Project report on

Digital Classroom Management (DCM)

Group number: 17

Group members:

Walid Hasan Tanim (ID: 1702139)

Takia Esha (ID: 1802129)

Md. Mostafijur Rahman (ID:1802161)

Submitted to:

Md. Abu Marjan Sir

Lecturer

Dept. of CSE, Faculty of CSE

Hajee Mohammad Danesh Science and Technology University

Project name: Digital Classroom Management (DCM)

Overview of project:

We live in the period of technology boosting where everybody is running after like a race, No Time to Waste. So considering this, we made this device for all purpose of people from remote to nearer to get their instant information up to regular schedule in a very effective way.

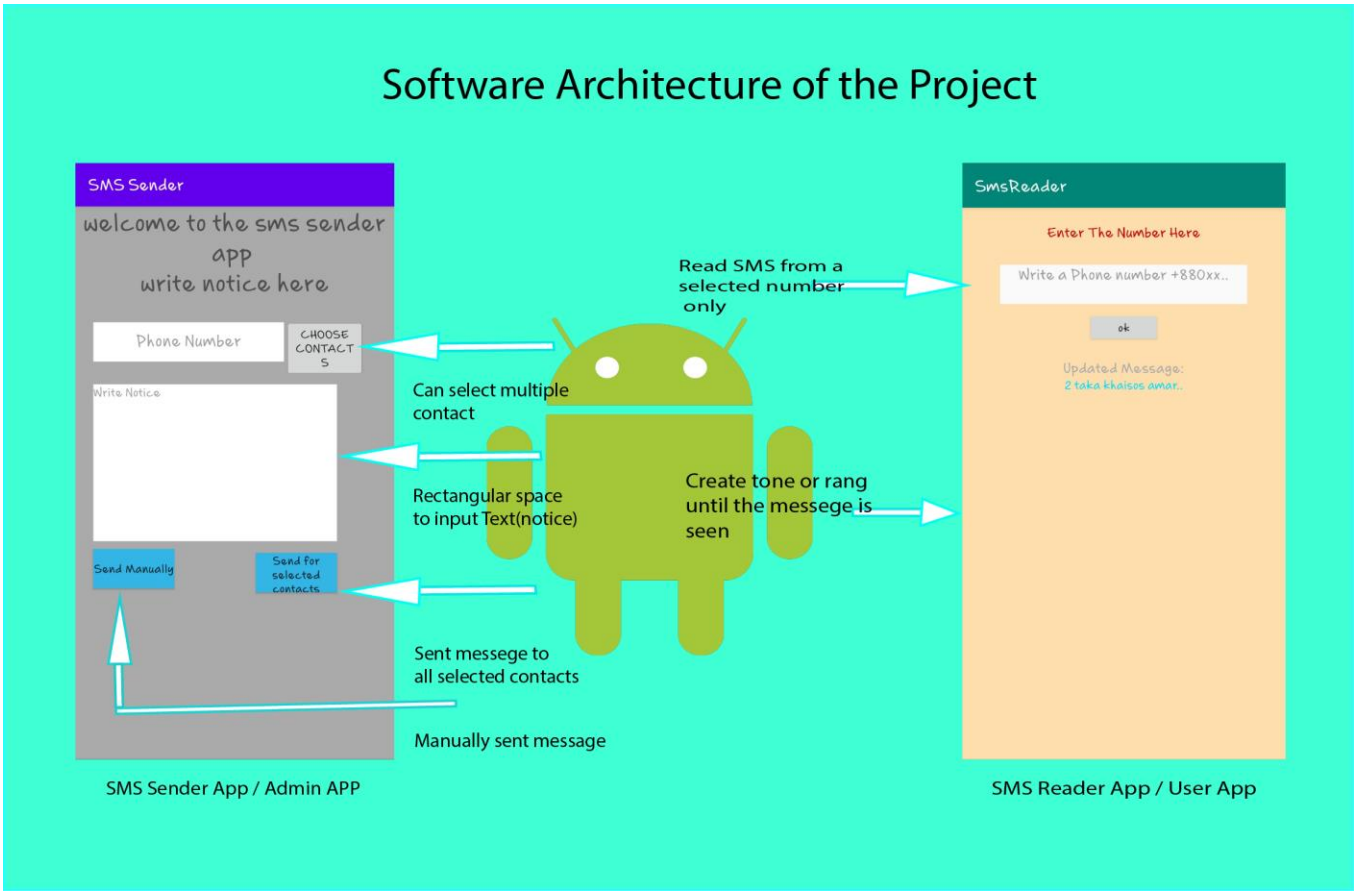
Purpose of the project:

To ensure updated information about any particular sector to their consumer within a short span of time. Our proposed project called “DCM” basically helps teachers and students about their daily schedule of class time, special occurrence, meeting and so on in a short way with less cost.

Required equipment:

1. Android studio.
2. Java NetBeans IDK 8.2
3. Intellij IDEA 20.0

Software architecture of the project:



Analyzing of the project:

Basically our project is assembled with two major parts. They are -

1. Admin app/SMS sender app.
2. User app/SMS reader app.

Functional part of admin app:

- ❖ It uses Dexter which is an Android library that simplifies the process of requesting permissions at runtime.
- ❖ Dexter frees permission code from activities and lets user write that logic anywhere they want.
- ❖ If we want to request multiple permissions we just need to call withPermissions() and register an implementation of MultiplePermissionsListener()
- ❖ Uses setOnClickListener( ) to listen button action
- ❖ Uses sendTextMessage() method of SmsManager class to send sms to specified number
- ❖ Uses MultiContactPicker class, A simple, material design multi-contact picker using RecyclerView and Alphabetical Fast Scrolling! The picker will read the device contacts
- ❖ (**NOTE**: Runtime permissions for retrieving contacts) and allow you to select 1-many and return them in a convenient list

Functional part of User app:

- ➢ Uses intended ReceiveSms class to receive sms.

- Uses separate method writeFile() method to store sms and readFile() to read stored sms.
- Uses storePhoneNum() to store and readFile2() to read phone number.
- Uses foreground service to inform use about the updated sms.
- Repeatedly uses a counter/timer to always run the app in the background services so that it can always check the incoming sms.

Non-functional part:

- Easy and well-designed user interface that everybody (minimum idea to operate smart phone) can use it.
- Quality issues are good enough to launch it widely.
- Project performance is very fast rather than before. It takes around 1 sec to send any particular notice to user through a single call or SMS. More particularly, taken example from our class, ECE-18 with 70 students notifies everyday about their class schedule. Unexpectedly if some changes happen then class captain will reschedule that notice through this device required only 70 sec for 70 students.
- In security perspective, only user can access it by putting a pin number or effective password. It is also used in a certain area who belongs this device and people who are listed in database.

Features of OOP used in our project:

Inheritance: Inheritance, in Java, is a mechanism in which one object acquires all the properties and behaviors of a parent object. Inheritance is used for two purposes, i. method overriding so that run time polymorphism can be achieved and ii. For code reusability. This code reusability features use most of the cases like main activity, SMS receiver in SMS Reader app.

Polymorphism: Polymorphism is a feature of the object-oriented programming language, Java, which allows a single task to be performed in different ways. In the technical world, polymorphism in java allows one to do multiple implementations by defining one interface. Method overriding directly operates in lots of features like button and receiver etc.

File input stream: Java FileInputStream class obtains input bytes from a file. It is used for reading byte-oriented data (streams of raw bytes) such as image data, audio, video etc. You can also read character-stream data. But, for reading streams of characters, it is recommended to use FileReader class. It has wide range of methods like int available(), void close(), int read() etc. are used.

Final discussion of the project:

Our project has various advantages in accordance with the modern technological advancement. This arrangement belongs to the admin where he or she types the information and send it to its user who listed in its dynamic ArrayList. This text must be converted into string and through SMS manager class's method which is SendTextMessage() that processed the particular text. It contains four parameters. However, the admin selects the button send for selected contacts and sends them. Then the SMS reader app will read that SMS from a selected number and most importantly, this device ensures that users have seen it with a tone or vibration. This is as simple as to operate a smart phone. Basically our aim is to be added a new dimension to our daily communication through this project.

Advantages of the project:

Benefits are that suppose you have a meeting tomorrow sharp at 10 am. You have to present there before 9 am. But anyhow your meeting is being cancelled just before at 9 am. But somehow you don't know this or you are on your way. So if your Boss or Host have this device, then just type "Meeting Cancelled" text to its all listed employees. This message will be delivered with a tone to all the employees through User App required around 1sec.

So this will save both of your time and cost. This will largely help students, teachers, corporate workers etc.

Limitations of the project:

1. Need to turn off battery optimization.
2. Always run the app at backend.
3. Security issues are not fully confirmed.
4. Auto-start apps management > SMS Reader > Allowed must be done in SMS Reader App.