

# **Chapter 1**

# **SOFTWARE REQUIREMENT SPECIFICATIONS**

# **1. SOFTWARE REQUIREMENT SPECIFICATION**

## **1.1 Introduction**

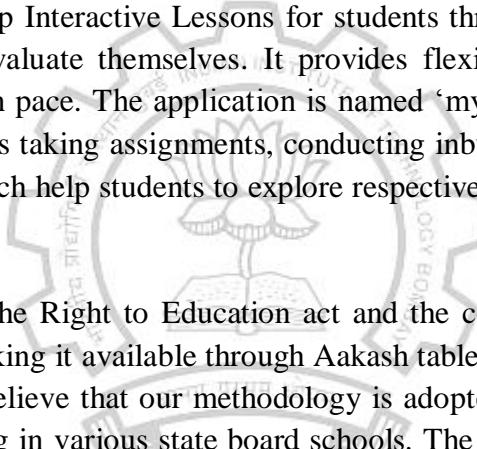
The document aims at defining the overall design, software requirements and features of Interactive Lesson Learner - “MY POINT” *version 1.0*. Efforts have been made to define the requirement exhaustively and accurately.

## **1.2 Document Purpose**

The purpose of this document is to present a detailed description of the **Interactive Lesson Learner - “MY POINT”**. It will explain the purpose and features of the system and what the system will do and also explain how the various modules work and how they communicate with each other for the successful working of the application.

## **1.3 Product Scope**

The product aims to develop Interactive Lessons for students through which they can learn, understand, practice and evaluate themselves. It provides flexibility of studying anytime, anywhere, and at one’s own pace. The application is named ‘myPoint’ which also works as an aid in teaching. It enables taking assignments, conducting inbuilt tests by teachers. It also provides functionalities which help students to explore respective lesson beyond the scope of textbook.



Given the implications of the Right to Education act and the continued scarcity of quality content, we believe that making it available through Aakash tablet will enhance the quality of education in schools. We believe that our methodology is adopted on a large scale and will benefit the children studying in various state board schools. The ability of this application is to establish vision and direction in order to help students and teachers, to empower and inspire them to achieve results or success. It is all about learning, teaching and getting things done, providing freedom to individuals and ultimately allowing people to develop and helping them discover their own strengths. The purpose of creation of this project is to provide a platform through which students can learn and teachers can teach Cartesian coordinate concepts in an easy and interactive way and building upon their skills and capabilities.

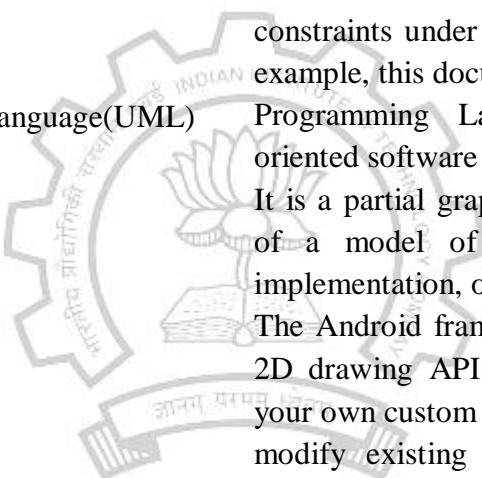
## **1.4 Intended Audience And Document Overview**

The intended audience for this document is the development team, testing team and the end users of the product. The users are students of ninth standard, Maharashtra State Board. The rest of this SRS contains first and foremost the introduction part, which is further subdivided into different sections which includes purpose, then scope of the product. Here the scope of the product is specified including relevant benefits, objectives, and goals. The second section, the Overall Description section, of this document gives an overview of the functionality of

the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next section. The third section, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

## 1.5 Definitions, Acronyms And Abbreviations

<u>Term</u>	<u>Definition</u>
• Android	Linux based operating system.
• Android Canvas	A drawing surface that handles compositing of the actual bits against a Bitmap or Surface object.
• Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
• Unified Modelling Language(UML)	Programming Language used for object oriented software development
• UML Diagram	It is a partial graphical representation (view) of a model of a system under design, implementation, or already in existence.
• Android 2D	The Android framework APIs provides a set 2D drawing APIs that allow you to render your own custom graphics onto a canvas or to modify existing Views to customize their look and feel

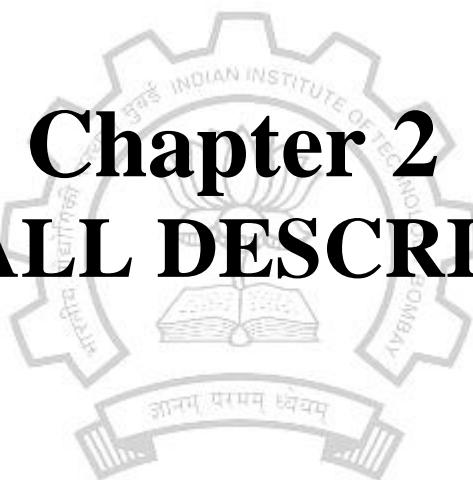


## 1.6 Document Conventions

It uses Times New Roman font style throughout the document for text. Font size 16 for headings, 14 for subheadings and 12 for text. Line spacing 1.5 and margin of 1".

# **Chapter 2**

## **OVERALL DESCRIPTION**



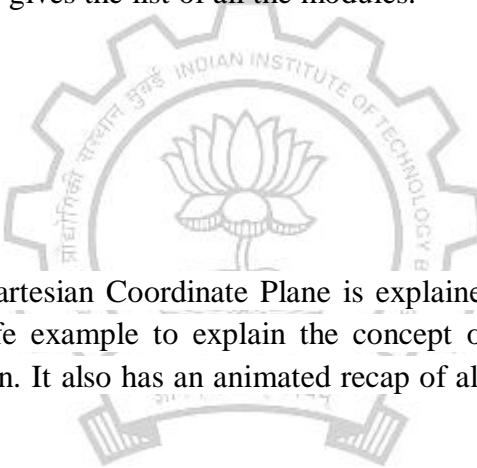
## 2. OVERALL DESCRIPTION

### 2.1 Product Perspective

The Interactive Lesson Builder is a new concept which has been introduced to Aakash Tablet. This is the first time an application is developed which will help the teachers and students to learn/teach concepts in an interactive way using Aakash. The product will first be introduced with the lesson Cartesian Coordinate System. This lesson aims at teaching the students all the concept right from the basics. The product also has provision for teachers to test the students on the concepts learnt. The tests can be conducted at any level that the teacher desires i.e. basic or advanced. This ensures that the student has a good grasp of basic ideas before moving on to advanced concepts. As the product is developed, we plan to introduce more lessons to cover other aspects of secondary education.

### 2.2 Product Functionality

1. The basic home page gives the list of all the modules.
  - a) C-plane
  - b) Plotting
  - c) Scaling
  - d) Line
  - e) Quiz-it



- **C-plane** denoting Cartesian Coordinate Plane is explained in this module, further it includes practical life example to explain the concept of graph, x and y axes and quadrants explanation. It also has an animated recap of all the concept studied in this module.
- **Plotting** module explains process of plotting a point on Cartesian Coordinate System. It includes sub modules as: Practical Life Examples, Point Basics, Solved Examples, Practice and Quiz.
- **Scaling** explains process of finding & using scale for a graph. It includes sub modules as: Practical life examples, Scale Basics, Solved Examples, Practice and Quiz.
- **Line** module explains different process of plotting a point on Cartesian Coordinate System such as using equations, using slope etc. It includes sub modules as: Practical Life examples, Line Basics, Solved Examples, Practice and Quiz.
- **Quiz-it** module provides a vast number of randomly generated questions. It has choice for two levels of quizzes i.e. basic & advanced. It comprises of interactive quiz, multiple choice quiz. It also generates a test report on the submission of the quiz in the form of a pie-chart.

## **2.3 Users And Characteristics**

➤ **Students:**

- Do not have any previous knowledge about the modules implemented.
- Should be able to use touch-screen devices.

➤ **Teacher:**

- Construct a lesson plan using the product to effectively teach students the concepts in the various modules of the application.

## **2.1 Operating Environment**

This product is designed to work specifically on Aakash Tablet with Android ( version Ice Cream Sandwich).

## **2.2 Design & Implementation Constraints**

- The small size of the device screen limits the amount of content visible at a given time.
- The application has been developed fully in English. Its effectiveness will depend upon the user's proficiency in English.
- It will work only on Android devices.
- It is developed specifically for Aakash tablet.

## **2.3 User Documentation**

A detailed user manual and possibly on-line help will be delivered along with the software.

## **2.4 Assumption And Dependencies**

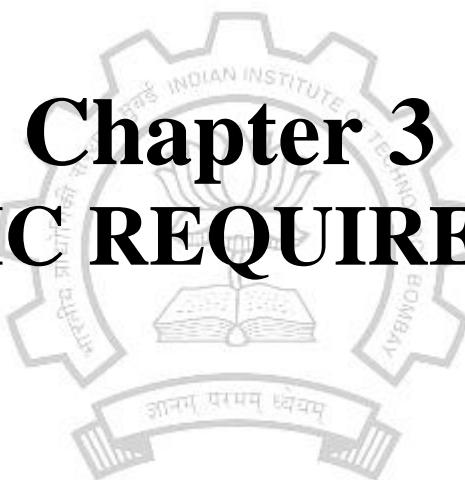
1. The deadline must be met.
2. The product must be reliable.
3. The architecture must be open so that additional functionality may be added later.
4. The product must be user-friendly.

- From the very start of this project we are aware of time constraints so the main emphasis is on extensibility and parallel development. We shall try our best to ensure that project deadlines are met.



# **Chapter 3**

# **SPECIFIC REQUIREMENTS**



### **3 SPECIFIC REQUIREMENTS**

#### **3.1 External Interface Requirements**

##### **3.1.1 User Interface**

User interface must be user-friendly. The user interface shall be designed using various components available in ADT plugins such as View Pager for swapping among different modules, expandable list view to display the list of questions in Multiple Choice Quiz, sliding drawer for hiding and showing the different options in Practice module etc.

##### **3.1.2 Hardware Interface**

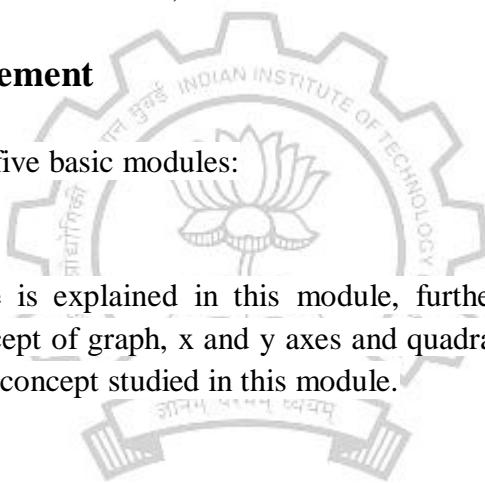
The only hardware required is the Aakash Tablet. Users can also use headphones to listen to the audio associated with each lesson more clearly and SD-Card to store Screenshots.

##### **3.1.3 Software Interface**

- Android 4.0 (Ice Cream Sandwich).

#### **3.2 Functional Requirement**

Home page is divided into five basic modules:



##### **a) C-plane:**

Cartesian Coordinate Plane is explained in this module, further it includes practical life example to explain the concept of graph, x and y axes and quadrants explanation. It also has an animated recap of all the concept studied in this module.

##### **b) Plotting:**

- Practical life examples: Quoting the examples of day today life so that user can relate point with his/her real life.
- Point basics: Explain the basic concepts of plotting point using images and text.
- Solved Examples: Quoting several solved examples to make concept easy to understand.
- Practice: Plotting a point according to coordinates given by user and also gives the coordinate of the point generated by the user's touch on graph displayed on screen.
- Quiz: Users can assess themselves by going through different types of test including multiple choice questions as well.

##### **c) Scaling:**

- Practical life examples: Quoting the examples of day today life so that user can relate scaling with his/her real life.
- Scale basics: Explain the basic concepts of finding & using scaling and its importance.

- Solved Examples: Quoting several solved examples to make concept easy to understand.
- Practice: Calculating different scale and use of kink on graph. Plots graph of specified scale with the specified kink.
- Quiz: Users can assess themselves by going through different types of test.

#### **d) Line**

- Practical life examples: Quoting the examples of day today life so that user can relate line with his/her real life.
- Line basics: Explains the concept of Line from very basic including slope, line parallel to X-axis and Y-axis, by joining two or more different points, drawing line using equation of form :  $ax + by + c = 0$  and  $y = mx + c$
- Solved Examples: Quoting several solved examples to make concepts more clear
- Practice: Practicing line equation of the form  $ax + by + c = 0$ , calculation of X, Y coordinate pairs, and hence lastly line of the respective equation will be drawn and displayed on the graph including steps.
- Quiz: Users can assess themselves by going through different types of test including multiple choice questions as well.

#### **e) Quiz-it:**

This gives a final evaluation which includes the questions of all types. It has choice for two levels of quizzes i.e. basic & advanced. It comprises of interactive quiz, multiple choice quiz. It also generates a test report on the submission of the quiz in the form of a pie-chart with several difficulty levels i.e. Basic and Advanced level. It consists of fixed number of 15 questions.

#### **3.2.2 Additional functionalities:**

- **Scratch Pad:**

The user can write notes, do calculations and can draw graphs. User can also save their work or they can also use this for their rough work

- **Screen shot:**

The user can take screen shots of their practice which get saved into SD card of their tablet. They can refer it later at any time. Screen shots could also be used for assignments.

- **Audio explanation:**

Audio explanation is provided for each module including instructions in various Indian regional languages for example Marathi for the users who are not very acquainted with English.

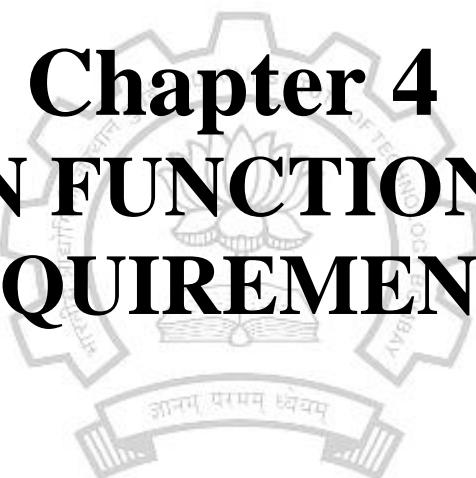
- **Web references:**

For more or extra information, web references are provided. User can directly select them to refer internet links. It allows user to explore & study the concepts even beyond the application.



# **Chapter 4**

## **NON FUNCTIONAL REQUIREMENTS**



## 4 NON FUNCTIONAL REQUIREMENT

### 4.1 Software Quality Attributes

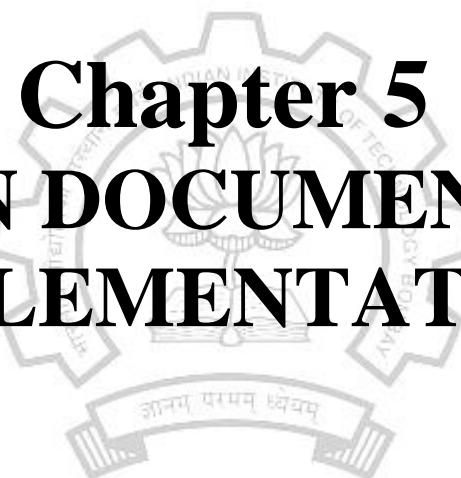
**4.1.1 Reliability:** System must be reliable and data should persist even after suffering some system crashes or booting of android supported devices.

**4.1.2 Maintainability:** Software needs to be upgraded if required in future.



# **Chapter 5**

## **DESIGN DOCUMENT AND IMPLEMENTATION**



## **5 DESIGN DOCUMENT AND IMPLEMENTATION**

### **5.1 Resource Requirements**

#### **5.1.1 H/W Requirements**

The following hardware configuration is required for this project:

- Minimum RAM Required: 512 MB
- Minimum Capacity Required:
  - Internal: 4 GB
  - External: 2 to 32 GB
- Display Resolution: 800 x 480 Pixels
- Microphone
- Touch screen
- Net connectivity

#### **5.1.2 S/W Requirements**

Following software are required for this project:

- Eclipse Juno , ADT Plugin along with Android sdk.
- Microsoft Windows and Ubuntu as our operating system.
- Adobe Dreamweaver to design HTML and CSS part and, this interface was tested on Internet Explorer 9, Mozilla Firefox 11 (and above) and Google Chrome 17 and above).
- We used SqLite as the database to store the Quiz questions, answers and its explanation.

### **5.2 Software Development Life Cycle Model**

The systems development life cycle (SDLC), or software development life cycle in systems engineering, information systems and software engineering, is a process of creating or altering information systems, and the models and methodologies that people use to develop these systems.

In software engineering the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system

A software development process is a structure imposed on the development of a software product. Similar terms include software life cycle and software process. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. Some people consider lifecycle model a general term and software life cycle development a specific term.

Iterative and Incremental development is at the heart of a cyclic software development process developed in response to the weaknesses of the waterfall model. It starts with an initial planning and ends with deployment with the cyclic interactions in between.

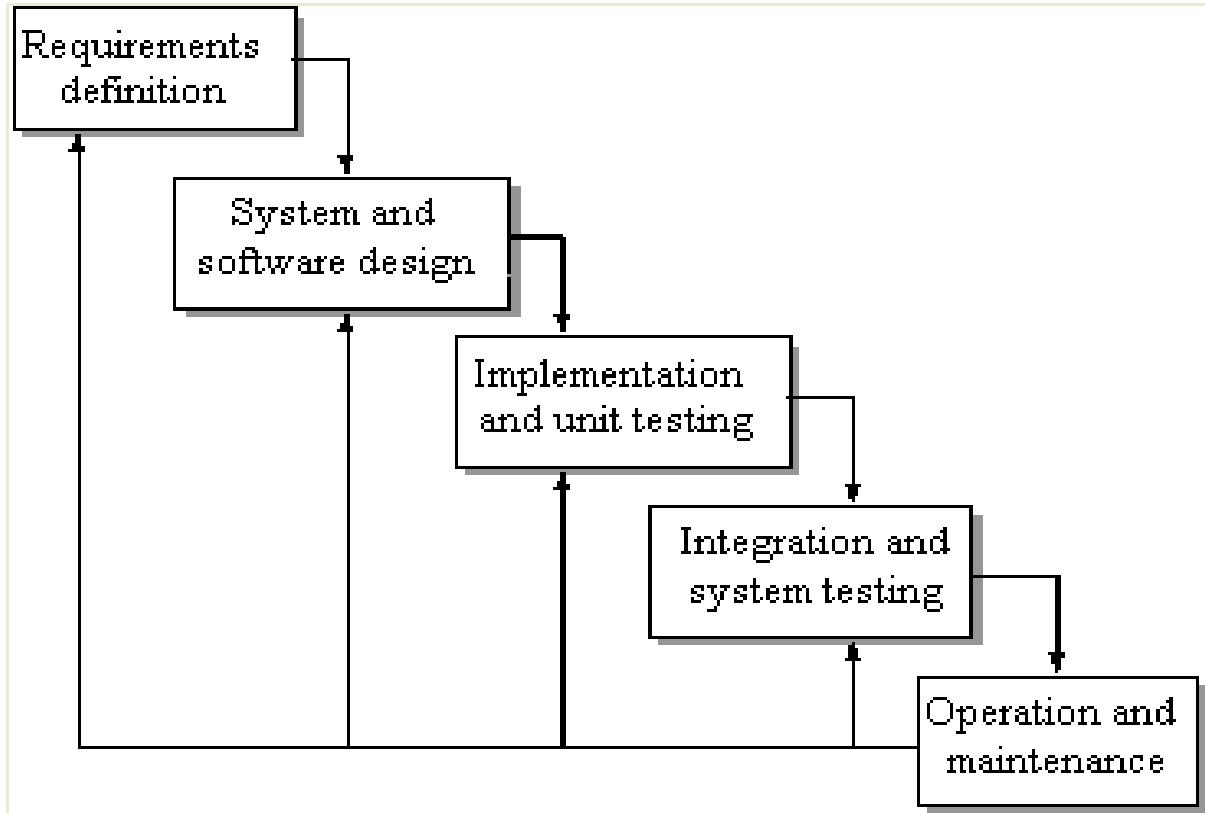
Incremental development slices the system functionality into increments (portions). In each increment, a slice of functionality is delivered through cross-discipline work, from the requirements to the deployment. The unified process groups increments/iterations into phases: inception, elaboration, construction, and transition.

- a) Inception identifies project scope, risks, and requirements (functional and non-functional) at a high level but in enough detail that work can be estimated.
- b) Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.
- c) Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.
- d) Transition delivers the system into the production operating environment.

Each of the phases may be divided into 1 or more iterations, which are usually time-boxed rather than feature-boxed. Architects and analysts work one iteration ahead of developers and testers to keep their work-product backlog full.

The system has been developed using **Iterative Waterfall Model**.

The **waterfall model** is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance.



**Figure 5.0:** SDLC - Iterative Waterfall Model



## Advantages

- Simple goal. Simple to understand and use.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.
- Easy to manage. Each phase has specific deliverable and a review.
- Works well for projects where requirements are well understood.
- Works well when quality is more important than cost/schedule.
- Customers/End users already know about it.

## Disadvantages

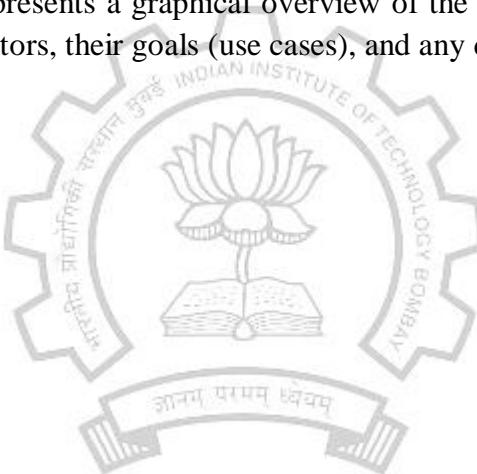
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- No working software is produced until late in the life cycle.
- Risk and uncertainty is high with this process model.

- Adjusting scope during the life cycle can end a project
- Not suitable for complex projects
- Not suitable for projects of long duration because in long running projects requirements are likely to change.
- Integration is done as a "big-bang" at the very end, which doesn't allow to identify any technological or business bottleneck or challenges early.
- Attempt to go back 2 or more phases is very costly.
- Percentage completion of functionality could not be determined in mid of the project because each functionality is undergoing some phase.
- Very risky, since one process can't start before finishing the other.

## 5.3 High Level Design Document

### 5.3.1 Use Case Diagram

A use case diagram presents a graphical overview of the functionality provided by a system in terms of actors, their goals (use cases), and any dependencies between those use cases.



# My Point - APPLICATION



**Figure 5.1:** Use Case Diagram

### 5.3.2 Use Case Description

User is the main actor of the system. The user first launches the application and selects the appropriate module as required. Then, he/she can also practice or can evaluate themselves.

#### ➤ Start My Point:

**Brief Description:** Start My Point function enables user to enter into the main modules of the application.

**Flow of Events:**

- **Basic Flow:** User enters application with default module selected as C-plane.
- **Alternate Flow:** User can view ‘Application Guide’ or ‘About Us’.

**Pre-condition:** User opens myPoint application.

**Post-condition:** Different modules are displayed.

#### ➤ About Us:

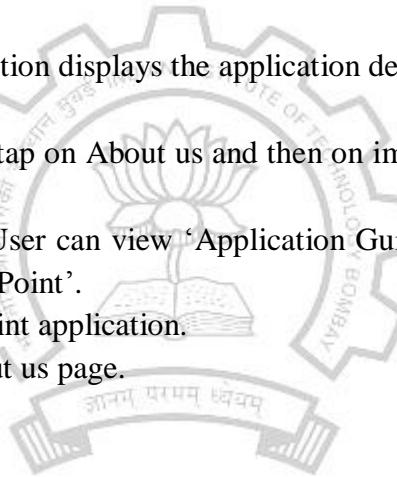
**Brief Description:** About us function displays the application development team details.

**Flow of Events:**

- **Basic Flow:** User tap on About us and then on images & views the respective person’s detail.
- **Alternate Flow:** User can view ‘Application Guide’ or enter into application through ‘Start My Point’.

**Pre-condition:** User opens myPoint application.

**Post-condition:** User views About us page.



#### ➤ Application Guide:

**Brief Description:** This function displays instructions for efficient use of the application.

**Flow of Events:**

- **Basic Flow:** User tap on Application Guide & reads the instructions.
- **Alternate Flow:** User can view ‘About Us’ or enter into application through ‘Start My Point’.

**Pre-condition:** User opens myPoint application.

**Post-condition:** User views instructions.

#### ➤ C-plane:

**Brief Description:** This function displays four sub modules namely location, x-y axes, properties, recap to start learning about Cartesian coordinate plane.

**Flow of Events:**

- **Basic Flow:** User opens one of the sub modules.
- **Alternate Flow:** User can swipe to open other main modules like plotting, scaling etc.

**Pre-condition:** User enters into application by taping on ‘Start My Point’.

**Post-condition:** User reads & practice c-plane concepts.

➤ **Plotting:**

**Brief Description:** This function displays five sub modules namely practical examples, point basics, solved examples, practice and quiz-it which explains concept of plotting points on graphs.

**Flow of Events:**

- **Basic Flow:** User opens one of the sub modules.
- **Alternate Flow:** User can swipe to open other main modules like scaling, c-plane etc.

**Pre-condition:** User enters into application by taping on ‘Start My Point’.

**Post-condition:** User reads & practice or take test for point plotting concepts.

➤ **Scaling:**

**Brief Description:** This function displays five sub modules namely practical examples, Scale basics, solved examples, practice and quiz-it which explains concept of scale on graphs.

**Flow of Events:**

- **Basic Flow:** User opens one of the sub modules.
- **Alternate Flow:** User can swipe to open other main modules like plotting, c-plane etc.

**Pre-condition:** User enters into application by taping on ‘Start My Point’.

**Post-condition:** User reads & practice or take test for scale concepts.

➤ **Line:**

**Brief Description:** This function displays five sub modules namely practical examples, line basics, solved examples, practice and quiz-it which explains concept on different ways of plotting line on a graph.

**Flow of Events:**

- **Basic Flow:** User opens one of the sub modules.
- **Alternate Flow:** User can swipe to open other main modules like plotting, c-plane etc.

**Pre-condition:** User enters into application by taping on ‘Start My Point’.

**Post-condition:** User reads & practice or take test for line concepts.

➤ **Quiz-it:**

**Brief Description:** This function provides two choices on difficulty levels for taking test namely basic, advanced. User select the desired level & take quiz.

**Flow of Events:**

- **Basic Flow:** User select one of the two difficulty levels and takes test.
- **Alternate Flow:** User can swipe to open other main modules like plotting, c-plane etc.

**Pre-condition:** User enters into application by taping on ‘Start My Point’.

**Post-condition:** User quizzes and gets a score.

➤ **Practical example:**

**Brief Description:** This function provides real life examples on each respective concept.

**Flow of Events:**

- **Basic Flow:** User reads, understand and relates the concept to real life situation.
- **Alternate Flow:** User may open other sub-modules like basics, solved examples, practice, quiz-it.

**Pre-condition:** User must enter in one of the modules like c-plane, plotting, scaling or line.

**Post-condition:** User reads the practical examples.

➤ **Concept basics:**

**Brief Description:** This function provides detailed explanation on basic concepts of each respective module.

**Flow of Events:**

- **Basic Flow:** User reads, understand the basic concept of the respective module.
- **Alternate Flow:** User may open other sub-modules like basics, solved examples, practice, quiz-it.

**Pre-condition:** User must enter in one of the modules like c-plane, plotting, scaling or line.

**Post-condition:** User reads the concept basics.

➤ **Solved examples:**

**Brief Description:** This function provides detailed solved examples of each respective module.

**Flow of Events:**

- **Basic Flow:** User reads, understand the solved examples of the respective module.
- **Alternate Flow:** User may open other sub-modules like basics, practical examples, practice, quiz-it.

**Pre-condition:** User must enter in one of the modules like c-plane, plotting, scaling or line.

**Post-condition:** User reads the solved examples.

➤ **Practice:**

**Brief Description:** This function provides an interactive practice module for plotting, scaling & line modules.

**Flow of Events:**

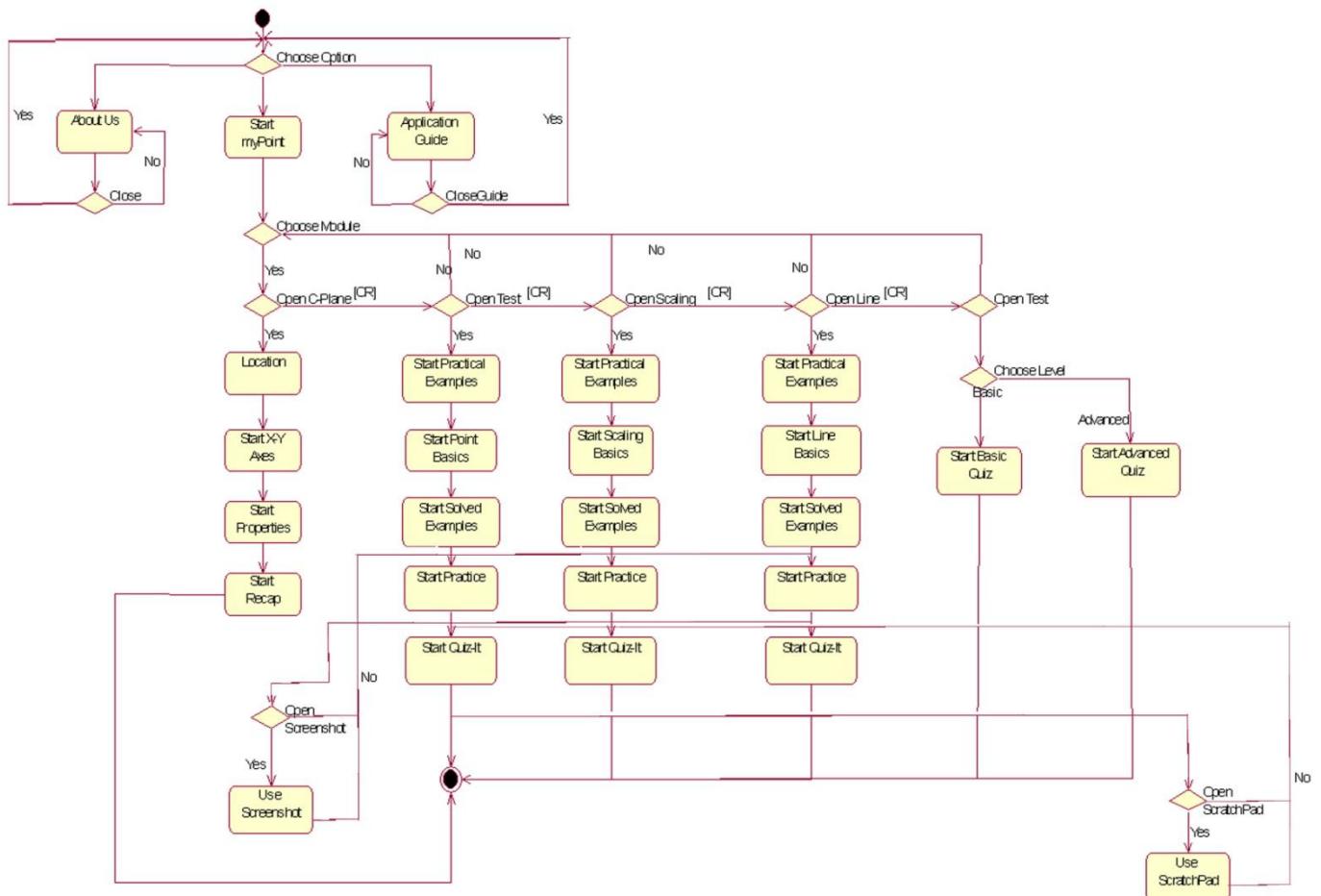
- **Basic Flow:** User reads instructions & practice respective concepts.
- **Alternate Flow:** User may open other sub-modules like practical examples, basics, solved examples, quiz-it.

**Pre-condition:** User must enter in one of the modules like c-plane, plotting, scaling or line.

**Post-condition:** User practices the concept.

### 5.3.3 Activity Diagram

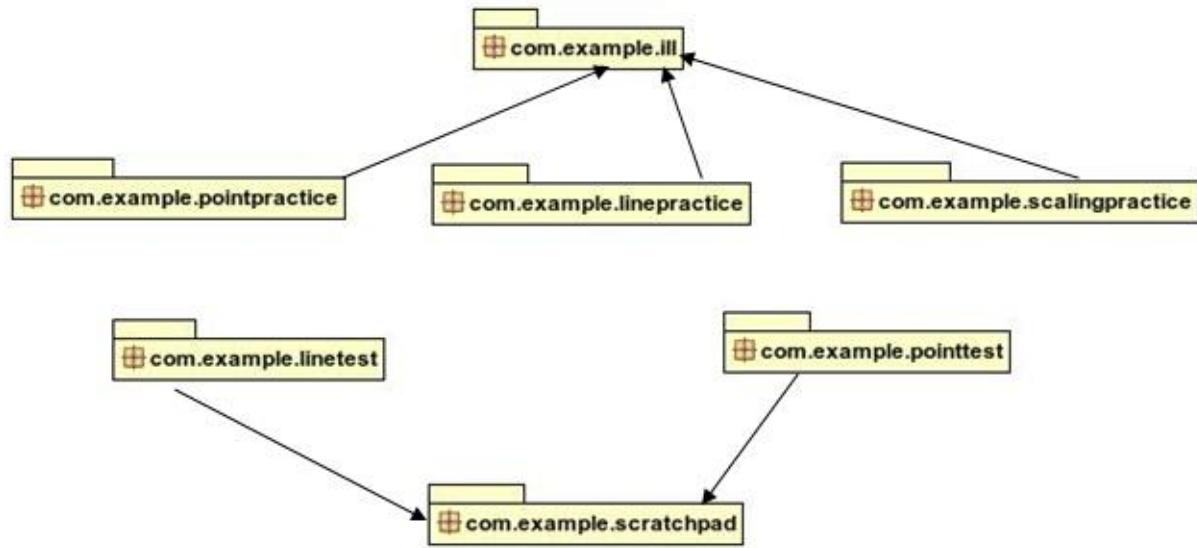
**Activity diagrams** are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.



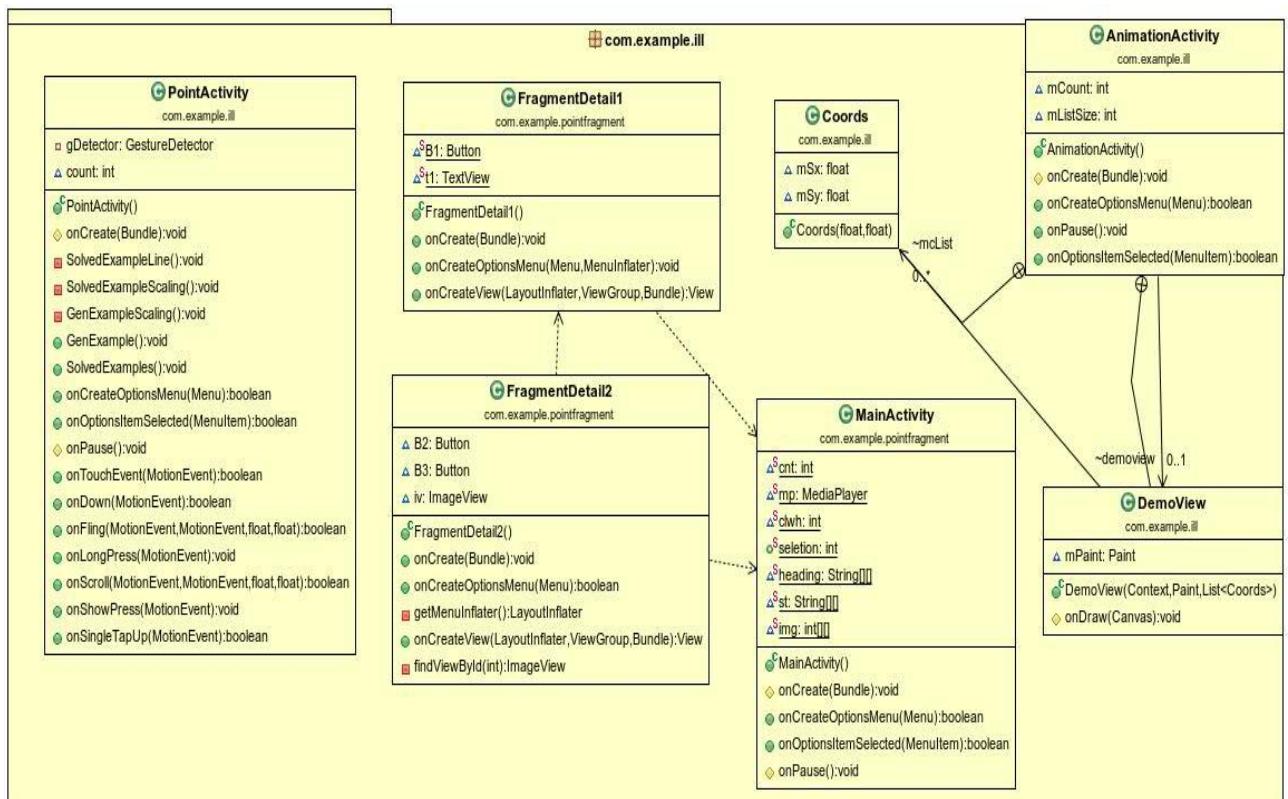
**Figure 5.2:** Activity Diagram

### 5.3.4 Class Diagram :

In software engineering, a **class diagram** in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations(or)methods and the relationships between the classes.



**Figure 5.3 – Class Diagram**



**Figure 5.3.1 – Class Diagram**

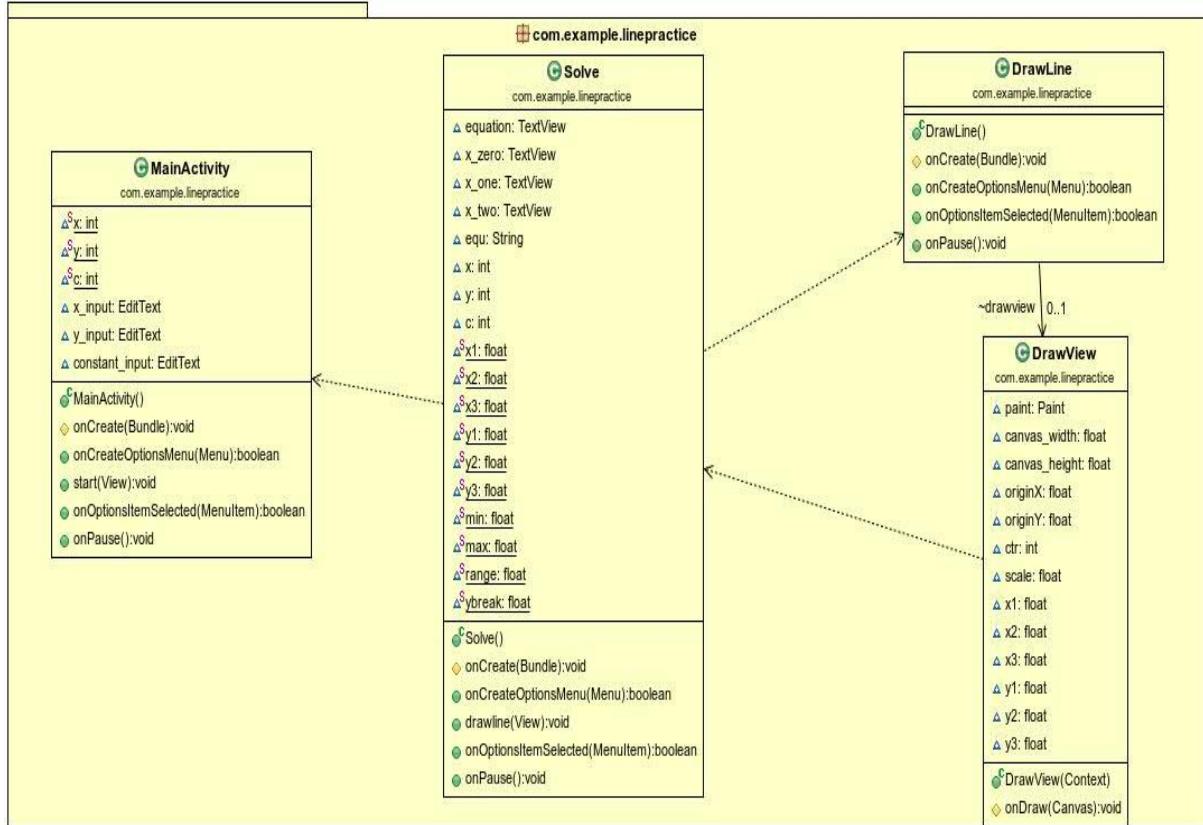


Figure 5.3.2 – Class Diagram

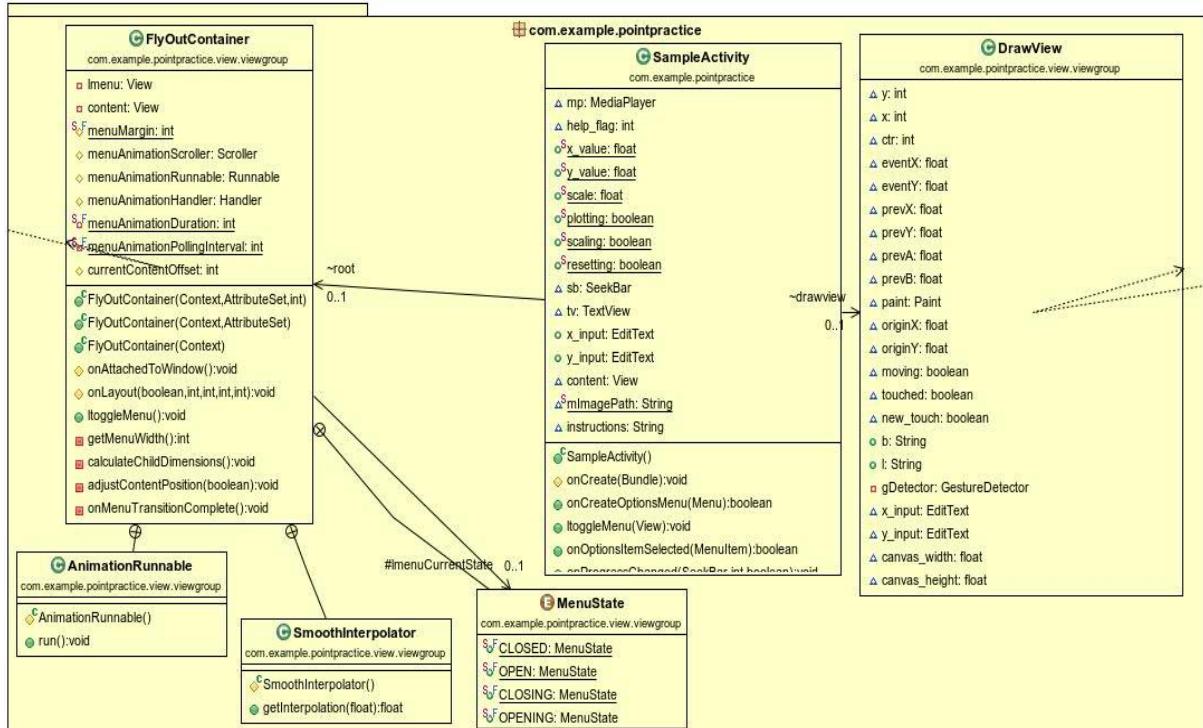


Figure 5.3.3 – Class Diagram

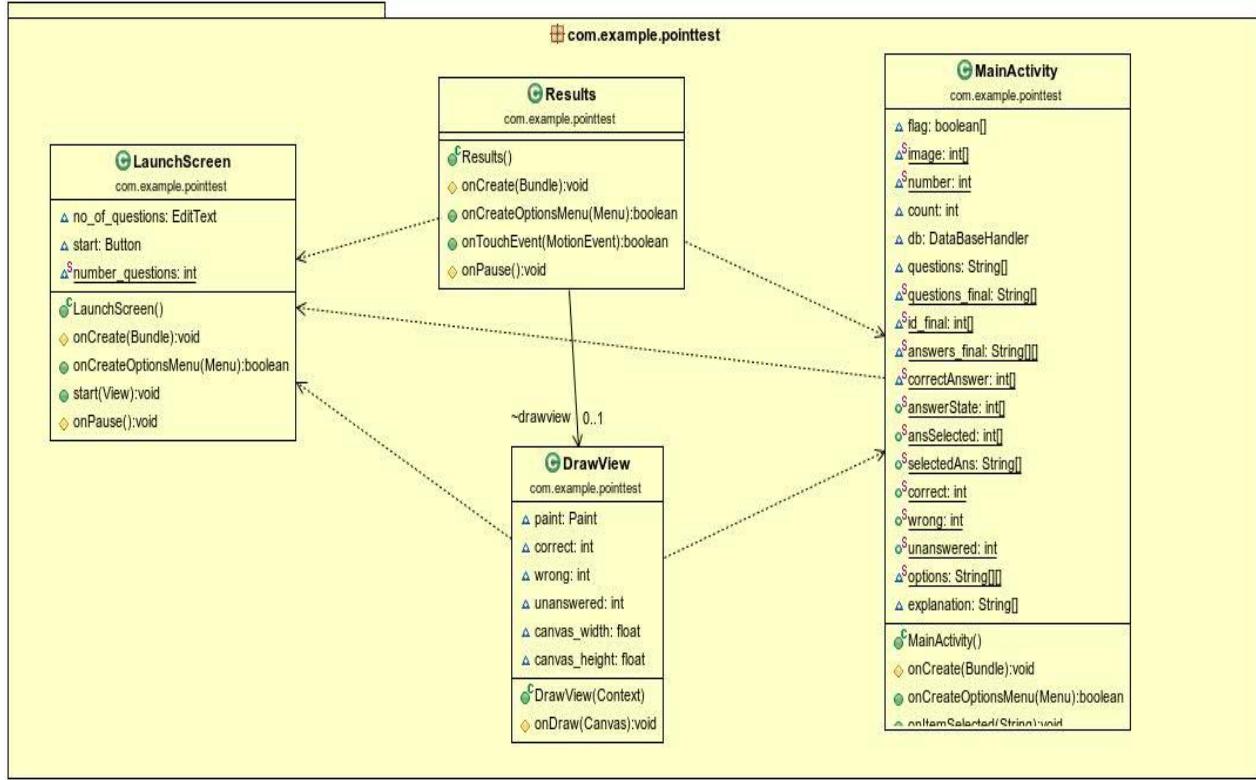


Figure 5.3.4 – Class Diagram

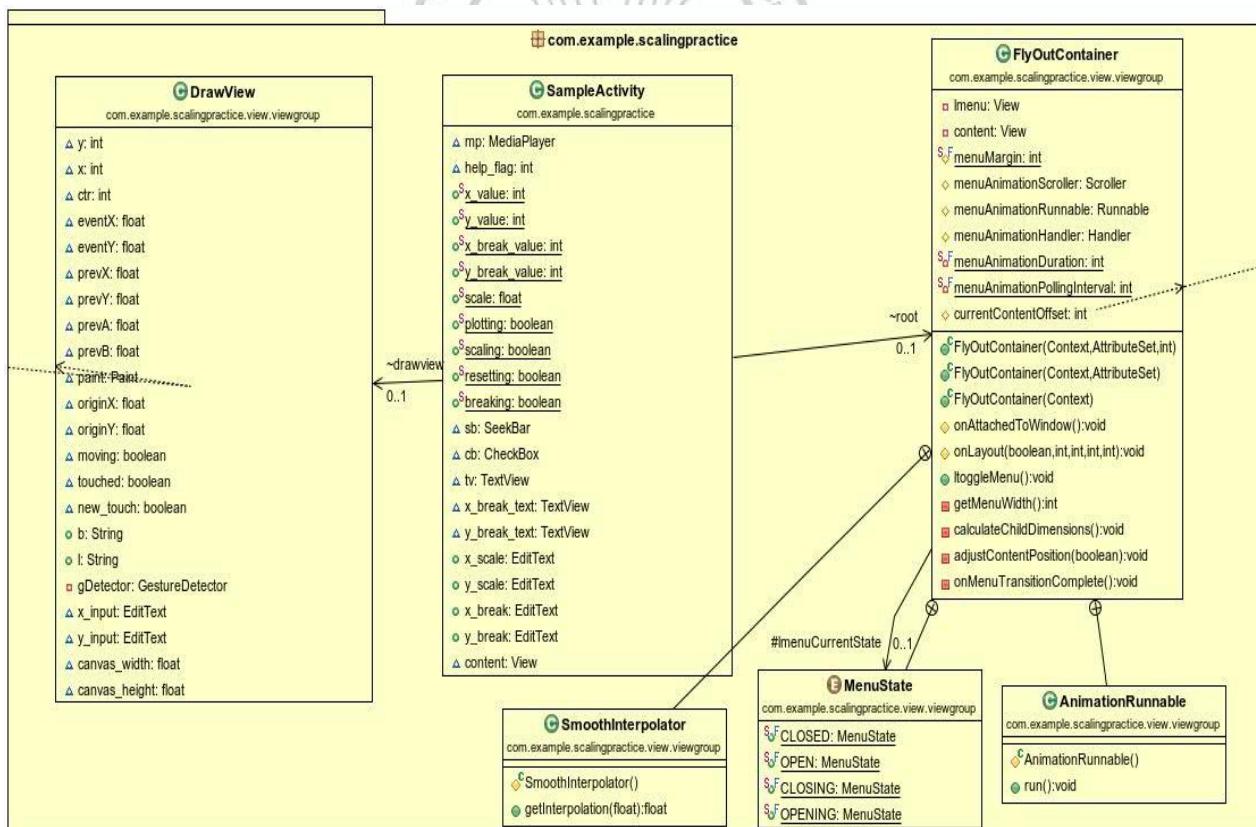
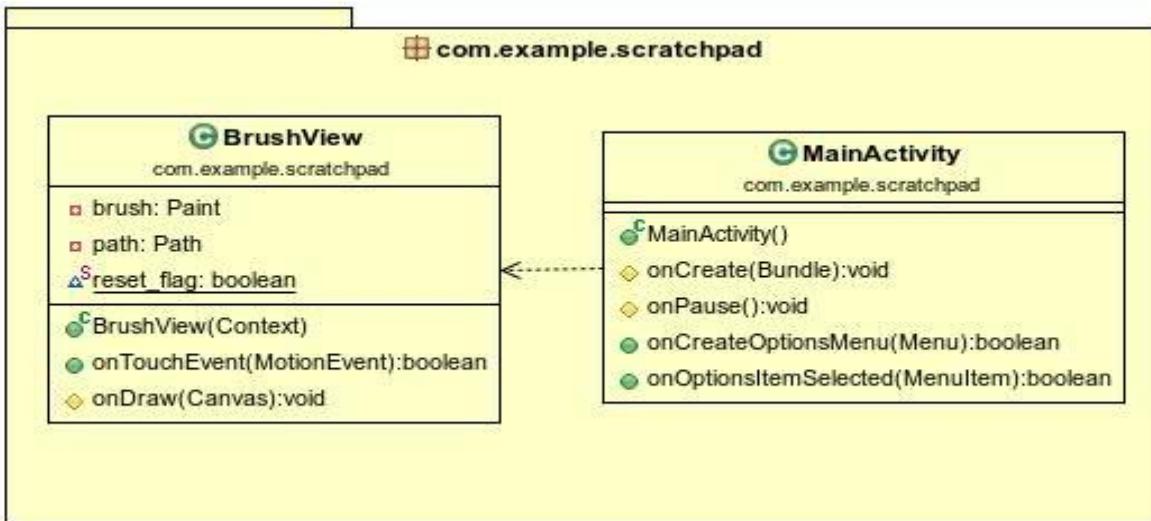
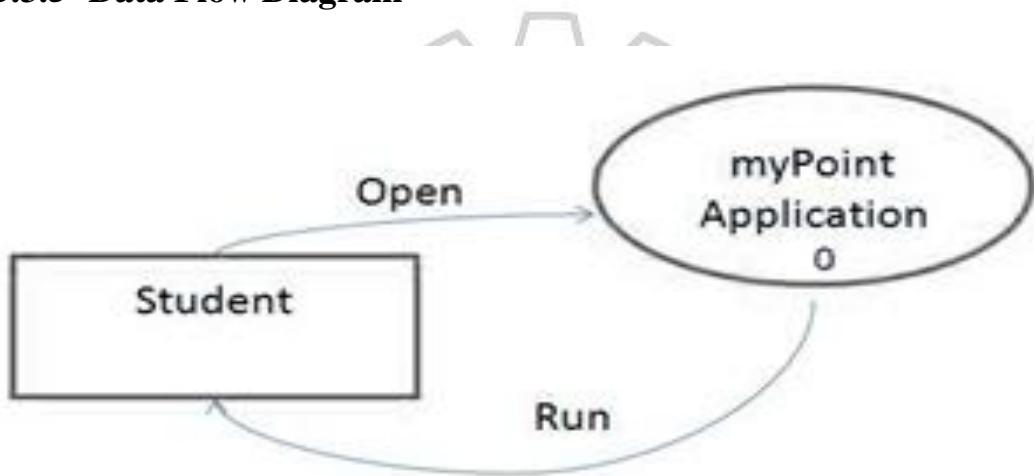


Figure 5.3.5 – Class Diagram

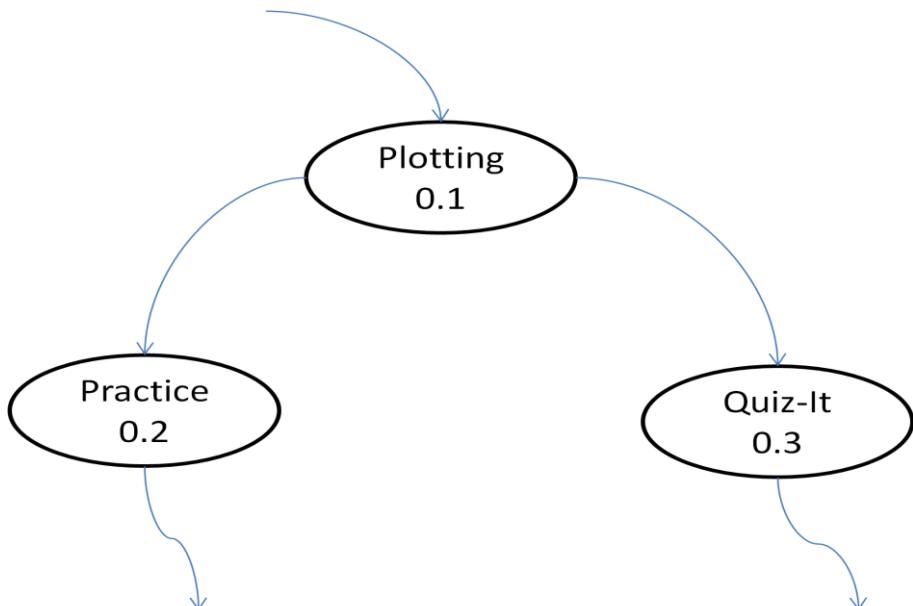


**Figure 5.3.6 – Class Diagram**

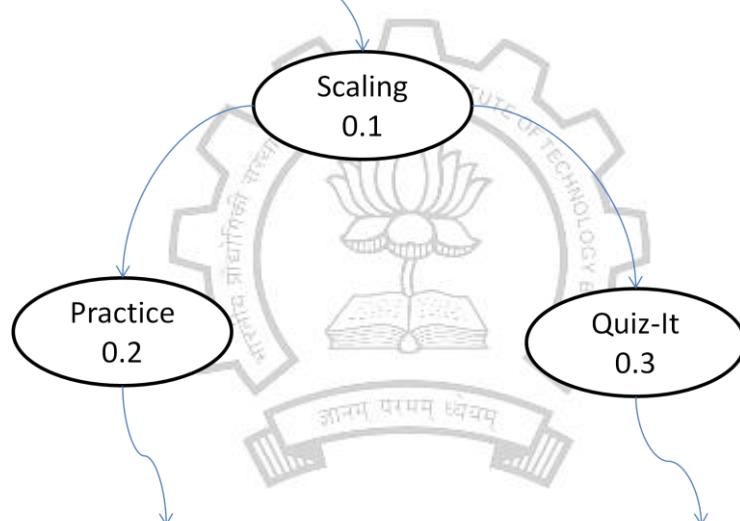
### 5.3.5 Data Flow Diagram



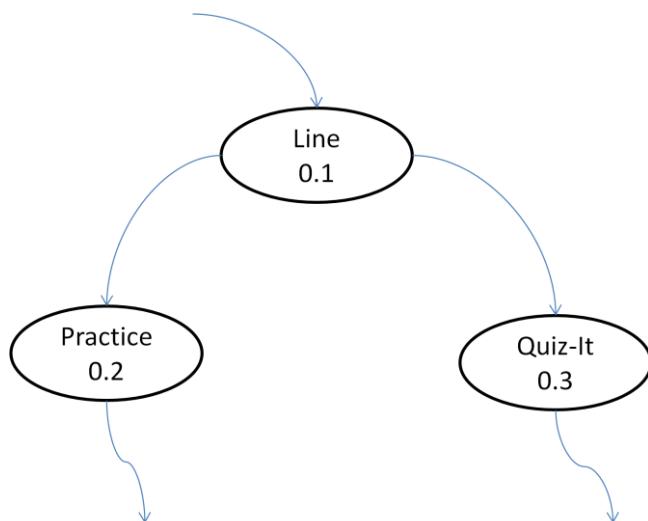
**Figure 5.4 - Level 0 DFD**



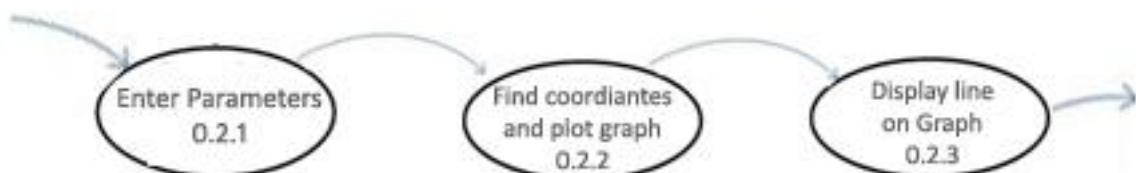
**Figure 5.5.1 - Level 1 DFD**



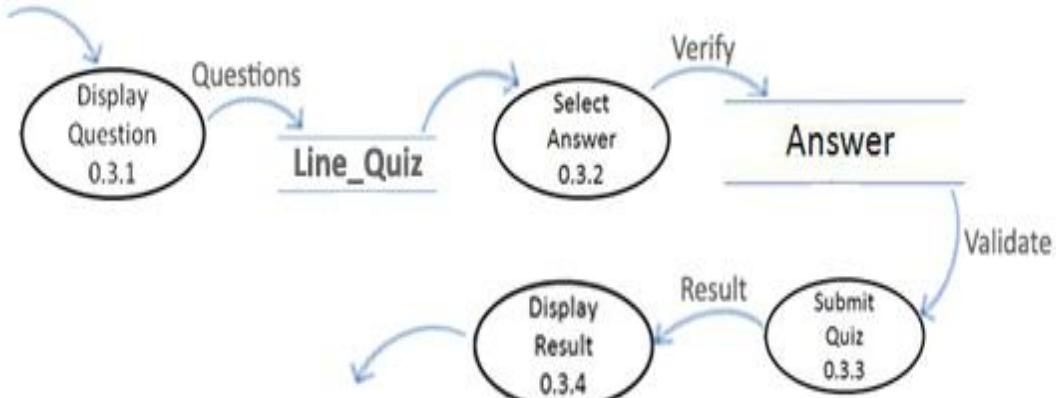
**Figure 5.5.2 - Level 1 DFD**



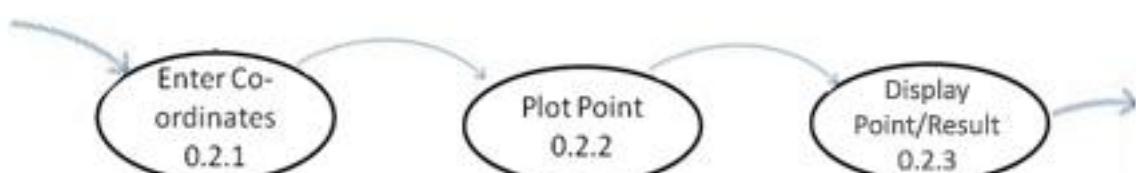
**Figure 5.5.3 - Level 1 DFD**



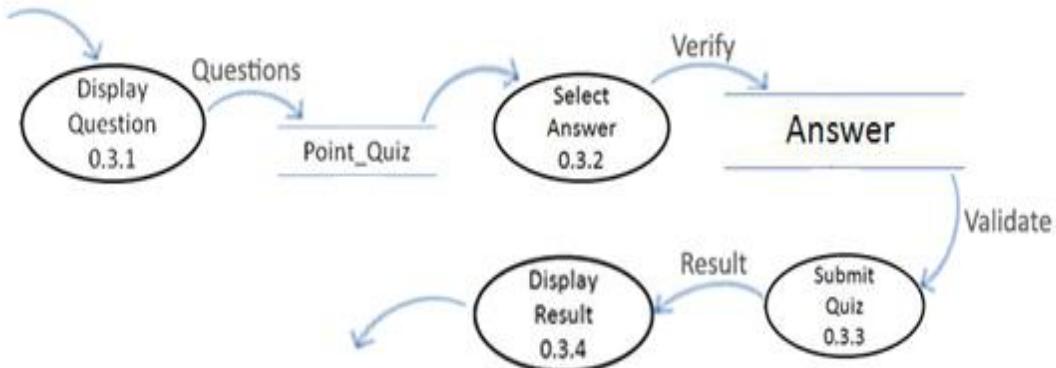
**Figure 5.6.1 - Level 2 DFD**



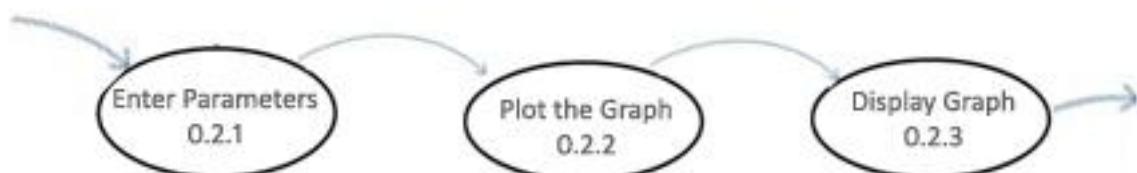
**Figure 5.6.2 - Level 2 DFD**



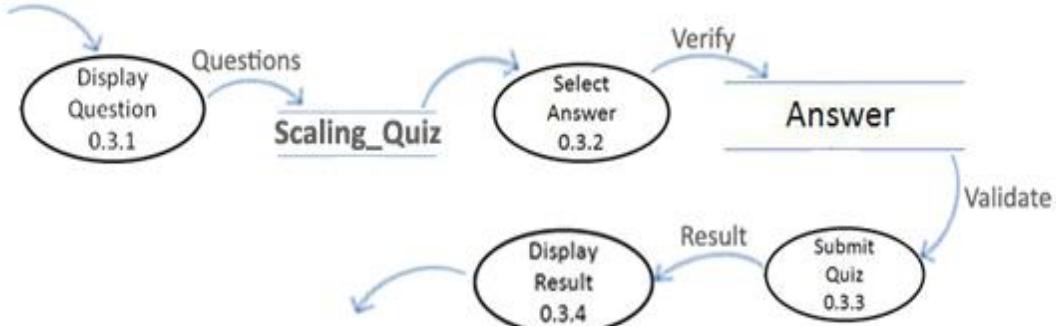
**Figure 5.6.3 - Level 2 DFD**



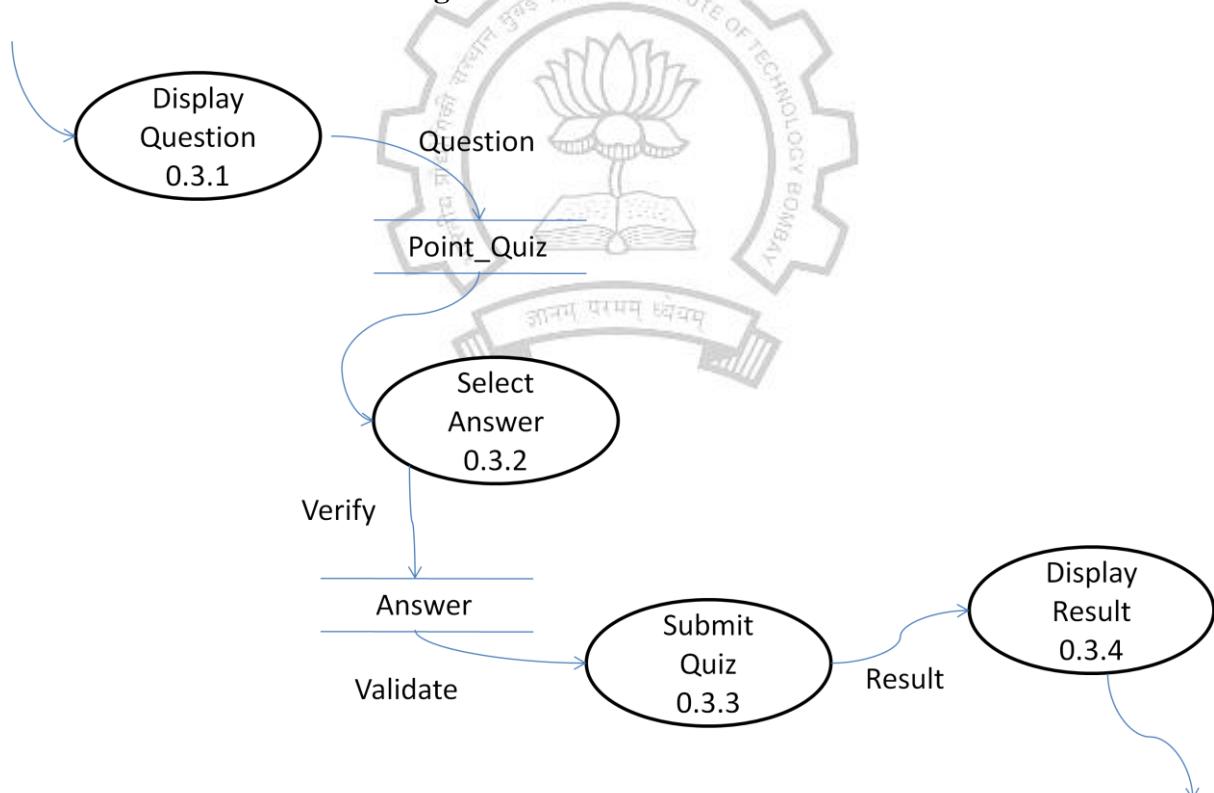
**Figure 5.6.4 - Level 2 DFD**



**Figure 5.6.5 - Level 2 DFD**

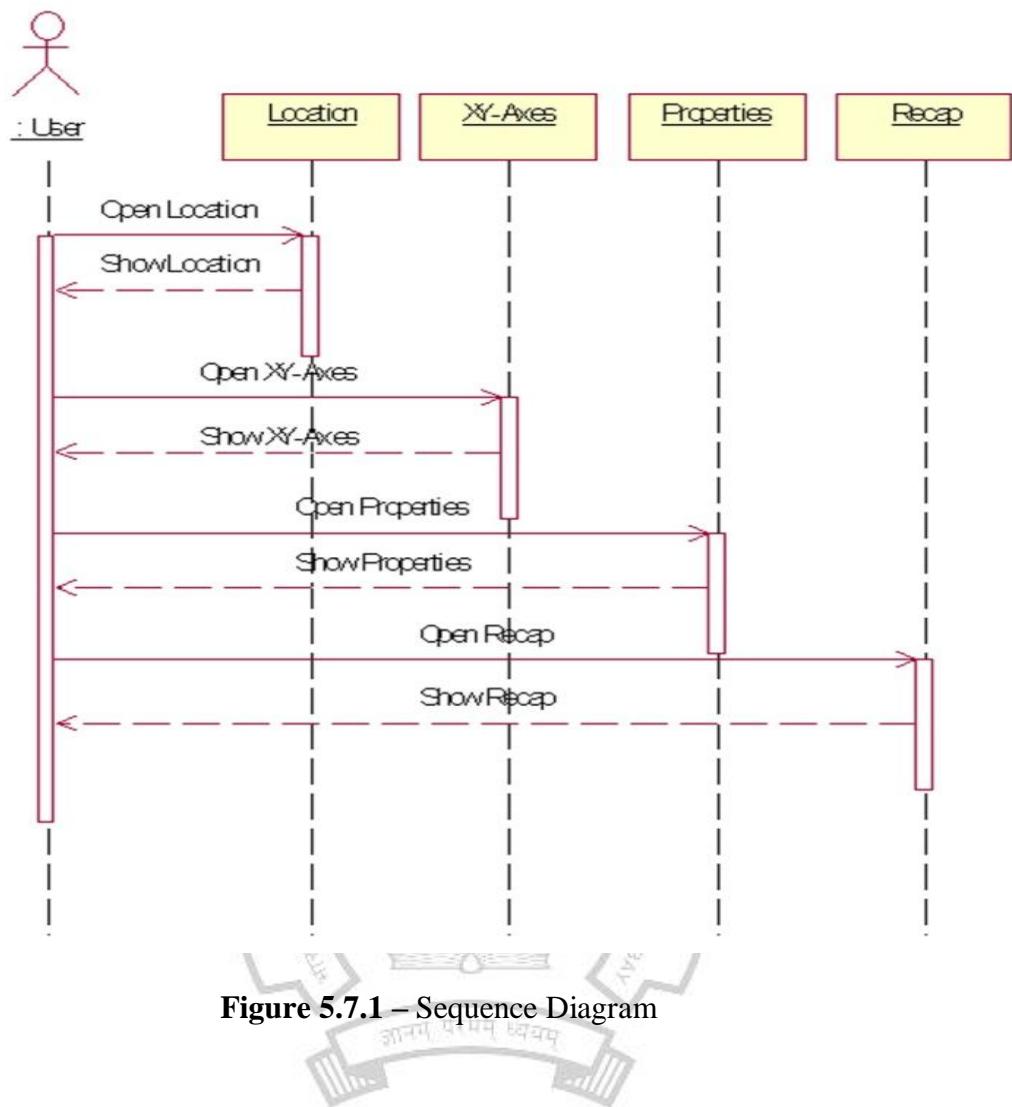


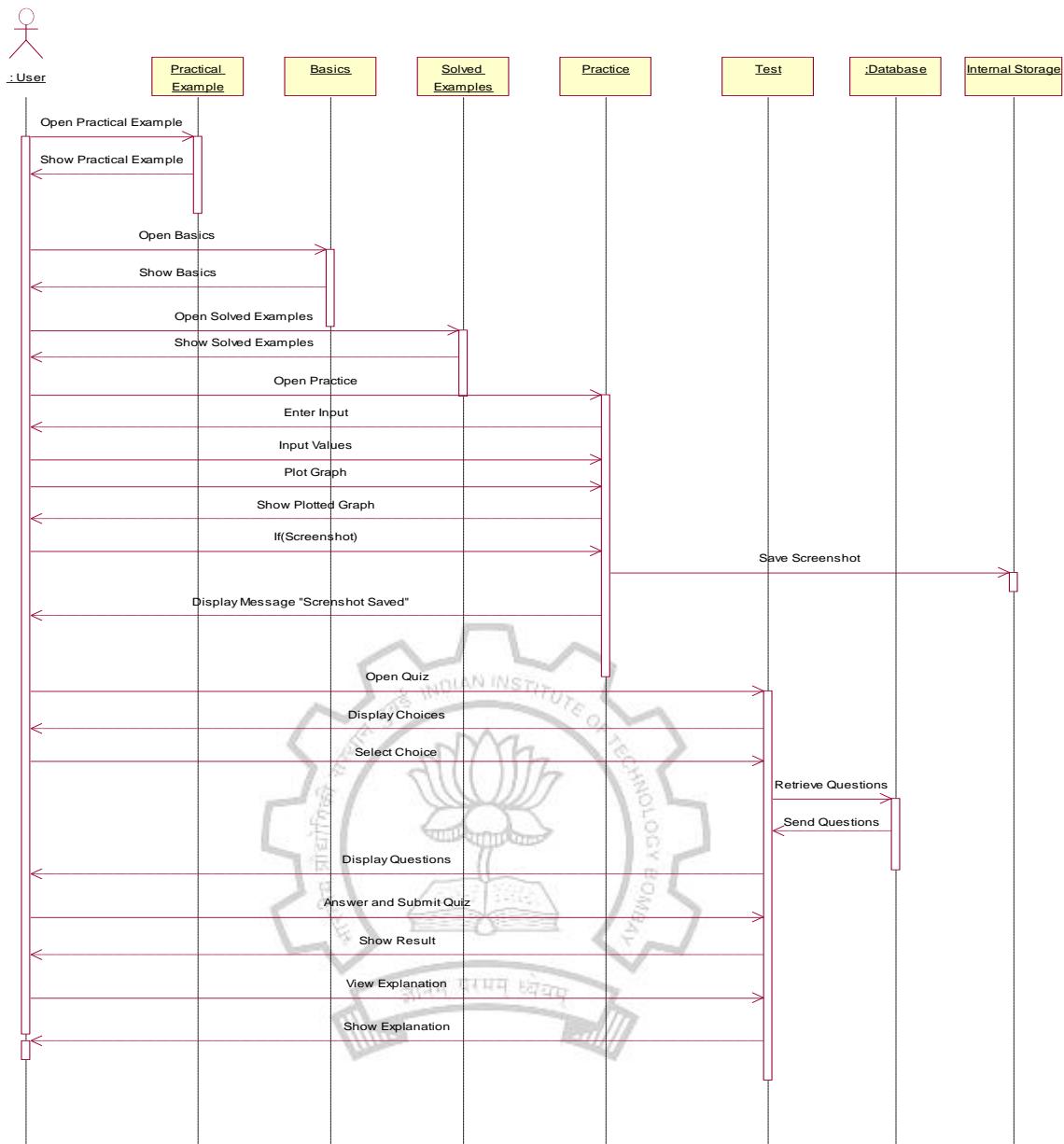
**Figure 5.6.6 - Level 2 DFD**



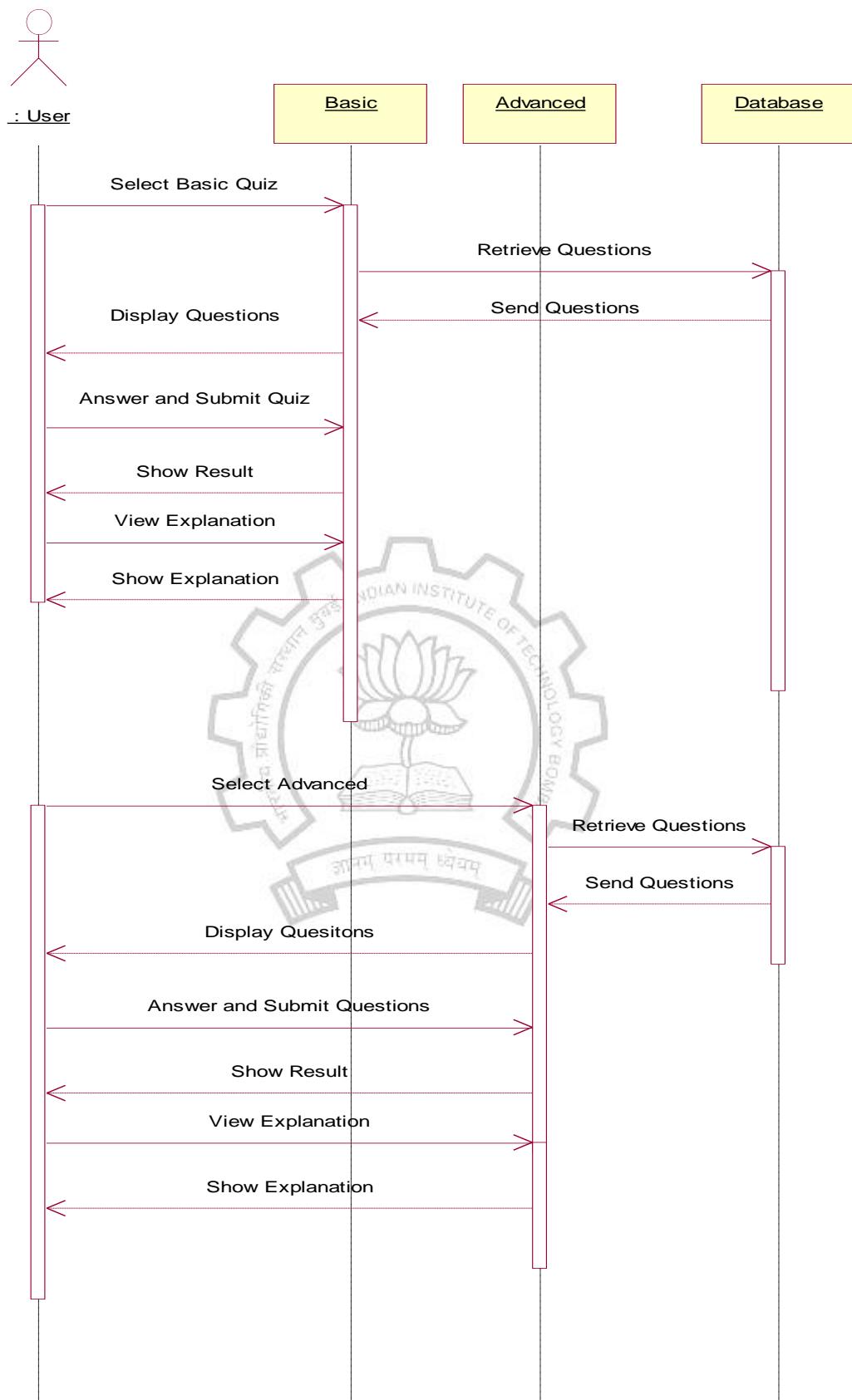
**Figure 5.6.7 - Level 2 DFD**

### 5.3.6 Sequence Diagram



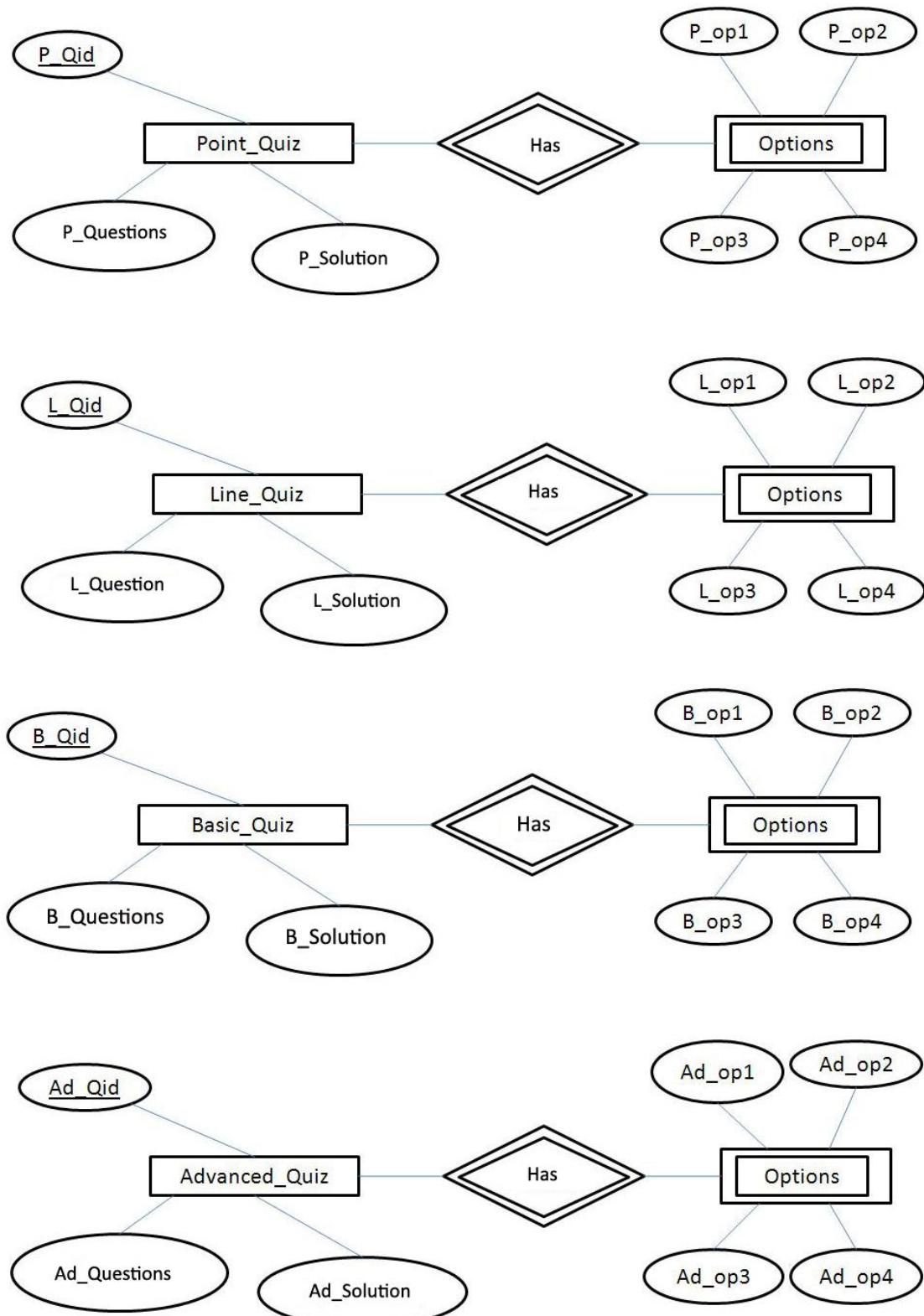


**Figure 5.7.2 – Sequence Diagram**



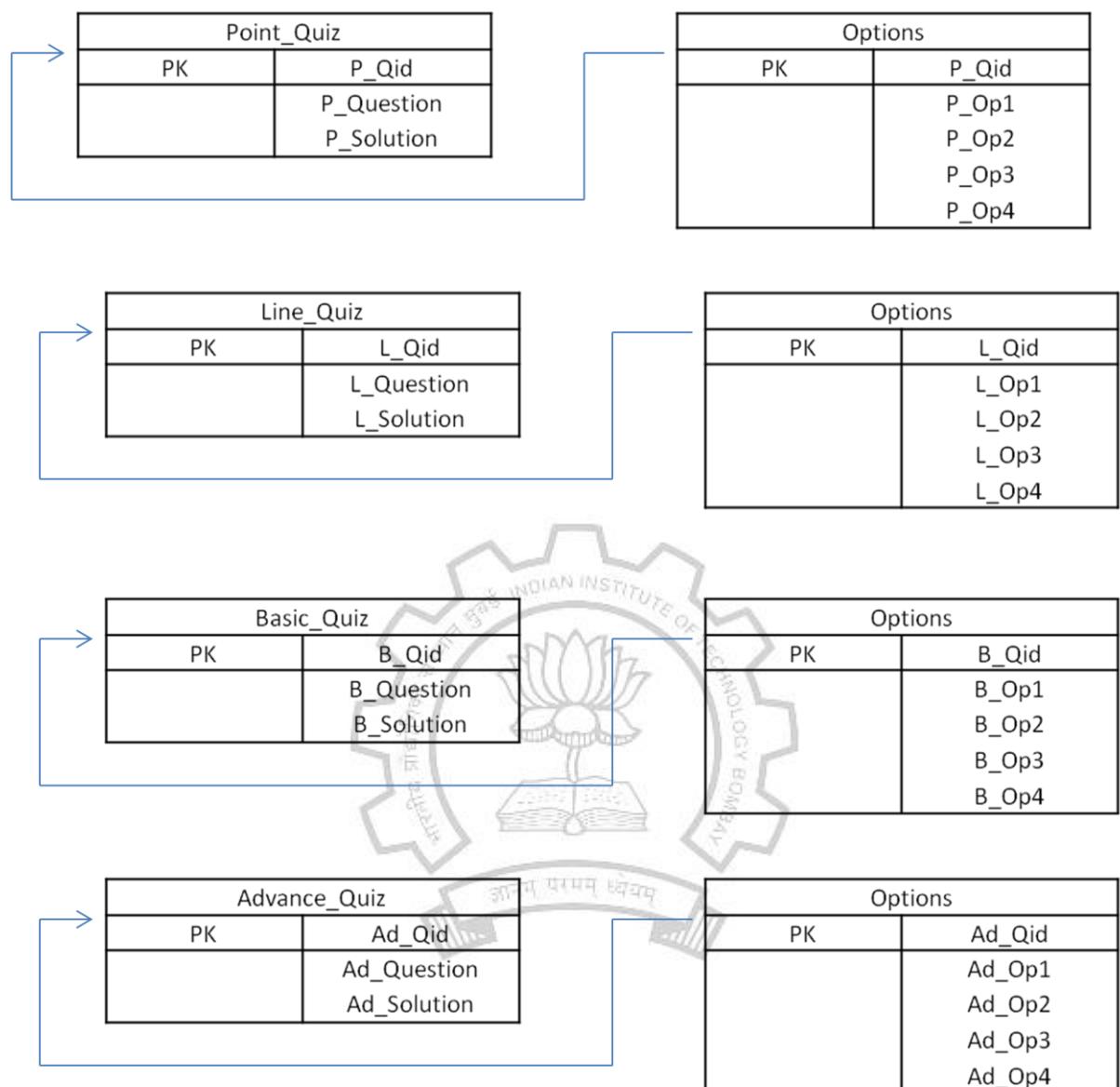
**Figure 5.7.3 – Sequence Diagram**

### 5.3.7 E-R Diagram



**Figure 5.8 – E-R Diagram**

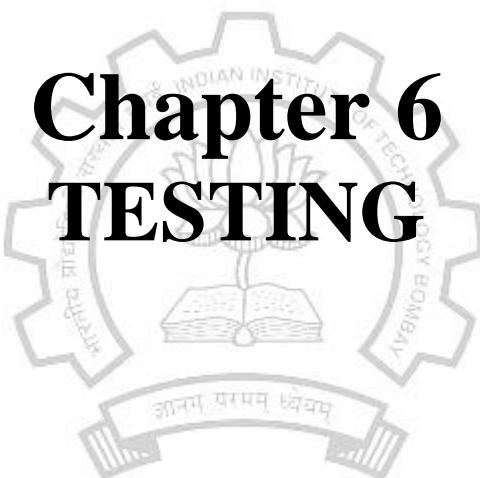
## 5.4 Database Schema



**Figure 5.9 – Database Schema Diagram**

# **Chapter 6**

## **TESTING**



## **6 TESTING**

Software testing is an investigation conducted to provide information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test technique includes the process of executing a program or application with the intent of finding software bugs.

Software testing can be stated as the process of verifying and validating that a software program/application/product:

1. meet the requirements that guide its design and development,
2. work as expected, and
3. can be deployed with the same characteristics.

Software testing can be implemented at any time depending on the testing method employed in the development process. However, most of the test effort traditionally occurs after the requirements have been defined and the coding process has been completed. It is observed that fixing a bug is less expensive if found earlier in the development process. Although in the agile approaches most of the test effort is, conversely, on-going. As such, the methodology of the test is governed by the software development methodology adopted.

### **6.1 Testing technique used**

Several testing techniques have been performed on the system such as unit testing, integration testing and system testing.

#### **6.1.1 Unit Testing:**

Unit testing, also known as component testing refers to tests that verify the functionality of a specific section of code, usually at the function level. Unit testing is a software development process that involves synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs.

#### **6.1.2 Integration Testing**

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Integration testing works to expose defects in the interfaces and interaction between integrated components (modules).

1. **Black-box testing-** It is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings.

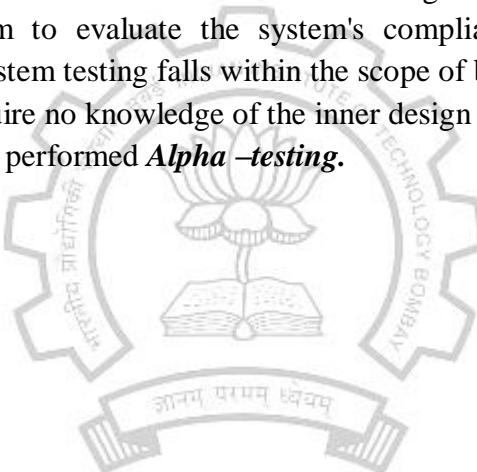
2. **White-box testing-** It is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.  
In this testing, we performed ***Boundary Value Analysis***.

### 6.1.3 System Testing

System testing tests a completely integrated system to verify that it meets its requirements. In addition, the software testing should ensure that the program, as well as working as expected, does not also destroy or partially corrupt its operating environment or cause other processes within that environment to become inoperative.

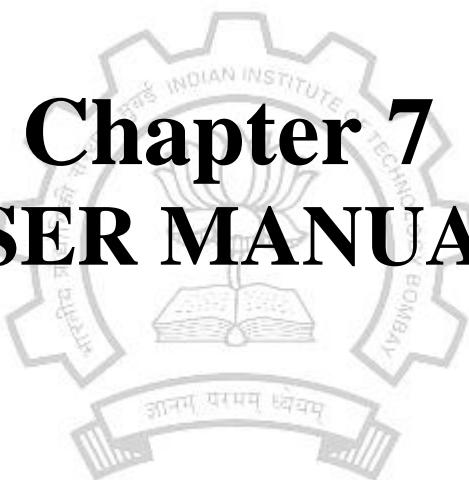
**System testing** of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

In this testing we performed ***Alpha -testing***.



# **Chapter 7**

# **USER MANUAL**



## **7 USER MANUAL**

**1.** Home Screen (Application Guide, About Us, Start myPoint).

**2.** C-Plane

- a.** Location
- b.** X-Y axes
- c.** Properties
- d.** Recap

**3.** Plotting

- a.** Practical Example
- b.** Point basics
- c.** Solved Example
- d.** Practice
- e.** Quiz-It

**4.** Scaling

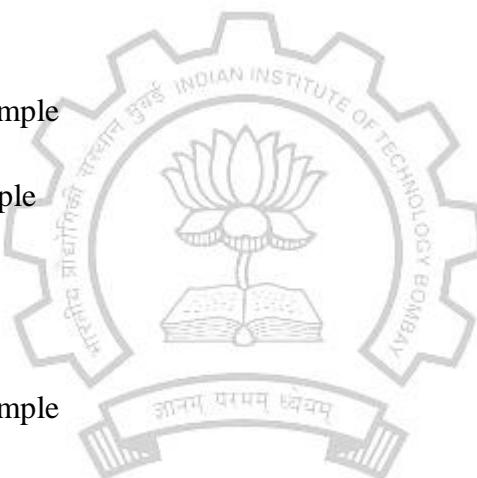
- a.** Practical Example
- b.** Scale basics
- c.** Solved Example
- d.** Practice
- e.** Quiz-It

**5.** Line

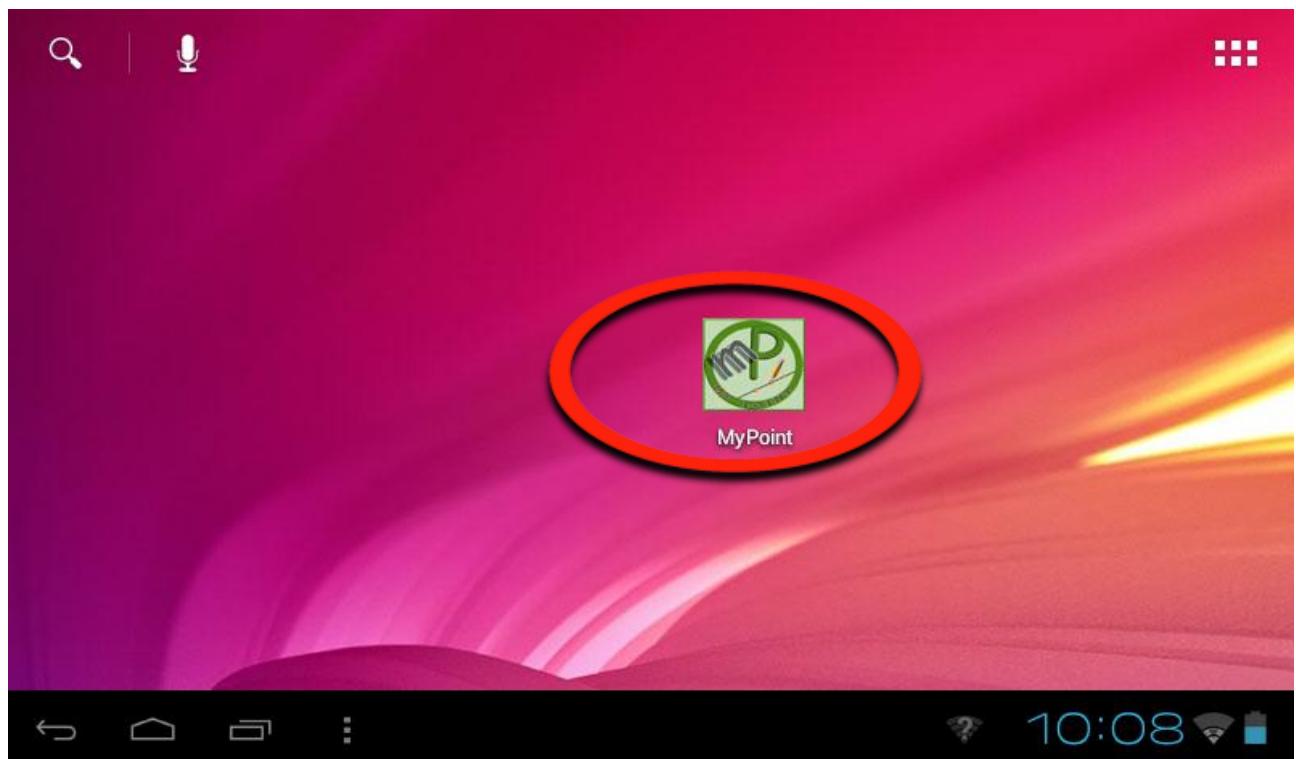
- a.** Practical Example
- b.** Line basics
- c.** Solved Example
- d.** Practice
- e.** Quiz-It

**6.** Quiz-it

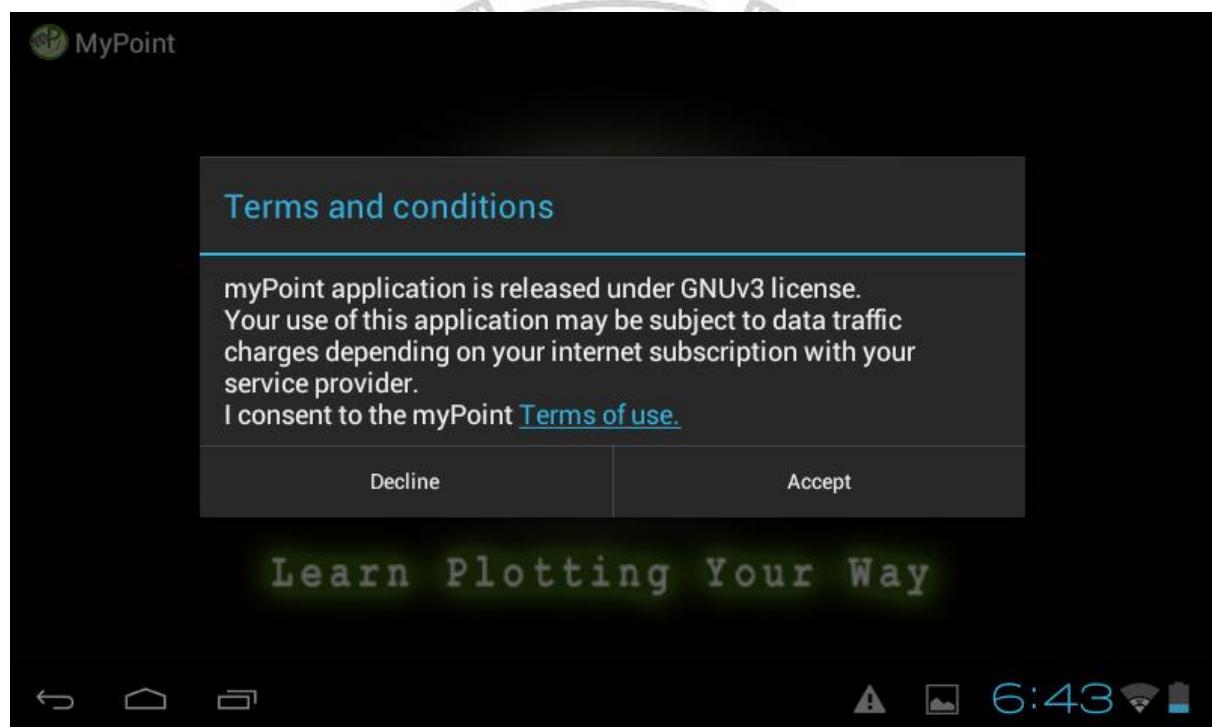
- a.** Basic
- b.** Advanced



1. Tap the Icon to Launch the app.



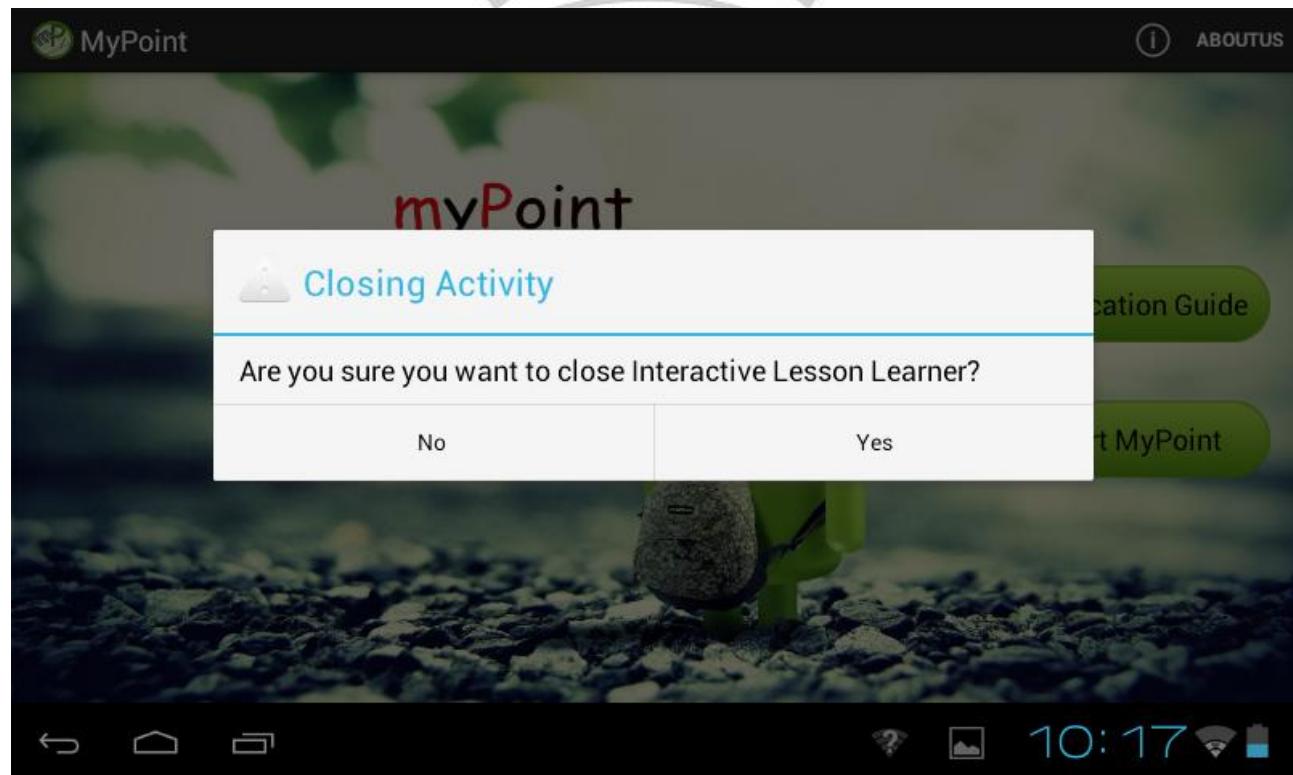
The screen below is visible only for the first launch of the application, Accept terms and conditions to open the application after going through Terms of use.



2. Home Screen consists of three buttons: Application Guide, About Us and Start MyPoint.

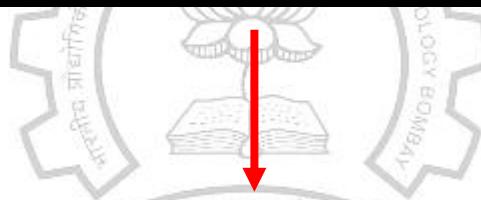


This screen shown below is visible while closing the application, Select yes to exit and no to remain inside the application.



Three modules of Home Screen:

- a. **About-Us:** Tap to open, it displays information about developer team.



The 'About Us' screen displays a family tree diagram. At the top is a portrait of a man, connected by lines to a woman and a man below him. From this middle couple, lines descend to four children: a boy, a girl, a boy, and a girl. Below the tree, the text 'Tap on image to view details' is centered. To the right is a circular logo for 'MY POINT' featuring a stylized 'MP' with a pencil and a graph line.

**About Us**

Tap on image to view details

Learn Plotting Your Way

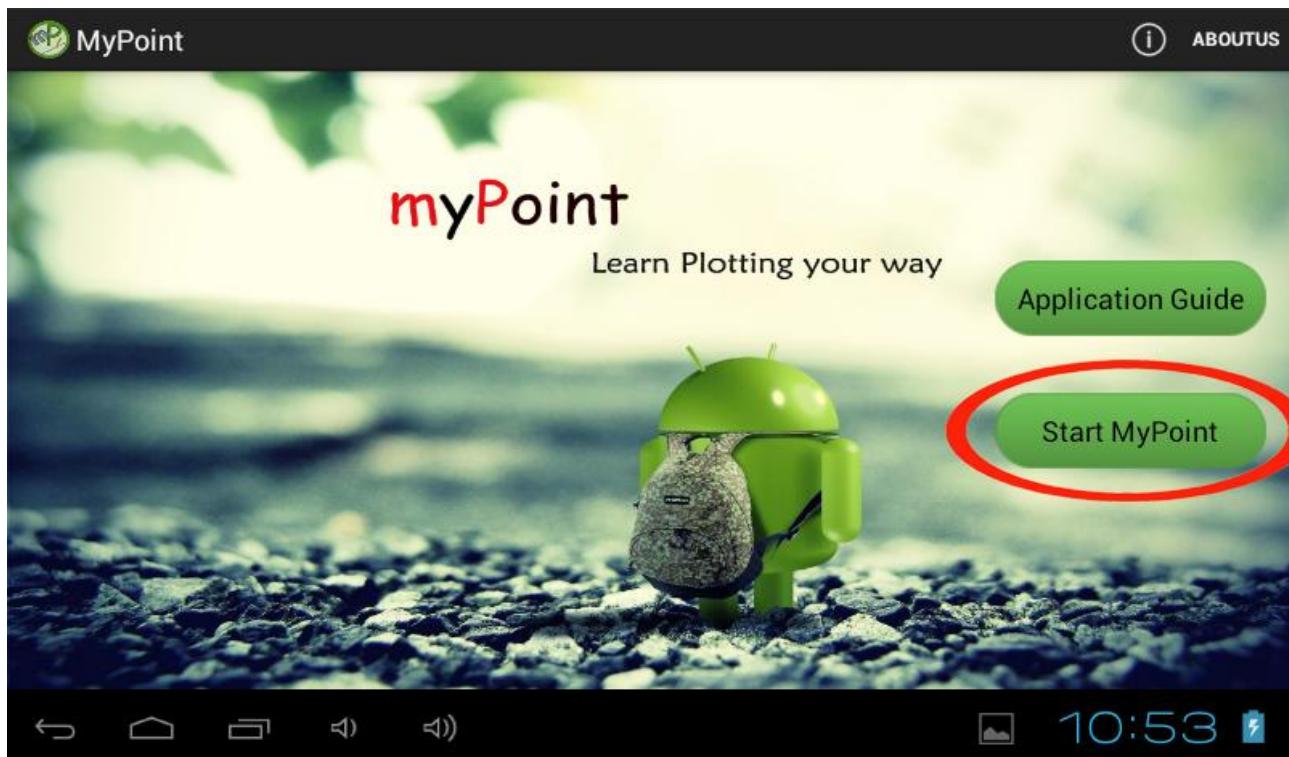


**b. Application Guide:** Tap to open. It is a complete guide on how to efficiently use MyPoint App.

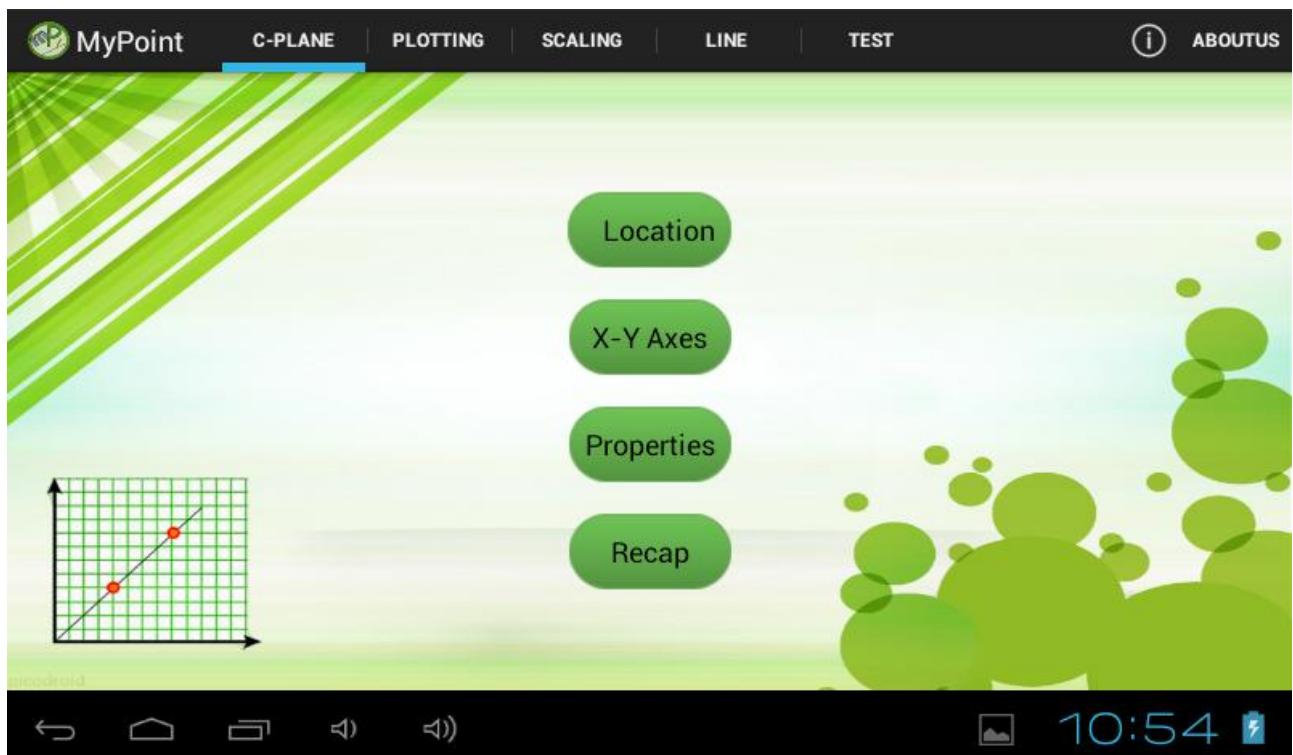


This image shows the 'Application Guide' screen of the MyPoint app. It features a title 'myPoint – Application Guide' at the top. Below it is a section titled 'Instructions To Use My Point App :'. The text explains that the app provides an efficient platform for learning, practising &amp; quizzing Cartesian Coordinate System. It then states that the app is categorized into 5 modules: 'C-plane', 'Plotting', 'Coordinate System', 'Quizzes', and 'Help'. A red arrow points from the text 'The app has been categorized into 5 modules named as:' down towards the list of modules. The bottom of the screen includes a 'Close' button and a clock reading 10:54.

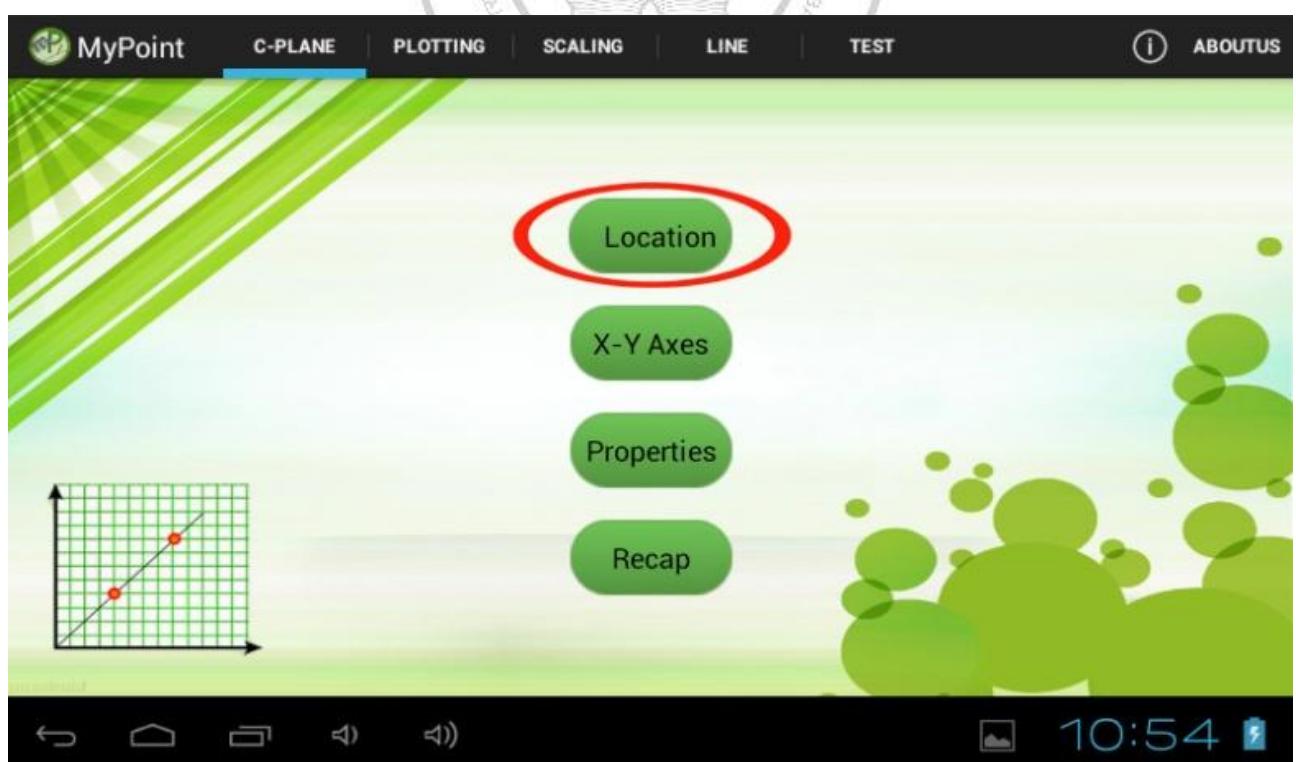
c. Start myPoint: Tap to enter Main Application.



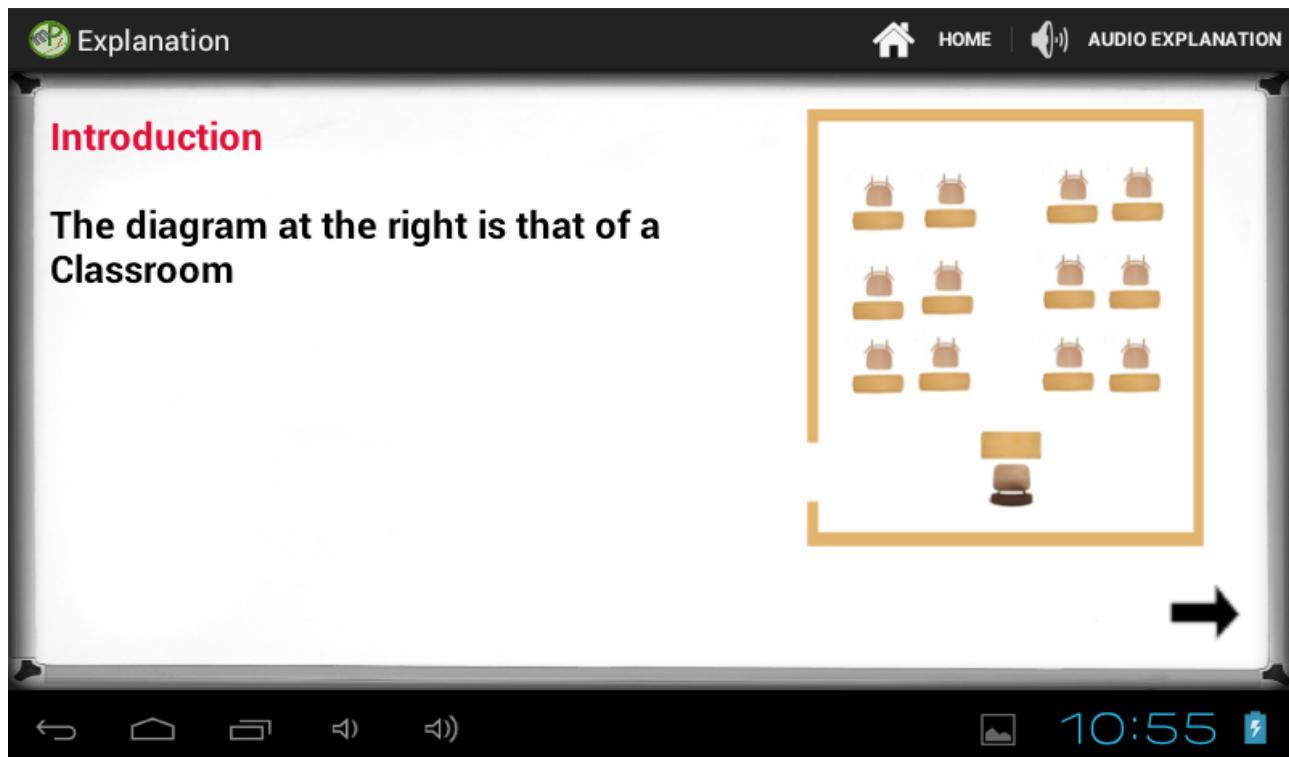
1. **C-Plane** module consists of four sub-modules: Location, X-Y Axes, Properties and Recap.



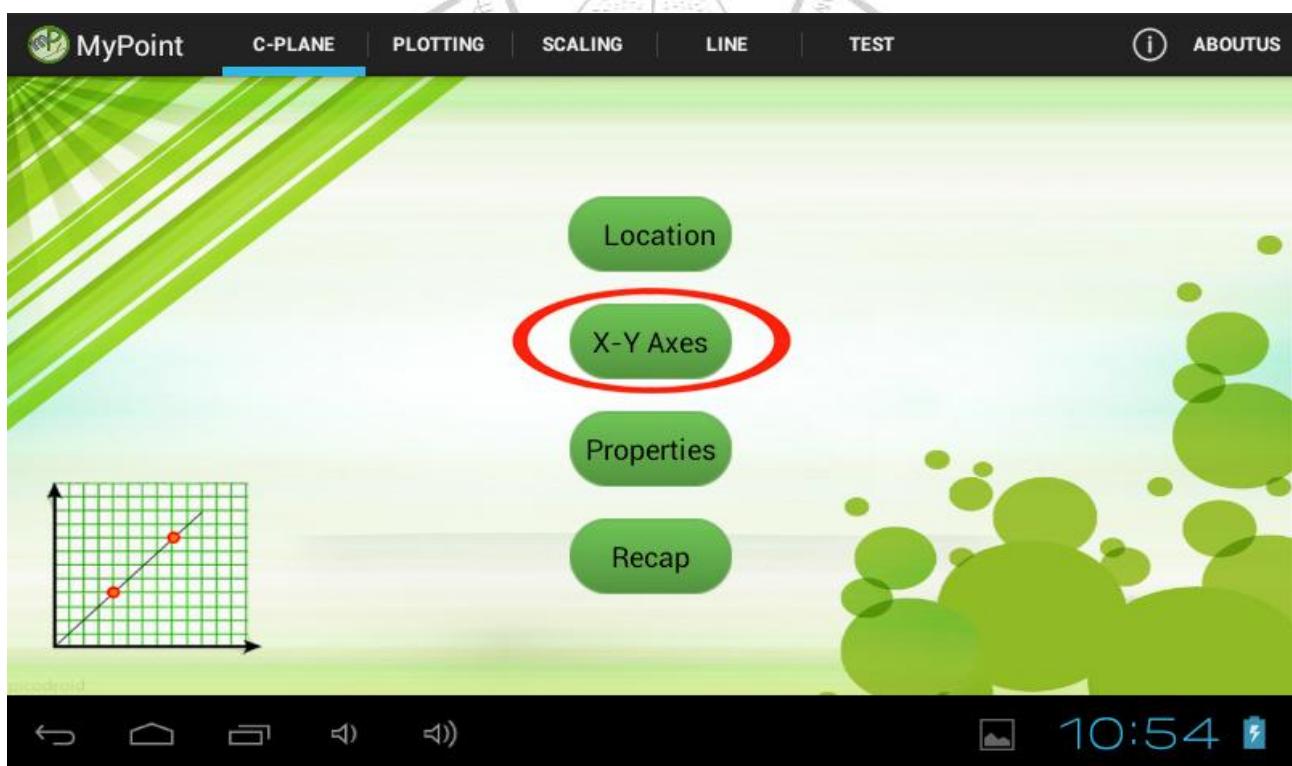
**Location:** Tap to open. It explains practical examples relative to Cartesian plane.



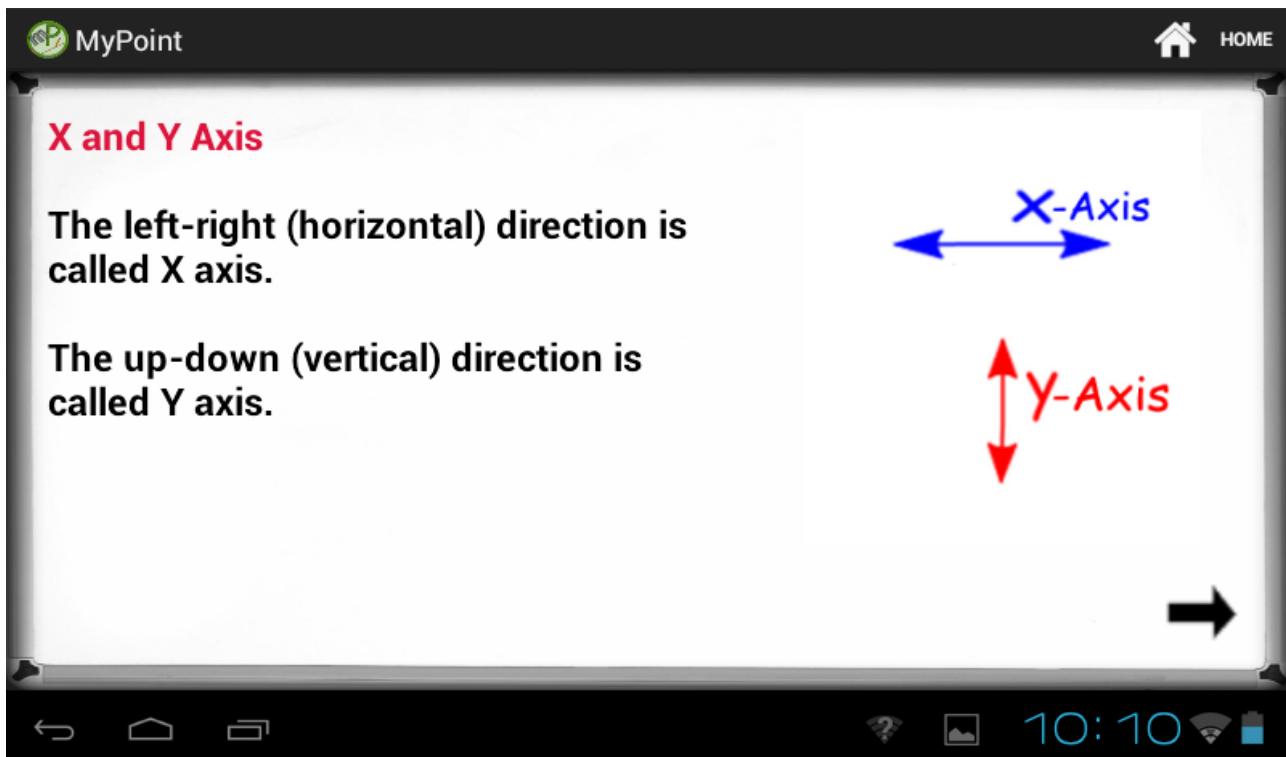
Use arrow button shown at the bottom right of the screen below to change screens. Tap on Home menu icon to return to c-plane module. Tap on Audio menu explanation for Audio narration of respective screen.



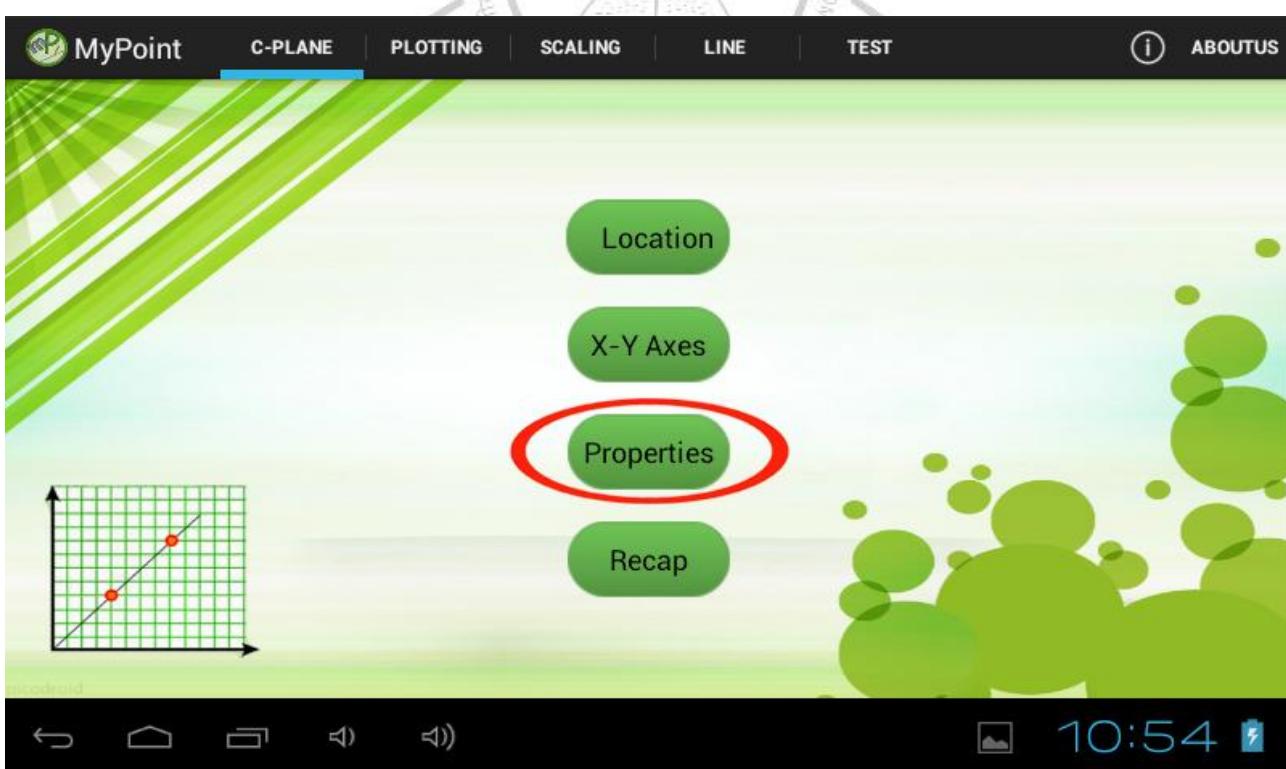
**X-Y Axes:** Tap to open X-Y axes explanation.



Use arrow button shown at the bottom right of the screen below to change screens. Tap on Home menu icon to return to c-plane module.



**Properties:** Tap to open Cartesian plane properties explanation.



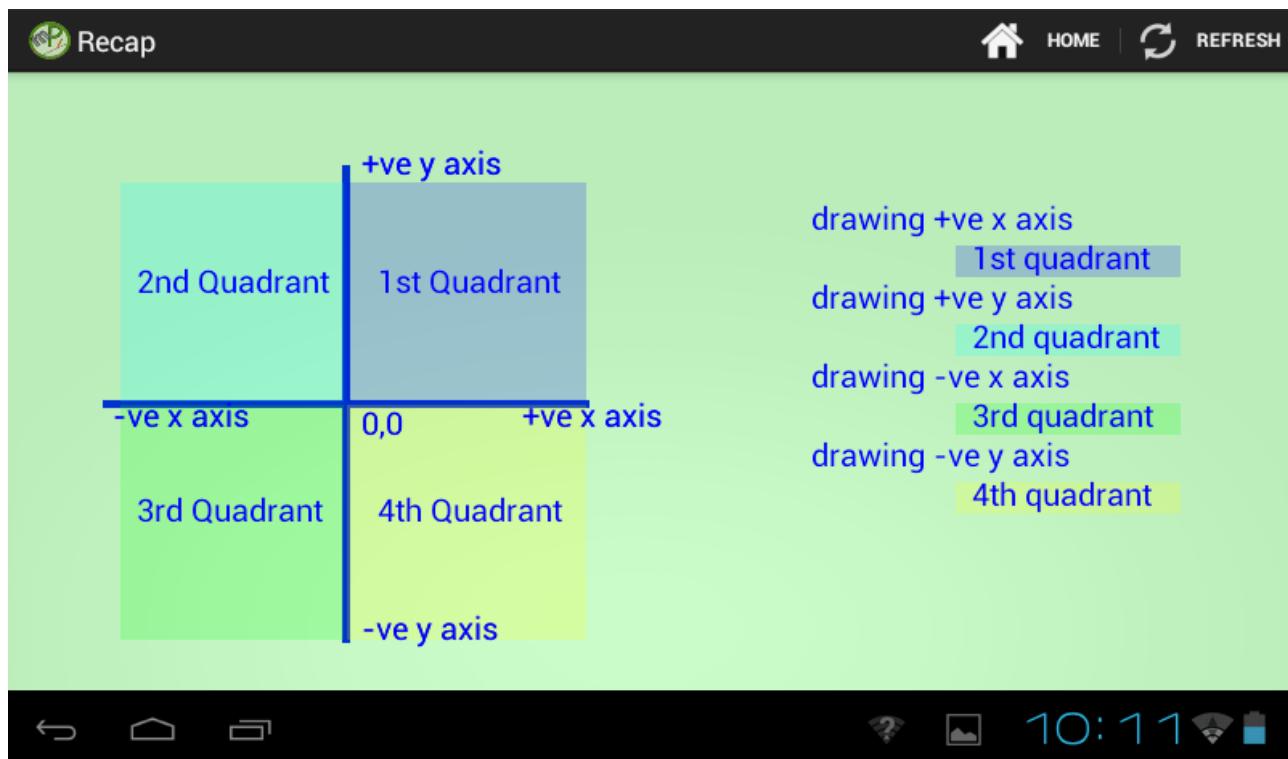
Use arrow button shown at the bottom right of the screen below to change screens. Tap on Home menu icon to return to c-plane module.

The screenshot shows a mobile application interface. At the top, there is a navigation bar with a logo, the text "MyPoint", and a "HOME" button. Below this, the main content area has a title "Quadrants of a Graph" in red. To the right of the title is a diagram of a Cartesian coordinate system divided into four quadrants. The quadrants are labeled: I Quadrant (+ve x axis, +ve y axis), II Quadrant (+ve y axis, -ve x axis), III Quadrant (-ve x axis, -ve y axis), and IV Quadrant (+ve x axis, -ve y axis). The axes are labeled X and Y. A large black arrow points to the right at the bottom right corner of the content area. The bottom of the screen features a standard Android-style navigation bar with icons for back, home, and recent apps, along with a battery and signal status indicator. The time is displayed as 10:10.

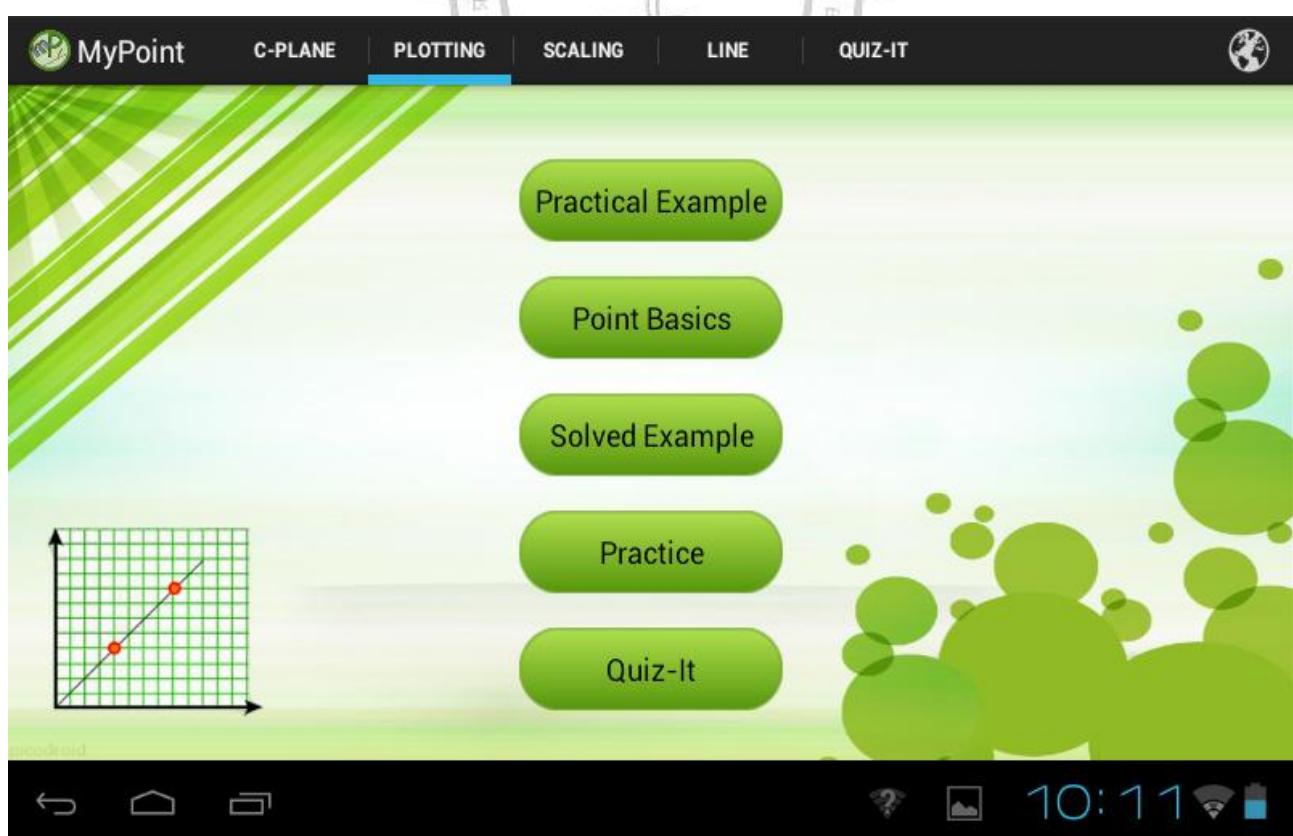
**Recap:** Tap on Recap button to revise the concepts studied in c-plane module.

This screenshot shows the "C-PLANE" tab selected in the MyPoint app. On the left, there is a small grid diagram with two red dots connected by a line. To the right of the grid are four green rounded rectangular buttons with white text: "Location", "X-Y Axes", "Properties", and "Recap". The "Recap" button is circled in red. The background has a green and white abstract design with bubbles. The bottom navigation bar includes icons for back, home, and recent apps, along with a battery and signal status indicator. The time is displayed as 10:54.

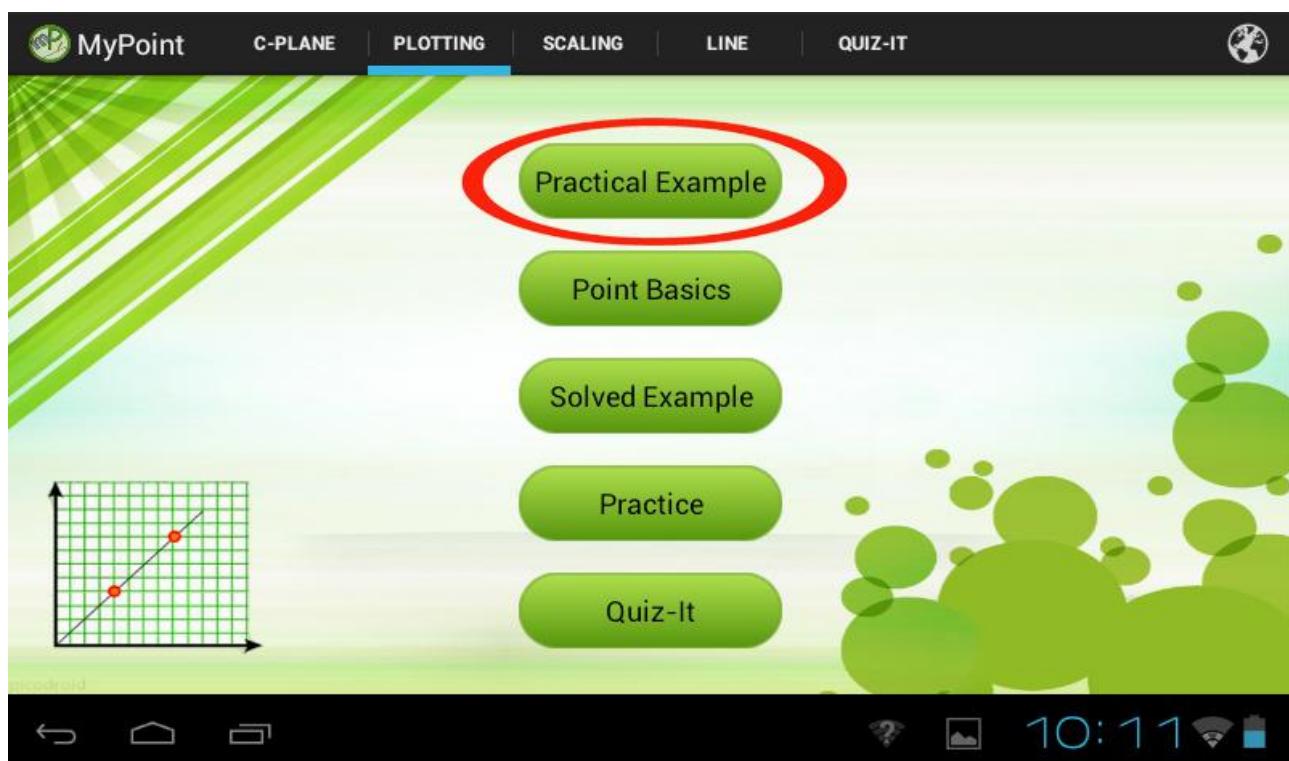
Animated recap being displayed.



2. Plotting consists of five modules: Practical Example, Point basics, Solved Example, Practice and Quiz-It.



**Practical Example:** Tap on button to start reading, learning, analysing real life examples.



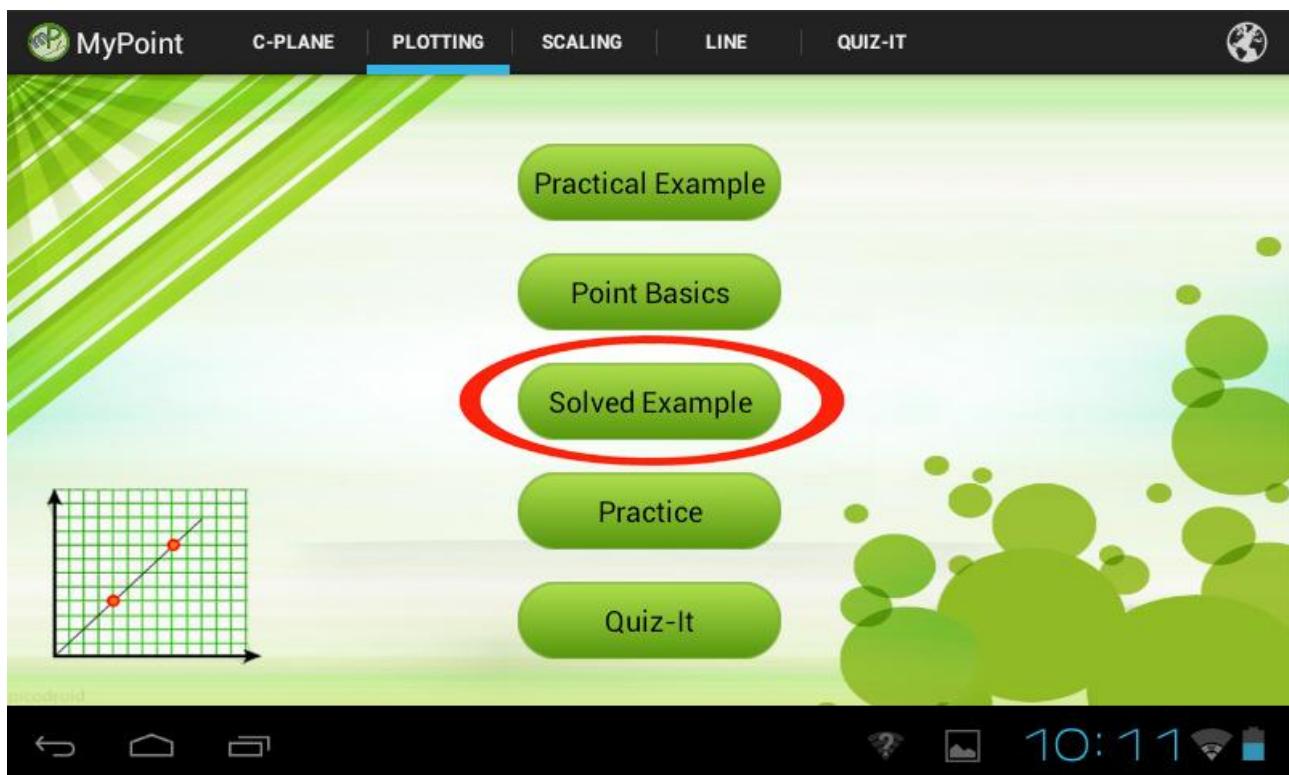
The screenshot displays the 'Practical Example' content page. The background is a blackboard illustration. In the center, there is a red arrow pointing to a small red dot labeled 'P'. To the left of the arrow, there is text: 'The teacher puts a dot "P" on blackboard. and asked the students,"How would you describe the location of the point?"'. Below this, another text block says: 'Raiza said,"The dot is very close to the right upper corner of the board"'. At the bottom of the screen, there is a black navigation bar with icons for back, home, and recent apps, and a message: 'Saving screenshot... Screenshot is being saved.'

**Point Basics:** Tap on button to start reading, learning point basics.



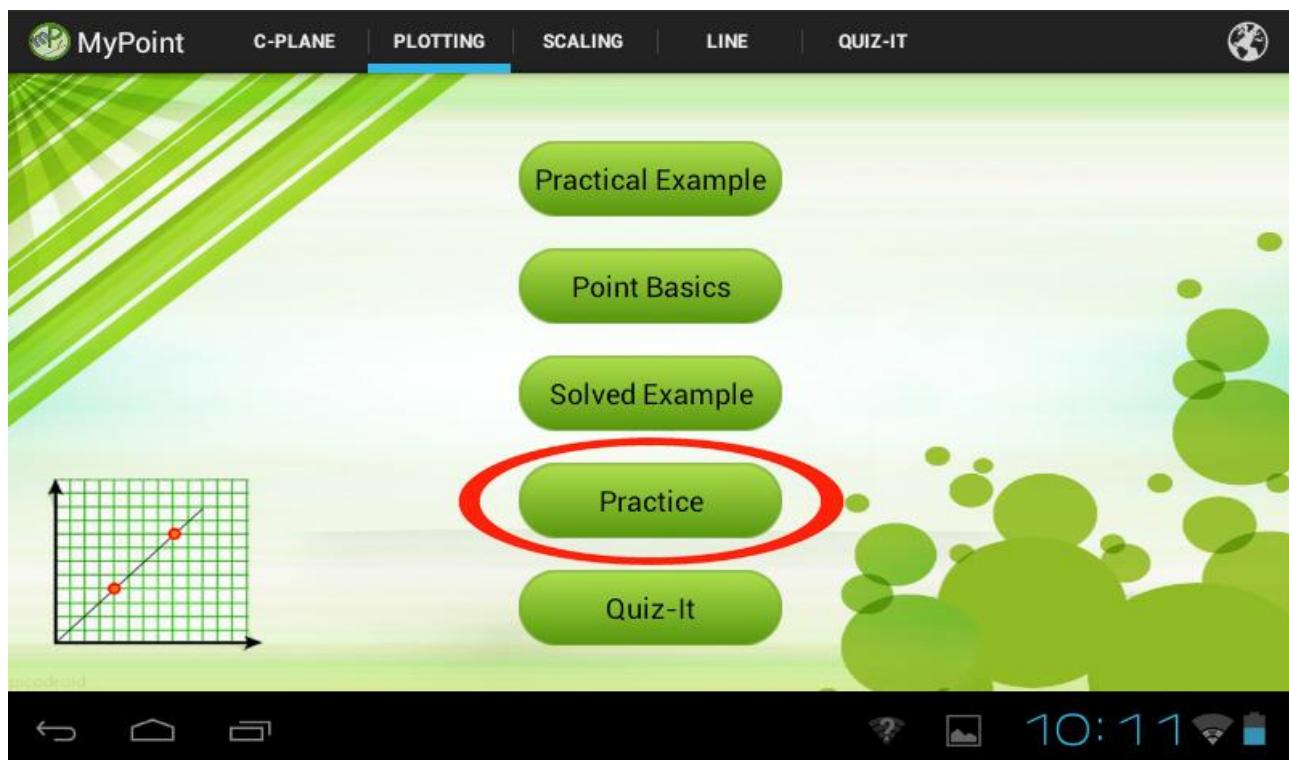
The screenshot shows a page titled 'Coordinates of Points' in red text. The main text on the left is: 'A location of a point on a graph always represented by these 'ordered pairs' which are called Coordinates. Thus coordinates is one set of numbers define to identify the location of a point on a graph.' To the right of the text is a dark gray box with the title 'ORDERED PAIRS' in white. Inside the box, there is a coordinate system with a grid. Two points are plotted: one at (2, 2) and another at (4, 7). The text '(2, 2)' is written above the first point, and '(4, 7)' is written below the second point. At the bottom right of the page is a large black arrow pointing right. The bottom navigation bar is identical to the one in the previous screenshot, showing the same icons and time.

**Solved Examples:** Tap on button to start reading, learning solved examples.

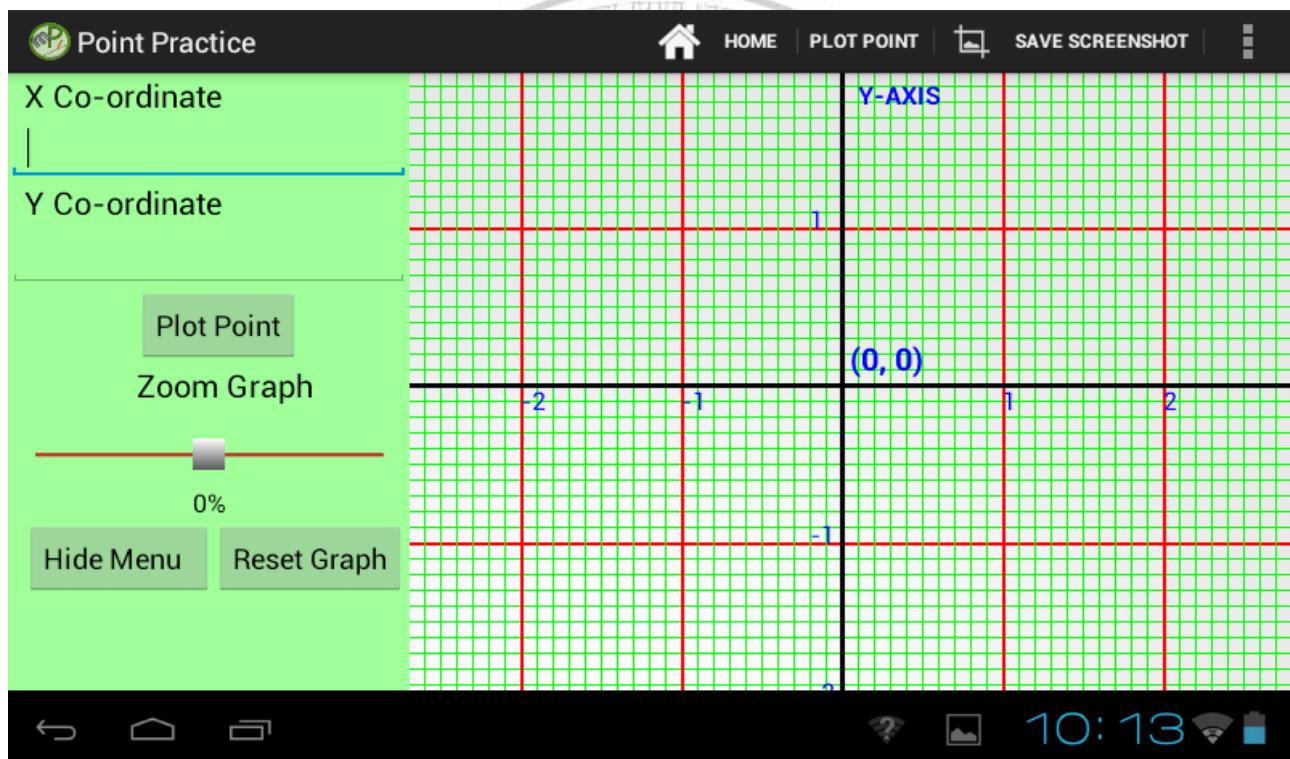


The screenshot displays a solved example from the MyPoint app. At the top, it says 'Ex. In which quadrant or on which axis do the following points lie ?'. Below this, there is a list of ten coordinate pairs: (i) (5,3), (ii) (-2,4), (iii) (-3,-6), (iv) (2,-5), (v) (0,4), (vi) (-4,0), (vii) (1/2,-3), (viii) (-2,5/2), (ix) (-7/2,-5/2), and (x) (7,4). In the bottom right corner of the main content area, there is an illustration of a black pen resting on a white notepad. At the very bottom of the screen, there is a black navigation bar with icons for back, home, and recent apps, along with a digital clock showing 10:13 and some signal strength indicators.

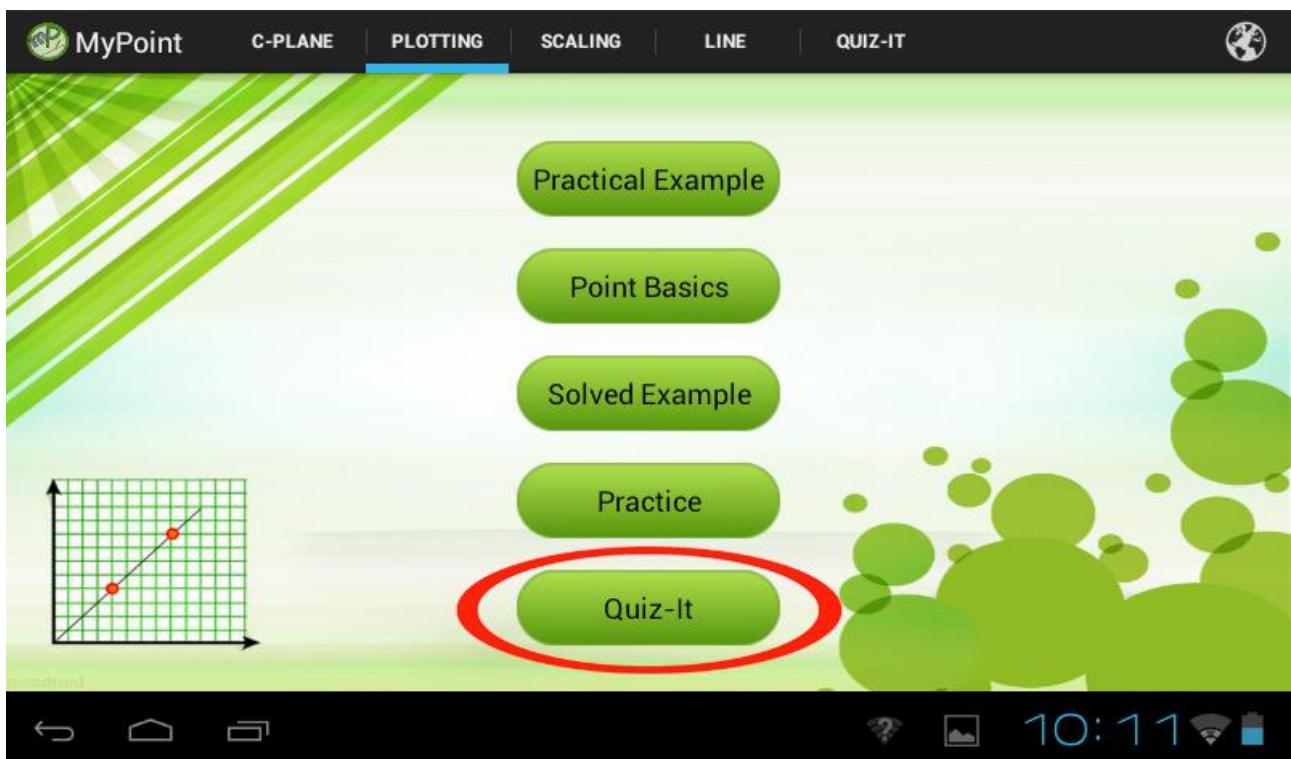
**Practice:** Tap on button to start practicing different ways of plotting a point on graph.



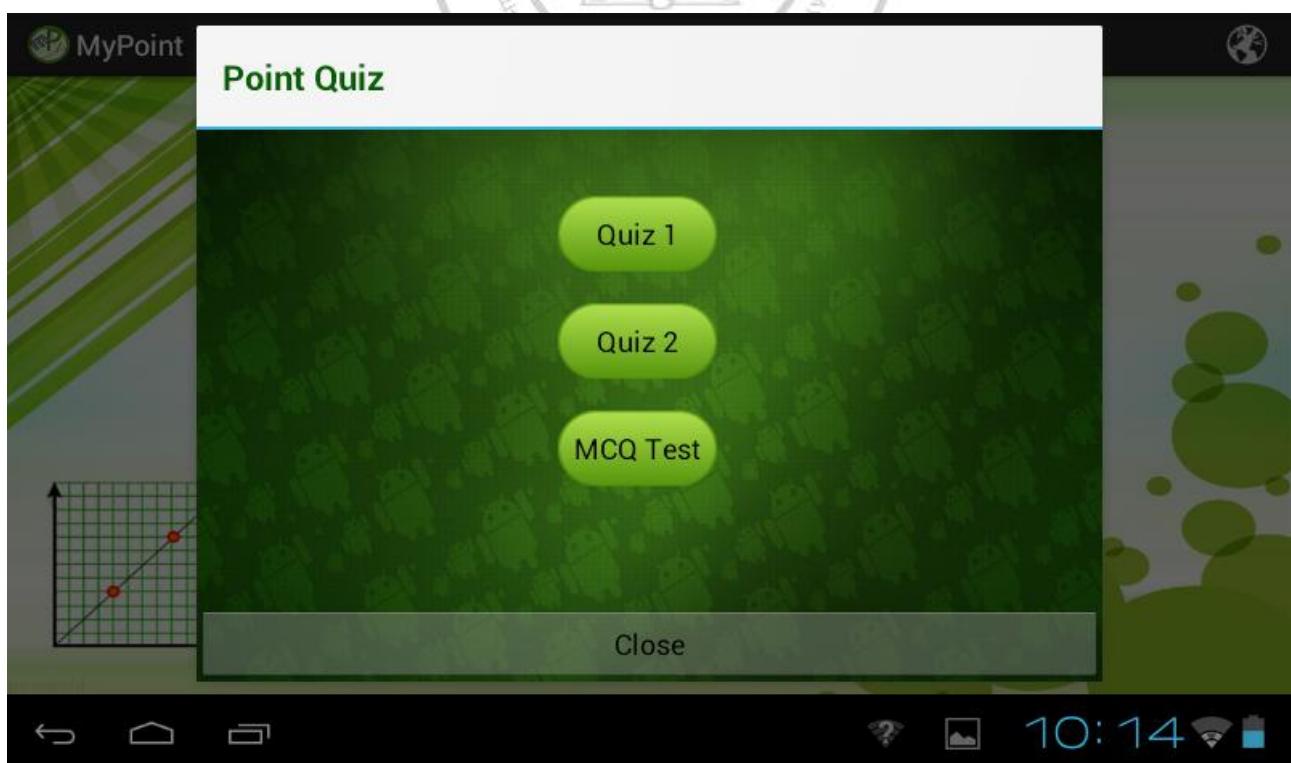
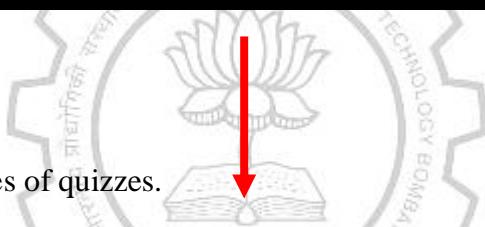
Enter x, y coordinates & plot point. Also can tap on screen to know a particular point's coordinates.



**Quiz-it:** Test module. Tap to open.

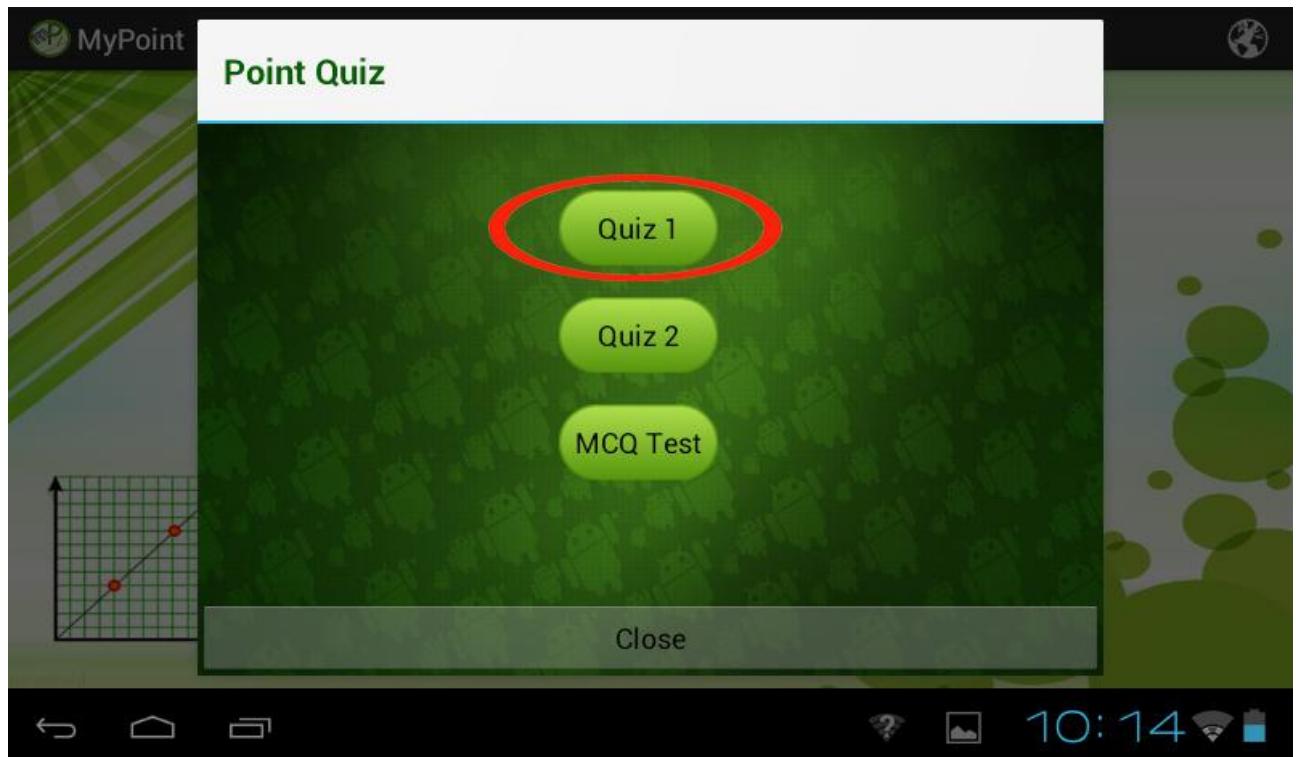


Choose one of the three types of quizzes.



## Different types of quiz:

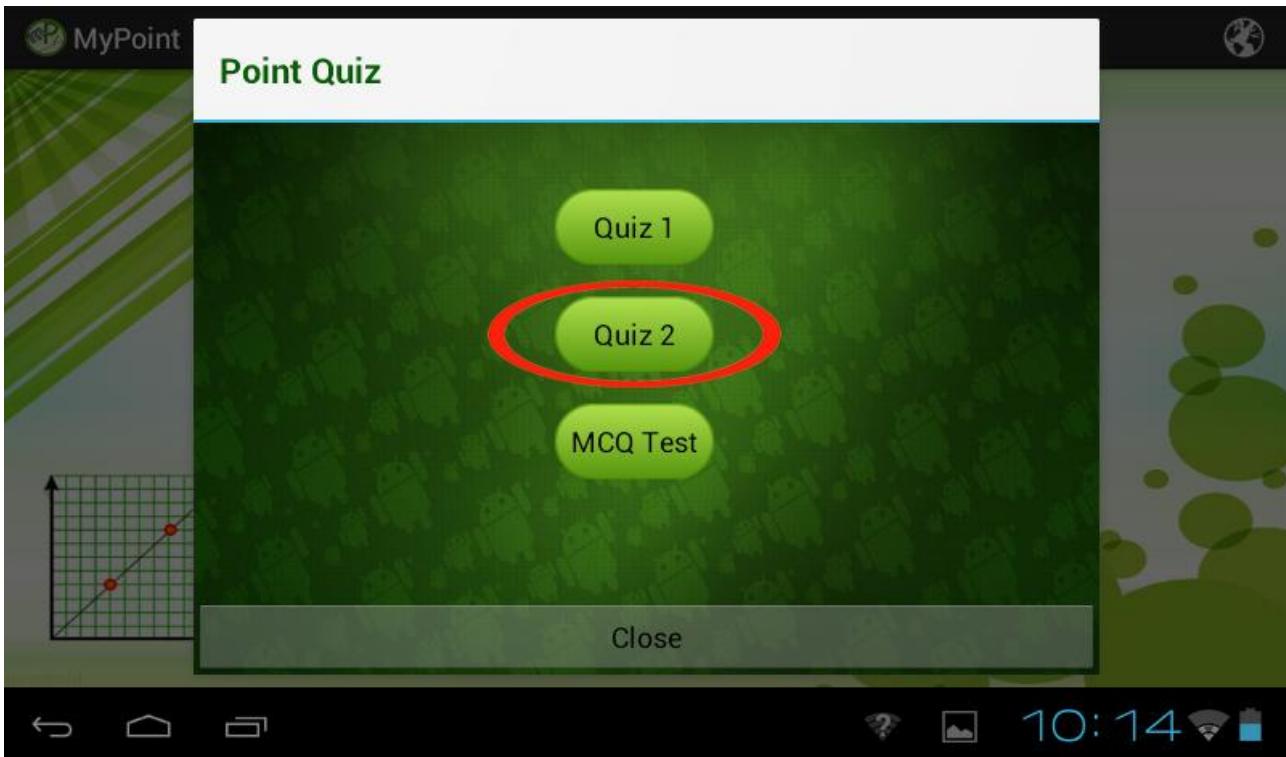
- 1st type



Tap on the point coordinates asked in the question. Use scratchpad for any calculations.

- 2nd type

A screenshot of a quiz interface titled "Quiz". The main area features a Cartesian coordinate system with a grid. The X-axis is labeled "X-axis" and has tick marks at -2, -1, 1, and 2. The Y-axis is labeled "Y-axis" and has tick marks at -1, 0, and 1. The origin is marked with "(0,0)". To the right of the grid, there is text providing instructions and parameters:  
Total Time: 2 sec  
SCALE:  
X axis : 1 cm = 1 unit  
X axis : 1 cm = 1 unit  
you are on question no. 1  
Locate the point from given coordinates  
X : 0  
Y : 0.4  
At the bottom right are "NEXT" and "SUBMIT" buttons. A progress bar at the bottom indicates "Saving screenshot... Screenshot is being saved." The status bar at the very bottom shows navigation icons.



Give the coordinates of the point asked in the question. Use scratchpad for any calculations.

Click on next for new question. It generates new random question each time.

Quiz      HOME | SCRATCH PAD

Total Time: 2 sec  
SCALE:  
X axis : 1 cm = 1 unit  
X axis : 1 cm = 1 unit  
you are on question no. 1  
Locate the point from given coordinates  
X : 0  
Y : 0.4

**NEXT**      **SUBMIT**

←    ⌂    ⌃      Saving screenshot...  
Screenshot is being saved.

'Submit' the test to finish & view result.

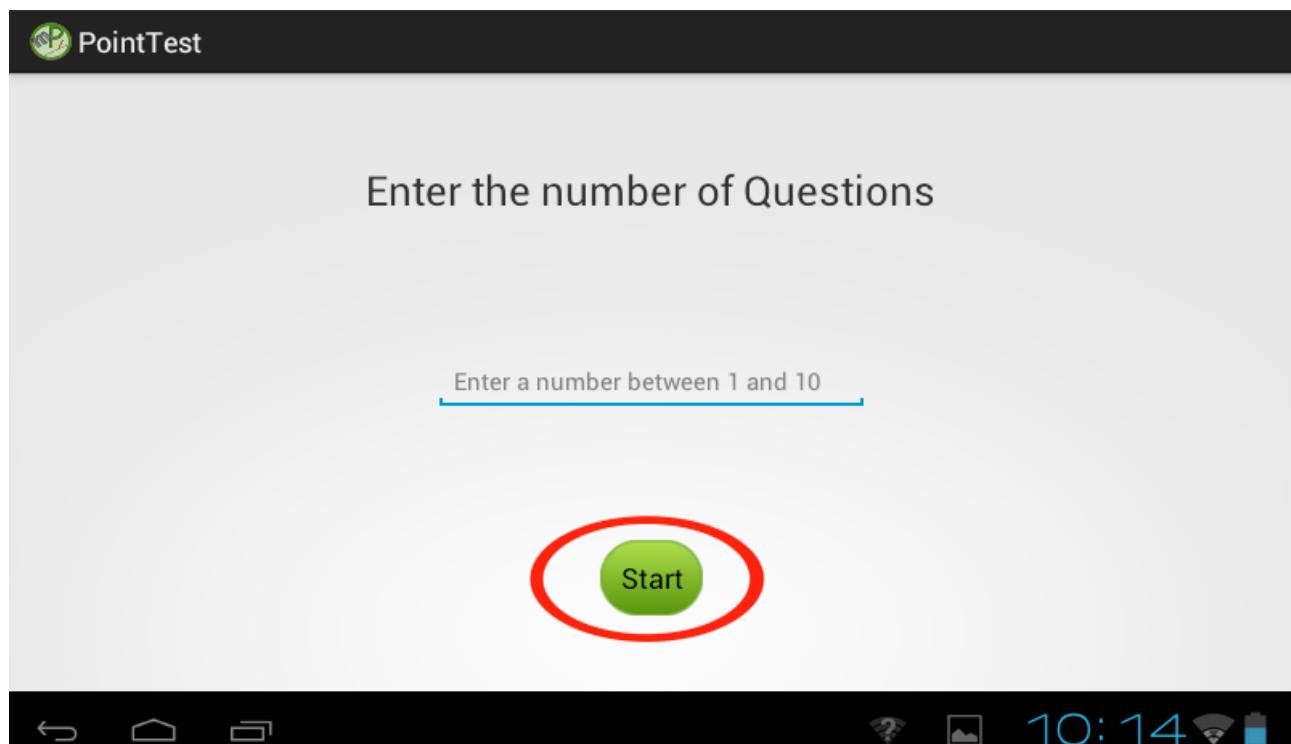
Quiz      HOME | SCRATCH PAD

Total Time: 2 sec  
SCALE:  
X axis : 1 cm = 1 unit  
X axis : 1 cm = 1 unit  
you are on question no. 1  
Locate the point from given coordinates  
X : 0  
Y : 0.4

**NEXT**      **SUBMIT**

←    ⌂    ⌃      Saving screenshot...  
Screenshot is being saved.

Enter desired number of questions to take a quiz.



Tap on question list at left to open them & answer.

A screenshot of the "PointTest" application showing a list of questions on the left and a detailed question on the right. The list includes: Question1, Question2, Question3, Question4, Question5, Question6, Question7, and Question8. The question details are as follows:

Q1.  
What are the Cartesian coordinates of the point A?

A diagram shows a Cartesian coordinate system with a grid. The x-axis is labeled "x" and the y-axis is labeled "y". Point A is plotted in the first quadrant at the coordinates (4, 2).  
A red arrow points from the text "Tap on question list at left to open them & answer." down to the "SUBMIT" button.

Question1

Question2

Question3

Question4

Question5

Question6

Question7

Question8

Q1.  
What are the Cartesian coordinates of the point A?

(4,2)

(2,4)

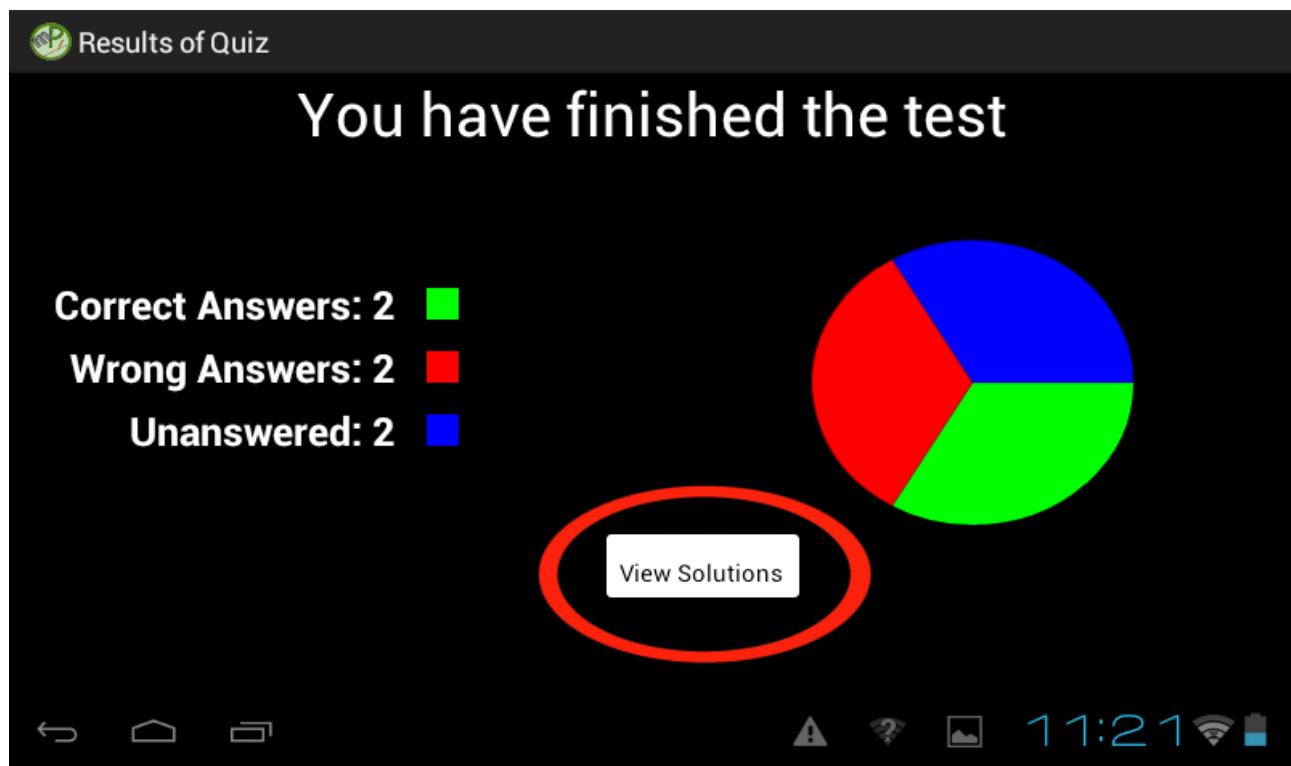
(-4,2)

(4,-2)

SUBMIT

10:15

Test report is generated in the form of pie chart displayed below as well as the time taken to finish the test. Tap on 'view solutions' to view detailed explanation of answers.



The figure shows a mobile application interface for an explanation. At the top, it says 'Explanation'. Below that, a question is asked: 'The point (1,-2) lies on which of the following lines?'. Underneath, it says 'The correct answer is  $x+y=-1$ ' and 'You selected:  $x+y=-1$ '. A section titled '**Explanation:**' follows, containing text about finding points on a line by substituting coordinates into the equation. At the bottom, there are two green buttons: 'Previous' on the left and 'Next' on the right, both circled in red. The bottom of the screen shows standard Android navigation icons and a status bar with the time '10:16'.

Previous and Next button can be used to view the explanation of previous and next question

PointTest

✓ SUBMIT

Question1

Question2

Question3

Question4

Question5

Question6

Question7

Question8

Q1. What are the Cartesian coordinates of the point A?

(4,2) (2,4) (-4,2) (4,-2)

10:15

List at left side gives the number of all the questions, through which we can move from any question to another

3. Scaling consists of five modules: Practical Example, Scaling basics, Solved Example, Practice and Quiz-It. These modules work in similar way as the sub-modules of Plotting.

MyPoint C-PLANE PLOTTING SCALING LINE QUIZ-IT

Practical Example

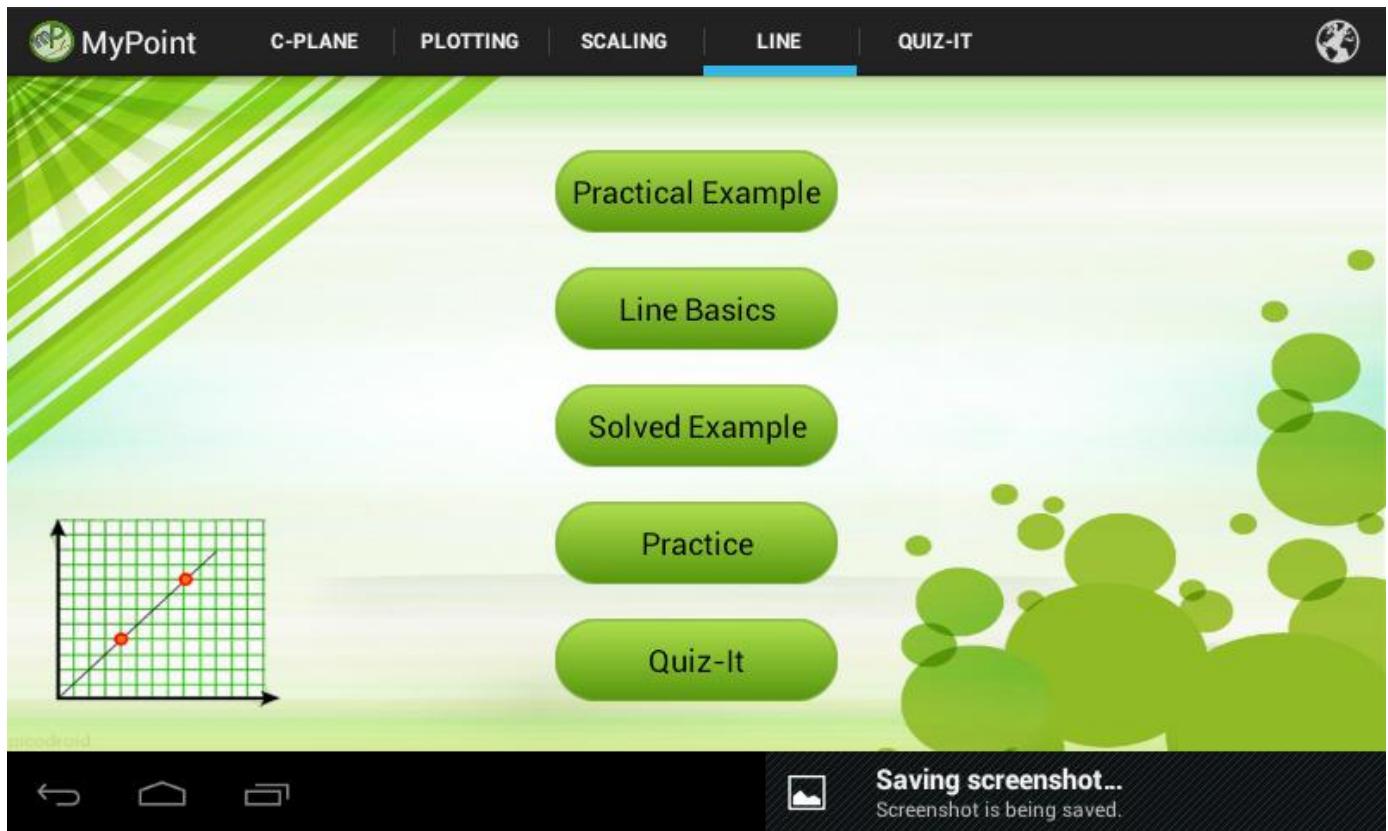
Scaling Basics

Solved Example

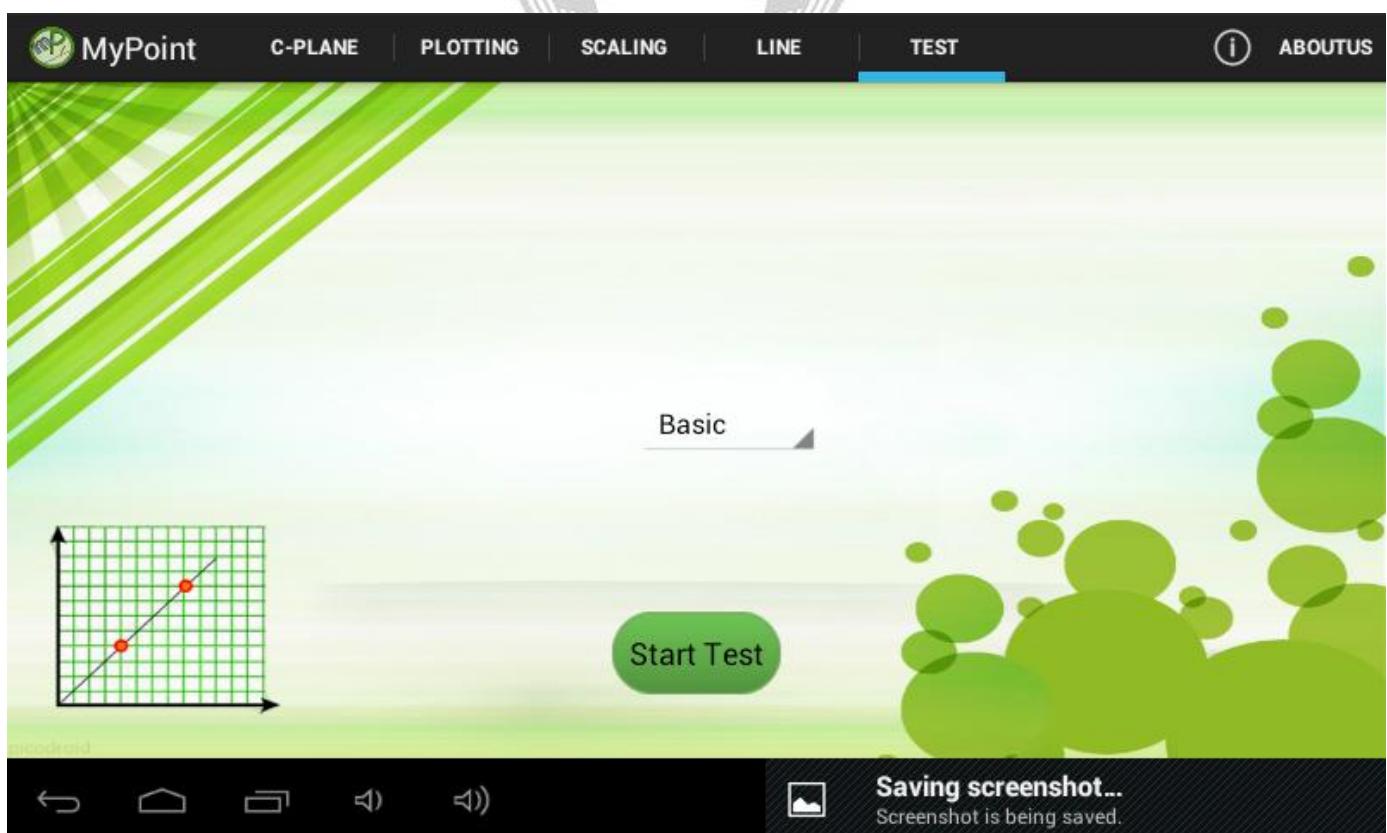
Practice

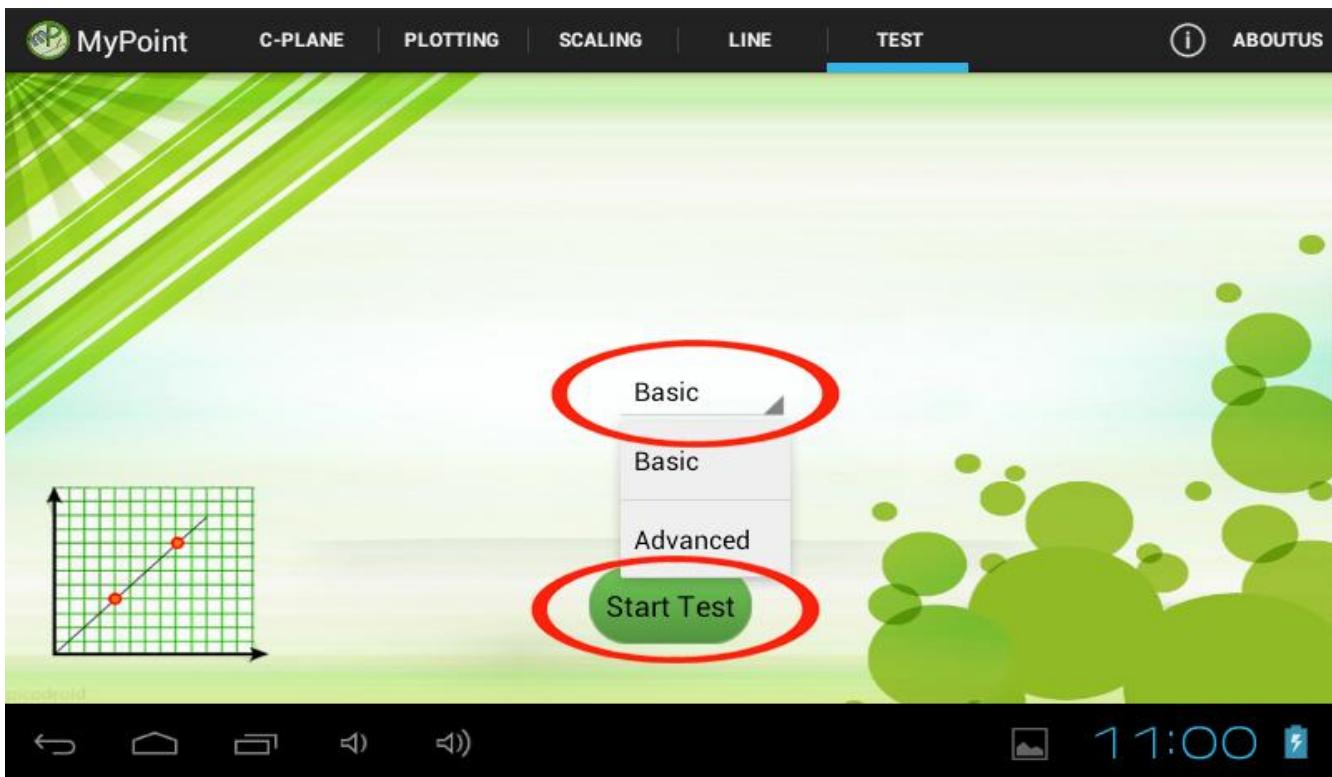
Quiz-It

4. Line consists of five modules: Practical Example, Line basics, Solved Example, Practice and Quiz-It. These modules work in similar way as the modules of Plotting



5. Quiz-it consists of Basic and Advanced levels of test.





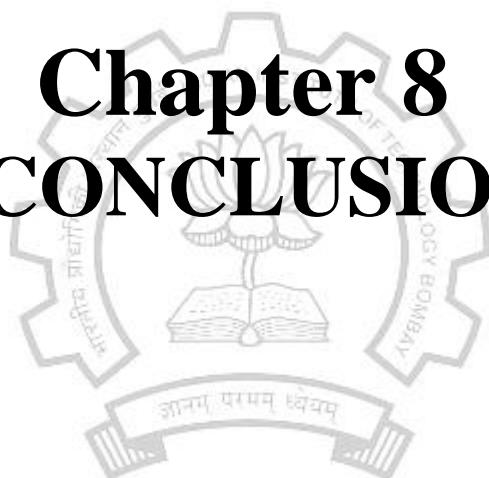
User choose between different levels of quiz using drop down menu and “Start Test” button can be used to start a quiz of selected level with fixed number of questions. These quiz consist questions related to all modules.

The screenshot shows the 'PointTest' app interface. On the left, there is a vertical list of questions: Question1, Question2, Question3, Question4, Question5, Question6, Question7, and Question8. On the right, a question is displayed: "Q1. What are the Cartesian coordinates of the point A?". Below the question is a Cartesian coordinate system with a grid from -4 to 4 on both axes. A point labeled 'A' is plotted in the first quadrant at the coordinates (4, 2). To the right of the grid are four answer options, each with a radio button:  
 (4,2)  
 (2,4)  
 (-4,2)  
 (4,-2"

We can submit and can view solutions of the questions  
Each time we will get a different sequence of questions as well as different sequence of options

# **Chapter 8**

# **CONCLUSION**



## 8 CONCLUSION

### 8.1 User Feedback

The acceptance testing conducted with the actual users of the application, resulted into following feedbacks:



INTERACTIVE LESSON LEARNER "myPoint"

IIT Bombay Summer Interns Program, 2013

29<sup>th</sup> June, 2013

#### FEEDBACK FORM

NAME : MANJUJA RANA

DESIGNATION : V.P

EMAIL-ID : mranas.organic@gmail.com

- How much helpful do you think the application would be to students?  
 1. Very helpful      3. Somewhat helpful  
2. Helpful      4. Not so helpful
  
- Do you think it can help the students understand the concept better?  
 1. Yes      2. May Be      3. No
  
- Will it help in teaching different concepts more exhaustively and efficiently?  
 1. Definitely      2. May be      3. Don't know
  
- How did you find the module sequence of the application?  
1. very good  
 2. Good  
3. Needs improvement  
4. Not so good
  
- Any Suggestions for improvement

Well prepared, perfect

Manjula Rana  
Signature

Thank you for your support and patience  
Interactive lesson learner Team

## Feedback 1



INTERACTIVE LESSON LEARNER "myPoint"

IIT Bombay Summer Interns Program, 2013

29<sup>th</sup> June, 2013

### FEEDBACK FORM

NAME : JAYASHREE S.

DESIGNATION : TGT (MATHS)

EMAIL-ID : jayashree128@gmail.com

- How much helpful do you think the application would be to students?  
 1. Very helpful       3. Somewhat helpful  
 2. Helpful       4. Not so helpful
  
- Do you think it can help the students understand the concept better?  
 1. Yes       2. May Be       3. No
  
- Will it help in teaching different concepts more exhaustively and efficiently?  
 1. Definitely       2. May be       3. Don't know
  
- How did you find the module sequence of the application?  
 1. very good  
 2. Good  
 3. Needs improvement  
 4. Not so good
  
- Any Suggestions for improvement

Jayashree S.  
Signature

Thank you for your support and patience  
Interactive lesson learner Team

## Feedback 2



## INTERACTIVE LESSON LEARNER "myPoint"

### IIT Bombay Summer Interns Program, 2013

29<sup>th</sup> June, 2013

#### FEEDBACK FORM

NAME : Sonika Sehrawat

DESIGNATION : TGT Maths

EMAIL-ID : sonikaseh@hotmail.com

- How much helpful do you think the application would be to students?  
 1. Very helpful      3. Somewhat helpful  
2. Helpful      4. Not so helpful
  
- Do you think it can help the students understand the concept better?  
 1. Yes      2. May Be      3. No
  
- Will it help in teaching different concepts more exhaustively and efficiently?  
 1. Definitely      2. May be      3. Don't know
  
- How did you find the module sequence of the application?  
 1. very good  
2. Good  
3. Needs improvement  
4. Not so good
  
- Any Suggestions for improvement

.....None.....

Sonika  
Signature

Thank you for your support and patience  
Interactive lesson learner Team

#### Feedback 3



## INTERACTIVE LESSON LEARNER "myPoint"

### IIT Bombay Summer Interns Program, 2013

V.R.PATIL  
P.G.T (MATHS)  
springbato1@gmail.com

29<sup>th</sup> June, 2013

### FEEDBACK FORM

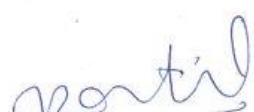
NAME :

DESIGNATION :

EMAIL-ID :

- How much helpful do you think the application would be to students?  
 1. Very helpful      3. Somewhat helpful  
2. Helpful      4. Not so helpful
  
- Do you think it can help the students understand the concept better?  
 1. Yes      2. May Be      3. No
  
- Will it help in teaching different concepts more exhaustively and efficiently?  
 1. Definitely      2. May be      3. Don't know
  
- How did you find the module sequence of the application?  
1. very good  
 2. Good  
3. Needs improvement  
4. Not so good
  
- Any Suggestions for improvement

It is already given sys.  
Mathematically it is nice



Signature

Thank you for your support and patience  
Interactive lesson learner Team

### Feedback 4

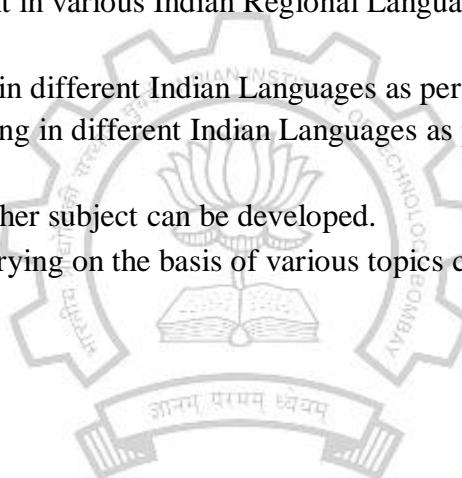
## 8.2 Conclusion

The application aim to develop Interactive Lessons for students through which they can learn, understand, practice and evaluate themselves is successfully achieved. It thus also provides flexibility of studying anytime, anywhere, and at one's own pace. The application also works as an aid in teaching. It enables taking assignments, conducting inbuilt tests by teachers. It also provides functionalities which help students to explore respective lesson beyond the scope of textbook.

Although we have been successful in implementing our desired aim of the project with a total of **16,367 lines of code (LOC)**, there is always a future maintenance possible in order to overcome the limitations in the current system.

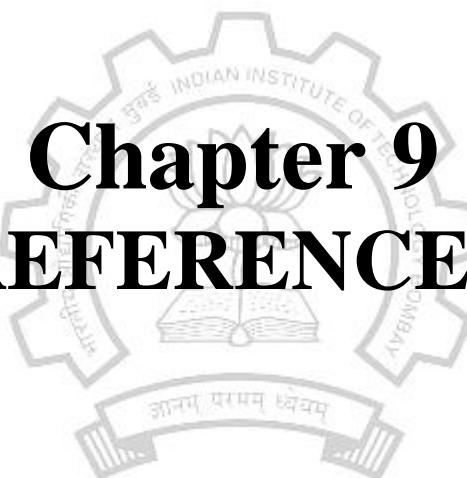
## 8.2 Future Enhancement

- Embedding of content in various Indian Regional Languages.
  - Text rendering in different Indian Languages as per the requirements.
  - Audio embedding in different Indian Languages as per the requirements.
- Similar lessons for other subject can be developed.
- More interactivity varying on the basis of various topics can be added.



# **Chapter 9**

# **REFERENCES**



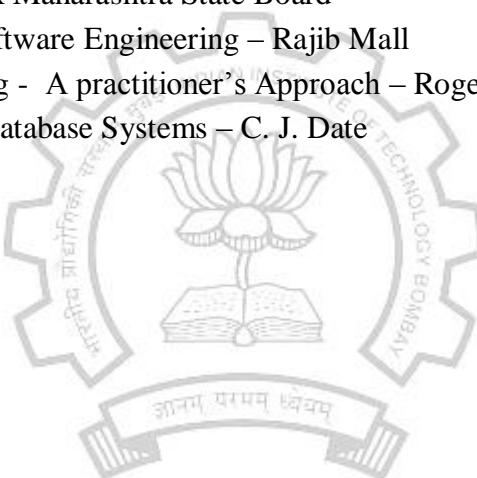
## **9 REFERENCES**

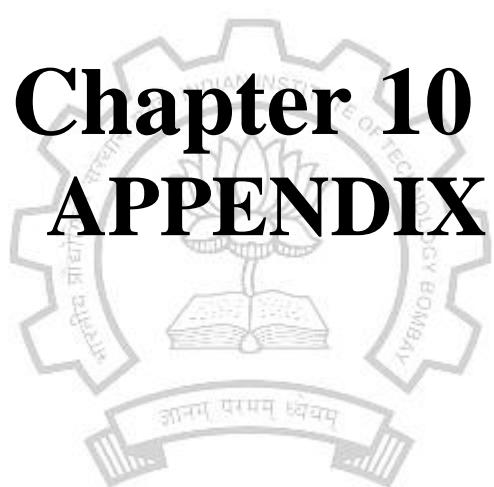
### **➤ Web References:**

- Android Developer Website, <http://developer.android.com/>
- Custom List View Website, <http://www.thenewboston.org/>
- Android Tutorial Website, <http://en.wikipedia.org/wiki/Android>
- Wikipedia, [http://en.wikipedia.org/wiki/Cartesian\\_coordinate\\_system](http://en.wikipedia.org/wiki/Cartesian_coordinate_system)
- KhanAcademy, [https://www.khanacademy.org/math/algebra/introduction-to-algebra/overview\\_hist\\_alg/v/descartes-and-cartesian-coordinates](https://www.khanacademy.org/math/algebra/introduction-to-algebra/overview_hist_alg/v/descartes-and-cartesian-coordinates)

### **➤ Book References:**

- Mathematics, Standard IX CBSE
- Algebra, Standard IX Maharashtra State Board
- Fundamentals Of Software Engineering – Rajib Mall
- Software Engineering - A practitioner's Approach – Roger S. Pressman
- An Introduction to Database Systems – C. J. Date





# Chapter 10

# APPENDIX

## 10 APPENDIX

Acronym or Abbreviation	Description
ADT	Android Development Tools
API	Application Programming Interface
APK file	Android Application Package file
AVD	Android virtual Device
GUI	Graphical user Interface
IDE	Integrated Development Environment
SDD	Software Design Document
SDK	Software Development Kit
SRS	Software Requirement Specification
SDLC	Software Development Life Cycle