ACTIVITY VISUALIZATION

By

Capt Izzat

Capt Usman

Capt Noshad

Submitted to the Faculty of Computer Software Engineering Department

National University of Sciences and Technology, Islamabad

in partial fulfillment for the requirements of a B.E Degree in

Software Engineering

July 2018

ABSTRACT

In this project we intended to create an android app which will help health conscious persons to maintain a healthy life .It will help doctors to check their patients (Diabetic and Obesity) that they follow the routine which is prescribed by them by using a smart phone. This is developed by using the internal sensors embedded in the smart phones. This project will use Step Counter and Step Detector sensors, accelerometer and magnetometer which are in android phone. This application would also count steps taken and number of calories burned accordingly. This app also reduces the routine visit of patients to their doctor by sending email to doctor when they have accomplished their target. This will reduce the burden of doctors as well as patients.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that work contained in the thesis –Activity Visualization carried out by **Capt Izzat, Capt Usman** and **Capt Noshad** under supervision of **A/P Touseef Ahmed Rana** for partial fulfillment of Degree of Bachelor of Software Engineering is correct and approved.

Approved By

A/P Touseef Ahmed Rana

Department of CSE, MCS

Dated: ----------

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere. We hereby declare that this document "**Activity Visualization** " neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers especially our supervisor **A/P Touseef Ahmed Rana.**

DEDICATION

First and foremost, We are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project. We have to thank our supervisor, A/P Touseef Ahmed Rana and Evaluation Committee. Without their assistance and dedicated involvement in every step throughout the process, this project would have never been accomplished. Most importantly, none of this could have happened without our Parents. We would like to thank our Parents, whose love and guidance are with us in whatever we pursue.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Introduction

Our project is to build an android app which will help health conscious persons to maintain a healthy life as well as monitoring for doctors to check their patients (Diabetic and Obesity) that they follow the routine which is prescribed by them by using a smart phone. This is developed by using the internal sensors embedded in the smart phones. This project will use Step Counter and Step Detector sensors, accelerometer and magnetometer which are in android phone. This application would also count steps taken and number of calories burned accordingly.

- Purpose of this app is to keep track of a person's physical activity . This app would also calculate steps taken by a person through step counter as well as calculate calories by using steps data.
- It would also show weekly chart of steps taken and calories burned
- This would also help to maintain healthy routine.Walking is an excellent form of exercise that most people can take part in. The average person walks 3000-4000 steps per day. This app would help user to burn appropriate amount of calories.
- It would also provoke its user to take as more steps as possible and keep track of its walk and calories burned through chart. Intended users of this project are health conscious persons and this application would be developed in android mobile phone.
- Use of this app would make its user to set a routine i.e. user would start burning extra calories by developing habit of walking.
- Activity Visualization is a software based project that aims to visualize activity of its user (i.e. walking). It also shows number of steps taken by user (while performing activity) and calories burned accordingly in the form of statistics.
- This app also help patients to tell their doctors that they have meet the goal(e.g walking 1000 steps daily) suggested by them in the form of email to doctor.

1.1 Background

Aim to develop an application that would integrate step counter and calorie counter as well as show path that user followed while maintaining weekly chart for steps and calories data. It will also be helpful to Diabetic as well as Obesity patients.

1.2 Motivations and Challenges

Motivations

Motivations for this project are:

- Good mixture of Step counter and Calorie counter provoke user to maintain healthy life style.
- This application would combine features of counting calories, counting steps as the steps are taken .
- Step counter and calorie counter would provoke user to take more and more steps and know about calories burned relative to each step.
- This would persuade user to be more conscious about his/her health that is a positive aspect.
- This application can be used by anyone anywhere. Even user doesn’t need to be much literate to use this.

Challenges

Challenges faced during accomplishment of desired goals are mentioned below:

- Sensors take values even at no device movement.
- Accelerometers measure the acceleration which represents the actual dynamics of a moving body, but may also be contaminated by components of earth's gravity which may also be sensed.
- The step detector sensor has very low latency in reporting the steps, which is generally within 1 to 2 seconds.
- Version updating may make previous programs difficult to access .
- Running emulator may sometimes hang pc.

1.3 Goals and Objectives

Main objective of this project/application is learning. Other objectives are:

- To develop something new. No other application shows all these features that are exhibited in this application
- Objective is to show number of calories burned in relation to steps taken.
- Maintains a weekly chart of activity.
- Help patients to send their activity data to doctor for further treatment.

Goals include:

- Successful implementation of application
- Cover functional and non functional requirements
- Completion of an application that would be easy to use by the end user

1.4 Solution Overview

Activity Visualization is a software based project that aims to visualize activity of its user (i.e. walking). It also shows number of steps taken by user (while performing activity) and calories burned accordingly in the form of statistics. This application counts steps and also helps us to control weight. The built-in step sensor tracks steps throughout the day. The app keeps track of steps taken even if user phone is in backpack or purse which is a good feature. User will be able to record steps, and burned calories with this app. User can also send his activity

CHAPTER 2

LITERATURE /MARKETSURVEY

2.1 Introduction

Android calorie counting application is an application that helps users to know about their burned calories according to steps taken. This application works by providing necessary information such as weight in pounds and steps taken. The number of calories burned by user's activities will be counted and recorded in the application.

2.2 Market Survey

Market Survey is defined as the study of the spending characteristics and purchasing power of the consumer who are within your business's geographic area of operation; a research method for defining the market parameters of a business[1].

2.2.1 Types of Market Surveys

Market surveys can be administered in several ways. Paper surveys or questionnaires that businesses hand out in public or mail to consumers are common forms of market surveys. Businesses such as restaurants and service companies often use questionnaires to gather feedback about consumer experience. Some businesses conduct oral market surveys over the phone, while others conduct surveys electronically via email, on official websites or through third-party websites dedicated to market surveys.

2.2.2 Benefits of Surveys

Market surveys help businesses make better decisions about the types of products and services they offer, prices, how to deal with competitors and whether to enter or exit markets. Analysis of market surveys can prevent a business from making a costly mistake such as launching a new product or service that doesn't fulfill a need in the market, getting into a market that is saturated with competitors and setting prices too high or too low. Surveys can help entrepreneurs assess the viability of new ideas[2.]

During conducting market survey for this project, we found that:

- Physicians have limited time for weight-loss counseling, and there is a lack of resources to which they can refer patients for assistance with weight loss.
- Weight-loss mobile applications (apps) have the potential to be a helpful tool, but their complex layout may be not helpful for most of users.

- This app would help user to keep track of their calories burned at their own mobile phone and user plan to take more steps to burn extra calories.

2.3 Literature Review

A literature review is a text of a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. Literature reviews are secondary sources, and do not report new or original experimental work. Most often associated with academic-oriented literature, such reviews are found in academic journals, and are not to be confused with book reviews that may also appear in the same publication. Literature reviews are a basis for research in nearly every academic field[3].

Very few applications keep record of user path with help of sensors. Most of the applications use Google maps to keep track of route taken.

CHAPTER 3

REQUIREMENT ANALYSIS

Requirement Analysis

3.1 Introduction

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product[4].

- These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications.

- Requirements analysis involves frequent communication with system.

- Requirements analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people

- Requirements analysis of this android application have attractive interface, simple layout, that is not complicated and easy to use. This application can be used by anyone anywhere. Even user doesn't need to be much literate to use this.

- The requirement is that hierarchical structure of android application is correct.

- Permissions for sensors is required in manifest files and permissions must be added according to proper syntax.

- Our android application have reasonable response time and give accurate results.

- Requirements analysis involves hardware and software like Android studio ,Java language, Step Sensor , Accelerometer, Magnetometer i.e. a mobile device with IMU.

- Android studio/java language
- Samsung Galaxy Series/ Android phone

- Internet & Devices

3.2 Problem Scenarios

Problem Scenarios regarding project are mentioned below:

- Sensors give constant values even when device is not moving.
- Sensors are contaminated by gravitational forces and may produce wrong output.
- Version updating may make previous programs difficult to access.
- Running emulator may sometime hang pc.

3.3 Architecture and Use Case Diagram

Architecture defines the way in which an application or a program is developed Three level architecture is used for designing this application.

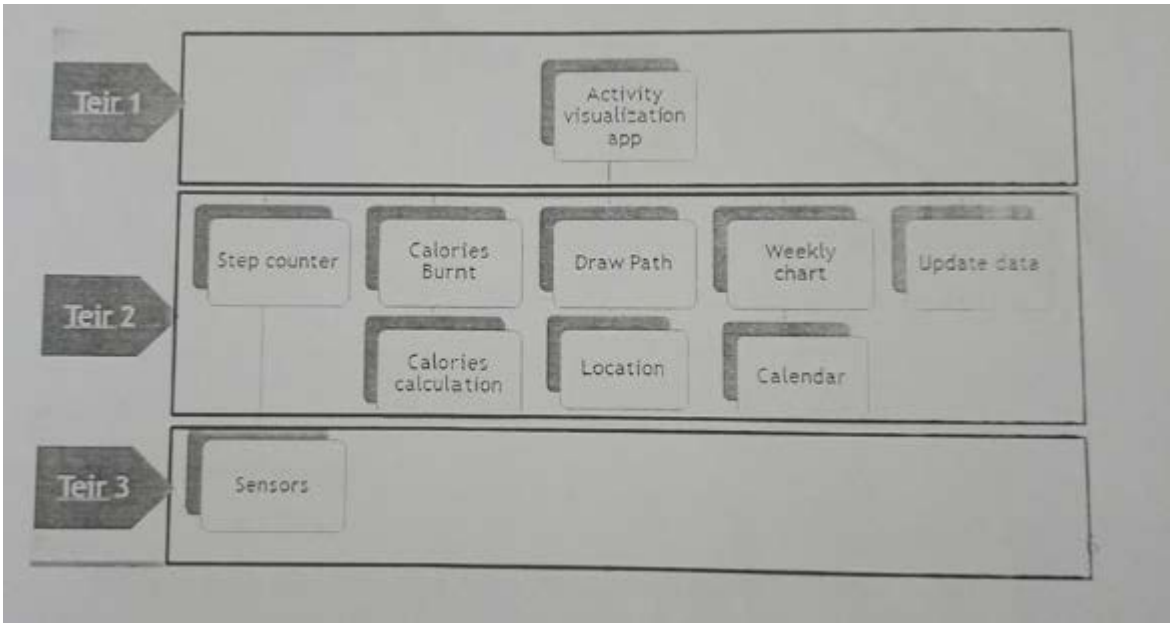


Figure 1 . Architecture Diagram

Use Case diagram illustrates system from user point of view. It overview the usage requirements for a system. They are useful for presentations to management and/or project stakeholders, but for actual development use cases provide significantly more value [5]. Use Case Diagram is shown below in figure 1 which represents three main modules of application. Steps and calorie counting module includes add/update data and weekly chart module. Update/add data module includes taking user name and weight in pounds.

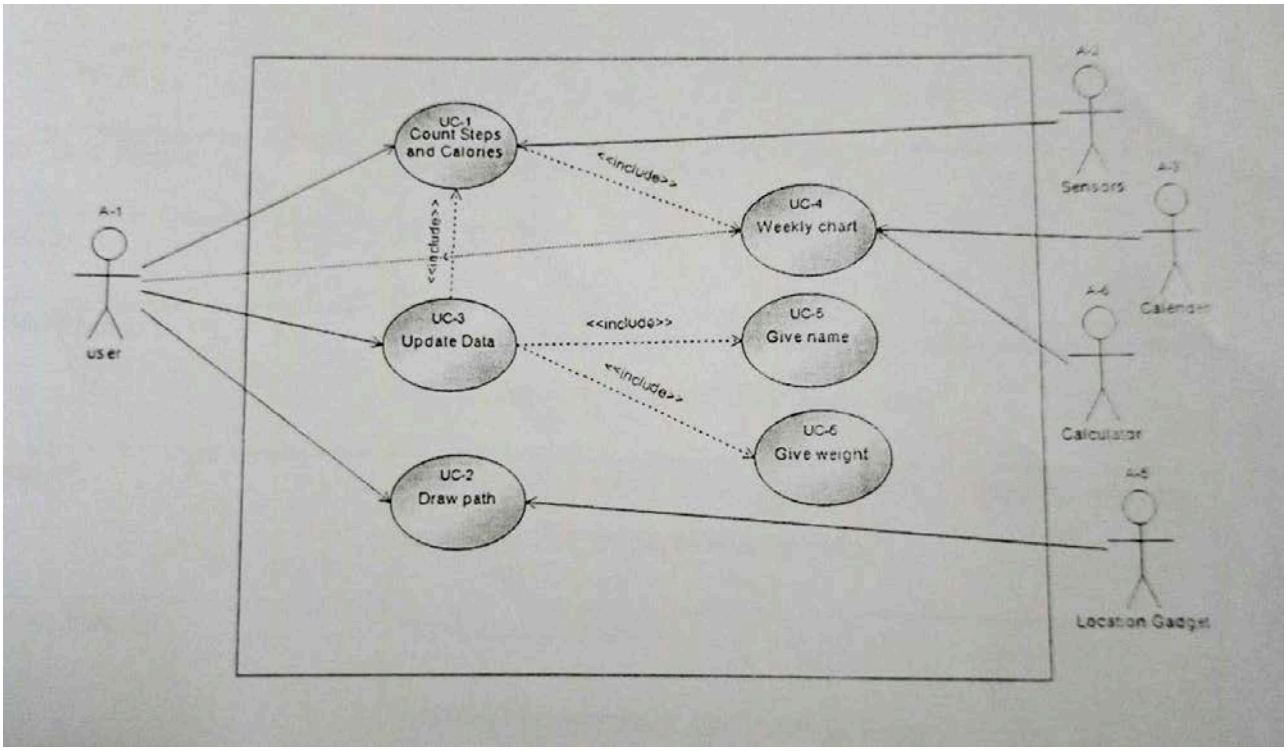


Figure 2. Use Case Diagram

3.4 Use Case List

Below is given list of use case scenarios in which primary actors and uses cases are mentioned.

Primary Actors	**Use Cases**
User	Count Steps & Calories
User	Draw Path

User	Update Data
User	Weekly Chart

3.4.1 Count Steps & Calories

Use Case Id 01 describes Steps and Calorie Counting Module.

Use Case Id	01
Use Case Name	Count Steps & Calories

Actors :	User
Description :	This use case would count user steps and calories burned according to user number of steps taken.
Trigger:	This event would be triggered when user would tap specified button on mobile screen.
Pre-conditions:	Pre-Condition is • Android phone must have step counter
Post-Conditions:	• Steps will be updated. • As well as calories burned according to steps taken .
Normal Flow:	When module is executed , user steps are recorded and numbered step counter sensors. Updated steps & calories burned would be shown in statistical format .

Exceptions :	If user won't enter its data accurately or previously entered data is utilized for new user, wrong calorie burned information would be delivered, as it uses information about user weight .
Includes:	This use case includes update data and weekly chart use case
Special Requirements:	Instead of sensors ,other requirements are : • User must know his weight in pounds accurately to calculate accurate calories burned

3.4.2 Draw Path

Use Case Id 02 describes User path drawn modules with the help of sensors fitted in android phone.

Use Case id:	02
Use Case Name:	Draw Path

Actors:	User
Description:	This use case would draw path taken by user as he moves forward, left or right .
Trigger:	This use case would be triggered when specified button is pressed.
Pre-condition:	• Magnetometer Sensors • Accelerometer Sensors • Must be present in user's device.
Post-condition:	• Path lines would be drawn in red color. • Array of angle changed would be shown on screen.
Normal Flow:	Angle of user movement would be recorded and

	coordinates would be calculated. With the help of this data user path would be shown by lines of fixed displacement.
Exceptions:	Erroneous output would be generated if user show huge multiple hand movement (in which he has grasped device)
Special Requirements:	• User must consider timer i. e within specified time period he has to complete his route. • User must not do unnecessary device movement .

3.4.3 Update Data

Use Case Id 03 describes adding and updating of user data for counting calories regarding each step.

Use Case ID:	03
Use Case Name:	Update Data

Actors:	User
Description:	This use case would collect user data in order to calculate calories burned.
Trigger:	This use case would be triggered whenever user wants to change his data or new user may use the applications .
Pre-condition:	User must know accurately his weight in pounds.
Post-condition:	User data would be saved permanently unless changed.
Normal Flow:	User would enter his name so the step counter module may wish his user, and weight to find burned calories by taking steps .User data would be saved by tapping save button.
Exception:	Error would occur if name and weight are entered in

	alphanumeric format and system may not count calories burned.

3.4.4 Weekly Chart

Use Case Id 04 describes weekly steps and calorie chart maintained by application.

Use Case ID:	04
Use Case Name:	Weekly Chart

Actors:	User
Description:	This use case would show weekly chart for calories burned and steps taken
Trigger:	This use case would be triggered whenever user wants to view this chart.
Pre-condition:	User must trigger firstly step counter and calorie counter module.
Post-condition:	Chart data would be saved and displayed unless week days changed.
Normal Flow:	User may view daily or weekly chart of his activities.
Exception:	Incorrect data would be displayed if wrong data be intended from Step and Calorie Counter activity.

3.5 Functional Requirements:

Functional requirements explain what has to be done by identifying the necessary task, action or activity that must be accomplished. Functional requirements analysis will be used as the top level functions for functional analysis. Functional requirements include:

- Number of steps taken are shown as user moves.
- Target steps to be taken and on completion the notification will appear.
- Calories burned are calculated and shown after user enters his weight in pounds.
- A weekly chart is maintained by saving ever day data regarding steps taken and calories burned.

3.6 Non-functional Requirements:

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors. Non- Functional requirements of this project include:

- Interactive interface
- Graphical icons i.e. buttons, image views and lines drawn.
- Accurate data processing and calculation
- Accessing sensors as and when required.
- Appropriate Response time.
- Comprehensive text
- Simple architecture of application

CHAPTER 4

SYSTEM DESIGN

System design

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering[5].

4.1 Introduction

System design is based on logical design and physical design. As java is an Object Oriented Programing Language that is composed of objects and classes, so logical design is structured.

Java is a modular programing language that is the subset of Procedural Programing. Structured Programing is more efficient and easier to understand and modify

Physical design is basically layout of application, capabilities of sensors and android phone used i.e. its features in relation to run application efficiently.

Below is described Architectural and Detailed design of the application.

4.2 Architectural Design

The architectural design of a system emphasizes the design of the system architecture that describes the structure, behavior and more views of that system and analysis.

4.2.1 Structure of the System

Structure of the system as described above is Object Oriented. Classes are defined and then their objects are used to access them. Structured Languages provides a wide variety of data types and easy to modify and debug. In the source code of Activity Visualization, just after declaration of main activity, data types are allocated to different variables so that according to types, memory would be allocated to that variable. In java, variables for layout views are also specified and initialized in a certain way. After variable declaration/initialization, classes are defined and different functions and operations are performed in them.

4.2.2 Behavior of the System

Our application behaves in a very interactive way. It responds to user input (i.e. entry of weight in pounds) in a timely manner and displays output (i.e. display number of steps taken and calories burned) in a comprehensive way.

4.2.3 Views of the System

A layout defines the visual structure for a user interface, such as the UI for an activity or app widget.

In this application, user can interact with the help of buttons (press button to call a specific piece of functionality) and editText or textView components (output is displayed on these views) of xml layout file of java.

In android applications developed in android studio, there are a number of interactive views from with user can interact with application. These views include TextView, Button, ToggleButton, CheckBox, RadioButon, PlainText, Password, E-mail, Phone, Postal Adress, Time, Date, Number, ListView, GridView, ScrollView, ImageView, Date/TimePicker e.t.c.

4.2.4 Analysis of the System

This application is analyzed to be coded by using features of structured programing and easy to use by the end user.

4.3 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application [6]. Class Diagram is shown in figure 2 which represents five classes of User, Draw Path, Count Steps and Calories, Update data and Weekly Chart. Relevant functions regarding each class are also shown.

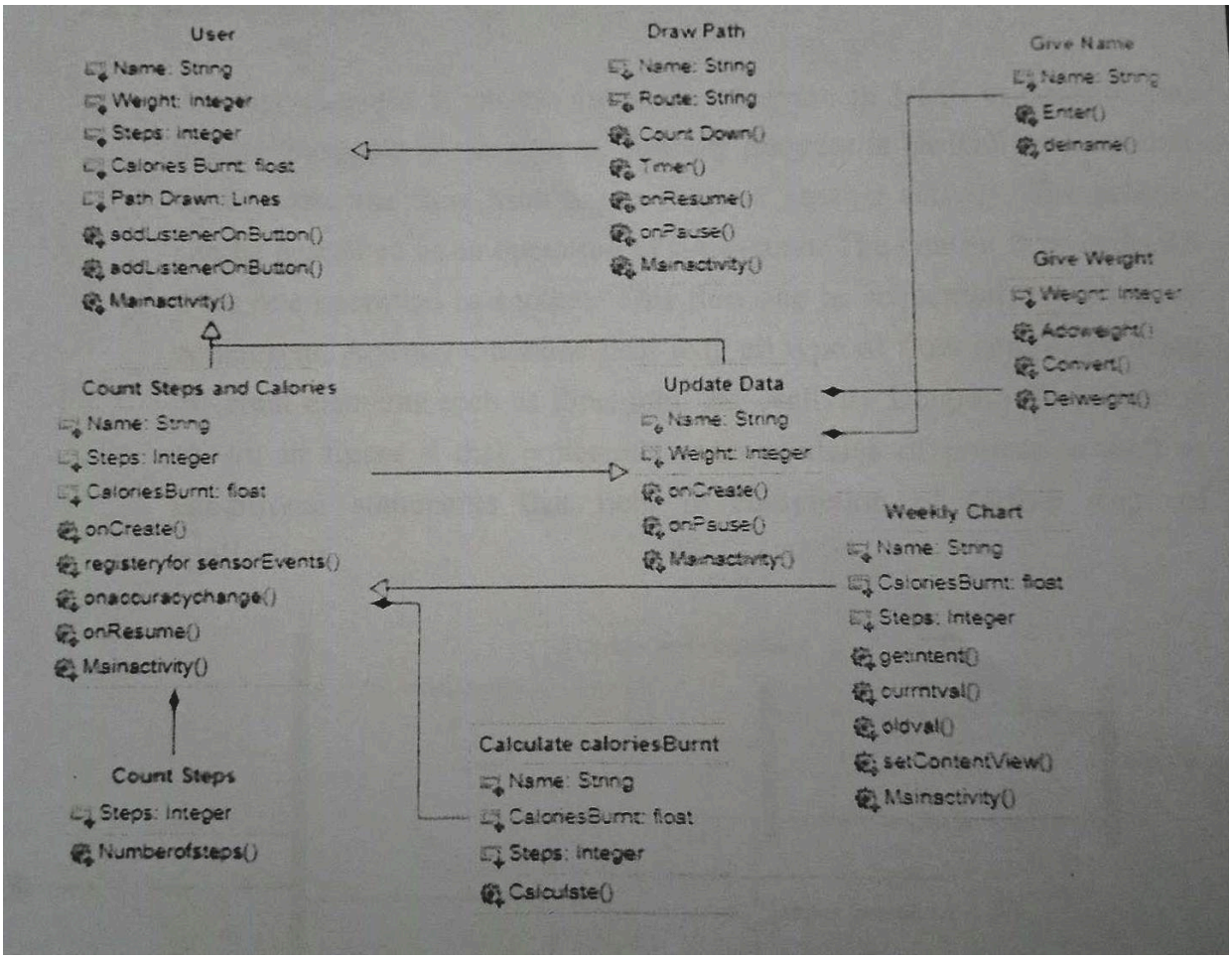


Figure 1. Class Diagram

4.4 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

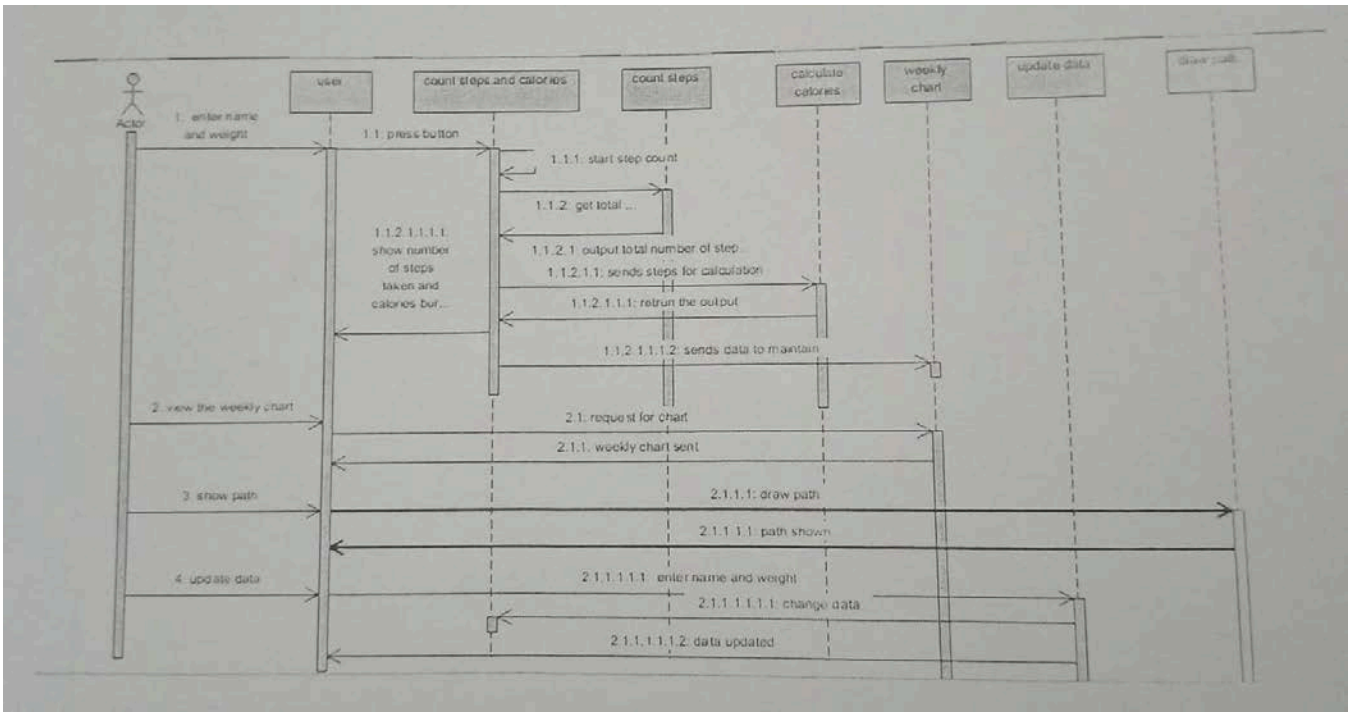


Figure 3. Sequence Diagram

4.5 Detailed Design[6]

Detailed Design is composed of each component interfaces and design descriptions.

4.5.1 Layout Views

Layout views are specified in xml file along with unique identities and different attributes attached to that.

4.5.2 Import Classes

Necessary classes and libraries are imported in java file right after the package name.

4.5.3 Variable Declaration/Definition

After defining main activity, different variables and arrays are declared and/or initialized to be used in the body of the code. Instead of variables of java, xml layout view variables are also defined.

4.5.4 Definition of Classes

Different classes are defined in main activity each to serve a specific purpose. onCreate( ) class is executed when the activity is first created. onResume( ) is used to bring application to previous state after some interruption. onPause( ) is used to terminate application, unregister sensors and interrupt a function running. onSensorChange( ) is used to keep

track of all changing values of sensors used. onAccurcyChanged is called when the values of sensor has been changed.

4.5.5 Function Definition

Functions may be defined outside these classes and called as and when required. A function ShX( )[a] is defined at the end which is composed of series of if-statements.

4.6 Process Model

The project is accomplished by following waterfall Model approach. Waterfall model is also called Linear-sequential life cycle model.in waterfall model, it is necessary to complete previous phase before moving on to next phase and no two or more phases may overlap. By using this model, entire project is divided into different standard phases. Usually in waterfall model, outcome of one phase acts as input of next phase in a sequence. Waterfall Process model is selected so that each phase of project is entirely completed before moving on to next phase. The process model of project is shown in figure 3 which represents all phases from which project is being passed.

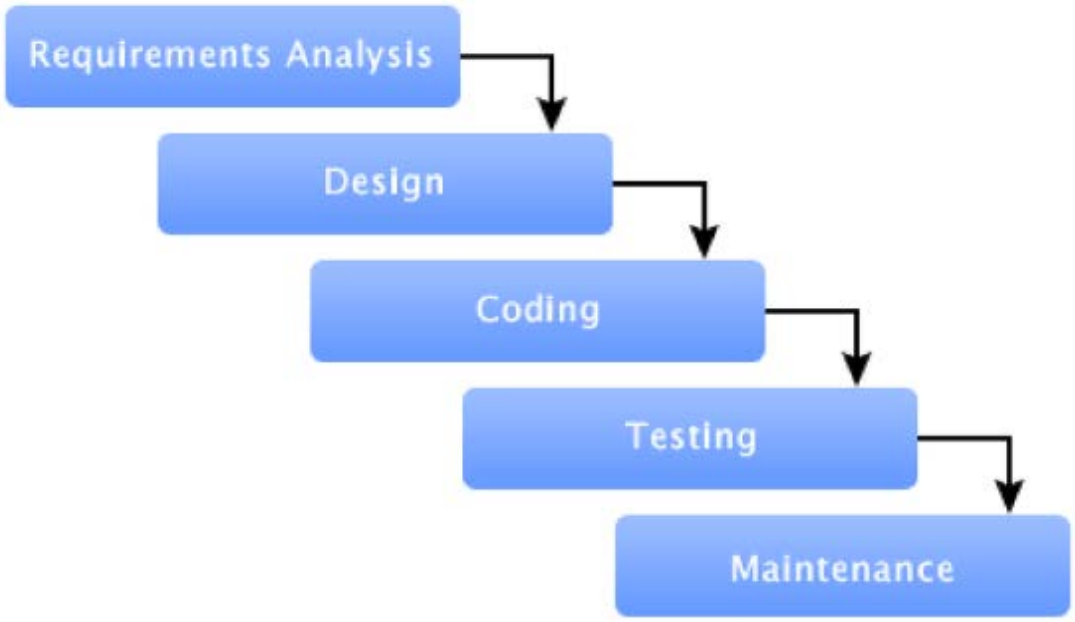


Figure 2. Process Model

- **4.6.1 Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase. In this model, phases do not overlap.

4.7 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. [8]. Activity Diagram for project is shown in figure 4 that represents main modules of project as well as conditional statements that help in completion of further steps of application.

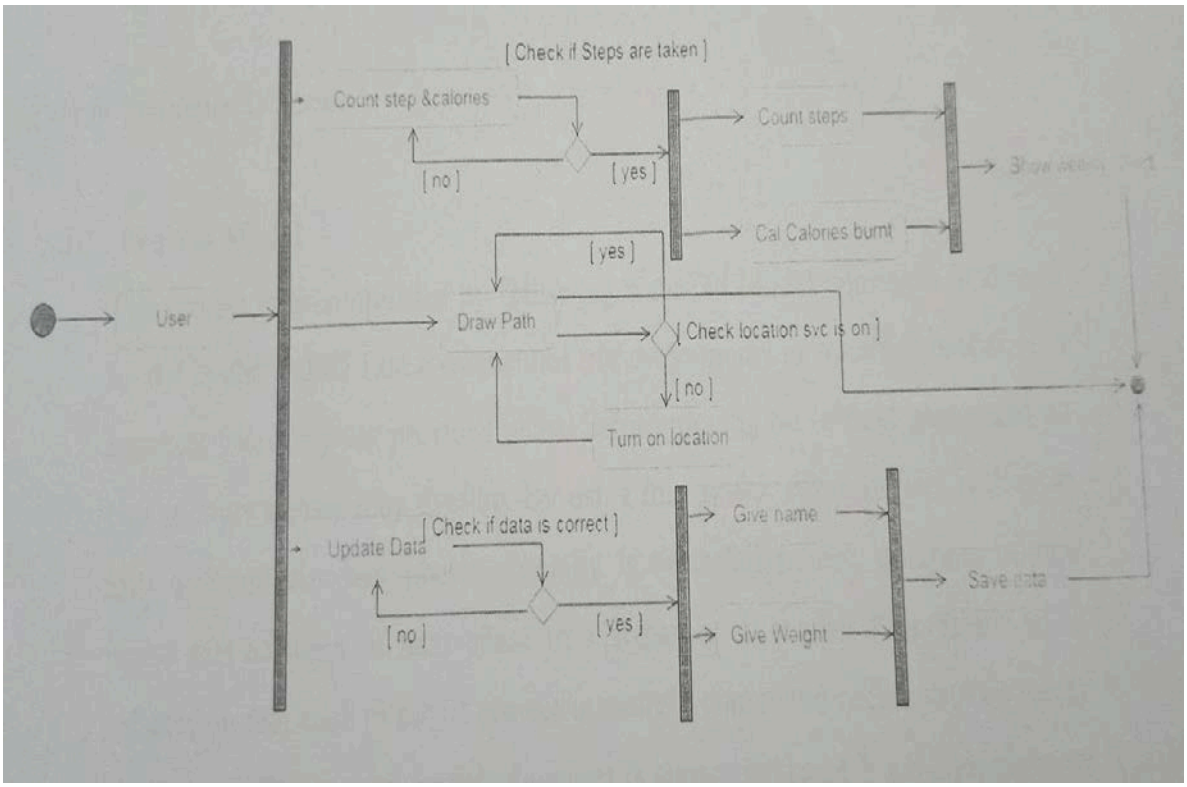


Figure 3. Activity Diagram

CHAPTER 5

IMPLEMENTATION

5.1 Components

The research made to implement this project includes selecting appropriate device, latest version of tool and API level settings.

Device research includes selecting a mobile device with IMU on it. Sensors needed for this project are Step Counter, Magnetometer and Accelerometer. Last two sensors are placed in IMU board and IMU board is not present in every mobile phone. Mobile phone used for this project is Samsung Galaxy S4 that contains all required sensors.

Version of Tool Used is Android Studio 2.3.3. As Android Studio used to recommend updating and updating bring more easiness and advancement, so, possible latest version is used.

API Level Settings are done in accordance with device used. Mobile device has API level 19 so project API level is also 19. Contradictions in API levels halt application run. Each API level has its features and advancements.

5.2 Coding Standards

Coding Standards used in this project follows coding standards presented in the Java Language Specification , from Sun Microsystems, Inc. Major contributions are from Peter King, Patrick Naughton, Mike DeMoney, Jonni Kanerva, Kathy Walrath, and Scott Hommel.

- According to these conventions, java code is composed of blank lines and optional comment statements
- Single line and multiple line comments are used
- Package statement is placed at the top line of main activity in java
- After that short cut of classes and libraries are imported
- Classes and interfaces are declared and defined
- Most of the variables are declared per line
- Each line contains one statement at one time
- Compound and return statements are also used
- If-else statements are used in source code
- For loop is used
- Try-catch statements are used.[6]

5.3 Deployment Environment

Deployment Environment of this project is divided into three tiers:

5.3.1 First Tier/Layout

Layout of this application is first thing with whom user may interact. Layout is designed to be interactive and easy to comprehend. User may see his route followed, number of steps taken and accordingly calories burned with the help of paint function, edit text view and button view respectively. Application would be installed on mobile phone with API level 19 as an android application.

5.3.2 Second Tier/ Source Code

This deployment layer is consist of application actual java or source code. All kind of processing, function calling and formula calculation is done here. Sensors send their value changed and accuracy changed notifications to this tier and this tier is responsible of all formula calculation, processing from raw input data to output result and mechanisms of showing that output to layout screen.

5.3.3 Third Tier/Sensors Working

Data to be utilized in different formulae and functions is mostly based upon sensors working and detection of different angles as well as steps taken. This sensor data is used to generate output lines and statistics.

5.4 Summary

Deployment components include device upon which application would run, tool upon which application is developed and API level. Sun Microsystems coding conventions are followed while developing application. Application would work in three layers: first layer includes mobile screen from which user may interact with application, second layer performs processing upon user input and sensor data and from third layer sensors values are generated to be helpful in calculations.

CHAPTER 6

TESTING AND EVALUATION

6.1 Introduction

Software Testing is done in order to know whether product meets with functional and non-functional requirements as well as to check whether product meets with stakeholders demands. System Testing helps in the production of error free, reliable and efficient final product. Testing is held to find logical, run time, syntax errors etc. in the product. Test Cases are devised for the accomplishment of this task. A test case is a set of data items that the system processes as normal input. Success of test case is related to proportion of maximum errors found that were undiscovered. Correct and wrong data is passed to system in order to check its validation. Mostly real like data is presented for testing. Testing can be done before or after Implementation phase in order to know that does system meets the expectations of designer or not.

There are many approaches to software testing.

- Unit testing
- Interface testing
- Integration testing
- Module testing

6.1.1 Unit Testing

Every module of system is tested before integration in order to check functionality and expected results. All Buttons, Image Buttons, Intended Buttons are tested. Data entry and its results are also validated.

6.1.2 Interface Test

Interface Testing is done to check correct sequence and flow of data between different activities of application. It is checked that Buttons and ImageButtons lead to expected activity, Interface is designed properly, text be readable and helpful for user etc.

6.1.3 Integration Testing

Integration Testing is done when system is put into integration i.e. all modules and sub-modules are combined and executed in unity. This type of testing is done to check that may system not produces errors and wrong output when integrated. Oftenly, Integration of a system produces logical errors that are very important to be removed so that end user may not suffer of it.

6.1.4 Testing Methods

Testing Methods to be used are:

6.1.4.1 Black Box Testing

White Box testing is done when designer don't knows anything about internal working of the system. Black Box testing is done by giving valid and invalid data to application and noted down the output as well as tested correct flow of application.

6.1.4.2 White Box Testing

White Box Testing is carried out when tester know the source code, structure and internal working of system or application. Test cases are developed to do White Box Testing of a system.

6.1.4.3 Grey Box Testing

In this kind of testing method, tester knows the internal working or source code of application but he tests the system as Black Box Testing.

6.2 Test Cases

Following test cases are devised to test the application.

6.2.1 Test Case: Collect User Data

This test case would test user data collection module i.e. name and weight in pounds that will be used in formula calculation of calories burned.

Pre-Condition	The user hast to be on intended activity.
Action	User would enter his/her name and weight in pounds.
Expected Result	User data should be saved accurately and permanently until changed.
Actual result	User data get saved accurately and permanently until changed.
Tested By	Team Members
Result	Pass

6.2.2 Test Case: Count Steps and Calories

This test case would test step and calorie counting module so that user may get correct and useful data.

Pre-Condition	User has to be on intended activity
Action	User would take steps to let sensor determine number of steps taken.
Expected Result	Number of Steps taken by user would be shown as well as calories burned are calculated and displayed on screen in relation to steps taken and body weight entered by user.
Actual result	Correct number of steps taken and calories burned are shown.
Tested By	Team Members
Result	Pass

6.2.3 Test Case: Weekly Chart

This test case would test display of weekly chart regarding user steps taken and calories burned.

Pre-Condition	User has to be on intended activity.
Action	User would press WeeklyChart button on Step and Calorie Counter Activity and user would view chart.
Expected Result	Whole week steps taken and calories burned should be shown along with respective days.
Actual result	Whole week steps taken and calories burned are shown along with respective days.
Tested By	Team Members.

Result	Pass

6.2.4 Test Case: Draw User Path

This test case would test Draw Path module so that it is ensured that user path is tracked and drawn correctly.

Pre-Condition	User has to be on intended activity.
Action	User would walk by holding android device in hand.
Expected Result	Route of user should be drawn on screen by pressing draw button.
Actual result	Route of user draws on screen by pressing draw button.
Tested By	Team Members
Result	Pass

6.3 Performance And Evaluation

By evaluating the system, it is clear that all the objectives are met i.e. number of steps taken, calories burned, user data is collected, weekly chart is shown, and user route is drawn. Almost all functional and non-functional requirements are met. The only deficiency is that sometimes sensors generate wrong values that would be used in formula calculation and hence, sometimes results are not much accurate.

System performance is satisfactory. Application generate output by calculating given formulae and data saved.

6.4 Summary

Testing is a crucial part of useful software development. There are many types and methods to test a software. All the testing methods are applied for testing of this application as well as use cases are also devised for thorough testing.

CHAPTER 7

CONCLUSION AND OUTLOOK

Conclusion and Outlook

In this chapter, final conclusion and overall view about project is stated in the following lines.

7.1 Introduction

Output of the project is shown in the form of lines drawn through Paint(), showing steps taken and calories burned, displaying data of week days in a chart. It is concluded that application almost fulfills its objectives and meets requirements set. It also fulfills functional and non-functional requirements.

7.2 Achievements and Improvements

The achievements that are obtained at the end of this project are:

- Steps are recorded through sensors as user walks
- Calories burned are displayed in an edit text view of android studio by manipulating number of steps taken.
- Weekly data is saved and displayed successfully
- Notification is shown as the target steps are completed

Improvements that are needed are:

- Steps taken per mile are used as averaged i.e. 200 steps per mile. Instead steps/mile can be calculated by user by his self and entered to calculate calorie formula.
-

7.3 Critical Review

This project was selected to develop an android app in java that would make use of sensor values and performs functionality that would be useful for the end user. By analyzing this project, it is found that this application would be of great interest for end user as he knows his calories burned and be provoked to take more and more steps. Object Oriented Programing methodology is used in source code as java is a structured language. Concept behind this project is to combine features of three useful modules and take advantage of them on a single layout. Lesson Learnt from this project is that making use of sensors is not easy and they constantly give values even when device is not in motion due to gravitational force.

7.4 Future Recommendations/Outlook

Future recommendations for this project are:

- Calorie counter would be more accurate if user enters his own steps taken per mile instead of using average steps/miles in formula
- Blood pressure ,sugar and temperature monitoring can also be added at later stage

Outlook of this project is defined as:

- Interface is designed cleanly without adding complex icons
- Most of the data for processing is generated by sensors
- This application is really helpful for people who wants to calculate their calories burned in relation to steps taken

.7.5 Summary

Most of the objectives are achieved but there is still space for further improvements. This application would be of great interest for end user as he knows his calories burned and be provoked to take more and more steps. This application can be made more efficient if higher API levels would be used. As a whole, this project is easy to understand and application is comprehensive for user as well as data processing and formula calculation produces desired accurate results.

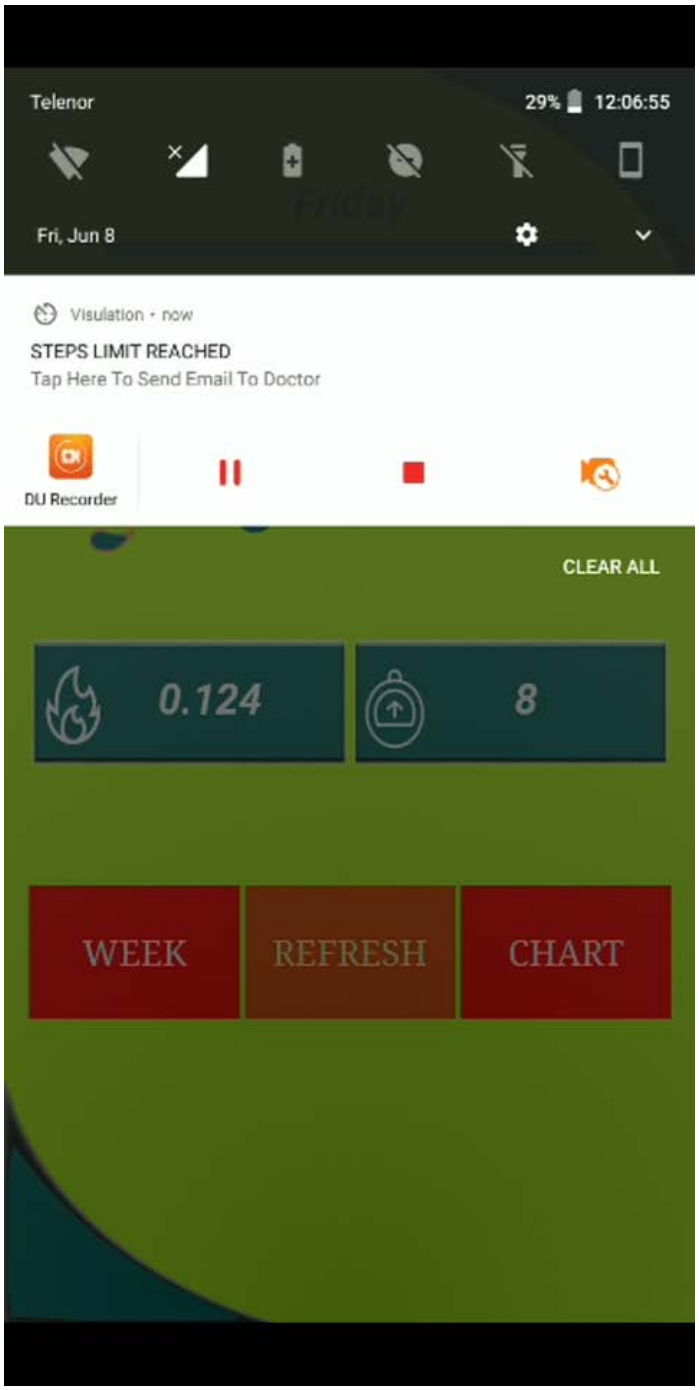
PROJECT SCREEN SHOTS

In this interface user will first enter his name, weight in pounds and steps limit .The required data will be saved for calculation by pressing save button as shown in interface .If the user will not enter data in any of the field a notification will be there to tell user to enter data so that further process will be done.

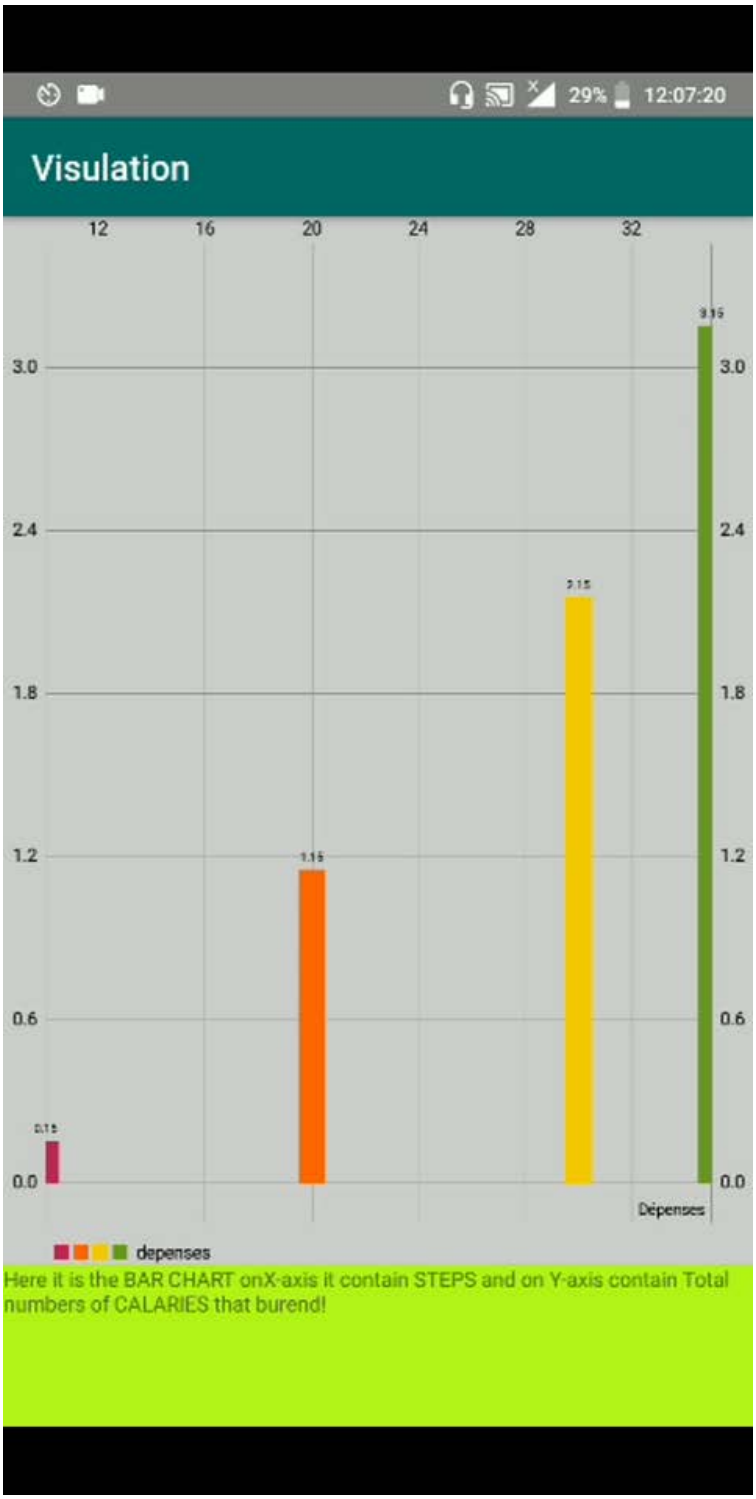
In this interface user will have to click count steps button for application to count the steps taken by user and draw path for showing the direction of user.

This interface shows the day on which user has walked for number of steps and calories burnt in relation to his action. It has weekly chart button for further processing to weekly chart interface as well as refresh button for counting the steps from start(zero).

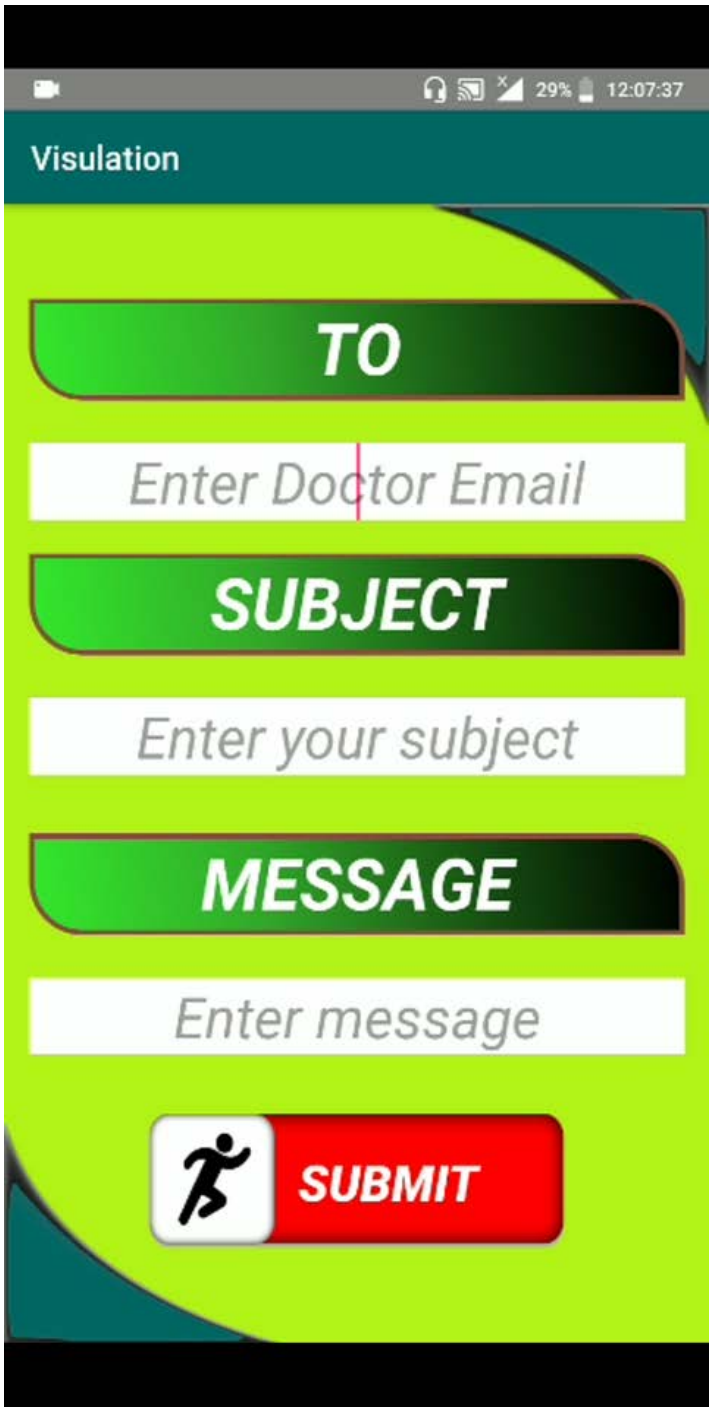
This interface shows that notification will appear as step limit has reached or the target given in the start has achieved .By clicking on this notification will further leads to interface for emailing information to doctor.



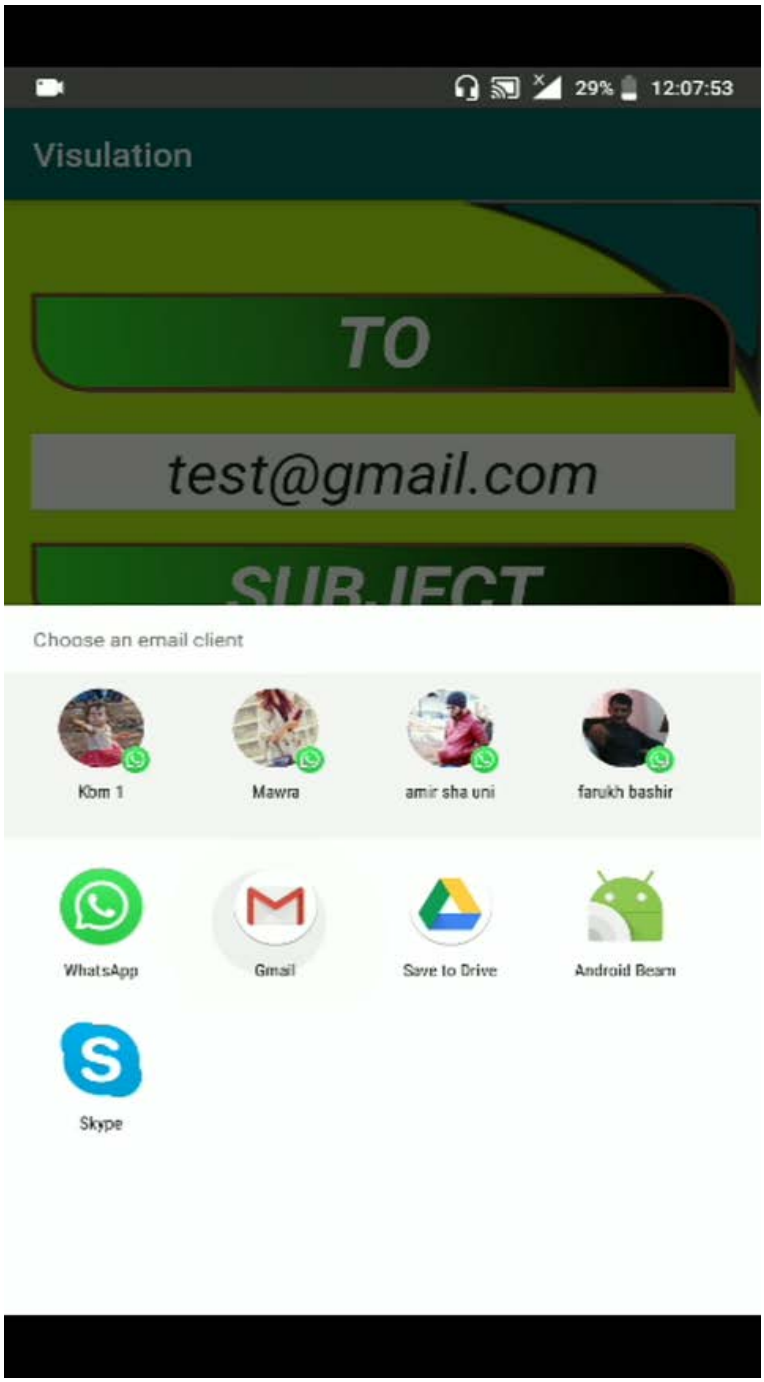
This interface shows the bar chart of activity .Steps taken are shown on x-axis and calories burnt are shown on y-axis.



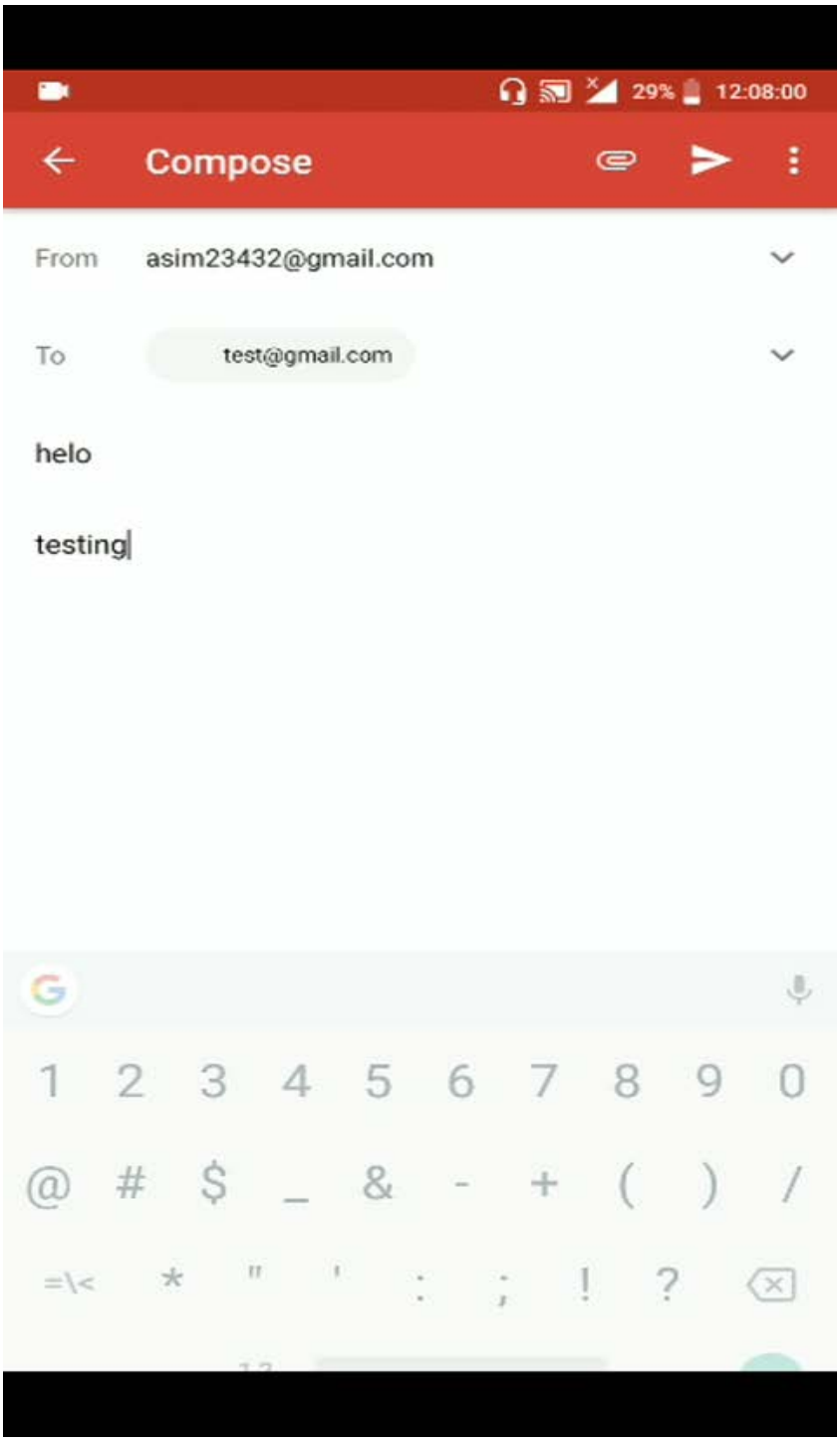
In this interface user will enter doctor email address, subject for which email is sent (i.e steps completion report) and message which the user want to add. User will press the submit button which will redirect to various options to user in next interface.



On clicking the submit button in previous interface, this interface will appear which will give options to user to send his message to doctor through watt Sapp, email etc. User will click on one of option (Gmail) will redirect to Gmail of user shown in next interface.



In this interface user will see his composed email through application and have the option to send it to the doctor .



REFERENCE AND BIBLIOGRAPHY

[1] Sensor Listener, 2017

[1] Entrepreneur Staff .(2017). Market Surveys [Online]. Available: https://www.entrepreneur.com/encyclopedia/market-surveys, 15 Oct 2017

[2] (2017).Definition of a Market Survey[Online]. Available: http://smallbusiness.chron.com/definition-market-survey-40925.html, 15 Oct 2017

[3] Literature review[Online]. Available: https://en.wikipedia.org/wiki/Literature_review, 15 Feb 2018

[4] "Best pedometer apps for Android – AndroidPIT" AndroidPIT [Online]. Available: http://www.androidpit.com/best-android-pedometer-apps, 20 july 2017.

[5] www.agilemodeling.com/artifacts/useCaseDiagram.htm

[5] "What is requirements analysis (requirements engineering) ? - Definition from WhatIs.com" SearchSoftwareQuality, [Online]. Available: http://searchsoftwarequality.techtarget.com/definition/requirements-analysis, 20 july2017.

[6] https://www.tutorialspoint.com/uml/uml_class_diagram.htm

[6] "Systems design" En.wikipedia.org, [Online]. Available: https://en.wikipedia.org/wiki/Systems_design, 20 july2017.

[6]" Detailed Design Documentation Template" Openacs.org, [Online]. Available: http://openacs.org/doc/filename, 20 july2017.

[7] https://en.wikipedia.org/wiki/Flowchart

[7] SDLC Waterfall Model[Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm, 15 Oct 2017

[8] https://www.tutorialspoint.com/uml/uml_activity_diagram.htm

[8]” Layouts | Android Developers” Developer.android.com, [Online]. Available: https://developer.android.com/guide/topics/ui/declaring-layout.html, 20 july2017.

[9]” Code Conventions for the Java Programming Language: Contents” Oracle.com, [Online]. Available: http://www.oracle.com/technetwork/java/codeconvtoc-136057.html, 20 july2017.

[a] Sh(x) is function in Draw Path java code that is used to control minute difference calculations in angle change in order to maintain output.