A PROJECT REPORT
ON

"PERSONAL TRACKER"

Submitted to
SnT COUNCIL, IIT KANPUR

BY

AYUSHI KUSHWAHA	**160184**
RAGHAV GARG	**160535**
SAGAR CHAUDHARY	**160603**
SHIKHAR MAHAJAN	**150669**

UNDER THE GUIDANCE OF
ROBOTICS CLUB

WINTER SCHOOL, SnT COUNCIL

DECEMBER 2016

IIT KANPUR

ARDUINO CIRCUITS

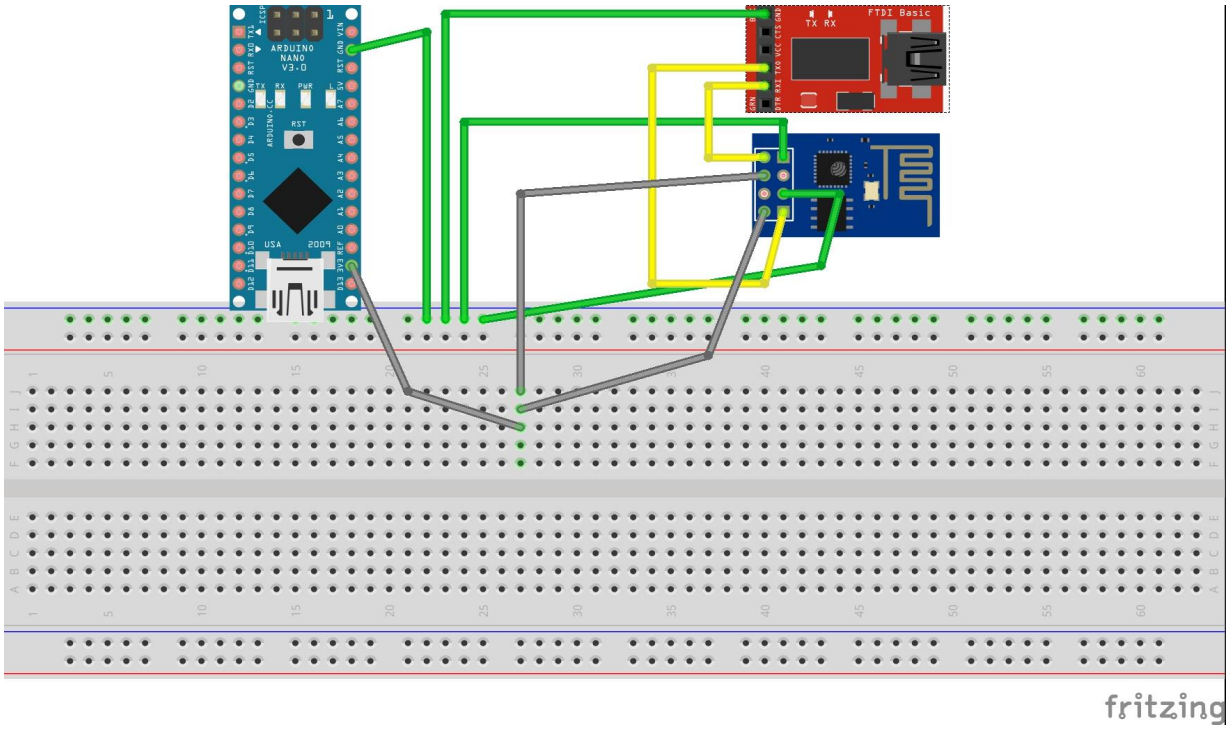


Figure 1: Circuit during the programming of Arduino

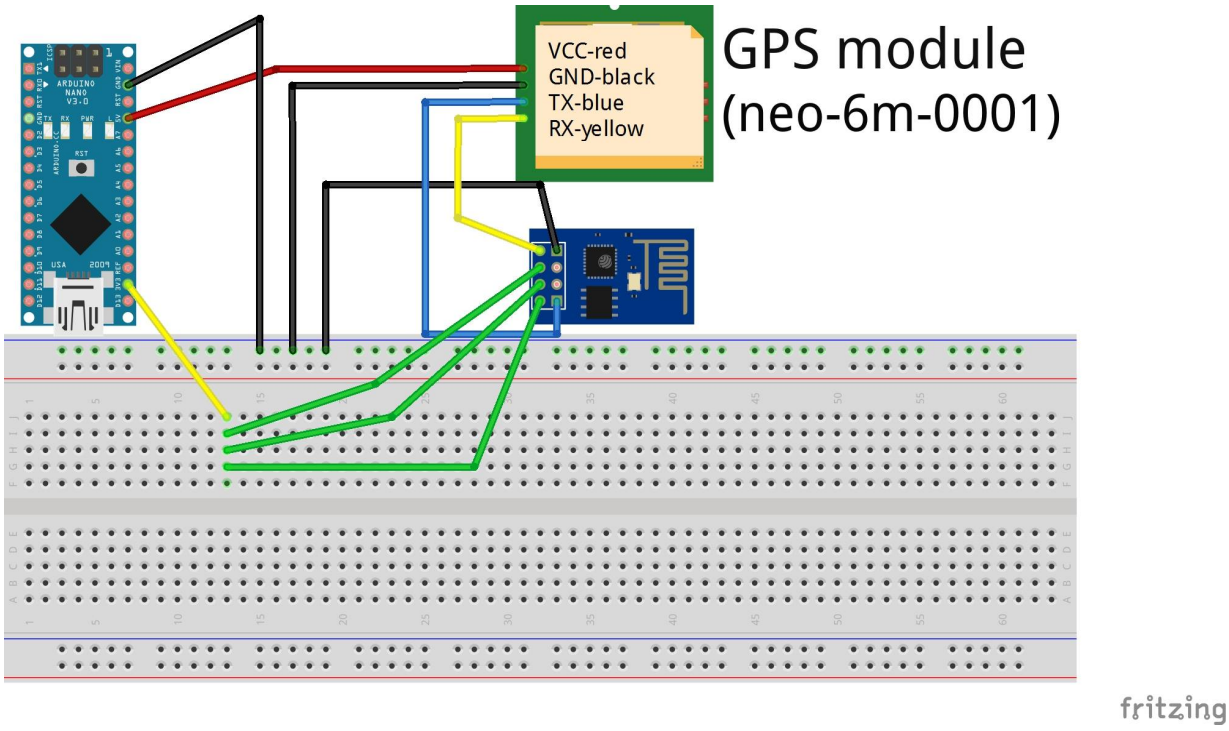


Figure 2: Setup which sends its location

NAME OF COLLEGE

Department of Computer Engineering
LOCATION IN PUNE, Pune PINCODE

CERTIFICATE

This is certify that the project entitled

"NAME OF PROJECT"

submitted by

GROUP MEMBER A	**ROLL NUMBER A**
GROUP MEMBER B	**ROLL NUMBER B**
GROUP MEMBER C	**ROLL NUMBER C**
GROUP MEMBER D	**ROLL NUMBER D**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at NAME OF COLLEGE, Pune under the University of Pune. This work is done during year 2012-2013, under our guidance.

Date: / /

(Prof. GUIDE NAME)
Project Guide

(Prof. PROJECT COORD NAME)
Project Coordinator

(Prof. HOD NAME)
HOD, Computer Department

(Dr. PRINCIPAL NAME)
Principal

External Examiner

Acknowledgements

We are profoundly grateful to **Coordinators, Robotics Club** for their expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

We would like to express deepest appreciation towards **Mr. Ayush Sakhya**, General Secretary, SnT Council, IIT KANPUR whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the co-workers and mentors of Robotics Club who helped me directly or indirectly during this course of work.

AYUHSI KUSHWAHA
RAGHAV GARG
SAGAR CHAUDHARY
SHIKHAR MAHAJAN

OBJECTIVES

To program the Arduino using processing language from Arduino Integrated Development Environment

To compute and send the current location of Arduino MEGA through the GPS module, firstly through the external GSM shield by replying to a predefined text request via a sms, and secondly through an ESP module connected to a specific wifi SSID to a thingspeak.com API

Contents

List of Figures

REAL TIME CIRCUITS

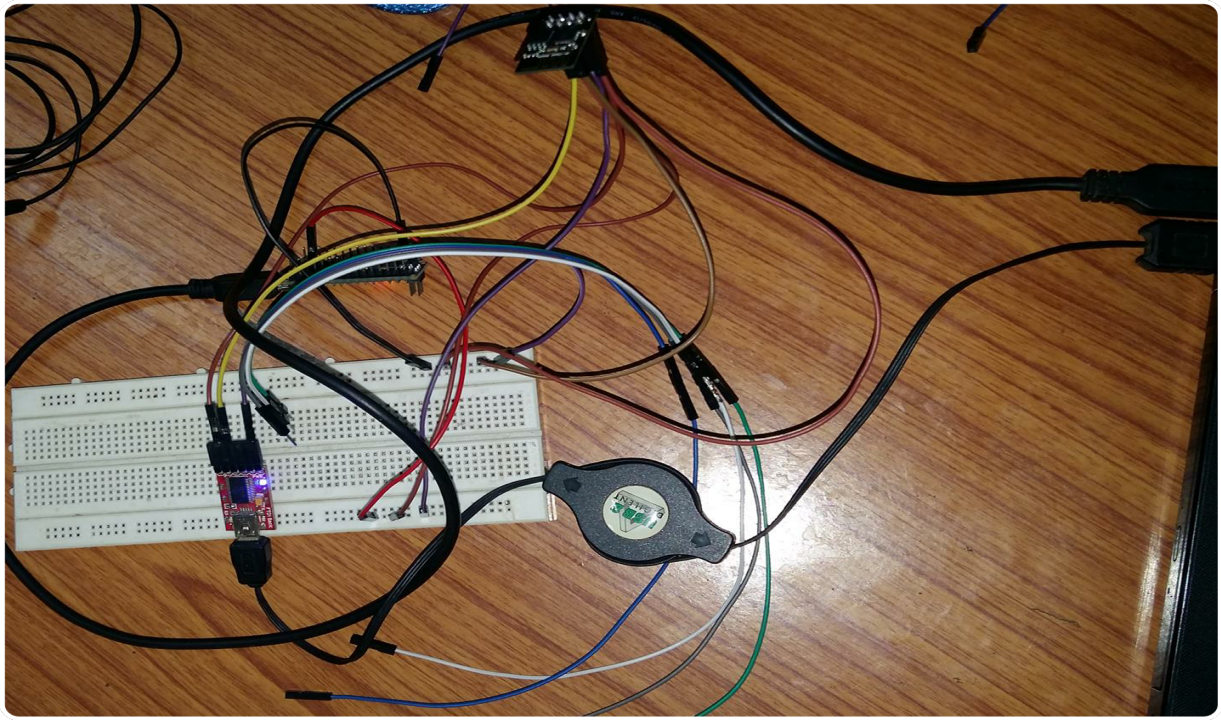


Figure 3: Circuit during the programming of Arduino

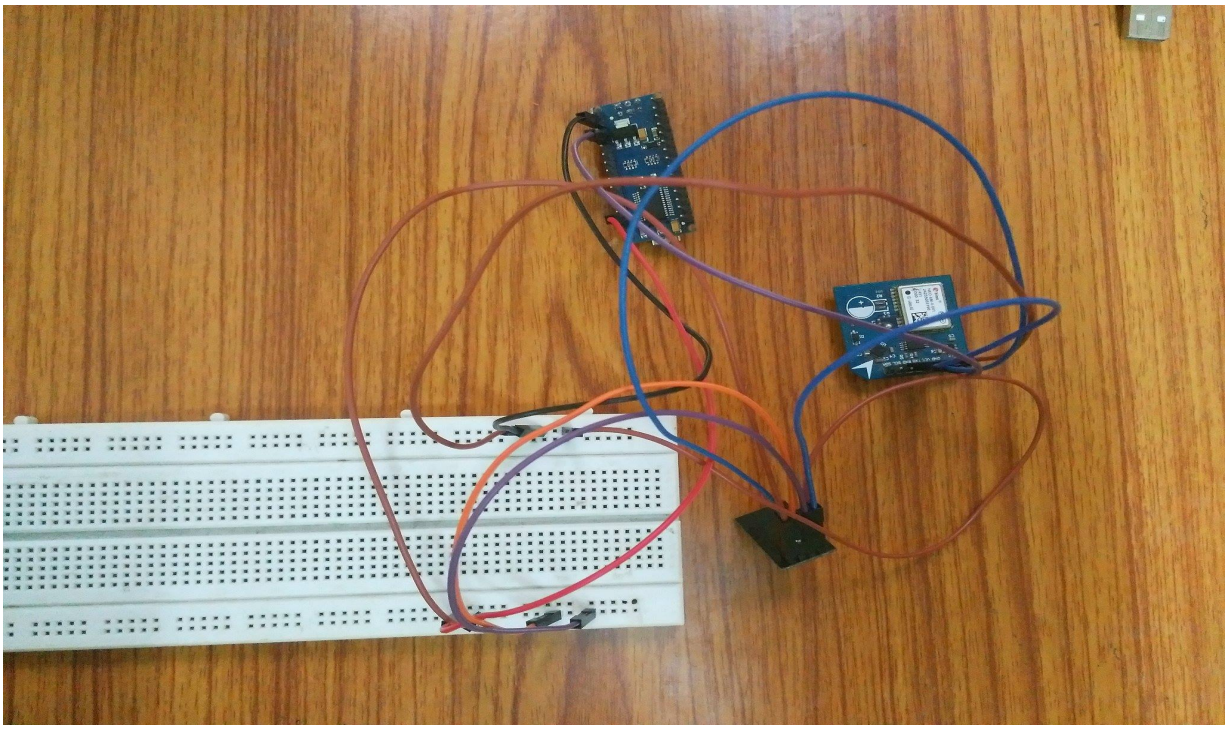


Figure 4: Setup which sends its location

CODE FOR GSM SHIELD

```
#include <TinyGPS.h>
#include <GSM.h>

#define PIN ""

long lati, lon;
unsigned long tym, dat, lastUpd;

TinyGPS gps;

GSM gsmAccess;
GSM_SMS smsRec;
GSM_SMS smsSend;

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
  while (!Serial);

  boolean notConnected = true;
  while (notConnected){
    if (gsmAccess.begin(PIN) == GSM_READY){
      notConnected = false;
    }
    else {
      Serial.println("SIM not connected");
      delay(1000);
    }
  }

  Serial.println("GSM connected. Initialized!");
}

void loop() {
  Serial.println("No new requests received!");

  char c;
  char senderNumber[20];
  String sendTextMsg;

  int countRec = 0;
  int recStatus;

  char xpctdRecMsg[] = "Please tell my location";

  if (smsRec.available()){
    Serial.println("Message received from: ");
    smsRec.remoteNumber(senderNumber, 20);
    Serial.println(senderNumber);
  }

```

```
52  while (c = smsRec.read()) {
53    if ( countRec < 23){
54      if ( c == xpctdRecMsg[countRec] ){
55        countRec = countRec + 1;
56      }
57      else{
58        Serial.println("Invalid Request Received!");
59        recStatus = 0;
60        break;
61      }
62    }
63    if ( countRec == 23){
64      recStatus = 1;
65      Serial.println(" Valid Request Received :D ");
66      break;
67    }
68  }
69
70  if (recStatus == 1){
71
72    smsRec.flush();
73    Serial.println("Message Flushed");
74
75    /*
76      senderNumber complexity!
77     */
78
79    Serial.println("Preparing to fetch data from GPS Shield");
80    while(Serial1.available()){
81     if (gps.encode(Serial1.read())){
82       gps.get_position( &lati, &lon );
83       gps.get_datetime( &dat, &tym, &lastUpd);
84       Serial.print("Position: ");
85       Serial.print("lati: ");Serial.print(lati);Serial.print(" ");
86       Serial.print("lon: ");Serial.println(lon);
87       Serial.print("Date: "); Serial.print(dat); Serial.print(" ");
88       Serial.print("Time: "); Serial.println(tym);
89       Serial.print("Time since last update: "); Serial.print(lastUpd); Serial.
             println(" ms");
90     }
91    }
92
93    Serial.println("Now preparing to send fetched data via SMS");
94    sendTextMsg = sendTextMsg + "Latitude: " + lati + " Longitude: " + lon
95                  + " on Date: " + dat + " at time: " + tym
96                  + ". Time since last updated data: " + lastUpd
97                  + " ms.";
98
99    smsSend.beginSMS(senderNumber);
100   smsSend.print(sendTextMsg);
101   smsSend.endSMS();
102
103   Serial.println("SMS sent successfully");
104 }
```

```
105
106   delay(1000);
107 }
```

CODE FOR ESP SHIELD

```
1  #include <ESP8266WiFi.h>
2  #include <TinyGPS.h>
3
4  String apiKey = "CP2IX9ZRBIN2EM4B";
5  const char* ssid = "GenPuchchu";
6  const char* password = "motamagga";
7  const char* server = "api.thingspeak.com";
8
9  long lati, lon;
10 unsigned long tym, lastUpd;
11
12 TinyGPS gps;
13
14 WiFiClient client;
15
16 void setup(){
17   Serial.begin(9600);
18   delay(10);
19
20   WiFi.begin(ssid,password);
21
22   while(WiFi.status() != WL_CONNECTED){
23   delay(500);
24   }
25 }
26
27 void loop(){
28
29   while(Serial.available()){
30        if (gps.encode(Serial.read())){
31          gps.get_position( &lati, &lon );
32          gps.get_datetime( NULL, &tym, &lastUpd);
33        }
34     }
35   if (client.connect(server,80)){
36     String postStr = apiKey;
37     postStr +="&field1=";
38     postStr += String(lati);
39     postStr +="&field2=";
40     postStr += String(lon);
```

```
41      postStr +="&field3=";
42      postStr += String(tym);
43      postStr +="&field4=";
44      postStr += String(lastUpd);
45      postStr += "\r\n\r\n";
46
47      client.print("POST /update HTTP/1.1\n");
48      client.print("Host: api.thingspeak.com\n");
49      client.print("Connection: close\n");
50      client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
51      client.print("Content-Type: application/x-www-form-urlencoded\n");
52      client.print("Content-Length: ");
53      client.print(postStr.length());
54      client.print("\n\n");
55      client.print(postStr);
56    }
57    client.stop();
58
59    delay(15000);
60  }
```

CHALLENGES & LIMITATIONS

First of all challenges that we faced was the testing of hardware that were provided to us, specifically the GPS module. Winter was thriving and fog was immense due to which the GPS module was not able to fetch data in evening and afterwards and we had to work on it during daytime under clear sky on the rooftops carrying our whole setup.

The reset pin of ESP module has to be toggled after the program has been uploaded to the ESP module from LOW to HIGH. Sometimes, this didn't worked until we removed the wire connections and setup again

Making a server by our own was a major task as it required extra knowledge of PHP and other web dev skills. So, rather than composing a fully customized server with alluring GUI, we decided to make use of thingspeak.com API with built in features which recorded and displayed the data fetched (latitudes, longitudes, time latency) graphically with time axis.

A major limitation while using the ESP module setup is that the location observer as well as the setup has to be connected to the Internet while transferring the data.

SCREENSHOT OF RESULTS

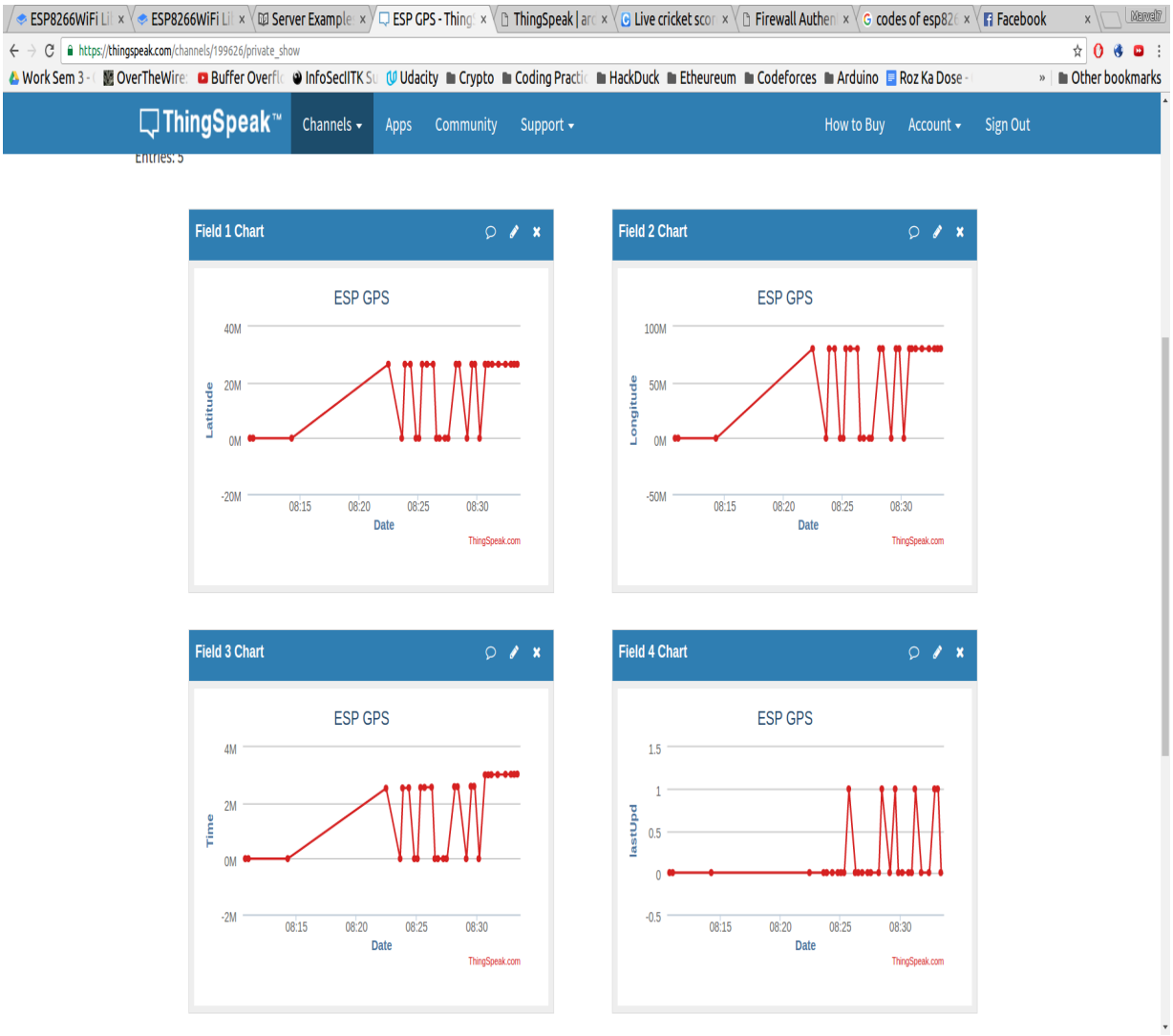


Figure 5: Data fetched and displayed by thingspeak API

Chapter 1

System Testing

WRITE HERE.

1.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	AAAA	BBBB	CCCC	DDDD
T02	AAAA	BBBB	CCCC	DDDD
T03	AAAA	BBBB	CCCC	DDDD

Note: Testing should be performed manually

Chapter 2

Project Planning

2.1 SECTION 1

WRITE HERE.

Chapter 3

Implementation

WRITE HERE, PARAGRAPH 1.

WRITE HERE, PARAGRAPH 2.

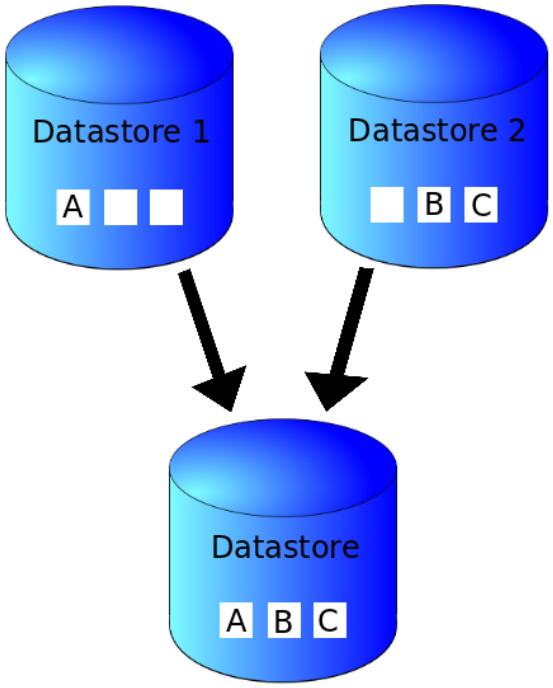


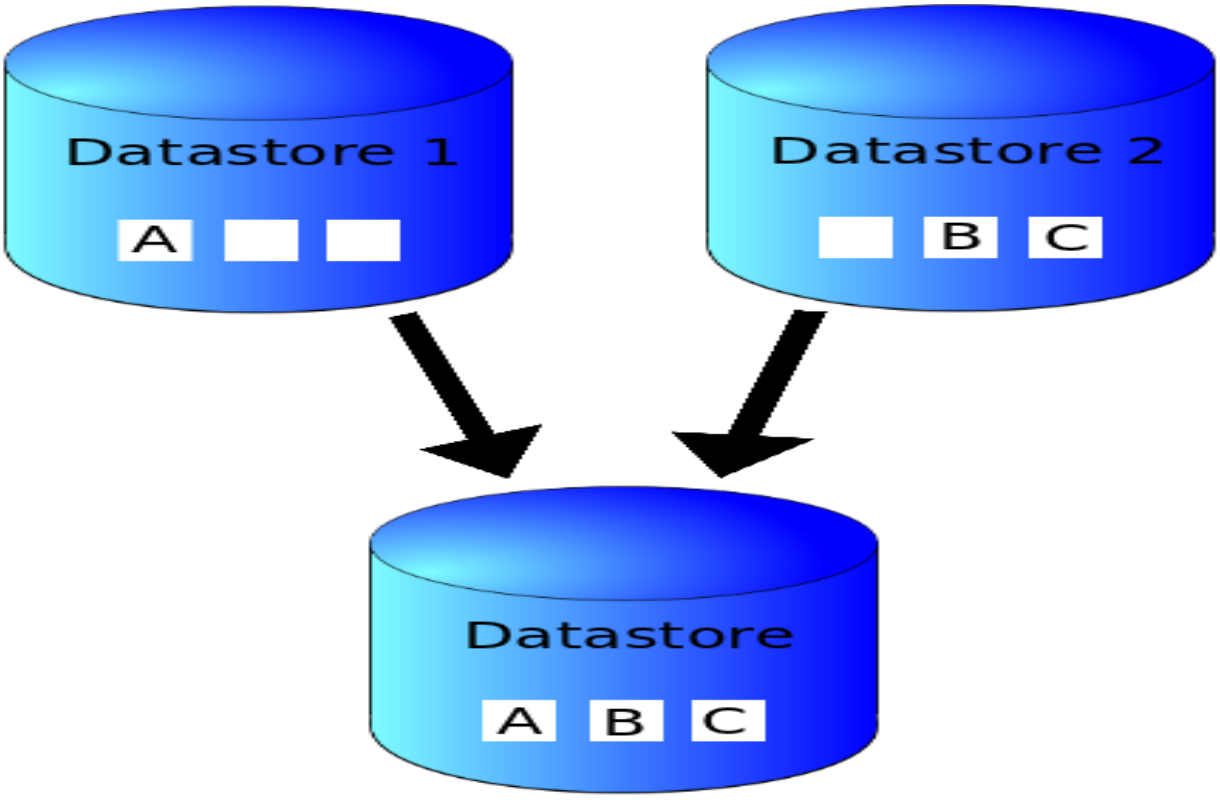
Figure 3.1: IMAGE CAPTION

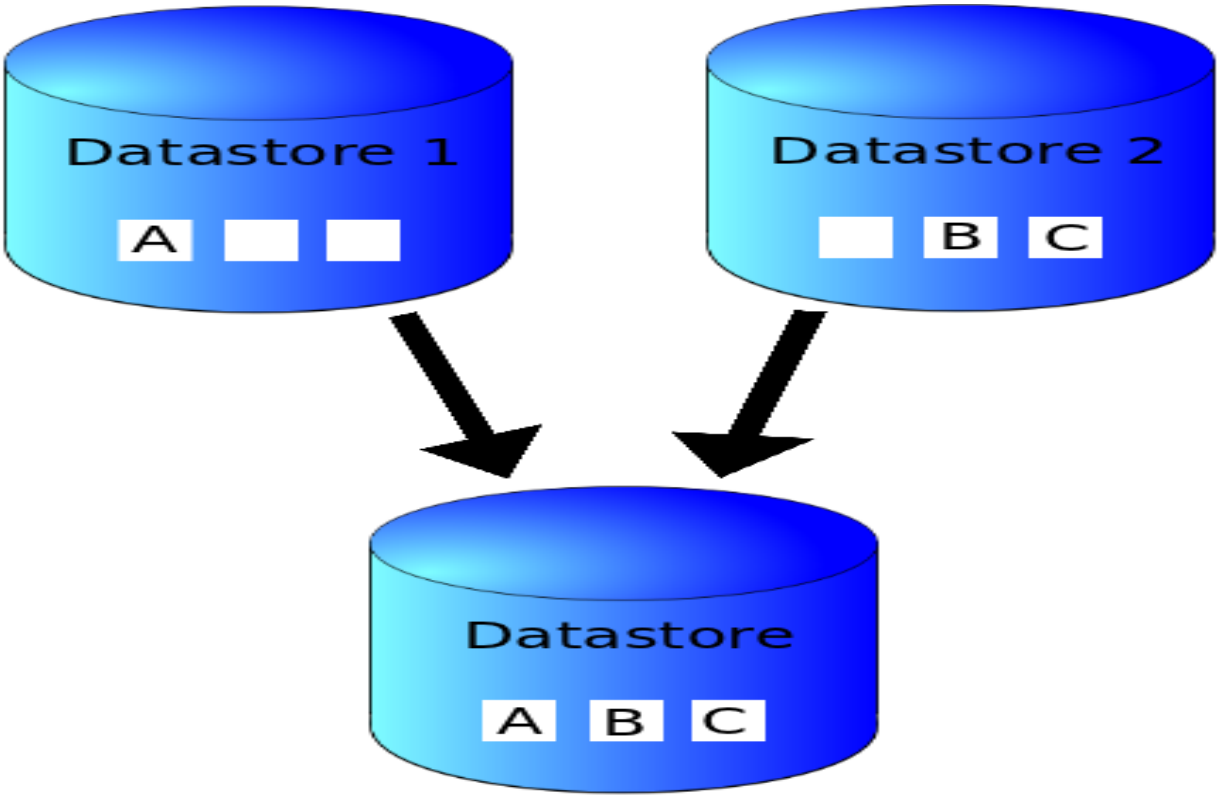
```
PASTE YOUR CODE HERE
```

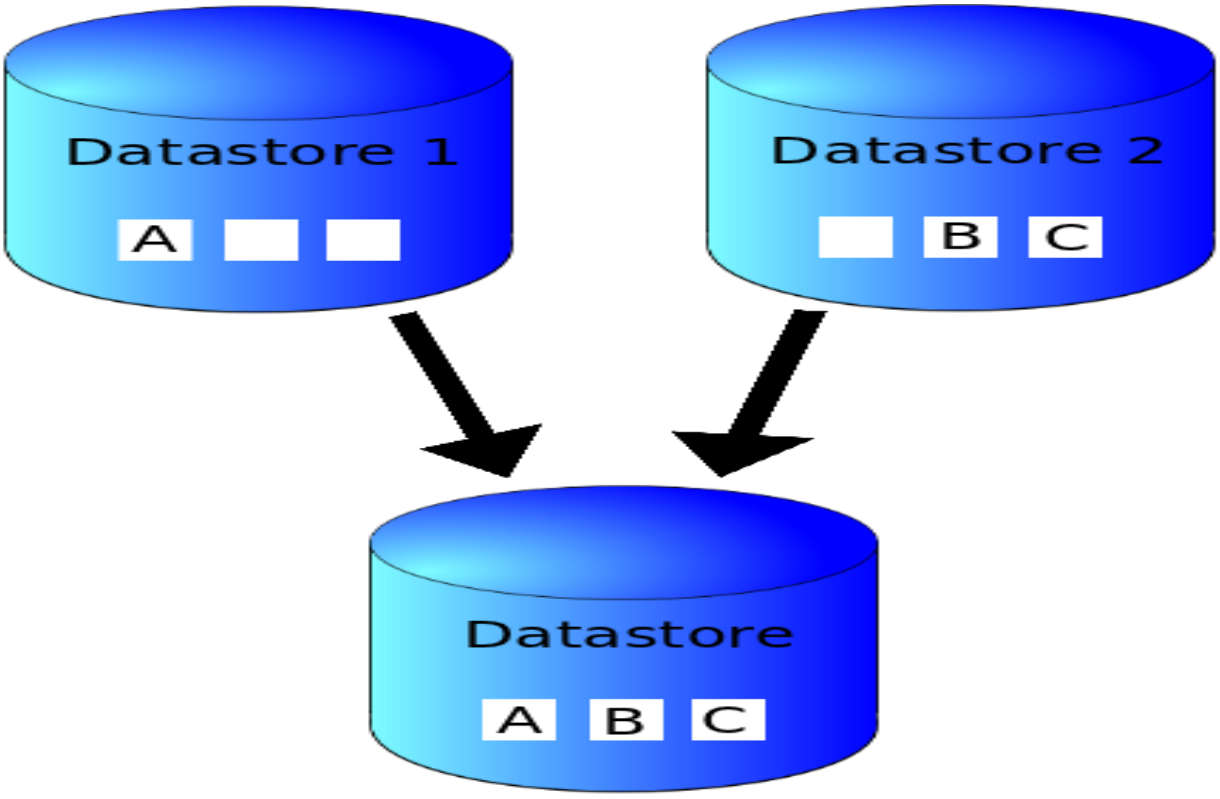
Chapter 4

Screenshots of Project

4.1 SECTION NAME




Datastore 1
A
Datastore 2
B C
Datastore
A B C


Datastore 1
A
Datastore 2
B C
Datastore
A B C

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

WRITE HERE.

5.2 Future Scope

WRITE HERE.

- ITEM 1
- ITEM 2
- ITEM 3

References

[1] *NAME OF IEEE PAPER*; NAME OF AUTHORS

[2] `http://EXAMPLE.com`