

SMART ALARM

**A Mini-Project Report
Under
Project Workshop**

Submitted by

**ANMOL KHANDEPARKAR
DEBKANYA MAZUMDER
AKANKSHA MANUJ**

Under The Guidance Of
Ms. Prachi Natu

*in partial fulfillment for the award of the degree
of*

**B.TECH
IN
COMPUTER SCIENCE**

**at
MUKESH PATEL SCHOOL OF TECHNOLOGY
MANAGEMENT AND ENGINEERING,
NMIMS UNIVERSITY
APRIL 2014**

CERTIFICATE

This is to certify that the project entitled “Smart Alarm” is the bonafide work carried out by Anmol Khandeparkar(B046),Debkanya Mazumder(B052),Akanksha Manuj(B060) of B.Tech (Computer Engineering), MPSTME (NMIMS), Mumbai, during the VI semester of the academic year 2014, in partial fulfillment of the requirements for the award of the Degree of Bachelors of Technology as per the norms prescribed by NMIMS. The mini-project work has been assessed and found to be satisfactory.

Prachi Natu

Internal Mentor

Examiner 1

Examiner 2

Dean

Dr. S. Y. Mhaiskar

DECLARATION

We, Anmol Khandeparkar, Debkanya Mazumder, Akanksha Manuj, Roll Nos. B046 , B052 ,B060 respectively , B.Tech (Computer Engineering), VI semester understand that plagiarism is defined as anyone or combination of the following:

1. Un-credited verbatim copying of individual sentences, paragraphs or illustration (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.
2. Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did wrote what. (Source: IEEE, The institute, Dec. 2004)
4. I have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.
5. I affirm that no portion of my work can be considered as plagiarism and I take full responsibility if such a complaint occurs. I understand fully well that the guide of the seminar/ project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature of the Student:

Name: Anmol Khandeparkar Debkanya Mazumder Akanksha Manuj

Roll No. B046 B052 B060

Place:

Date:

ACKNOWLEDGEMENT:

We take this opportunity to express our gratitude to our guide Mrs. Prachi Natu for her excellent guidance throughout the course of this project. Also we would like to thank our colleagues for their constant encouragement throughout the project. We would also like to thank Miss Supriya Agarwal for her constant guidance and suggestions for improvements in our project throughout the semester.

ABSTRACT:

The hardest thing to do is get up early for school, college or work every day. Our application 'Smart Alarm' serves the same purpose as other alarms to wake up but by doing it in a different unique and creative way. A person can simply press the snooze button in a regular alarm and go back to dreaming! The purpose of our android application is to make going back to sleep a little more challenging. Our application is unique in a way that a person has to answer a simple mathematical question for switching off the alarm instead of pressing the snooze button. It also supports tones which are already present in the user's device.

The methodology that we used was the waterfall model methodology. We first prepared a general idea to go about our project and the various requirements for our project and if the project could be completed in the stipulated time. We analysed all the requirements of our project and moved onto the design of our project. During the designing phase, we designed a simple Graphical User Interface for our project on paper and then started working on that Graphical User Interface trying to improvise on it. After an improvised Graphical User Interface was made, we started implementing the Graphical User Interface in the coding phase. We coded all the modules of the project and after completion of the coding of the project, we tested the project and the test was successful.

CHAPTER NO.	TITLE	PAGE NO.
	List of Figures	i
	List of Tables	li
	Abbreviations	lii
	Abstract	lv
1.	INTRODUCTION	9
1.1	Project Overview	9
1.2	Hardware Specification	9
1.3	Software Specification	9
2.	INSTALLATION & SETUP	10
2.1	Installing ADT Bundle	10
2.2	Setup	13
3.	ANALYSIS & DESIGN	15
3.1	External Interface Requirements	15
3.1.1	User Interfaces	15
3.1.2	Software Interfaces	15
3.2	Functional Requirements	15
3.3	Analysis Models	15
3.3.1	Use Case Diagram	16
3.3.2	Data Flow Diagram	17
3.3.3	State Diagram	17
3.4	Non Functional Requirements	18
3.4.1	Performance	18
3.4.2	Reliability	18
3.4.3	Availability	18
3.4.4	Security	18
3.4.5	Maintainability	18
3.4.6	Portability	18
3.5	Design Constraints	18
3.5.1	Time	18
3.5.2	Knowledge of Android	18
3.6	Overall Description Diagram of the Project	19
4.	PROJECT IN DETAIL	20
4.1	HomeScreen	20
4.1.1	Working of the application	20
5.	CONCLUSION & FUTURE SCOPE	28
5.1	Conclusion	28
5.2	Future Scope	28
	REFERENCES	29

List of Figures		
CHAPTER NO .	TITLE	PAGE NO.
1.	INTRODUCTION	-
2.	INSTALLATION & SETUP	
Fig 2.1.1	Downloading Android SDK	10
Fig2.1.2	Run the SDK Setup File	11
Fig 2.1.3	Making a New Project	12
Fig 2.1.4	Launching Eclipse	13
Fig 2.2.1	Running the Application	14
3.	ANALYSIS AND DESIGN	
	3.3.1 Use Case Diagram	16
	3.3.2 Dataflow Diagram	17
	3.3.3 State Chart Diagram	17
	3.6.1 Overall Description	18
4	PROJECT DETAILS	
	4.1.1 Home Screen	20
	4.1.2 Setting the time	21
	4.1.3 Setting the day of Repeat	22
	4.1.4 Tone Settings	23
	4.1.5 Setting the difficulty	24
	4.1.6 Alarm about to ring	25
	4.1.7 Maths problem displayed	26
	4.1.8 Incorrect answer	27
	4.1.9 Report Bug	27

Abbreviations

Abbreviation	Description
SDK	Software Development Kit
IDE	Integrated Development Environment
DFD	Data Flow Diagram
AVD	AndroidVirtual Device

1. INTRODUCTION

1.1 Project Overview

The application that we have developed is an alarm application called the Smart Alarm. The alarm fulfills the various functions of a normal alarm like creating an alarm for a particular time or deleting an alarm which was previously set. The alarm can be set to repeat on particular days or all the days of the week and the mode of the alarm can also be set to vibrate or not to vibrate the device using the alarm application. But the unique feature that distinguishes our application from other alarm applications is the addition of the mathematical problem. Whenever the user wishes to stop the alarm from ringing, instead of pressing the snooze button the user has to solve the mathematical problem displayed on the screen. The level of difficulty of the problem can be set by the user prior to setting the alarm. Once the user has entered the right answer, the alarm stops ringing fulfilling its purpose of waking up the user.

Our application is a basic application which runs from Android 2.3 to Android 4.4. It does not require a lot of memory space and can run on a very basic android device.

1.2 Hardware Specifications

Recommended 4 GB memory for Eclipse Integrated development Environment and its various plugins.

1.3 Software Specifications

Operating Systems

- Windows Vista (32- or 64-bit), Windows 7 (32- or 64-bit)
- Mac OS X 10.5.8 or later (x86 only)
- Linux (Ubuntu Linux)

Eclipse IDE

- Eclipse 4.0 or higher
- Eclipse JDT plugin
- JDK 6 and JRE
- Android Development Tools plugin

2. INSTALLATION AND SETUP

2.1 Installing ADT Bundle (Android SDK + Eclipse IDE)

Steps to install Android SDK and other important components:

Log on to <http://developer.android.com/sdk/installing/bundle.html> and download the SDK (ADT bundle).

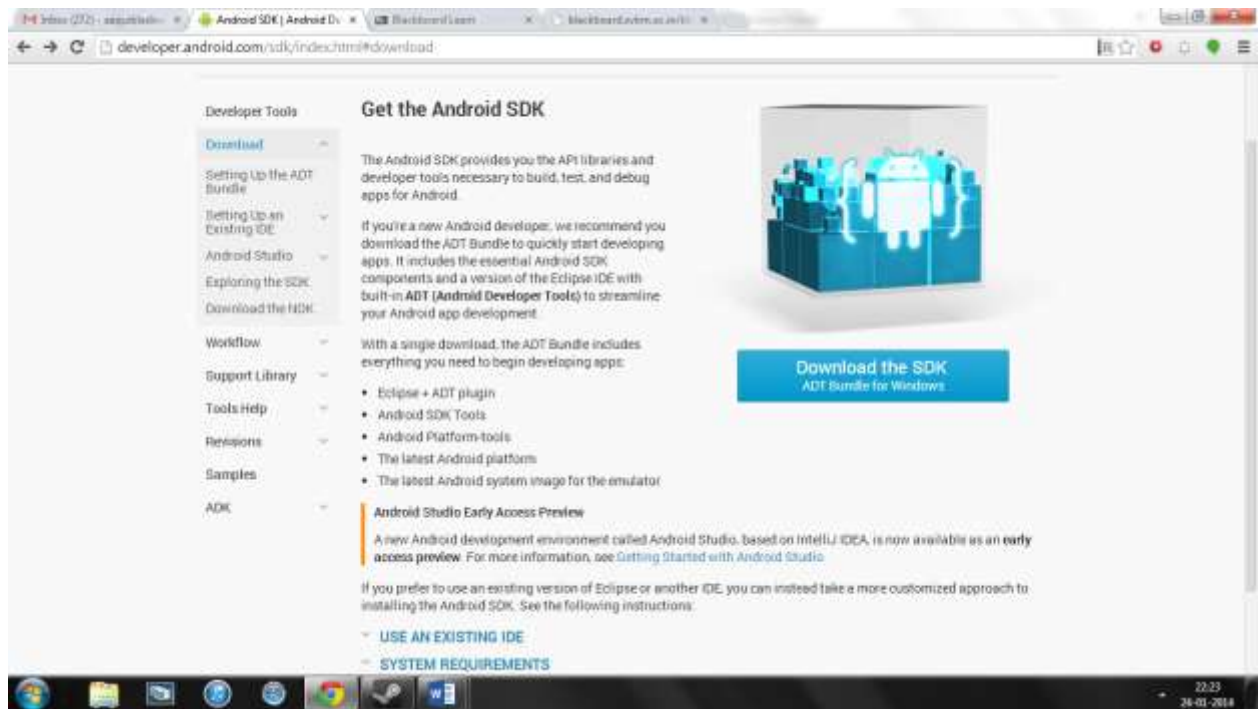


Fig 2.1.1 Downloading Android SDK

After the download has been completed install the SDK and Eclipse IDE as follows:

1. Unpack the ZIP file (named `adt-bundle-<os_platform>.zip`) and save it to an appropriate location, such as a "Development" directory in your home directory.
2. Open the `adt-bundle-<os_platform>/eclipse/` directory and launch **eclipse**.

After the SDK and Eclipse IDE has been installed the next step is to setup the IDE. This can be done by installing the eclipse plug-in and additional platforms and packages.

You should have already downloaded the Android SDK Tools. The SDK Tools package is not the complete SDK environment. It includes only the core SDK tools, which you can use to download the rest of the SDK packages.

Your download package is an executable file that starts an installer. The installer checks your machine for required tools, such as the proper Java SE Development Kit (JDK) and installs it if necessary. The installer then saves the Android SDK Tools into a default location (or you can specify the location).

1. Double-click the executable (.exe file) to start the install.

2. Make a note of the name and location in which it saves the SDK on your system—you will need to refer to the SDK directory later, when setting up the ADT plugin and when using the SDK tools from the command line.
3. Once the installation completes, the installer offers to start the Android SDK Manager. If you'll be using Eclipse, **do not** start the Android SDK Manager, and instead move on to installing the Eclipse Plugin.

The installation of eclipse plug-in can be done as follows:

Android offers a custom plugin for the Eclipse IDE, called Android Development Tools (ADT). This plugin provides a powerful, integrated environment in which to develop Android apps. It extends the capabilities of Eclipse to let you quickly set up new Android projects, build an app UI, debug your app, and export signed (or unsigned) app packages (APKs) for distribution.

Run the SDK Setup File

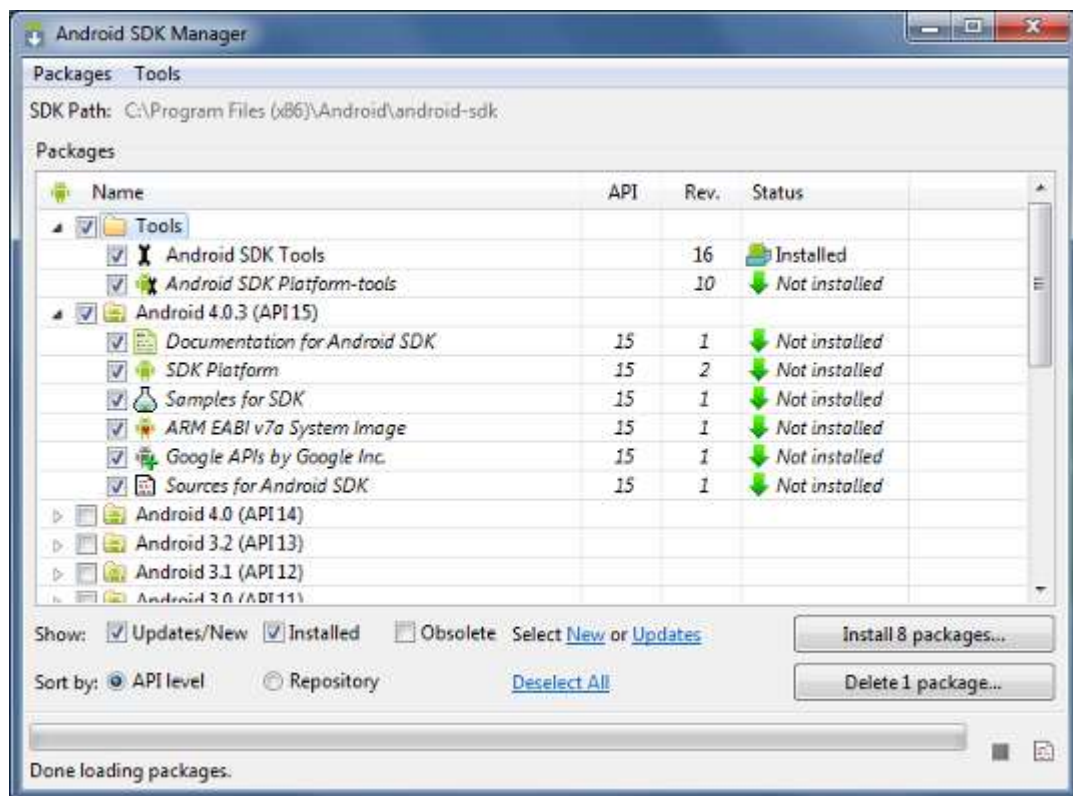


Fig 2.1.2 Run the SDK Setup File

Test your installation:

1. Download HelloWorld project and extract the contents to a folder.
2. Create an Android Virtual Device (AVD)
 - In Eclipse, you can access the “**Android Virtual Device (AVD)**” in Eclipse toolbar. Click “new” to create an AVD.

- Choose a AVD Name and set a Target listed in the drop down and click on Create AVD.

Later, Eclipse will deploy the application into this AVD.

1. Choose the AVD device and click on Start button to launch it. If the emulator launches then your installation is successful.
2. Next, open Eclipse. Click on **File -> Import -> General -> Existing Projects into Workspace** -> Browse to the HelloWorld folder. Then click Finish.
3. Check if the project successfully imports into your workspace without any errors.

STEPS TO CREATE THE HELLO WORLD APPLICATION.

Create Android Application

The first step is to create a simple Android Application using Eclipse IDE. Follow the option **File -> New -> Project** and finally select **Android New Application** wizard from the wizard list. Now name your application as **HelloWorld** using the wizard window as follows:

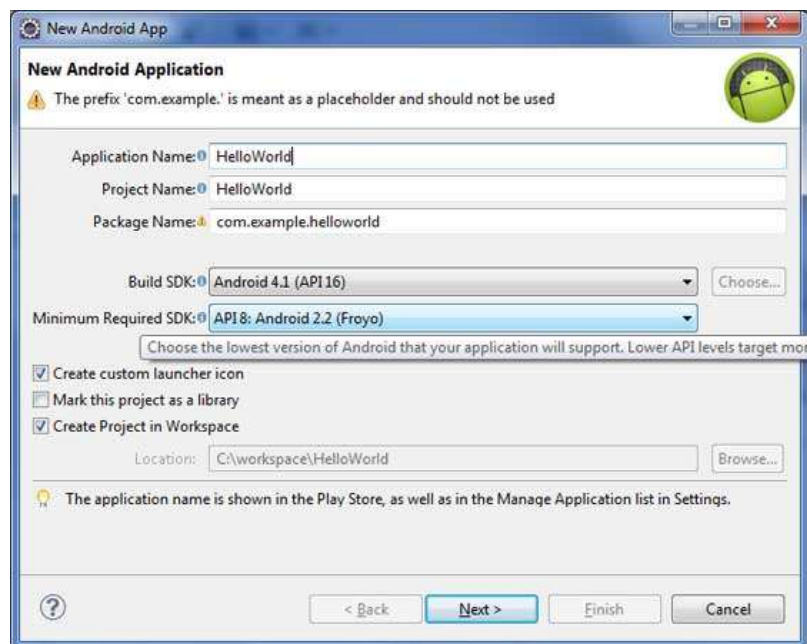


Fig 2.1.3 Making A New Project

Next, follow the instructions provided and keep all other entries as default till the final step. Once your project is created successfully, you will have following project screen:

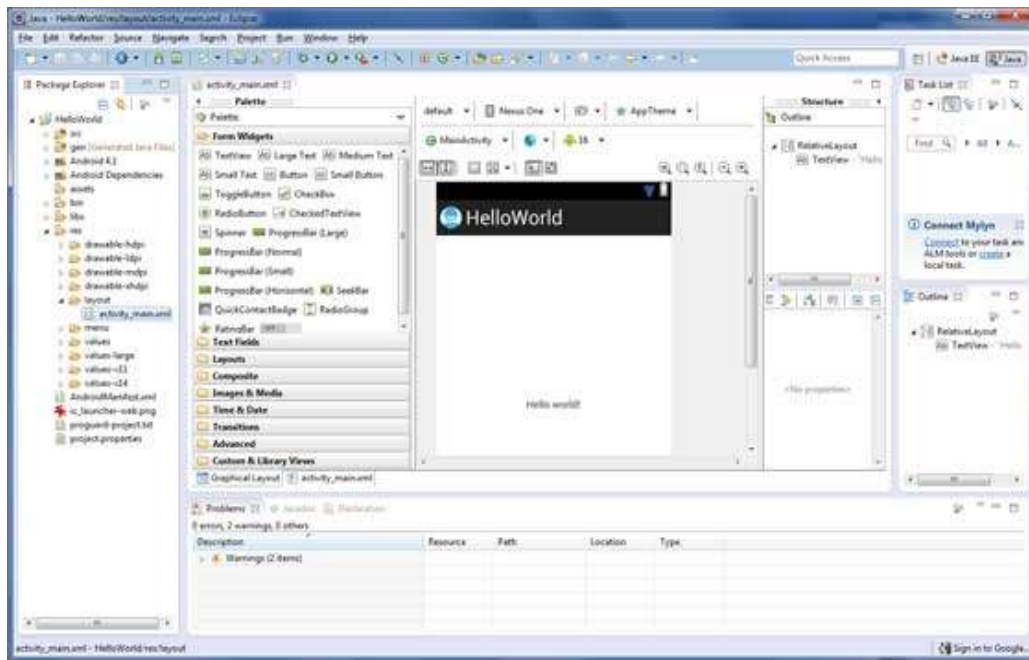



Fig 2.1.4 Launching Eclipse

2.2 Set-up (Running Test App)

Running the Application:

Let's try to run our **Hello World!** application we just created. I assume you had created your **AVD** while doing environment setup. To run the app from Eclipse, open one of your project's activity files and click Run  icon from the toolbar. Eclipse installs the app on your AVD and starts it and if everything is fine with your setup and application, it will display following Emulator window:

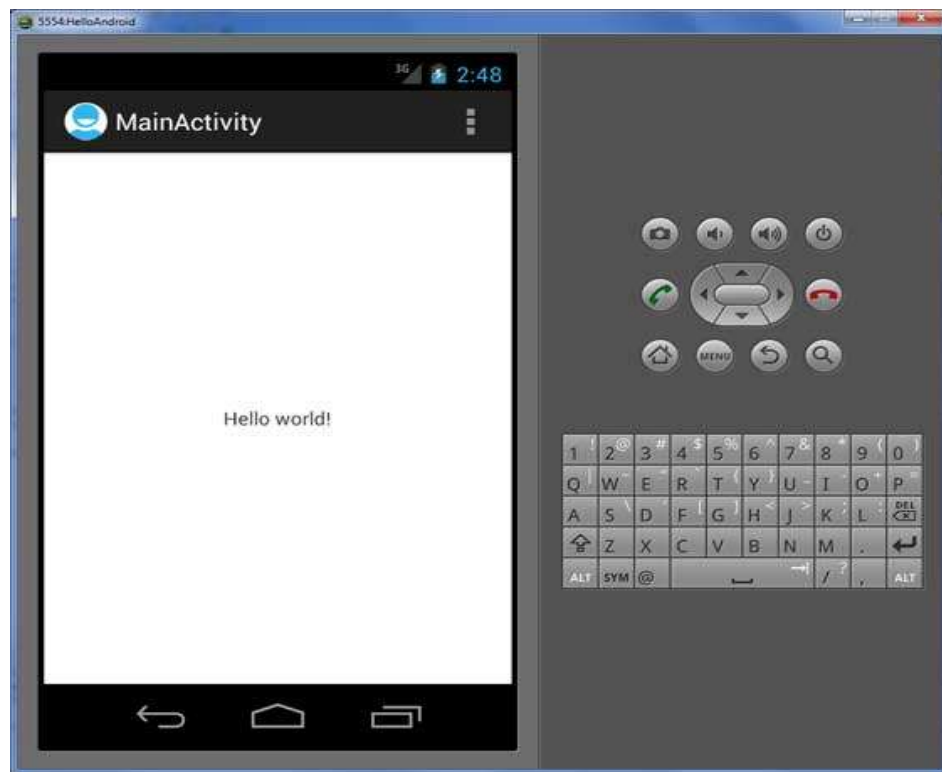


Fig 2.2.1 Running The Application

3. ANALYSIS AND DESIGN

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 User Interfaces

The user will be an active entity participating in the usage of alarm regularly and using the application to his advantage in waking up efficiently. The system is programmed to make the user interface friendly.

3.1.2 Software Interfaces

The system is programmed to be user friendly and as easy as possible for users so that people who haven't used android can use it without any pain and efforts required.

3.2 FUNCTIONAL REQUIREMENTS

1. The user's device should be equipped with Android 4.4 Kitkat to 2.3 Gingerbread.
2. Though it won't affect the users battery but your device should have a good battery life so when the alarm is set, your device is in a condition to run and let the alarm do its job.
3. The audio of the devices should be good to play the loud tones clearly for special cases.

3.3 ANALYSIS MODELS

3.3.1 Use Case

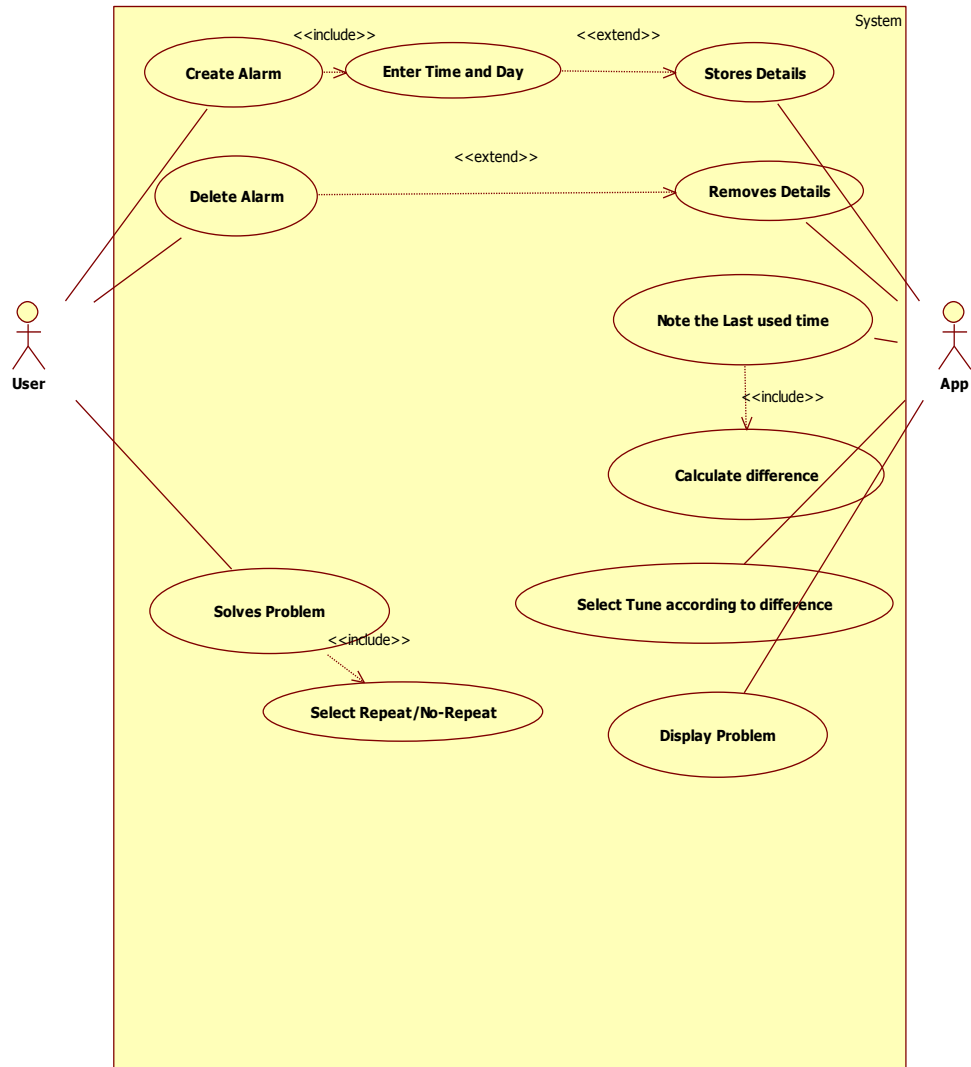


Fig 3.3.1 : Use Case Diagram

3.3.2 Data Flow Diagram (DFD)

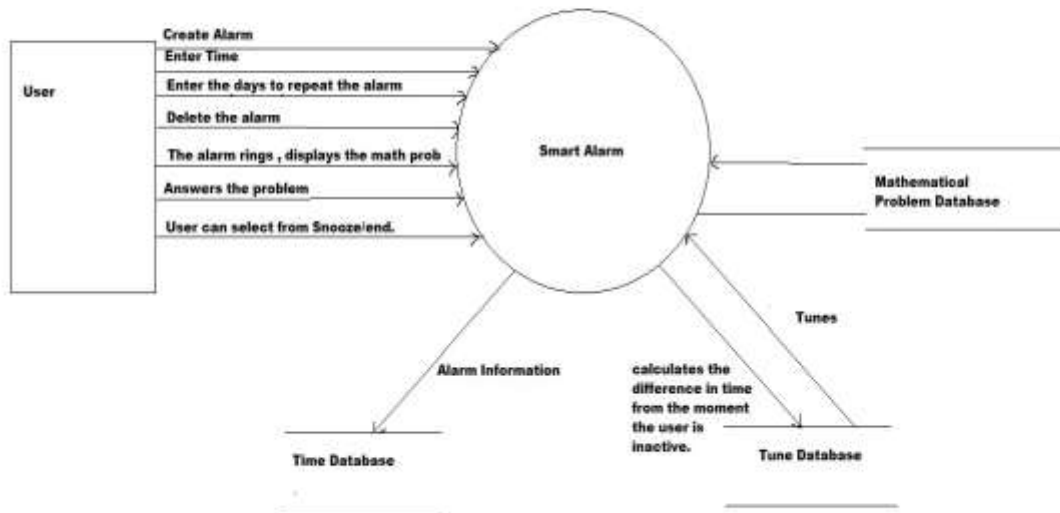


Fig 3.3.2 Data Flow Diagram

3.3.3 State Diagram

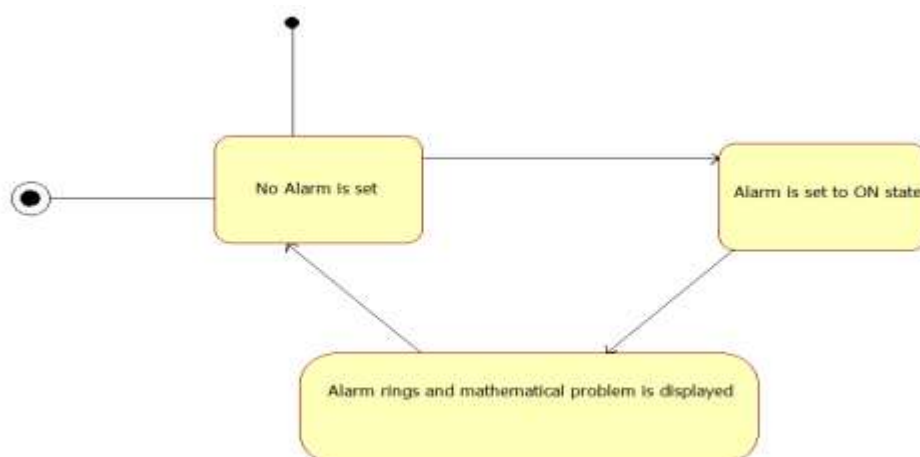


Fig 3.3.3 State Diagram

3.4 NON-FUNCTIONAL REQUIREMENTS

3.4.1 Performance

The alarms performance will be consistent. It will never malfunction and wake the user up at the time for which the alarm was set. The alarm will set the tune from the tunes present in the user's android device. The user will have to solve a mathematical problem in order to stop the alarm from ringing instead of just pressing the snooze button which further enhances and distinguishes the performance of the application from other alarm applications.

3.4.2 Reliability

The application is very reliable as mentioned above. It will give the user the option to solve a math problem so that if the user is a persistent sleeper, the user solves the math problem, and it will ensure that he is awake.

3.4.3 Availability

The application will be widely available through google play store and there will be no restrictions in its usage as it can be used by anyone belonging to any country and speaking any language as long as the alarm is set to his country time.

3.4.4 Security

No personal data of the user will be stored. No login is required so security provided is 100%.

3.4.5 Maintainability

There will be updates done frequently to try to surpass the previous version but there won't be much maintainability required and if any problems faced by the user. The user can write his problem and it will be sent to the developer immediately through the inbuilt feature of help.

3.4.6 Portability

The application can be embedded in any android device with specifications mentioned above and can even replace the previous alarm stored in the device.

3.5 DESIGN CONSTRAINTS

3.5.1 Time

There are a lot of features which are required to be added. Time is a major constraint as even user manuals, documentations will be needed to be made.

3.5.2 Knowledge of android

As it's an android application, knowledge of android needs to be more than basic to implement functions of alarm and to make it a successful application so times needed again to polish android programming skills.

3.6 OVERALL DESCRIPTION OF THE APPLICATION:

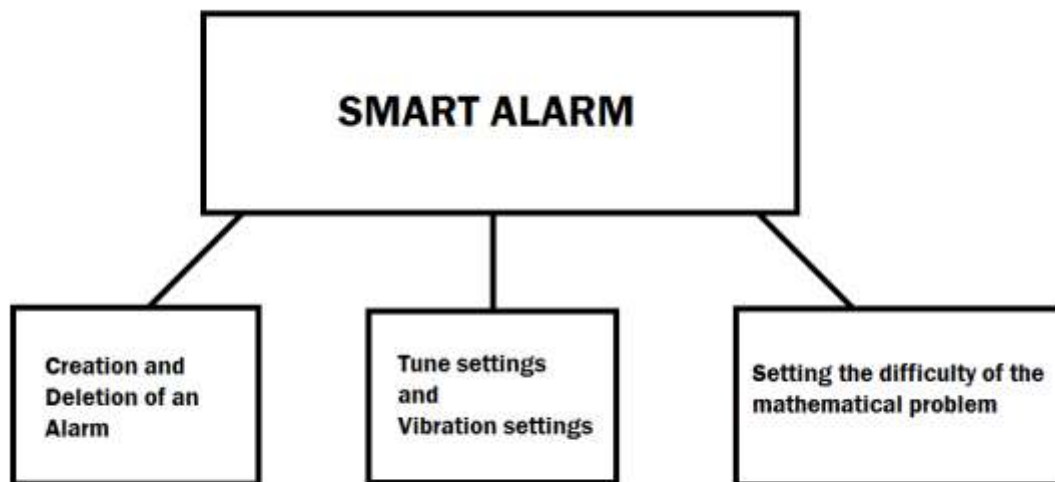


Fig 3.6.1 Overall Description of the application

4.PROJECT DETAILS:

4.1 Home Screen

When we click on the alarm application we are directed to the home screen of the application.



Fig 4.1.1 Home Screen

4.1.1 Working of the application

The working of the application is very user friendly and straightforward like a regular alarm application with the addition of a unique feature in the guise of the solving of the mathematical problem:

1. Initial Setup for Creating an alarm

Creating an alarm consists of the following functions:

1.1 Setting the time and day of the alarm

The time at which the user wants the alarm to ring is set initially by the user

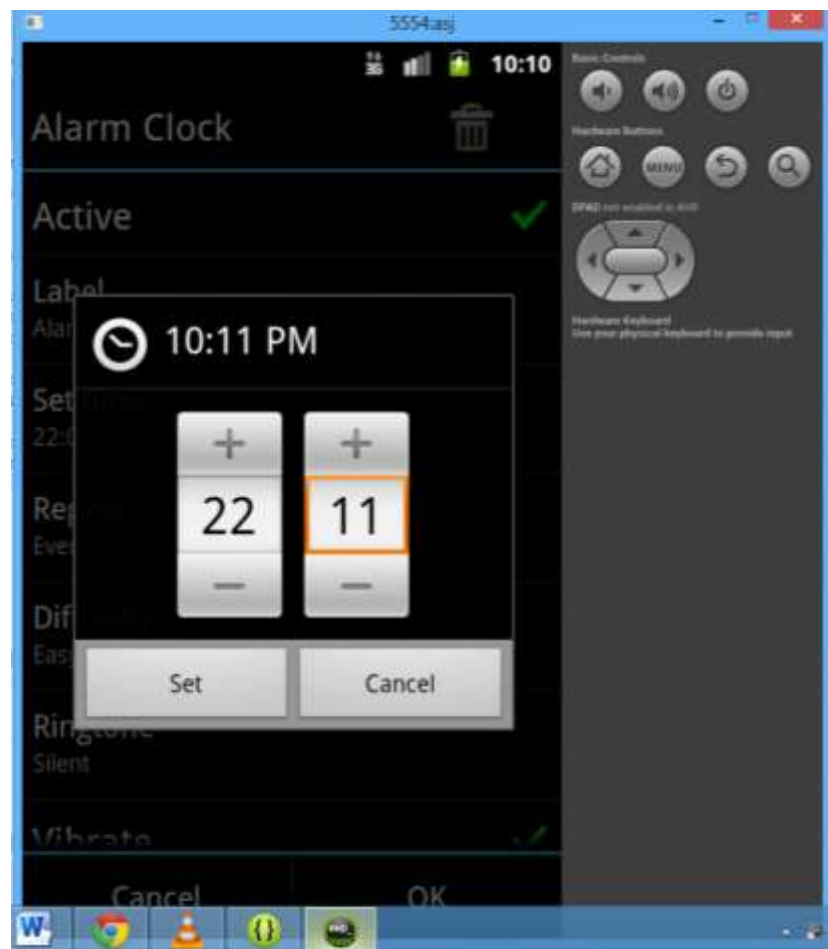


Fig 4.1.2 : Setting the time

He can also select the days at which he wants the alarm to repeat according to his own choice.

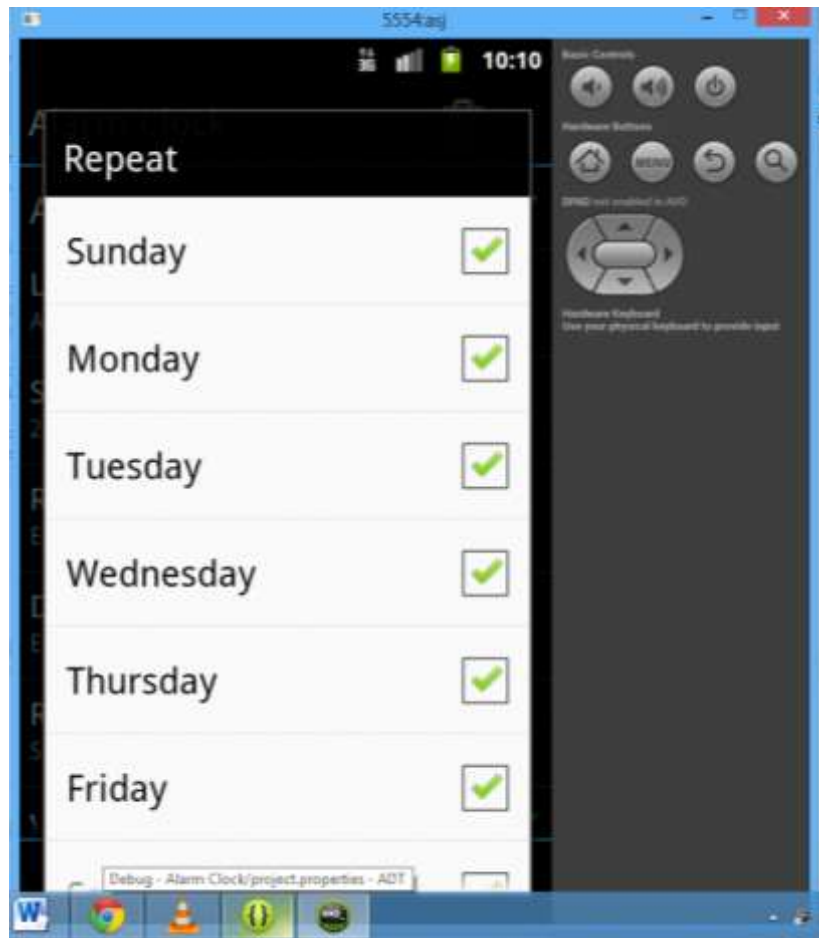


Fig 4.1.3 : Setting the day of repeat

1.2 Setting the tone and vibration settings:

The user can set the tone from the tones available in the user's android device and the user can also choose whether the device will vibrate or it will be on silent when the alarm begins ringing

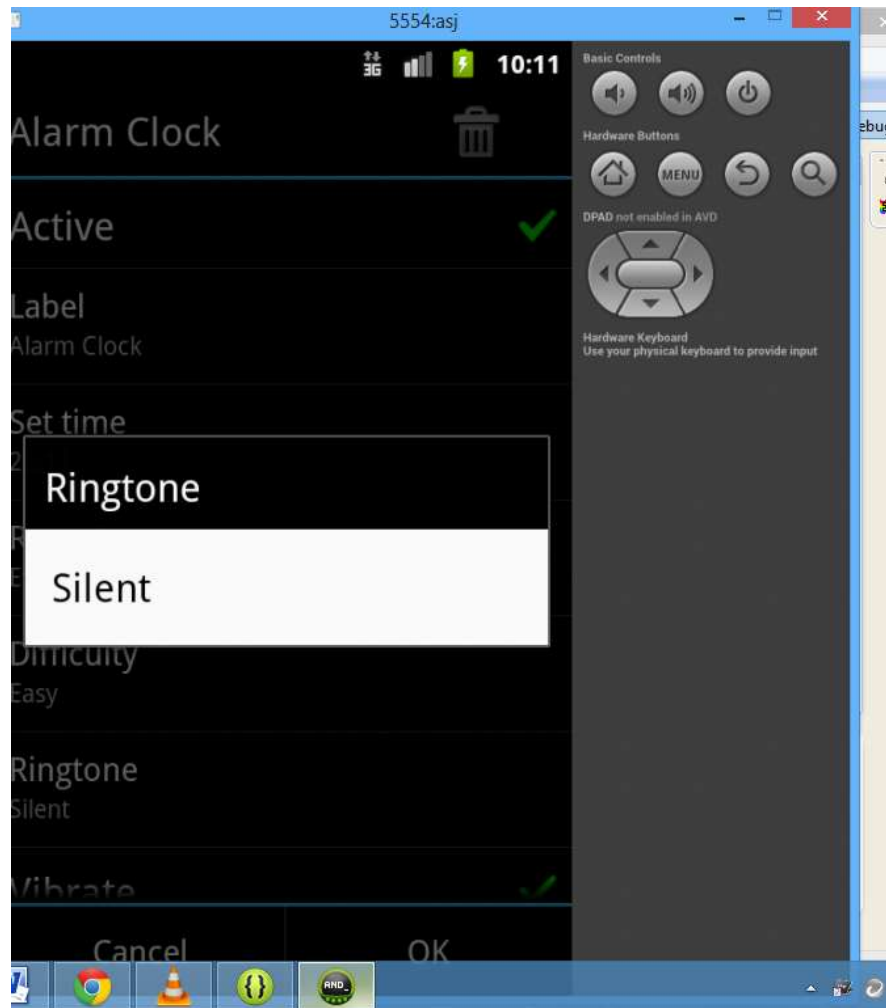


Fig 4.1.4: the Tone settings

1.3. Setting the difficulty of the mathematical problem

The user can set the level of mathematical problem according to his/her convenience. The user can choose from a set of levels: Easy , Medium , Hard.

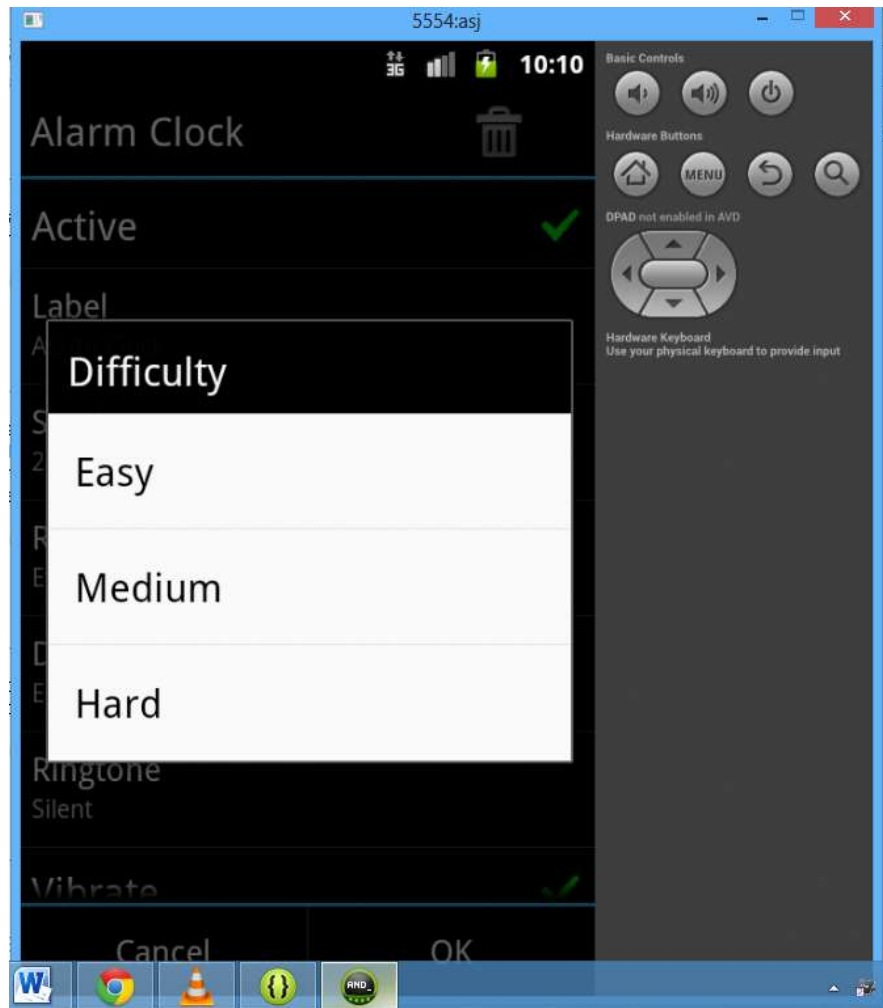


Fig 4.1.5 : Setting the difficulty of the mathematical problem

2. Ringing of the Alarm:

Once the alarm is set , it will ring at the particular time for which it was set and instead of the snooze button, a mathematical problem of the chosen difficulty level is displayed on the screen.The alarm stops ringing only when the mathematical problem has been correctly answered.



Fig 4.1.6 : Alarm about to ring



Fig 4.1.7: Mathematical problem displayed on the screen

If the problem is not answered correctly the alarm displays a popup message on the screen.

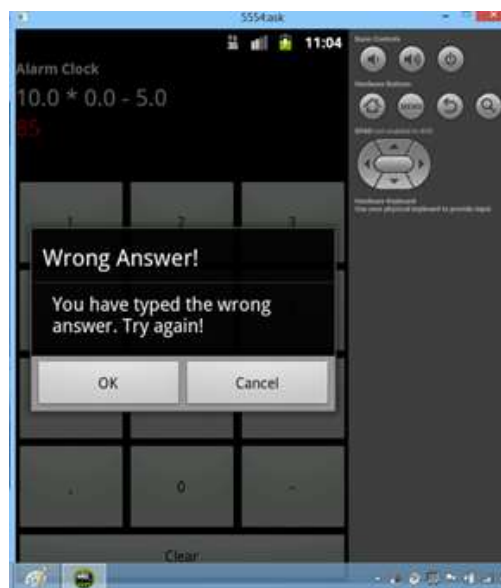


Fig 4.1.8 : Incorrect answer to the mathematical problem



Fig 4.1.9 : Report a bug and go to google playstore

5. CONCLUSION & FUTURE SCOPE

5.1 CONCLUSION

Thus we have created a user friendly alarm application which distinguishes itself from other alarm applications with its mathematical problem feature.

5.2 FUTURE SCOPE

We would like to add some more additional feature into our application. We would like to incorporate a feature where the alarm becomes more independent of the user and truly defines the word 'Smart'.

This includes a last seen feature incorporated into the alarm which helps the alarm to know when the user was last active and thus helps the alarm to calculate the difference in time for which the user set the alarm to the time where the user was last active. This difference is assumed to be the time for which the user has slept and the alarm will then select a tune based on the time for which the user has slept. If the user has slept for less than 5 hours the alarm will select a loud tune and if the user has slept for more than 5 hours the alarm will select a softer tune. We will try to incorporate this feature into our alarm.

REFERENCES:

- [1] <http://developer.android.com/training/index.html>
- [2] <http://www.youtube.com/watch?v=SUOWNXGRc6g&list=PL2F07DBCDC01493A>
- [3] <http://www.youtube.com/watch?v=j-IV87qQ00M>