A REPORT

ON

‘DIGITIZATION OF DMRC WITH RESPECT TO DEPRECIATED PROCEDURES’

BY

Kartik Sethi (2014A7PS130H)

PREPARED IN PARTIAL FULFILMENT OF

PRACTICE SCHOOL-1

AT

DELHI METRO RAIL CORPORATION LIMITED, DELHI

A PRACTICE SCHOOL-I STATION

OF

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

MAY - JULY 2016

दिल्ली मेट्रो रेल कॉरपोरेशन लिमिटेड
Delhi Metro Rail Corporation Limited

ACKNOWLEDGEMENT

It has been a great honor and privilege to undergo training at **Delhi Metro Rail Corporation Limited, Delhi.**

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along the completion of our project work. In finalizing our Project, we sincerely thank for the invaluable assistance and guidance provided by experts and officials of Delhi Metro.

We are also very much thankful to **Mr. Rajneesh Pandey** for providing all the facilities and support to meet our project requirements.

We would like to take opportunity to express our humble gratitude to **Mr. Sajeev Maheshwari and Ms. Sasha Luthra,** under whom we executed this project. Their constant guidance and willingness to share their vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

We are highly thankful to our Practice School instructor, **Mr. Dinesh Yadav** and our co-instructor, **Mr. Karthik Srinivasan**, for their invaluable guidance which helped us understand the project better.

Although there may be many who remain unacknowledged in this humble note of gratitude, there are none who remain unappreciated.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
PRACTICE SCHOOL DIVISION

Station: Delhi Metro Rail Corporation Limited **Centre:** Barakhamba Road, Delhi
Duration: 7 weeks **Date of Start:** 23rd May, 2016
Date of Submission: 15th July, 2016

Title of the Project:
'DIGITIZATION OF DMRC WITH RESPECT TO DEPRECIATED PROCEDURES'

Students involved in the Project:

Anirudh Kumar Bansal	2014A7PS081P
Ayush Sawarni	2014A3PS253P
Kartik Sethi	2014A7PS130H
Rishabh Mittar	2014A7PS141P
Siddharth Agarwal	2014A7PS025P

Expert(s): Mr. Sajeev Maheshwari (AGM/IT), Ms. Sasha Luthra (Asst. Manager/IT)
PS Faculty: Mr. Dinesh Yadav (instructor), Mr. Karthik Srinivasan (co-instructor)

Key Words: Android Application, DMRC, GPS (Global Positioning Service), Attendance, Web Portal, Fault Reporting System

Project Areas: Android Application Development, Frontend and Backend Web Development

Abstract:

This project report highlights the working of digitized systems for simplifying reporting procedures at DMRC. The **Android Application** is used to create a GPS based attendance system for the DMRC employees. It also highlights the functions of a **Web Based Application**, to efficiently file complaint reports of faults occurring at various metro stations.

For the **Android application**, the employees need to login through the app and send a report of their work to their supervisor(s). Simultaneously, the location of the employee is captured through the GPS or through his/her mobile network, on click of a button and the report filled by the employee is sent to his/her respective supervisor through an internal mailer.

Through the **Web Based Application,** the station controller will report about a fault in one of the equipment/machines installed at his/her respective metro station. A detailed report of the complaint will be sent to the concerned repairmen (through email and application). Similarly, after rectifying a fault the repairman can send a diagnostic report back to the station controller.

Date:

PS Instructors:
Dinesh Yadav
Karthik Srinivasan

TABLE OF CONTENTS

INTRODUCTION

Modern handheld devices such as smart phones and PDAs have become increasingly powerful in recent years. Spectacular advances in processing power along with the number of additional features included in these devices have opened the door to a wide range of commercial possibilities. Now there are various smartphone applications for different jobs as well for recreational purposes, which one could install in his/her phone.

An application that can be used for an effective reporting and attendance is '**DMRC Attendance Application**' developed for Google Android phones. The prime objective of

'DMRC Attendance Application' is to create a full-fledged attendance and reporting system for the DMRC officials. In DMRC, the employees belonging to different departments are supposed to go to different project sites to do their work. These sites are present at different geographical locations. An employee can therefore wrongly inform the supervisor of his/her attendance at a project site, as the supervisor may not be present at the location site at all times. Hence there arises a need to have a GPS (Global Positioning Service) attendance system for employees to curb such kinds of notorious activities.

The process of fault reporting in machines/equipments installed at various metro stations involves the station controller to notify the Operation Control Centre (OCC) about the error, which logs the complaint and then further informs the repairmen about such complaints. Then the repairmen again inform the OCC about the diagnostic of the error.

A **Web based application** (**CSS Complaint Portal**) was developed to make this process of fault reporting, fault diagnostic and rectification much simpler and easier.

OBJECTIVE OF STUDY

The objective of building an **Android application** was to make the process of reporting of employees at a particular site location, easy and genuine. In the hierarchy of construction and other related works, a department at higher level needs a confirmation from the lower level, to start its own work. This application can provide a platform so that the employee at a lower level in the hierarchy can notify and report about the completion of his/her required work so that the above levels in the hierarchy can start with their own respective work. Thus creating an efficient flow of information between employees and their respective supervisors.

The employee can report using the application after he/she is done with his/her work. A detailed report comprising of his/her location, time of reporting, employee's review and comments, etc. will be sent to his/her supervisor through an internal mailer. A GPS is also added in the application which is used to capture the location of the employee, when the employee reports from the application. This feature allows to maintain a check on the employee whether he/she has actually visited the site at the required time or not.

The objective of building the **Web Based Application (CSS Complaint Portal)** was to make the process of fault reporting and correction, painless and less complex. The main aim was to digitalize this process. Using the application, the station controller can easily lodge a complaint report about some fault in one of the machines present in his/her allotted station and the concerned repairmen will be directly notified (without the need of OCC) through an email notification, and once the repairmen are done diagnosing the type of error and the subsequent rectification, they can inform about the status of the equipment to the station controller through the web application itself

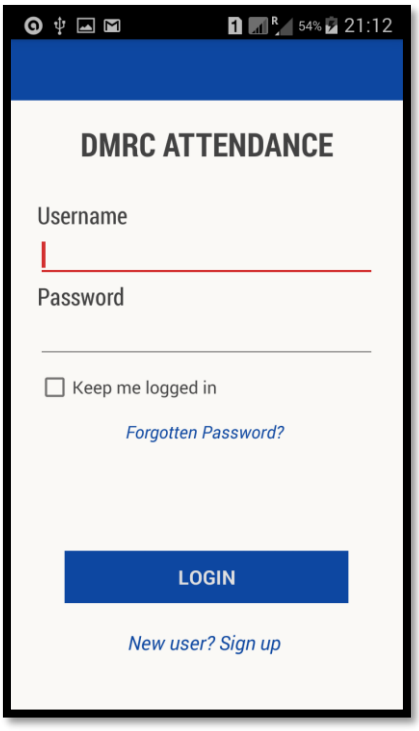
MAJOR COMPONENTS AND INPUTS INTO PROJECT

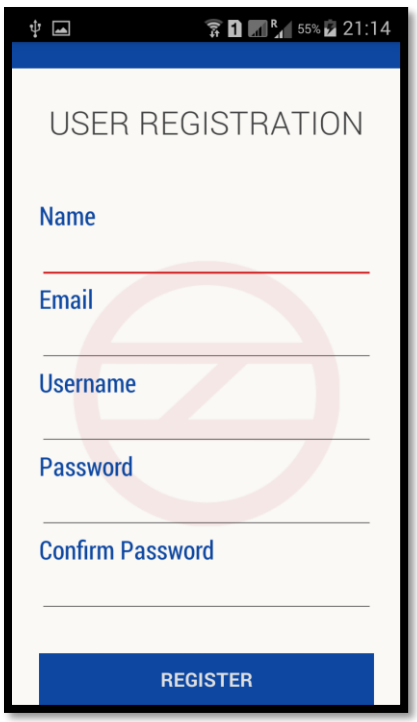
The major components of the **Android application** include:

1. **A User Registration and Login activity (page):**

The registration page is used to create a new entry for an employee in the application database hosted on the DMRC server. The employee needs to provide his name, username, password and email as input. Apart from these inputs, the application internally captures the IMEI (International Mobile Equipment Identity) which is unique for every phone.
All this information of the employee is taken and sent to the server where it gets stored in the database. Before submitting this data to database, it is checked that password and confirm password fields are same, all entries are filled in and email is in proper format. At server side, the database is checked, to ensure that a new user is not created using the same device or has the same username. If the IMEI number and username are unique then an employee is successfully registered in the database.

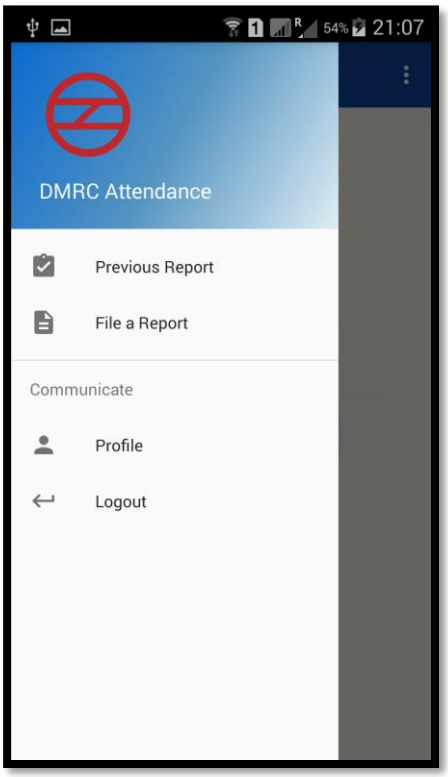
A user which has registered himself on the database can login in the app. The login activity requires the employee to add his/her username and password. The app will only allow login if both of the two fields (and the IMEI number) matches the entries in the user database. The IMEI number is automatically detected by the application (the user does not need to enter it). So as a result, Employee A cannot login into his/her account using Employee B's phone, which will make it essential for the registered user to log in from his/her own phone.





2. Navigation drawer

A navigation drawer is a rectangular box which contains various options to navigate to various activities (pages) present in the Android application (app).
The navigation drawer can be accessed once the user has successfully logged into the app. It can be accessed using the menu button (three horizontal bars) present on the left side of the Application Bar (topmost bar of the app) and it can be also accessed by swiping rightwards. It contains four options (Previous Reports, File a Report, Profile, Logout). Each of these options when clicked open a new fragment (screen).



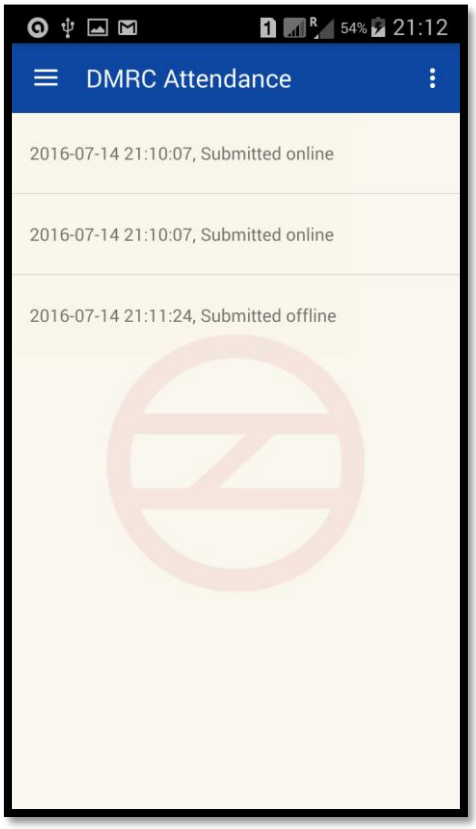
The four options available to the user are as follows:

- **Previous Reports**

This option provides the user to check and verify the reports submitted by him/her. After clicking on this option the user is taken to a screen which displays a list of reports. This may include both offline and online reports. Online reports are those that were successfully submitted by the employee to the server in the first attempt itself. Offline reports are those that could not be submitted in the first attempt due to server problem or internet connection problem. These reports are saved in the phone's local database and get automatically submitted to the server

whenever internet connection becomes available.

The list containing the submitted reports which gets displayed on clicking this option, contains their timestamp (time and date of submission) and also the information regarding whether it was submitted online or offline. On clicking the relevant item in the list, the employee can view all the details of the corresponding report, including the content of the report (the text as well as the image if sent).



- **File a Report**

This option enables the employee to submit a report to his/her supervisor. Upon selecting this option, the employee is directed to the report page where the user can enter a detailed report in the text area provided. There is also a camera icon below the text area, upon clicking which the user goes to the device's camera application and he/she can attach the clicked image along with the report.

When the user reports using the 'Report' button present in the bottom of this screen, his/her location's coordinates (latitude and longitude) are captured by the app in the background. So, along with the report content and the clicked image (if required), the user's location coordinates

are sent using GPS if it is available at that moment else these coordinates are obtained using cellular network from the service provider.

- **Profile**

This option can be used by the employee to view his/her profile page. This page welcomes the user and displays the employee's name, department and the list of his/her supervisor(s).

- **Logout**

This option can be used to logout the user from his/her account, directing the user to the login page.

The major components of the CSS Web Portal include:

1) **Login Page:** This page provides users with option to access the web portal. The user type is determined automatically from the User ID and respective home pages are loaded for every user type. This also provides with an option to reset password in case someone forgets it.

2) **WebMaster Page:** This is the home page for the WebMaster. It has options to add new users, systems, SUDB systems, equipments and sub-equipments. It also provides the option of viewing the webmaster's profile.

3) **Reporter Page:** This is the home page for the station controller. It provides an option for lodging a new complaint or view pending ones. Besides this, it gives option for viewing the user's profile.

4) **Repair Page:** This is the home page for the Repair Staff. It provides options to view received complaints and submit repair reports for the same. Besides this, it gives option for viewing the user's profile.

The major inputs into the project are:

1. **Software and Hardware:**

Android Studio

We used Android Studio 2.1.2**,** which is the official integrated development environment (IDE) for Android platform development, to build, debug and test our application. We also used the Android Software Development Kit (SDK) and JDK (Java development kit), which includes a variety of custom tools that help us develop mobile applications on the Android platform. Apart from this we used the built in android emulator of the Android Studio and our own android devices to test the application (app).

PHP MyAdmin (XAMPP server)

XAMPP was used to create a localhost server (a server on our local machine). A PHP script was also used to write the backend part of our application. The backend code included the functionality to insert a new record in a database and to find out the name of supervisor and department, checking user credentials during login etc.

Code Igniter:

Its a framework of PHP, it was used to provide an MVC framework to the application and make the scripts more secure.

Ngrok:

We used ngrok application to make our local server accessible over the internet and at the same time to get detailed debugging reports.

Bootstrap:

It was used to make the web application more elegant and also make it compatible with all screen sizes.

JQuery and Ajax:

These were used to make the web portals more responsive and reduce data load on the server.

2. **Android application development tutorials**

We used a variety of online and offline tutorials which were based on Android

application development. Example: the book ‘Headfirst Android Development’, Udacity beginners course on Android Development, The New Boston Android App Development tutorials, various YouTube channels, etc.

We also used the official android developers website, stack overflow, stack exchange, android hive and other online resources to help us learn new concepts and functionalities in android development and to help us resolve the bugs and errors that occurred along the way in the project.

We also used tutorials for JQuery, ajax and Bootstrap to learn the basics of making responsive web applications

3. **Internal mail server**:

The internal mail server was provided by DMRC. It is used to send a report entered by the employee to his/her supervisor along with his whereabouts when he/she has used the application to report to his/her supervisor.

DATA COLLECTION AND ANALYSIS

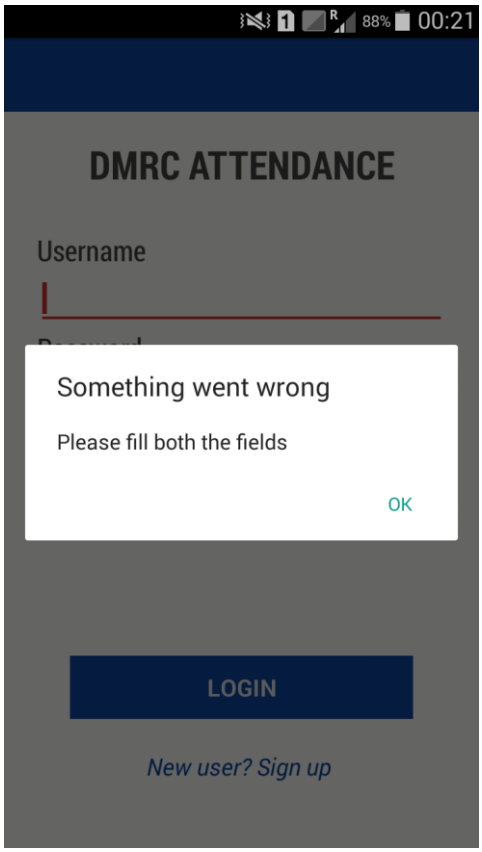
Analysis of the application:

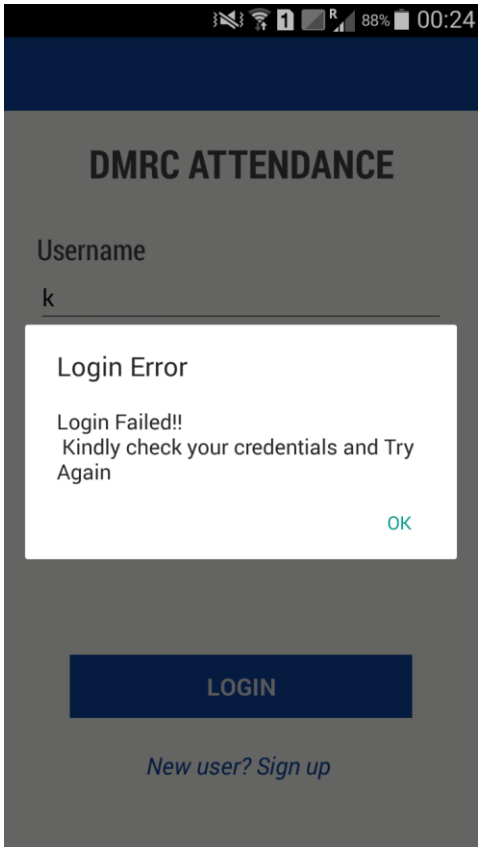
1. **Splash screen**:

   A splash screen with the DMRC logo is displayed along with a train animation, when the user opens the application. The duration of the splash screen is about three to four seconds after which the login activity begins or directly the profile (home activity) is displayed, depending on whether the user checked the 'Keep me logged in' check box button in his/her previous login attempt.

2. Login screen:

This activity is used to login into the application. If the user tries to login without filling all the fields then an error message is displayed. Similarly, if the user tries to login with someone else's phone or fail to input the right credentials then another dialog box is displayed, prompting the failed login attempt. On filling the proper credentials and clicking the 'LOGIN' button, the user credentials (username, password and IMEI number) are matched with the ones stored in the database and if it is a match then the BackgroundTask (part of Java code) is called and user is shown his/her profile (home activity).

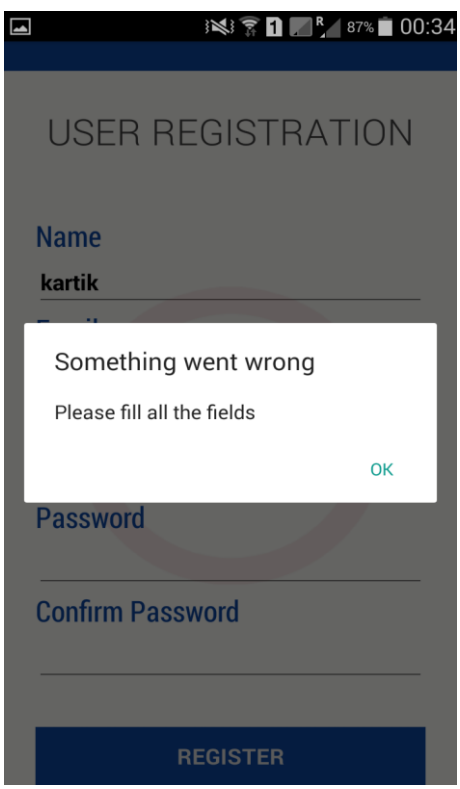


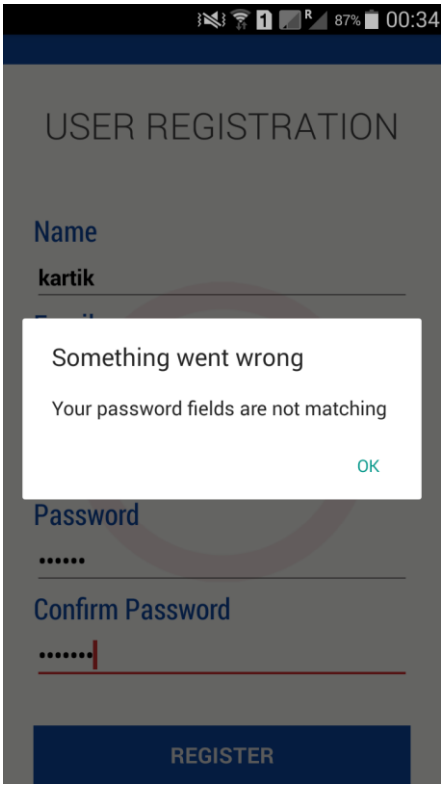


3. Registration activity screen:

It first checks whether all the fields are filled or not and whether the 'Password' and 'Confirm Password' fields are matching or not. It also checks whether email address is entered in the correct format (with @ sign). If not, then appropriate alert dialog boxes are displayed. Otherwise BackgroundTask (part of Java Code) is called with action (register),

name, username, password and IMEI (again found by the app itself) as arguments and the user is registered.





4. **BackgroundTask (part of the JAVA code of the app):**

This is a .java file which incorporates the AsyncTask Java class, which is created to write the code for the background tasks that happen in the application which the user cannot actually see, while he/she is interacting with the app. It has four methods- onPreExecute, doInBackground, onProgressUpdate (which is not used by us) and onPostExecute.

5. **Drawer Activity (Page):**

This page shows a navigation drawer at the side of the page with four options- previous report, file a report, profile and logout as were discussed in brief in the previous section.

6. **Report Activity (Page):**

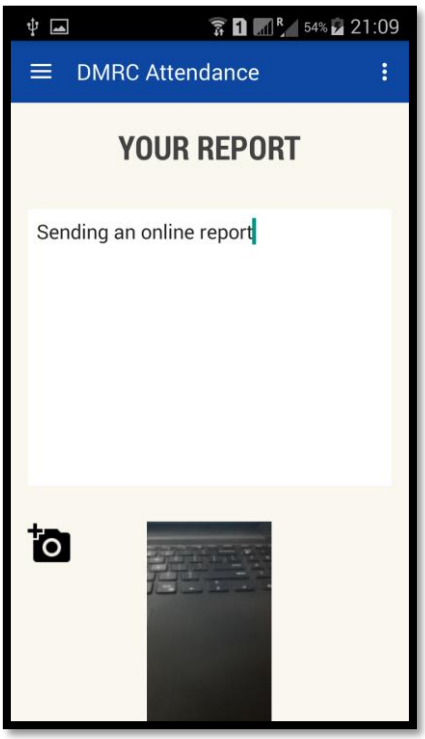
This page is displayed when the user chooses ‘File a Report’ option in the drawer menu. The user can enter a report to be sent to his supervisor on this page. Once the employee clicks the ‘REPORT’ button, his report along with his current location and current time (using Server Time) is sent to the server, which is stored in the database as well as, an

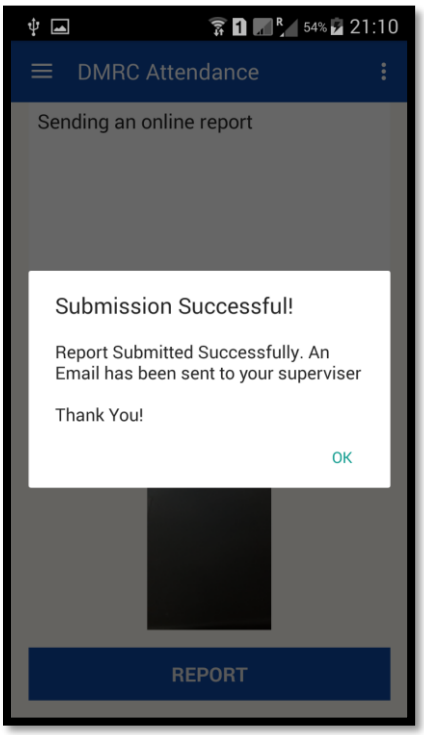
automated e-mail (containing the report) is also sent to his supervisor using the internal mailer. The app also allows for the possibility of an employee having multiple supervisors. In such a case, the automated e-mail containing the report is sent to all of them.

The location is captured in the form of two Coordinates-Latitude and Longitude, which together uniquely determine any location on the Earth. The app first tries to obtain these coordinates using GPS. If it is not able to do so in a given period of time, indicating that either GPS is taking too long to 'lock' location or GPS is not working at all, it uses a built-in Android class called Network Provider to obtain the location. This class uses the location available from the nearest cellular network of the service provider to obtain the approximate coordinates. This can sometimes also give accurate estimates of the current location in closed indoor buildings and project sites, where GPS may not work properly.
At the server side, a class called Geocoder is used to map these two coordinates to a unique address on the Earth.

Reports can be sent to the supervisor on clicking the Report button only if internet connection is available. The reports submitted this way are called online reports. However, in certain cases, access to internet connection may not be immediately available. To tackle this problem, all reports are stored in the local database as well, irrespective of whether it can be sent to the server or not. Reports which could not be immediately sent are called offline reports. They are automatically sent to the server whenever internet connection becomes available and the server is able to respond, using the details of the report (text, image (if necessary), timestamp etc.) stored in the database. Just as in online reports, an automated e-mail to the supervisor(s) is sent in case of offline reports also with a note that this was an online report which is being sent right now. All these report details can be viewed using the 'Previous Reports' option.

An optional feature in this page is the facility to attach an image with the report which may be used to apprise the supervisor of the actual site conditions or depict the situation more accurately. There is a camera icon below the report text area. Clicking this icon opens any camera application available in the device itself. It asks you to select one of the camera present in the phones (for phones with multiple camera apps installed). After clicking the image, there is a choice of saving or discarding it. On saving it, the image is shown in the report page below the text as a thumbnail and it can be sent along with the report if required.





7. Profile page

This page is displayed when 'Profile' option is chosen in the drawer menu. It displays the name, department and list of supervisor(s) of the employee. This data is fetched from the database using the login credentials of the employee (username and imei). The list of supervisors is flexible and the user can scroll if required to view the entire list. Once a person logs in into his/her profile, these details also get stored in the local database on the device itself, so that the application doesn't have to ping the server every time to get these basic details.

8. Auto login, forgotten password and logout features

The login page has two extra features-**auto login and forgotten password**.

The **'Keep me logged in'** checkbox, if checked by the user saves the credentials (username and password) locally in the app using a feature called 'SharedPreferences' which is a predefined JAVA class. It can be used to store data in the form of key-value pairs and retrieve the value using its corresponding key whenever required. Thus when the user, who has checked this option, opens the app again (after logging out of the app), his/her credentials are already retrieved and auto-filled in the fields in the login page.

The **'Forgotten Password'** feature allows the user to reset his/her password through an auto-generated email which is sent to his/her registered email address (through which the user registered in the app).

The logout option in the drawer menu can be used by the employee to logout of the app. The information regarding an employee's status (automatically logged in or not) is stored as a boolean in the app and it is changed appropriately whenever the auto-login or the logout feature is used.

9. Fetching location: GPS vs Network Provider

Obtaining user location from a mobile device can be complicated. There are several reasons why a location reading (regardless of the source) can contain errors and be inaccurate. Some sources of error in the user location include:

- **Multitude of location sources**

  GPS, Cell-ID, and Wi-Fi can each provide a clue to user's location. Determining which to use and trust is a matter of trade-offs in accuracy, speed, and battery-efficiency.

- **User movement**

  Because the user location changes, we must account for movement by re-estimating user location every so often.

- **Varying accuracy**

  Location estimates coming from each location source are not consistent in their accuracy. A location obtained 10 seconds ago from one source might be more accurate than the newest location from another or same source.

To get as accurate an estimate as possible we make use of both GoogleApiClient to get GPS location as well as Network Provider to get nearest cellular tower location. For network provider we request location updates every 2 min and at a minimum distance change of 2 meter from the previous location.

Comparing GPS and Network Provider

GPS: Although GPS is most accurate, it only works outdoors, it quickly consumes battery power, and doesn't return the location as quickly as users want.

Network Provider: Android's Network Location Provider determines user location using cell tower and Wi-Fi signals, providing location information in a way that works indoors and outdoors, responds faster, and uses less battery power. This comes at a cost of being less accurate than GPS.

10. Offline database using SQLite

An important feature of our app is the ability to save reports offline allowing the app to retrieve those later and to send these reports automatically to the server in the presence of internet during the time the app is loaded and the Splash Screen is displayed This storage is done using local storage available on the app in the form of SQLite database system.

SQLite is an **open-source relational database** i.e. used to perform database operations on android devices such as storing, manipulating or retrieving persistent data from the database .It is embedded in android by default. So, there is no need to perform any database setup or administration task.

Why SQLite?

- **It's lightweight.**

Most database systems need a special database server process in order to work. SQLite doesn't, a SQLite database is just a file. When you're not using the database, it doesn't use up any processor time. This is important on a mobile device, because
we don't want to drain the battery.

- **It's optimized for a single user.**

Our app is the only thing that will talk to the database and SQLite ensures that the user does not have to identify himself/herself with a separate username and password.

- **It's stable and fast.**

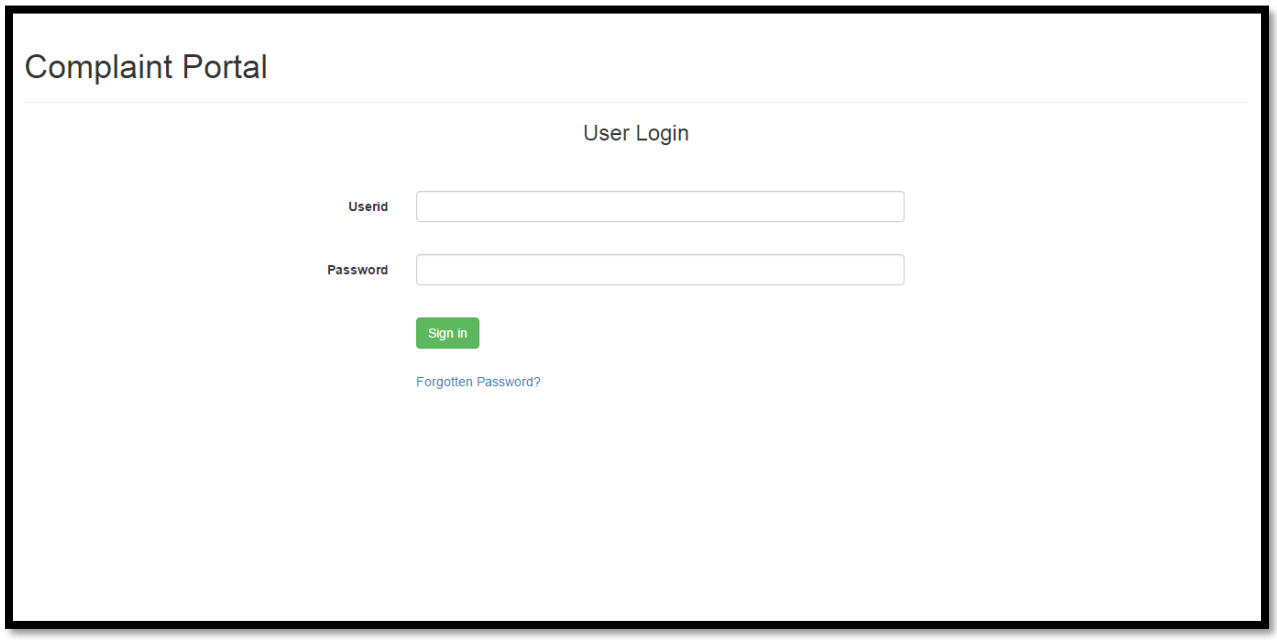
SQLite databases are amazingly stable. They can handle database transactions, which means that we could roll the data back anytime if we wanted to revert after erroneously updating several data items at once. Also, the code that reads and writes the data is written in optimized C code. Not only is it fast, but it also reduces the amount of processor power it needs.

A user defined class called DBHelper is used which implements all the basic CRUD (create, read, update and delete) operations required in the SQLite local database.

Analysis of the CSS Web Application:

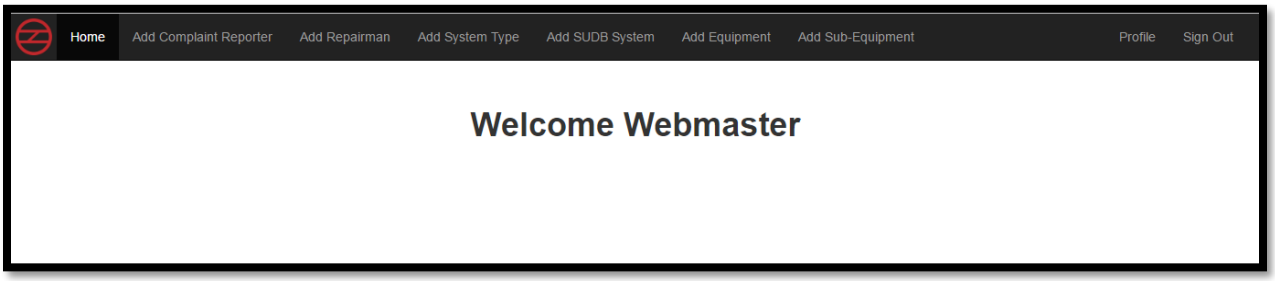
1) **Login Screen:**

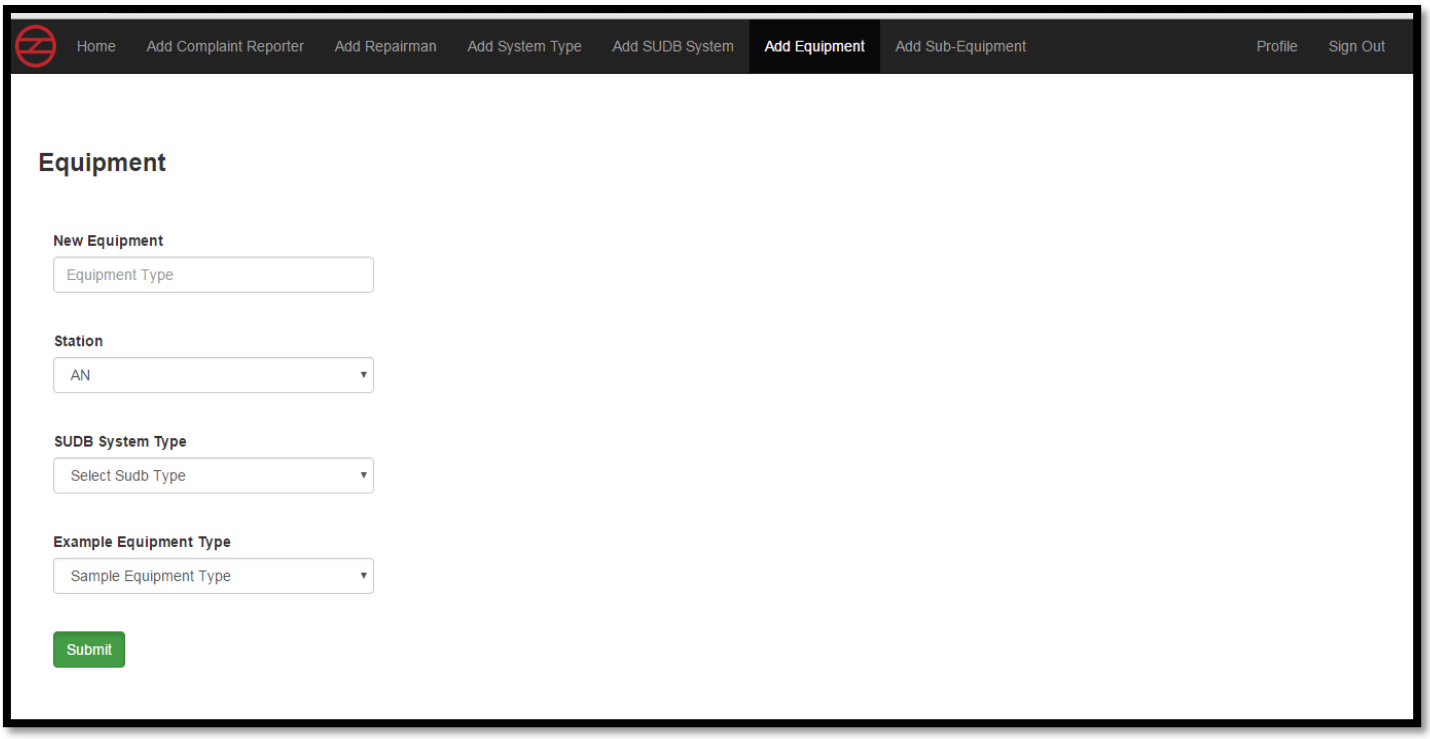
The login screen is the home page of the web portal. A user is required to login every time to access the web portal. This has been kept to make
verify access is given only to the authorized people. This also provides a functionality where the user can reset his password in case he forgets it. On login the type of user is automatically detected through the user id and is then redirected to the required page



2) Webmaster Home Page:

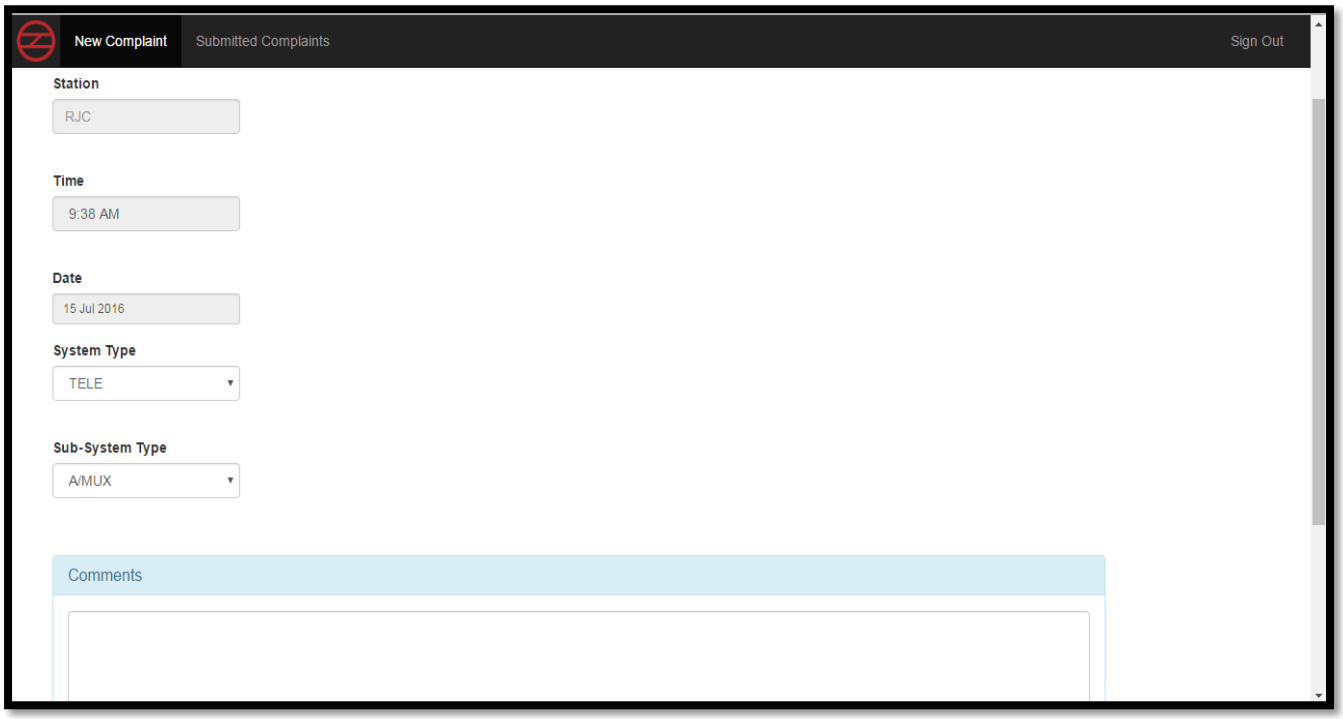
This page is used by the webmaster. It grants the webmaster exclusive rights of adding new equipments, systems and users.





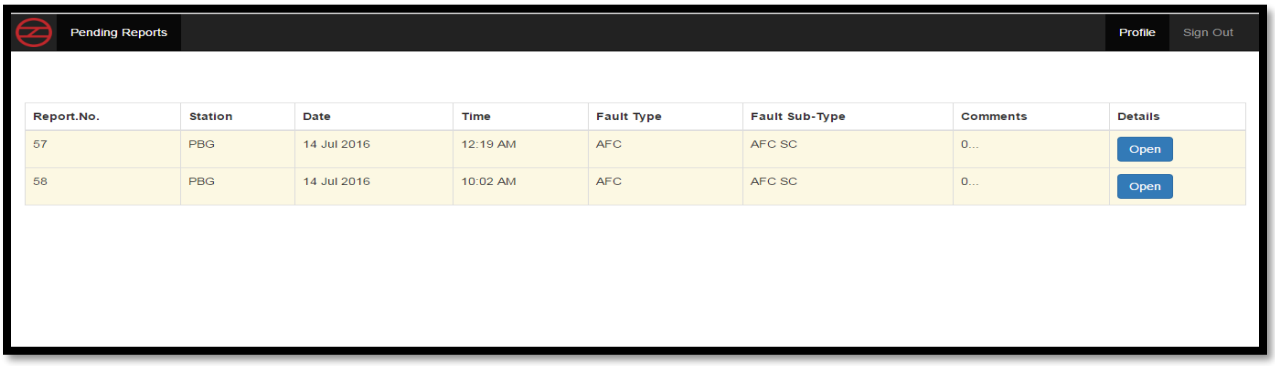
3) **Report Page:**

This is the home page for the Station Controller who can login and lodge complaints. He can also view status of already registered complaints by him. While lodging a complaint. The system automatically fills in his station, date and time. It also provides intelligent suggestions as to what the SUDB system type might be depending on the System type filled in by the Station controller.



4) **Repair Page:**

This is the home page for the repair staff. It shows the pending complaints for the user on login. He can then click on the complaint to open a detailed view and also submit the repair report for the same. Most required fields are automatically filled in by the system.



FINAL OBSERVATIONS AND CONCLUSIONS

The DMRC attendance application has undergone rigorous testing and has passed them with flying colors. We were told that it would be a live project and they will be implement within their system.

The CSS Web Portal is currently undergoing Phase I testing.

During testing of both the applications, it has been observed that the required tasks are performed more smoothly than the manual procedures. Hence it can be concluded that all procedures that require data handling should be shifted to digital means not only to reduce labor and time but also to increase data management efficiency.

Though digitization does make our lives simpler, it also opens a new domain for malpractices and thefts of data. Moreover, with the expanding domain and procedures, which get increasingly complex as time progresses, the developed systems can easily get out of date within a matter of years. Hence decisions regarding implementations should be taken keeping all the factors in mind.

FINAL REMARKS AND RECOMMENDATIONS

Successful demonstrations and testing phases of both the applications have given us an insight to the pros and cons of the applications. Keeping those in mind, the following recommendations are suggested:

1. The systems have to be regularly updated with appropriate security measures so as to curb data thefts and other security breaches. For this, it is necessary that the organization is up to date with the global guidelines on information security.

2. A team of developers should be created so that more functionalities can be added in the system as and when required. This will ensure that the system is up to date with the latest technical advancement.

3. For CSS Web Application, Machine Learning and Artificial Intelligence can be integrated with Internet of things so that a complaint is registered automatically as soon as a device fails and there is no requirement for the station controller to register complaints.

Bibliography

- Google. (2016). *Android Developers*. Retrieved from Android Developers: https://developer.android.com/index.html
- Google. (2016). *Android Developers YouTube Channel*. Retrieved from YouTube: https://www.youtube.com/user/androiddevelopers
- Google. (2016). *Developing Android Apps*. Retrieved from Udacity: https://classroom.udacity.com/courses/ud853/lessons
- Griffiths, D. G. (2015). *Head First Android Development.* O'Reilly Media.
- Roberts, B. (n.d.). *Android Application Development*. Retrieved from The New Boston: https://thenewboston.com/videos.php?cat=6
- *Stack Exchange Q & A*. (n.d.). Retrieved from Stack Exchange: http://stackexchange.com/
- *Stack Overflow Android Tags* . (n.d.). Retrieved from Stack Overflow : http://stackoverflow.com/questions/tagged/android