VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI
INSTITUTE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institute, NAAC Accredited with 'A' Grade
NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech Courses
Approved by AICTE, New Delhi, Affiliated to JNTUH
Recognized as "College with Potential for Excellence" by UGC
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyd – 500090,TS, India.
Telephone No: 040-2304 2758/59/60, Fax: 040-23042761
E-mail: postbox@vnrvjiet.ac.in, Website: www.vnrvjiet.ac.in

TAMASOMA JYOTIRGAMAYA

Estd.1995

Best Lab Oriented Course Based Projects for the Academic Year 2017-18 CSE

S.NO	H.T.NO	NAME OF STUDENT	PROJECT TITLE	GUIDE NAME	Subject	Year
1	14071A05A6 14071A0567 14071A05A7 14071A0590 15075A0515	Praneetha G Nuthan Kumar Priya Bhandari Venkatesh M S manish	Data mining for weather Prediction and climate change studies.	N V Sailaja	Data Mining & Data ware Housing	IV
2	14071A0508 15075A0508 14071A0507 15075A0510	B Srikanth B Madhava Reddy B Raju V Tirumalesh	Estimating the data using the training set and cross validation	Dr. Ch Suresh	Data Mining & Data ware Housing	IV
3	15071A05N5	P. Venkat	Shell Simulator in C	S.Kranthi Kumar	Operating System	III
4	15071A05J0 15071A05L5 15071A05K6 15071A05J7 15071A05I4	C Vishnuvardhan P Varun M Swaroop G Rohit A Rama Krishna	Creation of Chat Monitor with Multiple Connected Clients in a Network using Threads	S Kranthi Kumar	Operating System & Computer Networks	III
5	16075A0520 16075A0521 16075A0522 16075A0523 16075A0524	P.Ramakanth Ramesh Sainath Nagaraj Manoj Kumar	rulAR in Unity	P.Rama Krishna	Object Oriented Analysis And Design	III

6	15071A0585 15071A0591 15071A05A6 15071A05B1	K.Sai Chand Md.Sulaiman P.Lakshmipathy Sk Mujahed	**Implementation of Sub Netting Strategy on Physical Networks for Creation of Multiple Logical Networks**	Tejaswi Potluri	Computer Networks	III
7	15071A05I7 15071A05I8 15071A05K5 15071A05L2	B.Sravanthi B.Nisha K.Deepthi Devi N.Lahari	Implementation of SLR parser	M.Gangappa	Complier Design	III
8	16075A0501 16075A0502 16075A0503 16075A0504 16075A0505	Ch Pooja Ch Sunil P Harish B Prashanth Arshaque Md	Minimization Of DFA	Priya Bhatnagar	Complier Design	III
9	16071A0586 16071A0583	K.Sahithi K.Shylesh	Online Movie Ticket Booking System Reservation	A.Madhavi	Data Structures	II

HOD-CSE

Data Mining For Weather Prediction and Climate Change Studies

1	**14071A05A6** **14071A0567** **14071A05A7** **14071A0590** **15075A0515**	**Praneetha Gutta** **Nuthan Kumar** **Priya Bhandari** **Venkatesh** **M S Manish**	**Data mining for weather Prediction and climate Change studies.**	**N V Sailaja**

Name of the course : Data Mining and Data warehousing

Year / Semester : IV- B.Tech I-semester

Project Title : Weather Prediction and Climate Change Studies

Aim : Apply classification to predict the weather prediction and climate changes

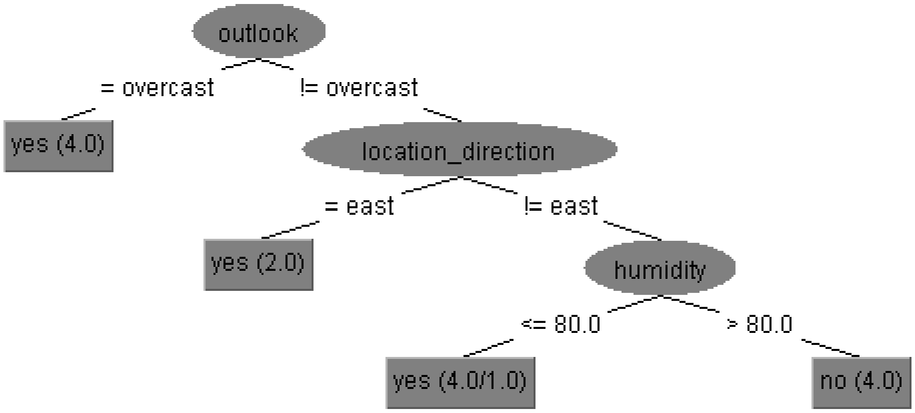
ABSTRACT:

The purpose of this project is to apply classification to predict the weather prediction and climate changes. Weather forecasting is a vital application in meteorology and has been one of the most scientifically and technologically challenging problems around the world in the last century. In this project, we investigate the use of data mining techniques in forecasting maximum temperature, rainfall, and wind speed by applying preprocessing steps and classification algorithms. We apply unsupervised filter for discretization of data as preprocessing step and then few classification algorithms. This was carried out using Naïve-bayes algorithms, Decision Tree algorithms and also some meteorological data collected. A data model for the meteorological data was developed and this was used to train the classifier algorithms. The performances of these algorithms were compared using standard performance metrics, and the algorithm which gave the best results was used to generate classification rules for the mean weather variables. The results show that given enough case data, Data Mining techniques can be used for weather forecasting and climate changes.

ALGORITHM:

1. Study of various pre-processing techniques for improving the quality of raw data
2. Use of un-supervised filter for discretization of data as preprocessing step
3. Understanding and exploring the various algorithms for classification technique
4. Analyzing the results of J48 and Naïve-bayes algorithm

RESULT:



LIMITATIONS:

It cannot predict the weather short term efficiently. We used only small limited areas for weather forecasting. Accurate weather prediction is a difficult task due to dynamic change of atmosphere. It is susceptible for predict weather in large areas at a time.

CONCLUSION:

Naive Bayes algorithm is based on probability and j48 algorithm is based on decision tree. From the result we see time to build the model is less when using Naive Bayes and correctly classified instances are more when using Naive Bayes and prediction accuracy is also greater in Naive Bayes than of J48. Hence it is concluded that Naïve Bayes perform better than of J48 on diabetes dataset.

Estimating the Data Using the Training Set and Cross Validation

2	**14071A0508** **15075A0508** **14071A0507** **15075A0510**	**Bhamidipati Srikanth** **B Madhava Reddy** **B Raju** **V Tirumalesh**	**Estimating the data using the training set and cross validation**	**Dr. Ch Suresh**	**IV**

Name of the course : Data Mining and Data warehousing

Year / Semester : IV- B.Tech I-semester

Project Title : Weather Prediction and Climate Change Studies

Aim : Classify data to find out the percentage of correct classified instances and also to find out which estimate is more realistic.

ABSTRACT:

Estimating the data using Training Set and Cross Validation is used to classify data to find out the percentage of correct classified instances and also to find out which estimate is more realistic. We experiment this on iris.arff dataset. We perform this classification using Weka tool which is an open source software issued under the GNU General Public License. Weka tool consists of 5 applications namely Explorer, Experimenter, KnowledgeFlow, Workbench, SimpleCLI. To perform the task, we first study the various pre-processing techniques for improving the quality of raw data. We use the unsupervised filter for discretization of data as pre-processing step. Now, we understand and explore the various algorithms for classification. Then, we perform the classification on the data and analyze the results.

CONCLUSION:

From the above classification, we observe the following

- ✓ Correctly classified instances using Training Set is 50(33.33%)
- ✓ Incorrectly classified instances using Training Set is 100(66.67%)
- ✓ Correctly classified instances using Cross Validation with 10 folds is 50(33.33%)
- ✓ Incorrectly classified instances using Cross Validation with 10 folds is 100(66.67%)
- ✓ Correctly classified instances using Cross Validation with 20 folds is 40(26.67%)
- ✓ Incorrectly classified instances using Cross Validation with 20 folds is 110(73.33%)

So, from the above observations we can conclude that Training Set is more realistic because correctly classified instances in Cross Validation change with respect to number of folds. But correctly a classified instance in Training Set is consistent for a dataset.

Shell Simulator in C Language

3	**15071A05N5**	**P. Venkat**	**Shell Simulator in C**	**S.Kranthi Kumar**	**Operating System**	**III**

Name of the course : Operating Systems

Year / Semester : III-B.Tech I-semester

Aim : To simulate a shell in C language.

ABSTRACT:

Every Linux distro has a built-in Terminal in it, which isn't an open source in many cases. In such case, the user of the distro needs to implement own version of shell for customization. This C program helps to achieve that requirement.

TECHNOLOGICAL STACK:

- Basic knowledge on Linux
- C language control statements, functions.

ALGORITHM:

1) The C program is written to be compatible with gcc.
2) The program runs infinitely until the exit command is encountered.
3) The inputs are scanned until EOF or NewLine.
4) If the command exists, then execution is bound to /dev/null so as to check if the command exists.
5) If it exists, the user is notified regarding the same.
6) If it exits, the command is executed by interacting with the Environment.

LIMITATIONS:

1. The program requires a base to execute until used as standalone program
2. The Shell is on a single process, but doesn't fork to child process.
3. Background process is a limitation as of now.
4. Limited commands are handled internally through C language.

Creation of Chat Monitor with Multiple Connected Clients in a Network using Threads

4	**15071A05J0 15071A05L5 15071A05K6 15071A05J7 15071A05I4**	**C Vishnu Vardhan Reddy P Varun M Swaroop G Rohit A Rama Krishna**	**Creation of Chat Monitor with Multiple Connected Clients in a Network using Threads**	**S Kranthi Kumar**	**III**

Name of the course : Operating Systems and Computer Networks

Year / Semester : III- B.Tech I-semester

Project Title : Multi-Threading Application in C.

Aim : Creation of Chat Monitor with Multiple Connected Clients in a N/W using Threads

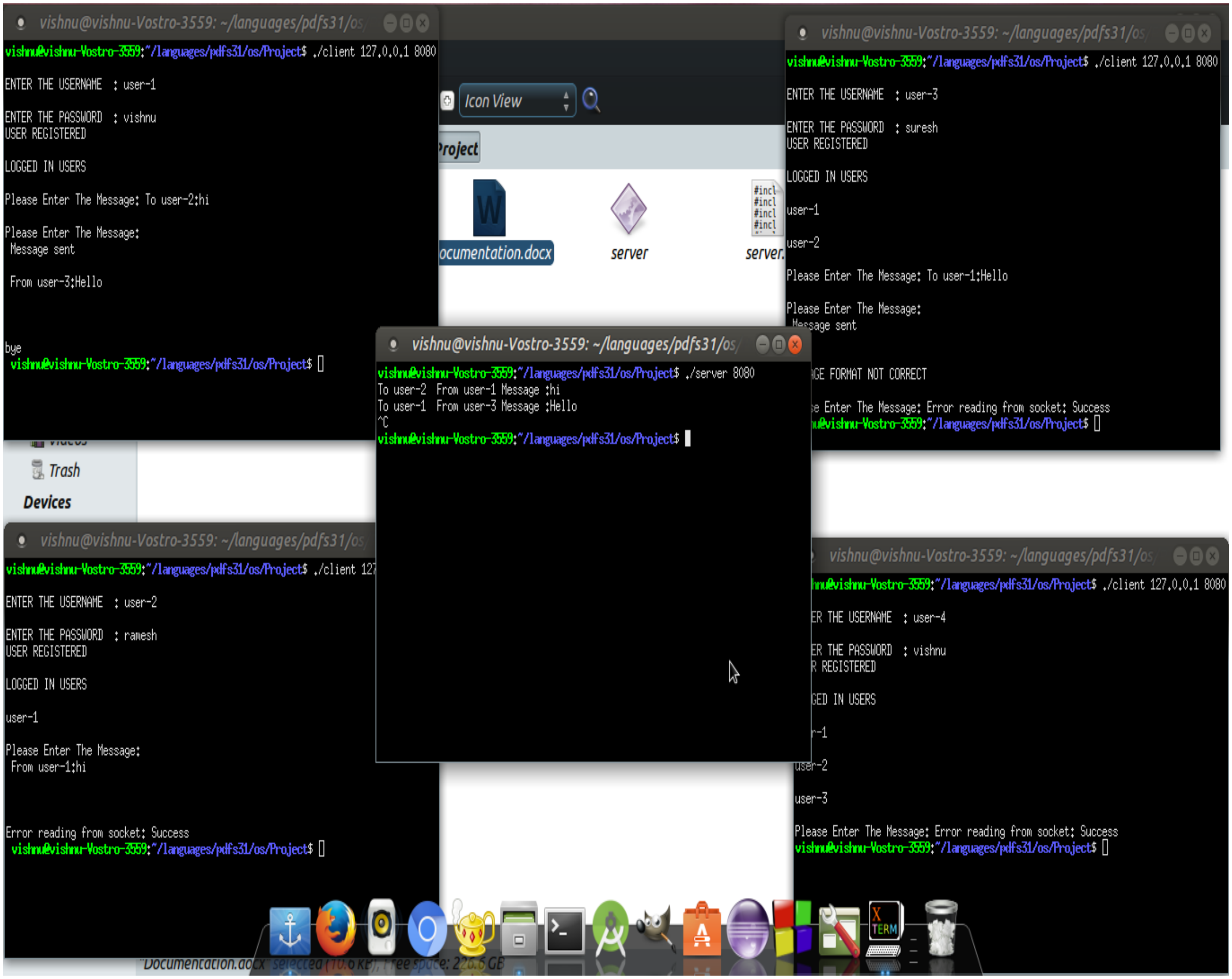
ABSTRACT:

This project deals the implementation of client program for the multi-threaded chat application. Till now all examples in socket programming assume that client first sends some information and then server or other clients responds to that information.

In real world, this might not be the case. It is not required to send someone a message in order to be able to receive one. A client should readily receive a message whenever it is delivered to it i.e. sending and receiving must be implemented as separate activities rather than sequential. There is a very simple solution which uses threads to achieve this functionality. In the client side implementation we will be creating two threads

OUTCOME OF THE PROJECT:

The Chat Server Monitor the Activities of Multiple Connected Clients. Clients connected to the server can send messages to any of the Connected Users in the Specified Format. A file able to store the Information of All the registered Users for every new Session. A Console I/O Bases Application Which Creates a Chat Room for the devices connected through a Common Network.



```
vishnu@vishnu-Vostro-3559:~/languages/pdfs31/os/Project$ ./client 127.0.0.1 8080

ENTER THE USERNAME  : user-1

ENTER THE PASSWORD  : vishnu
USER REGISTERED

LOGGED IN USERS

Please Enter The Message: To user-2:hi

Please Enter The Message:
 Message sent

 From user-3:Hello



bye
 vishnu@vishnu-Vostro-3559:~/languages/pdfs31/os/Project$ []
```

```
vishnu@vishnu-Vostro-3559:~/languages/pdfs31/os/Project$ ./client 127.0.0.1 8080

ENTER THE USERNAME  : user-3

ENTER THE PASSWORD  : suresh
USER REGISTERED

LOGGED IN USERS

user-1

user-2

Please Enter The Message: To user-1:Hello

Please Enter The Message:
 Message sent


AGE FORMAT NOT CORRECT

se Enter The Message: Error reading from socket: Success
nu@vishnu-Vostro-3559:~/languages/pdfs31/os/Project$ []
```

```
vishnu@vishnu-Vostro-3559:~/languages/pdfs31/os/Project$ ./server 8080
To user-2  From user-1 Message :hi
To user-1  From user-3 Message :Hello
^C
vishnu@vishnu-Vostro-3559:~/languages/pdfs31/os/Project$ █
```

```
vishnu@vishnu-Vostro-3559:~/languages/pdfs31/os/Project$ ./client 127

ENTER THE USERNAME  : user-2

ENTER THE PASSWORD  : ramesh
USER REGISTERED

LOGGED IN USERS

user-1

Please Enter The Message:
 From user-1:hi


Error reading from socket: Success
 vishnu@vishnu-Vostro-3559:~/languages/pdfs31/os/Project$ []
```

```
hnu@vishnu-Vostro-3559:~/languages/pdfs31/os/Project$ ./client 127.0.0.1 8080

ER THE USERNAME  : user-4

ER THE PASSWORD  : vishnu
R REGISTERED

GED IN USERS

r-1

user-2

user-3

Please Enter The Message: Error reading from socket: Success
vishnu@vishnu-Vostro-3559:~/languages/pdfs31/os/Project$ []
```

rulAR in Unity

5	**16075A0520 16075A0521 16075A0522 16075A0523 16075A0524**	**P.Ramakanth Ramesh Sainath Nagaraj Manoj Kumar**	**rulAR in Unity**	**P.Rama Krishna**	**III**

Name of the course : OBJECT ORIENTED ANALYSIS AND DESIGN

Year / Semester : III-B.Tech I-semester

Project Title : rulAR in Unity

Aim : create an integrated measurement system with the help of two or more image targets

ABSTRACT:

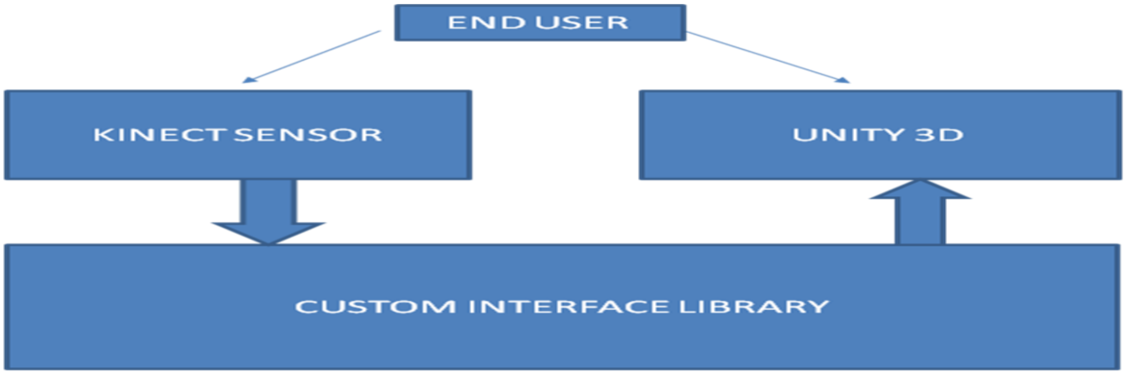
This project will create an integrated measurement system by a small app called " rulAR " with the help of two or more image targets we can check or get the complete measurements Like length, area, perimeter & shape of the object_as how image targets are placed . By using this application we can measure objects measurements with the help of a simple app & image targets. rulAr in Unity is setting the basis for the customer to be virtually immersed in a complex augmented reality, just as any other customer should carry all measuring accessories.

PRODUCT FUNCTIONALITY:

- Unity 3D, vuforia based rulAR app used to access the camera.
- The screens display the augmented reality as an integrated panoramic viewing window.
- The augmented environment's viewable segment will be based on the perspective of the user according to the user; s location.
- The user's object measurement is calculated based on the distances and angles in reference to the rulAR
- This rulAR are mounted on the screens.
- The system is scalable, meaning multiple rulAR can be added and the system will remain functional and stable.

USERS AND CHARACTERISTICS:

The primary users of this product are going to be academics in the fields of science. Anyone can use this application.



DATA FLOW DIAGRAM.....

User Interfaces:-

CONCLUSION:

The main intention of this project is to allow a user to realistically experience an visual measurements that all measuring devices could not otherwise be physically present in .Marias technologies wanted to simplify the process of tracking movements by Cameras and transferring these movements to an augmented reality, Simplifying the process entails decreasing the cost of the system as a whole. Our project provides the foundation for this significant benefit.

Implementation of Sub Netting Strategy on Physical Networks for Creation of Multiple Logical Networks

6	15071A0585 15071A0591 15071A05A6 15071A05B1	K.Sai Chand Md.Sulaiman P.Lakshmipathy Sheikh Mujahed	Implementation of Sub Netting Strategy on Physical Networks for Creation of Multiple Logical Networks	Tejaswi Potluri	III

Name of the course : Computer Networks

Year / Semester : III-B.Tech I-semester

Project Title : Implementation of Sub Netting Strategy on Physical Networks for Creation of Multiple Logical Networks

Aim : To analyze the output of sub netting and write C++ program .To obtain subnet mask for given IP Address with respective Classes.

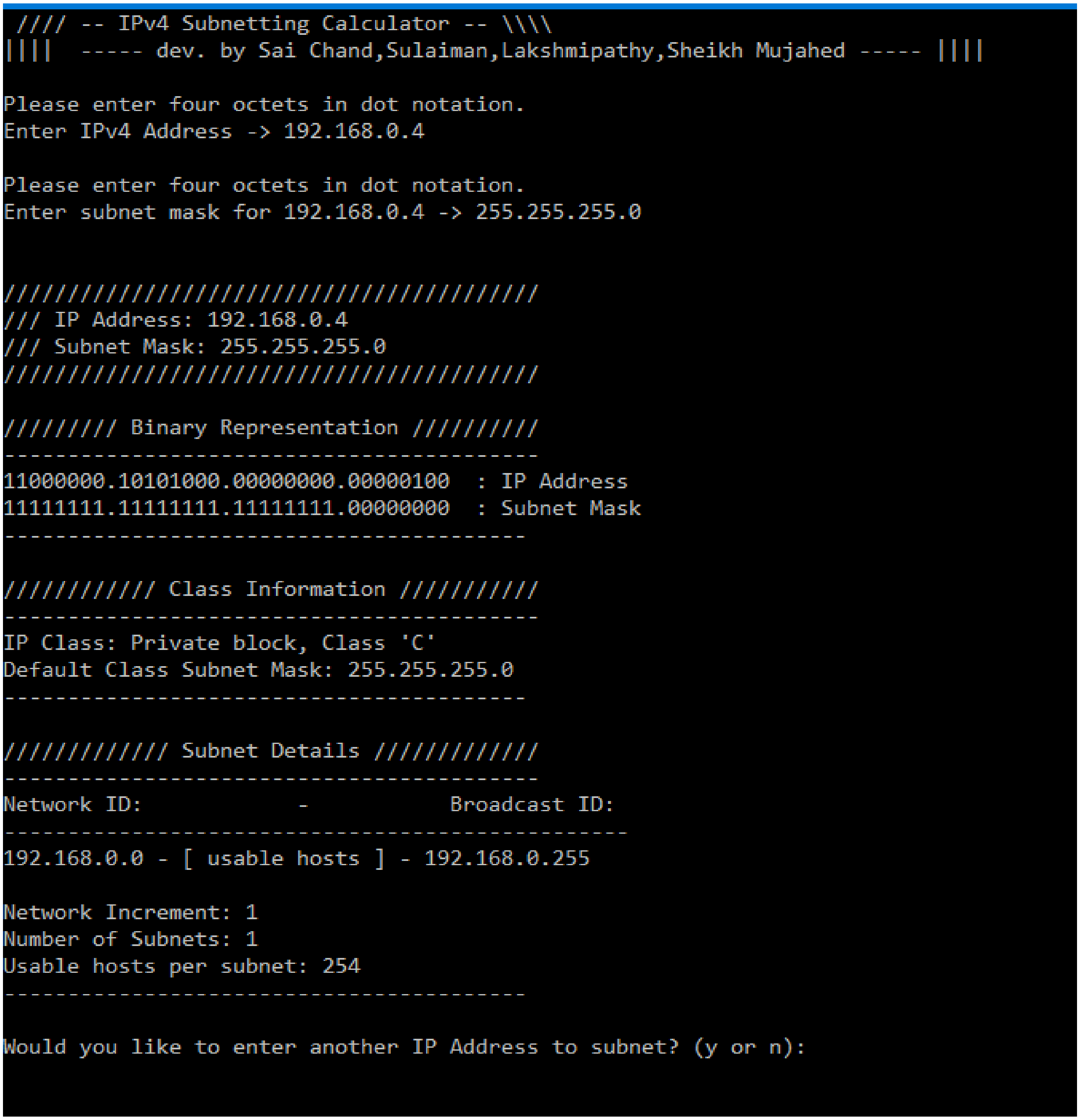
ABSTRACT:

Sub netting is the strategy used to partition a single physical network into more than one smaller logical sub-network (subnets). A subnet allows the flow of network traffic between hosts to be segregated based on a network configuration. By organizing hosts into logical groups, sub netting can improve network security and performance. An IP address includes a network segment and a host segment. Sub netting allows network administrators some flexibility in defining relationships among network hosts.

ALGORITHM:

1) Taking a IPv4 address and subnet mask to perform sub netting
2) Clear the octet mask vector
3) Determine and get IP Binary Notation and Subnet Representation
4) Perform ANDing of IP and Subnet mask to generate network ID range
5) Turn Binary back to decimal
6) Get the network increment, Network ID range
7) Get Subnets, hosts per subnet and Subnet masks and octets
8) Print Initial user IP and subnet mask
9) Print Binary Representation
10) Print IP Class and Subnetting Details.

SAMPLE OUTPUT:



```
 //// -- IPv4 Subnetting Calculator -- \\\\
||||  ----- dev. by Sai Chand,Sulaiman,Lakshmipathy,Sheikh Mujahed ----- ||||

Please enter four octets in dot notation.
Enter IPv4 Address -> 192.168.0.4

Please enter four octets in dot notation.
Enter subnet mask for 192.168.0.4 -> 255.255.255.0


//////////////////////////////////////////
/// IP Address: 192.168.0.4
/// Subnet Mask: 255.255.255.0
//////////////////////////////////////////

///////// Binary Representation //////////
------------------------------------------
11000000.10101000.00000000.00000100  : IP Address
11111111.11111111.11111111.00000000  : Subnet Mask
-----------------------------------------

//////////// Class Information ///////////
------------------------------------------
IP Class: Private block, Class 'C'
Default Class Subnet Mask: 255.255.255.0
-----------------------------------------

///////////// Subnet Details /////////////
------------------------------------------
Network ID:            -           Broadcast ID:
-------------------------------------------------
192.168.0.0 - [ usable hosts ] - 192.168.0.255

Network Increment: 1
Number of Subnets: 1
Usable hosts per subnet: 254
-----------------------------------------

Would you like to enter another IP Address to subnet? (y or n):
```

LIMITATIONS:

Not implemented for IP V6 address

CONCLUSION:

Sub netting implemented successfully.

Implementation of SLR parser

7	15071A05I7 15071A05I8 15071A05K5 15071A05L2	B.Sravanthi B.Nisha K.Deepthi Devi N.Lahari	Implementation of SLR parser	M.Gangappa	III

Name of the course : Compiler Design

Year / Semester : III-B.Tech I-semester

Project Title : **Implementation of SLR parser**

Aim : To design and customize C program that simulates SLR parser

ABSTRACT:

SLR and the more-general methods LALR and CLR have identical methods and similar tables at parse time; they differ only in the mathematical grammar analysis algorithms used by the parser generator tool. SLR and LALR generators create tables of identical size and identical parser states. SLR generators accept fewer grammars than do LALR generators like YACC and Bison. Many computer languages don't readily fit the restrictions of SLR, as is. Bending the language's natural grammar into SLR grammar form requires more compromises and grammar hackery. So LALR generators have become much more widely used than SLR generators, despite being somewhat more complicated tools. SLR methods remain a useful learning step in college classes on compiler theory.

PARSING ALGORITHM:

A grammar is said to be SLR(1) if the following simple LR parser algorithm results in no ambiguity.

1. If state *s* contains1 any item of the form $A \rightarrow a \bullet Xb$, where X is a terminal, and X is the next token in the input string, then the action is to shift the current input token onto the stack, and the new state to be pushed on the stack is the state containing the item $A \rightarrow aX \bullet b$.
2. If state *s* contains the complete item $A \rightarrow y \bullet$, and the next token in the input string is in *Follow(A)*, then the action is to reduce by the rule $A \rightarrow y$. A reduction by the rule $S' \rightarrow S$, where *S*is the start state, is equivalent to acceptance; this will happen only if the next input token is *$*. In all other cases, the new state in computed as follows. Remove the string *y* and all of its corresponding states from the parsing stack. Correspondingly, back up in the DFA to the state from which the construction of *y* began. By construction, this state must contain an item of the form $B \rightarrow a \bullet Ab$. Push *A* on to the stack, and push the state containing the item $B \rightarrow aA \bullet b$.
3. If the next input token is such that neither of the above two cases applies, an error is declared.

LIMITATIONS:

1. It does not parse all kinds of grammars.
2. It possess shift/reduce issues if a grammar is ambiguous

CONCLUSION:

Simple LR is a type of LR parser with small parse tables and a relatively simple parser generator algorithm. As with other types of LR(1) parser, an SLR parser is quite efficient at finding the single correct bottom-up in a single left-to-right scan over the input stream, without guesswork or backtracking. The parser is mechanically generated from a formal grammar for the language.

MINIMIZATION OF DFA

8	**16075A0501** **16075A0502** **16075A0503** **16075A0504** **16075A0505**	**Ch Pooja** **Ch Sunil** **P Harish** **B Prashanth** **Arshaque Md**	**Minimization Of DFA**	**Priya Bhatnagar**	**Complier Design**	**III**

Name of the course **:** Compiler Design

Year / Semester **:** III-B.Tech I-semester

Project Title **:** MINIMIZATION OF DFA

Aim **:** To design DFA with minimum states for the given regular expression

ABSTRACT:

In automata theory (a branch of computer science), DFA minimization is the task of transforming a given deterministic finite automaton (DFA) into an equivalent DFA that has a minimum number of states. Here, two DFAs are called equivalent if they recognize the same regular language.

FEATURES:

- Regex →NFA
- NFA → DFA
- DFA Minimization

ALGORITHMS:

1) Conversion from NFA to DFA

Suppose there is an NFA N < Q, ∑, q0, δ, F > which recognizes a language L. Then the DFA D < Q', ∑, q0, δ',F'> can be constructed for language L as :

Step 1: Initially Q' = ϕ.

Step 2: Add q0 to Q'.

Step 3: For each state in Q', find the possible set of states for each input symbol using transition function of NFA. If this set of states is not in Q', add it to Q'.

Step 4: Final state of DFA will be all states with contain F (final states of NFA)

2) Minimization of DFA

Suppose there is a DFA D < Q, ∑, q0, δ, F > which recognizes a language L. Then the minimized DFA D < Q’, ∑, q0, δ’, F’> can be constructed for language L as:

Step 1: We will divide Q (set of states) into two sets. One set will contain all final states and other set will contain non-final states. This partition is called P0.

Step 2: Initialize k = 1

Step 3: Find P_k by partitioning the different sets of P_{k-1}. In each set of P_{k-1}, we will take all possible pair of states. If two states of a set are distinguishable, we will split the sets into different sets in P_k.

Step 4: Stop when $P_k = P_{k-1}$ (No change in partition)

Step 5: All states of one set are merged into one. No. of states in minimized DFA will be equal to no. of sets in P_k.

LIMITATIONS:

There is no direct method to convert a regular expression into minimized DFA.

CONCLUSION:

DFA Minimization is a fairly understandable process, and is useful in several areas

- Regular expression matching implementation
- Very similar algorithm is used for compiler optimization to eliminate duplicate computations.

ONLINE MOVIE TICKET BOOKING SYSTEM RESERVATION

9	**16071A0586** **16071A0583**	**K.Sahithi** **K.Shylesh**	**Online Movie Ticket Booking System Reservation**	**A.Madhavi**	**Data Structures**	**II**

Name of the course : ADVANCED DATA STRUCTURES THROUGH C++

Year / Semester : II-B.Tech I-semester

Project Title : ONLINE MOVIE TICKET BOOKING SYSTEM RESERVATION

Aim : The project is aimed at developing an online ticket reservation system for a cinema hall.

ABSTRACT:

This online ticket reservation system provides a website for a cinema hall where any user of internet can access it. Watching movies with family and friends in theatre is one of the best medium of entertainment. But all this excitement vanishes after standing in hours in long queue to get tickets booked. This project helps to provide complete information regarding currently running movies with details of show timings, available seats and charger.

ALGORITHM:

1) Considering a structure to denote the details required like movie, show timings and seats.
2) The details are inserted with the help of FILE and the movie details are stored in a file.
3) The required movie can be found by entering the movie code.
4) After giving the required details the movie tickets are booked and required transactions are made.
5) After successful transaction print allocated seats, movie and show timings.

LIMITATIONS:

1) We are using files to store information of movie details instead of a database.
2) It was developed using C++, so the performance is poor as compared to the real time online systems.

CONCLUSION:

The functions like booking, cancelling etc. of the movie booking system online has been developed successfully and the performance of this system has been found satisfactory.