Bus Tracking and Monitoring System Using RFID Technology

Mosab Wadea 201021320
Omar Amin 201073280
Ahmed Bajubair 201152850
Mahdi Sahel 201152070

***Abstract*—Put the text of your abstract here.**

I. Introduction

II. Problem

The main transportation method in KFUPM is the bus system that is supervised and ran by the transportation department. The problem with the system depends on the knowledge of the student about the timing of the busses movements and when busses are located in any station. If a small delay or error happens in the system it will affect all the time schedule and the flow of the busses. This error is most likely to occur and can not be prevented. The problem with the students is that they will not be able to identify when a delay occurs and wither a bus is available or not.

III. Solution

The solution proposed in this project is to provide a monitoring system for both the student and the administrators. Where students can access a website to check for busses and their availability, as will as the administrators who can view the tracking results and evaluate the efficiency of the system.

A. Requirements

In order for the project to fulfill the needs it must sustain the following requirements:

- Identify the busses and their assigned lines.
- Detect the busses and their movements.
- Detect busses on the run without the need to stop.
- Users can interact with the system and submit complaints.
- Administrators can access the system to evaluate the efficiency.
- Ability to accommodate multiple lines.

B. Specifications

In order to make these requirements feasible the project has to implement the following specifications:

- Attach RFID passive tag to each bus.
- Install RFID reader and antenna at each bus station.
- Develop web based application that connects to the readers and fitch data and displays them.

IV. Project Design

The system will depend on the bus stations as scanning locations for the busses. Each bus is tagged with an RFID passive tag. These tag are identified in the system by their unique ID (UID). When the bus approaches a bus station it will check into that station. This will let the system know when the bus is checked in and it will know what is the estimated time of arrival (ETA) to the next station. The users can access all these information from the web-base front-end interface. In Fig. 1 the block diagram shows how the components of the system are integrated together.

A. Block diagram

In the block diagram seen in Fig. 1 there are three main components that interact with the system, the bus station which contains the readers and the antennas, the front-end that consists of the students interface and the administrators panel, the last component is the bus that is tagged with an RFID tag. All these components are linked together and processed in the server. The server

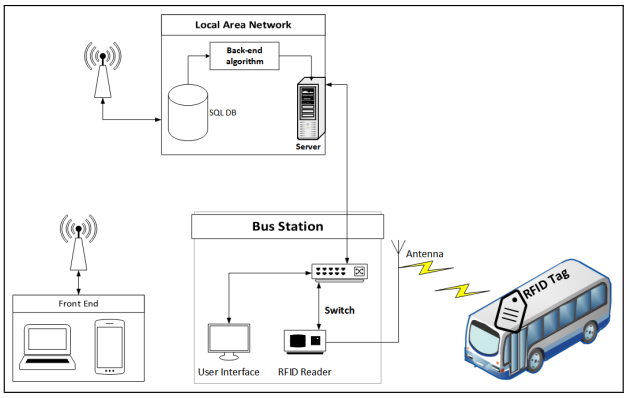

Fig. 1. The block diagram of the system

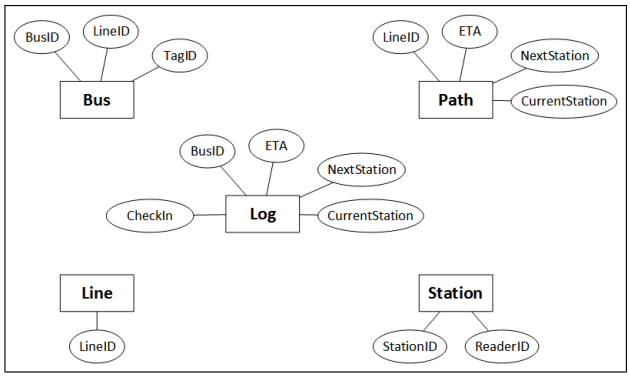

Fig. 2. The entity model of the database

contains the software that will be described in details in the next section.

B. Software architecture

The software part is divided into three main parts as shown in Fig. 1 in the local area network tear, these three main parts are:

- Back-end logic.
- The database.
- Web server.

The back-end contains a script that will communicate with the reader to check the busses availability at each station and it will write these information to a database. The web server will communicate with the database to get the latest information and it will update the front-end interface without the need to refresh the page by the user. The database is constructed of the following tables:

- Bus.
- Path.
- Line.
- Station.
- Lo

The entity model shown in Fig. 2 shows the database structure and the attributes for each entity.

V. Methodology

The system is divided into two main parts, software and hardware. The software will contain the database as well as the communication with the reader as part of the back-end and the front-end which is the website of the service. The hardware consists mainly from the reader and the antennas as well as the tags attached to each bus.

A. Hardware

A selection of appropriate hardware was made to match all the requirements of the project, here is a brief description for each hardware part and how it is tested and implemented:

1) Sirit RFID reader:
2) RFID antenna:
3) RFID tags:

B. Software

The implementation of the software was mainly done in C# using Visual Studio. The components of the software consists of the front-end and the back-end logic. These components communicate with each other using SQL database.

1) Back-end logic: In order to connect to the RFID readers a script was developed by the team in C#. This Script connects to the reader and sends a command that asks the reader about any tag in the range. The following code establishes the connection and starts an infinite loop to keep reading the tags using another method until the process is manually canceled:

```
string hostName = "172.16.10.99";
int hostPort = 50007;

IPAddress host = IPAddress.Parse(hostName);
IPEndPoint hostep = new IPEndPoint(host,
    hostPort);
Socket socket = new
    Socket(AddressFamily.InterNetwork,
    SocketType.Stream, ProtocolType.IP);
while (!socket.Connected){
 socket.Connect(hostep);
}
while (true){
 try{
   NetworkStream nw = new
       NetworkStream(socket);
   sw = new StreamWriter(nw);
```

```
    sr = new StreamReader(nw);
    Tag[] tags = readTags();
  }
  catch (Exception){
    Console.WriteLine("something went wrong");
  }
```

The following method is the method that is responsible of sending the read command to the reader, when it fitches the tags it will check for the "ok" string to make sure that there are tags detected, if tags were detected it will start splitting the information and assign the values received into a custom type called "Tag".

```
static Tag[] readTags(){
 string command = "tag.db.scan_tags(100)";
 writeCommand(command);
 Tag[] tags = new Tag[15];

 if (sr.ReadLine().Equals("ok")){
  char[] delimiterChars = { '=', ',' };
  int i = 0;
  string[] words;
  while ((words =
      sr.ReadLine().Split(delimiterChars)).Length
      != 1){
   tags[i] = new Tag();
   tags[i].id = words[1];
   tags[i].readTime =
       DateTime.Parse(words[5]).TimeOfDay;
   tags[i].antenna = Int32.Parse(words[9]);
   var x = tags[i].id;
   int antenna = tags[i].antenna;
```

After that the code will write the new data to the database using MS SQL connection string.

2) SQL database:

3) Front-end implementation:

VI. Discussion

A. Challenges

B. limitations

VII. Conclusion