Lab report on :

Mobile application and game development project.

Submitted to:

Md.Rakib Hasan

Lecturer

Department of ICT

Comilla University

Submitted By:

Sinthia Akter

ID:11609046

Session:2016-17

Dept. of ICT

Comilla University

Project : 01

Name of the project : Switching Lights.

Introduction :

**Android Switch** or SwitchCompact widget is a custom On-Off slider which is commonly seen in phone settings. . LightSwitch is a Nintendo **Switch** emulator with which we can run and play the games of this video console in ROM format on our smartphone or tablet .The best smart **light switch** is arguably the Wemo **Light Switch.**

**Motivation:**

This project presents the design of Home Automation System (HAS) with low cost and wireless system. This is a simple home automation tool that relies on HTTP requests being sent back and forth from an Android app to a Python script running on a Raspberry Pi to remotely turn lights or other devices on and off.

Description:

The best way to control LED Bulbs, Including Home and Lock screen widgets .To get LED lights, it is easy to find using the Light switch more often than not because of the need to open the app in order to simply turn on the lights. this was developed specifically in order to give the light switches where they are most accessible .We can choose where we want the switches, be it on one of home screens, on the lock screen or both.

The widget gives us easy access to switch each zone on and off as well as global control. also included is easy access to the main application for more advanced control.

The application enables control of both RGBW bulbs and Dual White Bulbs, And we can now choose to turn on one or both sets of lighting zones.

We have started working on some extra lighting modes also light listening mode, where it will attempt to detect the beat of the music playing and pulse different colours to the music. A candle mode to emulate the flickering of a candle, along with the built in disco modes.

Control LED Bulbs from:

* AppLamp
* EasyBulb
* Limitless LED
* MiLight
* ilux

and some others also.

Some snaps:



**Project : 02**

Name of the project : Sending Email App.

Introduction:

Email sender app is an **Android app** that allows you to set up your **email** account once and then use it to **send emails** or quick notes to an address of your choice and receive email from the sender.

In **android**, we can easily **send** an **email** from our **android application** using existing **email** clients such as GMAIL, Outlook, etc. ... Generally, the Intent object in **android** with proper action (ACTION\_SEND) and data will help us to launch the available **email** clients to **send** an **email** in our **application**.

**Motivation:**

**This project presents a email sending app which can be used** to send an email from our android device to another android device we don’t have to implement an email client from the beginning, but we can use an existing one like the default Email app provided from Android. We will use **ACTION\_SEND** action to launch an email client installed on your Android device.

Description:

For making a basic [android application](https://www.geeksforgeeks.org/android-app-development-fundamentals-for-beginners/) which can be used to send email through your android application.

It can be done so with the help of [**Intent**](https://www.geeksforgeeks.org/android-implicit-and-explicit-intents-with-examples/) with action as **ACTION\_SEND** with extra fields:

* **email id** to which you want to send mail,
* **subject of email** and
* **body of the email**.

Basically **Intent** is a simple message object that is used to communicate between [android components such as activities, content providers, broadcast receivers and services](https://www.geeksforgeeks.org/components-android-application/), here use to send the email.

This application basically contains one activity with EditText to take input of email address, subject and body of the email from user and button to send that email:

**Step 1. activity\_main.xml**  
activity\_main.xml contains a Relative Layout which contains three Edit texts for receiver mail id, other for the subject of the mail and last one for the body of the email and three TextViews for the label and a button for starting intent or sending mail.

**Step 2. MainActivity.java**  
In Main activity Intent object is created and its action is defined to ACTION\_SEND to send email, with Intent three extra fields are also added using putExtra function.These fields are:

* Email of receiver
* Subject of email
* Body of email

**setOnClickListener** is attached to button with intent object in it to make intent with action as ACTION\_SEND to send email and intent type as shown in code.

Some snaps:



