Benha University

Shoubra Faculty of Engineering

"Benha University Mobile Application"

2016-2017

Submitted in Partial fulfillment of the requirement for the award of

Bachelor of Engineering

In

Computer Engineering

Submitted to

Electrical Engineering Department

Submitted by

Ahmed Nabil Hosny Abd EL-Motaleb
Bassma Ali El-Den Yahia Ali
Kareem Mohammed Tawfeek Mohammed
Mahmoud Mohammed Taha
Manal Hesham Hassan Korany

Under the Guidance Of

Dr. Ahmed Bayoumi

Assistant Professor at Computer Systems Engineering Department, Faculty of Engineering at Shoubra, Benha University.

Mr. Shady Yehia EL-Mashad

Assistant Professor at Computer Systems Engineering Department, Faculty of Engineering at Shoubra, Benha University.

Abstract

Modern hand held devices such as smart phones and PDAs have become increasingly powerful in recent years. Dramatic breakthroughs in processing power along with the number of extra features included in these devices have opened the doors to a wide range of commercial possibilities. In particular, most cell phones regularly include cameras, processors comparable to PCs from only a few years ago, and internet access. However, even with all these added abilities, there are few applications that allow much passing of the environmental information and location based services.

As mobile devices become more like PCs they will come to replace objects we tend to carry around such as checkbooks, credit cards, cameras, planners, mp3 players, etc. In short, we will be using them to accomplish our daily tasks. One application that falls into this category is the Benha University APP developed for the Google Android Phones.

The Project is developed in Java Programming Language by using the Android Studio Development tool. We use the Android Software Development Kit (SDK) which includes a variety of custom tools that help us develop mobile applications on the Android platform. The most important of these are the Android Emulator and the Android Development Tools (ADT) plug-in for Eclipse.

- First we have The Main menu activity should help us navigate the whole application with more than one way like icons, drop menus and quick search methods. We Added Attendance Recorder as recording is a mass need for any educational system and developing this process to be easier and faster.
- As each college has a lot of classes, rooms and floors this can cause long time to get to your destination room, using our App will help you providing the best route.

- We have used the synchronized calendar to reminder the student all his quizzes, lectures and important events in his college.
- Helping students keep in touch with all information about university, student can surf the University Portal inside the app.

- Benha App has a very smart target that it wants to propagate mission and vision of the university and for all its colleges through our app.
- We have created setting list to help user of App with additional tasks like sign in with his academic mail and password, change language and so on and News timeline helps students keeping in touch with the latest university news.

- Our app provides the student with directions to all university buildings with outdoor navigation with GPS and indoor navigation using graphs and A* search technique. also we have created Emergency numbers are provided in the application to make for emergency calls faster.

- Benha University has a news and photos gallery but not many students are aware of this fact, so we added gallery module to bring the gallery closer to the student.
- Benha App is smart and interactive as it's provided with dimensions for multiple devices and supports many languages

- We are glad that our App is the first App to serve an Egyptian University and we know that database is an important part from any system so we needed to keep it safe, fast and easy to use

Table of Contents

Problem Statement

The Real need of "Benha University APP" is to create a full-fledged Android application which could help students, university staff and normal users to get and use all data they need from university. The user not only finds all the colleges of the university and any information he needs, but also he can make a lot of actions like indoor navigation where user can navigate inside university buildings easily, Attendance recording which makes recording the attendance for every file easier and can also get notification about his staff. The user can also map the location of the colleges on Google Maps rendered to the user on the phone & find the path from his current location or from any other location to the college; the user has the facility to make a call directly to the college and can also obtain the detailed data about this college. Also Staff members may take some photos that they want to share with students but unfortunately the gallery on the websites is relatively out of students' reach.so our App gallery provides fast and easy way to share all university news and innovations

- We asked for a web service to access data from Benha university database but for our demo we have built a server on tomcat using eclipse and SQLite database and have built database scheme for some modules like sing in , sing out , upload data and notifications and so on

Why Android…?

Devices… devices everywhere!

There is something for everyone on the Android platform. The sheer variety of Android smartphones from manufacturers like Samsung, HTC, Sony, Motorola, LG, Huawei, ZTE, and others is staggering. You can get a compact phone, something with a huge touchscreen, a stylus, a rotating camera, an edge screen, or even a physical keyboard such as found on the Blackberry Priv. Niche demands like dual SIM are catered for, and the flagship devices are on the cutting edge when it comes to specs. There's also features you just can't get with Apple's devices, like microSD and removable battery. Yes, some Android manufacturers have moved away from offering these extras, but the nice thing about options is there are manufacturers that still do.

The same story exists largely for the tablet market too, with all sorts of different devices from the Pixel C to the Nexus 9, Xiaomi's tablets, Honor's tablets, Samsung Note tablets, and the list goes on.

Attracting a huge range of manufacturers and giving them license to run amok in terms of imagination has resulted in the widest variety of devices on any platform, even blurring the line between smartphone and tablet. Compared to the limited iPhone and iPad lineup, Android represents choice on a grand scale.

Sure, Apple has a few sizes to offer, but size is really the only differentiator there. The iPhone 6S and iPhone 6S Plus, for example, have pretty much most of the same specs with the main difference being screen size and resolution. The same general argument applies to the iPads, even though there are more choices than with the smartphone counterparts.

Prices to fit your needs

This naturally follows on from the first point on our list. A wide variety of devices with different designs and specs means that Android has something for you at just about any budget. The exclusive nature of Apple's products is in stark contrast to the inclusive nature of Android. Almost anyone can afford an Android phone. It might not be all-singing and dancing, but there are solid budget options that give people a true smartphone experience.

Customization

One of the strong points of Android has always been the level of customization it allows. While Apple wants to keep control of default apps in order to maintain a homogenous software and hardware experience, Android lets you pick your own level of customization. This extends all the way from simple things like live wallpapers, to alternative keyboards, to custom ROM installs.

Launchers

Grab an iPhone, sit next to another iOS user and compare your home screens. Oh wait, they look exactly the same! That's not the story with Android as in figure 1.

If you want control over how your Android smartphone or tablet looks then you've probably tried out a custom launcher. You can choose from a wide variety of custom launcher apps in Google Play and tweak everything from your home screen layout, to your page transitions, to effects and even gestures. There's no risk involved with launcher apps and you can really open up a world of possibilities.

Figure 1 - different launchers for different Android devices

Results and discussions

Introduction

This chapter highlights the results of the project & the snapshots for each of the activities are shown along with the discussion of each activity describing its working. Each snapshot describes every single step of the Benha University APP. Many main activities as well as the options provided to the users in each activity such as menu options and the activities which are created on the click of these options are also shown and described in brief.

Snapshot of outputs

The highlighted icon is the image icon representation of the Benha University App, on clicking it would open the application which results in the creation of the first activity.

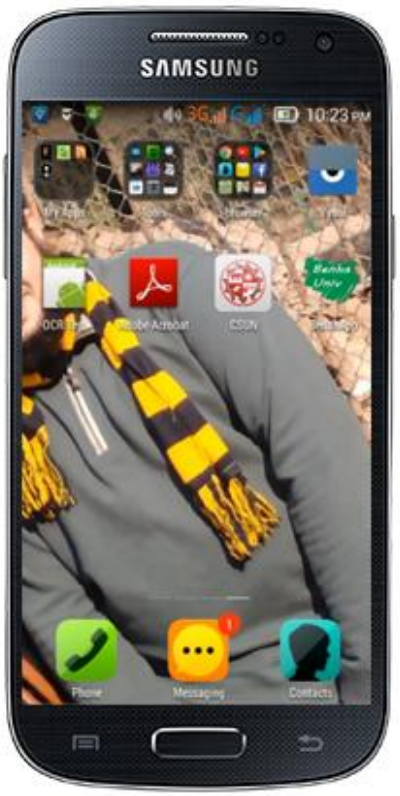


Figure 3- Benha App on the launcher

Figure 2 - logo of Benha App

Splash screen

Figure 4 - splash screen of Benha App

This is the first screen of the project output, which has user interface for the splash screen of the application

Main Menu

Figure 5 - start menu of application

This is the first Activity of the project output, which has user interface for main menu of the application it has many good technical and design features see it in figure 5

Technical details

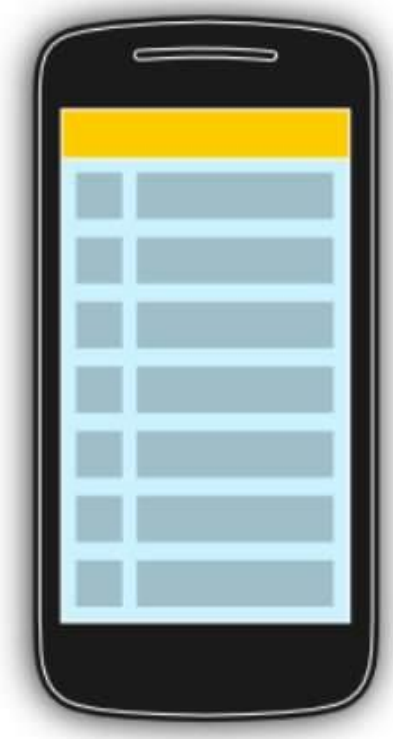
1- This activity has many features to help user
   - Several tabs to navigate between the menu and the other features like mission and vision

Figure 6 - many fragments in the Start Activity of the application

- Menu of all app parts and each of them is a separate module that will be discussed later
  - The smart part about this menu that we used the custom list and recycled view of Android which lead to high performance and maximum benefit

Using lists in Android

The display of elements in a list is a very common pattern in mobile applications. The user sees a list of items and can scroll through them. Such an activity is depicted in the following picture like in figure 7.



Typically the user interacts with the list via the toolbar, for example, via a button which refreshes the list. Individual list items can be selected. This selection can update the toolbar or can trigger a detailed screen for the selection. The following graphic sketches that. On the selection of a list item another activity is started.

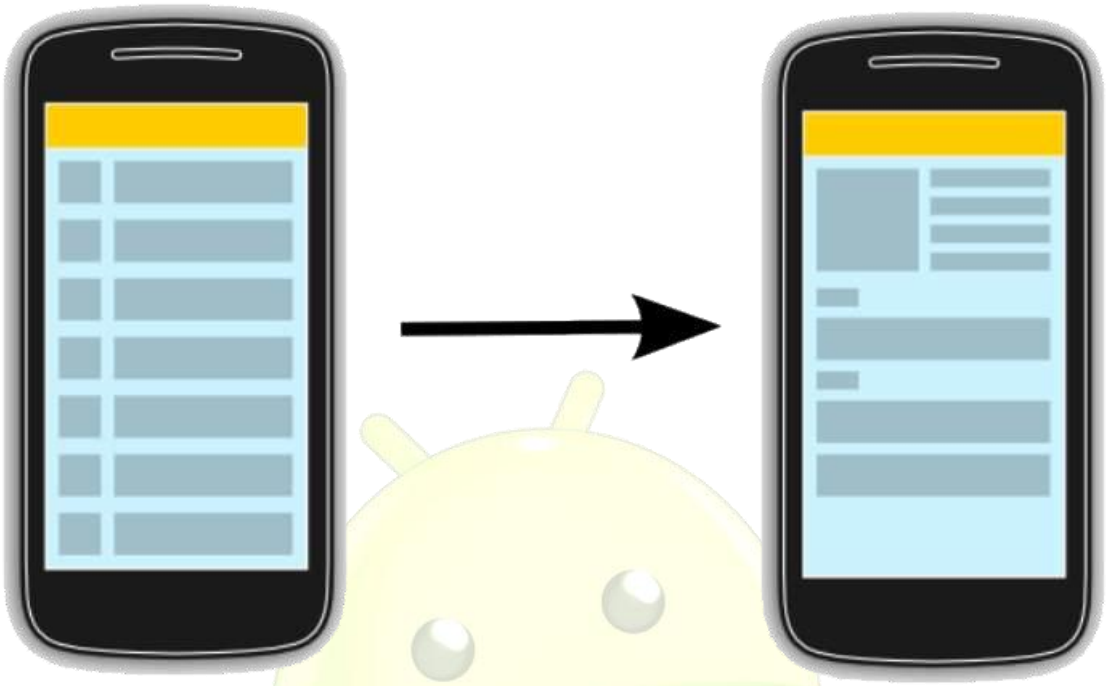


Figure 7 - moving from list data to detailed data of a list item

Motivation for Custom lists

Performance is especially important on Android as users expect fast reaction times. Compared to desktop computers an Android device is relatively slow from the hardware perspective.

This part describes how to reduce these operations to implement your custom list adapter efficiently. The default Android adapters like `ArrayAdapter` are already performance optimized.

Time consuming operations

-Every view which gets inflated from an XML layout file will result in a Java object. Inflating layouts and creating Java objects are expensive with regards to time and memory consumption.

In addition using the `findViewById()` method is relatively time consuming, even though it is not as bad as XML inflating.

Avoiding layout inflation and object creation

A `ListView` typically contains more data than the number of displayed rows. If the user scrolls the list, then the rows and their associated views will be scrolled out of the visible area. The Java objects which represent the rows can be reused for newly visible rows.

If Android determines that a row is not visible anymore, it allows the `getView()` of the adapter method to reuse the associated view via the `convertView` parameter.

The adapter can assign new data to the views contained in the view hierarchy of the `convertView`. This avoids inflating an XML file and creating new Java objects.

In case Android cannot reuse a row, the Android system passes `null` to the `convertView` parameter. Therefore the adapter implementation needs to check for this.

View holder pattern

A *ViewHolder* implementation allows to avoid the `findViewById()` method in an adapter.

A `ViewHolder` class is typically a static inner class in your adapter which holds references to the relevant views in your layout. This reference is assigned to the row view as a tag via the `setTag()` method.

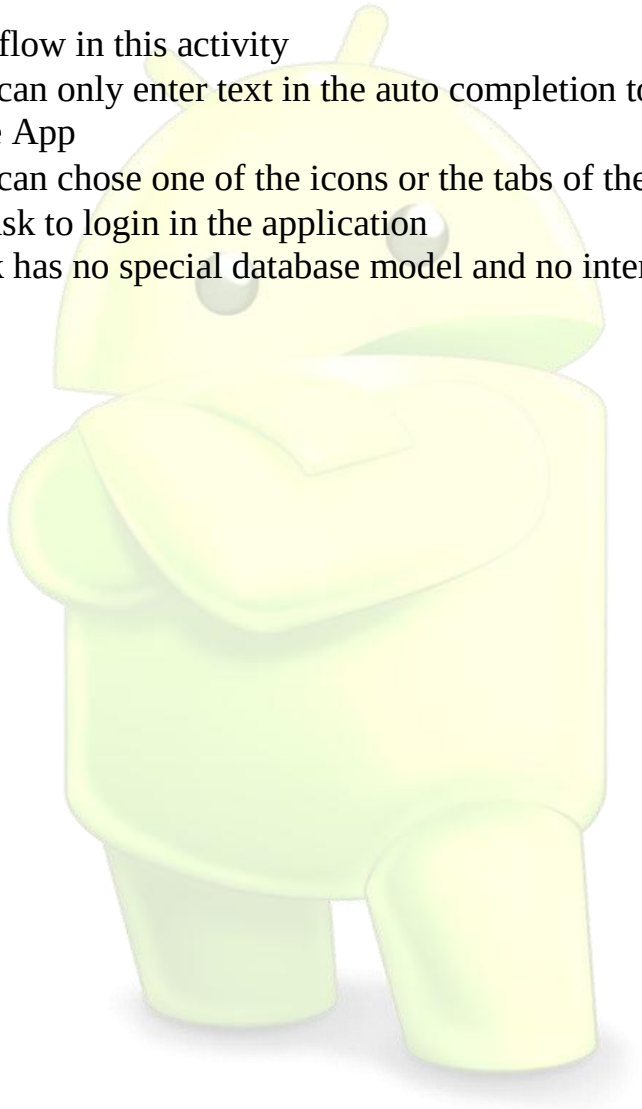
If we receive a `convertView` object, we can get the instance of the `ViewHolder` via the `getTag()` method and assign the new attributes to the views via the `ViewHolder` reference.

While this sounds complex this is approximately 15 % faster than using the `findViewById()` method.

Example from the Application

- The menu fragment code see example Appendix A.1.1
- The Adaptor class see example Appendix A.1.2

2- The dataflow in this activity
   - User can only enter text in the auto completion to access some part of the App
   - User can chose one of the icons or the tabs of the Activity
   - Can ask to login in the application
3- This task has no special database model and no interfaces with other systems



Auto Complete Text View

- Auto completion search window to help quick access to any part of the application
- This is a list of all auto completion keywords used in this system

```
public static final String ALERT = "Notification";
    public static final String LIBRARY = "Library";
    public static final String NAVI = "Navigate";
    public static final String MAP = "Maps";
    public static final String ATTENDANCE = "Attendance";
    public static final String STAFF = "Staff";
    public static final String OFFICE = "Office 365";
    public static final String EKB = "EKB";
    public static final String PORTAL = "Portal";
    public static final String GALLERY = "News";
    public static final String COURSE = "Courses";
    public static final String CALENDER = "Time Table";
    public static final String RESULT = "Exam Results";
    public static final String EMERG = "Emergency";
    public static final String SAFE = "safety";
```

- See figure 8 for task design

Figure 8 - auto complete TextView for quick search

Attendance Recorder

Problem Statement

- Attendance recording needs easy and secure way to record attendance and check no cheating in it
- Routine in attendance recording takes a lot of efforts papers and has a great predictions for cheating

Technical Details

- The idea for solving this problem is to use new technologies, programming and mobile sensors to get the task as required

Identifications

- There is many ways to uniquely identify each person like: RDFI , barcodes , finger print and QR codes and we have chosen the last one like figure 9

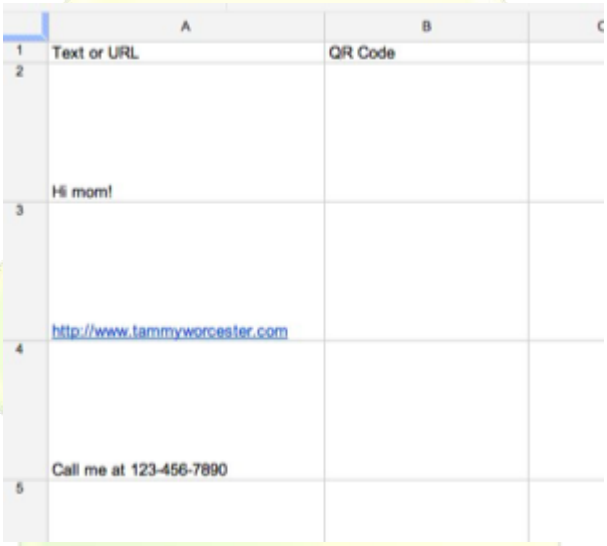
Figure 9 - QR code with coded name in it

- They come to us from Japan where they are very common. QR is short for Quick Response (they can be read quickly by a cell phone). They are used to take a piece of information from a transitory media and put it in to your cell phone. You may soon see QR Codes in a magazine advert, on a billboard, a web page or even on someone's t-shirt. Once it is in your cell phone, it may give you details about that business (allowing users to search for nearby locations), or details about the person wearing the t-shirt, show you a URL which you can click to see a trailer for a movie, or it may give you a coupon which you can use in a local outlet.

- The reason why they are more useful than a standard barcode is that they can store (and digitally present) much more data, including URL links, geo coordinates, and text. The other key feature of QR Codes is that instead of requiring a chunky hand-held scanner to scan them, many modern cell phones can scan them

Google spread sheets batch QR code generator

- We enter all student names in one column



	A	B
1	Text or URL	QR Code
2	Hi mom!	
3	http://www.tammyworcester.com	
4	Call me at 123-456-7890	
5		

Figure 10 - Google spread sheets with all students' names to get bulk QR codes for them

- Then, In Cell B2, enter this formula

  =image("https://chart.googleapis.com/chart?chs=150×150&cht=qr&chl="&A2)
- Then we can download all QR codes at once
- Result of this code in figure 11

Figure 11 - Google spread sheets after bulk QR generation

- This is the interface of the QR code Activity in figure 12

Figure 12 - Attendance recorder main activity

- We have also used external QR code reader activity (QR scanner) and we had interface with it to get read data from it – see figure 13

Figure 13 - ZXING QR code reader

- Code to interface with ZXING QR code scanner see example Appendix A.2.1

Database

- Our attendance recorder works on any academic year or staff
- You need to provide *.csv file with all students names that will record attendance in the current session
- On click on new session button it will open file chooser to choose the *.csv file then it will be loaded in the system and now you can start attendance recording

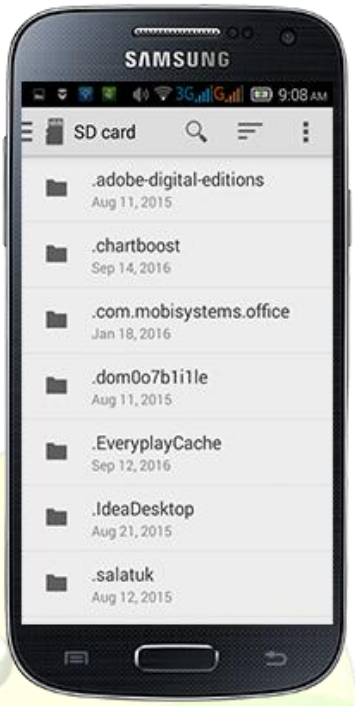


Figure 14 - file chooser for attendance recorder absence file

- On click on present button it will open the QR code reader and read the QR of the student

- Each QR contains data about the student and when read it will save and update file in the /documens/BenhaApp with the current session date and time and update this student attendance

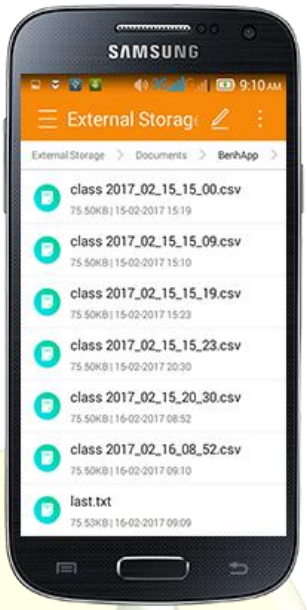


Figure 15 - all saved files by the attendance recorder

Figure 17 - last saved data from App

Figure 16 - session attendance save on SD card in .csv file

- Code saving and loading data from files see example Appendix A.2.2

Cheating detection

- There are many models for checking security on the system and there is no cheating in attendance recording
- We have used smart idea as a way to reduce cheating and not to perfectly stop it with the least need of cost and time
- We have used the mobile sensors mainly accelerometer

Accelerometer sensor in Android

- Acceleration is the measurement of the change in velocity, or speed divided by time. For example, a car accelerating from a standstill to 60 mph in six seconds is determined to have an acceleration of 10 mph per second (60 divided by 6).

The purpose of the accelerometer

- The application of accelerometers extends to multiple disciplines, both academic and consumer-driven. For example, accelerometers in laptops protect hard drives from damage. If the laptop were to suddenly drop while in use, the accelerometer would detect the sudden free fall and immediately turn off the hard drive to avoid hitting the reading heads into the hard drive platter. Without this, the two would strike and cause scratches to the platter for extensive file and reading damage. Accelerometers are likewise used in cars as the industry method way of detecting car crashes and deploying airbags almost instantaneously.

How they work

- An accelerator looks like a simple circuit for some larger electronic device. Despite its humble appearance, the accelerometer consists of many different parts and works in many ways, two of which are the piezoelectric effect and the capacitance sensor. The piezoelectric effect is the most common form of accelerometer and uses microscopic crystal structures that become stressed due to accelerative forces. These crystals create a voltage from the stress, and the accelerometer interprets the voltage to determine velocity and orientation.
- The capacitance accelerometer senses changes in capacitance between

- Microstructures located next to the device. If an accelerative force moves one of these structures, the capacitance will change and the accelerometer will translate that capacitance to voltage for interpretation.
- Accelerometers are made up of many different components, and can be purchased as a separate device. Analog and digital displays are available, though for most technology devices, these components are integrated into the main technology and accessed using the governing software or operating system.

- Typical accelerometers are made up of multiple axes, two to determine most two-dimensional movement with the option of a third for 3D positioning. Most smartphones typically make use of three-axis models, whereas cars simply use only a two-axis to determine the moment of impact. The sensitivity of these devices is quite high as they're intended to measure even very minute shifts in acceleration. The more sensitive the accelerometer, the more easily it can measure acceleration.
- Accelerometers, while actively used in many electronics in today's world, are also available for use in custom projects. Whether you're an engineer or tech geek, the accelerometer plays a very active role in a wide range of functionalities. In many cases you may not notice the presence of this simple sensor, but odds are you may already be using a device with it.

The way we will use the accelerometer

- It normal that mobile will move between students to record the QR codes of each student
- If one student has taken more than QR code and used it to submit attendance for absent student, so mobile will still be in the same place without great movements
- If accelerometer find this that there is more than attendance recording without great movement it will put this students on cheating list

- After instructor open the cheating list he can check himself that suspected people are in class or not

- On click on cheating list button it will open the list of suspected students like in figure 18

Figure 18 - cheating list of attendance recorder

- If instructor find that one of the suspected students in the cheating list is really not in class, he can click on his name to remove his attendance recording

- Code for accelerometer see example Appendix A.2.3

- Finally, we have tried to use the front camera to prevent cheating by taking image at each attendance recording, but it gave us low performance , low quality and slow reaction so we removed it

Indoor Navigation

Problem Statement

- We need to define a way to know the users current and destination locations
- We need a way to define each collage to our App and total data about it
- We need to define search technique to reach the best route between the source and the destination

Technical Details

Indoor navigation

- An **indoor positioning system** (**IPS**) is a system to locate objects or people inside a building using radio waves, magnetic fields, acoustic signals, or other sensory information collected by mobile devices. There are several commercial systems on the market, but there is no standard for an IPS system.

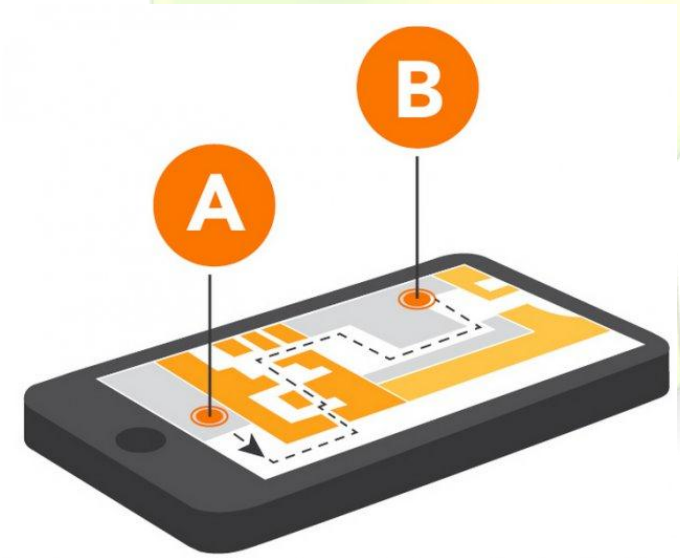


Figure 19- some example on indoor navigation tasks

- IPSes use different technologies, including distance measurement to nearby anchor nodes (nodes with known positions, e.g., WiFi access points), magnetic positioning, dead reckoning. They either actively locate mobile devices and tags or provide ambient location or environmental context for devices to get sensed.

- The localized nature of an IPS has resulted in design fragmentation, with systems making use of various optical, radio, or even acoustic technologies.

➢ The start Activity of Indoor Navigation in figure 20

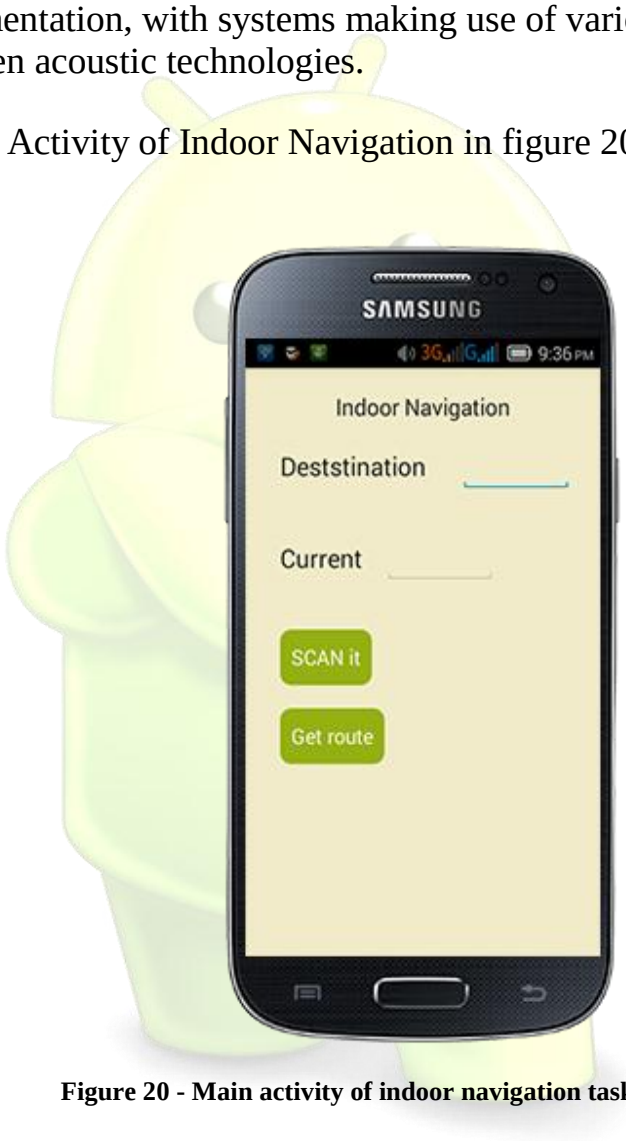


Figure 20 - Main activity of indoor navigation task

Figure 21 - logo of indoor navigation task in the application

Why not to use GPS …?

- Global navigation satellite systems (GPS or GNSS) are generally not suitable to establish indoor locations, since microwaves will be attenuated and scattered by roofs, walls and other objects. However, in order to make positioning signals ubiquitous, integration between GPS and indoor positioning can be made.

- Currently, GNSS receivers are becoming more and more sensitive due to increasing microchip processing power. High Sensitivity GNSS receivers are able to receive satellite signals in most indoor environments and attempts to determine the 3D position indoors have been successful. Besides increasing the sensitivity of the receivers, the technique of A-GPS is used, where the almanac and other information are transferred through a mobile phone.

- However, proper coverage for the required four satellites to locate a receiver is not achieved with all current designs (2008–11) for indoor operations. Beyond, the average error budget for GNSS systems normally is much larger than the confinements, in which the locating shall be performed.

Smart building model

- This idea that we can represent any collage programmatically as a graph of nodes and each node is one of the classes and the cost between nodes indicates the distance , upstairs and downstairs between the classes

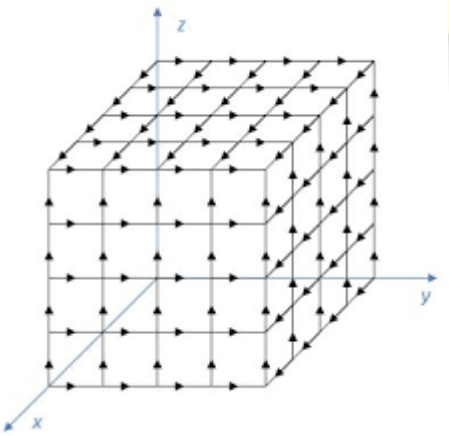


Figure 22 - some example of building a graph model for a building each node here is a class

A-star search technique

- As a good way to fine the best route between the source and the destination is to use the A-star search technique

- A* is an informed search algorithm, or a best-first search, meaning that it solves problems by searching among all possible paths to the solution (goal) for the one that incurs the smallest cost (least distance travelled, shortest time, etc.), and among these paths it first considers the ones that appear to lead most quickly to the solution. It is formulated in terms of weighted graphs: starting from a specific node of a graph, it constructs a tree of paths starting from that node, expanding paths one step at a time, until one of its paths ends at the predetermined goal node.

- At each iteration of its main loop, A* needs to determine which of its partial paths to expand into one or more longer paths. It does so based on an estimate of the cost (total weight) still to go to the goal node. Specifically, A* selects the path that minimizes

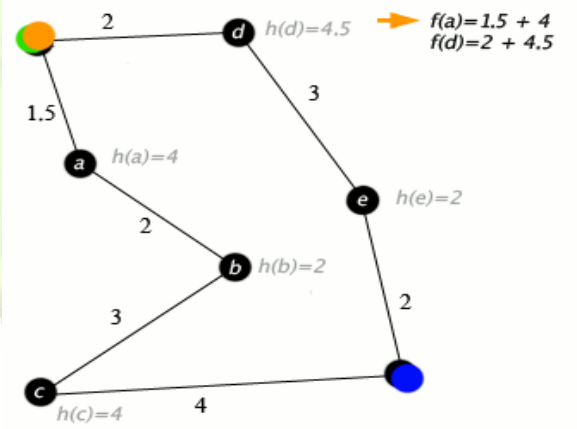


Figure 23 - examples on graph in A-star search

- where n is the last node on the path, g(n) is the cost of the path from the start node to n, and h(n) is a heuristic that estimates the cost of the cheapest path from n to the goal. The heuristic is problem-specific. For the algorithm to find the actual shortest path, the heuristic function must be admissible, meaning that it never overestimates the actual cost to get to the nearest goal node.

- Typical implementations of A* use a priority queue to perform the repeated selection of minimum (estimated) cost nodes to expand. This priority queue is known as the open set or fringe. At each step of the algorithm, the node with the lowest f(x) value is removed from the queue, the f and g values of its neighbors are updated accordingly, and these neighbors are added to the queue. The algorithm continues until a goal node has a lower f value than any node in the queue (or until the queue is empty).[a] The f value of the goal is then the length of the shortest path, since h at the goal is zero in an admissible heuristic.

- Code of our A-star see example Appendix A.3.1
- Class to build graph with the collage model and cost of travels see example Appendix A.3.2

- Easy to use
  - Some new student are not aware with the addressing system of the classes so we help them by OCR system
  - So they need only to use this system to take a photo to name plate of their current location and our system will lead them

Optical Character Recognition

- OCR is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example from a television broadcast). It is widely used as a form of information entry from printed paper data records, whether passport documents, invoices, bank statements, computerised receipts, business cards, mail, printouts of static-data, or any suitable documentation. It is a common method of digitising printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

Figure 24 - advanced OCR pen

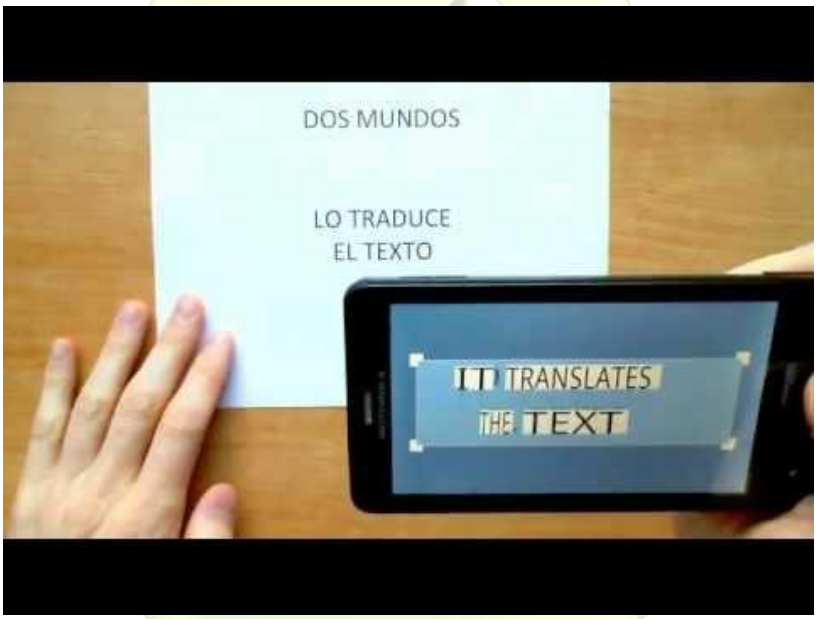
- Early versions needed to be trained with images of each character, and worked on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common, and with support for a variety of digital image file format inputs some systems are capable of reproducing formatted output that closely approximates the original page including images, columns, and other non-textual components.

➢ We have used OCR Android library to read class names

Figure 25 - Tesseract java library for OCR recognition

Android-ocr package

- An experimental app for Android that performs optical character recognition (OCR) on images captured using the device camera.
- Runs the Tesseract OCR engine using tess-two, a fork of Tesseract Tools for Android.
- Most of the code making up the core structure of this project has been adapted from the ZXing Barcode Scanner. Along with Tesseract-OCR and Tesseract Tools for Android (tesseract-android-tools), several open source projects have been used in this project, including leptonica, google-api-translate-java, microsoft-translator-java-api, and jtar.



Requirements

- A Windows Azure Marketplace Client ID and Client Secret (for translation) - Documentation
- A Google Translate API key (for translation) - Documentation

Training data for OCR

- A data file is required for every language you want to recognize. For English, this data file is included in the application assets and is automatically installed when the app is first run.

- For other languages (Spanish, French, Chinese, etc.), the app will try to download the training data from an old Google Code repository that is no longer available, and the download fails. So if you want to use training data for other languages, you'll need to package the appropriate training data files in the app or change the code to point to your own download location.

Introducing Tesseract

- Tesseract OCR is quite powerful, but does have the following limitations:

  - Unlike some OCR engines (like those used by the U.S. Postal Service to sort mail), Tesseract is unable to recognize handwriting and is limited to about 64 fonts in total.
  - Tesseract requires a bit of preprocessing to improve the OCR results; images need to be scaled appropriately, have as much image contrast as possible, and have horizontally-aligned text.
  - Finally, Tesseract OCR only works on Linux, Windows, and Mac OS X.

How does Tesseract work

- The first step is a connected component analysis in which outlines of the components are stored.
- This is a computationally expensive design decision at the time, but has a significant advantage: by inspection of the nesting of outlines, and the number of child and grandchild outlines, it is simple to detect inverse text and recognize it as easily as black-on-white text.

- Tesseract is probably the first OCR engine able to handle white-on-black text so trivially. At this stage, outlines are gathered together, purely by nesting, into Blobs.
- Blobs are organized into text lines, and the lines and regions are analyzed for fixed pitch or proportional text.
- Text lines are broken into words differently according to the kind of character spacing. Fixed pitch text is chopped immediately by character cells.

- Proportional text is broken into words using definite spaces and fuzzy spaces.
- Recognition then proceeds as a two-pass process.
- In the first pass, an attempt is made to recognize each word in turn. Each word that is satisfactory is passed to an adaptive classifier as training data. The adaptive classifier then gets a chance to more accurately recognize text lower down the page. Since the adaptive classifier may have learned something useful too late to make a contribution near the top of the page, a second pass is run over the page, in which words that were not recognized well enough are recognized again. A final phase resolves fuzzy spaces, and checks alternative hypotheses for the x-height to locate small-cap text.

➢ Then we can display all instruction to user to reach his target location
- On click on SCAN it button you will open OCR system to recognize you current location
- On click on Get route button you will get instruction to get the best route from you current location to your required destination see figure 26

Figure 26 - indoor navigation routing algorithm and giving instructions to user

Student planner Activity

Problem Statement

- Sometimes there's a gap between students and staff when it comes to determining deadlines and quizzes times, etc. So the purpose of this task is to keep students up-to-date about the above mentioned.

Technical Details

- Using (CompactCalendarView) class from a Github repository we create an empty calendar that can be filled with events using the function (eventMaker) which has the following signature void eventMaker(String date, String event), so this function takes an event and the date when event occurs then it stores those events in a Shared Preference, this way the user will be able to access it anytime even offline. When the user clicks any day on the calendar the events on that day are displayed in a box under the calendar.

Figure 27 - student planner icon in application

Figure 28 - example of day with events

Figure 29 - example on day without events

- See days with events in figure 28
- See days without events in figure 29

Code from the Application

- Code of the main activity of the calendar see example Appendix A.4.1
- The XML file of the calendar see example Appendix A.4.2
- Code of my_border.xml file see Appendix A.4.3

Summary

Very important user of the application is the students, and one of the most important things to any students is his academic time table, so providing it to him with online updates is a great contribution in this application

Courses Activity

Features

- Screen of this activity contains buttons for courses for all departments' courses

Problem statement

- We want to help students to get all information about all courses of every department that studied in the university
- By selecting courses button you can choose the department you want to know information about it

Results

- We have used guided links to all the courses in all academic years see figure 30,31
- Another integrations in this task can get data directly from the database of the server and display it directly in the application

Figure 30 - main activity of courses task

Figure 31- some course on Benha website

Some code of the task

- Start activity of the courses activity see example Appendix A.5.

Exam Results

- As a small task we can lead student to the results website to get their exam results
- See figure 32 for design of the task

Figure 32 - results activity on Benha website

Portal of UNIVERSITY ACTIVITY

Problem Statement

- A student may find it difficult to have to switch between apps often to get some information about a certain course or a test result.

Technical Details

- Using Web View we can open any website inside any activity using this line of code `webView.loadUrl("http://www.bu.edu.eg/");`.
- Then we can add some features, like zooming like the following: `webView.getSettings().setBuiltInZoomControls(true);` , adding Key Listener to be able to go back to the previous page when you hit the back button

Code from Application

- Portal Activity see example Appendix 6.1
- And here's a screenshot from the app running see figure 34,35

Figure 33 - portal activity to open portal of Benha website portal

Figure 34 - logo of portal Activity on App

Educational Mails Activities

Technical Details

- We used the Android browser to display and open many important educational mails like: EKB (Egyptian Knowledge Bank) , student Microsoft mail in office 356 and the Egyptian Universities Libraries website

Figure 36 - logo of office 365

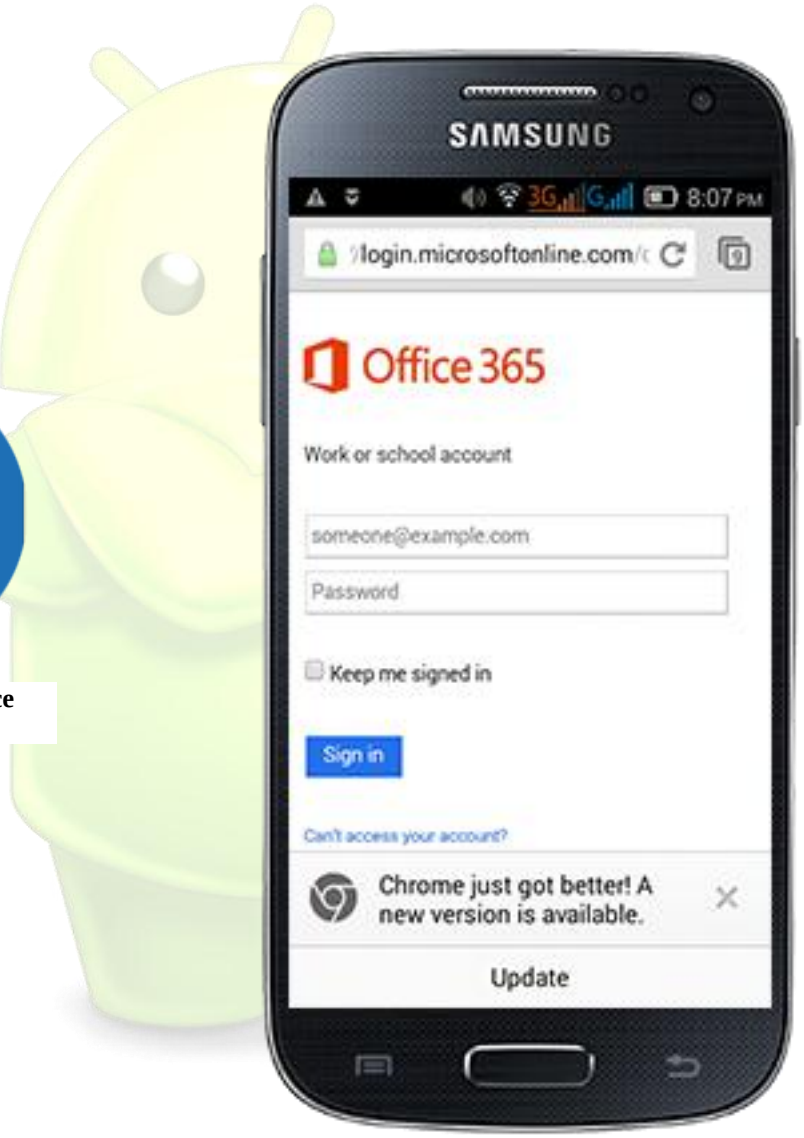


Figure 35 - office 365 on App

Figure 37 - EKB on App

Figure 38 - logo of EKB on App

Figure 39 - ICTP on App

Figure 40 - logo of ICTP on App

Mission and Vision

Technical Details

- We have gave separated tabs for the mission and vision and this tab contains list of all colleges of Benha University and you can navigate to your college to know about its mission and vision
- On the click on any college the mission and vision will displayed in the upper text view
- Tabs in main screen see figure 42

Figure 41 - Mission and vision fragment on Main Activity

- List of all colleges in Benha university

Figure 42 - list of all collages in Benha university

- Mission and Vision of college see figure 44

Figure 43 - showing Mission and Vision of each collage in Benha University

- Here we have used more than one layer of layouts relative and linear layouts (vertical and horizontal) and all of this is to guarantee the easy access to any part of the activity

Code from the Application

- Code of main Activity that contains the tabs for menu and the anther tab for mission and vision see example Appendix A.7.1
- Code of the Mission and vision fragment see example Appendix A.7.2
- Part from fragment_main_pagefrag.xml that contains the all missions and vision see example Appendix A.7.3

News gallery

Technical Details

- Utilizing Recycler View and Card View we can achieve the required form of our gallery. Every card has 4 components: Image View which is a container for the photo, Text View for the description of the photo, two buttons one for the Like, another for dislike.

- Code the contents of each card see example Appendix A.8.1

- The most complex part is the adapter; the ItemAdapter is the class which does that function. This class extends RecyclerView.Adapter and overrides the following methods: onCreateViewHolder, onBindViewHolder, getItemCount.
- When you click the image a new activity is created and here's the executed code, notice that we use Serializable to transfer data between the adapter class and this class

- Code of the gallery Adaptor class see Appendix A.8.2
- And here's a screenshot of the gallery running.
- For recycled view see figure 45
- For detailed view data see figure

Figure 44 - The activity with Recycler View

Figure 45 - The activity images, description, likes and hates

Figure 46 - logo of news gallery

Emergency Info Activity

Technical Details

- We use android calls to call emergency numbers
- Photos from the activity

Figure 47 - logo of emergency info in App

Figure 49 - emergency numbers in App

Figure 48 - emergency calls in App

Dimensions

Technical Details

Dimensions

- Android is a mobile operating system with very few limitations on its devices' hardware. Manufacturers can create devices of almost any screen shape, size, and density. Devices can have physical keyboards and buttons, or only virtual keyboards and buttons. While this flexibility is great for allowing device customization, it creates a few hurdles for application developers. First, how can apps support all of these various devices configurations with a consistent experience? Also, how can apps take advantage of devices that have higher end hardware or more features than others? Android was built with this in mind, and gives developers tools to support all device configurations, and optimize the experience for other device configurations, which will be covered later.

- In order for an application to be flexible and compatible with any device configuration, careful thought is required to make sure the user experience is appropriate across configurations. When creating an Android application, designers and developers must create user interfaces that work well with the small space of a phone, and the large space of a tablet. They also must take into account having image resources optimized for high and low density screens. While there are many ways to create optimized user interfaces for different screen sizes and orientations, the focus of this blog is how Android supports different screen densities.

- The basic design concept of Android is to have user interfaces that have elements that are about the same physical size, regardless of the screen density. Why? Simple, a user's finger is the same physical size no matter what the screen density is. A button or clickable item should render about the same physical size (larger than a fingertip) on any device. This also goes for text, words should render about the same (readable) font size across devices..

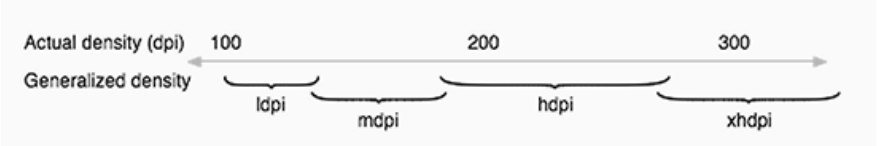
Screen Density, Not Resolution

- Screen density is a ratio of resolution and display size, which can be quantified as dots per inch, or dpi. The higher the dpi, the smaller each individual pixel is, and the greater clarity. Simply put, a higher dpi means more detail is displayed per inch, but does not necessarily correlate with a higher screen resolution. For example, the Galaxy Nexus (4.65" diagonal) has a 720x1280 px resolution, while the Nexus 7 (7" diagonal) has an 800x1280 px resolution. It is a common misconception to assume that they have about the same screen density, since their resolutions are almost identical. However, the Galaxy Nexus has a screen density of about 316 dpi and the Nexus 7 has a screen density of 216 dpi, not even close. This is because while they are displaying the same resolution, they are also displaying it in different amounts of space. Again, screen density is a ratio of resolution and display size, and both factors contribute to the density.

Density Buckets

- There is a myriad of Android devices with varying screen densities, which can range from 100 dpi to over 480 dpi. In order to optimize images for all these screen densities, images need to be created at different resolutions. However, trying to optimize every image resource for every possible density would be incredibly tedious, cause app sizes to be enormous, and simply is not a feasible solution. As a compromise, Android uses density "buckets" that are used to group devices together within certain screen density ranges. This way, apps are only required to optimize images for each density bucket, instead of every possible density.

- This keeps the workload reasonable for designers and developer, and also prevents the application size from ballooning. Of course, there is a tradeoff, leading to variance in the physical rendered size of images depending on device density, which will be shown later.



- dp - A density independent pixel. This is a density independent unit, however the physical size of a single "dp" is only approximately the same on every screen density. There are approximately 160 "dp" in an inch. A scaling factor, depending on the density bucket of the device, is applied to convert "dp" to the number of pixels at 160 dpi. The number of pixels a single "dp" translates to varies depending on the pixel on screen density and the density bucket the device falls into.

- The reason "dp" tends to vary in physical size is due to the same scaling factor being applied for the entire density bucket. The scaling factor is computed with the density bucket's dpi, and not the device's actual dpi. When the device's dpi is not exactly the same as its density bucket's dpi, the same amount of "dp" converts to the same amount "px". This leads to the same amount of "px" being displayed on different density screens, which render at different sizes.

- We used two files of dimensions to be flexible with different devices

- Dimen.xml and dimen(600dp).xml so if the device is bigger than 600dp in width it will take its dimension variables from dimen(600dp)
- Dimen.xml file see appendix A.9.1

Style

- Styles is an important part of any android application because this is the part which attracts the user to the application and without it the application is useless
- We need to observe this task in the styles and from the concept of the modular programming we tend to use one or group of style rules to manage all the application so it's easy to change manual or with programming

Technical Details

- We have used unified color xml file to choose color for any component of the application

- Color.xml file see example Appendix A.10.1
- We have used unified style files to choose give unified design for any component of the application

- One of the style file draw.xml for styling buttons see example Appendix A.10.2

Multi-lingual

Technical details

- We have created two files with the same variable names but different values, so when we start activity we can prompt user to choose language
- For the App working with Arabic language see figure 51
- For the App working with English language see figure 52
- We change language by a button in the setting buttons in figure 53
- Code to change language see example Appendix A.11.1
- One of the two files of applications names see Appendix A.11.2

Figure 51- English language selected

Figure 50- Arabic language selected

Setting list

Figure 53 - setting buttons

Figure 52 - moving news timeline TextView

Technical details

- Our settings contains
  - Sign in to lead to sign in activity in figure to check user see figure 55
  - Important website to lead to activity with many important educational websites see figure 56
  - Change language button as mentioned our App supports English and Arabic
  - Contact us to help contact with MIS of the project
  - Top of the page button to easy navigate in the long activity for code of the top of the page button see example A.12.1

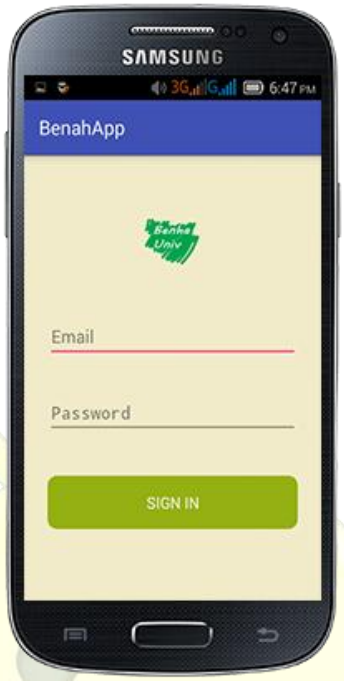


Figure 54 - Sign in Activity

Figure 55 - important websites Activity

News Timeline

Technical details

- It a TextView that takes text and rotate text in from of the user see figures 53 , 54
- For code of time line TextView see Appendix A.12.2

Safety rules

Technical details

- See figures 57 , 58 for details about our safety rules task

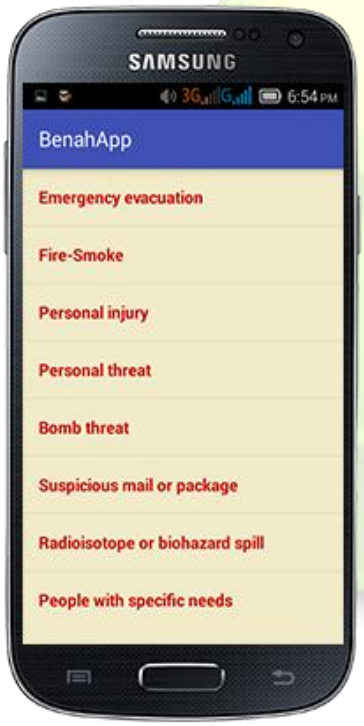


Figure 57 - menu of all safety rules

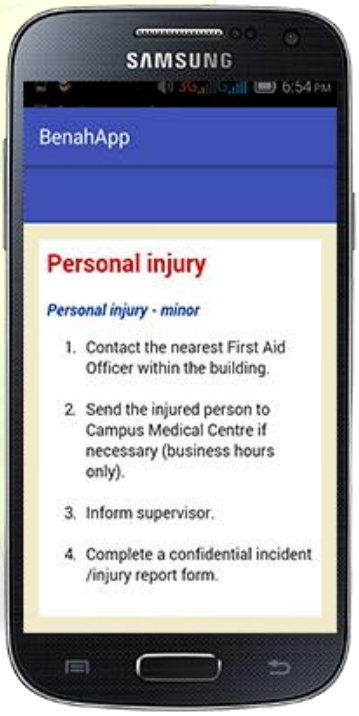


Figure 56 - details of Personal injury item

Outdoor Navigation

Technical details

- See figure 60 , 59 for details about list of all collages to help student choose collage he wants to reach
- On clicking on any collage name it will navigate to its location on Google maps
- See figure for Google Map navigation to help student reach his destination

Figure 59 - menu of all collages of Benha University

Figure 58 - location of Shoubra faculty of Engineering, EL-Kalafawy department on Google Maps

Questionnaire

Technical details

- Adding online questionnaire will increase interaction and adds new values to our system to help student and officials to discuss problems and solve it by participation Develop this part from app by answer some question about college staff and the college facilities
- We will need to use web service from university server to save this data on the university database for development and other usage
  See figures 60 for questionnaire data

Figure 60 - start Activity of Questionnaire

Database

Technical details

- We need database to search for :

1. faculty departments
2. faculty staff
3. faculty courses
4. courses from selecting staff names
5. doctors can select course to upload files for this course
6. Student can select course to get notification

Figure 61 - start of login activity

Figure 62 - choosing account parameters

Database mobile side

We have created android class to build tables for local database see example Appendix A.13.1

-We made join between tables to get the target. We create object of DatabaseHelper class see Appendix A.13.1 and used this tables to fill spinner for search

Database Web service side

- For testing, we used tomcat server that is connected to SQL server and Android is the client that sends data to server. Server receives data to store it in SQL server or to verify if this data is correct or not from SQL server database then return response to the client

- We need server to upload data to be accessed by all users via internet, and any update happens by doctors, all users can see it from notification.
- Server needs to store data that will be received. We used SQL server, server connects to SQL server and calls data from SQL server then returns response to android client .

- we create many tables in SQL server

  - signup and login tables

    Users can login with educational mail directly but we don't have a live database to check if this email is correct or not so we create signup button and put if condition to type only educational mail see figures 63.

    Users can sign up with:
    1-the email of university
    2-password
    3-job: Student or doctor

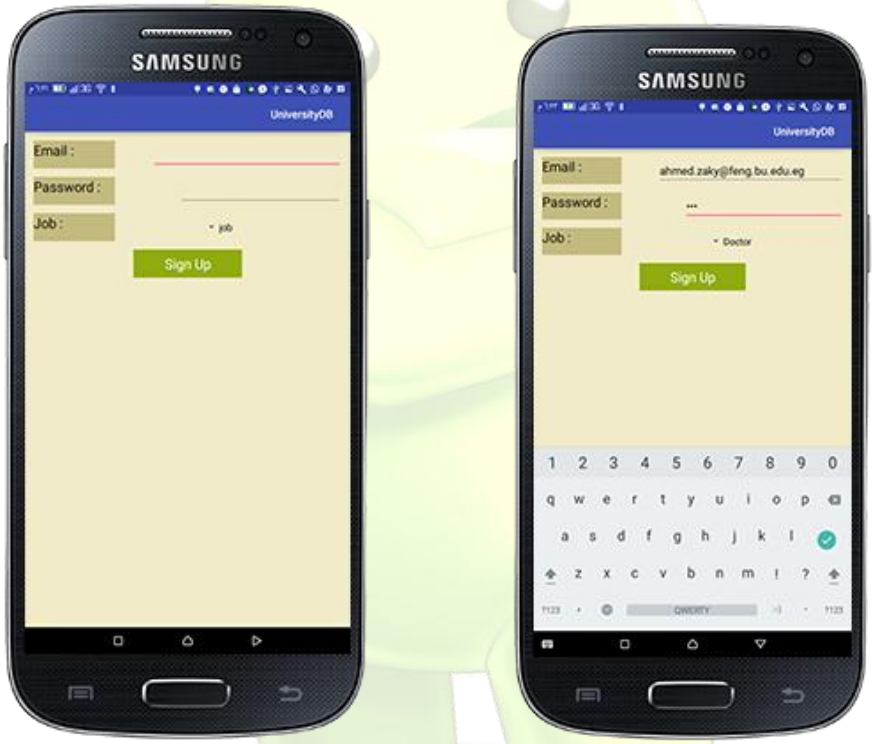


Figure 63 - sign up with valid educational mail

This data is sent to server and server inserts it in database SQL server, when user login User data is sent to server then database, then server responses with the job of user

- In detail:
  - Android sends and receives data from server with the URL that consists of http:\\Ip address: port number \web project name\servlet class name
  - code of sing up class see example Appendix A.13.2
  - code of login class see example Appendix A.13.3

- Notification
  - Doctors can upload files for students; Files will be uploaded on the server with the name of course. If students send notification request for particular course, server will send response with notification of name and path uploaded file.
  - See figures 64, 65 for the notifications activity

Figure 65 - choose notification course

Figure 64 - notification of programing course

- Upload files to server

  Doctor should select course firstly, then browse button opens file chooser and when doctor upload files to server so Server receives file name, course name
  , then save them with file URL in database see example Appendix A.13.4 and see figures 66, 67

Figure 66 - choose course to upload data

Figure 67 - uploading files for courses

- This is server output and SQL server see example Appendix
- We created upload File class that will take parameters: file path , URL server , course name from Upload Activity see example Appendix A.13.5

- This class:
  1. creates http connection
  2. Reads data from selected file by file input stream.
  3. writes this data by output stream on the connection
  4. connects connection
  5. On the other hand server will read that is written on this connection and generates a new file with particular path to write that is read in it.
  6. server will save its path in data base

➢ Server code

- On the other hand student in its student home page can select course to get notification for this course, server will return response to student with list of notification if there is notification see example Appendix A.13.6 and see figure

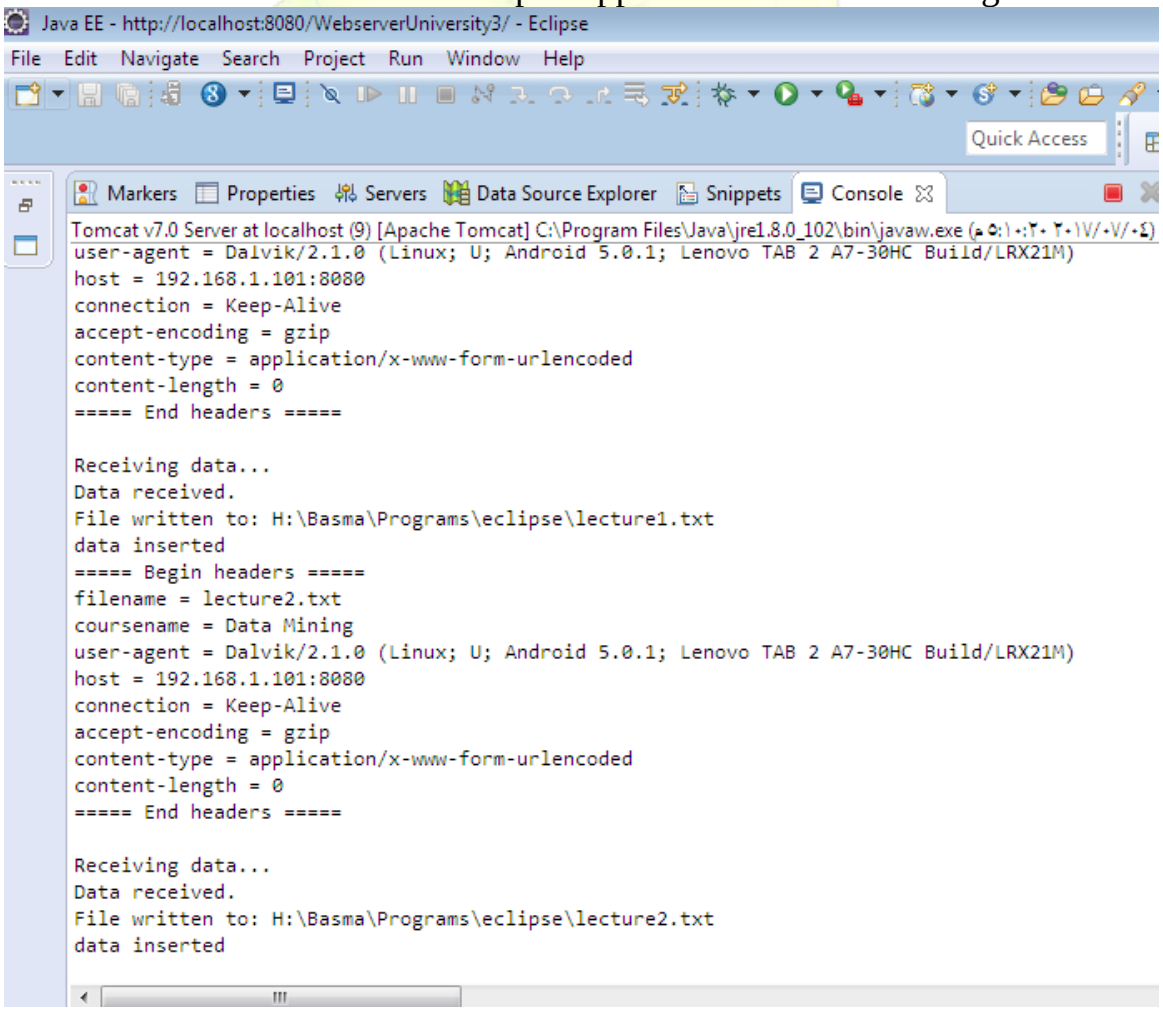


Figure 68 - response from tomcat server on eclipse

Some basics classes in android to deal with server

1- loginData class
   - To take the URL and data that will be sent to server and set them in list see Appendix A.13.7
2- loginParser class To send and receive data see example Appendix A.13.8

Permissions and Dependencies

- We have used many android permissions in our applications to help using camera , internet , external SD card , on device applications and so on
- Some of used permissions

```
<uses-feature android:name="android.hardware.camera"
android:required="false" />
    <uses-feature android:name="android.hardware.camera.front"
android:required="false" />

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-feature android:name="android.hardware.camera.autofocus"
/>
<uses-permission android:name="android.permission.CALL_PHONE"/>
```

- We also have used many libraries from android studio and used gradle to synchronize them
- Some of used dependencies

```
dependencies {
      compile fileTree(dir: 'libs', include: ['*.jar'])
      testCompile 'junit:junit:4.12'
      compile 'com.android.support:appcompat-v7:23.4.0'
      compile 'com.google.zxing:core:3.2.1'
      compile 'com.android.support:design:+'
      compile 'com.android.support:cardview-v7:21.+'
      compile "com.squareup.picasso:picasso:2.4.0"
      compile 'com.github.sundeepk:compact-calendar-view:1.8.4'

      // libraries for OCR
      compile 'com.rmtheis:tess-two:6.0.4'
      compile files('libs/google-api-translate-java-0.98-
      mod2.jar')
      compile files('libs/json_simple-1.1.jar')
      compile files('libs/jtar-1.0.4.jar')
      compile files('libs/microsoft-translator-java-api-0.6-
      mod.jar')
      compile 'com.jakewharton:butterknife:6.1.0'
}
```

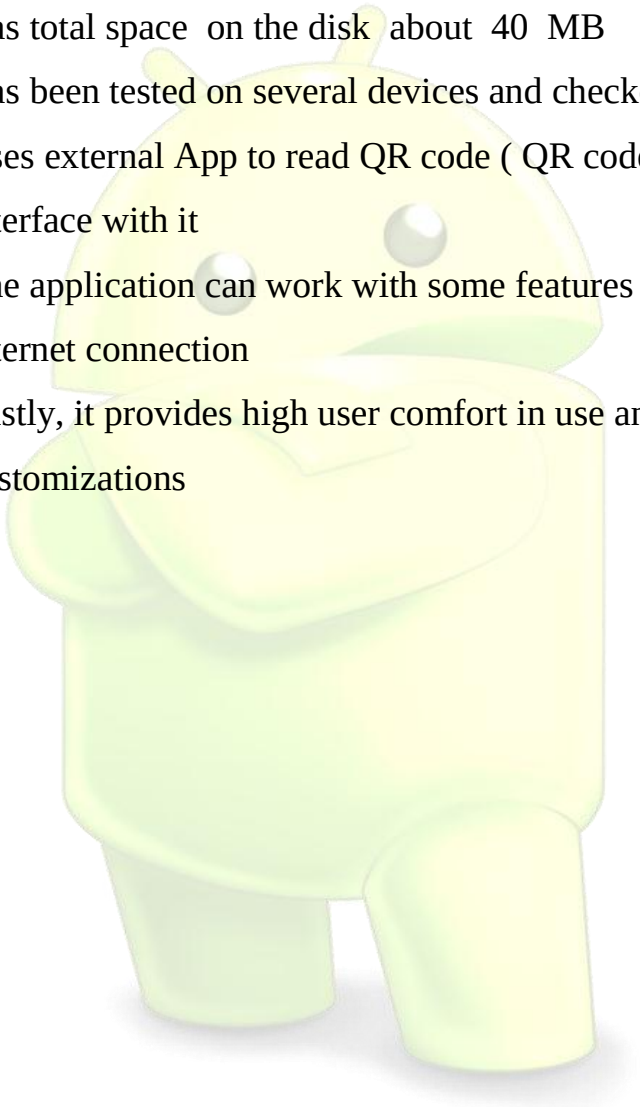
- External applications our App need
  - Csv viewer to view csv files of attendance
  - Barcode scanner to interface with the App
  - Vysor to view and control android from PC for presentation
  - Browser of the device to open some websites like the EKB, office 365

Conclusion

In this chapter we saw all the snap shots of all the activities, along with the features provided to the user. The image icon is representation of the University Application. The first activity has a user interface for the users to activity he needs to do. The second activity is the indoor navigation to help new students navigate through the university. The third activity would give us brief details about the University; it includes the Image & the Full details, vision, mission, phone numbers and maps of the colleges, Review, Location & Phone Number. It also provides the user with various options such as to Map Location, Full Review List in the Browser which allows the user to get to the edu mails, Egyptian EKB and University portal. The application was tested with various inputs and the results were obtained as expected.

Statics about Benha App

- About 2500 line code
- About 25 separated module and more
- About 15 separated java packages
- Supports 2 languages (Arabic , English)
- Has total space on the disk about 40 MB
- Has been tested on several devices and checked for portability
- Uses external App to read QR code (QR code scanner) and has interface with it
- The application can work with some features off when no internet connection
- Lastly, it provides high user comfort in use and some customizations



Appendix A (CODE)

A.1 – Main menu

Example A.1.1 - Main menu fragment

```
public class studentservicefragment extends Fragment {

    public static final String ALERT = "ALERT";
    public static final String LIBRARY = "LIBRARY";
    public static final String MAP = "MAP";
    public static final String ATTENDANCE = "ATTENDANCE";
    public static final String STAFF = "STAFF";
    public static final String OFFICE = "OFFICE 365";
    public static final String EKB = "EKB";
    public static final String PORTAL = "PORTAL";
    public static final String GALLERY = "Gallery";
    public static final String COURSE = "courses";
    public static final String CALENDER = "Time Table";

    View v;
    Toolbar toolbar;
    public List<ViewHolder> ArrayOfData=new ArrayList<>();
    public studentservicefragment() {
        // Required empty public constructor
    }



    private static final String[] COUNTRIES = new String[] {
            "Belgium", "France", "Italy", "Germany", "Spain"
    };

    public void searchAutoComplete2()
    {
        ArrayAdapter<String> adapter = new
ArrayAdapter<String>(getActivity(),
                android.R.layout.simple_dropdown_item_1line,
COUNTRIES);
        AutoCompleteTextView textView = (AutoCompleteTextView)
                v.findViewById(R.id.autoCompleteTextView);
        textView.setAdapter(adapter);
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
      v=inflater.inflate(R.layout.fragment_startfragment,
container, false);


        Button Go = (Button) v.findViewById(R.id.buttonGo);

        Go.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getActivity() ,
pluserActivity.class);
```

```
                startActivity(intent);
            }
        });
        searchAutoComplete2();

        GridView g=(GridView)v.findViewById(R.id.gridView);

        ViewHolder v1=new ViewHolder();
        v1.settitle(ALERT);
        v1.setid(R.drawable.ic_alert);

        ViewHolder v2=new ViewHolder();
        v2.settitle(LIBRARY);
        v2.setid(R.drawable.library);



        ArrayOfData.add(v1);
        ArrayOfData.add(v2);
        ArrayOfData.add(v3);
        ArrayOfData.add(v4);

        ImageAdapter a=new
ImageAdapter(getContext(),R.layout.griditem,ArrayOfData);
        g.setAdapter(a);

        g.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {

                ViewHolder vh=ArrayOfData.get(position);
                Log.v("xtest" , "onclick items");


                if(vh.gettitle().equals(OFFICE))
                {
                    Intent i=new Intent(Intent.ACTION_VIEW,
Uri.parse("https://outlook.office.com/owa/"));
                    startActivity(i);
                }
                else if(vh.gettitle().equals(STAFF))
                {
                    Intent i=new Intent(Intent.ACTION_VIEW,
Uri.parse("http://www.bu.edu.eg/portal/index.php?act=104"));
                    startActivity(i);
                }
                else if(vh.gettitle().equals(ATTENDANCE))
                {
                    Intent i=new Intent( getActivity(),
AttendenceActivity.class);
                    startActivity(i);
                }
                else if(vh.gettitle().equals(LIBRARY))
                {
                    Intent i=new Intent(Intent.ACTION_VIEW,
Uri.parse("http://srv3.eulc.edu.eg/eulc_v5/libraries/start.aspx
n"));
                    startActivity(i);
                }else if(vh.gettitle().equals(EKB))
                {
```

```
                        Intent i=new Intent(Intent.ACTION_VIEW,
        Uri.parse("http://www.ekb.eg/login"));
                        startActivity(i);
                    }
                    else if(vh.gettitle().equals(PORTAL))
                    {
                        Intent i=new Intent( getActivity(),
        portalMain.class);
                        startActivity(i);
                    }
                    else if(vh.gettitle().equals(GALLERY))
                    {
                        Intent i=new Intent( getActivity(),
        galleryMain.class);
                        startActivity(i);
                    }else if(vh.gettitle().equals(COURSE))
                    {
                        Intent i=new Intent( getActivity(),
        courseMain.class);
                        startActivity(i);
                    }
                    else if(vh.gettitle().equals(CALENDER))
                    {
                        Intent i=new Intent( getActivity(),
        calenderMain.class);
                        startActivity(i);
                    }
                }
                });

            Go.requestFocus();


            return v;
        }
    }
```

Example A.1.2 – Adaptor class

```
public class ImageAdapter  extends BaseAdapter {

    Context context;
    private int layoutResourceId;
    List<ViewHolder> ArrayOfData =new ArrayList<>();
    LayoutInflater inflater;


    public ImageAdapter(Context context,int layoutResourceId,List
ArrayOfData)
    {

        this.context= context;
       Log.v("xtest" , context + " ImageAdapter");
        this.layoutResourceId=layoutResourceId;
        this.ArrayOfData=ArrayOfData;

    }

    @Override
    public int getCount() {
        return ArrayOfData.size();

    }
```

```
        @Override
        public ViewHolder getItem(int i) {
           return ArrayOfData.get(i);

        }

        @Override
        public long getItemId(int i) {

            return i;
        }

        @Override
        public View getView(int i, View view, ViewGroup parent) {
            View v=view;
            ImageView img ;
            TextView t;
            Log.v("xtest" , "getview view == null " + (v==null) );

            if(v==null)
            {


                inflater = (LayoutInflater)
    context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

                v = inflater.inflate(R.layout.griditem, null);
                img=(ImageView)v.findViewById(R.id.grid_item_image) ;
                 t=(TextView)v.findViewById(R.id.grid_item_title);



            }
            else
            {
                img = (ImageView)v.findViewById(R.id.grid_item_image);
                t=(TextView)v.findViewById(R.id.grid_item_title);
            }
            img = (ImageView) v.findViewById(R.id.grid_item_image);
            img.setImageResource(ArrayOfData.get(i).getid());
             t.setText(ArrayOfData.get(i).gettitle());
            return v;
        }

    }
```

A.2 – Attendance recorder

Example A.1.2 – Code to interface with ZXING QR code scanner

```
public class QR_factory {


    private static final int BLACK = 0;
    private static final int WHITE = 1;
    private static final int WIDTH = 50;
    public static String data = "";
    public static AppCompatActivity app ;
    public static String QR_data = "none";
    public static void open_QR_reader(AppCompatActivity ma) {
        try {

            Intent intent = new
Intent("com.google.zxing.client.android.SCAN");
            intent.putExtra("SCAN_MODE", "QR_CODE_MODE"); //
"PRODUCT_MODE for bar codes
            app = ma;
            ma.startActivityForResult(intent, 0);
        } catch (Exception e) {

            Uri marketUri =
Uri.parse("market://details?id=com.google.zxing.client.android");
            Intent marketIntent = new
Intent(Intent.ACTION_VIEW,marketUri);
            ma.startActivity(marketIntent);

        }

    }

    public static void create_QR_code(String STR ,
AppCompatActivity ma) {
       // ImageView imageView = (ImageView)
ma.findViewById(R.id.imageView);
        Toast.makeText(ma, "creator here",
                Toast.LENGTH_LONG).show();
        try {
            Bitmap bitmap = encodeAsBitmap(STR);
        //    imageView.setImageBitmap(bitmap);
        } catch (WriterException e) {
            e.printStackTrace();
        }
    }

    public static Bitmap encodeAsBitmap(String str) throws
WriterException {
        BitMatrix result;
        try {
            result = new MultiFormatWriter().encode(str,
                    BarcodeFormat.QR_CODE, WIDTH, WIDTH, null);
        } catch (IllegalArgumentException iae) {
            // Unsupported format
            return null;
        }
        int w = result.getWidth();
        int h = result.getHeight();
        int[] pixels = new int[w * h];
```

```
            for (int y = 0; y < h; y++) {
                int offset = y * w;
                for (int x = 0; x < w; x++) {
                    pixels[offset + x] = result.get(x, y) ? BLACK :
WHITE;
                }
            }
            Bitmap bitmap = Bitmap.createBitmap(w, h,
Bitmap.Config.ARGB_8888);
            bitmap.setPixels(pixels, 0, WIDTH, 0, 0, w, h);
            return bitmap;
        }


        public static void process_result(int requestCode , int
resultCode , Intent data) {
            if (requestCode == 0) {

                if (resultCode == app.RESULT_OK) {
                    String contents =
data.getStringExtra("SCAN_RESULT");
                    Toast.makeText(app, contents,
Toast.LENGTH_LONG).show();

                    QR_data = contents.substring(contents.indexOf("-
"),contents.length())
                            .replaceAll("-"," ").trim();
//                  frontCameraController.checker();

                    attendance_servant.check_bad_student(QR_data);
                    if(!QR_factory.QR_data.equals("none"))

data_factory.setAttendence(QR_factory.QR_data);
                    data_factory.saveLast(data_factory.getLastData());
                }
                if(resultCode == app.RESULT_CANCELED){
                    //handle cancel
                }
                AttendenceActivity.cameraBt.setEnabled(true);
            }
        }
}
```

Example A.2.2 - Code saving and loading data from files

```
public class data_factory {

    public static AppCompatActivity app;
    public static ArrayList<String> studentData = new
ArrayList<>();
    public static ArrayList<String> pluserismData = new
ArrayList<>();
    public static File goalFile;
    public static String globalName = "class "+new
SimpleDateFormat("yyyy_MM_dd_HH_mm").format(new Date()) + ".csv";
    private static final int FILE_SELECT_CODE = 1;

    public static  void showFileChooser(AppCompatActivity ma) {
        app = ma;
        Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
        intent.setType("*/*");
```

```
            intent.addCategory(Intent.CATEGORY_OPENABLE);

            try {
                app.startActivityForResult(
                        Intent.createChooser(intent, "Select a File to
Upload"),
                        FILE_SELECT_CODE);
            } catch (android.content.ActivityNotFoundException ex) {
                // Potentially direct the user to the Market with a
Dialog
                Toast.makeText(app, "Please install a File Manager.",
                        Toast.LENGTH_SHORT).show();
            }
        }

        public static void check_file(int requestCode, int resultCode
, Intent data) {

            if ( requestCode ==  FILE_SELECT_CODE ){
                if (resultCode == app.RESULT_OK) {
                    // Get the Uri of the selected file
                    Uri uri = data.getData();

                    // Get the path
                    String path = null;
                    try {
                        path = getPath(app, uri);
                    } catch (URISyntaxException e) {
                        e.printStackTrace();
                    }

                    // Get the file instance
                    goalFile = new File(path);
                    // Initiate the upload
                    studentData = new ArrayList<String>();
                    pluserismData = new ArrayList<String>();
                    get_CSV_data();

                }
            }

        }

        public static String getPath(Context context, Uri uri) throws
URISyntaxException {
            if ("content".equalsIgnoreCase(uri.getScheme())) {
                String[] projection = { "_data" };
                Cursor cursor = null;

                try {
                    cursor = context.getContentResolver().query(uri,
projection, null, null, null);
                    int column_index =
cursor.getColumnIndexOrThrow("_data");
                    if (cursor.moveToFirst()) {
                        return cursor.getString(column_index);
                    }
                } catch (Exception e) {
                    // Eat it
                }
            }
            else if ("file".equalsIgnoreCase(uri.getScheme())) {
                return uri.getPath();
            }
```

```
            return null;
        }


        public static Boolean get_txt_data (File f) {

            boolean state ;

            BufferedReader reader ;
            try {
                reader = new BufferedReader(new FileReader(f));
                studentData = new ArrayList<>();
                pluserismData = new ArrayList<>();
                globalName = null;
                String line;
                globalName = reader.readLine();
                while ((line = reader.readLine()) != null) {
                    studentData.add(line);
                }
                Log.v("xtest get_txt_data", " last file "+
    studentData.size());
                state = true;
            } catch (IOException e) {
                Log.v("xtest get_txt_data", " cannot find last file");
                state = false;
                e.printStackTrace();
            }

            return state;
        }

        public static Boolean get_CSV_data () {

            boolean state ;

            BufferedReader reader = null;
            try {
                reader = new BufferedReader(new FileReader(goalFile));

                String line;
                reader.readLine();
                studentData = new ArrayList<String>();
                pluserismData = new ArrayList<String>();
                globalName = "class "+new
    SimpleDateFormat("yyyy_MM_dd_HH_mm").format(new Date()) + ".csv";

                while ((line = reader.readLine()) != null) {

                    String[] RowData = line.split(",");

                    if(RowData.length > 3) {
                        studentData.add(formatItem(RowData[0],
    RowData[1], RowData[RowData.length-1]));
                    }
                }

                saveLast(getLastData());
                Toast.makeText(app ,"working by " + studentData.size()
    + " student", Toast.LENGTH_SHORT).show();
                state = true;
            } catch (IOException e) {
                e.printStackTrace();
                state = false;
```

```
        }

        return state;
    }

    private static String formatItem(String id , String name ,
String num) {
        if(num == "")
            num = "0";

        return  id.trim() + " - "  + name.trim() + " - "+
num.trim();
    }

    public static boolean setAttendence(String item) {

        for (int i = 0; i < studentData.size(); i++) {
            if(studentData.get(i).contains(item))
            {
                updateItem(i);
                return true;
            }
        }

        return false;
    }

    public static boolean removeAttendence(String item) {
        for (int i = 0; i < studentData.size(); i++) {
            if(studentData.get(i).contains(item))
            {
                lowerItem(i);

                QR_factory.QR_data = "none";
                return true;
            }
        }

        return false;
    }

    public static boolean loadLast()
    {
        return get_txt_data(getOutputLastFile());
    }

    public static void updateItem(int index) {

        String str []= studentData.get(index).split("-");
        int tmp = Integer.parseInt(str[2].trim());
       studentData.set(index , formatItem(str[0] , str[1] ,
(++tmp) +"") );
    }
    public static void lowerItem(int index) {

        String str []= studentData.get(index).split("-");
        int tmp = Integer.parseInt(str[2].trim());
        studentData.set(index , formatItem(str[0] , str[1] , (--
tmp) +"") );
    }


    public static void saveLast(String data)
```

```
        {
            File dataFile = getOutputLastFile();

            if(dataFile == null){

                return;
            }

            try{

                FileOutputStream fos = new FileOutputStream(dataFile);
                Writer out = new BufferedWriter(new
    OutputStreamWriter(
                        fos, "UTF-8"));
                try {
                    out.write("");
                    out.append(data);
                } finally {
                    out.close();
                }

            }catch(FileNotFoundException e){

            } catch (IOException e){

            }

        }

        public static void saveAttendence(String data)
        {

            File dataFile = getOutputMediaFile();

            if(dataFile == null){

                return;
            }

            try{

                FileOutputStream fos = new FileOutputStream(dataFile);
                Writer out = new BufferedWriter(new
    OutputStreamWriter(fos,"utf-8"));
                try {
                    out.write(data);

                } finally {
                    out.close();
                }


            }catch(FileNotFoundException e){

            } catch (IOException e){

            }

        }


        private static File getOutputLastFile(){
            // To be safe, you should check that the SDCard is mounted
```

```
        // using Environment.getExternalStorageState() before
doing this.

        File mediaStorageDir = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIR
ECTORY_DOCUMENTS),"BenhApp");


        if(!mediaStorageDir.exists()){
            if(!mediaStorageDir.mkdirs()){
                return null;
            }
        }

        File lastFile;
        lastFile = new
File(mediaStorageDir.getPath()+File.separator+"last"+".txt");
        return lastFile;
    }


    private static File getOutputMediaFile(){
        // To be safe, you should check that the SDCard is mounted
        // using Environment.getExternalStorageState() before
doing this.

        File mediaStorageDir = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIR
ECTORY_DOCUMENTS),"BenhApp");

        // This location works best if you want the created images
to be shared
        // between applications and persist after your app has
been uninstalled.

        // Create the storage directory if it does not exist
        if(!mediaStorageDir.exists()){
            if(!mediaStorageDir.mkdirs()){
                return null;
            }
        }


        File mediaFile;
        mediaFile = new
File(mediaStorageDir.getPath()+File.separator+globalName);
        return mediaFile;
    }


    public static String getStringData() {
        String temp = "";
        for (int i = 0; i < studentData.size(); i++) {
            temp += studentData.get(i).replaceAll("-",",")+"\n";
            }
        return  temp;
        }

    public static String getLastData() {
        String temp = globalName + "\n";
        for (int i = 0; i < studentData.size(); i++) {
            temp += studentData.get(i) +"\n";
        }
        return  temp;    }
```

Example A.2.3 – Code for accelerometer

```
public class Movement_sensor {

    // Start with some variables
    private static SensorManager sensorMan;
    private static Sensor accelerometer;
    private static float[] mGravity;
    private static float mAccel;
    private static float mAccelCurrent;
    private static float mAccelLast;
    private static AppCompatActivity ACA;
    private static SensorEventListener SEL ;
    private static int index = 0 ;
    private static int timer = 0 ;
    private static int state = 0;

    public static void start_Accelerator(AppCompatActivity ma) {
        ACA = ma;
        SEL = (SensorEventListener) ma;
        sensorMan =
(SensorManager)ma.getSystemService(ACA.SENSOR_SERVICE);
        accelerometer =
sensorMan.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        mAccel = 0.00f;
        mAccelCurrent = SensorManager.GRAVITY_EARTH;
        mAccelLast = SensorManager.GRAVITY_EARTH;
    }

    public  static boolean is_stable()
    {
        return state < 2;
    }
    public static void setState() {
        Movement_sensor.state++;
    }

    public static void resetState() {
        Movement_sensor.state = 0;
    }

    public static void resume_Accelerometer() {
        sensorMan.registerListener(SEL , accelerometer,
SensorManager.SENSOR_DELAY_UI);
    }

    public static void pause_Accelerometer()
    {
        sensorMan.unregisterListener(SEL);
    }

    public static void sensor_changed(SensorEvent event) {

        if ( (timer++) % 8 == 0 ) {
            if ( event.sensor.getType() ==
Sensor.TYPE_ACCELEROMETER ) {
                mGravity = event.values.clone();
                // Shake detection
                float x = mGravity[0];
                float y = mGravity[1];
                float z = mGravity[2];
                mAccelLast = mAccelCurrent;
```

```
                    mAccelCurrent = (float) Math.sqrt( x * x +  y * y
    + z * z) ;

                    float delta = mAccelCurrent - mAccelLast;
                    //mAccel = mAccel * 0.9f + delta;
                    mAccel = Math.abs( delta);
                    // Make this higher or lower according to how much
                    // motion you want to detect
                    if (mAccel > 0.8) {
                        index++;

                    } else if ( mAccel < 0.1 && index > 2) {
                        Toast.makeText(ACA, "I have sensed move! " +
    (index), Toast.LENGTH_SHORT).show();
                        setState();
                        index = 0;

                    }
                }
            }
        }

    }
```

A.3 – Indoor Navigation

Example A.3.1 – Code of our A-star

```
public class ocr_servent {

    public static ArrayList<Node> classes = new ArrayList<Node>();

    public static boolean first_time = true ;

    public static boolean check_values(String current , String
dest) {
        boolean goal_b = false , dest_b = false ;

        if (dest.matches("SB\\d-\\d{2}") || dest.matches("SB\\d-
\\d")) {
            int floor =
Integer.parseInt(dest.substring(2).split("-")[0]);
            int num = Integer.parseInt(dest.split("-")[1]);
            if (num < 10 && num > 0 && dest.matches("SB\\d-\\d"))
{
                dest = dest.substring(0, 4) + "0" + num;
            }
            if (floor >= 0 && floor <= 9 && num >= 1 && num <= 20)
{
                goal_b = true;
            }
        }

            if (current.matches("SB\\d-\\d{2}") ||
current.matches("SB\\d-\\d")) {
                int floor =
Integer.parseInt(current.substring(2).split("-")[0]);
                int num = Integer.parseInt(current.split("-")[1]);
                if (num < 10 && num > 0 && current.matches("SB\\d-
\\d")) {
                    current = current.substring(0, 4) + "0" + num;
                }
                if (floor >= 0 && floor <= 9 && num >= 1 && num <=
20) {
                    dest_b = true;
                }

            }

        return goal_b && dest_b ;

        }

    public static String checkRoute(String current_p, String
dest_p) throws IOException {


        if (check_values(current_p, dest_p)) {

            if(first_time) {
                add_classes(dest_p);
                add_adjacent();
                first_time = false;
            }
            if (get_node(dest_p) != null) {
```

```
                    AstarSearch(get_node(current_p),
get_node(dest_p));
                    List<Node> path = printPath(get_node(dest_p));
                    String route = path.toString();
                    route = route.substring(1, route.length() - 1);
                    return get_detailed_instructions(route);
                }
            }else {
                    return "Not a valid source or destination!";
                }

            return "Not a valid process!";
        }
    /*
        public static String get_instructions(String route) throws
    IOException {
            String inst = "";
            String steps[] = route.split(",");

            for (int i = 0; i < steps.length - 1; i++) {
                int c = getCost(steps[i].trim(), steps[i + 1].trim());
                int floor_s =
    Integer.parseInt(steps[i].trim().substring(2).split("-")[0]);
                int floor_d = Integer.parseInt(steps[i +
    1].trim().substring(2).split("-")[0]);
                int num_s = Integer.parseInt(steps[i].trim().split("-
    ")[1]);
                int num_d = Integer.parseInt(steps[i +
    1].trim().split("-")[1]);

                if (c == 4) {
                    if (floor_d > floor_s) {
                        inst += "- go upstairs\n";
                    } else if (floor_d < floor_s) {
                        inst += "- go downstairs\n";
                    } else {
                        if ( num_s == 11 ) {
                            inst += "- go upstairs\n";
                        } else {
                            inst += "- go downstairs\n";
                        }
                    }
                } else if (c == 2) {
                    inst += "- move to the other side\n";
                } else if (c == 1) {
                    inst += "- move 4m ";
                    if (num_d == 1 || num_s == 1) {
                        inst += "backwards ";
                    } else if (num_d >= 14 || num_d <= 21) {
                        if (num_d > num_s) {
                            inst += "left to class door";
                        } else {
                            inst += "right to class door";
                        }
                    } else if (num_d >= 3 || num_d <= 10) {
                        if (num_d > num_s) {
                            inst += "left to class door";
                        } else {
                            inst += "right to class door";
                        }
                    }
                    inst += "\n";
                }
            }
```

```
            return inst;
        }
    */

        public static String get_detailed_instructions(String route)
    throws IOException {
            String inst = "";
            String steps[] = route.split(",");

            for (int i = 0; i < steps.length - 1; i++) {
                int c = getCost(steps[i].trim(), steps[i + 1].trim());
                int floor_s =
    Integer.parseInt(steps[i].trim().substring(2).split("-")[0]);
                int floor_d = Integer.parseInt(steps[i +
    1].trim().substring(2).split("-")[0]);
                int num_s = Integer.parseInt(steps[i].trim().split("-
    ")[1]);
                int num_d = Integer.parseInt(steps[i +
    1].trim().split("-")[1]);

                if (c == 4) {
                    if (floor_d > floor_s) {
                        inst += "- go upstairs";
                    } else if (floor_d < floor_s) {
                        inst += "- go downstairs";
                    } else {
                        if ( num_s == 11 ) {
                            inst += "- go upstairs";
                        } else {
                            inst += "- go downstairs";
                        }
                    }
                } else if (c == 2) {
                    inst += "- move to the other side";
                } else if (c == 1) {
                    inst += "- move 4m ";
                    if (num_d == 1 || num_s == 1) {
                        inst += "backwards ";
                    } else if (num_d >= 14 || num_d <= 21) {
                        if (num_d > num_s) {
                            inst += "left to class door";
                        } else {
                            inst += "right to class door";
                        }
                    } else if (num_d >= 3 || num_d <= 10) {
                        if (num_d > num_s) {
                            inst += "left to class door";
                        } else {
                            inst += "right to class door";
                        }
                    }
                }
                inst += " >>> to class ( " + steps[i] + " )\n\n";
            }
            return inst;
        }


        public static void add_classes(String des) {
            int limit = 21;
            for (int i = 0; i < 10; i++) {
                if (i == 1) {
                    limit = 36;
                }
```

```
                for (int j = 0; j < limit; j++) {
                    String name = "SB" + i + "-" + ((j < 10) ? ("0" +
j) : j);
                    classes.add(new Node(name,
collage_model.get_heuristic(name, des)));
                }

            }
        }


        public static Node get_node(String t) {
            for (int i = 0; i < classes.size(); i++) {
                if (classes.get(i).value.equals(t)) {
                    return classes.get(i);
                }
            }
            return null;
        }

        public static void add_relation(int i, int num, int i2, int
num2, int cost) {
            String n = (num > 9) ? "" + num : "0" + num;
            String n2 = (num2 > 9) ? "" + num2 : "0" + num2;
            get_node("SB" + i + "-" + n).add_link(new
Edge(get_node("SB" + i2 + "-" + n2), cost));
            get_node("SB" + i2 + "-" + n2).add_link(new
Edge(get_node("SB" + i + "-" + n), cost));
        }

        private static void add_adjacent() {

            //11,13,  14,10  18,6
            for (int i = 1; i < 10; i++) {

                add_relation(i, 6, i - 1, 6, 4);
                add_relation(i, 18, i - 1, 18, 4);

                add_relation(i, 14, i, 13, 4);
                add_relation(i, 10, i, 11, 4);
                add_relation(i, 13, i, 11, 2);

                add_relation(i, 1, i, 2, 0);
                add_relation(i, 1, i, 20, 2);
                add_relation(i, 1, i, 4, 1);

                for (int j = 15; j < 22; j++) {
                    add_relation(i, j, i, j - 1, 1);
                    int tmp = 10 - j + 15;
                    add_relation(i, j - 1, i, tmp, 2);
                }

                for (int j = 4; j < 11; j++) {
                    add_relation(i, j, i, j - 1, 1);
                }

                if (i != 9) {
                    add_relation(i, 14, i + 1, 11, 4);
                    add_relation(i, 14, i + 1, 13, 4);

                    add_relation(i, 10, i + 1, 11, 4);
                    add_relation(i, 10, i + 1, 13, 4);
                }
```

```
            }

            for (int i = 0; i < classes.size(); i++) {

            }


            return;
        }

        public static List<Node> printPath(Node target) {
            List<Node> path = new ArrayList<Node>();

            for (Node node = target; node != null; node = node.parent)
    {
                path.add(node);
            }

            Collections.reverse(path);

            return path;
        }

        public static void AstarSearch(Node source, Node goal) {

            Set<Node> explored = new HashSet<Node>();

            PriorityQueue<Node> queue = new PriorityQueue<Node>(20,
                    new Comparator<Node>() {
                        //override compare method
                        public int compare(Node i, Node j) {
                            if (i.f_scores > j.f_scores) {
                                return 1;
                            } else if (i.f_scores < j.f_scores) {
                                return -1;
                            } else {
                                return 0;
                            }
                        }

                    }
            );


            //cost from start
            source.g_scores = 0;

            queue.add(source);

            boolean found = false;

            while ((!queue.isEmpty()) && (!found)) {

                //the node in having the lowest f_score value
                Node current = queue.poll();

                explored.add(current);
                Log.e("www" , "checkpoint 2-4");
                //goal found
                if (current.value.equals(goal.value)) {
                    found = true;
                }
                Log.e("www" , "checkpoint 2-5 >>> current.adj " +
    current.adjacencies);
```

```
                //check every child of current node
                for (Edge e : current.adjacencies) {
                    Node child = e.target;
                    double cost = e.cost;
                    double temp_g_scores = current.g_scores + cost;
                    double temp_f_scores = temp_g_scores +
child.h_scores;
                    Log.e("www" , "checkpoint 2-6");

                    /*if child node has been evaluated and
                     the newer f_score is higher, skip*/
                    if ((explored.contains(child))
                            && (temp_f_scores >= child.f_scores)) {
                        continue;
                    } /*else if child node is not in queue or
                     newer f_score is lower*/ else if
((!queue.contains(child))
                            || (temp_f_scores < child.f_scores)) {

                        child.parent = current;
                        child.g_scores = temp_g_scores;
                        child.f_scores = temp_f_scores;

                        if (queue.contains(child)) {
                            queue.remove(child);
                        }

                        queue.add(child);

                    }

                }

            }

        }

        public static int getCost(String s1, String s2) {

            return (int) get_node(s1).get_cost(get_node(s2));
        }


    }

    class Node {

        public final String value;
        public double g_scores;
        public double h_scores;
        public double f_scores = 0;
        public Edge[] adjacencies;
        public Node parent;

        public Node(String val, double hVal) {
            value = val;
            h_scores = hVal;
        }

        public void seth_scores(double g_scores) {
            this.h_scores = g_scores;
        }

        public String toString() {
```

```
            return value;
        }

        public double get_cost(Node n) {
            for (int i = 0; i < adjacencies.length; i++) {
                if (adjacencies[i].target == n) {
                    return adjacencies[i].cost;
                }
            }
            return -1;
        }

        public void add_link(Edge e) {

            int size = 0;
            if (adjacencies == null) {
                size = 1;
            } else {
                size = adjacencies.length + 1;
            }
            Edge[] temp = new Edge[size];
            for (int i = 0; i < size - 1; i++) {
                temp[i] = adjacencies[i];
            }
            temp[size - 1] = e;
            adjacencies = temp;
        }

    }

    class Edge {

        public final double cost;
        public final Node target;

        public Edge(Node targetNode, double costVal) {
            target = targetNode;
            cost = costVal;
        }
    }
```

Example A.3.2 - Class to build graph with the collage model and cost of travels

```
    package com.example.dell.benahapp.navigate.ocr;
    public class collage_model {

        static int[] pos_s = new int[3];

        static int[] pos_t = new int[3];

        public static void get_pos_s(String p) {
            pos_s[2] = Integer.parseInt(p.substring(2).split("-")[0]);
            int num = Integer.parseInt(p.split("-")[1]);
            if (num == 1 || num == 2 || num == 35 || num == 34 || num
    == 33) {
                pos_s[0] = 0;
                pos_s[1] = 0;
            }
            else if(num < 12)
            {
                pos_s[0] = 1;
                pos_s[1] = num - 3;
            }
```

```
            else if(num > 12)
            {
                pos_s[0] = -1;
                pos_s[1] = 21 - num ;
            }
        }

        public static void get_pos_t(String dest) {
            pos_t[2] = Integer.parseInt(dest.substring(2).split("-
")[0]);
            int num = Integer.parseInt(dest.split("-")[1]);
            if (num == 1 || num == 2 || num == 35 || num == 34 || num
== 33) {
                pos_t[0] = 0;
                pos_t[1] = 0;
            }
            else if(num < 12)
            {
                pos_t[0] = 1;
                pos_t[1] = num - 3;
            }
            else if(num > 12)
            {
                pos_t[0] = -1;
                pos_t[1] = 21 - num ;
            }
        }

        public static int get_eclidean() {
            return (int) Math.sqrt(Math.pow(pos_s[0] - pos_t[0], 2) +
Math.pow(pos_s[1] - pos_t[1], 2) + Math.pow(pos_s[2]- pos_t[2],
2)) * 10;
        }

        public static int get_heuristic(String p, String dest) {
            get_pos_s(p);
            get_pos_t(dest);
            return get_eclidean();  } }
```

A.4 – Student planner Activity

Example A.4.1 – Code of the main activity of the calendar

```
public class calenderMain extends AppCompatActivity {

    CompactCalendarView compactCalendar;


    SimpleDateFormat fullDate = new SimpleDateFormat("MMM dd yyyy
HH:mm");
    SimpleDateFormat evTime = new SimpleDateFormat("HH:mm");
    SimpleDateFormat month = new SimpleDateFormat("MMMM - yyyy");
    SimpleDateFormat day = new SimpleDateFormat("MMM dd yyyy");

    public calenderMain() throws IOException {
    }


    void saveData(String fileName, String key, Set<String>
values){
        SharedPreferences sharedPref =
getSharedPreferences(fileName, Context.MODE_PRIVATE);
        SharedPreferences.Editor editor = sharedPref.edit();
        editor.putStringSet(key, values);
        editor.apply();
    }

    Set<String> loadData(String fileName, String key){
        SharedPreferences sharedPref =
getSharedPreferences(fileName, Context.MODE_PRIVATE);
        Set<String> vals = sharedPref.getStringSet(key, new
HashSet<String>());
        return vals;
    }

    void eventMaker(String d, String ev)  {

        Date date = null;

        try {
            date = fullDate.parse(d);
        } catch (ParseException e) {
            e.printStackTrace();
        }

        String time = evTime.format(date);

        try {
            date = day.parse(d);
        } catch (ParseException e) {
            e.printStackTrace();
        }

        long epoch = date.getTime();
        Event event = new Event(Color.RED,epoch,ev);
        compactCalendar.addEvent(event);

        if(loadData("eventList", d)==null){
            Set<String> eventList = new HashSet<>();
            eventList.add(ev);
            saveData("eventList",date.toString(),eventList);
```

```
                Set<String> eventTimes = new HashSet<>();
                eventTimes.add(time);
                saveData("eventTimes",date.toString(),eventTimes);
            }
            else{

                Set<String> eventList = loadData("eventList",
date.toString());
                eventList.add(ev);
                saveData("eventList",date.toString(),eventList);

                Set<String> eventTimes = loadData("eventTimes",
date.toString());
                eventTimes.add(time);
                saveData("eventTimes",date.toString(),eventTimes);
            }

            }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.calender_main);

        SharedPreferences  mPrefs = getPreferences(MODE_PRIVATE);


        final ActionBar actionBar = getSupportActionBar();
        actionBar.setDisplayHomeAsUpEnabled(false);

//        Gson gson = new Gson();
//        String json = mPrefs.getString("MyObject", "");
//
//        if(!json.equals(""))
//            compactCalendar = gson.fromJson(json,
CompactCalendarView.class);
//
//        else
            compactCalendar = (CompactCalendarView)
findViewById(R.id.compactcalendar_view);



actionBar.setTitle(month.format(compactCalendar.getFirstDayOfCurre
ntMonth()));
        compactCalendar.setUseThreeLetterAbbreviation(true);


//        Uncomment for code testing
        eventMaker("May 08 2017 10:30","Quiz");
        eventMaker("May 08 2017 08:45","Lecture");
        eventMaker("May 18 2017 10:15", "Revision" );
        eventMaker("Jun 18 2017 12:30", "Final Exam" );



        compactCalendar.setListener(new
CompactCalendarView.CompactCalendarViewListener() {
            TextView events = (TextView)
findViewById(R.id.events);
            @Override
            public void onDayClick(Date dateClicked) {
```

```
                    Context context = getApplicationContext();

        if(!loadData("eventList",dateClicked.toString()).isEmpty()) {
            events.setText("");

            Set<String> eventList
=loadData("eventList",dateClicked.toString());
            Set<String> eventTimes =
loadData("eventTimes",dateClicked.toString());

            for(int i=0; i<eventList.size(); i++){
                String output = (i+1)+"-"+eventList.toArray()[i]+" at
"+eventTimes.toArray()[i]+"\n\n";
                events.append(output);
            }

        }
                    else{
                        events.setText("NO EVENT PLANNED FOR THIS
DAY");
                        Toast.makeText(context,"No
Event",Toast.LENGTH_SHORT).show();

            }

            }
            @Override
                public void onMonthScroll(Date firstDayOfNewMonth) {
                    events.setText("");

actionBar.setTitle(month.format(firstDayOfNewMonth));
                }
            });

//          SharedPreferences.Editor prefsEditor = mPrefs.edit();
//          gson = new Gson();
//          json = gson.toJson(compactCalendar);
//          prefsEditor.putString("MyObject", json);
//          prefsEditor.commit();


        } }
```

Example A.4.2 - Code of the main activity of the calendar

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.awarrior.calendar.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        >
```

```
<com.github.sundeepk.compactcalendarview.CompactCalendarView
xmlns:app="http://schemas.android.com/apk/res-auto"
            android:id="@+id/compactcalendar_view"
            android:layout_width="fill_parent"
            android:layout_height="250dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            app:compactCalendarBackgroundColor="#ffe95451"
            app:compactCalendarCurrentDayBackgroundColor="#B71C1C"

app:compactCalendarCurrentSelectedDayBackgroundColor="#E57373"
            app:compactCalendarMultiEventIndicatorColor="#fff"
            app:compactCalendarTargetHeight="250dp"
            app:compactCalendarTextColor="#fff"
            app:compactCalendarTextSize="12sp" />

        <TextView
            android:id="@+id/events"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:paddingLeft="25dp"
            android:paddingTop="35dp"
            android:textSize="20dp"
            android:textColor="#ffe95451"
            android:textStyle="bold"
            android:shadowColor="#272728"
            android:shadowDx="5"
            android:shadowDy="2"
            android:shadowRadius="2"
            android:background="@drawable/my_border"
            />

    </LinearLayout>


</RelativeLayout>
```

Example A.4.3 - my_border.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >

    <!-- View background color -->
    <solid
        android:color="#fff" >
    </solid>

    <!-- View border color and width -->
    <stroke
        android:width="10dp"
        android:color="#E57373" >
    </stroke>


</shape>
```

A.5 – courses Activity

Example A.5.1 – start activity of courses

```
public class courseMain extends AppCompatActivity {



    public void courses_data() {

     Button b1;
     Button b2;
     Button b3;
     Button b4;
     Button b5;
     Button b6;
     Button b7;
     Button b8;

                b1 = (Button) findViewById(R.id.button11);
                b1.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Intent intent = new Intent();
                        intent.setAction(Intent.ACTION_VIEW);

intent.addCategory(Intent.CATEGORY_BROWSABLE);

intent.setData(Uri.parse("http://www.feng.bu.edu.eg/feng/index.php
/mathematical-and-physical-engineering-academic-programs-and-
courses"));
                        startActivity(intent);

                    }


                });



        b2 = (Button) findViewById(R.id.button22);
        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent();
                intent.setAction(Intent.ACTION_VIEW);
                intent.addCategory(Intent.CATEGORY_BROWSABLE);

intent.setData(Uri.parse("http://www.feng.bu.edu.eg/feng/index.php
/civil-academic-programs-and-courses"));
                startActivity(intent);

            }


        });
        b3 = (Button) findViewById(R.id.button33);
        b3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent();
```

```
                    intent.setAction(Intent.ACTION_VIEW);
                    intent.addCategory(Intent.CATEGORY_BROWSABLE);

    intent.setData(Uri.parse("http://www.feng.bu.edu.eg/feng/index.php
    /architectural-academic-programs-and-courses"));
                    startActivity(intent);

                }


            });
            b4 = (Button) findViewById(R.id.button44);
            b4.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Intent intent = new Intent();
                    intent.setAction(Intent.ACTION_VIEW);
                    intent.addCategory(Intent.CATEGORY_BROWSABLE);

    intent.setData(Uri.parse("http://www.feng.bu.edu.eg/feng/index.php
    /survey-sciences-academic-programs-and-courses"));
                    startActivity(intent);

                }


            });
            b5 = (Button) findViewById(R.id.button55);
            b5.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Intent intent = new Intent();
                    intent.setAction(Intent.ACTION_VIEW);
                    intent.addCategory(Intent.CATEGORY_BROWSABLE);

    intent.setData(Uri.parse("http://www.feng.bu.edu.eg/feng/index.php
    /electrical-academic-programs-and-courses"));
                    startActivity(intent);

                }


            });
            b6 = (Button) findViewById(R.id.button66);
            b6.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Intent intent = new Intent();
                    intent.setAction(Intent.ACTION_VIEW);
                    intent.addCategory(Intent.CATEGORY_BROWSABLE);

    intent.setData(Uri.parse("http://www.feng.bu.edu.eg/feng/index.php
    /mechanical-academic-programs-and-courses"));
                    startActivity(intent);

                }        });}


        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.course_main);
            courses_data();
    }}
```

A.6 – Portal Activity

Example A.6.1 – start of portal task

```
public class portalMain extends AppCompatActivity {

    private WebView webView = null;

    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if (event.getAction() == KeyEvent.ACTION_DOWN) {
            switch (keyCode) {
                case KeyEvent.KEYCODE_BACK:
                    if (webView.canGoBack()) {
                        webView.goBack();
                    }
                    return true;
            }

        }
        return super.onKeyDown(keyCode, event);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.portal_main);

        final ProgressDialog pd = ProgressDialog.show(this, "",
"Loading...",true);

        webView = (WebView) findViewById(R.id.webView);

        webView.setWebChromeClient(new WebChromeClient() {
            public void onProgressChanged(WebView view, int
progress) {
                if(progress>=40)
                    pd.dismiss();
            }
        });
        webView.setWebViewClient(new WebViewClient(){
            @Override
            public void onReceivedError(WebView view, int
errorCode, String description, String failingUrl) {
                Context context = getApplicationContext();
                Toast.makeText(context,"Oh No!",
Toast.LENGTH_SHORT).show();
            }
        });

        //webView.setInitialScale(0);
        webView.getSettings().setBuiltInZoomControls(true);
        webView.getSettings().setDisplayZoomControls(false);
        webView.getSettings().setJavaScriptEnabled(true);
        webView.getSettings().setLoadWithOverviewMode(true);
        webView.getSettings().setUseWideViewPort(true);
        webView.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
        webView.setScrollbarFadingEnabled(false);
        webView.loadUrl("http://www.bu.edu.eg/");


    }
}
```

Example A.6.2 - Portal XML file

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/backColor"
tools:context="com.example.dell.benahapp.protal.portalMain">

<WebView
    android:id="@+id/webView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:scrollbars="vertical"
    android:background="@color/backColor"
    android:layout_marginRight="-37dp"
    android:padding="0dp"
    />
</RelativeLayout>
```

A.7 – Mission and Vision activity

Example A.7.1 – Code of main Activity that contains the tabs for menu and the anther tab for mission and vision

```
public class MainActivity extends AppCompatActivity {
    private Toolbar toolbar;
   // private TabLayout tabLayout;
    private ViewPager viewPager;
    public static Typeface face ;
    public static AppCompatActivity app;

        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);

          //  face =
Typeface.createFromAsset(getAssets(),"fonts/LGR.otf");
            // Setup Toolbar
            //android:background="?attr/colorPrimary"
            // app:titleTextColor="@color/colorAccent"
        app = this;
 //         toolbar = (Toolbar) findViewById(R.id.toolbar);
//            setSupportActionBar(toolbar);

            ImageView universityimg=
(ImageView)findViewById(R.id.universityimg);

            universityimg.setImageResource(R.drawable.benha);

            viewPager = (ViewPager) findViewById(R.id.pager);
            // Assign created adapter to viewPager
            setupViewPager(viewPager);

            TabLayout tabLayout = (TabLayout)
findViewById(R.id.tab_layout);
            //      This method setup all required method for
TabLayout with Viewpager
            tabLayout.setupWithViewPager(viewPager);

        }

    private void setupViewPager(ViewPager viewPager) {
        ViewPagerAdapter adapter = new
ViewPagerAdapter(getSupportFragmentManager());
        adapter.addFragment(new studentservicefragment(), "Main
page");
        adapter.addFragment(new MainPagefrag(), "Vision");
        viewPager.setAdapter(adapter);
    }

    public static class ViewPagerAdapter extends
FragmentPagerAdapter {

                private final List<Fragment> mFragmentList = new
ArrayList<>();
                private final List<String> mFragmentTitleList =
new ArrayList<>();

        public ViewPagerAdapter(FragmentManager manager) {
```

```
            super(manager);
        }

        @Override
        public Fragment getItem(int position) {
            return mFragmentList.get(position);
        }

        @Override
        public int getCount() {
            return mFragmentList.size();
        }

        public void addFragment(Fragment fragment, String title) {
            mFragmentList.add(fragment);
            mFragmentTitleList.add(title);
        }

        @Override
        public CharSequence getPageTitle(int position) {
            return mFragmentTitleList.get(position);
        }
            }


    public static TextView tv;

    public void showMission(View view){
        Button b = (Button)view;
        String buttonText = b.getText().toString();
        tv = (TextView) findViewById(R.id.missonData);
        tv.setText(buttonText+"\nMission:-
\n"+visionFactory.getMission(buttonText)+"\n\nVision:-
\n"+visionFactory.getVision(buttonText));
    }


}
```

Example A.7.2 – Code of the Mission and vision

```
public class MainPagefrag extends Fragment {
 View v;

    public MainPagefrag() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
       v= inflater.inflate(R.layout.fragment_main_pagefrag,
container, false);


        return v;
    }


}
```

Example A.7.3- Part from fragment_main_pagefrag.xml that contains the all missions and vision

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".Major.MainActivity"
    android:background="@color/backColor">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/facultyname"
        android:textSize="30dp"
        android:text="Mission and vission"
        android:background="@color/backColor2"
        android:layout_alignParentTop="true"
        />

    <ScrollView
        android:layout_width="wrap_content"
        android:layout_height="150dp"
        android:layout_alignParentTop="true">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/missonData"
            android:textSize="15dp"
            android:text=""
            android:background="@color/backColor"
            android:textColor="@color/textColor2"
            android:layout_alignParentTop="true"
            />

    </ScrollView>

    <ScrollView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/scrollView2"
        android:layout_alignParentTop="true">


        <LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:tools="http://schemas.android.com/tools"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            tools:context=".Major.MainActivity">


                <LinearLayout
                    android:orientation="horizontal"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent">
```

```
                    <Button
                        android:layout_width="217dp"
                        android:layout_height="wrap_content"
                        android:text="Faculty of Medicine"
                        android:layout_marginBottom="10dp"
                        android:background="@drawable/draw"
                        android:textColor="@color/textColor"
                        android:onClick="showMission"
                        />

                    <ImageView
                        android:layout_width="50dp"
                        android:layout_height="50dp"
                        android:layout_marginLeft="10dp"
                        android:background="@drawable/medicine"
                        />
                </LinearLayout>


                <LinearLayout
                    android:orientation="horizontal"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent">

                    <Button
                        android:layout_width="219dp"
                        android:layout_height="wrap_content"
                        android:text="Faculty of Veterinary"
                        android:layout_marginBottom="10dp"
                        android:background="@drawable/draw"
                        android:textColor="@color/textColor"
                        android:onClick="showMission"
                        />

                    <ImageView
                        android:layout_width="50dp"
                        android:layout_height="50dp"
                        android:layout_marginLeft="10dp"
                        android:background="@drawable/veter"
                        />

                </LinearLayout>


            </LinearLayout>


        </ScrollView>


    </LinearLayout>
```

A.8 – News gallery

Example A.8.1 – class of contents of each gallery card

```
package com.example.dell.benahapp.gallery;

import java.io.Serializable;

public class Item implements Serializable {

    public int id;

    public String text;

    public String img;

    public int loves;

    public int hates;

}
```

Example A.8.2 – class of adaptor of News gallery

```
public class ItemAdapter extends
RecyclerView.Adapter<ItemAdapter.ItemViewHolder> implements
Serializable{

    private Context context;
    private ArrayList<Item> itemList;

    public ItemAdapter(Context context, ArrayList<Item> itemList){
        this.context = context;
        this.itemList = itemList;
    }

    @Override
    public ItemViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        LayoutInflater inflater =
LayoutInflater.from(parent.getContext());
        View view = inflater.from(parent.getContext())
                .inflate(R.layout.item_cardview_layout, parent,
false);

        ItemViewHolder itemViewHolder = new ItemViewHolder(view);
        return itemViewHolder;
    }

    @Override
    public void onBindViewHolder(ItemViewHolder holder, int
position) {
        final Item item = itemList.get(position);
        Picasso.with(context)
                .load(item.img)
                .placeholder(R.drawable.rogue)
                .error(android.R.drawable.stat_notify_error)
                .into(holder.ivImg);

        holder.tvText.setText(item.text);
```

```
        holder.ivImg.setOnClickListener(new
View.OnClickListener(){

            @Override
            public void onClick(View v) {

                Intent intent = new Intent(context, Detail.class);
                intent.putExtra("item", item);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                context.startActivity(intent);
            }
        });

        holder.loveButton.setOnClickListener(new
View.OnClickListener(){

            @Override
            public void onClick(View v) {
                item.loves++;
            }
        });

        holder.hateButton.setOnClickListener(new
View.OnClickListener(){

            @Override
            public void onClick(View v) {
                item.hates++;
            }
        });

    }

    @Override
    public int getItemCount() {

        if(itemList != null)
            return itemList.size();

        return 0;
    }


    //ViewHolder Class
    public static class ItemViewHolder extends
RecyclerView.ViewHolder implements Serializable{

        public CardView cvItem;
        public ImageView ivImg;
        public TextView tvText;
        public Button loveButton;
        public Button hateButton;

        public ItemViewHolder(View itemView) {
            super(itemView);
            cvItem = (CardView)
itemView.findViewById(R.id.cvItem);
            ivImg = (ImageView) itemView.findViewById(R.id.ivImg);
            tvText = (TextView)
itemView.findViewById(R.id.tvText);
            loveButton = (Button)
itemView.findViewById(R.id.loveB);
```

```
                hateButton = (Button)
itemView.findViewById(R.id.hateB);      }    } }
```

A.9 - Dimensions

Examples A.9.1 – dimen.xml file

```
<resources>
    <!-- Default screen margins, per the Android Design
guidelines. -->
    <dimen name="small_margin">16dp</dimen>
    <dimen name="margin">25dp</dimen>
    <dimen name="extra_margin">60dp</dimen>
    <dimen name="huge_margin">340dp</dimen>
    <dimen name="wow_margin">500dp</dimen>
    <dimen name="text_small">11sp</dimen>
    <dimen name="text_mid">16sp</dimen>
    <dimen name="text_big">25sp</dimen>
</resources>
```

A.10 – style

Example A.10.1 – color.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="backColor">#f1ebc9 </color>
    <color name="backColor2">#c4bb81</color>
    <color name="textColor">#ffffff </color>
    <color name="textColor2">#000000 </color>
    <color name="special">#e78baa00 </color>
</resources>
```

Example A.10.2 – draw.xml file for style of button of Benha App

```
<?xml version="1.0" encoding="utf-8"?>
<selector
xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:state_pressed="false"><shape
android:shape="rectangle">
        <corners android:radius="10dp" />
        <gradient
            android:endColor="#fffffa"
            android:startColor="#ffffff"
            android:angle="270" />
        <solid android:color="@color/special" />
```

```
        <padding android:bottom="10.0dip" android:left="10.0dip"
android:right="10.0dip" android:top="10.0dip" />


    </shape></item>
    <item android:state_pressed="true"><shape
android:shape="rectangle">
        <corners android:radius="10dp" />

        <solid android:color="#ffffff" />
        <gradient
            android:endColor="#fffffa"
            android:startColor="#ffffff"
            android:angle="270" />
        <padding android:bottom="10.0dip" android:left="10.0dip"
android:right="10.0dip" android:top="10.0dip" />
    </shape></item>

</selector>
```

A.11 – Multi-lingual

Example A.11.1 – code to change language

```
if(temp.equals("English"))
                {
                    locale = new Locale("en");
                    Toast.makeText(getActivity(), "Language:
English", Toast.LENGTH_LONG).show();
                  }
                else
                {
                    locale = new Locale("ar");
                    Toast.makeText(getActivity(), " :اللغة
العربية", Toast.LENGTH_LONG).show();
                }

                Locale.setDefault(locale);
                Configuration config = new Configuration();
                config.locale = locale;

getActivity().getBaseContext().getResources().updateConfiguration(
config,
getActivity().getBaseContext().getResources().getDisplayMetrics())
;
                getActivity().finish();
                startActivity(getActivity().getIntent());

            }
```

Example A.11.2 – string.xml file for Arabic language

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">جامعة بنها</string>
    <string name="lang_btn">English</string>
    <string name="quick_earch">بحث سريع</string>
    <string name="go">بحث</string>
    <string name="email">بريد</string>
    <string name="password">كلمة السر</string>
    <string name="sign in">تسجيل دخول</string>
    <string name="cheating_list">قائمة الغش</string>
    <string name="new_session">محاضرة جديدة</string>
    <string name="OCR">OCR</string>
    <string name="translation">Transition</string>
    <string name="academic_years">الأعوام الدراسية</string>
    <string name="prep_year">فرقة إعدادى هندسة</string>
    <string name="civil">هندسة مدنية</string>
    <string name="archictect">هندسة معمارية</string>
    <string name="measure">هندسة مساحة</string>
    <string name="electrical">هندسة كهربية</string>
    <string name="mechanical">هندسة ميكانيكية</string>
    <string name="indutrial">هندسة صناعية</string>
    <string name="energy">هندسة الطاقة</string>
    <string name="emergency">أرقام الطوارئ</string>
    <string name="mission">الرؤية و الرسالة</string>
    <string name="midicine">كلية الطب</string>
    <string name="veterinary">كلية الطب البيطرى</string>
    <string name="nursing">كلية التمريض</string>
    <string name="science">كلية العلوم</string>
    <string name="education">كلية التربية</string>
    <string name="arts">كلية الفنون</string>
    <string name="agriculture">كلية الزراعة</string>
    <string name="physical">كلية التربية الرياضية</string>
    <string name="law">كلية الحقوق</string>
    <string name="shoubra">كلية هندسة شبرا</string>
    <string name="tech">كلية حاسبات و معلومات</string>
    <string name="engineer">هندسة بنها</string>
    <string name="applied">كلية الفنون التطبيقية</string>
    <string name="specific">كلية التربية النوعية</string>
    <string name="important_websites">مواقع هامة</string>
    <string name="contact">اتصل بنا</string>
    <string name="top">اتجه لأعلى الصفحة</string>
    <string name="done">تم</string>
    <string name="indoor_navigation">موجه الطرق</string>
    <string name="deststination">الوجهه</string>
    <string name="current">المكان الحالى</string>
    <string name="scan">صور اللوحة</string>
    <string name="route">حدد طريق</string>
    <string name="like">أعجبنى</string>
    <string name="dislike">لم يعجبنى</string>
    <string name="maps">الخرائط</string>
    <string name="khalfawy">الخلفاوى</string>
    <string name="rod_elfarg">روض الفرج</string>

</resources>
```

A.12 – setting lists and News timeline

Examples A.12.1 – Top of the page button code

```
Button b7 = (Button) v.findViewById(R.id.button7);
        final ScrollView myScrollView =
(ScrollView)v.findViewById(R.id.scrollViewmain) ;

        b7.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                myScrollView.scrollTo(0, 0);
            }
        });
```

Example A.12.2 – new time lines TextView code

```
<TextView
                android:text="START | lunch 20.00 | Dinner 60.00 |
Travel 60.00 | Doctor 5000.00 | lunch 20.00 | Dinner 60.00 |
Travel 60.00 | Doctor 5000.00 | END"
                android:id="@+id/newsTxt"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:singleLine="true"
                android:ellipsize="marquee"
                android:marqueeRepeatLimit="marquee_forever"
                android:scrollHorizontally="true"
                android:paddingLeft="15dip"
                android:paddingRight="15dip"
                android:focusable="true"
                android:focusableInTouchMode="true"
                android:freezesText="true"
                android:layout_below="@+id/gridView"
                android:background="@color/backColor2"
                />
```

A.13 – database and tomcat server

Examples A.13.1 – code to create database tables and insert into it

```
public class DatabaseHelper extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "University.db";

    public static final String TABLE1_NAME = "Faculty_table";
    public static final String COL_facID = "facID";
    public static final String COL_facName = "facNAME";

    public static final String TABLE3_NAME= "Doctor_table";
    public static final String COL_docID = "docID";
    public static final String COL_docNAME = "docNAME";

    public static final String TABLE4_NAME= "Course_table";
    public static final String COL_courseID = "courseID";
    public static final String COL_courseNAME = "courseNAME";

    public static final String TABLE7_NAME= "enroll_table";
    public static final String COL_userID = "userID";
    public static final String COL_username = "userNAME";

    public static final String TABLE8_NAME= "Doctor_table";

    public static final  String TABLE9_NAME= "SignUP_table";

    public static final  String TABLE10_NAME= "actionDr_table";

    public DatabaseHelper(Context context) {

        super(context, DATABASE_NAME, null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        db.execSQL("create table " + TABLE1_NAME +" (facID INTEGER
PRIMARY KEY AUTOINCREMENT,facNAME TEXT)");
        db.execSQL("INSERT INTO  "+TABLE1_NAME +" (facNAME)
VALUES('Engineering,Shoubra')");

        db.execSQL("create table " + TABLE2_NAME +" (depID INTEGER
PRIMARY KEY AUTOINCREMENT ," +
                "depNAME TEXT,facID  INTEGER NOT NULL,FOREIGN KEY
(facID) REFERENCES "+TABLE1_NAME +" (facID))" );
        db.execSQL("INSERT INTO  "+TABLE2_NAME +" (depNAME,facID)
VALUES('Electrical',1)");
        db.execSQL("INSERT INTO  "+TABLE2_NAME +" (depNAME,facID)
VALUES('Civil',1)");
        db.execSQL("INSERT INTO  "+TABLE2_NAME +" (depNAME,facID)
VALUES('Mechanical Engineering',1)");
        db.execSQL("INSERT INTO  "+TABLE2_NAME +" (depNAME,facID)
VALUES('Surveying Engineering',1)");
        db.execSQL("INSERT INTO  "+TABLE2_NAME +" (depNAME,facID)
VALUES('Architectural Engineering',1)");



    }
    @Override
    public void onOpen(SQLiteDatabase db) {
        super.onOpen(db);
        if (!db.isReadOnly()) {
            // Enable foreign key constraints
            db.execSQL("PRAGMA foreign_keys=ON;");
        }
    }
    @Override
```

```
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        db.execSQL("DROP TABLE IF EXISTS "+TABLE1_NAME);
        db.execSQL("DROP TABLE IF EXISTS "+TABLE2_NAME);
        onCreate(db);
    }

    public boolean insertfac(String name) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_facName, name);
        long result = db.insert(TABLE1_NAME, null, contentValues);
        if (result == -1)
            return false;
        else


        {

            return true;
        }
    }
    public Cursor getAllfac() {
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor res = db.rawQuery("select * from "+TABLE1_NAME,null);
        return res;
    }

    public boolean insertdep(String name,String cource) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_DepName, name);
        contentValues.put(COL_facID, cource);
        long result = db.insert(TABLE2_NAME, null, contentValues);
        if (result == -1)
            return false;
        else


        {

            return true;
        }
    }
    public Cursor getAlldep() {
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor res = db.rawQuery("select * from "+TABLE2_NAME,null);
        return res;
    }

    public  Cursor  getselectedjointable2And1(String name) {
        SQLiteDatabase db = this.getReadableDatabase();

       Cursor s = db.rawQuery("SELECT depNAME FROM "+TABLE2_NAME+ "
a"+"  JOIN "+TABLE1_NAME+" b "+" ON a.facID=b.facID WHERE b.facNAME =
?", new String[] {name});


        return s;
    }

    public  Cursor  getselectedstaffbyfac(String name) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor s = db.rawQuery("SELECT docNAME FROM "+TABLE3_NAME+ "
a"+"  JOIN "+TABLE1_NAME+" b "+" ON a.facID=b.facID WHERE b.facNAME =
?", new String[] {name});

        return s;
    }
    public  Cursor  getselectedstaffbydep(String name1,String name2) {
        SQLiteDatabase db = this.getReadableDatabase();
```

```
        Cursor s = db.rawQuery("SELECT docNAME FROM "+TABLE3_NAME+ "
a"+"  JOIN "+TABLE1_NAME+" b "+" ON a.facID=b.facID WHERE b.facNAME = ?
AND a.depNAME= ? ", new String[] {name1,name2});

        return s;
    }

    public  Cursor  getselectedcourse(String name1) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor s = db.rawQuery("SELECT courseNAME  FROM "+TABLE4_NAME+
" a"+"  JOIN "+TABLE1_NAME+" b "+" ON a.facID=b.facID WHERE b.facNAME =
?  ", new String[] {name1});

        return s;
    }
    public  Cursor  getselectedcoursebystaff(String name1,String name2)
{
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor s = db.rawQuery("SELECT  courseNAME  FROM "+TABLE4_NAME+
" a"+"  JOIN "+TABLE1_NAME+" b "+" ON a.facID=b.facID WHERE b.facNAME =
? AND a.docNAME= ? ", new String[] {name1,name2});

        return s;
    }

    public boolean enrollincourse(String name1,String name2) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_username, name1);
        contentValues.put("courseNAME", name2);

        long result = db.insert(TABLE7_NAME, null, contentValues);
        if (result == -1)
            return false;
        else
        {

            return true;
        }
    }

    public Cursor getAllRegistered() {
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor res = db.rawQuery("select username from
"+TABLE7_NAME,null);
        return res;
    }

    public Integer deleteData (String name) {
        SQLiteDatabase db = this.getWritableDatabase();
        return db.delete(TABLE7_NAME, "username = ?",new String[]
{name});
    }
    public boolean signup(String name1,String name2,String name3) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("userNAME", name1);
        contentValues.put("Password", name2);
        contentValues.put("Job", name3);

        long result = db.insert(TABLE9_NAME, null, contentValues);
        if (result == -1)
            return false;
        else


        {

            return true;
        }
    }
```

```
    public  Cursor selectJob(String name1,String name2) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor s = db.rawQuery("SELECT Job  FROM "+TABLE9_NAME+" WHERE
userNAME= ? AND Password= ? ", new String[] {name1,name2});

        return s;
    }

    public boolean actionslide(String name1,String name2,String name3)
{
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("courseNAME", name1);
        contentValues.put("courseNAMEAgain", name2);
        contentValues.put("action", name3);


        long result = db.insert(TABLE10_NAME, null, contentValues);
        if (result == -1)
            return false;
        else


        {

            return true;
        }
    }

    public  Cursor  selectnotification(String name1) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor s = db.rawQuery("SELECT  action,courseNAMEAgain  FROM
"+TABLE10_NAME+ " a"+"  JOIN "+TABLE7_NAME+" b "+" ON
a.courseNAME=b.courseNAME WHERE b.userNAME = ? ", new String[]
{name1});

        return s;
    }
```

Examples A.13.2 – sign up class

```
public class SignUp extends AppCompatActivity {

    EditText username;
    EditText password;

    Button signup;
    DatabaseHelper myDb;
    ArrayList<String> job;
     ProgressBar simpleProgressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.notify_activity_sign_up);

        myDb=new DatabaseHelper(this);


    username=(EditText)findViewById(R.id.sign_up_editText_username);

    password=(EditText)findViewById(R.id.sign_up_editText_pass);
            final Spinner spinnerjob = (Spinner)
```

```
findViewById(R.id.editText_job);
        final ArrayAdapter<String> dataAdapter4;
        job=new ArrayList<String>();
        job.add("job");
        job.add("Student");
        job.add("Doctor");

        simpleProgressBar  = (ProgressBar)
findViewById(R.id.simpleProgressBar);
        dataAdapter4 = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item,job);

dataAdapter4.setDropDownViewResource(android.R.layout.simple_spinn
er_dropdown_item);
        spinnerjob.setAdapter(dataAdapter4);

        signup=(Button)findViewById(R.id.signup);
        signup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {



                String n=username.getText().toString();
                String p=password.getText().toString();
                String j=spinnerjob.getSelectedItem().toString();
                String type="register";


                if(n.length()!=0 || p.length()!=0 ||
j.length()!=0) {


if(!n.contains("@feng.bu.edu")){showMessage("Error","Email is
incorrect,Type your edu mail");

simpleProgressBar.setVisibility(View.VISIBLE);}
                    else {
                        new LoginData().getdata(n, p, j, type);

simpleProgressBar.setVisibility(View.INVISIBLE);
                        Log.e("workers hhh    " , n +"$"+ p +"$"+ j
+"$"+ type);
                        Toast.makeText(SignUp.this, "Done you can
back to logIn", Toast.LENGTH_LONG).show();
                    }
                }
            }
        });

    }

    public void showMessage(String title, String Message) {
        AlertDialog.Builder builder = new
AlertDialog.Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(Message);
        builder.show();
    }
}
```

Examples A.13.3 – sign in / login class

```
public class SignIn extends AppCompatActivity {
```

```
    EditText username;
    EditText password;
    Button savebtn;
    Button login;
    SharedPreferences preferences;
    DatabaseHelper myDb;
    String job2,username2,username3="";
    boolean b=false;
    int count=0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.notify_activity_sign_in);
        final ProgressBar simpleProgressBar = (ProgressBar)
findViewById(R.id.simpleProgressBar);

        myDb=new DatabaseHelper(this);

username=(EditText)findViewById(R.id.sign_in_editText_username);
//email
        password=(EditText)findViewById(R.id.sign_in_editText_pass);
        login=(Button)findViewById(R.id.login);
        //check correct user

        username2=username.getText().toString();
for(int i=0;i<username2.length();i++)
{
    if(username2.charAt(i)!='@')
    username3+=username2.charAt(i);
    else
    {
        break;
    }
}

         login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {



                String n=username.getText().toString();//email
                String p=password.getText().toString();
                String type="login";

                if(n.length()!=0 || p.length()!=0) {

                    simpleProgressBar.setVisibility(View.VISIBLE);


                    new LoginData().getdata(n, p, " ", type);
                    String getjob = new
LoginParser().getparameter("http://192.168.1.101:8080/WebserverUniversi
ty3/Login");
                    Log.e("eeee",getjob);
                    if (getjob.equals("no job")) {

simpleProgressBar.setVisibility(View.INVISIBLE);
                        showMessage("Error", "Email or Password is
incorrect ");
                        return;
                    } else if (getjob.equals("Student")) {
                            Intent intent = new Intent(SignIn.this,
HomeStudent.class);
                            //intent.putExtra("username", username3);
                            startActivity(intent);

simpleProgressBar.setVisibility(View.INVISIBLE);
                        } else if (getjob.equals("Doctor")){
                            Intent intent = new Intent(SignIn.this,
```

```
HomeDoctor.class);
                                // intent.putExtra("username", username3);
                                 startActivity(intent);

simpleProgressBar.setVisibility(View.INVISIBLE);
                            }
                        else{

simpleProgressBar.setVisibility(View.INVISIBLE);
                            showMessage("Error", "Email or Password is
incorrect ");
                            return;
                        }
                        }

                    else {
                        simpleProgressBar.setVisibility(View.INVISIBLE);
                        showMessage("Error", "Required Email and
Password");

                    }

                    }

            });

        }



        public void showMessage(String title, String Message) {
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setCancelable(true);
            builder.setTitle(title);
            builder.setMessage(Message);
            builder.show();
        }
        }
```

Examples A.13.4 – upload Activity

```
public class UploadActivity extends AppCompatActivity {

    String path2;
    TextView t1;
    Button select,cansel,upload;
    public static final int PERMISSIONS_REQUEST_CODE = 0;
    private static final int FILE_SELECT_CODE = 1;
    filechooser f = new filechooser();
    String path,coursename;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.notify_activity_upload);

        Bundle b=getIntent().getExtras();

        if(b !=null)
        {
            coursename = b.getString("coursename");
        }

        select = (Button) findViewById(R.id.b1);
        cansel = (Button) findViewById(R.id.b2);
        upload = (Button) findViewById(R.id.b3);
```

```
        t1 = (TextView) findViewById(R.id.t1);


        // new UploadFile().execute(new String[] {
"http://192.168.1.3:8080/ReceiveFileServlet/RecFileServlet"});


        select.setOnClickListener(new View.OnClickListener() {
                                            @Override
                                            public void onClick(View
v) {
                                                Intent intent = new
Intent(Intent.ACTION_GET_CONTENT);

intent.setType("*/*");

intent.addCategory(Intent.CATEGORY_OPENABLE);

                                                try {

startActivityForResult(

Intent.createChooser(intent, "Select a File to Upload"),

FILE_SELECT_CODE);

setResult(notificationMain.RESULT_OK, intent);
                                                } catch
(ActivityNotFoundException ex) {
                                                    // Potentially
direct the user to the Market with a Dialog

Toast.makeText(UploadActivity.this, "Please install a File
Manager.",

Toast.LENGTH_SHORT).show();
                                                }

                                            }

                                        }

        );

        upload.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                if (t1.getText().toString().length() != 0) {
                    new UploadFile().execute(new String[] {
t1.getText().toString(),"http://192.168.1.101:8080/WebserverUniver
sity3/RecFileServlet",coursename});


                }
            }        });


        cansel.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent i=new
Intent(UploadActivity.this,HomeDoctor.class);
                startActivity(i);

            }
        });
            }
```

```
    @Override
    protected void onActivityResult(int requestCode, int
resultCode, Intent data) {

        switch (requestCode) {
            case 1:
                if (resultCode == RESULT_OK) {
                    // Get the Uri of the selected file
                    Uri uri = data.getData();
                    File myFile = new File(uri.toString());
                    path = myFile.getAbsolutePath();
                    String s = myFile.getName();

                    try {
                        path2 = filechooser.getPath(this,uri);
                    } catch (URISyntaxException e) {
                        e.printStackTrace();
                    }
                    t1.setText(path2);


                }
                break;
        }
        super.onActivityResult(requestCode, resultCode, data);
    }


    }
```

Examples A.13.5 – upload file class

```
public class UploadFile extends AsyncTask<String, Void, String> {

    static final int BUFFER_SIZE = 4096;
    @Override
    protected String doInBackground(String... urls) {
        HttpURLConnection httpConn=null;

        String file=urls[0];
        String pram1=urls[1];
        String coursename=urls[2];

        try
        {

            File uploadFile = new File(file);
            FileInputStream inputStream = new
FileInputStream(uploadFile);

            System.out.println("File to upload: " + file);

            // creates a HTTP connection
            URL url1 = new URL(pram1);
            httpConn = (HttpURLConnection) url1.openConnection();
            httpConn.setUseCaches(false);
            httpConn.setDoOutput(true);
            httpConn.setRequestMethod("POST");
            httpConn.setRequestProperty("fileName",
uploadFile.getName());
            httpConn.setRequestProperty("coursename", coursename);

            httpConn.connect();
```

```
                // sets file name as a HTTP header
                Log.i("fileName", uploadFile.getName());

                // opens output stream of the HTTP connection for
        writing data

                OutputStream outputStream =
        httpConn.getOutputStream();

                // Opens input stream of the file for reading data


                byte[] buffer = new byte[BUFFER_SIZE];
                int bytesRead = -1;

                System.out.println("Start writing data...");

                while ((bytesRead = inputStream.read(buffer)) != -1) {
                    outputStream.write(buffer, 0, bytesRead);
                }

                System.out.println("Data was written.");
                outputStream.close();
                inputStream.close();
            }
            catch(SocketTimeoutException e)
            {
                Log.e("Debug", "error: " + e.getMessage(), e);
            }
            catch (MalformedURLException ex)
            {
                Log.e("Debug", "error: " + ex.getMessage(), ex);
            }
            catch (IOException ioe)
            {
                Log.e("Debug", "error: " + ioe.getMessage(), ioe);
            }

            try
            {

                // always check HTTP response code from server
                int responseCode = httpConn.getResponseCode();
                if (responseCode == HttpURLConnection.HTTP_OK) {
                    // reads server's response
                    BufferedReader reader = new BufferedReader(new
        InputStreamReader(httpConn.getInputStream()));
                    String response = reader.readLine();
                    System.out.println("Server's response: " +
        response);
                } else {
                    System.out.println("Server returned non-OK code: "
        + responseCode);
                }
            }
            catch (IOException ioex){
                Log.e("Debug", "error: " + ioex.getMessage(), ioex);
            }
            return null;
        }
    }
```

Examples A.13.6 – server code on eclipse

```
@WebServlet("/Register")
public class Register extends HttpServlet {
        private static final long serialVersionUID = 1L;
```

```
        protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {

                response.setContentType("application/json");
                String n=request.getParameter("name");
         String p=request.getParameter("pass");
         String j=request.getParameter("job");
         System.out.println(n);
         System.out.println(p);
         System.out.println(j);

         String driver="com.mysql.jdbc.Driver";
         String url="jdbc:mysql://localhost:3306/universitydb";
         String username="root";
         String password="123";
                String sql= "insert into signup_login
(Email,password,job) values(?,?,?)";

                try{
                     Class.forName(driver);
                     Connection
con=DriverManager.getConnection(url,username,password);
                     PreparedStatement ps=con.prepareStatement(sql);
                     ps.setString(1,n);
                      ps.setString(2,p);
                      ps.setString(3,j);
                      int a=ps.executeUpdate();

                      if(a==1)
                      {
                          System.out.println("data inserted");
                      }
                       con.close();



                }catch (ClassNotFoundException e){
                e.printStackTrace();}
                       catch(SQLException e){e.printStackTrace();}


}
```

Examples A.13.7 – Login data

```
        public class LoginData {
            String s;
            LoginParser Logparser;
            String
        urlReg="http://192.168.1.101:8080/WebserverUniversity3/Register";
            String
        urlLog="http://192.168.1.101:8080/WebserverUniversity3/Login";
            String
        urlNotif="http://192.168.1.101:8080/WebserverUniversity3/Notification";
            String
        fileurl="http://192.168.1.101:8080/WebserverUniversity3/dowonloadFile";

            public void getdata(String nm,String p ,String j,String type)
            {
                List<NameValuePair> list=new ArrayList<NameValuePair>();
```

```
            list.add(new BasicNameValuePair("name",nm));
            list.add(new BasicNameValuePair("pass",p));

            try {
                if(type.equals("login"))
                    Logparser.LoginParser(list,urlLog);
                else {

                    list.add(new BasicNameValuePair("job",j));
                    Logparser.LoginParser(list, urlReg);
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

        public void setdownloadfile(String n)
        {
            List<NameValuePair> list=new ArrayList<NameValuePair>();
            list.add(new BasicNameValuePair("fileurl",fileurl));

            try {

                    Logparser.LoginParser(list,fileurl);

            } catch (IOException e) {
                e.printStackTrace();
            }
        }


        public void setNotif(String coursename)
        {
            List<NameValuePair> list=new ArrayList<NameValuePair>();
            list.add(new BasicNameValuePair("coursename",coursename));
            //list.add(new BasicNameValuePair("res",s));
            try {

                Logparser.LoginParser(list,urlNotif);

            } catch (IOException e) {
                e.printStackTrace();
            }
        }


    }
```

Examples A.13.8 – Login parser

```
    public class LoginParser {

        String line = "";
        String s = "";

        public static void LoginParser(List<NameValuePair> lst, String url)
    throws IOException {
            try {

                DefaultHttpClient httpClient = new DefaultHttpClient();
                HttpPost httpPost = new HttpPost(url);
                httpPost.setEntity(new UrlEncodedFormEntity(lst));
                HttpResponse httpResponse = httpClient.execute(httpPost);
                HttpEntity httpEntity = httpResponse.getEntity();
                Log.e("x-ray",httpResponse.getStatusLine().toString());
            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            } catch (ClientProtocolException e) {
                e.printStackTrace();
```

```
            } catch (IOException e) {
                e.printStackTrace();
            }


        }

    //getparm---> return if user is student or doctor
        public String getparameter(String url) {
            HttpClient client = new DefaultHttpClient();

                HttpGet request = new HttpGet(url);
                HttpResponse response = null;
                try {
                    response = client.execute(request);
                } catch (IOException e) {
                    e.printStackTrace();
                }
    //Here i try to read the response
                BufferedReader rd = null;
                try {
                    String line;
                    rd = new BufferedReader(new
    InputStreamReader(response.getEntity().getContent()));

                    while ((line = rd.readLine()) != null) {
                        s += line;
                    }

                } catch (IOException e) {
                    e.printStackTrace();
                }
                return s;
            }

    }
```

References

Programmer for Android developer website

1. *Introduction to Android:* http://developer.android.com/guide/index.html.
2. *Android API:* http://developer.android.com/reference/packages.html
3. *Java 6 API:* http://docs.oracle.com/javase/6/docs/api/
4. *Android Fundamentals:* http://developer.android.com/guide/components/fundamentals.html
5. *The Java Tutorials:* http://docs.oracle.com/javase/tutorial/
6. *Android User Interfaces:* http://developer.android.com/guide/topics/ui/index.html
7. *Layout:* http://developer.android.com/guide/topics/ui/declaring-layout.html

Books

1. *Wei-Meng Lee* (2015). *BEGINNING ANDROID™4 APPLICATION DEVELOPMENT*
2. *Reto Meier (2012) Professional Android 4 Application Development*
3. *Mark L. Murphy (2010) The Busy Coder's Guide to Android Development*

Other References

1. Stack overflow https://stackoverflow.com/
2. Udacity course material for Android MAL
3. Some repositories on git-hub https://github.com/