Nazir Ahmad

Formal Gathering Plus

An ICT Add-on for Formal Gatherings

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

27 April 2015

Helsinki Metropolia University of Applied Sciences

Author(s) Title Number of Pages Date	Nazir Ahmad Formal Gathering Plus 34 pages 27 April 2015
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Embedded Engineering
Instructor(s)	Sami Sainio, Senior lecturer

The goal of this project was to develop an application that would provide assistance for conducting events, seminars or any other type of formal gathering. The initial functional requirements were that at the top level the system administrator would define which persons could conduct an event. Then a record would be made for start / end time, location and subject of event, and who were the participants including their time of arrival and departure.

During the project, a client-server based application was developed. The server was built using a PHP-enabled Apache web server and MySQL database server. The RFID technology was used to recognise the event's host person and participants. A complete ID-Reader device was made using an RFID reader module and Arduino board. This device passes the IDs of the host and participants to the server and displays acknowledgement to the user regarding the state of the event.

All initial functional requirements were implemented with a few enhancements. There are also a few possibilities highlighted for further development. For example, a web-page for RFID tag allotment to participants, sending notification to the host person and participants regarding an upcoming event, a page for participants' feedback, and inclusion of an LCD screen in the ID-Reader-Unit for providing more user-friendly acknowledgements.

Keywords	events, participants, Arduino, RFID, Web-based ICT application

Contents

1 Introduction

Formal gathering is a regular activity in almost every organisation. Some of the terms used to describe those gatherings include; meeting, seminar, class, convention, and session. All of these events are conducted with a purpose (or objective) and have great importance in business, academic and other social sectors.

The frequency and duration of those types of events are variable, depending on the nature of work done at a particular organisation or department. In most cases the date, time and subject (along with other parameters) are pre-planned but an unplanned gathering can be called to discuss some emergency issue.

No matter what type the organisation is, every formal gathering has a few features in common:

- At least one host who controls the proceedings
- Few to many participants
- Timings: when the event started and ended, when a participant arrived at / departed from the event
- Topic (or subject) of the event

The goal of this bachelor's thesis project is to design and develop an ICT application that would be helpful in achieving the objectives of a formal gathering. The first version of the product should be able to collect and store the above listed data of an event, without requiring any extra effort from the host and participants and display the stored information in a simple and comprehensive form.

2 Background

2.1 Use of ICT Applications in Today's World

ICT applications are used by people for objective processing of information by using relevant technologies. The main sections of an ICT application are shown in Figure 1 in a simplified way. Experts, engineers and researchers have developed these applications by observing the demands of a specific sector [1]. In some cases, these are invented to create new values. The users gain many advantages such as accuracy, simplicity, convenience / speed.

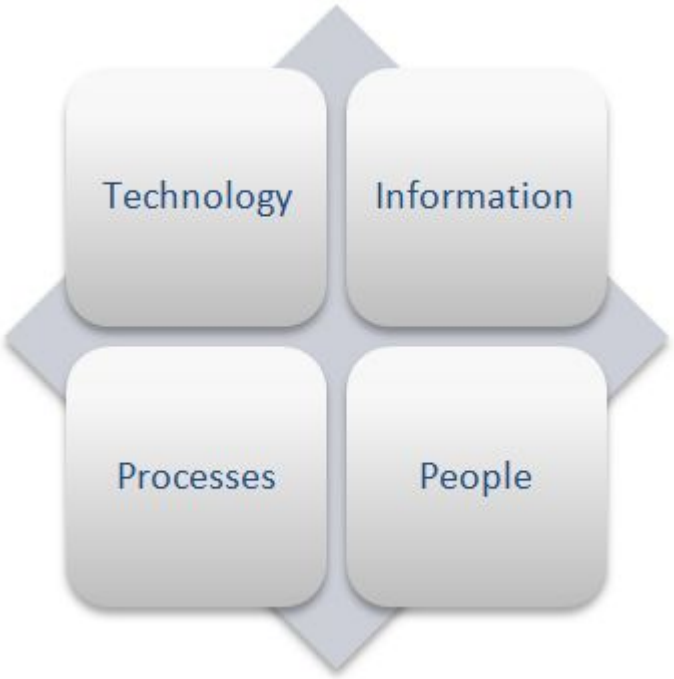

Figure 1. Main sections of an ICT application

Reprinted from [2]

Due to the important role of ICT applications in businesses and organisations, employers give preference to candidates who are more ICT-literate according to the requirements. Also candidates especially mention all of their ICT knowledge and skills in their CV when they apply for a job.

2.2 Potential for ICT Applications in a Selected Sector

Formal gathering is a complicated process and has many aspects to be addressed. There are a number of aspects which could be considered (*before, during and after the session)*, in order to improve the overall outcome of an event. For example:

- Before includes: Resource allocation (room, time), topic, agenda, leader, list of participants.
- During: Note taking, timing of proceedings, record of participants.
- After: Making reports based on the recorded information for analysis of a completed event and future planning.

Since there is large amount of information processing is involved, it makes this sector a good candidate of ICT applications. Use of proper technologies and software can make the task simple, quick, and error-free.

2.3 Project Feasibility

According to information presented in Figure 2 about types of new products, the intended product comes in the category of New Product Line. It is not completely new to the market but it is new to us. It will have some differences in terms of features, functionality, and cost from the existing products.

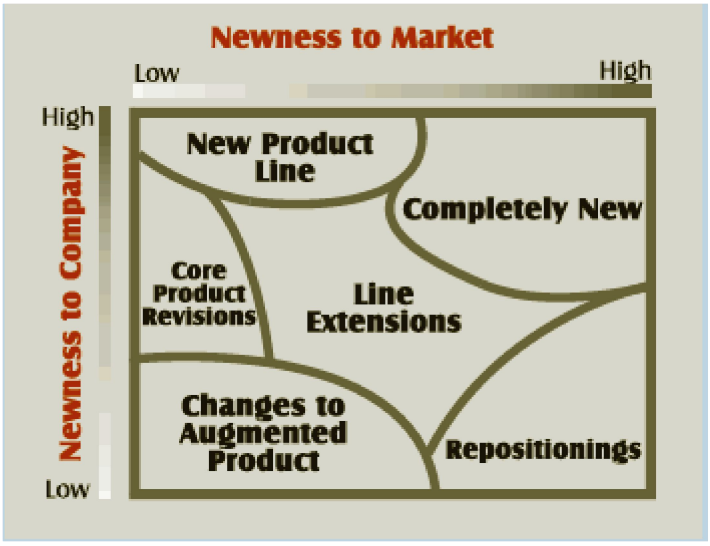

Figure 2. Types of New Products

Reprinted from [3]

Its cost will be significantly lower than that of available products. Functionality will be simple and user-friendly. It will include such features which are fundamental in nature and common in various types of events. So the same product could be utilized in different events and organizations. Finally, it is possible to develop it from a technical point of view. All these factors make this project a feasible project and work should continue until completion.

3 Project Overview

3.1 Functional Requirements

On completion of this project, the new product should provide at least the functionality mentioned below:

- The system administrator will be able to define a future event on the server through a web browser. Here he/she will mention the subject, host-person, location, and time of that event.
- An event will begin and end by an identified host person.
- Participants will also be identified and their sign-in and sign-out time will be recorded.
- If an undefined event gets conducted by a valid host then all information will be stored in the same way but the event's subject will have the text "untitled".
- The subject text will be stored in an editable form.
- The recorded information will be accessible to authenticated users via a web interface.
- The system administrator will be able to create a new (event) host, and web user accounts as well.

3.2 System Block Diagram

A system model is made to implement the required functionality. It is shown in Figure 3. The role of each block is as follows:

- The server machine will receive information from its clients: ID-Reader-Unit(s), and web browser machine(s). It will process the information and generate a response accordingly. It will also be able to store the information in a well-organized form.

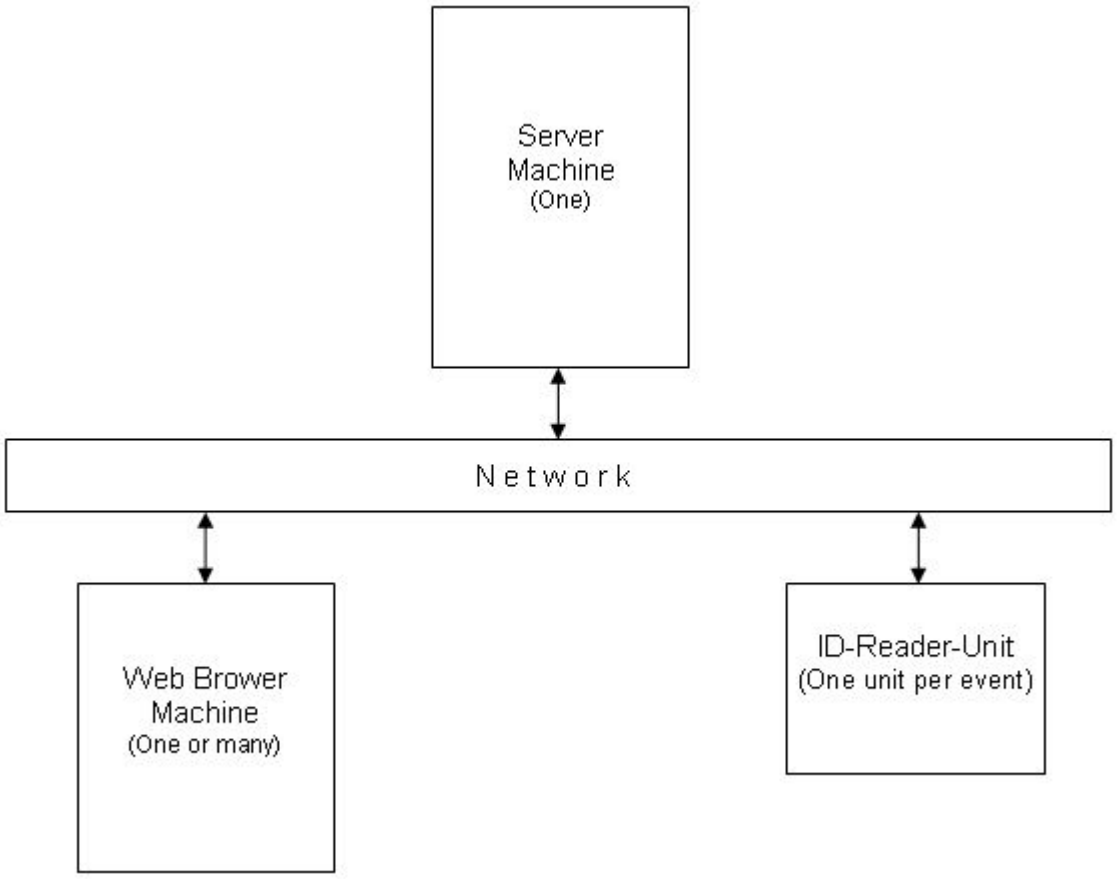

Figure 3. A simplified view of the system

- The ID-Reader-Unit will read the ID of the event's host and participants, one at a time. This unit will make an indication upon a successful ID reading process. If the event has not started and the identified individual is a valid host person, then the event will begin. If the event is already started, and the identified individual is the same host (who started the event earlier), then the event will end. Each participant has to present their ID to the ID-Reader-Unit upon arrival and departure from the event, in order to make a record of their presence.

- The web browser machine(s) will be used by the system administrator and web users.

3.3 Review of Chosen Technologies

3.3.1 Arduino Platform

An Arduino board is chosen to build the ID-Reader-Unit. The official web site for Arduino products and software is “http://www.arduino.cc”. Various types of boards are offered by them. Each type has some differences from the other in terms of processing capability, physical dimensions, and I/O options.

Mostly Arduino boards are based on ATMEGA series of microcontrollers made by Atmel Corporation. There are other alternatives available in the market but Arduino boards have some advantages over the others, for example:

- Arduino boards comparatively less expensive
- Multi platform support (Linux, Windows, Mac)
- Clear and straight forward programming environment, which is equally useful for new and expert users.
- Open-source extensible nature of software and hardware. [4]

The model of the chosen Arduino board is Arduino Uno Revision 3. It is shown in Figure 4. The main features of this board are: ATmega328 microcontroller, 16 MHz Clock Speed, 32 KB Flash Memory, 14 Digital I/O Pins (6 PWM outputs), 6 Analog Inputs, Input voltage 7-12V [5]. It is supported by the current stable version (*1.0.6*) of Arduino Integrated Development Environment (IDE).

Figure 4. Arduino Uno Revision 3 board
Reprinted from [5]

A view of Arduino IDE is presented in Figure 5. The written program will be transferred to the microcontroller's program memory via USB / serial connection and the boot loader firmware [6]. The board will start blinking an on-board LED connected to pin#13. The LED will keep on blinking until the board is powered ON.

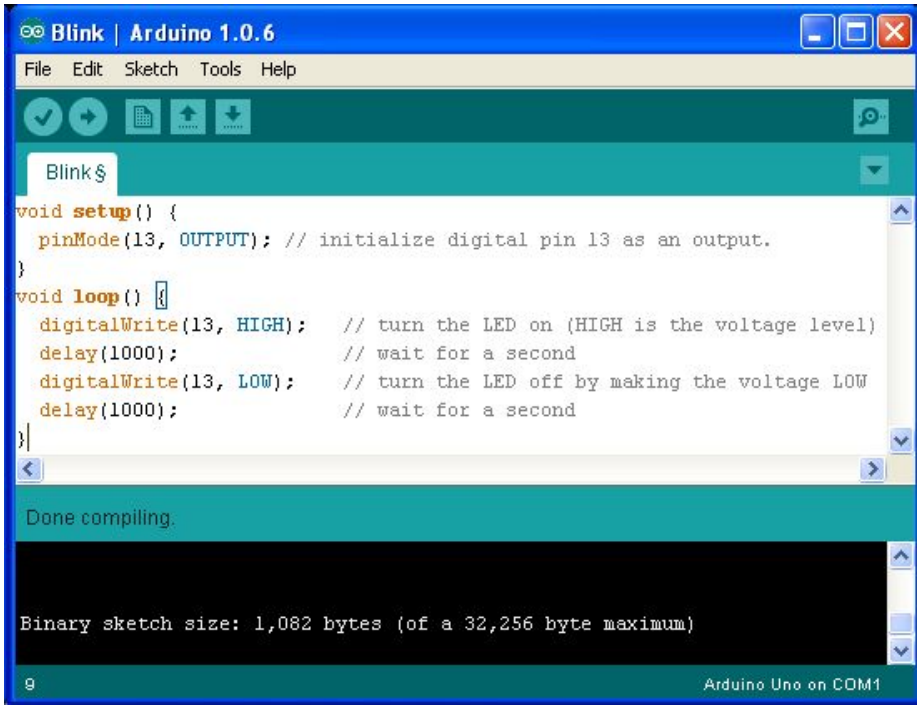

Figure 5. Arduino IDE 1.0.6 and a small program

Program modified from [7]

In order to make the ID-Reader-Unit capable of communicating over the network, another Arduino product is selected, which is called Ethernet shield. A view of this product is shown in Figure 6. This extension board will only utilize pin 4, 10, 11, 12, and 13. [8] The rest of the I/O pins will remain free for other extension boards and electronic components.

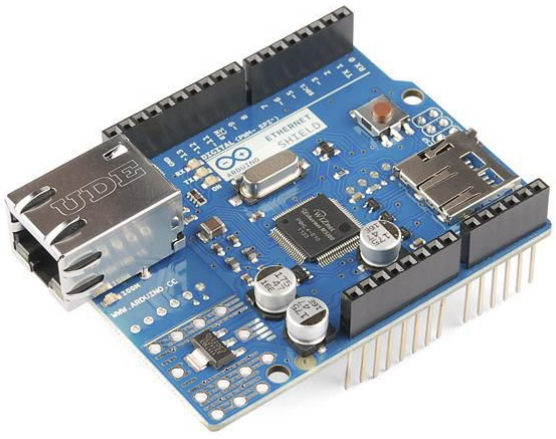

Figure 6. Arduino Ethernet Shield, reprinted from [8]

3.3.2 Radio-frequency Identification

The radio frequency identification process consists of two basic components; RFID tag, and RFID reader. A simplified view of a RFID reading system is presented in Figure 7. Every tag holds a unique number, which is its ID. The role of the RFID tag is to send the stored ID through a radio wave. The role of the RFID reader is to receive the transmission and reproduce that ID number. The RFID reader passes this ID number to its parent device, where it is processed according to the requirements. For successful detection, the tag and reader should be compatible with each other. [9; 10, 5-6]

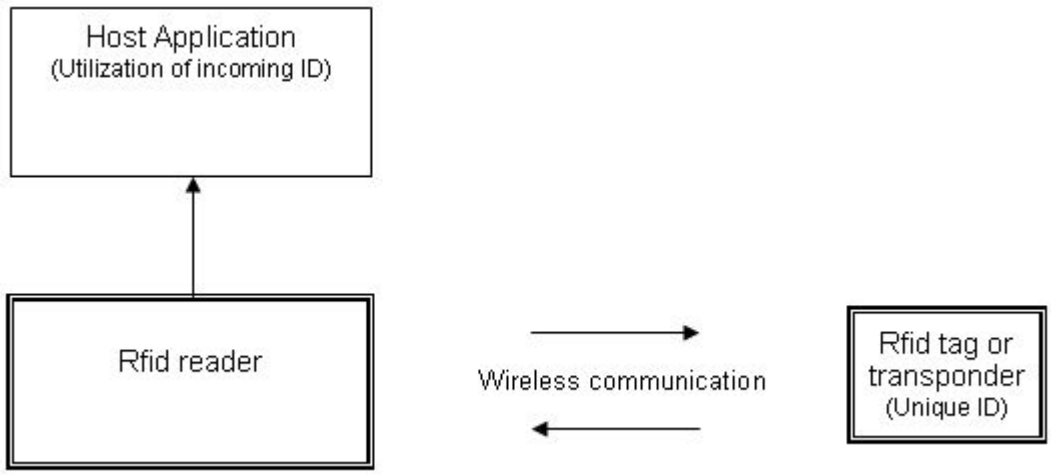

Figure 7. A generic view of RFID reading system

RFID solutions are available in different variations to fill the requirements of different applications for example, variation in power source for tags, operating frequency, and physical dimensions of tags. Power source variations include:

- Active tags have their own power supply (a battery) for internal operations and for transmitting a reply to the RFID reader. They offer a longer reading distance (up to 1500 feet), better immunity, reliability, and more memory. However these tags are expensive, bigger in size and need periodic checking.
- Passive tags do not have their own power source. They fulfill their energy requirements by generating electricity from the incoming radio waves from RFID reader. Their physical size can be small / thin and the cost is lesser than active tags but their reading distance is shorter (inches to 22 feet).
- Semi-active tags use their own battery power for internal operations only. They generate electricity from incoming field of the RFID reader for transmitting their reply. These tags offer better performance than passive tags. [9; 10, 5-6]

Some of the frequency-related aspects of the RFID system are presented in Table 1. Here we notice that the data transfer rate and reading range increase with a higher frequency and also the required antenna size decreases accordingly but reading becomes difficult at higher frequencies if the tag is positioned near some metallic object.

Table 1. Some frequency related characteristics of RFID System

Data collected from [9; 11]

Frequency Range	LF 125-134.2KHz	HF 13.56 MHz	UHF 860 - 960 MHz	Microwave 2.4 GHz
Reading range	< 0.33 meter	Up to ~1 meter	1 to 10 meters	3 meters upwards
Data Speed	Slower ⟷ Faster			
Ability to read near metal	Better ⟷ Worst			
Antenna coil (size)	Longer ⟷ Shorter			

The selected RFID reader module is shown in Figure 8. It sends ID data to the host application through a serial-out pin (SOUT) @ 2400bps, when it is enabled and a tag is present in its reading range. This RFID reader; has a reading range up to 4 inches, operates at a frequency of 125 KHz, is designed for passive tags, and needs a 5V DC input [10].

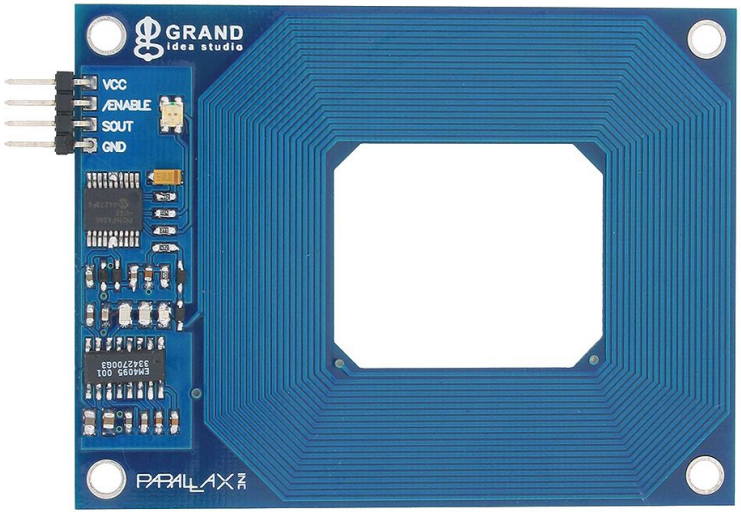

Figure 8. RFID Reader Module, Parallax Inc.

Reprinted from [12]

3.3.2 Server Setup

The server will perform its duties by using the following chosen software packages:

- Apache Web Server is free, open-source, and the most popular web server software. It is an active project of the Apache Software Foundation. [13; 14]
- PHP module is free software from the PHP Group for executing server side scripts written in the PHP programming language. PHP stands for Hypertext Preprocessor. [15]
- MySQL Server from Oracle Corporation for enabling the application to store data. [16]

Further application specific configuration and development will be done on top of these packages. A simplified arrangement of the main sections inside the server is shown in Figure 9.

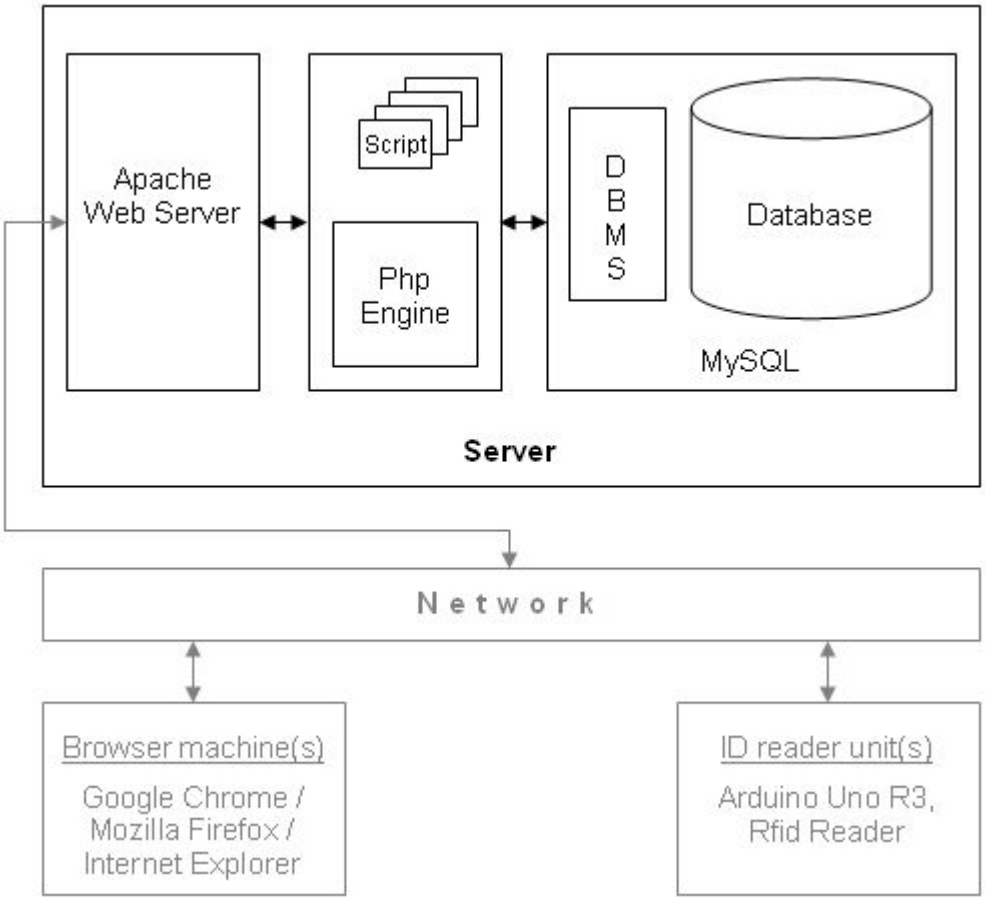

Figure 9. Server block diagram

The web server will receive all of the requests from client devices. It will serve the requests for the static contents itself. But requests for the dynamic contents and database operations will be handled through the PHP engine. Appropriate sever side script(s) will be executed through the PHP engine. The outcome of this execution will be embedded in an HTML page. Finally the web server will send this HTML response to the same client device which initiated the request for these dynamic contents. [17]

All these three mentioned software products Apache Web Server, PHP module, and MySQL Server can be downloaded individually from their vendor's websites. The downloads are available for most of the popular operating systems such as Linux, Windows, and Mac. In addition, to make work easier these products are also provided inside a single package by some of the vendors. For this project a similar unified package is used, which is called "XAMPP" from Apache Friends organization. This package is a free release of the Apache web server including MySQL, PHP, and Perl another programming language. [18]

4 Development

4.1 Database Design and Implementation

The database unit has a central role in this project. This unit will be a part of most of the server operations for example information storage / retrieval, user authentication. At the beginning of database design stage, the whole system will be analyzed through the process of data modeling, in light of the product's functional requirements and project setup. By the end of the data modeling process, it will be clear which information will be stored in the database and how the information will be structured to fulfill the data-related requirements of the application. [19]

During the data modeling process, work will be done at three different levels in a sequence. At each level, a data model will be made. These models are named as conceptual data model, logical data model, and physical data model. The three levels and aspects of the database design / structure to be covered (at each level) are shown in Table 2 and discussed briefly below. [19; 20]

Table 2. Levels of data modelling and aspects in focus

Copied with minor editing from [20]

Aspect of database design / structure	Level-1 Conceptual data model	Level -2 Logical data model	Level -3 Physical data model
Entity Names	✓	✓	
Entity Relationships	✓	✓	
Attributes		✓	
Primary Keys		✓	✓
Foreign keys		✓	✓
Table Names			✓
Column Names			✓
Column Data Types			✓

The work at each level of database modeling will go as follows:

1. At the first level, a conceptual data model will be made. This model will highlight entities and their relationship with each other. The term entity refers to an object or concept for which information can be stored [21] and that object has a meaningful place in the system.

2. At the second level, the conceptual model will be reviewed for improvement and more detail will be added for example defining application-specific attributes or properties of each entity including primary keys and foreign keys. The purpose of making a primary key is to uniquely identify every instance of an entity. An attribute or a set of attributes can be declared as the primary key if its value will always be unique and non-null for every instance of the entity [22]. A foreign key is a copy of primary key and a component of the entity relationship mechanism. This key becomes a member of the related entity's or child table's attributes and acts as a pointer to the original entity or parent table [23].

3. At the third level, a physical data model will be made, based on the work done before. This model will contain complete details of every aspect of the database and will be ready for implementation. Here all entities will be converted into tables and the attributes of entities will become columns within those tables. Next, the data type for each column will be specified. These specifications will be according to functional requirements and compatible with the database server software.

A simplified view of the finalized database structure for this application is shown in Figure 10. It consists of six related tables. The role of each table is as follows:

- "meeting_host" table will store information about event-host persons.
- "guest" table will store information about participants.
- "meeting" table will store information about conducted events.
- "mg_data_buffer" table will store the participants' sign-in and sign-out information during active events.
- "mg_data" table will store the participants' sign-in and sign-out information of conducted events.
- "meeting_plan" table will store information about events to be conducted in the future.

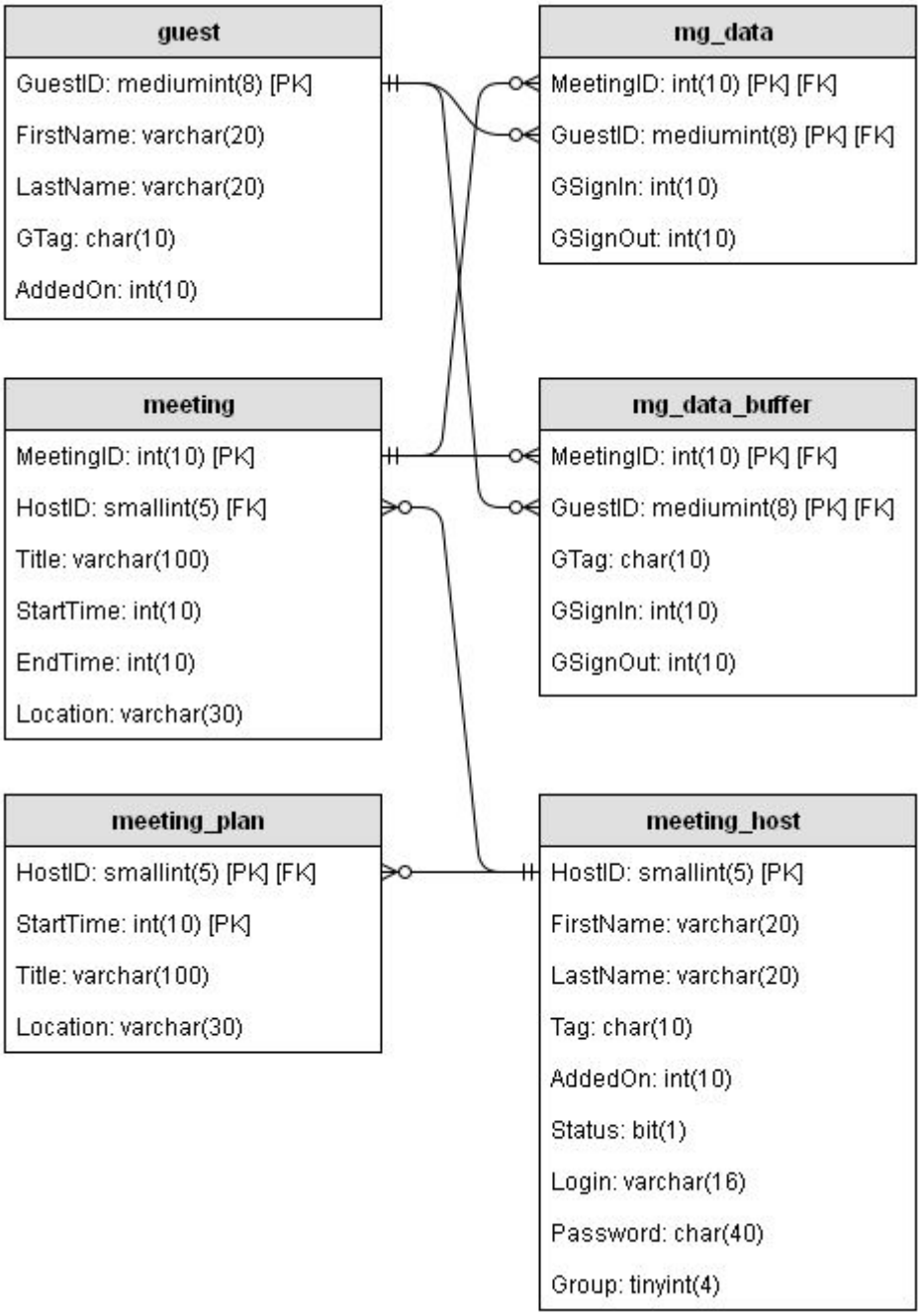

Figure 10. Database structure in the form of physical data model

The finalized physical data model will be implemented on the MySQL database server, which is a subcomponent of the server machine for this application. The implementation can be done through the MySQL command prompt by executing SQL language commands or by using a graphical user interface (GUI) tool. Some of the popular GUI tools for database administration and development are MySQL Workbench, phpMyAdmin, Navicat. [24]. For this project phpMyAdmin tool is used. This

tool is provided inside the XAMPP package along with other modules. In case we use the command line option, the scenario will be as shown in Figure 11 (here a new table is made inside the “fgplusdb” database on the MySQL server).

```
mysql> USE fgplusdb;
Database changed
mysql> CREATE TABLE guest (GuestID mediumint(8) unsigned PRIMARY KEY
AUTO_INCREMENT NOT NULL, FirstName varchar(20) NOT NULL, LastName varchar(20)
NOT NULL, GTag char(10) NOT NULL, AddedOn int(10) unsigned NOT NULL);
Query OK, 0 rows affected (0.14 sec)

mysql> DESCRIBE guest;
+-----------+----------------------+------+-----+---------+----------------+
| Field     | Type                 | Null | Key | Default | Extra          |
+-----------+----------------------+------+-----+---------+----------------+
| GuestID   | mediumint(8) unsigned | NO   | PRI | NULL    | auto_increment |
| FirstName | varchar(20)          | NO   |     | NULL    |                |
| LastName  | varchar(20)          | NO   |     | NULL    |                |
| GTag      | char(10)             | NO   |     | NULL    |                |
| AddedOn   | int(10) unsigned     | NO   |     | NULL    |                |
+-----------+----------------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql>
```

Figure 11. Execution of a few SQL commands at the MySQL command prompt

Along with the database structure, four user accounts are also created for this database through root user of the MySQL server. These accounts are;

- initlogin - will be used at beginning of web browser and server interaction.
- admin - will be used if the administrator (*of this application*) connects through browser.
- mthost - will be used if meeting host person(s) connects through the browser.
- idreader - will be used for id-reader-unit(s) connection.

The data access rights of these accounts are limited according to the tasks to be performed through them. No limit is applied on simultaneous connections.

4.2 ID Reader Unit

At the beginning of the ID-Reader-Unit development, all the chosen components are connected according to the wiring diagram shown in Figure 12. Then behavior of each component is tested with small test programs to see that everything is working properly. Here, two LED indicators are also added to the hardware setup for informing the users about the current state of the system. SSN_LED will only turn ON at the start of a new session (or meeting) and turn OFF when the host person closes the session. TAG_LED will turn ON whenever the system takes RFID tag reading to inform the user that his/her presented tag has been read successfully. This LED will turn OFF after a period of minimum 300ms.

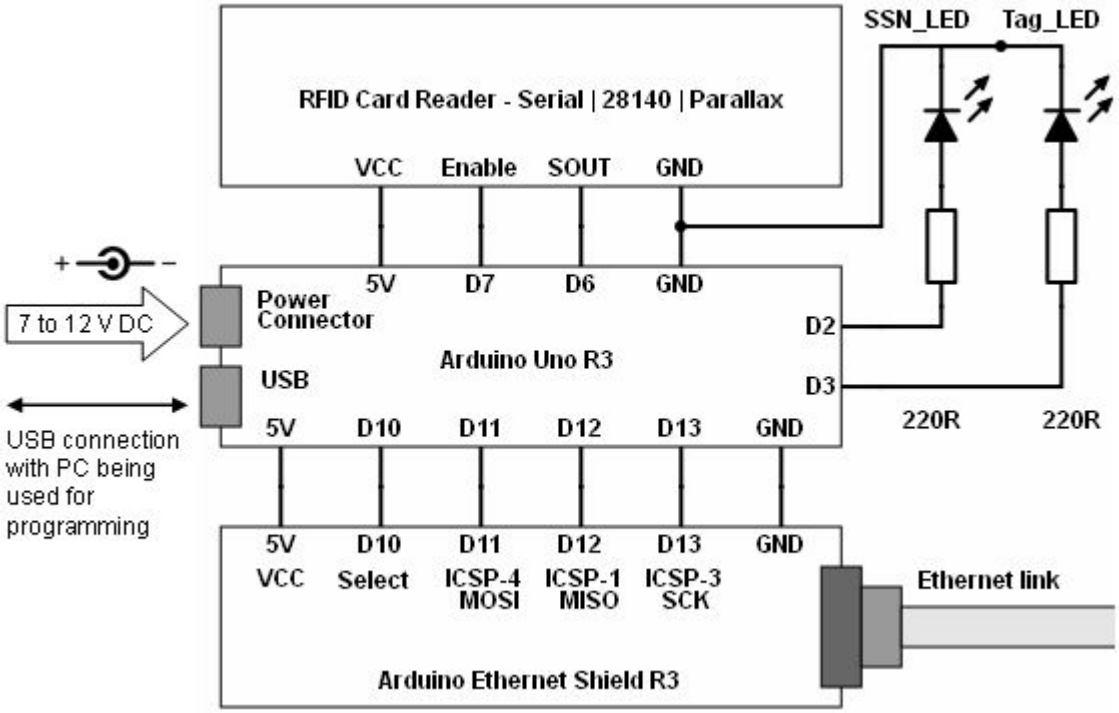

Figure 12. Wiring diagram of ID-Reader-Unit components

RFID reader [10], Arduino Uno R3 [7; 25], Arduino Ethernet Shield R3 [8; 26; 27]

(Unused pins are not displayed and location of pins is rearranged for simplification)

In this hardware setup the Arduino board is acting like a motherboard of a personal computer, while other components can be referred to PC peripherals. Information will arrive at Arduino board through these peripheral, some decisions will be made after some processing and the outcome in the form of control signal and information will be sent to the appropriate peripheral. Then the system will wait for the new input. The whole process will be repeated in a permanent loop.

4.2.1 ID-Reader-Unit Enclosure

Next all of the components shown in Figure 12 are installed inside a plastic enclosure for safe and easy handling. The finalized form of the ID-Reader-Unit from a physical point of view is shown in Figure 13. A number of aspects have been considered during the component installation process, such as:

- The box setup should not place negative impact on the reading range of the RFID reader.
- For sufficient ventilation extra tiny holes are drilled on the bottom and back side of the box.
- The process should be convenient from the user's point of view, which was also one of the reasons for selecting the RFID technology for identification purposes: no manual entry through keypad, no line of sight requirement like in barcode reading and no physical contact requirement like in the fingerprint authentication technologies.

Figure 13. ID-Reader-Unit (*left*) and a compatible RFID tag (*right*) [12]

This enclosure is made by the Hammond Manufacturing Company, part# 1591S. [28] External dimensions of the enclosure in length x width x height format are 110 x 82 x 44 mm. Now this device needs an Ethernet network connection, 7-12V DC power supply and Arduino programming to become an active component of the system.

4.2.2 Arduino programing for ID-Reader-Unit

Arduino programs are written using the Arduino programming language which is based on C/C++. This language contains all the required ingredients for enabling the programmer to start working with Arduino hardware. These ingredients can be classified into three main groups: [29]

1. Structure: control structures, operators.
2. Values: constants, data types, type conversion.
3. Functions: I/O, time, communication.

Furthermore, it is possible to enhance programming capabilities by including libraries for specific hardware or data processing tasks. Several libraries are provided within the Arduino IDE package and additional can be downloaded from the Arduino website or users can make their own libraries. [30]

An Arduino program is known as a “sketch” in arduino terminology. The sketch is written in a text editor and saved as a text file having the extension of “.ino”. The completed sketch is first passed through the code verification stage. If the sketch has some error(s) then appropriate error message(s) will be returned to the user. An error-free sketch is transformed in C (or C++) program. This C program is compiled through the avr-gcc compiler. The compiler converts the user program into a machine-language format (object file). Next the user program is linked with Arduino standard libraries and a single file is produced in the Intel hex file format. Finally this file is uploaded to the target board connected at the pre-selected port. The compilation processing is board-type specific, so it is necessary to select the proper board model before executing the compiler program. [31]

The flow diagram of the finalized Arduino program for the ID-Reader-Unit is shown in Figure 14. The whole program will go through the verification, compiling and uploading stages and program execution will begin from the “void setup function”. This function will be called once (whenever the ID-Reader-Unit is powered ON) for basic initializations and configurations. Next the program will enter into “void loop function”. This function will remain in a permanent loop and provide the required functionality by using local and server resources.

By the end of the setup function, the ID-Reader-Unit will be ready to receive a new tag input and take appropriate actions through the loop function as follows:

- At the start of the loop function the Arduino program will wait for a new RFID tag reading.

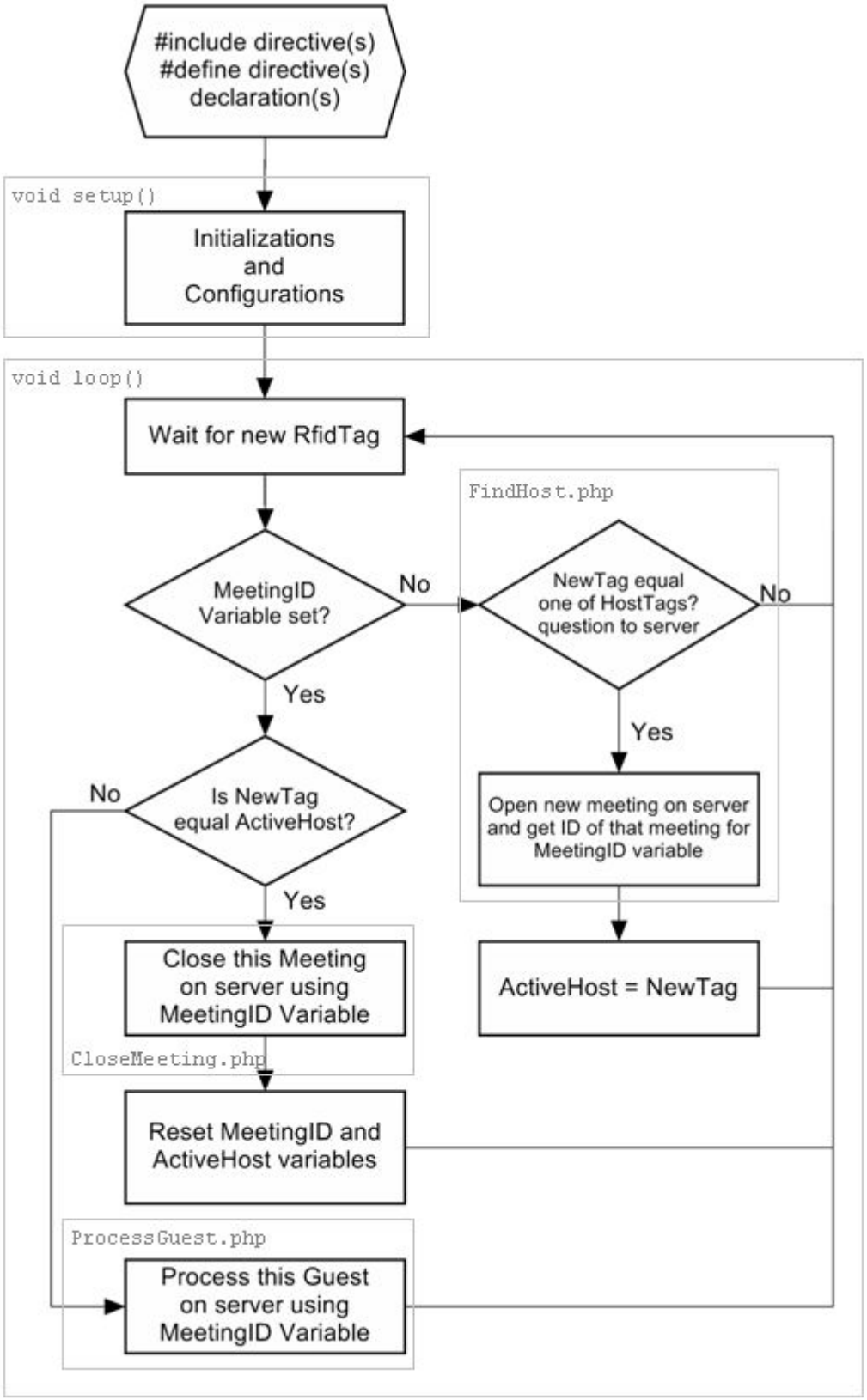

Figure 14. Main sections of Arduino program for ID-Reader-Unit

- On reading a new tag, the Arduino program will determine the current operating state by examining the value of MeetingID variable. An initialized MeetingID variable means that a meeting (or session) has already started and an unset MeetingID variable means no meeting is going on.
- If no meeting is going on then the Arduino program has to figure out if this new presented tag belongs to a valid host person or not. In this case the Arduino program will call a server side script “FindHost.php” along with the ID number of the new tag.
- In response to this call, the server will either return a negative reply if the new tag does not belong to a valid host or the server will create a new meeting and return the MeetingID number of the newly created meeting if the new Tag belongs to a valid host.
- If the server reply is negative, the MeetingID variable will remain uninitialized. In other cases the returned ID number will be stored in the MeetingID variable. The new tag’s number will be stored in another local variable named ActiveHost.

In the second case if the MeetingID variable is already initialized, then a new tag input will be processed as follows:

- Now the Arduino program has to figure out, if this new tag is presented by the same host person who started this meeting or not. This will be done by comparing the new tag ID number with the ActiveHost variable.
- If the new Tag is not equal to the ActiveHost variable, it means that the new Tag has come from a guest (or participant). In this case the system will call a server side script named “ProcessGuest.php” along with the ID number of this guest and the value of MeetingID variable.
- If the NewTag ID number is equal to the ActiveHost variable, this means the meeting host person wants to close the meeting. In this case the Arduino program will ask the server to close the on-going meeting by calling another server side script named “CloseMeeting.php” along with the value of the MeetingID variable (which was initialized at the start of this meeting). Next the system will unset the two control variables MeetingID and ActiveHost.

After processing a tag input the Arduino program will wait for another new tag input.

4.3 ID-Reader-Unit Related Server-Side Processing

The ID-Reader-Unit related server side processing is done using three sever-side scripts; FindHost.php, ProcessGuest.php, and CloseMeeting.php. All of these scripts will connect to the database through a predefined account idreader to perform their database-related tasks. A flow diagram of the FindHost script is shown in Figure 15. Input for this script is the RFID value of NewTag sent from the ID-Reader-Unit using the HTTP GET request method. The script will search for this tag in the meeting_host table of database. If no match is found then the script will send "NO" in reply to the ID-Reader-Unit. If a match is found then script will search for a plan for this host in meeting_plan table for this time. If a plan exists then the script will insert a record in the meeting table for this host and the title will be as provided in the meeting_plan table. If no plan exists then title will be “untitled”. Then the script will send the ID number of this newly opened meeting from the meeting table to the ID-Reader-Unit.

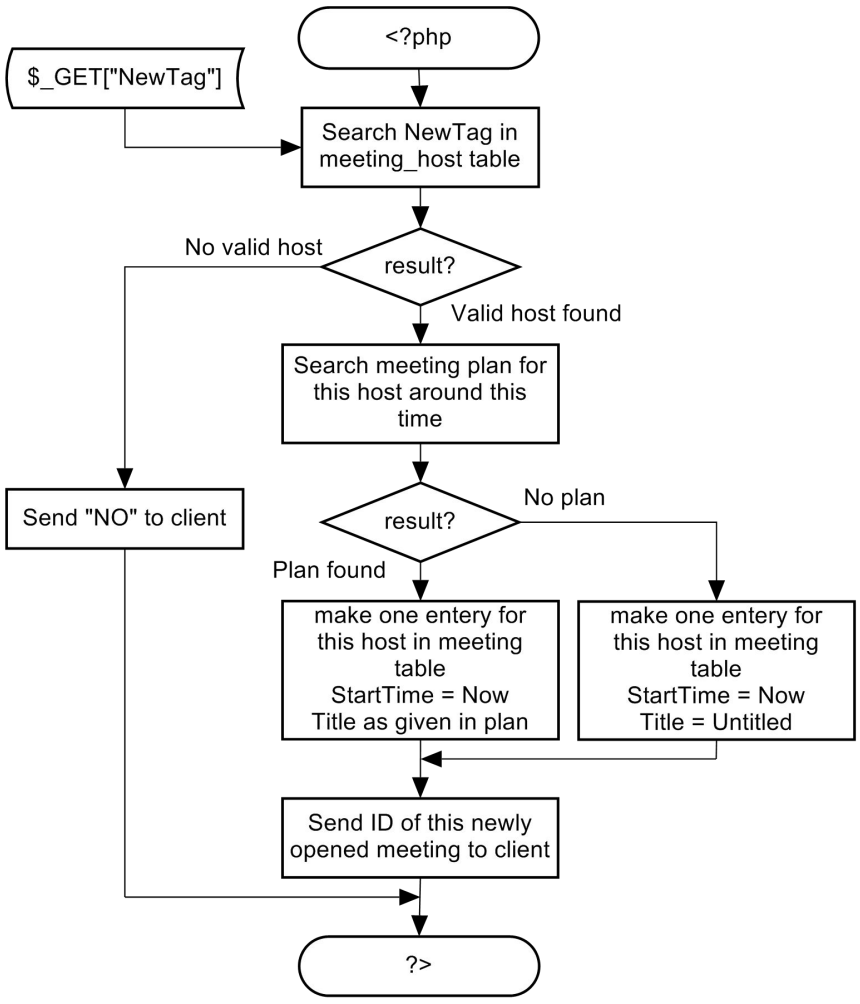

Figure 15. Main sections of FindHost.php server-side script

A flow diagram of the ProcessGuest script is shown in Figure 15. This script will receive two inputs from the ID-Reader-Unit; RFID value of NewTag, and MeetingID number. This script will search mg_data_buffer table to see if this guest has already signed out for the given meeting number. If yes then the script will end without any further processing. If this guest has already signed in then the script will sign him/her out in the mg_data_buffer table using the current time.

If there is no record for the NewTag number in the mg_data_buffer table then the script will search this tag in the guest table. If found then this guest will be signed in using MeetingID, guestID, RFID, and the current time. If not found then this guest will be added first to the guest table with a default name of New Partic. and next he/she will be singed in the mg_data_buffer table.

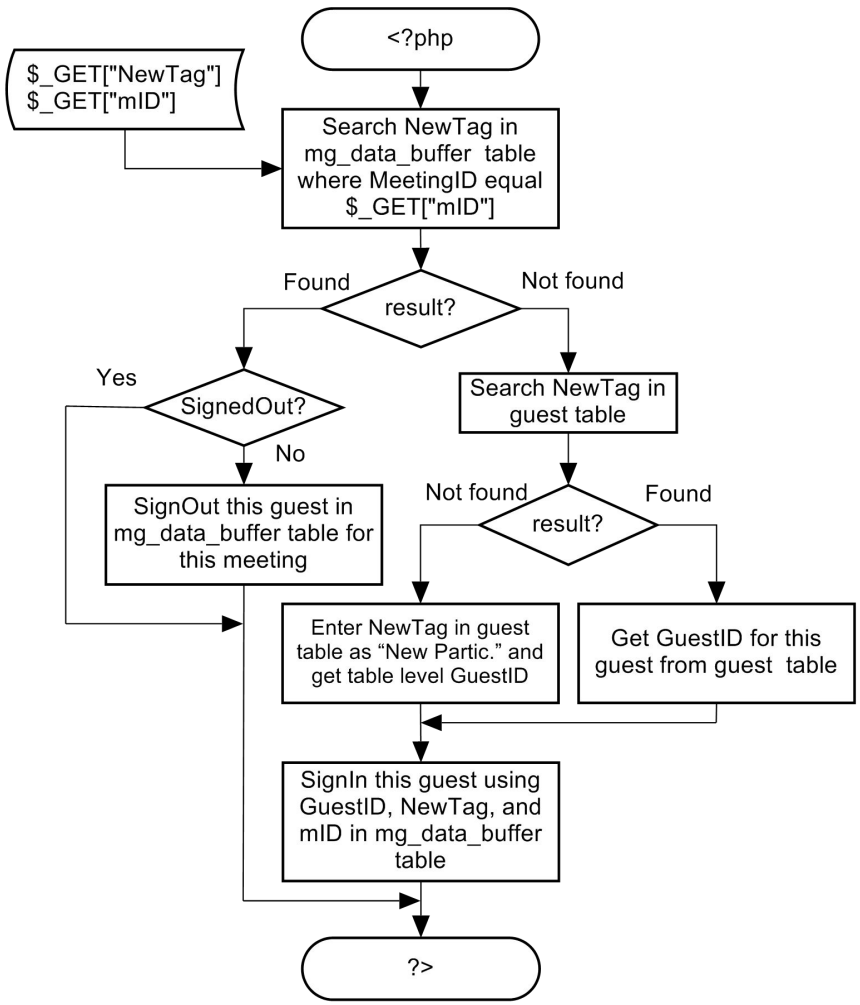

Figure 16. Main sections of ProcessGuest.php server-side script

A flow diagram of the CloseMeeting script is shown in Figure 15. The input for this script is the MeetingID number sent from the ID-Reader-Unit. This script will copy all records of the given MeetingID from the mg_data_buffer table to the mg_data table. The attributes included in copy operation are: MeetingID, GuestID, GSignIn, and GSignOut time. Next all of these records will be deleted from the mg_data_buffer table. Finally the script will enter current time in the EndTime attribute in the meeting table for the given MeetingID number.

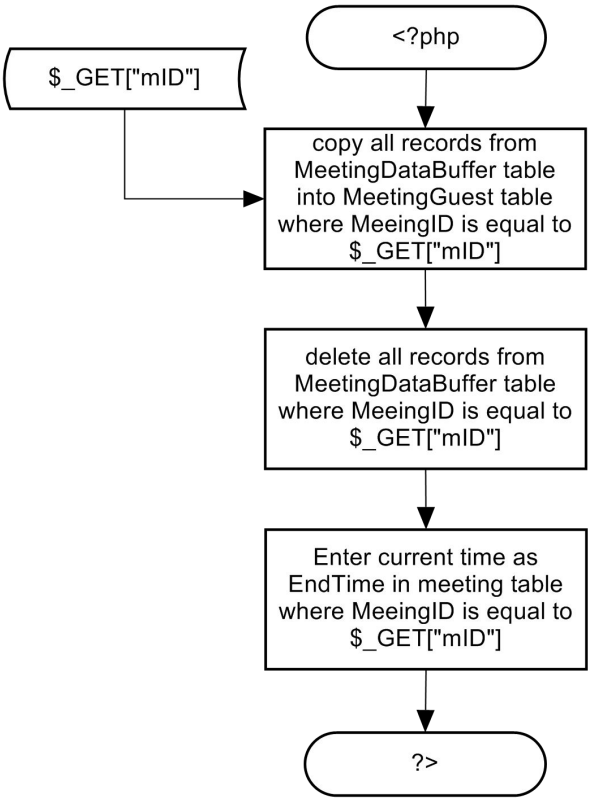

Figure 17. Main sections of CloseMeeting.php server-side script

4.4 Administration and Information Display Setup

System administration and presentation of the stored information is provided using several PHP based WebPages. Hierarchical arrangement of these WebPages is shown in Figure 18. Here the first page index.php is available to everyone while all other pages are only accessible after a successful login. The index page takes username and password information and compares it with the information stored in the meeting_host table. On successful comparison two session variables are created and initialized. One session variable stores the group number and the other stores the database level ID of the user. These two variables are used as a log-in check and for dynamic content generation. At present, two groups are implemented. One group is admin having numerical value 2 and the other group is host having numerical value 3.

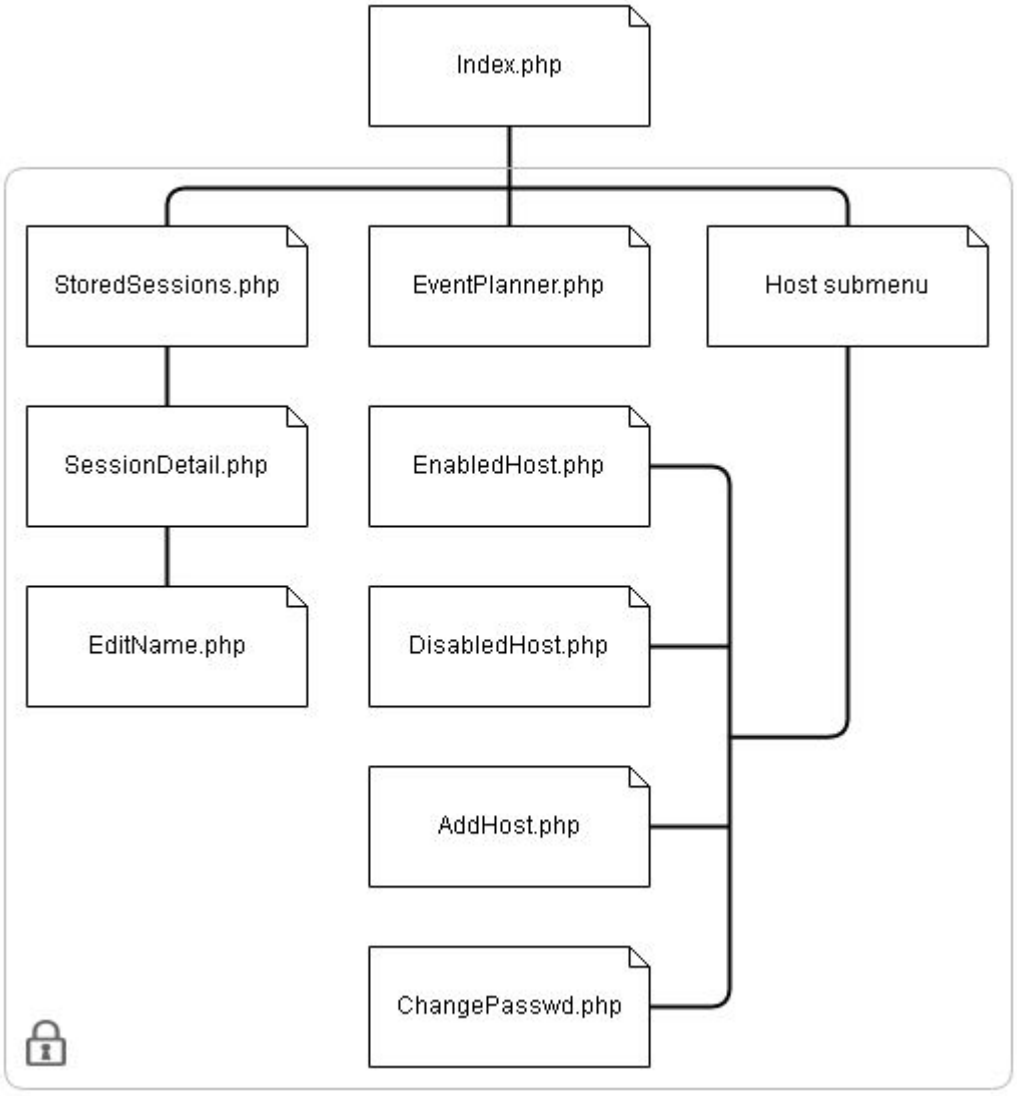

Figure 18. System administration / information display pages

After proper login, the user is redirected to “StoredSession.php” page. In case of the admin user (group# 2), this page will show a list of the latest 50 sessions conducted by all host persons. Here the admin user also has the option of viewing the stored sessions of a specific host person by selecting him/her from a dropdown list. If the user is a basic host person (group #3) the system will show the latest 50 sessions conducted by this host only. Both admin and host users can choose if they want to see a list of all or latest 50 stored sessions. The session data delete option is also provided to the admin user only.

The presented list of sessions contains four columns: Title of the session, Hosted by (name of host person), Start Time, and End Time. By clicking on the title of a session, another page will open named SessionDetail.php. This page shows a record of the participants (name, sign-in and sign-out time) who participated in this session. The session title (and location) is editable on this page after a single click on it. If a participant is new, his/her name will appear as “New Partic.”. Participants’ names are editable through EditName.php page.

On EventPlanner.php page, the admin user can define an upcoming session by submitting starting time, host name, title, and location of the session. All host names are provided in the form of a dropdown list. This page also shows a list of earlier defined sessions. In the case of a basic host user, this page will only show a list of those sessions defined for this user only. A timely conducted session will be deleted automatically from this list.

There is a hyperlink menu provided on top of every page with links to Stored Session page, Event Planner page, Host-sub-menu, and logout page. By clicking on the Host-sub-menu link, a submenu will open with links to Enabled-Host, Disabled-Host, Add-Host, and Change-Password pages for the admin user. In the case of a basic host user the submenu will only provide link to the Change-Password page. On the Add-Host page the admin user can create a new host account by entering RFID, First Name, Last Name, Login Name, and Password information. A newly added host is enabled by default. He/she can conduct a session or log into the Formal gather plus web server.

The enabled-Host page shows a list of all enabled host persons. Here the administrator has two options one of which is disabling a host account. A disabled host cannot conduct a session neither log into the system. The other option is to apply a new

password to a host account if that person has forgotten his/her password. The Disabled-Host page provides a list of all disabled host accounts. Here the admin can re-enable a host account if required.

All users can reset their own passwords from the Change-Passwd page. On this page they have to type in their current password once and a new password twice. After completing their work users can logout by a click on the logout hyperlink. During the logout process all the session variables which were created at the beginning will be destroyed. Data sources and destinations for all pages are shown in Table 3.

Table 3. Data sources and destinations at page level

Page	Source table in D.B.	Destination table in D.B.
index.php	meeting_host	
StoredSessions.php	meeting, meeting_host	
SessionDetail.php	meeting, meeting_host, mg_data, guest	meeting
EditName.php	guest	guest
EventPlanner.php	meeting_plan	meeting_plan
EnabledHost.php	meeting_host	meeting_host
DisabledHost.php	meeting_host	meeting_host
AddHost.php		meeting_host
ChangePasswd.php	meeting_host	meeting_host

5 Discussion

Here the admin user is going to log in (Figure 19). For event host users there will be changes in interface as described in chapter 4.

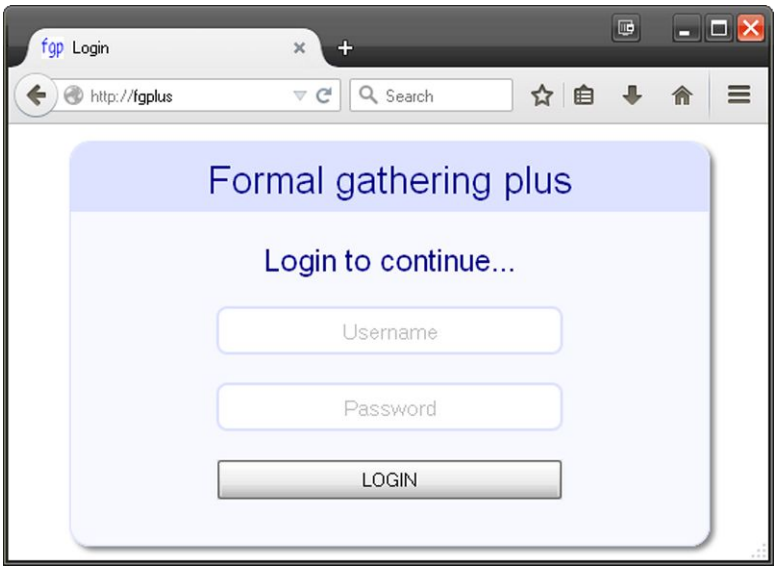

Figure 19. Login page

The next page after successful login is show in Figure 20. If the given user name / password is invalid then an error message will be displayed on top of the login page.

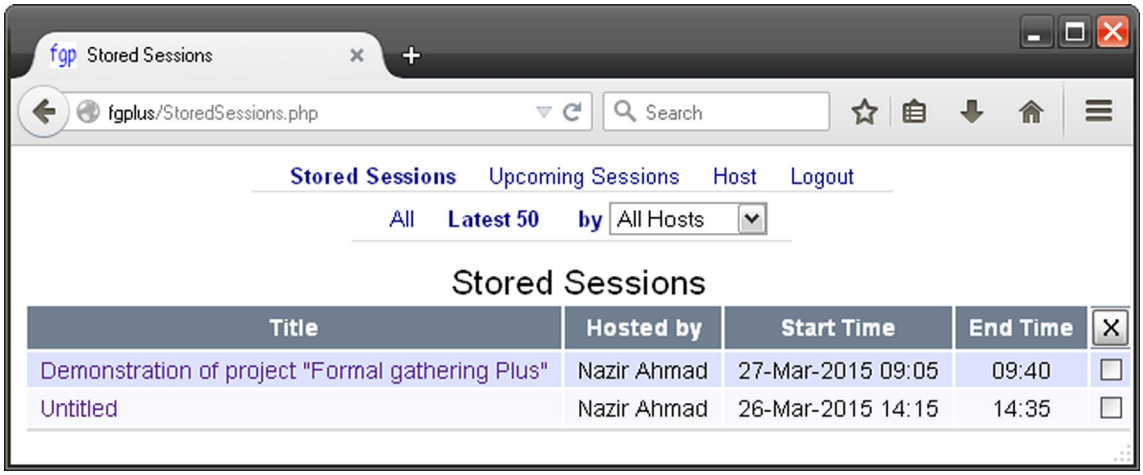

Figure 20. Stored Sessions page

On the stored sessions page (Figure 20), at present there are only two options provided for selecting the size of the list (latest 50 or all sessions). Later more options can be implemented. For example, the user should be able to decide how much

records he/she wants to view on one web page such as 25, 50, 75 or 100. In addition, there should be buttons for the first 25, the next 25, the previous 25, and the last 25 records. A per month view option can also be implemented where a user can pick a month from a dropdown list. A search option can be included along with choosable search criteria for example as provided on the Youtude website where a user can choose from a number of search options such as upload date or duration.

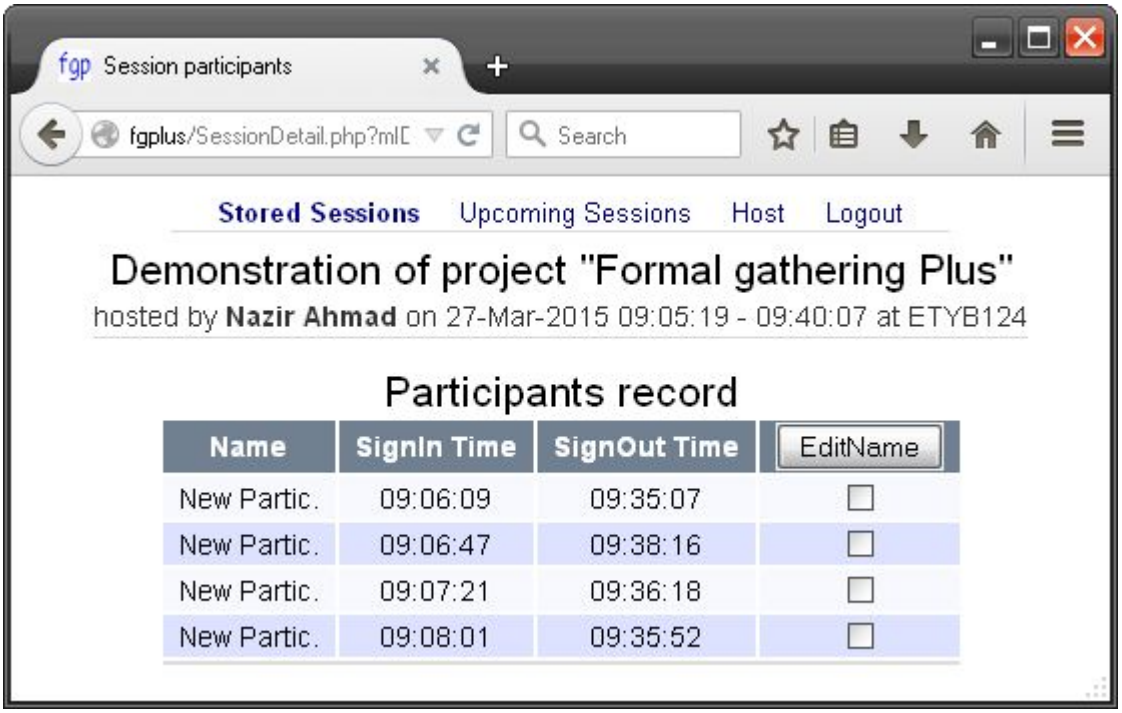

Figure 21. Details of selected session

In the session detail page (Figure 21), the color of the session's title will change into blue and the mouse pointer will turn into a hand symbol on the "mouseover" event, for signaling to the user that this text is not static. Then by the "onclick" event, a JavaScript prompt box will open with the current title already loaded in it. Here the user can edit the current title or enter a completely new title. The changed title will be posted to this same PHP script SessionDetail.php. This script will update the given title in the database and then provide updated information back to the user. The update functionality for location of event (*at the end of line below main title*) is also implemented in the similar way. The user can go forward to the participant name editing page (Figure 22) by selecting at least one participant checkbox and then clicking on the EditName button.

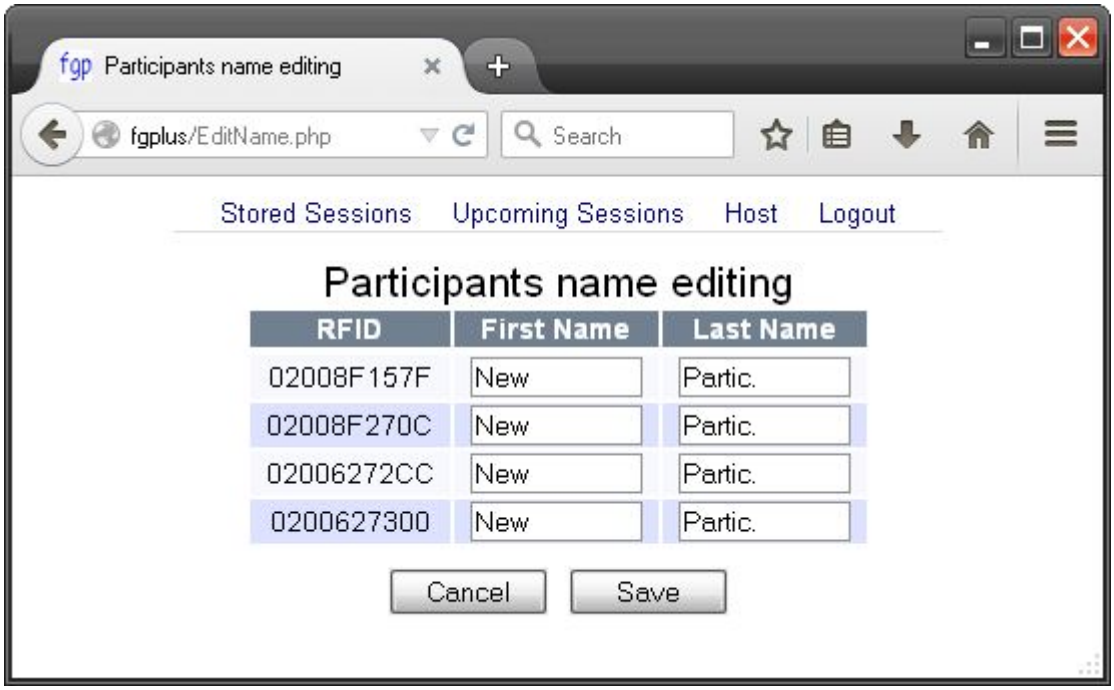

Figure 22. Participants' name editing

Currently participants can record their presence by simply presenting their RFID to the ID-Reader-Unit during an open session. No restriction is applied on participants for a specific event through the system itself. This functionality can be considered during further development. Also one more page can be made through which RFID tags can be issued by a dedicated officer and the participant's name would start appearing from day one. In parallel, this implemented functionality where the participant's name has to be entered once through the participants' name editing page is shown in Figure 22.

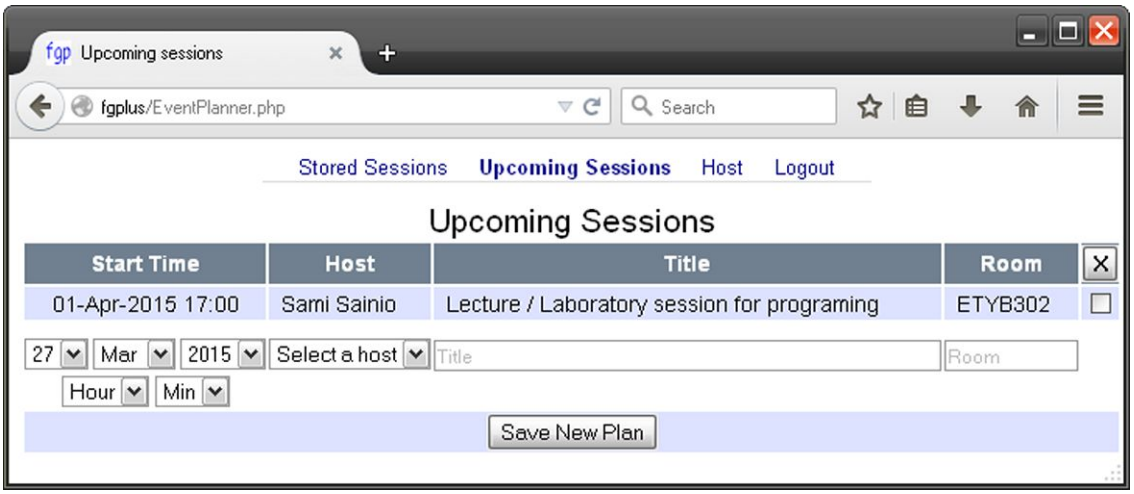

Figure 23. Event planner page with a list of predefined event

The page of upcoming sessions (Figure 23) acts as a link between the administrator and event host persons. The host persons can gain information for any upcoming event assignment. Further, a feature can be added for sending an email notification to the host person (and also to a pre-selected set of participants) regarding a newly defined event. When the system admin will submit parameters of a new event/session, first the given date and time will be tested through a JavaScript function and only a future date and timing entry will be processed. The other three fields (host, title and room/location) are also compulsory for saving a new plan. Here a dropdown list of the most common locations for events can also be provided.

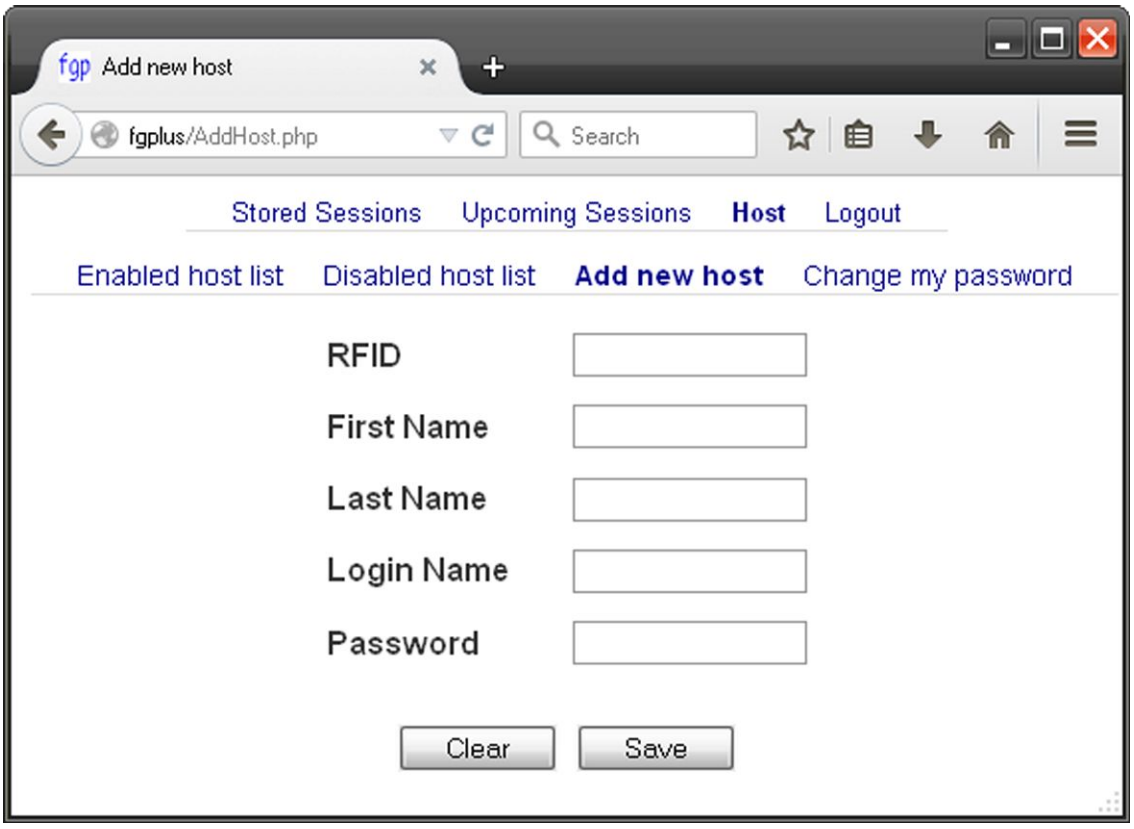

Figure 24. Adding a new host, page accessible to admin only

All the five inputs (in Figure 24) are compulsory for creating a new host person account. Values for other attributes related to the host account will be inserted automatically. The same RFID tag or Login name can not be added more than once. Passwords are stored in the database after the SHA1 encryption [32], which is why, the size of the password attribute (*in meeting_host table*) is set to 40 characters to accommodate the encrypted form of the password.

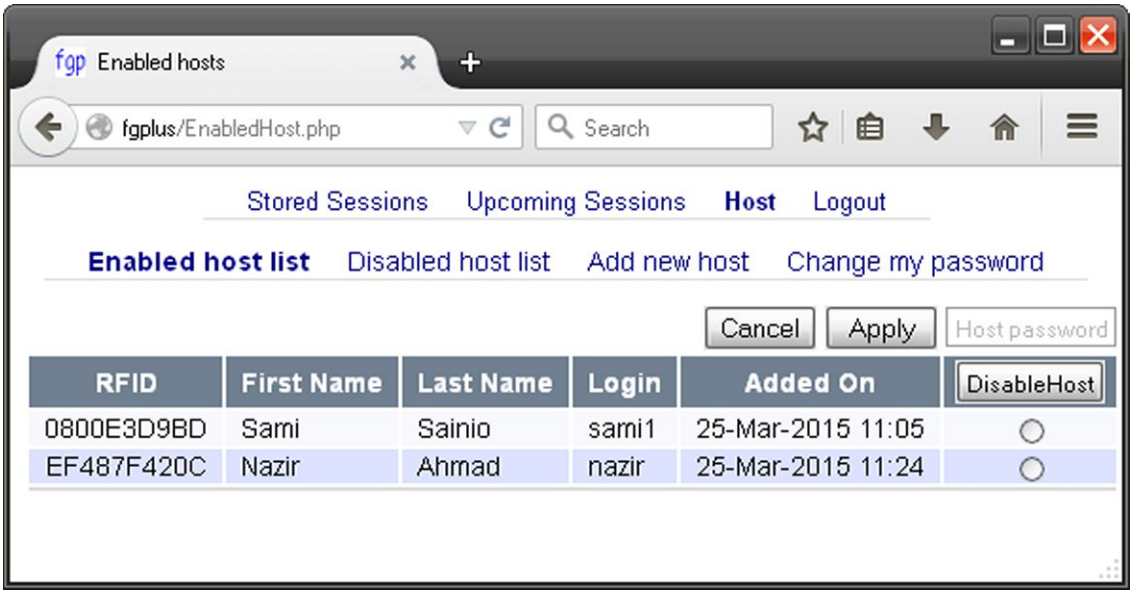

Figure 25. Enabled host list page (accessible to admin only)

The advantage of implementing the Disable-Host feature (Figure 25) is that a host account can be deactivated instantly without loosing the data of all those sessions which have been conducted by the same host. Later that host account can be reactivated from the same data state and with the same credentials, if required, through the page shown in Figure 26. This host enable/disable control is mapped on the status bit attribute of the meeting_host table. This attribute is checked during the login process and when a host person tries to open a session through ID-Reader-Unit. In further development, the delete functionality can also be implemented for deleting extra host accounts.

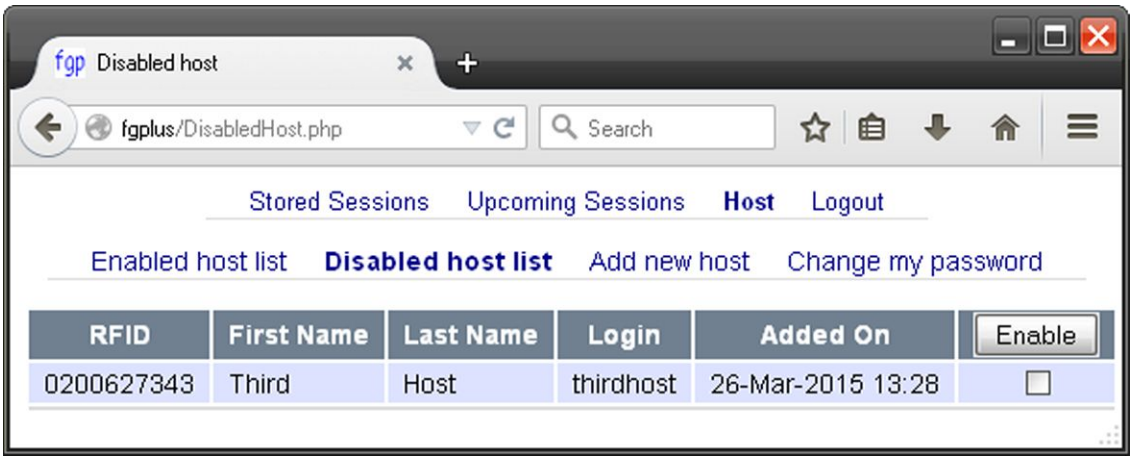

Figure 26. Disabled host list page (accessible to admin only)

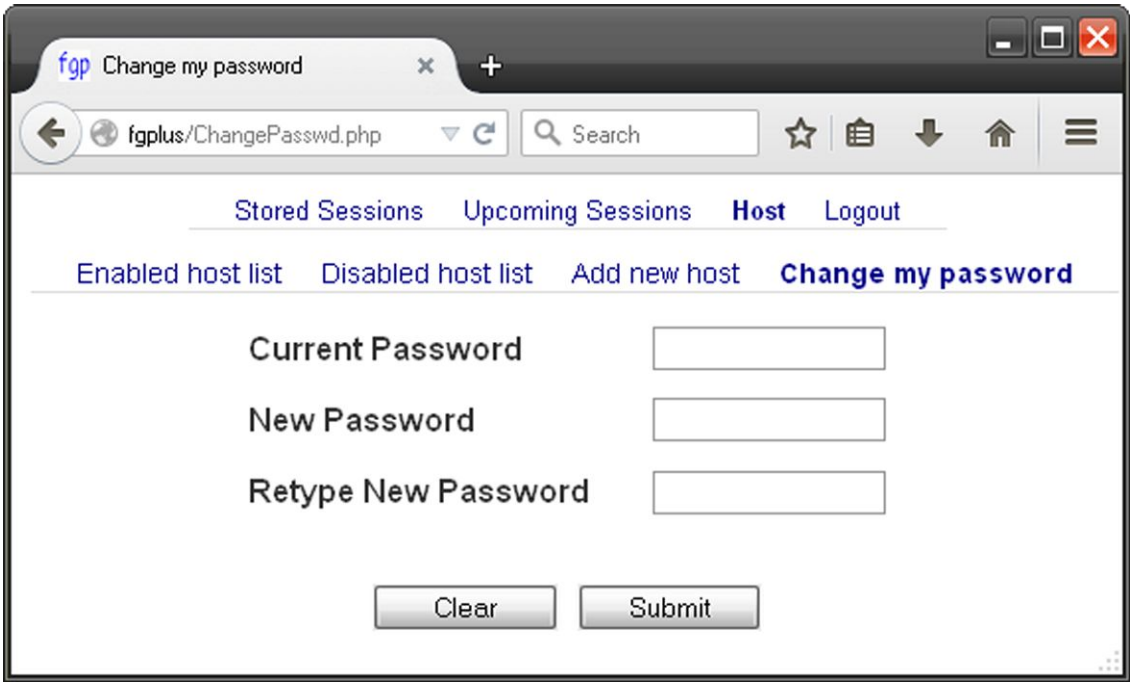

Figure 27. Password changing page

The password changing option (Figure 27) is provided in the similar style as generally provided on authentication requiring websites. With a proper current password input, the system will go ahead to apply a new password. The minimum acceptable size of the password is set to five characters. The allowed character set includes small or upper-case alphabets and digits 0 to 9 only. In addition to the present set of web pages, another page can be made where a host person could write notes about a session and the participants could give their feedback. Also a page can be made to view the participation history of a participant.

Currently, the ID-Reader-Unit has only two LED indicators for communicating with host person and participants. A small LCD screen can be considered a replacement for LEDs. Helpful messages can be shown on the LCD screen according to the ongoing situation for example;

- Waiting for host
- Session open
- Participant signed-in
- Participant signed-out
- Session closed

In order to test the processing speed of the system a small test session is conducted. Here two participants signed in and later signed out as follows:

- First, one tag of a participant was presented to the RFID reader unit. This tag was taken away right after viewing acknowledgement from the Tag-LED indicator (first participant signed in).
- Soon after, the second tag of another participant was presented to the RFID reader and taken away similarly (second participant signed in).
- Next both participants were signed out quickly by presenting the tag of the first participant and later the tag of the second participant.

Here is the session detail page of this small speed-testing session (in Figure 28):

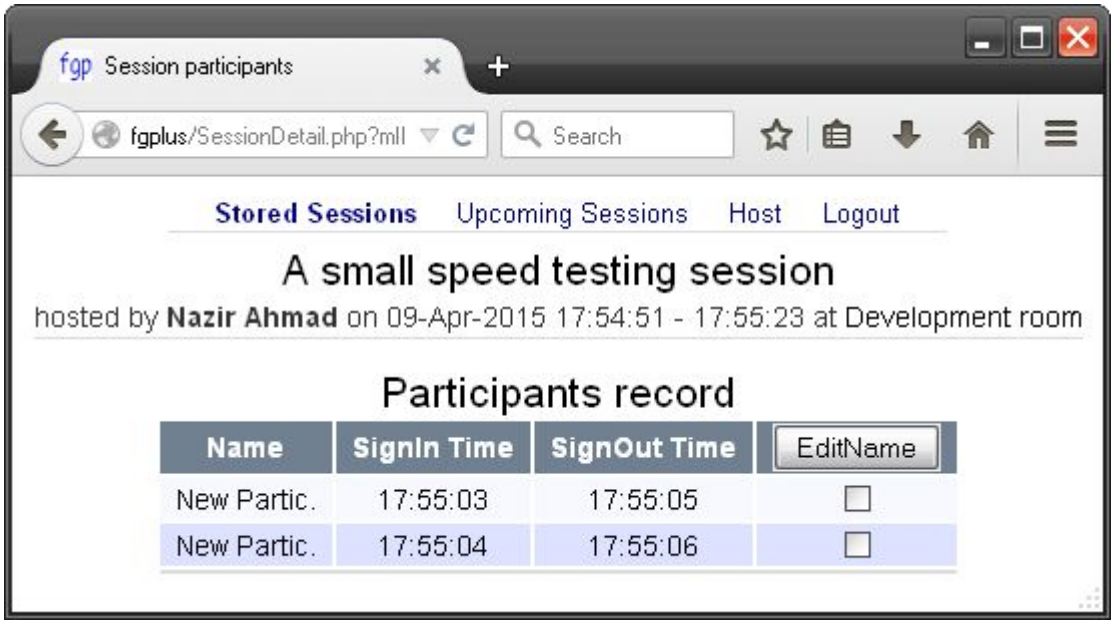

Figure 28. Details of speed testing session

During this test session the ID-Reader-Unit was directly connected to the server machine (*Dell D630*) using a patch cable (*Dell 2R512*). The Participants' record for this session shows that each tag input was processed and saved in the duration of approximately one second. This shows that if there are 60 participants lined up properly and holding their RFID tags already in hand, they can sign-in in a time slot of 1+ minute approximately. However in real life the chance is very little that 60 or any number of the participants will be attempting to sign in or sign out so quickly.

6 Conclusion

The goal of this bachelor's thesis project was to design and develop an ICT application that would act as a helpful tool in achieving the objectives of a formal gathering for example; event, seminar, meeting, concert, class or convocation. At the end of the current phase of development all of the initial functional requirements have been implemented with a few enhancements: for example, recording participants' data with ease and speed, and later presenting them in a well organised way to authenticated users through a web interface, event-host account management, defining a future event, and editing of some attributes of stored information such as subject and location of event, participants' names. Each of the implemented features has been successfully tested and the product is ready for utilization.

A number of possibilities were also highlighted for further development, like a web-page for allotment of RFID tags to participants by a fixed officer, sending a notification to an event's host person and target participants regarding an upcoming event, a page for notes and feedback text input, and inclusion of an LCD screen in ID-Reader-Unit for providing more user-friendly acknowledgment upon every RFID tag input. Furthermore, customer feedback, in-housing analytical testing, and a periodic review of the fundamental selected and similar technologies will be necessary for ensuring the success of the product. Lesser resource requiring and most demanding features by customers could be considered first for implementation.

References

1. ICT Educator Conference (7 January, 2011), Mid-Pacific ICT Center
   [online] [Accessed 2 January 2015]
   URL: http://www.mpict.org/newsletters/q111/article_winter_ict_educator_conference_a_success.html
2. Glossary - ICT (2012), The University of Nottingham
   [online] [Accessed 2 January 2015]
   URL: http://www.nottingham.ac.uk/gradschool/sict/toolkit/glossary/
3. Types of New Products (2005), Guangzhou Tai Feng Yuan Enterprise Co., Ltd.
   [online] [Accessed 2 January 2015]
   URL: http://www.tfy.com/product_development/product_types.htm
4. Arduino - Introduction (2015)
   [online] [Accessed 19 January 2015]
   URL: http://arduino.cc/en/guide/introduction
5. Arduino Uno - R3 - DEV-11021 - SparkFun Electronics
   [online] [Accessed 19 January 2015]
   URL: https://www.sparkfun.com/products/11021
6. Arduino Build Process, (2015), Learning - Arduino
   [online] [Accessed 19 January 2015]
   URL: http://arduino.cc/en/Hacking/BuildProcess
7. Arduino - Blink (2015), learning example - Arduino
   [online] [Accessed 19 January 2015]
   URL: http://arduino.cc/en/Tutorial/Blink
8. Arduino Ethernet Shield - DEV-09026 - SparkFun Electronics
   [online] [Accessed 19 January 2015]
   URL: https://www.sparkfun.com/products/9026
9. RFID - FAQ (2009) ALEXAN TECH. INC
   [online] [Accessed 22 January 2015]
   URL: http://www.alexan-tech.com/faq.html

10. RFID Card Reader Documentation (version 2.3) - Parallax Inc
[online] [Accessed 22 January 2015]
URL: http://www.parallax.com/sites/default/files/downloads/28140-28340-RFID-Reader-Documentation-v2.3.pdf

11. Ian Poole (2015), "RFID Frequencies and Frequency Bands", (Radio-Electronics.com) [online] [Accessed 22 January 2015]
URL: http://www.radio-electronics.com/info/wireless/radio-frequency-identification-rfid/low-high-frequency-bands-frequencies.php

12. RFID Card Reader - Serial | 28140 | Parallax Inc (OEM)
[online] [Accessed 22 January 2015]
URL: http://www.parallax.com/product/28140

13. The Apache HTTP Server Project - Apache Software Foundation
[online] [Accessed 26 January 2015]
URL: http://httpd.apache.org/

14. World Wide Web Technology Surveys - W3Techs
[online] [Accessed 26 January 2015]
URL: http://w3techs.com/

15. Introduction - PHP Manual (2015) The PHP Group
[online] [Accessed 26 January 2015]
URL: http://php.net/manual/en/intro-whatis.php

16. Download MySQL Community Server - Oracle Corporation
[online] [Accessed 26 January 2015]
URL: http://dev.mysql.com/downloads/mysql/

17. Kevin Yank (January 24, 2002), "Build Your Own Database Driven Website Using PHP & MySQL", Database Journal [online] [Accessed 26 January 2015]
URL: http://www.databasejournal.com/features/mysql/article.php/1402281/Build-Your-Own-Database-Driven-Website-Using-PHP--MySQL-Pt-4.htm

18. XAMPP package - Apache Friends Organization
[online] [Accessed 26 January 2015]
URL: https://www.apachefriends.org/index.html

19. Resham Parikh (29 August 2014), "Data Modeling Basics: Definition and Usage", (Real-time apps, simplified. Syncano) [online] [Accessed 21 March 2015]
URL: http://www.syncano.com/data-modeling-basics/

20. Data Modeling - Conceptual, Logical, And Physical Data Models, (2015), (1Keydata Tutorials) [online] [Accessed 21 March 2015]
URL: http://www.1keydata.com/datawarehousing/data-modeling-levels.html

21. Vangie Beal (2015), “entity-relationship model (diagram)“, (tech dictionary, Webopedia) [online] [Accessed 21 March 2015]
URL: http://www.webopedia.com/TERM/E/entity_relationship_diagram.html

22. Gary F. Andrus, (2015), "Primary and Foreign Keys", (DePaul University) [online] [Accessed 21 March 2015]
URL:http://condor.depaul.edu/gandrus/240IT/accesspages/primary-foreign-keys.htm

23. Using FOREIGN KEY Constraints - MySQL 5.6 Reference Manual [online] [Accessed 21 March 2015]
URL: http://dev.mysql.com/doc/refman/5.6/en/create-table-foreign-keys.html

24. Rob Gravelle, (12 May, 2010), "Top 10 MySQL GUI Tools", (Database Journal) [online] [Accessed 21 March 2015]
URL: http://www.databasejournal.com/features/mysql/article.php/3880961/Top-10-MySQL-GUI-Tools.htm

25. Arduino Uno R3, (2015), Product page - Arduino [online] [Accessed 29 March 2015]
URL: http://www.arduino.cc/en/Main/ArduinoBoardUno

26. Arduino Ethernet Shield R3, (2015), Product page - Arduino [online] [Accessed 29 March 2015]
URL: http://www.arduino.cc/en/Main/ArduinoEthernetShield

27. SPI library, (2015), Arduino Reference Manual [online] [Accessed 29 March 2015]
URL: http://www.arduino.cc/en/Reference/SPI

28. Plastic Enclosure 1591S, Hammond Manufacturing Ltd [online] [Accessed 29 March 2015]
URL: http://www.hammondmfg.com/pdf/1591S.pdf

29. Arduino - Language Reference, (2015) [online] [Accessed 12 April 2015]
URL: http://www.arduino.cc/en/Reference/HomePage

30. Arduino - Libraries, (2015)
[online] [Accessed 12 April 2015]
URL: http://arduino.cc/en/reference/libraries

31. Arduino Build Process, (2015), Learning - Arduino
[online] [Accessed 12 April 2015]
URL: http://arduino.cc/en/Hacking/BuildProcess

32. Encryption and Compression Functions, MySQL 5.5 Reference Manual
[online] [Accessed 25 April 2015]
URL:https://dev.mysql.com/doc/refman/5.5/en/encryption-functions.html#function_sha1