

## Criterion B: Design

### Basic Structure of the Android application

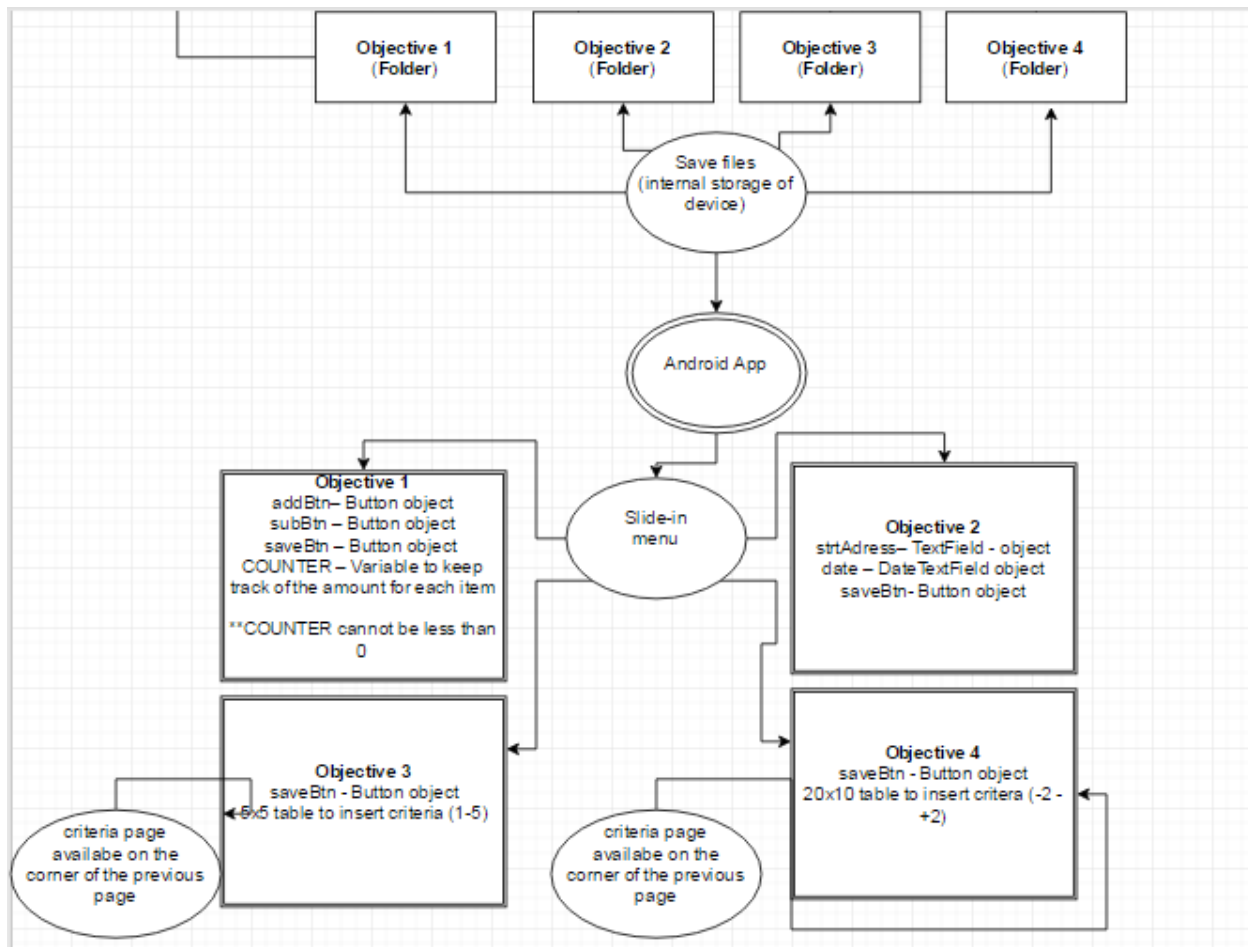


Figure 1 : Flowchart of

## The application

The programming language is of course java, but there is also a visual aspect to it. XML is used to place objects such as buttons, text fields or tables relative to each other and the device (type of device has to be specified before starting the application).

This is what the hierarchy looks like for this application

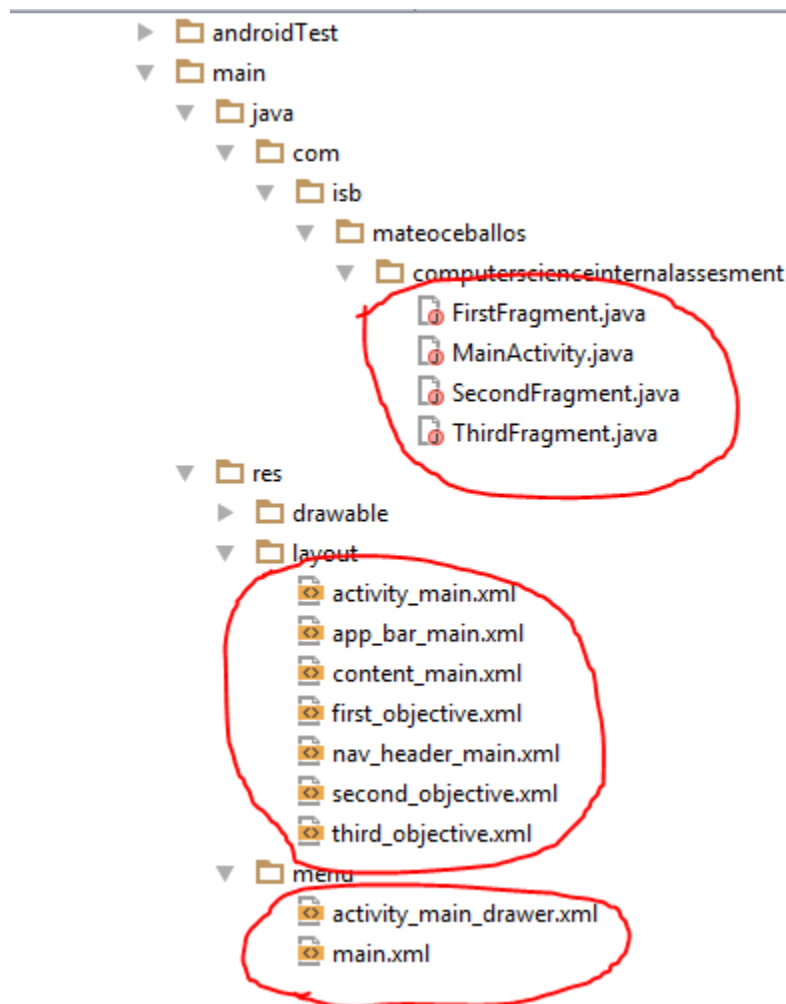


Figure 2 : Xxxk hjlkgjlkdgdlf

Each java class is associated to one XML file, known as the layout, because it shows the layout for that class. Then there are a few other layout files which are combined together to create the slide-in menu. The MainActivity.java is the main class of the entire application, in that class; the menu is created and is told what layout file it should display based on which option is pressed in the slide-in menu.

**Interface (Menu)**

The menu will consist of the following:

Objective 1	Objective 2	Objective 3	Objective 4
-------------	-------------	-------------	-------------

**OBJECTIVE 1**

NAME	TYPE	DESCRIPTION
addBtn	Button object	Increase the value of COUNTER by 1, COUNTER cannot be less than 0
subBtn	Button object	decrease the value of COUNTER by 1, COUNTER cannot be less than 0
saveBtn	Button object	Save values of COUNTER to internal storage as txt file
COUNTER1	Integer	Keep track of number of pedestrians count
COUNTER2	Integer	Keep track of number of traffic count

**OBJECTIVE 2**

NAME	TYPE	DESCRIPTION
strtAdress	TextField object	The address of the street at which the student is at
saveBtn	Button object	Saves data as 2d array into text file located in the internal storage
DATE	DateTextField object	Write down the date of recording

**OBJECTIVE 3**

NAME	TYPE	DESCRIPTION
saveBtn	Button object	Saves data as 3d array into text file located in the internal storage // the type of building, the value (1-5) and which out of 5 of that building type
TABLE	TableView Object	6x5 length

**OBJECTIVE 4**

NAME	TYPE	DESCRIPTION
saveBtn	Button object	Saves data as 2d array into text file located in the internal storage // the type of building, the value (1-5) and which out of 5 of that building type // creates file if there isn't one already
TABLE	TableView Object	20x10 length, each element is an integer from -2 to +2

**Internal Storage (invisible to the user)**

When I first came across having to decide among different types of storage, I immediately thought of using the internal storage of the device rather than SQL for example because of the amount of data overall.

When an application is installed on a device, a folder is automatically created in the storage containing everything about that application. That would be the ideal place to store text files with data.

MAIN APPLICATION FOLDER			
Objective 1 FOLDER	Objective 2 FOLDER	Objective 3 FOLDER	Objective 4 FOLDER

The files would be organized with a date and time like

OBJ[X]\_YYYYMMDD\_HHMMSS

e.g.

OBJ1\_20170219\_134257

This is to keep track of when data was recorded