, sub-elements, comments and processing instructions. An empty element may still have attributes, as shown below.

<my_empty_element att_1="1**" att_2="**2**" />**

Document Type Definition:- The Uniform Resource Identifier (URI) in a document type declaration can point to a document known as a document type definition (DTD). The format for a DTD is defined in the XML Specification and is not the same as for an XML document. A DTD may contain a set of rules that specify how the different tags in an XML document can be used together and the attributes that may belong to each tag. Most XML processors provide checking of XML documents against a DTD, allowing applications to quickly and painlessly check that the structure of an XML document is roughly correct. DTDs do not allow the specification of constraints on element and attribute content like "*the value of the **att_1** attribute must be a number*". This kind of validation can be handled by using XML Schema [http://www.w3.org/XML/Schema], the successor to DTDs which defines an XML-based file format.

Comment: - A document author can place comments in XML documents to add annotations intended for other humans reading the document. The contents of a comment are not regarded as part of the document's data. A comment is started with a less-than sign, exclamation mark, and two hyphens, and is ended with two-hyphens and a greater-than sign, as shown below. Comments may not be placed inside start- or end-tags.

<my_tag> content <!-- comment on content --**></my_tag>**

XML Namespace: - Namespaces in XML [http://www.w3.org/TR/REC-xml-names/] is a companion specification to the main XML specification. It provides a facility for associating the elements and/or attributes in all or part of a document with a particular schema, as indicated by a URI. The key aspect of the URI is that it is unique. The value of the URI need not have anything to do with the XML document that uses it, although typically it would be a good location for the XML Schema or DTD that defines the rules for the document type. The URI may be mapped to a prefix which may then be used in front of tag and attribute names, separated by a colon. If not mapped to a prefix, the URI sets the default schema for the current element and all of its children. A namespace declaration looks like an attribute on a start tag, but may be identified by the keyword xmlns. In the following example, the default namespace is

set to the CellML namespace, and the MathML namespace is declared and mapped to the **mathml** prefix, which is then used on a **<math>** element. Note that the **<model>** element and any children elements with no default namespace declaration or namespace prefix (such as the **<component>** element) will be in the CellML namespace.

<model
xmlns="http://www.cellml.org/cellml/1.1#**"xmlns:mathml="**http://www.w3.org/1998/Math/MathML**">**

<component> ... **</component>**

<mathml:math> ... math goes here ... **</mathml:math>**

</model>

5.2 Data Access on PDA/Pocket PC

- The data sent by the server in XML format is uploaded onto an online web server (dcetech.com)
- The XML data is read by an application and represented in a user friendly format.
- This is accessed by the user on Windows based PDA/Pocket PC.

A sample XML File which is uploaded at the dcetech.com server is as given below:

```
<?xml version="1.0" encoding="utf-8" ?>
<trafficmanagement>
        <record>
          <position>Rajouri Garden</position>
          <node>1</node>
        </record>

        <record>
         <position>Uttam Nagar</position>
         <node>3</node>
        </record>

        <record>
         <position>Subash Nagar</position>
         <node>2</node>
         </record>
 </trafficmanagement>
```

There is a root element called Traffic Management. Under this root element, there are sub-elements called Record. Each Record consists of two sub-elements. Position and Node. Position contains the position of location on the Stretch of the Road. Node contains the unique node each stretch has been given for the EBOX. The Graphical

User Interface for this Module contains a list of Destination and Source Locations as given by the XML file “nodeposition.xml”. The user selects the source and destination addresses in the List Boxes and on submitting the Query the Module locates the Nodes for given Locations/Positions. These node values are then passed onto the Optimization Module where a text file containing the total number of nodes and the distances of each node from every other node is given. On application of this module, the shortest route from Source to Destination with the intermediate nodes is given, along with the expected time it would take to travel that distance. These are displayed to the users who are using this KAYO application on their Mobile Pocket PC / PDA.

Once the module gives the shortest route through the intermediate nodes, a reverse mechanism is used to find out the location/position using the same XML File ‘nodeposition.xml’.

Pocket PC/PDA for which the application is being made is using Windows CE Operating Systems from Microsoft Corporation.

A traveler who wishes to reach at the earliest possible time from a Source Location to a Destination Location that is being covered using KAYO simply uses his Pocket PC/PDA having Internet Connectivity to connect to the online database whose data is being updated regularly from various Ebox on various stretches of Roads. He enters his locations using a user-friendly GUI and is instructed by KAYO on the best shortest possible path from a Source Location to a Destination Location.

6. Conclusion & References

6.1 Conclusion

KAYO uses the already existing infrastructure by a better management of traffic by using the proposed technology, and will be able to make world a better place to live in by relieving commuters of their ever growing woes.

The results of KAYO were impressing. It can efficiently and reliably track vehicle speeds as they travel down the road, and can do so with a reasonable degree of accuracy. Noticeably, the performance of the tracking can be very fast, and can handle 20 frames per second in real time. The system will keep itself updated within every t_s units of time.

Apart from giving the user the details of the shortest path KAYO will also provide other information like the emergency helps available on the given route, the food outlets and other small utilities which can be useful to a commuter. KAYO will cover the whole city under its non-deceivable eye. Since the number of users dependent on KAYO will keep on growing, the system must be at its best at each instance of its functioning. Also care would be taken that due to some unforeseen events the traffic on certain tracks may increase, so proper partitioning of the traffic have to be done to make the commuters of the city free of their ever growing woes. KAYO will work in any type of environment (i.e. any weather) and wide range of temperatures etc. conditions that are possible in highly dynamic environment.

6.2 KAYO Uniqueness

- KAYO is being designed keeping in mind that system should not be expensive to implement.
- The project KAYO is dynamic in nature i.e. it keeps track of the latest available data and gives its user the latest information.
- KAYO modifies itself (during image comparison) according to the changes taking in the weather condition on the road stretch (for example we have parameters which we can modify according to the conditions on the road).
- KAYO is simple and cost effective and implementable with good scalability.
- KAYO does not interfere with the privacy of the commuters as in the case of customers assigned RFID’s where the position of vehicle can be traced.

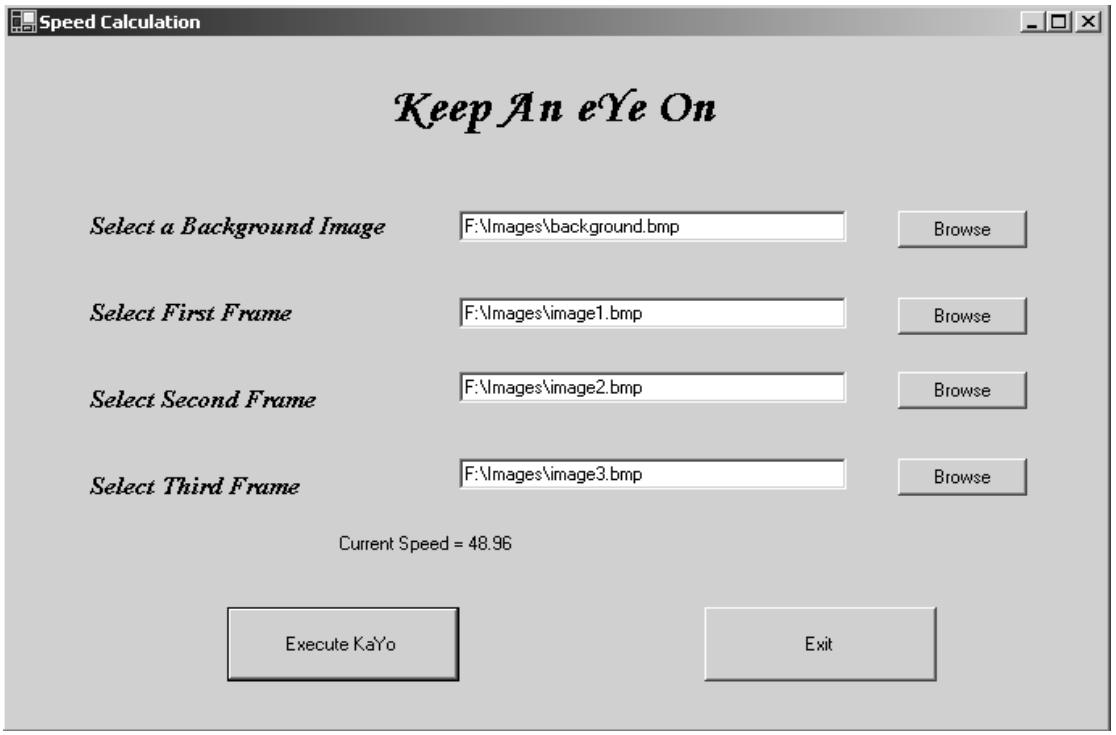
6.3 Further Work

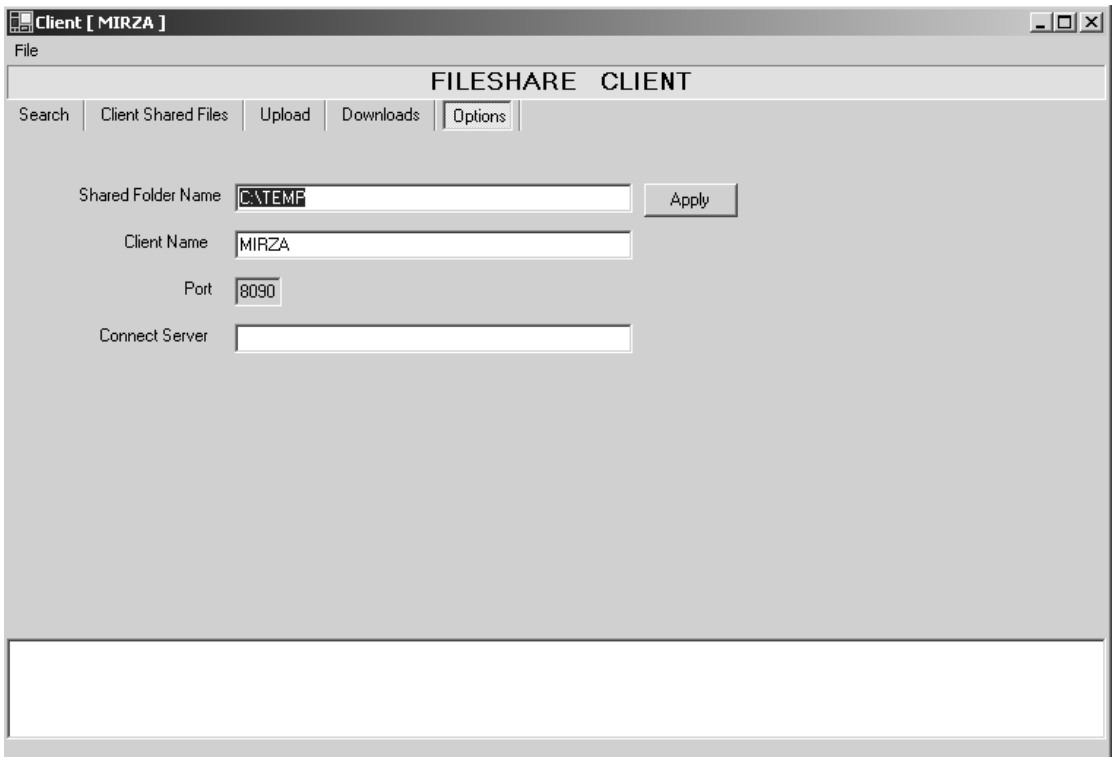
Some additional work could also be implemented. It might be useful to track multiple lines per lane, so that smaller vehicles, such as motorcycles could be tracked more robustly. Additional logic can also be added to track vehicles across lanes. There are some issues with robustness, such as a tall truck, but overall the system showed a strong degree of reliability.

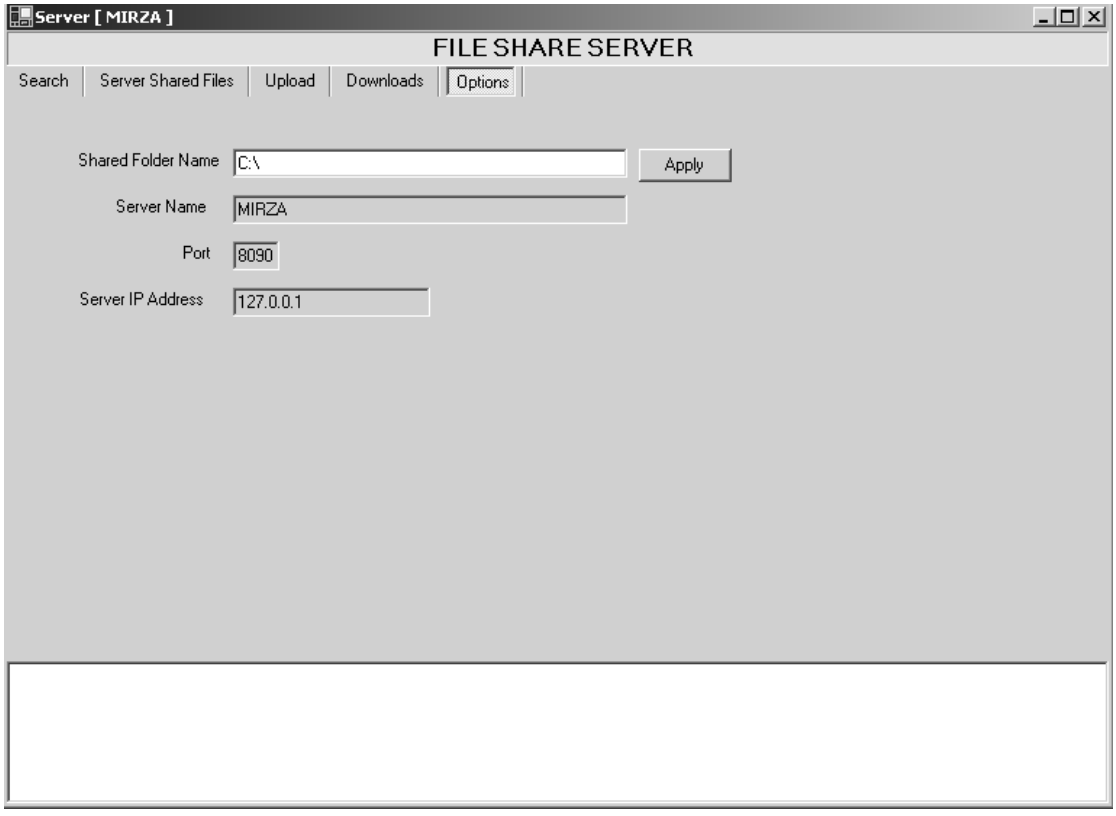
6.4 References

- Mike Burns, Christopher De Coro, Ananya Misra “Speed Trap.”
- Bevilacqua, Alessandro. "Effective Object Segmentation in a Traffic Monitoring Application."
- Francois, Alexandre and Gerard Medioni. "Adaptive Color Background Modeling for Real-Time Segmentation of Video Streams."
- Zhu, Zhigang, G. Xu, B. Yang, D. Shi and X. Lin. "VISATRAM: A real-time vision system for automatic traffic monitoring."
- Ivan Joseph, www.weblogs.asp.net/ivanjoseph
- Professional C# — WROX.
- HowStuffWorks, http://www.howstuffworks.com
- http://www.ukspeedcameras.co.uk
- http://www.trafficmaster.co.uk
- http://www.w3.org/XML— the W3C's XML page.
- http://www.ucc.ie/xml — the official XML FAQ.

7. Some Screen Shots of KAYO




Client [MIRZA]
File
FILESHARE CLIENT
Search
Client Shared Files
Upload
Downloads
Options
Shared Folder Name
C:\TEMP
Apply
Client Name
MIRZA
Port
8090
Connect Server


Server [MIRZA]
FILE SHARE SERVER
Search
Server Shared Files
Upload
Downloads
Options
Shared Folder Name
C:\
Apply
Server Name
MIRZA
Port
8090
Server IP Address
127.0.0.1

8. Implementation

Kayo GUI Form1.cs

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.IO;
namespace KayoGui
{
	public class Form1 : System.Windows.Forms.Form
	{
		private System.Windows.Forms.Label label1;
		private System.Windows.Forms.Label label4;
		private System.Windows.Forms.Label label2;
		private System.Windows.Forms.Label label3;
		private System.Windows.Forms.Label label5;
		private System.Windows.Forms.Button button1;
		private System.Windows.Forms.Button button2;
		public System.Windows.Forms.TextBox txtbackground;
		public System.Windows.Forms.TextBox txtframe1;
		public System.Windows.Forms.TextBox txtframe2;
		public System.Windows.Forms.TextBox txtframe3;
		private System.Windows.Forms.Button btbbackgroundbrowse;
		private System.Windows.Forms.Button btbframe1browse;
		private System.Windows.Forms.Button btbframe2browse;
		private System.Windows.Forms.Button btbframe3browse;
		private System.Windows.Forms.OpenFileDialog fbackground;
		private System.Windows.Forms.OpenFileDialog fframe1;
		private System.Windows.Forms.OpenFileDialog fframe2;
		private System.Windows.Forms.OpenFileDialog fframe3;

		string sbackground_path;
		string sframe1_path;
		string sframe2_path;
		string sframe3_path;
		private System.Windows.Forms.Label lblcspeed;

		public Form1()
		{
			InitializeComponent();
		}

		protected override void Dispose( bool disposing )
		{
			if( disposing )
			{
				if (components != null)
				{
					components.Dispose();
```

```
                }
            }
            base.Dispose( disposing );
        }

        #region Windows Form Designer generated code
        private void InitializeComponent()
        {
            this.label1 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.button1 = new System.Windows.Forms.Button();
            this.button2 = new System.Windows.Forms.Button();
            this.txtbackground = new System.Windows.Forms.TextBox();
            this.txtframe1 = new System.Windows.Forms.TextBox();
            this.txtframe2 = new System.Windows.Forms.TextBox();
            this.txtframe3 = new System.Windows.Forms.TextBox();
            this.btbbackgroundbrowse = new System.Windows.Forms.Button();
            this.btbframe1browse = new System.Windows.Forms.Button();
            this.btbframe2browse = new System.Windows.Forms.Button();
            this.btbframe3browse = new System.Windows.Forms.Button();
            this.fbackground = new System.Windows.Forms.OpenFileDialog();
            this.fframe1 = new System.Windows.Forms.OpenFileDialog();
            this.fframe2 = new System.Windows.Forms.OpenFileDialog();
            this.fframe3 = new System.Windows.Forms.OpenFileDialog();
            this.lblcspeed = new System.Windows.Forms.Label();
            this.SuspendLayout();

            this.label1.Font = new System.Drawing.Font("Monotype Corsiva", 24F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold | System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
            this.label1.Location = new System.Drawing.Point(232, 24);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(216, 40);
            this.label1.TabIndex = 0;
            this.label1.Text = "Keep An eYe On";

            this.label4.Font = new System.Drawing.Font("Times New Roman", 12F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold | System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
            this.label4.Location = new System.Drawing.Point(48, 112);
            this.label4.Name = "label4";
            this.label4.Size = new System.Drawing.Size(208, 24);
            this.label4.TabIndex = 3;
            this.label4.Text = "Select a Background Image";

            this.label2.Font = new System.Drawing.Font("Times New Roman", 12F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold | System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
            this.label2.Location = new System.Drawing.Point(48, 168);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(208, 24);
            this.label2.TabIndex = 4;
            this.label2.Text = "Select First Frame";
```

```
            this.label2.Click += new System.EventHandler(this.label2_Click);

            this.label3.Font = new System.Drawing.Font("Times New Roman", 12F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold | System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
            this.label3.Location = new System.Drawing.Point(48, 224);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(208, 24);
            this.label3.TabIndex = 5;
            this.label3.Text = "Select Second Frame";

            this.label5.Font = new System.Drawing.Font("Times New Roman", 12F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold | System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
            this.label5.Location = new System.Drawing.Point(48, 280);
            this.label5.Name = "label5";
            this.label5.Size = new System.Drawing.Size(208, 24);
            this.label5.TabIndex = 6;
            this.label5.Text = "Select Third Frame";

            this.button1.Location = new System.Drawing.Point(136, 368);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(144, 48);
            this.button1.TabIndex = 7;
            this.button1.Text = "Execute Ka&Yo";
            this.button1.Click += new System.EventHandler(this.button1_Click);

            this.button2.Location = new System.Drawing.Point(432, 368);
            this.button2.Name = "button2";
            this.button2.Size = new System.Drawing.Size(144, 48);
            this.button2.TabIndex = 8;
            this.button2.Text = "E&xit";
            this.button2.Click += new System.EventHandler(this.button2_Click);

            this.txtbackground.Location = new System.Drawing.Point(280, 112);
            this.txtbackground.Name = "txtbackground";
            this.txtbackground.Size = new System.Drawing.Size(240, 20);
            this.txtbackground.TabIndex = 9;
            this.txtbackground.Text = "";

            this.txtframe1.Location = new System.Drawing.Point(280, 168);
            this.txtframe1.Name = "txtframe1";
            this.txtframe1.Size = new System.Drawing.Size(240, 20);
            this.txtframe1.TabIndex = 10;
            this.txtframe1.Text = "";

            this.txtframe2.Location = new System.Drawing.Point(280, 216);
            this.txtframe2.Name = "txtframe2";
            this.txtframe2.Size = new System.Drawing.Size(240, 20);
            this.txtframe2.TabIndex = 11;
            this.txtframe2.Text = "";

            this.txtframe3.Location = new System.Drawing.Point(280, 272);
            this.txtframe3.Name = "txtframe3";
            this.txtframe3.Size = new System.Drawing.Size(240, 20);
            this.txtframe3.TabIndex = 12;
```

```
            this.txtframe3.Text = "";

            this.btbbackgroundbrowse.Location = new System.Drawing.Point(552, 112);
            this.btbbackgroundbrowse.Name = "btbbackgroundbrowse";
            this.btbbackgroundbrowse.Size = new System.Drawing.Size(80, 24);
            this.btbbackgroundbrowse.TabIndex = 13;
            this.btbbackgroundbrowse.Text = "Browse";
        this.btbbackgroundbrowse.Click += new System.EventHandler(this.btbbackgroundbrowse_Click);

            this.btbframe1browse.Location = new System.Drawing.Point(552, 168);
            this.btbframe1browse.Name = "btbframe1browse";
            this.btbframe1browse.Size = new System.Drawing.Size(80, 24);
            this.btbframe1browse.TabIndex = 14;
            this.btbframe1browse.Text = "Browse";
        this.btbframe1browse.Click += new System.EventHandler(this.btbframe1browse_Click);

            this.btbframe2browse.Location = new System.Drawing.Point(552, 216);
            this.btbframe2browse.Name = "btbframe2browse";
            this.btbframe2browse.Size = new System.Drawing.Size(80, 24);
            this.btbframe2browse.TabIndex = 15;
            this.btbframe2browse.Text = "Browse";
        this.btbframe2browse.Click += new System.EventHandler(this.btbframe2browse_Click);

            this.btbframe3browse.Location = new System.Drawing.Point(552, 272);
            this.btbframe3browse.Name = "btbframe3browse";
            this.btbframe3browse.Size = new System.Drawing.Size(80, 24);
            this.btbframe3browse.TabIndex = 16;
            this.btbframe3browse.Text = "Browse";
        this.btbframe3browse.Click += new System.EventHandler(this.btbframe3browse_Click);

            this.fbackground.FileOk += new
System.ComponentModel.CancelEventHandler(this.fbackground_FileOk_1);

        this.fframe1.FileOk += new System.ComponentModel.CancelEventHandler(this.fframe1_FileOk);

        this.fframe2.FileOk += new System.ComponentModel.CancelEventHandler(this.fframe2_FileOk);

        this.fframe3.FileOk += new System.ComponentModel.CancelEventHandler(this.fframe3_FileOk);

            this.lblcspeed.Location = new System.Drawing.Point(200, 320);
            this.lblcspeed.Name = "lblcspeed";
            this.lblcspeed.Size = new System.Drawing.Size(312, 32);
            this.lblcspeed.TabIndex = 17;

            this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
            this.ClientSize = new System.Drawing.Size(680, 445);
            this.Controls.Add(this.lblcspeed);
            this.Controls.Add(this.btbframe3browse);
            this.Controls.Add(this.btbframe2browse);
            this.Controls.Add(this.btbframe1browse);
            this.Controls.Add(this.btbbackgroundbrowse);
            this.Controls.Add(this.txtframe3);
            this.Controls.Add(this.txtframe2);
            this.Controls.Add(this.txtframe1);
            this.Controls.Add(this.txtbackground);
            this.Controls.Add(this.button2);
```

```
                this.Controls.Add(this.button1);
                this.Controls.Add(this.label5);
                this.Controls.Add(this.label3);
                this.Controls.Add(this.label2);
                this.Controls.Add(this.label4);
                this.Controls.Add(this.label1);
                this.Name = "Form1";
                this.Text = "Speed Calculation";
                this.Load += new System.EventHandler(this.Form1_Load);
                this.ResumeLayout(false);

        }
        #endregion

        [STAThread]
        static void Main()
        {
                Application.Run(new Form1());
        }

        private void button2_Click(object sender, System.EventArgs e)
        {
                Application.Exit();
        }

        private void btbbackgroundbrowse_Click(object sender, System.EventArgs e)
        {
                fbackground.ShowDialog();
        }

private void fbackground_FileOk_1(object sender, System.ComponentModel.CancelEventArgs e)
        {
                txtbackground.Text = fbackground.FileName;
        }

        private void btbframe1browse_Click(object sender, System.EventArgs e)
        {
                fframe1.ShowDialog();
        }

        private void btbframe2browse_Click(object sender, System.EventArgs e)
        {
                fframe2.ShowDialog();
        }

        private void btbframe3browse_Click(object sender, System.EventArgs e)
        {
                fframe3.ShowDialog();
        }

private void fframe1_FileOk(object sender, System.ComponentModel.CancelEventArgs e)
        {
                txtframe1.Text = fframe1.FileName;
        }

private void fframe2_FileOk(object sender, System.ComponentModel.CancelEventArgs e)
```

```
		{
			txtframe2.Text = fframe2.FileName;
		}

	private void fframe3_FileOk(object sender, System.ComponentModel.CancelEventArgs e)
		{
			txtframe3.Text = fframe3.FileName;
		}

	private void button1_Click(object sender, System.EventArgs e)
		{
			sbackground_path = txtbackground.Text;
			sframe1_path = txtframe1.Text;
			sframe2_path = txtframe2.Text;
			sframe3_path = txtframe3.Text;

			// create the gray scale images
			sbackground_path = sbackground_path.Replace(".bmp" , "1.bmp");
functions.SaveGIFWithNewColorTable(sbackground_path.Replace("1.bmp" ,
".bmp"),sbackground_path,256,false);

			sframe1_path = sframe1_path.Replace(".bmp" , "1.bmp");
functions.SaveGIFWithNewColorTable(sframe1_path.Replace("1.bmp" , ".bmp"),sframe1_path,256,false);

			sframe2_path = sframe2_path.Replace(".bmp" , "1.bmp");
functions.SaveGIFWithNewColorTable(sframe2_path.Replace("1.bmp" , ".bmp"),sframe2_path,256,false);

			sframe3_path = sframe3_path.Replace(".bmp" , "1.bmp");
functions.SaveGIFWithNewColorTable(sframe3_path.Replace("1.bmp" , ".bmp"),sframe3_path,256,false);
			//compare the images and delete the gary scale images

functions.image_comparison(sframe1_path,sbackground_path,sframe1_path.Replace(".bmp" , "1.bmp"));
			sframe1_path = sframe1_path.Replace(".bmp" , "1.bmp");
functions.image_comparison(sframe2_path,sbackground_path,sframe2_path.Replace(".bmp" , "1.bmp"));
			sframe2_path = sframe2_path.Replace(".bmp" , "1.bmp");

functions.image_comparison(sframe3_path,sbackground_path,sframe3_path.Replace(".bmp" , "1.bmp"));
			sframe3_path = sframe3_path.Replace(".bmp" , "1.bmp");

			//claculate the speed
			int [ ]a=new int[2];
			int [ ]b=new int[2];

			a=functions.coordinates_calculation(sframe1_path);
			b=functions.coordinates_calculation(sframe2_path);

			float c_speed=0;
			c_speed=functions.speed_return(a,b);

			//show the result
			lblcspeed.Text= " Current Speed = "+c_speed;

		}

	}
}
```

Kayo GIU Functions.cs

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.IO;
using System.Drawing.Imaging;

namespace KayoGui
{
	public class functions
	{
		static int nncolors=256;
		static public ColorPalette pal_quant = functions.GetnewColorPalette(nncolors);
		static public  void SaveGIFWithNewColorTable(string originalpath,string filename,int
		nColors,bool fTransparent)
		{

			Image image = Image.FromFile(originalpath);
			if (nColors > 256)
				nColors = 256;
			if (nColors < 2)
				nColors = 2;
			if (nColors == 256)
				nColors = 256;
			int   Width = image.Width;
			int   Height = image.Height;

			Bitmap  bitmap = new Bitmap(Width,
				Height,
				PixelFormat.Format8bppIndexed);
			ColorPalette pal = functions.GetnewColorPalette(nColors);
			for (int i = 0; i < nColors; i++)
			{
				int Alpha = 0XFF;                    // Colors are opaque.
				int Intensity = (i * 0XFF) /(nColors-1);    // Even distribution.
				if ( i == 0 && fTransparent) // Make this color index...
					Alpha = 0;          // Transparent
			pal.Entries[i] = Color.FromArgb((int)Intensity, (int)Intensity , (int)Intensity);
			}

			bitmap.Palette = pal;
			Bitmap BmpCopy = new Bitmap(Width,
				Height,
				PixelFormat.Format32bppArgb);
		{
			Graphics g = Graphics.FromImage(BmpCopy);

			g.PageUnit = GraphicsUnit.Pixel;

			// Transfer the Image to the Bitmap
			g.DrawImage(image, 0, 0, Width, Height);
```

```
            g.Dispose();
}

            // Lock a rectangular portion of the bitmap for writing.
            BitmapData  bitmapData;
            Rectangle   rect = new Rectangle(0, 0, Width, Height);

            bitmapData = bitmap.LockBits(
                        rect,
                        ImageLockMode.WriteOnly,
                        PixelFormat.Format8bppIndexed);
            IntPtr pixels = bitmapData.Scan0;

unsafe
{
            // Get the pointer to the image bits.
            // This is the unsafe operation.
            byte *  pBits;
            if (bitmapData.Stride > 0)
                        pBits = (byte *)pixels.ToPointer();
            else
                        pBits = (byte *)pixels.ToPointer() + bitmapData.Stride*(Height-1);
            uint stride = (uint)Math.Abs(bitmapData.Stride);

            for ( uint row = 0; row < Height; ++row )
            {
                        for ( uint col = 0; col < Width; ++col )
                        {
                                    Color     pixel;    // The source pixel.
                                    byte *    p8bppPixel = pBits + row*stride + col;
                                    pixel = BmpCopy.GetPixel((int)col, (int)row);
            double luminance = (pixel.R *0.299) + (pixel.G *0.587) +(pixel.B *0.114);
                                    *p8bppPixel = (byte)(luminance * (nColors-1)/255 +0.5);

                        } /* end loop for col */
            } /* end loop for row */
} /* end unsafe */

            bitmap.UnlockBits(bitmapData);
            bitmap.Save(filename);
            BmpCopy.Dispose();
            bitmap.Dispose();

}

static public ColorPalette GetnewColorPalette( int nColors )
{
            PixelFormat     bitscolordepth = PixelFormat.Format1bppIndexed;
            ColorPalette    palette;    // The Palette we are stealing
            Bitmap          bitmap;     // The source of the stolen palette
            if (nColors > 2)
                        bitscolordepth = PixelFormat.Format4bppIndexed;
            if (nColors > 16)
                        bitscolordepth = PixelFormat.Format8bppIndexed;
```

```
        // Make a new Bitmap object to get its Palette.
        bitmap = new Bitmap(320, 288, bitscolordepth );
        palette = bitmap.Palette;   // Grab the palette
        bitmap.Dispose();          // cleanup the source Bitmap
        return palette;            // Send the palette back
}

static public bool compare(Color a,Color b)
{
        byte b1,r1,g1,b2,r2,g2;
        b1 = a.B;
        r1 = a.R;
        g1 = a.G;
        b2 = b.B;
        r2 = b.R;
        g2 = b.G;
        if((abs(b1-b2))<65&&(abs(r1-r2))<65&&(abs(g1-g2))<65)
                return true;
        else
                return false;

}

static public byte abs(int a)
{
        if(a<0)
                return (byte)(a*(-1));
        else
                return (byte) a;
}

static public void image_comparison(string img1,string img2,string comp_image)
{

        Image image1 = Image.FromFile(img1);
        Image image2 = Image.FromFile(img2);
        Bitmap b_image1 = new Bitmap(image1);
        Bitmap b_image2 = new Bitmap(image2);
        for(int x = 0 ; x < image1.Width ; x++)
        {
                for ( int y =0 ; y < image1.Height ; y++)
                {
                        Color c_image1 = b_image1.GetPixel(x,y);
                        Color c_image2 = b_image2.GetPixel(x,y);
                        if(compare(c_image1,c_image2))
                        {
                                b_image2.SetPixel(x,y,Color.Black);

                        }
                        else
                        {
                                b_image2.SetPixel(x,y,Color.White);
                        }
                }
        }
        b_image2.Save(comp_image);
```

```
}

static public float speed_return(int [ ] a , int [ ] b )
{
        float speed = 0;
        if(a[0]> 130 && a[0] < 200 && b[0] >130 && b[0] <200)
        {
if(a[1]> 100 && a[1] < 200 && b[1] >100 && b[1] <200 && (abs(b[0]-a[0])<10))
                {
                        //fps is 20 fps
                        speed = (float)((abs(b[1]-a[1])* 100.0 * 3600.0) / 125000.0);
                }
        }
        return speed;
}

static public int[ ] coordinates_calculation(string spath)
{
        Image img = Image.FromFile(spath);
        Bitmap b = new Bitmap(img);
        int width=img.Width;
        int height=img.Height;
        int [ ] col = new int[width];
        int [ ] row = new int[height];

        int i,j;
        for(i=0;i<width;i++)
                col[i]=0;

        for(j=0;j<height;j++)
                row[j]=0;

        for(i=10;i<width-10;i++)
                for(j=10;j<height-10;j++)
                {
                        if(compare(b.GetPixel(i,j),Color.White))
                        {
                                col[i]++;
                                row[j]++;
                        }
                }

        int p1x=0 ;
        int p1y=0 ;
        int p2x=0 ;
        int p2y=0 ;
        int p3x=0 ;
        int p3y=0 ;
        int p4x=0 ;
        int p4y=0 ;

        for(i=0;i<width;i++)
        {
                if(col[i]>10)
                        break;
        }
```

```
                    p3x=p2x=i;

                    for(;i<width;i++)
                    {
                        if(col[i]>10)
                            p1x=p4x=i;
                    }

                    for(j=0;j<height;j++)
                    {
                        if(row[j]>10)
                            break;
                    }
                    p3y=p4y=j;

                    for(;j<height;j++)
                    {
                        if(row[j]>10)
                            p1y=p2y=j;
                    }


                    int d1  = 2;
                    int d2 = 2;
                    int d3 = 3;
                    int d4 = 3;
                    int d5 = 1;
                    int d6 = 1;

                    int q1x = 0;
                    int q1y = 0;
                    int q2x = 0;
                    int q2y = 0;

                    int[ ]  q = new int[2];

                    q1x = ((p1x * d4 ) + (p4x * d1))/ (d1+d4);
                    q1y = ((p1y * d4 ) + (p4y * d1))/ (d1+d4);
                    q2x = ((p2x * d3 ) + (p3x * d2))/ (d2+d3);
                    q2y = ((p2y * d3 ) + (p3y * d2))/ (d2+d3);

                    q[0] = (q1x + q2x ) / 2;
                    q[1] = (q1y + q2y ) / 2;

                    return q;

                }
            }
}
```

Optimization -

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.IO;
using System.Text;

namespace Optimization
{
	public class Form1 : System.Windows.Forms.Form
	{
		private System.Windows.Forms.Button btbexecute;
		private System.Windows.Forms.Button btbexit;
		private System.Windows.Forms.TextBox txta;
		private System.Windows.Forms.TextBox txtb;
		private System.Windows.Forms.Label label1;
		private System.Windows.Forms.Button btbbrowse;
		private System.Windows.Forms.TextBox txtfilename;
		private System.Windows.Forms.OpenFileDialog ffilename;
		private System.Windows.Forms.Label label2;
		private System.Windows.Forms.Label label3;

		struct mat
		{
			public int v ;
			public int path;
		};
		private System.ComponentModel.Container components = null;

		public Form1()
		{
			InitializeComponent();
		}

		protected override void Dispose( bool disposing )
		{
			if( disposing )
			{
				if (components != null)
				{
					components.Dispose();
				}
			}
			base.Dispose( disposing );
		}

		private void InitializeComponent()
		{
			this.btbexecute = new System.Windows.Forms.Button();
			this.btbexit = new System.Windows.Forms.Button();
			this.txta = new System.Windows.Forms.TextBox();
			this.txtb = new System.Windows.Forms.TextBox();
```

```
            this.txtfilename = new System.Windows.Forms.TextBox();
            this.label1 = new System.Windows.Forms.Label();
            this.btbbrowse = new System.Windows.Forms.Button();
            this.ffilename = new System.Windows.Forms.OpenFileDialog();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.SuspendLayout();

            this.btbexecute.Location = new System.Drawing.Point(88, 232);
            this.btbexecute.Name = "btbexecute";
            this.btbexecute.Size = new System.Drawing.Size(112, 40);
            this.btbexecute.TabIndex = 0;
            this.btbexecute.Text = "&Execute";
            this.btbexecute.Click += new System.EventHandler(this.btbexecute_Click);

            this.btbexit.Location = new System.Drawing.Point(320, 232);
            this.btbexit.Name = "btbexit";
            this.btbexit.Size = new System.Drawing.Size(112, 40);
            this.btbexit.TabIndex = 1;
            this.btbexit.Text = "E&xit";
            this.btbexit.Click += new System.EventHandler(this.btbexit_Click);

            this.txta.Location = new System.Drawing.Point(80, 160);
            this.txta.Name = "txta";
            this.txta.Size = new System.Drawing.Size(120, 20);
            this.txta.TabIndex = 2;
            this.txta.Text = "";

            this.txtb.Location = new System.Drawing.Point(304, 160);
            this.txtb.Name = "txtb";
            this.txtb.Size = new System.Drawing.Size(136, 20);
            this.txtb.TabIndex = 3;
            this.txtb.Text = "";

            this.txtfilename.Location = new System.Drawing.Point(176, 48);
            this.txtfilename.Name = "txtfilename";
            this.txtfilename.Size = new System.Drawing.Size(240, 20);
            this.txtfilename.TabIndex = 4;
            this.txtfilename.Text = "";
        this.txtfilename.TextChanged += new System.EventHandler(this.textBox1_TextChanged);

            this.label1.Font = new System.Drawing.Font("Monotype Corsiva", 18F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold | System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
            this.label1.Location = new System.Drawing.Point(19, 43);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(144, 32);
            this.label1.TabIndex = 5;
            this.label1.Text = "Select File";
            this.label1.Click += new System.EventHandler(this.label1_Click);

            this.btbbrowse.Location = new System.Drawing.Point(440, 48);
            this.btbbrowse.Name = "btbbrowse";
            this.btbbrowse.Size = new System.Drawing.Size(96, 24);
            this.btbbrowse.TabIndex = 6;
            this.btbbrowse.Text = "Browse";
```

```
                this.btbbrowse.Click += new System.EventHandler(this.btbbrowse_Click);

                this.ffilename.FileOk += new
System.ComponentModel.CancelEventHandler(this.ffilename_FileOk);

                this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
                this.label2.Location = new System.Drawing.Point(104, 120);
                this.label2.Name = "label2";
                this.label2.Size = new System.Drawing.Size(64, 24);
                this.label2.TabIndex = 7;
                this.label2.Text = "From";

                this.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
                this.label3.Location = new System.Drawing.Point(352, 120);
                this.label3.Name = "label3";
                this.label3.Size = new System.Drawing.Size(64, 24);
                this.label3.TabIndex = 8;
                this.label3.Text = "To";

                this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
                this.ClientSize = new System.Drawing.Size(552, 317);
                this.Controls.Add(this.label3);
                this.Controls.Add(this.label2);
                this.Controls.Add(this.btbbrowse);
                this.Controls.Add(this.label1);
                this.Controls.Add(this.txtfilename);
                this.Controls.Add(this.txtb);
                this.Controls.Add(this.txta);
                this.Controls.Add(this.btbexit);
                this.Controls.Add(this.btbexecute);
                this.Name = "Form1";
                this.Text = "Form1";
                this.ResumeLayout(false);

        }

        [STAThread]
        static void Main()
        {
                Application.Run(new Form1());
        }

        unsafe static int [ ] func1(mat [,] m,int a,int b)
        {
                int [ ]p;
                int [ ]q;
                int [ ]r;
                int [ ]temp;
                int [ ]temp1;
                int x=0,y=0;

                if(m[a,b].path==b)
                {
                        p=new int[2];
```

```
                p[0]=a;
                p[1]=b;
                return p;
        }

        temp=q=func1(m,a,m[a,b].path);
        while(temp[x]!=m[a,b].path)
        {
                x++;
        }

        temp=r=func1(m,m[a,b].path,b);
        while(temp[y]!=b)
        {
                y++;
        }
        p=new int[x+y+1];
        temp=p;
        temp1=q;
        int i=0;
        while(x!=0)
        {
                temp[i]=q[i];
                i++;
                x--;
        }

        temp1=r;
        int j=0;
        while(y!=0)
        {
                temp[i]=r[j];
                i++;
                j++;
                y--;
        }

        temp[i]=b;
        return(p);
}

unsafe static mat[,] func(int [,] arr,int n)
{
        int i,j,k,temp;
        int flag=1;
        int a,b;
        int [ ] p;
        mat[,] m = new mat[n,n];
        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                {
                        m[i,j].v=arr[i,j];
                        m[i,j].path=j;
                }
        }
```

```
		while(flag!=0)
		{
			flag=0;
			for(k=0;k<n;k++)
				for(i=0;i<n;i++)
					for(j=0;j<n;j++)
					{
						if(k==i || k==j)
							continue;
						temp=Math.Min(m[i,j].v,m[i,k].v+m[k,j].v);
						if(temp==m[i,j].v)
							continue;
						flag=1;
						m[i,j].v=temp;
						m[i,j].path=k;
					}
		}

		return m;

	}

	private void btbexit_Click(object sender, System.EventArgs e)
	{
		Application.Exit();
	}

	private void btbexecute_Click(object sender, System.EventArgs e)
	{
		int [ ] arra = new int[1];
		int [,] arr = file_load(txtfilename.Text,arra);
		int nodes = arra[0];

		int a , b;
		mat[,] z;
		z=func(arr,nodes);
		string a_text , b_text;
		a_text = txta.Text;
		b_text = txtb.Text;
		a = Int32.Parse(a_text);
		b = Int32.Parse(b_text);

		int [ ] p ;
		p=func1(z,a,b);
	}


	private void btbbrowse_Click(object sender, System.EventArgs e)
	{
		ffilename.ShowDialog();
	}

private void ffilename_FileOk(object sender, System.ComponentModel.CancelEventArgs e)
	{
		txtfilename.Text = ffilename.FileName;
```

```
        }

        static int[,] file_load(string filename, int [ ] arra)
        {
                FileStream fs = new FileStream(filename,FileMode.Open, FileAccess.Read);
                StreamReader sr = new StreamReader(fs);
                int nodes = Int32.Parse(sr.ReadLine());
                arra[0] = nodes;
                int [,] arr = new int[nodes,nodes];
                char[ ] b = new char[nodes];
                int [ ] number;
                for(int i = 0 ; i< nodes ; i++)
                {
                        string str;
                        str = sr.ReadLine();
                        number = intvalues(str , nodes);
                        for(int j =0 ; j<nodes ; j++)
                        {
                                arr[i,j] = number[j];
                        }
                }
                fs.Close();

                return arr;

        }

        static int[ ] intvalues(string str, int nodes)
        {
                int length = 0;
                int [ ] number = new int [nodes];
                int count=-1;
                 length = str.Length;
                char [ ] ch = str.ToCharArray();
                string stemp= null;
                int i = 0;
                while( (i<length) && (count < nodes))
                {
                        if(ch[i] != ' ')
                        {
                                stemp += ch[i].ToString();
                        }
                        else
                        {
                                number[++count] = Int32.Parse(stemp);
                                stemp = null;

                        }
                        i++;

                }
                number[++count] = Int32.Parse(stemp);
                return number;
        }
    }
}
```

Traffic Management

```
using System;
using System.Drawing;
using System.Collections;
using System.Windows.Forms;
using System.Data;
using System.Xml;

namespace TrafficManagement
{
	public class StartApplication : System.Windows.Forms.Form
	{
		private System.Windows.Forms.Label lblposition_x;
		private System.Windows.Forms.Label lblposition_y;
		private System.Windows.Forms.ListBox lstposition_x;
		private System.Windows.Forms.ListBox lstposition_y;
		private System.Windows.Forms.Label label1;
		private System.Windows.Forms.Button btbexit;
		private System.Windows.Forms.TextBox txtpath;
		private System.Windows.Forms.Label label2;
		private System.Windows.Forms.OpenFileDialog fbrowse;
		private System.Windows.Forms.Button cmdsubmit;
		private System.Windows.Forms.OpenFileDialog fbrowsenode;
		public string node_path = null;
		string from;
		string to;
		int node_from;
		int node_to;

		public StartApplication()
		{
			InitializeComponent();
		}

		protected override void Dispose( bool disposing )
		{
			base.Dispose( disposing );
		}
		#region Windows Form Designer generated code
		private void InitializeComponent()
		{
			this.cmdsubmit = new System.Windows.Forms.Button();
			this.lblposition_x = new System.Windows.Forms.Label();
			this.lblposition_y = new System.Windows.Forms.Label();
			this.lstposition_x = new System.Windows.Forms.ListBox();
			this.lstposition_y = new System.Windows.Forms.ListBox();
			this.label1 = new System.Windows.Forms.Label();
			this.btbexit = new System.Windows.Forms.Button();
			this.txtpath = new System.Windows.Forms.TextBox();
			this.label2 = new System.Windows.Forms.Label();
			this.fbrowse = new System.Windows.Forms.OpenFileDialog();
			this.fbrowsenode = new System.Windows.Forms.OpenFileDialog();

			this.cmdsubmit.Font = new System.Drawing.Font("Times New Roman", 9F,
			System.Drawing.FontStyle.Regular);
```

```
this.cmdsubmit.Location = new System.Drawing.Point(144, 280);
this.cmdsubmit.Size = new System.Drawing.Size(128, 32);
this.cmdsubmit.Text = "S&ubmit Query";
this.cmdsubmit.Click += new System.EventHandler(this.cmdsubmit_Click);

this.lblposition_x.Font = new System.Drawing.Font("Times New Roman", 9F,
System.Drawing.FontStyle.Regular);
this.lblposition_x.Location = new System.Drawing.Point(136, 72);
this.lblposition_x.Size = new System.Drawing.Size(120, 16);
this.lblposition_x.Text = "Starting Destination";

this.lblposition_y.Font = new System.Drawing.Font("Times New Roman", 9F,
System.Drawing.FontStyle.Regular);
this.lblposition_y.Location = new System.Drawing.Point(328, 72);
this.lblposition_y.Size = new System.Drawing.Size(120, 16);
this.lblposition_y.Text = "Final Destination";

this.lstposition_x.Location = new System.Drawing.Point(136, 112);
this.lstposition_x.Size = new System.Drawing.Size(128, 30);

this.lstposition_y.Location = new System.Drawing.Point(312, 112);
this.lstposition_y.Size = new System.Drawing.Size(136, 30);

this.label1.Font = new System.Drawing.Font("Times New Roman", 15.75F,
System.Drawing.FontStyle.Bold);
this.label1.Location = new System.Drawing.Point(136, 16);
this.label1.Size = new System.Drawing.Size(288, 32);
this.label1.Text = "Mobile Interface for KAYO";

this.btbexit.Font = new System.Drawing.Font("Times New Roman", 9F,
System.Drawing.FontStyle.Regular);
this.btbexit.Location = new System.Drawing.Point(320, 280);
this.btbexit.Size = new System.Drawing.Size(128, 32);
this.btbexit.Text = "E&xit";
this.btbexit.Click += new System.EventHandler(this.btbexit_Click);

this.txtpath.Location = new System.Drawing.Point(144, 208);
this.txtpath.Size = new System.Drawing.Size(312, 22);
this.txtpath.Text = "";

this.label2.Font = new System.Drawing.Font("Times New Roman", 11.25F,
System.Drawing.FontStyle.Bold);
this.label2.Location = new System.Drawing.Point(152, 176);
this.label2.Size = new System.Drawing.Size(296, 24);
this.label2.Text = "Path and its Cost from Source to Destintion";

this.Controls.Add(this.label2);
this.Controls.Add(this.txtpath);
this.Controls.Add(this.btbexit);
this.Controls.Add(this.label1);
this.Controls.Add(this.lstposition_y);
this.Controls.Add(this.lstposition_x);
this.Controls.Add(this.lblposition_y);
this.Controls.Add(this.lblposition_x);
this.Controls.Add(this.cmdsubmit);
this.Text = "Form1";
```

```
                this.Load += new System.EventHandler(this.StartApplication_Load);

        }
        #endregion

        static void Main()
        {
                Application.Run(new StartApplication());
        }

        private void StartApplication_Load(object sender, System.EventArgs e)
        {
node_path = @"F:\Documents and Settings\Amrit\Desktop\Microsoft Project\Data\nodeposition.xml";

                XmlTextReader trnode = new XmlTextReader(node_path);
                while(!trnode.EOF)
                        {
        if((trnode.MoveToContent()==XmlNodeType.Element) && (trnode.Name=="position"))
                                {
                                lstposition_x.Items.Add(trnode.ReadElementString());
                                lstposition_y.Items.Add(trnode.ReadElementString());
                                        trnode.Read();
                                }
                                else
                                {
                                        trnode.Read();
                                }
                        }
        }

        private void cmdsubmit_Click(object sender, System.EventArgs e)
        {
                from = lstposition_x.SelectedItem.ToString();
                to = lstposition_y.SelectedItem.ToString();
                XmlTextReader trnode = new XmlTextReader(node_path);
                while(!trnode.EOF)
                {
        if((trnode.MoveToContent()==XmlNodeType.Element) && (trnode.Name=="record"))
                        {
    if((trnode.MoveToContent() == XmlNodeType.Element) && (trnode.Name=="position"))
                                {
                                        if(trnode.ReadElementString()==from)
                                        {
                                                trnode.Read();
                                node_from = Int32.Parse(trnode.ReadElementString());
                                        }
                                        if(trnode.ReadElementString()==to)
                                        {
                                                trnode.Read();
                                        node_to = Int32.Parse(trnode.ReadElementString());
                                        }
                                }
                        }
                        trnode.Read();
                }
```

```
                    Console.WriteLine(node_from );
                    Console.WriteLine(node_to);
                }

                private void btbexit_Click(object sender, System.EventArgs e)
                {
                    Application.Exit();
                }
        }
}
```

Client

ClientInfo.cs

```
using System;
using System.Collections;
namespace CLIENT
{
        public class ClientInfo
        {
                public string username ;
                public string password ;
                public string ipaddress ;
                public string sharedfileName ;
                public string sharedfilesSize ;
                public string sharedfilesPath ;
        }
}
```

ClientCollections.cs

```
using System;
using System.Collections;
using System.Text.RegularExpressions ;

namespace CLIENT
{
        public class ClientCollection: System.Collections.CollectionBase
        {

                public void AddClient(Object client)
                {
                    List.Add(client);
                }

                public int GetCount()
                {
                    return List.Count;
                }

                public Object GetAt(int npos)
                {
                    Object obj = List[npos];
```

```
                        if ( npos > GetCount() )
                                return null ;
                        return obj ;
                }

                public ArrayList SearchFiles(string pat)
                {
                        ArrayList objList = new ArrayList();
                        for ( int i=0; i < List.Count ; i++)
                        {
                                ClientInfo obj = (ClientInfo)List[i];
                                int pos = obj.sharedfileName.IndexOf(pat);
                                if ( pos >= 0 )
                                        objList.Add(obj);
                        }
                        return objList;
                }
        }
}
```

Form1.cs

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Net;
using System.Net.Sockets;
using System.Threading;
using System.Text;
using System.IO ;
using System.Text.RegularExpressions ;

namespace CLIENT
{
        public class Form1 : System.Windows.Forms.Form
        {
                private System.ComponentModel.Container components = null;
                string server = "";
                ClientCollection clientcollection =null;

                string[ ] DFSCommandList ;
                Socket serversocket;
                private System.Windows.Forms.Label label1;
                Socket clientsock = null ;
                string[ ] cmdList = null ;
                private string shared_file_name ;
                private string shared_file_path ;
                private string shared_file_size ;
                private System.Windows.Forms.TabControl tabControl1;
                private System.Windows.Forms.TabPage tabPage1;
                private System.Windows.Forms.TabPage tabPage2;
                private System.Windows.Forms.ListView listView1;
                private System.Windows.Forms.ColumnHeader SFILENAME;
```

```
private System.Windows.Forms.ColumnHeader SFILESIZE;
private System.Windows.Forms.ColumnHeader SFILEPATH;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.TextBox textBox2;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.TextBox textBox3;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.TextBox textBox4;
private string local_shared_dir = "C:\\TEMP" ;
private System.Windows.Forms.ListView listView3;
private System.Windows.Forms.ColumnHeader STFILENAME;
private System.Windows.Forms.ColumnHeader STFILESIZE;
private System.Windows.Forms.ColumnHeader STCLIENT;
private System.Windows.Forms.Button button3;
private System.Windows.Forms.ListView listView4;
private System.Windows.Forms.ColumnHeader UPFILENAME;
private System.Windows.Forms.ColumnHeader UPFILESIZE;
private System.Windows.Forms.ColumnHeader UPCURRENTSIZE;
private System.Windows.Forms.ColumnHeader UPCLIENT;
private System.Windows.Forms.ColumnHeader ST_BYTES_COPIED;
private System.Windows.Forms.ColumnHeader ST_PERCENT;
private System.Windows.Forms.ColumnHeader UPPERCENT;
private System.Windows.Forms.TabPage tabPage4;
private System.Windows.Forms.TabPage tabPage3;
private System.Windows.Forms.TabPage tabPage5;
private System.Windows.Forms.MainMenu mainMenu1;
private System.Windows.Forms.MenuItem menuItem1;
private System.Windows.Forms.MenuItem menuItem2;
private System.Windows.Forms.MenuItem menuItem3;
private System.Windows.Forms.MenuItem menuItem4;
private System.Windows.Forms.Splitter splitter1;
private System.Windows.Forms.Splitter splitter2;
private System.Windows.Forms.ListBox listBox1;
private System.Windows.Forms.Panel panel1;
private System.Windows.Forms.TextBox SearchText;
private System.Windows.Forms.Button button1;
private System.Windows.Forms.Button button2;
private System.Windows.Forms.Panel panel2;
private System.Windows.Forms.ListView listView2;
private System.Windows.Forms.ColumnHeader SCFILENAME;
private System.Windows.Forms.ColumnHeader SCFILESIZE;
private System.Windows.Forms.ColumnHeader SCFILEPATH;
private System.Windows.Forms.ColumnHeader SCUSER;
private System.Windows.Forms.TextBox textBox1;
private System.Windows.Forms.Label label2;
private int total_clients_connected =0;


public Form1()
{
        InitializeComponent();
}

protected override void Dispose( bool disposing )
{
        if ( serversocket != null )
```

```
                serversocket.Close();

        if( disposing )
        {
                if (components != null)
                {
                        components.Dispose();
                }
        }
        base.Dispose( disposing );
}

#region Windows Form Designer generated code
private void InitializeComponent()
{
        this.ST_PERCENT = new System.Windows.Forms.ColumnHeader();
        this.STFILESIZE = new System.Windows.Forms.ColumnHeader();
        this.STCLIENT = new System.Windows.Forms.ColumnHeader();
        this.UPCURRENTSIZE = new System.Windows.Forms.ColumnHeader();
        this.textBox2 = new System.Windows.Forms.TextBox();
        this.textBox3 = new System.Windows.Forms.TextBox();
        this.tabPage4 = new System.Windows.Forms.TabPage();
        this.listView3 = new System.Windows.Forms.ListView();
        this.STFILENAME = new System.Windows.Forms.ColumnHeader();
        this.ST_BYTES_COPIED = new System.Windows.Forms.ColumnHeader();
        this.tabPage5 = new System.Windows.Forms.TabPage();
        this.button3 = new System.Windows.Forms.Button();
        this.label4 = new System.Windows.Forms.Label();
        this.label3 = new System.Windows.Forms.Label();
        this.label5 = new System.Windows.Forms.Label();
        this.textBox4 = new System.Windows.Forms.TextBox();
        this.textBox1 = new System.Windows.Forms.TextBox();
        this.label2 = new System.Windows.Forms.Label();
        this.tabPage2 = new System.Windows.Forms.TabPage();
        this.listView1 = new System.Windows.Forms.ListView();
        this.SFILENAME = new System.Windows.Forms.ColumnHeader();
        this.SFILESIZE = new System.Windows.Forms.ColumnHeader();
        this.SFILEPATH = new System.Windows.Forms.ColumnHeader();
        this.tabPage3 = new System.Windows.Forms.TabPage();
        this.listView4 = new System.Windows.Forms.ListView();
        this.UPFILENAME = new System.Windows.Forms.ColumnHeader();
        this.UPFILESIZE = new System.Windows.Forms.ColumnHeader();
        this.UPPERCENT = new System.Windows.Forms.ColumnHeader();
        this.UPCLIENT = new System.Windows.Forms.ColumnHeader();
        this.tabPage1 = new System.Windows.Forms.TabPage();
        this.panel2 = new System.Windows.Forms.Panel();
        this.listView2 = new System.Windows.Forms.ListView();
        this.SCFILENAME = new System.Windows.Forms.ColumnHeader();
        this.SCFILESIZE = new System.Windows.Forms.ColumnHeader();
        this.SCFILEPATH = new System.Windows.Forms.ColumnHeader();
        this.SCUSER = new System.Windows.Forms.ColumnHeader();
        this.splitter1 = new System.Windows.Forms.Splitter();
        this.panel1 = new System.Windows.Forms.Panel();
        this.SearchText = new System.Windows.Forms.TextBox();
        this.button1 = new System.Windows.Forms.Button();
        this.button2 = new System.Windows.Forms.Button();
```

```
                this.label1 = new System.Windows.Forms.Label();
                this.tabControl1 = new System.Windows.Forms.TabControl();
                this.mainMenu1 = new System.Windows.Forms.MainMenu();
                this.menuItem1 = new System.Windows.Forms.MenuItem();
                this.menuItem2 = new System.Windows.Forms.MenuItem();
                this.menuItem3 = new System.Windows.Forms.MenuItem();
                this.menuItem4 = new System.Windows.Forms.MenuItem();
                this.splitter2 = new System.Windows.Forms.Splitter();
                this.listBox1 = new System.Windows.Forms.ListBox();
                this.tabPage4.SuspendLayout();
                this.tabPage5.SuspendLayout();
                this.tabPage2.SuspendLayout();
                this.tabPage3.SuspendLayout();
                this.tabPage1.SuspendLayout();
                this.panel2.SuspendLayout();
                this.panel1.SuspendLayout();
                this.tabControl1.SuspendLayout();
                this.SuspendLayout();
                this.ST_PERCENT.Text = "Percent";
                this.STFILESIZE.Text = "FileSize";
                this.STFILESIZE.Width = 100;

                this.STCLIENT.Text = "ClientName";
                this.STCLIENT.Width = 100;
                this.UPCURRENTSIZE.Text = "Completed Size";
                this.UPCURRENTSIZE.Width = 150;
                this.textBox2.Location = new System.Drawing.Point(152, 32);
                this.textBox2.Name = "textBox2";
                this.textBox2.Size = new System.Drawing.Size(272, 20);
                this.textBox2.TabIndex = 1;
                this.textBox2.Text = "";
                this.textBox3.Location = new System.Drawing.Point(152, 64);
                this.textBox3.Name = "textBox3";
                this.textBox3.Size = new System.Drawing.Size(272, 20);
                this.textBox3.TabIndex = 3;
                this.textBox3.Text = "";
                this.tabPage4.Controls.AddRange(new System.Windows.Forms.Control[ ] {
                                                        this.listView3});
                this.tabPage4.Location = new System.Drawing.Point(4, 25);
                this.tabPage4.Name = "tabPage4";
                this.tabPage4.Size = new System.Drawing.Size(744, 331);
                this.tabPage4.TabIndex = 3;
                this.tabPage4.Text = "Downloads";
        this.listView3.Columns.AddRange(new System.Windows.Forms.ColumnHeader[ ] {
this.STFILENAME,
this.STFILESIZE,
this.ST_BYTES_COPIED,
this.ST_PERCENT,
this.STCLIENT});
                this.listView3.Dock = System.Windows.Forms.DockStyle.Fill;
                this.listView3.FullRowSelect = true;
                this.listView3.Name = "listView3";
                this.listView3.Size = new System.Drawing.Size(744, 331);
                this.listView3.TabIndex = 0;
                this.listView3.View = System.Windows.Forms.View.Details;
```

```
this.STFILENAME.Text = "FileName";
this.STFILENAME.Width = 150;

this.ST_BYTES_COPIED.Text = "Bytes Copied";
this.ST_BYTES_COPIED.Width = 100;

this.tabPage5.Controls.AddRange(new System.Windows.Forms.Control[ ] {
                                                    this.button3,
                                                    this.textBox3,
                                                    this.label4,
                                                    this.textBox2,
                                                    this.label3,
                                                    this.label5,
                                                    this.textBox4,
                                                    this.textBox1,
                                                    this.label2});
this.tabPage5.Location = new System.Drawing.Point(4, 25);
this.tabPage5.Name = "tabPage5";
this.tabPage5.Size = new System.Drawing.Size(744, 331);
this.tabPage5.TabIndex = 2;
this.tabPage5.Text = "Options";

this.button3.Location = new System.Drawing.Point(432, 32);
this.button3.Name = "button3";
this.button3.Size = new System.Drawing.Size(64, 23);
this.button3.TabIndex = 4;
this.button3.Text = "Apply";
this.button3.Click += new System.EventHandler(this.button3_Click);

this.label4.Location = new System.Drawing.Point(40, 64);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(100, 16);
this.label4.TabIndex = 2;
this.label4.Text = "Client Name";
this.label4.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.label3.Location = new System.Drawing.Point(32, 32);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(120, 16);
this.label3.TabIndex = 0;
this.label3.Text = "Shared Folder Name";
this.label3.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.label5.Location = new System.Drawing.Point(40, 96);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(100, 16);
this.label5.TabIndex = 2;
this.label5.Text = "Port";
this.label5.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.textBox4.Location = new System.Drawing.Point(152, 96);
this.textBox4.Name = "textBox4";
this.textBox4.ReadOnly = true;
this.textBox4.Size = new System.Drawing.Size(32, 20);
this.textBox4.TabIndex = 3;
this.textBox4.Text = "8090";
```

```
                        this.textBox1.Location = new System.Drawing.Point(152, 128);
                        this.textBox1.Name = "textBox1";
                        this.textBox1.Size = new System.Drawing.Size(272, 20);
                        this.textBox1.TabIndex = 3;
                        this.textBox1.Text = "";

                        this.label2.Location = new System.Drawing.Point(40, 128);
                        this.label2.Name = "label2";
                        this.label2.Size = new System.Drawing.Size(100, 16);
                        this.label2.TabIndex = 2;
                        this.label2.Text = "Connect Server";
                        this.label2.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

                        this.tabPage2.Controls.AddRange(new System.Windows.Forms.Control[ ] {
                                                                this.listView1});
                        this.tabPage2.Location = new System.Drawing.Point(4, 25);
                        this.tabPage2.Name = "tabPage2";
                        this.tabPage2.Size = new System.Drawing.Size(744, 331);
                        this.tabPage2.TabIndex = 1;
                        this.tabPage2.Text = "Client Shared Files";

                        this.listView1.Columns.AddRange(new
System.Windows.Forms.ColumnHeader[ ] {
        this.SFILENAME,
        this.SFILESIZE,
        this.SFILEPATH});
                        this.listView1.Dock = System.Windows.Forms.DockStyle.Fill;
                        this.listView1.FullRowSelect = true;
                        this.listView1.Name = "listView1";
                        this.listView1.Size = new System.Drawing.Size(744, 331);
                        this.listView1.TabIndex = 4;
                        this.listView1.View = System.Windows.Forms.View.Details;

                        this.SFILENAME.Text = "FileName";
                        this.SFILENAME.Width = 250;

                        this.SFILESIZE.Text = "File Size";
                        this.SFILESIZE.Width = 100;

                        this.SFILEPATH.Text = "FilePath";
                        this.SFILEPATH.Width = 350;

                        this.tabPage3.Controls.AddRange(new System.Windows.Forms.Control[ ] {
                                                                this.listView4});
                        this.tabPage3.Location = new System.Drawing.Point(4, 25);
                        this.tabPage3.Name = "tabPage3";
                        this.tabPage3.Size = new System.Drawing.Size(744, 331);
                        this.tabPage3.TabIndex = 4;
                        this.tabPage3.Text = "Upload";

                        this.listView4.Columns.AddRange(new
System.Windows.Forms.ColumnHeader[ ] {
        this.UPFILENAME,
        this.UPFILESIZE,
        this.UPCURRENTSIZE,
```

```
        this.UPPERCENT,
        this.UPCLIENT});
                        this.listView4.Dock = System.Windows.Forms.DockStyle.Fill;
                        this.listView4.FullRowSelect = true;
                        this.listView4.MultiSelect = false;
                        this.listView4.Name = "listView4";
                        this.listView4.Size = new System.Drawing.Size(744, 331);
                        this.listView4.TabIndex = 0;
                        this.listView4.View = System.Windows.Forms.View.Details;

                        this.UPFILENAME.Text = "FileName";
                        this.UPFILENAME.Width = 150;

                        this.UPFILESIZE.Text = "FileSize";
                        this.UPFILESIZE.Width = 150;

                        this.UPPERCENT.Text = "Percent";

                        this.UPCLIENT.Text = "Client Uploading";
                        this.UPCLIENT.Width = 200;

                        this.tabPage1.Controls.AddRange(new System.Windows.Forms.Control[ ] {
                                                        this.panel2,
                                                        this.splitter1,
                                                        this.panel1});
                        this.tabPage1.Location = new System.Drawing.Point(4, 25);
                        this.tabPage1.Name = "tabPage1";
                        this.tabPage1.Size = new System.Drawing.Size(744, 331);
                        this.tabPage1.TabIndex = 0;
                        this.tabPage1.Text = "Search";

                        this.panel2.Controls.AddRange(new System.Windows.Forms.Control[ ] {
                                                        this.listView2});
                        this.panel2.Dock = System.Windows.Forms.DockStyle.Fill;
                        this.panel2.Location = new System.Drawing.Point(0, 43);
                        this.panel2.Name = "panel2";
                        this.panel2.Size = new System.Drawing.Size(744, 288);
                        this.panel2.TabIndex = 11;

                        this.listView2.Columns.AddRange(new
System.Windows.Forms.ColumnHeader[ ] {
        this.SCFILENAME,
        this.SCFILESIZE,
        this.SCFILEPATH,
                                                                this.SCUSER});
                        this.listView2.Dock = System.Windows.Forms.DockStyle.Fill;
                        this.listView2.FullRowSelect = true;
                        this.listView2.Name = "listView2";
                        this.listView2.Size = new System.Drawing.Size(744, 288);
                        this.listView2.TabIndex = 10;
                        this.listView2.View = System.Windows.Forms.View.Details;

                        this.SCFILENAME.Text = "FileName";
                        this.SCFILENAME.Width = 250;

                        this.SCFILESIZE.Text = "FileSize";
```

```
            this.SCFILESIZE.Width = 100;

            this.SCFILEPATH.Text = "FilePath";
            this.SCFILEPATH.Width = 250;

            this.SCUSER.Text = "User";
            this.SCUSER.Width = 100;

            this.splitter1.Dock = System.Windows.Forms.DockStyle.Top;
            this.splitter1.Location = new System.Drawing.Point(0, 40);
            this.splitter1.Name = "splitter1";
            this.splitter1.Size = new System.Drawing.Size(744, 3);
            this.splitter1.TabIndex = 10;
            this.splitter1.TabStop = false;

            this.panel1.Controls.AddRange(new System.Windows.Forms.Control[ ] {
                                                    this.SearchText,
                                                    this.button1,
                                                    this.button2});
            this.panel1.Dock = System.Windows.Forms.DockStyle.Top;
            this.panel1.Name = "panel1";
            this.panel1.Size = new System.Drawing.Size(744, 40);
            this.panel1.TabIndex = 8;

            this.SearchText.Location = new System.Drawing.Point(72, 8);
            this.SearchText.Name = "SearchText";
            this.SearchText.Size = new System.Drawing.Size(216, 20);
            this.SearchText.TabIndex = 8;
            this.SearchText.Text = "";

            this.button1.Enabled = false;
            this.button1.Location = new System.Drawing.Point(8, 8);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(56, 23);
            this.button1.TabIndex = 7;
            this.button1.Text = "Search";
            this.button1.Click += new System.EventHandler(this.button1_Click_1);

            this.button2.Enabled = false;
            this.button2.Location = new System.Drawing.Point(296, 8);
            this.button2.Name = "button2";
            this.button2.Size = new System.Drawing.Size(72, 23);
            this.button2.TabIndex = 7;
            this.button2.Text = "Download";
            this.button2.Click += new System.EventHandler(this.button2_Click_2);

            this.label1.BackColor =
System.Drawing.Color.FromArgb(((System.Byte)(224)), ((System.Byte)(224)), ((System.Byte)(224)));
            this.label1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
            this.label1.Dock = System.Windows.Forms.DockStyle.Top;
            this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(752, 24);
            this.label1.TabIndex = 2;
            this.label1.Text = "FILESHARE    CLIENT";
```

```
            this.label1.TextAlign = System.Drawing.ContentAlignment.TopCenter;

            this.tabControl1.Appearance =
System.Windows.Forms.TabAppearance.FlatButtons;
            this.tabControl1.Controls.AddRange(new System.Windows.Forms.Control[ ] {
                  this.tabPage1,
                  this.tabPage2,
                  this.tabPage3,
                  this.tabPage4,
                  this.tabPage5});
            this.tabControl1.Dock = System.Windows.Forms.DockStyle.Top;
            this.tabControl1.Location = new System.Drawing.Point(0, 24);
            this.tabControl1.Name = "tabControl1";
            this.tabControl1.SelectedIndex = 0;
            this.tabControl1.Size = new System.Drawing.Size(752, 360);
            this.tabControl1.TabIndex = 8;
            this.tabControl1.SelectedIndexChanged += new
System.EventHandler(this.tabControl1_SelectedIndexChanged);

      this.mainMenu1.MenuItems.AddRange(new System.Windows.Forms.MenuItem[ ] {
                                                                  this.menuItem1});
            this.menuItem1.Index = 0;
      this.menuItem1.MenuItems.AddRange(new System.Windows.Forms.MenuItem[ ] {
            this.menuItem2,
            this.menuItem3,
            this.menuItem4});
            this.menuItem1.Text = "File";

            this.menuItem2.Index = 0;
            this.menuItem2.Text = "Connect";
            this.menuItem2.Click += new System.EventHandler(this.menuItem2_Click);

            this.menuItem3.Index = 1;
            this.menuItem3.Text = "Disconnect";
            this.menuItem3.Click += new System.EventHandler(this.menuItem3_Click);

            this.menuItem4.Index = 2;
            this.menuItem4.Text = "Exit";
            this.menuItem4.Click += new System.EventHandler(this.menuItem4_Click);

            this.splitter2.Dock = System.Windows.Forms.DockStyle.Top;
            this.splitter2.Location = new System.Drawing.Point(0, 384);
            this.splitter2.Name = "splitter2";
            this.splitter2.Size = new System.Drawing.Size(752, 8);
            this.splitter2.TabIndex = 9;
            this.splitter2.TabStop = false;

            this.listBox1.Dock = System.Windows.Forms.DockStyle.Fill;
            this.listBox1.Location = new System.Drawing.Point(0, 392);
            this.listBox1.Name = "listBox1";
            this.listBox1.Size = new System.Drawing.Size(752, 69);
            this.listBox1.TabIndex = 11;

            this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
            this.ClientSize = new System.Drawing.Size(752, 473);
            this.Controls.AddRange(new System.Windows.Forms.Control[ ] {
```

```
                                                this.listBox1,
                                                this.splitter2,
                                                this.tabControl1,
                                                this.label1});
                        this.Menu = this.mainMenu1;
                        this.Name = "Form1";
                        this.Text = "Client";
this.Closing += new System.ComponentModel.CancelEventHandler(this.Form1_Closing);
                        this.Load += new System.EventHandler(this.Form1_Load);
                        this.tabPage4.ResumeLayout(false);
                        this.tabPage5.ResumeLayout(false);
                        this.tabPage2.ResumeLayout(false);
                        this.tabPage3.ResumeLayout(false);
                        this.tabPage1.ResumeLayout(false);
                        this.panel2.ResumeLayout(false);
                        this.panel1.ResumeLayout(false);
                        this.tabControl1.ResumeLayout(false);
                        this.ResumeLayout(false);

                }
                #endregion

                [STAThread]
                static void Main()
                {
                        Application.Run(new Form1());
                }

                private void PopulateServer_MyFiles()
                {
                        listView1.Items.Clear();
                        string[ ] mCList = Directory.GetFiles(local_shared_dir);
                        Populate(mCList);
                }

                private void ShareClientFiles()
                {
                        for ( int i=0; i < clientcollection.Count;i++)
                        {
                                ClientInfo obj = (ClientInfo)clientcollection.GetAt(i);
                                string[ ] mCList = new String[4];
                                mCList[0] = obj.sharedfileName;
                                mCList[1] = obj.sharedfilesSize;
                                mCList[2] = obj.sharedfilesPath;
                                mCList[3] = obj.username;
                                listView1.Items.Add(new ListViewItem(mCList));
                        }
                        listView1.Refresh();
                }

                private void Populate(string[ ] mCList)
                {
                        for ( int i=0; i < mCList.Length; i++)
                        {
                                FileInfo fi = new FileInfo(mCList[i]);
                                string[ ] mDesc = new string[3];
```

```
                                mDesc[0] = fi.Name;
                                mDesc[1] = fi.Length.ToString();
                                mDesc[2] = fi.FullName;
                                listView1.Items.Add(new ListViewItem(mDesc));

                                ClientInfo obj = new ClientInfo() ;
                                obj.username = server;
                                obj.password = "";
                                obj.sharedfileName=mDesc[0];
                                obj.sharedfilesPath=mDesc[2];
                                obj.sharedfilesSize =mDesc[1];
                                clientcollection.AddClient(obj);
                        }
                }

                private void Form1_Load(object sender, System.EventArgs e)
                {
                        textBox3.Text = server = Dns.GetHostName();
                        textBox2.Text = local_shared_dir ;
                        this.Text += " [ " + server + " ]" ;
                        clientcollection = new ClientCollection() ;
                        tabControl1.SelectedIndex = 4 ;
                }

                private void Connect()
                {
                        if ( ! Directory.Exists(local_shared_dir) )
                                Directory.CreateDirectory(local_shared_dir);

                        try
                        {
        serversocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
                                serversocket.Blocking = true ;
                                IPHostEntry IPHost = Dns.Resolve(textBox1.Text);
                                string [ ]aliases = IPHost.Aliases;
                                IPAddress[ ] addr = IPHost.AddressList;
                                IPEndPoint ipepServer = new IPEndPoint(addr[0], 8090);
                                serversocket.Connect(ipepServer);
                                clientsock = serversocket ;
                                Thread MainThread= new Thread(new ThreadStart(listenclient));
                                MainThread.Start();
                                button1.Enabled = true ;
                                button2.Enabled = true ;
                                PopulateServer_MyFiles();
                        }
                        catch(SocketException se)
                        {
                                Console.WriteLine(se.Message);
                                AppendText(se.Message);
                        }
                        catch(Exception eee)
                        {
                                MessageBox.Show("Socket Connect Error.\n\n"+eee.Message
+"\nPossible Cause: Server Already running. Check the tasklist for running processes", "Startup Error" ,
MessageBoxButtons.OK , MessageBoxIcon.Error);
                                Application.Exit();
```

```
        }
}

void listenclient()
{
        Socket sock = clientsock ;
        string cmd = server ;
        byte[ ] sender = System.Text.Encoding.ASCII.GetBytes("CLIENT " + cmd) ;
        sock.Send(sender, sender.Length,0) ;
        try
        {
                while ( sock != null )
                {
                        cmd = "" ;
                        byte[ ] recs = new byte[32767];
                        int rcount = sock.Receive(recs,recs.Length,0) ;
                string clientmessage = System.Text.Encoding.ASCII.GetString(recs) ;
                        clientmessage = clientmessage.Substring(0,rcount);
                        string smk = clientmessage ;
                        cmdList = null ;
                        cmdList = clientmessage.Split(' ');
                        string execmd = cmdList[0];
                        AppendText("COMMAND==>" + execmd) ;
                        sender = null ;
                        sender = new Byte[32767];
                    Console.WriteLine("CLIENT COMMAND = " + execmd + "\r\n");
                        string parm1 = "";
                        if ( execmd == "SERVER" )
                        {
                                AppendText("Connected TO Server :" + cmdList[1]);
                                continue ;
                        }

                        if ( execmd == "GET" )
                        {
                                // GET <FileName> <FileSize>
                                for ( int i=1 ; i < cmdList.Length-1;i++)
                                        parm1 = parm1 + " " + cmdList[i];
                                parm1 = parm1.Trim();
                                FileInfo fi = new FileInfo(parm1);
                                if ( fi.Exists )
                                        cmd = "GETOK " ;
                                else
                                        cmd = "GETOK_FAILED " ;
                        sender = System.Text.Encoding.ASCII.GetBytes(cmd) ;
                                sock.Send(sender, sender.Length,0) ;
                                continue;
                        }

                        if ( execmd == "LISTING" )
                        {
                                PopulateList(clientmessage);
                                continue ;
                        }

                if ( execmd == "SERVER_LISTING" || execmd == "CLIENT_LISTING")
```

```
                {
                        sender = new Byte[32767];
                        cmd="LISTING \r\n";
                        parm1 = cmdList[1];
                        cmd = cmd + SearchDatabase(parm1);
                sender = System.Text.Encoding.ASCII.GetBytes(cmd) ;
                        sock.Send(sender, sender.Length,0) ;
                        continue ;
                }

                if ( execmd == "NOOP" )
                {
                        // do nothing
                        continue ;
                }

                if ( execmd == "DISCONNECT" )
                {
                        total_clients_connected++;
                        continue ;
                }


                if ( execmd == "FILEOK" )
                {
                        cmd = "NOOP ";
                sender = System.Text.Encoding.ASCII.GetBytes(cmd) ;
                        sock.Send(sender, sender.Length,0) ;
                        continue ;
                }

                if ( execmd == "GETOK" )
                {
        cmd = "BEGINSEND "  + shared_file_path + " " + shared_file_size ;
                        sender = new Byte[1024];
                        sender = Encoding.ASCII.GetBytes(cmd);
                        sock.Send(sender, sender.Length , 0 );
                        CLIENT_DOWNLOADING(sock);
                        continue ;
                }

                if ( execmd == "BEGINSEND" )
                {
                        //ClientDownloadingFromServer(cmdList , sock);
                        SERVER_DOWNLOADING(cmdList , sock);
                        continue ;
                }

                if ( execmd == "CLIENT_CLOSE" )
                {
                        break ;
                }
        }
}
catch(Exception Se)
{
```

```
                string s = Se.Message;
                Console.WriteLine(s);
        }
}

void AppendText(string mtxt)
{
        listBox1.Items.Add(mtxt);
}

private void button1_Click_1(object sender, System.EventArgs e)
{
        Socket sock = serversocket ;

        if ( SearchText.Text == "" ) return ;
        if ( sock == null )
        {
                MessageBox.Show("Client not connected" , "Connect Error" ,
                MessageBoxButtons.OK , MessageBoxIcon.Error);
                return ;
        }
        string cmd = "SERVER_LISTING " + SearchText.Text + " "  ;
        byte[ ] b = System.Text.Encoding.ASCII.GetBytes(cmd) ;
        sock.Send(b, b.Length,0) ;
}

private void button2_Click_2(object sender, System.EventArgs e)
{
        Socket sock = clientsock ;
        int nPos  = listView2.SelectedIndices.Count ;
        if (nPos <= 0 )
        {
                MessageBox.Show("Please select a file to download." , "Question" ,
                MessageBoxButtons.OK , MessageBoxIcon.Question );
                return ;

        }

        IEnumerator selCol = listView2.SelectedItems.GetEnumerator();
        selCol.MoveNext() ;
        ListViewItem lv = (ListViewItem)selCol.Current;
        shared_file_name = lv.SubItems[0].Text;
        shared_file_size = lv.SubItems[1].Text;
        shared_file_path = lv.SubItems[2].Text;

        // Send a Get to command .Server will reply GETOK if the file is avaialble
        string cmd = "GET " + shared_file_path + " " + shared_file_size;
        Byte[ ] sb = new Byte[1024];
        sb = Encoding.ASCII.GetBytes(cmd);
        sock.Send(sb , sb.Length , 0 );
}

void PopulateList(string servermessage)
{
        listView2.Items.Clear() ;
        string[ ] lines = servermessage.Split('\n');
```

```
                    for ( int i=1; i < lines.Length-1;i++)
                    {
                        lines[i] =lines[i].Substring(0,lines[i].Length-1);
                        string[ ] listviewitems = lines[i].Split(',');
                        listView2.Items.Add(new ListViewItem(listviewitems));
                    }
                }

                private void tabControl1_SelectedIndexChanged(object sender, System.EventArgs e)
                {
                    int pos = tabControl1.SelectedIndex;
                }

                private void button3_Click(object sender, System.EventArgs e)
                {
                    if ( ! Directory.Exists(textBox2.Text) )
                    {
                        MessageBox.Show("Directory does not exist" , "Folder Error" ,
MessageBoxButtons.OK , MessageBoxIcon.Error);
                        return ;
                    }

                    local_shared_dir = textBox2.Text ;
                    tabControl1.SelectedTab = tabPage2 ;
                    PopulateServer_MyFiles();
                }

                private void Disconnect()
                {
                    if ( clientsock != null )
                    {
                        if ( ! clientsock.Connected )
                            return ;
                    }
                    else
                        return ;

                    string cmd = "CLIENT_DISCONNECTING " ;
                    Byte[ ] sb = new Byte[1024];
                    sb = Encoding.ASCII.GetBytes(cmd);
                    clientsock.Send(sb , sb.Length , 0 );
                    clientsock = null;
                }

                private void Form1_Closing(object sender, System.ComponentModel.CancelEventArgs e)
                {
                }

                void CLIENT_DOWNLOADING(Socket s)
                {
                    int cnt = listView3.Items.Count;
                    Socket sock = s ;
                    string[ ] mDownloading = new String[5];

                    FileStream fout = new FileStream(local_shared_dir + "\\" + shared_file_name  ,
FileMode.Create, FileAccess.Write) ;
```

```
		NetworkStream nfs = new NetworkStream(sock) ;
		long size=int.Parse(shared_file_size) ;
		long rby=0 ;
		string login_client_machine="";

		mDownloading[0]=shared_file_name;
		mDownloading[1]=size.ToString();
		mDownloading[2]="0";
		mDownloading[3]="";
		mDownloading[4]=login_client_machine;

		listView3.Items.Add(new ListViewItem(mDownloading));

		try
		{
			while(rby < size)
			{
				byte[ ] buffer = new byte[1024] ;
				//Read from the Network Stream
				int i = nfs.Read(buffer,0,buffer.Length) ;
				fout.Write(buffer,0,(int)i) ;
				rby=rby+i ;

				int pc = (int)( ((double)rby/(double)size ) * 100.00) ;
				string perc = pc.ToString() + "%";
				listView3.Items[cnt].SubItems[3].Text = perc;
				listView3.Items[cnt].SubItems[2].Text = rby.ToString();

			}
			fout.Close() ;
			string cmd = "FILEOK" ;
			Byte[ ] sender = new Byte[1024];
			sender = new Byte[1024];
			sender = Encoding.ASCII.GetBytes(cmd);
			sock.Send(sender , sender.Length , 0 );
		}
		catch(Exception ed)
		{
		Console.WriteLine("A Exception occured in file transfer"+ed.ToString());
			MessageBox.Show(ed.Message);
		}
}

void SERVER_DOWNLOADING(string[ ] cmdList , Socket s)
{
		DFSCommandList = cmdList ;
		int cnt = listView4.Items.Count;
		string[ ] mUploading = new String[5];
		Socket sock =  s ;
		string parm1 = "";
		string parm2 = "";

		for ( int i=1 ; i < DFSCommandList.Length-1;i++)
			parm1 = parm1 + " " + DFSCommandList[i];
		parm1 = parm1.Trim();
		parm2 = DFSCommandList[DFSCommandList.Length-1];
```

```
                        try
                        {
                                FileInfo ftemp = new FileInfo(parm1);
                                long total=ftemp.Length;
                                long rdby=0 ;
                                int len=0 ;
                                string login_client_machine="";

                                mUploading[0]=parm1;
                                mUploading[1]=total.ToString();
                                mUploading[2]="0";
                                mUploading[3]="";
                                mUploading[4]=login_client_machine;

                                listView4.Items.Add(new ListViewItem(mUploading));

                                byte[ ] buffed = new byte[1024] ;
                        FileStream fin = new FileStream(parm1,FileMode.Open , FileAccess.Read) ;
                                NetworkStream nfs = new NetworkStream(sock) ;

                                while(rdby<total&&nfs.CanWrite)
                                {
                                        len =fin.Read(buffed,0,buffed.Length) ;
                                        nfs.Write(buffed, 0,len);
                                        rdby=rdby+len ;

                                        int pc = (int)( ((double)rdby/(double)total ) * 100.00) ;
                                        string perc = pc.ToString() + "%" ;
                                        listView4.Items[cnt].SubItems[3].Text = perc;
                                        listView4.Items[cnt].SubItems[2].Text = rdby.ToString();
                                }
                                fin.Close() ;
                        }
                        catch(Exception ed)
                        {
                                Console.WriteLine("A Exception occured in transfer"+ed.ToString()) ;
                                MessageBox.Show(ed.Message);
                        }
                }

                string SearchDatabase(string pattern)
        {
                        string retCmd ="";
                        ArrayList arr = clientcollection.SearchFiles(pattern);
                        for ( int i=0; i < arr.Count ;i++)
                        {
                                ClientInfo obj = (ClientInfo)arr[i];
                                retCmd  = retCmd  + obj.sharedfileName + "," + obj.sharedfilesSize +
"," + obj.sharedfilesPath + "," + obj.username + "\r\n";
                        }
                        return retCmd ;
        }

                private void menuItem2_Click(object sender, System.EventArgs e)
                {
```

```
                Connect();
        }

        private void menuItem3_Click(object sender, System.EventArgs e)
        {
                Disconnect();
        }

        private void menuItem4_Click(object sender, System.EventArgs e)
        {
                Application.Exit() ;
        }
    }
}
```

Server

Form1.cs

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Net;
using System.Net.Sockets;
using System.Threading;
using System.Text;
using System.IO ;
using System.Text.RegularExpressions ;

namespace SERVER
{
    public class Form1 : System.Windows.Forms.Form
    {
        private System.ComponentModel.Container components = null;
        string server = "";
        ClientCollection clientcollection =null;

        string[ ] DFSCommandList ;
        Socket serversocket;
        private System.Windows.Forms.Label label1;
        private string login_client_machine;
        Socket clientsock = null ;
        string[ ] cmdList = null ;
        private string shared_file_name ;
        private string shared_file_path ;
        private string shared_file_size ;
        private System.Windows.Forms.TabControl tabControl1;
        private System.Windows.Forms.TabPage tabPage1;
        private System.Windows.Forms.TabPage tabPage2;
        private System.Windows.Forms.ListView listView1;
        private System.Windows.Forms.ColumnHeader SFILENAME;
        private System.Windows.Forms.ColumnHeader SFILESIZE;
```

```
private System.Windows.Forms.ColumnHeader SFILEPATH;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.TextBox textBox2;
private System.Windows.Forms.Panel panel1;
private System.Windows.Forms.Button button1;
private System.Windows.Forms.Button button2;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.TextBox textBox3;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.TextBox textBox4;
private string local_shared_dir = "C:\\" ;
private System.Windows.Forms.ListView listView3;
private System.Windows.Forms.ColumnHeader STFILENAME;
private System.Windows.Forms.ColumnHeader STFILESIZE;
private System.Windows.Forms.ColumnHeader STCLIENT;
private System.Windows.Forms.TextBox SearchText;
private System.Windows.Forms.Button button3;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.ListView listView4;
private System.Windows.Forms.ColumnHeader UPFILENAME;
private System.Windows.Forms.ColumnHeader UPFILESIZE;
private System.Windows.Forms.ColumnHeader UPCURRENTSIZE;
private System.Windows.Forms.ColumnHeader UPCLIENT;
private System.Windows.Forms.ColumnHeader ST_BYTES_COPIED;
private System.Windows.Forms.ColumnHeader ST_PERCENT;
private System.Windows.Forms.ColumnHeader UPPERCENT;
private System.Windows.Forms.TabPage tabPage4;
private System.Windows.Forms.TabPage tabPage3;
private System.Windows.Forms.TabPage tabPage5;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.TextBox IP_ADDRESS;
private System.Windows.Forms.Panel panel2;
private System.Windows.Forms.ListView listView2;
private System.Windows.Forms.ColumnHeader SCFILENAME;
private System.Windows.Forms.ColumnHeader SCFILESIZE;
private System.Windows.Forms.ColumnHeader SCFILEPATH;
private System.Windows.Forms.ColumnHeader SCUSER;
private System.Windows.Forms.Splitter splitter1;
private System.Windows.Forms.RichTextBox textBox1;
private int total_clients_connected =0;

public Form1()
{
        InitializeComponent();

        textBox3.Text = server = Dns.GetHostName();
        textBox2.Text = local_shared_dir ;
        this.Text += " [ " + server + " ]" ;
        clientcollection = new ClientCollection() ;
        if ( ! Directory.Exists(local_shared_dir) )
                Directory.CreateDirectory(local_shared_dir);
}

protected override void Dispose( bool disposing )
{
        serversocket.Close();
```

```
            if( disposing )
            {
                        if (components != null)
                        {
                                    components.Dispose();
                        }
            }
            base.Dispose( disposing );
}

#region Windows Form Designer generated code
private void InitializeComponent()
{
            this.panel1 = new System.Windows.Forms.Panel();
            this.label2 = new System.Windows.Forms.Label();
            this.SearchText = new System.Windows.Forms.TextBox();
            this.button1 = new System.Windows.Forms.Button();
            this.button2 = new System.Windows.Forms.Button();
            this.ST_PERCENT = new System.Windows.Forms.ColumnHeader();
            this.STFILESIZE = new System.Windows.Forms.ColumnHeader();
            this.STCLIENT = new System.Windows.Forms.ColumnHeader();
            this.UPCURRENTSIZE = new System.Windows.Forms.ColumnHeader();
            this.textBox2 = new System.Windows.Forms.TextBox();
            this.textBox3 = new System.Windows.Forms.TextBox();
            this.tabPage4 = new System.Windows.Forms.TabPage();
            this.listView3 = new System.Windows.Forms.ListView();
            this.STFILENAME = new System.Windows.Forms.ColumnHeader();
            this.ST_BYTES_COPIED = new System.Windows.Forms.ColumnHeader();
            this.tabPage5 = new System.Windows.Forms.TabPage();
            this.label6 = new System.Windows.Forms.Label();
            this.IP_ADDRESS = new System.Windows.Forms.TextBox();
            this.button3 = new System.Windows.Forms.Button();
            this.label4 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.textBox4 = new System.Windows.Forms.TextBox();
            this.tabPage2 = new System.Windows.Forms.TabPage();
            this.listView1 = new System.Windows.Forms.ListView();
            this.SFILENAME = new System.Windows.Forms.ColumnHeader();
            this.SFILESIZE = new System.Windows.Forms.ColumnHeader();
            this.SFILEPATH = new System.Windows.Forms.ColumnHeader();
            this.tabPage3 = new System.Windows.Forms.TabPage();
            this.listView4 = new System.Windows.Forms.ListView();
            this.UPFILENAME = new System.Windows.Forms.ColumnHeader();
            this.UPFILESIZE = new System.Windows.Forms.ColumnHeader();
            this.UPPERCENT = new System.Windows.Forms.ColumnHeader();
            this.UPCLIENT = new System.Windows.Forms.ColumnHeader();
            this.tabPage1 = new System.Windows.Forms.TabPage();
            this.panel2 = new System.Windows.Forms.Panel();
            this.listView2 = new System.Windows.Forms.ListView();
            this.SCFILENAME = new System.Windows.Forms.ColumnHeader();
            this.SCFILESIZE = new System.Windows.Forms.ColumnHeader();
            this.SCFILEPATH = new System.Windows.Forms.ColumnHeader();
            this.SCUSER = new System.Windows.Forms.ColumnHeader();
            this.label1 = new System.Windows.Forms.Label();
            this.tabControl1 = new System.Windows.Forms.TabControl();
```

```
            this.splitter1 = new System.Windows.Forms.Splitter();
            this.textBox1 = new System.Windows.Forms.RichTextBox();
            this.panel1.SuspendLayout();
            this.tabPage4.SuspendLayout();
            this.tabPage5.SuspendLayout();
            this.tabPage2.SuspendLayout();
            this.tabPage3.SuspendLayout();
            this.tabPage1.SuspendLayout();
            this.panel2.SuspendLayout();
            this.tabControl1.SuspendLayout();
            this.SuspendLayout();

            this.panel1.Controls.Add(this.label2);
            this.panel1.Controls.Add(this.SearchText);
            this.panel1.Controls.Add(this.button1);
            this.panel1.Controls.Add(this.button2);
            this.panel1.Dock = System.Windows.Forms.DockStyle.Top;
            this.panel1.Location = new System.Drawing.Point(0, 0);
            this.panel1.Name = "panel1";
            this.panel1.Size = new System.Drawing.Size(752, 40);
            this.panel1.TabIndex = 8;

            this.label2.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
            this.label2.Dock = System.Windows.Forms.DockStyle.Right;
            this.label2.Font = new System.Drawing.Font("Comic Sans MS", 9.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
            this.label2.ForeColor = System.Drawing.Color.Blue;
            this.label2.Location = new System.Drawing.Point(528, 0);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(224, 40);
            this.label2.TabIndex = 9;
            this.label2.Text = "label2";
            this.label2.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;

            this.SearchText.Location = new System.Drawing.Point(72, 8);
            this.SearchText.Name = "SearchText";
            this.SearchText.Size = new System.Drawing.Size(216, 20);
            this.SearchText.TabIndex = 8;
            this.SearchText.Text = "";

            this.button1.Location = new System.Drawing.Point(8, 8);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(56, 23);
            this.button1.TabIndex = 7;
            this.button1.Text = "Search";
            this.button1.Click += new System.EventHandler(this.button1_Click_1);

            this.button2.Location = new System.Drawing.Point(296, 8);
            this.button2.Name = "button2";
            this.button2.Size = new System.Drawing.Size(72, 23);
            this.button2.TabIndex = 7;
            this.button2.Text = "Download";
            this.button2.Click += new System.EventHandler(this.button2_Click_2);

            this.ST_PERCENT.Text = "Percent";
```

```
                this.STFILESIZE.Text = "FileSize";
                this.STFILESIZE.Width = 100;

                this.STCLIENT.Text = "ClientName";
                this.STCLIENT.Width = 100;

                this.UPCURRENTSIZE.Text = "Completed Size";
                this.UPCURRENTSIZE.Width = 150;

                this.textBox2.Location = new System.Drawing.Point(152, 32);
                this.textBox2.Name = "textBox2";
                this.textBox2.Size = new System.Drawing.Size(272, 20);
                this.textBox2.TabIndex = 1;
                this.textBox2.Text = "textBox2";
            this.textBox2.TextChanged += new System.EventHandler(this.textBox2_TextChanged);

                this.textBox3.Location = new System.Drawing.Point(152, 64);
                this.textBox3.Name = "textBox3";
                this.textBox3.ReadOnly = true;
                this.textBox3.Size = new System.Drawing.Size(272, 20);
                this.textBox3.TabIndex = 3;
                this.textBox3.Text = "textBox3";

                this.tabPage4.Controls.Add(this.listView3);
                this.tabPage4.Location = new System.Drawing.Point(4, 25);
                this.tabPage4.Name = "tabPage4";
                this.tabPage4.Size = new System.Drawing.Size(752, 379);
                this.tabPage4.TabIndex = 3;
                this.tabPage4.Text = "Downloads";

                this.listView3.Columns.AddRange(new
System.Windows.Forms.ColumnHeader[ ] {
        this.STFILENAME,
        this.STFILESIZE,
        this.ST_BYTES_COPIED,
        this.ST_PERCENT,
        this.STCLIENT});
                this.listView3.Dock = System.Windows.Forms.DockStyle.Fill;
                this.listView3.FullRowSelect = true;
                this.listView3.Location = new System.Drawing.Point(0, 0);
                this.listView3.Name = "listView3";
                this.listView3.Size = new System.Drawing.Size(752, 379);
                this.listView3.TabIndex = 0;
                this.listView3.View = System.Windows.Forms.View.Details;

                this.STFILENAME.Text = "FileName";
                this.STFILENAME.Width = 150;

                this.ST_BYTES_COPIED.Text = "Bytes Copied";
                this.ST_BYTES_COPIED.Width = 100;

                this.tabPage5.Controls.Add(this.label6);
                this.tabPage5.Controls.Add(this.IP_ADDRESS);
                this.tabPage5.Controls.Add(this.button3);
                this.tabPage5.Controls.Add(this.textBox3);
                this.tabPage5.Controls.Add(this.label4);
```

```
this.tabPage5.Controls.Add(this.textBox2);
this.tabPage5.Controls.Add(this.label3);
this.tabPage5.Controls.Add(this.label5);
this.tabPage5.Controls.Add(this.textBox4);
this.tabPage5.Location = new System.Drawing.Point(4, 25);
this.tabPage5.Name = "tabPage5";
this.tabPage5.Size = new System.Drawing.Size(752, 379);
this.tabPage5.TabIndex = 2;
this.tabPage5.Text = "Options";

this.label6.Location = new System.Drawing.Point(40, 128);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(100, 16);
this.label6.TabIndex = 5;
this.label6.Text = "Server IP Address";
this.label6.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.IP_ADDRESS.Location = new System.Drawing.Point(152, 128);
this.IP_ADDRESS.Name = "IP_ADDRESS";
this.IP_ADDRESS.ReadOnly = true;
this.IP_ADDRESS.Size = new System.Drawing.Size(136, 20);
this.IP_ADDRESS.TabIndex = 6;
this.IP_ADDRESS.Text = "";

this.button3.Location = new System.Drawing.Point(432, 32);
this.button3.Name = "button3";
this.button3.Size = new System.Drawing.Size(64, 23);
this.button3.TabIndex = 4;
this.button3.Text = "Apply";
this.button3.Click += new System.EventHandler(this.button3_Click);

this.label4.Location = new System.Drawing.Point(40, 64);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(100, 16);
this.label4.TabIndex = 2;
this.label4.Text = "Server Name";
this.label4.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.label3.Location = new System.Drawing.Point(32, 32);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(120, 16);
this.label3.TabIndex = 0;
this.label3.Text = "Shared Folder Name";
this.label3.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.label5.Location = new System.Drawing.Point(40, 96);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(100, 16);
this.label5.TabIndex = 2;
this.label5.Text = "Port";
this.label5.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.textBox4.Location = new System.Drawing.Point(152, 96);
this.textBox4.Name = "textBox4";
this.textBox4.ReadOnly = true;
this.textBox4.Size = new System.Drawing.Size(32, 20);
```

```
                        this.textBox4.TabIndex = 3;
                        this.textBox4.Text = "8090";

                        this.tabPage2.Controls.Add(this.listView1);
                        this.tabPage2.Location = new System.Drawing.Point(4, 25);
                        this.tabPage2.Name = "tabPage2";
                        this.tabPage2.Size = new System.Drawing.Size(752, 379);
                        this.tabPage2.TabIndex = 1;
                        this.tabPage2.Text = "Server Shared Files";

                        this.listView1.Columns.AddRange(new
System.Windows.Forms.ColumnHeader[ ] {
        this.SFILENAME,
        this.SFILESIZE,
        this.SFILEPATH});
                        this.listView1.Dock = System.Windows.Forms.DockStyle.Fill;
                        this.listView1.FullRowSelect = true;
                        this.listView1.Location = new System.Drawing.Point(0, 0);
                        this.listView1.Name = "listView1";
                        this.listView1.Size = new System.Drawing.Size(752, 379);
                        this.listView1.TabIndex = 4;
                        this.listView1.View = System.Windows.Forms.View.Details;

                        this.SFILENAME.Text = "FileName";
                        this.SFILENAME.Width = 250;

                        this.SFILESIZE.Text = "File Size";
                        this.SFILESIZE.Width = 100;

                        this.SFILEPATH.Text = "FilePath";
                        this.SFILEPATH.Width = 350;

                        this.tabPage3.Controls.Add(this.listView4);
                        this.tabPage3.Location = new System.Drawing.Point(4, 25);
                        this.tabPage3.Name = "tabPage3";
                        this.tabPage3.Size = new System.Drawing.Size(752, 379);
                        this.tabPage3.TabIndex = 4;
                        this.tabPage3.Text = "Upload";

                        this.listView4.Columns.AddRange(new
System.Windows.Forms.ColumnHeader[ ] {
        this.UPFILENAME,
        this.UPFILESIZE,
        this.UPCURRENTSIZE,
        this.UPPERCENT,
        this.UPCLIENT});
                        this.listView4.Dock = System.Windows.Forms.DockStyle.Fill;
                        this.listView4.FullRowSelect = true;
                        this.listView4.Location = new System.Drawing.Point(0, 0);
                        this.listView4.MultiSelect = false;
                        this.listView4.Name = "listView4";
                        this.listView4.Size = new System.Drawing.Size(752, 379);
                        this.listView4.TabIndex = 0;
                        this.listView4.View = System.Windows.Forms.View.Details;

                        this.UPFILENAME.Text = "FileName";
```

```
                this.UPFILENAME.Width = 150;

                this.UPFILESIZE.Text = "FileSize";
                this.UPFILESIZE.Width = 150;

                this.UPPERCENT.Text = "Percent";

                this.UPCLIENT.Text = "Client Uploading";
                this.UPCLIENT.Width = 200;

                this.tabPage1.Controls.Add(this.panel2);
                this.tabPage1.Controls.Add(this.panel1);
                this.tabPage1.Location = new System.Drawing.Point(4, 25);
                this.tabPage1.Name = "tabPage1";
                this.tabPage1.Size = new System.Drawing.Size(752, 379);
                this.tabPage1.TabIndex = 0;
                this.tabPage1.Text = "Search";

                this.panel2.Controls.Add(this.listView2);
                this.panel2.Dock = System.Windows.Forms.DockStyle.Fill;
                this.panel2.Location = new System.Drawing.Point(0, 40);
                this.panel2.Name = "panel2";
                this.panel2.Size = new System.Drawing.Size(752, 339);
                this.panel2.TabIndex = 11;

                this.listView2.Columns.AddRange(new
System.Windows.Forms.ColumnHeader[ ] {
        this.SCFILENAME,
        this.SCFILESIZE,
        this.SCFILEPATH,
                                                                this.SCUSER});
                this.listView2.Dock = System.Windows.Forms.DockStyle.Fill;
                this.listView2.FullRowSelect = true;
                this.listView2.Location = new System.Drawing.Point(0, 0);
                this.listView2.Name = "listView2";
                this.listView2.Size = new System.Drawing.Size(752, 339);
                this.listView2.TabIndex = 10;
                this.listView2.View = System.Windows.Forms.View.Details;
                this.SCFILENAME.Text = "FileName";
                this.SCFILENAME.Width = 250;
                this.SCFILESIZE.Text = "FileSize";
                this.SCFILESIZE.Width = 100;
                this.SCFILEPATH.Text = "FilePath";
                this.SCFILEPATH.Width = 250;
                this.SCUSER.Text = "User";
                this.SCUSER.Width = 100;
                this.label1.BackColor =
System.Drawing.Color.FromArgb(((System.Byte)(224)), ((System.Byte)(224)), ((System.Byte)(224)));
                this.label1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
                this.label1.Dock = System.Windows.Forms.DockStyle.Top;
                this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
                this.label1.Location = new System.Drawing.Point(0, 0);
                this.label1.Name = "label1";
                this.label1.Size = new System.Drawing.Size(760, 24);
                this.label1.TabIndex = 2;
```

```
                this.label1.Text = "FILE SHARE SERVER";
                this.label1.TextAlign = System.Drawing.ContentAlignment.TopCenter;

            this.tabControl1.Appearance = System.Windows.Forms.TabAppearance.FlatButtons;
                this.tabControl1.Controls.Add(this.tabPage1);
                this.tabControl1.Controls.Add(this.tabPage2);
                this.tabControl1.Controls.Add(this.tabPage3);
                this.tabControl1.Controls.Add(this.tabPage4);
                this.tabControl1.Controls.Add(this.tabPage5);
                this.tabControl1.Dock = System.Windows.Forms.DockStyle.Top;
                this.tabControl1.Location = new System.Drawing.Point(0, 24);
                this.tabControl1.Name = "tabControl1";
                this.tabControl1.SelectedIndex = 0;
                this.tabControl1.Size = new System.Drawing.Size(760, 408);
                this.tabControl1.TabIndex = 8;
                this.tabControl1.SelectedIndexChanged += new
System.EventHandler(this.tabControl1_SelectedIndexChanged);

                this.splitter1.Dock = System.Windows.Forms.DockStyle.Top;
                this.splitter1.Location = new System.Drawing.Point(0, 432);
                this.splitter1.Name = "splitter1";
                this.splitter1.Size = new System.Drawing.Size(760, 3);
                this.splitter1.TabIndex = 9;
                this.splitter1.TabStop = false;

                this.textBox1.Dock = System.Windows.Forms.DockStyle.Fill;
                this.textBox1.Location = new System.Drawing.Point(0, 435);
                this.textBox1.Name = "textBox1";
                this.textBox1.Size = new System.Drawing.Size(760, 98);
                this.textBox1.TabIndex = 11;
                this.textBox1.Text = "";

                this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
                this.ClientSize = new System.Drawing.Size(760, 533);
                this.Controls.Add(this.textBox1);
                this.Controls.Add(this.splitter1);
                this.Controls.Add(this.tabControl1);
                this.Controls.Add(this.label1);
                this.Name = "Form1";
                this.Text = "Server";
        this.Closing += new System.ComponentModel.CancelEventHandler(this.Form1_Closing);
                this.Load += new System.EventHandler(this.Form1_Load);
                this.panel1.ResumeLayout(false);
                this.tabPage4.ResumeLayout(false);
                this.tabPage5.ResumeLayout(false);
                this.tabPage2.ResumeLayout(false);
                this.tabPage3.ResumeLayout(false);
                this.tabPage1.ResumeLayout(false);
                this.panel2.ResumeLayout(false);
                this.tabControl1.ResumeLayout(false);
                this.ResumeLayout(false);

            }
            #endregion

            [STAThread]
```

```
static void Main()
{
	Application.Run(new Form1());
}

private void PopulateServer_MyFiles()
{
	listView1.Items.Clear();
	string[ ] mCList = Directory.GetFiles(local_shared_dir);
	Populate(mCList);
}

private void ShareClientFiles()
{
	for ( int i=0; i < clientcollection.Count;i++)
	{
		ClientInfo obj = (ClientInfo)clientcollection.GetAt(i);
		string[ ] mCList = new String[4];
		mCList[0] = obj.sharedfileName;
		mCList[1] = obj.sharedfilesSize;
		mCList[2] = obj.sharedfilesPath;
		mCList[3] = obj.username;
		listView1.Items.Add(new ListViewItem(mCList));
	}
	listView1.Refresh();
}

private void Populate(string[ ] mCList)
{
	for ( int i=0; i < mCList.Length; i++)
	{
		FileInfo fi = new FileInfo(mCList[i]);
		string[ ] mDesc = new string[3];
		mDesc[0] = fi.Name;
		mDesc[1] = fi.Length.ToString();
		mDesc[2] = fi.FullName;
		listView1.Items.Add(new ListViewItem(mDesc));

		ClientInfo obj = new ClientInfo() ;
		obj.username = server;
		obj.password = "";
		obj.sharedfileName=mDesc[0];
		obj.sharedfilesPath=mDesc[2];
		obj.sharedfilesSize =mDesc[1];
		clientcollection.AddClient(obj);
	}
}

private void Form1_Load(object sender, System.EventArgs e)
{
	PopulateServer_MyFiles();
	Thread th = new Thread(new ThreadStart(ListenForPeers));
	th.Start();
}

public void ListenForPeers()
```

```
		{
			try
			{
				serversocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
				serversocket.Blocking = true ;

				IPHostEntry IPHost = Dns.Resolve(server);
				string [ ]aliases = IPHost.Aliases;
				IPAddress[ ] addr = IPHost.AddressList;
				IP_ADDRESS.Text = addr[0].ToString();
				IPEndPoint ipepServer = new IPEndPoint(addr[0], 8090);
				serversocket.Bind(ipepServer);
				serversocket.Listen(-1);
				label2.Text = "Server " +server + " is running.." ;
				while ( true )
				{
					clientsock = serversocket.Accept();
					if ( clientsock.Connected )
					{
						total_clients_connected++;
						AppendText("Client connected...") ;
					Thread tc = new Thread(new ThreadStart(listenclient));
						tc.Start();
					}
				}
			}
			catch(SocketException se)
			{
				Console.WriteLine(se.Message);
			}
			catch(Exception eee)
			{
				MessageBox.Show("Socket Connect Error.\n\n"+eee.Message
+"\nPossible Cause: Server Already running. Check the tasklist for running processes", "Startup Error" ,
MessageBoxButtons.OK , MessageBoxIcon.Error);
				Application.Exit();
			}
		}

		void listenclient()
		{
			Socket sock = clientsock ;
			string cmd = server ;
			byte[ ] sender = System.Text.Encoding.ASCII.GetBytes("SERVER " + cmd) ;
			sock.Send(sender, sender.Length,0) ;
			try
			{
				while ( sock != null )
				{
					cmd = "" ;
					byte[ ] recs = new byte[32767];
					int rcount = sock.Receive(recs,recs.Length,0) ;
				string clientmessage = System.Text.Encoding.ASCII.GetString(recs) ;
					clientmessage = clientmessage.Substring(0,rcount);
```

```
                string smk = clientmessage ;

                cmdList = null ;
                cmdList = clientmessage.Split(' ');
                string execmd = cmdList[0];
                AppendText("COMMAND==>" + execmd) ;
                sender = null ;
                sender = new Byte[32767];

        Console.WriteLine("CLIENT COMMAND = " + execmd + "\r\n");
                string parm1 = "";
                if ( execmd == "USER" )
                {

                        login_client_machine = cmdList[2];
                        string pass = cmdList[3];
                        string ip = cmdList[1];
AppendText("Client Connected IP:" + ip + " User :" + login_client_machine );
                        continue ;
                }

                if ( execmd == "GET" )
                {
                        // GET <FileName> <FileSize>
                        for ( int i=1 ; i < cmdList.Length-1;i++)
                                parm1 = parm1 + " " + cmdList[i];
                        parm1 = parm1.Trim();
                        FileInfo fi = new FileInfo(parm1);

                        if ( fi.Exists )
                                cmd = "GETOK " ;
                        else
                                cmd = "GETOK_FAILED " ;
                sender = System.Text.Encoding.ASCII.GetBytes(cmd) ;
                        sock.Send(sender, sender.Length,0) ;
                        continue;
                }

                if ( execmd == "LISTING" )
                {
                        PopulateList(clientmessage);
                        continue ;
                }

                if ( execmd == "SERVER_LISTING" )
                {
                        sender = new Byte[32767];
                        cmd="LISTING \r\n";
                        parm1 = cmdList[1];
                        cmd = cmd + SearchDatabase(parm1);

                sender = System.Text.Encoding.ASCII.GetBytes(cmd) ;
                        sock.Send(sender, sender.Length,0) ;
                        continue ;
                }
```

```
                        if ( execmd == "NOOP" )
                        {
                                // do nothing
                                continue ;
                        }

                        if ( execmd == "CLIENT_DISCONNECTING" )
                        {
                                cmd = "CLIENT_CLOSE";
                        sender = System.Text.Encoding.ASCII.GetBytes(cmd) ;
                                sock.Send(sender, sender.Length,0) ;

                                sock.Close();
                                AppendText("Client Disconnected");
                                continue ;
                        }


                        if ( execmd == "FILEOK" )
                        {
                                cmd = "NOOP ";
                        sender = System.Text.Encoding.ASCII.GetBytes(cmd) ;
                                sock.Send(sender, sender.Length,0) ;
                                continue ;
                        }

                        if ( execmd == "GETOK" )
                        {
                    cmd = "BEGINSEND "  + shared_file_path + " " + shared_file_size ;
                                sender = new Byte[1024];
                                sender = Encoding.ASCII.GetBytes(cmd);
                                sock.Send(sender, sender.Length , 0 );
                                ServerDownloadingFromClient(sock);
                                continue ;
                        }

                        if ( execmd == "BEGINSEND" )
                        {
                                ClientDownloadingFromServer(cmdList , sock);
                                continue ;
                        }
                    }
                }
                catch(Exception Se)
                {
                        string s = Se.Message;
                        Console.WriteLine(s);
                }
        }

        void AppendText(string mtxt)
        {
                mtxt = mtxt+"\r\n" ;
                textBox1.AppendText(mtxt);
                textBox1.Select(textBox1.Text.Length,0);
        }
```

```
		private void button1_Click_1(object sender, System.EventArgs e)
		{
			Socket sock = clientsock ;

			if ( SearchText.Text == "" ) return ;
			if ( sock == null )
			{
				MessageBox.Show("Client not connected" , "Connect Error" ,
MessageBoxButtons.OK , MessageBoxIcon.Error);
				return ;
			}
			string cmd = "CLIENT_LISTING " + SearchText.Text + " "  ;
			byte[ ] b = System.Text.Encoding.ASCII.GetBytes(cmd) ;
			sock.Send(b, b.Length,0) ;
		}

		private void button2_Click_2(object sender, System.EventArgs e)
		{
			Socket sock = clientsock ;
			int nPos  = listView2.SelectedIndices.Count ;
			if (nPos <= 0 )
			{
				MessageBox.Show("Please select a file to download." , "Question" ,
MessageBoxButtons.OK , MessageBoxIcon.Question );
				return ;

			}

			IEnumerator selCol = listView2.SelectedItems.GetEnumerator();
			selCol.MoveNext() ;
			ListViewItem lv = (ListViewItem)selCol.Current;
			shared_file_name = lv.SubItems[0].Text;
			shared_file_size = lv.SubItems[1].Text;
			shared_file_path = lv.SubItems[2].Text;

			// Send a Get to command .Server will reply GETOK if the file is avaialble
			string cmd = "GET " + shared_file_path + " " + shared_file_size;
			Byte[ ] sb = new Byte[1024];
			sb = Encoding.ASCII.GetBytes(cmd);
			sock.Send(sb , sb.Length , 0 );
		}

		void PopulateList(string servermessage)
		{
			listView2.Items.Clear() ;
			string[ ] lines = servermessage.Split('\n');
			for ( int i=1; i < lines.Length-1;i++)
			{
				lines[i] =lines[i].Substring(0,lines[i].Length-1);
				string[ ] listviewitems = lines[i].Split(',');
				listView2.Items.Add(new ListViewItem(listviewitems));
			}
		}

		private void tabControl1_SelectedIndexChanged(object sender, System.EventArgs e)
```

```
		{
			int pos = tabControl1.SelectedIndex;
		}

		private void button3_Click(object sender, System.EventArgs e)
		{
			if ( ! Directory.Exists(textBox2.Text) )
			{
				MessageBox.Show("Directory does not exist" , "Folder Error" ,
MessageBoxButtons.OK , MessageBoxIcon.Error);
				return ;
			}

			local_shared_dir = textBox2.Text ;
			tabControl1.SelectedTab = tabPage2 ;
			PopulateServer_MyFiles();
		}

		private void Disconnect()
		{
			if ( clientsock != null )
			{
				if ( ! clientsock.Connected )
					return ;
			}
			else
				return ;

			string cmd = "SERVER_DISCONNECT " ;
			Byte[ ] sb = new Byte[1024];
			sb = Encoding.ASCII.GetBytes(cmd);
			clientsock.Send(sb , sb.Length , 0 );
			clientsock = null;
		}


		void ServerDownloadingFromClient(Socket s)
		{
			int cnt = listView3.Items.Count;
			Socket sock = s ;
			string[ ] mDownloading = new String[5];

			FileStream fout = new FileStream(local_shared_dir + "\\" + shared_file_name  ,
			FileMode.Create, FileAccess.Write) ;
			NetworkStream nfs = new NetworkStream(sock) ;
			long size=int.Parse(shared_file_size) ;
			long rby=0 ;

			mDownloading[0]=shared_file_name;
			mDownloading[1]=size.ToString();
			mDownloading[2]="0";
			mDownloading[3]="";
			mDownloading[4]=login_client_machine;

			listView3.Items.Add(new ListViewItem(mDownloading));
```

```
		try
		{
			//loop till the Full bytes have been read
			while(rby<size)
			{
				byte[ ] buffer = new byte[1024] ;
				//Read from the Network Stream
				int i = nfs.Read(buffer,0,buffer.Length) ;
				fout.Write(buffer,0,(int)i) ;
				rby=rby+i ;

				int pc = (int)( ((double)rby/(double)size ) * 100.00) ;
				string perc = pc.ToString() + "%";
				listView3.Items[cnt].SubItems[3].Text = perc;
				listView3.Items[cnt].SubItems[2].Text = rby.ToString();

			}
			fout.Close() ;
			string cmd = "FILEOK" ;
			Byte[ ] sender = new Byte[1024];
			sender = new Byte[1024];
			sender = Encoding.ASCII.GetBytes(cmd);
			sock.Send(sender , sender.Length , 0 );
		}
		catch(Exception ed)
		{
		Console.WriteLine("A Exception occured in file transfer"+ed.ToString());
		}
}

void ClientDownloadingFromServer(string[ ] cmdList , Socket s)
{
		DFSCommandList = cmdList ;
		int cnt = listView4.Items.Count;
		string[ ] mUploading = new String[5];
		Socket sock =  s ;
		string parm1 = "";
		string parm2 = "";

		for ( int i=1 ; i < DFSCommandList.Length-1;i++)
			parm1 = parm1 + " " + DFSCommandList[i];
		parm1 = parm1.Trim();
		parm2 = DFSCommandList[DFSCommandList.Length-1];
		try
		{
			FileInfo ftemp = new FileInfo(parm1);
			long total=ftemp.Length;
			long rdby=0 ;
			int len=0 ;

			mUploading[0]=parm1;
			mUploading[1]=total.ToString();
			mUploading[2]="0";
			mUploading[3]="";
			mUploading[4]=login_client_machine;
```

```
                                listView4.Items.Add(new ListViewItem(mUploading));

                                byte[ ] buffed = new byte[1024] ;
                        FileStream fin = new FileStream(parm1,FileMode.Open , FileAccess.Read) ;
                                NetworkStream nfs = new NetworkStream(sock) ;

                                while(rdby < total && nfs.CanWrite)
                                {
                                        len =fin.Read(buffed,0,buffed.Length) ;
                                        nfs.Write(buffed, 0,len);
                                        rdby=rdby+len ;

                                        int pc = (int)( ((double)rdby/(double)total ) * 100.00) ;
                                        string perc = pc.ToString() + "%" ;
                                        listView4.Items[cnt].SubItems[3].Text = perc;
                                        listView4.Items[cnt].SubItems[2].Text = rdby.ToString();
                                }
                                fin.Close() ;
                        }
                        catch(Exception ed)
                        {
                                Console.WriteLine("A Exception occured in transfer"+ed.ToString()) ;
                        }
                }

                string SearchDatabase(string pattern)
        {
                        string retCmd ="";
                        ArrayList arr = clientcollection.SearchFiles(pattern);
                        for ( int i=0; i < arr.Count ;i++)
                        {
                                ClientInfo obj = (ClientInfo)arr[i];
                                retCmd  = retCmd  + obj.sharedfileName + "," + obj.sharedfilesSize +
"," + obj.sharedfilesPath + "," + obj.username + "\r\n";
                        }
                        return retCmd ;
        }


    }
}
```