CHAPTER – 1

INTRODUCTION

According to JNTU curriculum, students are capable of scoring 25% of the overall marks based on their performance in the internal i.e. Mid-Term examinations. In the internal examinations, each subject carries 25 marks, out of which 10 marks are allotted for subjective type questions, 5 marks are allotted for assignments and the rest of the 10 marks are allotted for the objective type questions. Students face many difficulties especially while preparing for the objective type questions as the existing system i.e. previous year’s objective type question papers and their keys available in pdf format on JNTU-related websites & blogs, class notes, textbooks etc. which does not provide an effective way of learning. The existing system is very time consuming the student has to visit multiple web sites to view and download previous question papers and most of the question papers are of pdf format where questions are listed on one page and answers are listed on another page which makes the student difficult to read and learn, he will have to switch to two different pages in order to read one objective type question .The student have to repeat this to learn all the questions which is time consuming and inefficient to learn .In order to overcome these problems, we need to develop a system that can accessed by the students(users) easily. We strongly believe that building an application on Android can solve the problems faced by the existing system.

These days majority of students own smartphones and are also familiar with the usage of smartphones. Android is an open source and Linux-based operating system for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android. Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide.

Android applications are usually developed in the Java language using the Android Software Development Kit. Hence we have proposed a system that is “an Android application for learning JNTU bits” can overcome all the difficulties of the existing system mentioned above. The detailed documentation of the application is provided here. The workflow of this application, testing, design etc. are also specified. The multiple aspects in which our proposal differs from the pre-existing system is also explained.

1.1 PROBLEM STATEMENT:

Every semester, JNTU conducts two mid-term examinations consisting of 10 marks worth of multiple choice questions and fill-in-the-blanks. In order to secure good marks in the internal examinations students must be prepared with 4 units of each subject. They can prepare from multiple sources including notes, textbooks, references and previous question papers. It is difficult and time-consuming to find and download all question papers as they are located at various sites. We have to download these question papers and the question papers are in pdf files and its respective keys are in another pdf files. Hence it is often time consuming and difficult for the students to prepare effectively for the bit paper of mid examination.

1.2 MOTIVATION:

The marks secured in internal examinations plays a vital role in improving the overall aggregate of students. Most of the students prepare well for the theory part of the internal exams as it has efficient and direct sources to prepare like notes, guides, textbooks etc. but sadly, they are lethargic in preparing for the bit paper of internal examination. The reason behind this is that the sources for preparing for bit papers are not clear and are difficult to use. Hence it is important to build a new system using a new approach through which the students can easily prepare for the bit papers. This system should also be in such a way that it should be available and accessible to majority of students. Therefore all above factors led to development of an android application for learning JNTU bits. This provides its users with a user friendly interface and an effective way of learning the JNTU bits.

1.3 SCOPE:

This software system will be an Android application for Learning JNTU bits. This system will be designed to maximize scope for the users to score higher grades in Mid-term examinations. This is done by preparing the user for Multiple Choice Questions (MCQ) and

Fill-in-the-blank questions worth 10 Marks. This system provides a user friendly environment for users. It can be accessed on-the-go at any time by the users. It provides a user interface in a hierarchical manner, hence it doesn’t lead to any confusion. This system also contains a relational database (using SQLite) containing a list of questions (BITS) and its corresponding answers for different branches and regulations.

1.4 OUTLINE:

This application serves as an effective solution for the students facing difficulties in preparing for bit paper of internal examination. There is no need of internet connectivity in order to use this application. This application overcomes all the drawbacks of the existing system. This application provides efficient user interface and it easy to operate. This application supports the features of android as it is built on Android. This application operates in either of the two modes (revision mode/learning mode) based on the requirement of the user. Therefore this application is user friendly and also provides high performance.

CHAPTER - 2

LITERATURE SURVEY

Introduction:

World is contracting with the growth of mobile phone technology. As the number of users are increasing day by day, facilities are also increasing. I began with simple regular handsets which were used just for making phone calls, today mobiles have changed our lives and have become part of it. Now they are not used just for making calls but they have innumerable uses. They can be used as a Camera, Music player, Tablet PC, T.V., Web browser etc. With new technologies, new software and operating systems are required. Operating Systems have progressed a lot in the last 15 years. Evolving from monochrome phones to recent smart phones or mini computers, mobile OS has come a long way. In the case of smart phones, Mobile OS spawned from Palm OS in 1996, leading to Windows pocket PC in 2000, which subsequently led to the birth of Blackberry and Android Operating Systems.

One of the most widely used mobile OS these days is Android. Android is a software stack comprising not only operating system but also middleware and key applications. Android Inc. was founded in Palo Alto of California, U.S. by Andy Rubin, Rich miner, Nick sears and Chris White in 2003. Later Android Inc. was acquired by Google in 2005. After original release there have been number of updates in the original version of Android. Android is a powerful Operating System supporting a large number of applications in Smart Phones. These applications make life more comfortable and less complex for the users. Hardware that support Android are mainly based on ARM architecture platform. However, some more hardware vendors are entering into mobile hardware such as Intel with its Atom processor. Some of the current features and specifications of android are: Application framework (It enables reuse and replacement of components), Dalvik Virtual Machine (Optimized for mobile applications), Integrated Web Browser (Based on the open source Web kit), Optimized framework (It is powered by a custom 2D and 3D graphic library) and many more useful features.

Android comes with an online marketplace which is an Android software store. It was developed by Google. It allows Android users to select, and download applications developed by third party developers and use them. There are around 2,00,000 plus games, applications and widgets available on the market for users. Android applications are written in java programming language. Android APIs are available as open source for developers to develop applications which can be sold in android market. There are around 200000 applications developed for android with over 3 billion plus downloads. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. For software development, Android provides Android SDK (Software development kit). The SDK involves an IDE, emulator and tools required by developers for the smooth development of Applications.

The Advantages of Android:

Multitasking: Android phones can run many applications, it means you can browse the web, download a file in the background while listening to a song.

Ease of Notification: When the user receives any SMS, Email, or even latest articles from an application, there will always be a notification on the Android phone's Home Screen, if an LED indicator is present even it indicates pending notifications by blinking, so you will not miss a single SMS, Email or even Missed-calls.

Google Play Store: One can access thousands of apps and games and easily find something you love, through Google's Android App Market. Anyone can download the applications for free. There are many thousands of applications and games that are ready for download on Android phones.

Phone options are diverse: Android phone feels 'different' than Apple's iOS, if iOS is limited to the iPhone from Apple, then Android is available on mobile phones from various manufacturers, from Sony Ericsson, Motorola, and HTC to Samsung. Each handset manufacturer also presents an Android phone in its own style, such as Motorola with its Motoblur, Sony Ericsson with its Timescape. So you can freely choose the Android phone in accordance with the 'brand' favourite.

Custom ROM: A user still not satisfied with the standard view of Android, need not worry as there are many custom ROMs that can be used in your mobile phone providing different features and functions.

Android Widgets: With the widgets on the home screen, one can easily access a variety of settings quickly and easily.

Google Maniac: Loyal users of Google services ranging from Gmail to Google Reader will benefit the most. Android phones come integrated with Google services, so you can quickly check e-mail from Gmail or view your appointments on calendar.

2.1 EXISTING SYSTEM:

According to JNTU, every semester will have two mid-term examinations (except for the first year, as it only has three mid exam for the academic year) consisting of 10 marks worth of multiple choice questions and fill-in-the-blanks. Students must prepare for 4 units for each Mid-Term exam. As per the existing system students prepare themselves for objective type questions from multiple sources including textbooks, references and previous question papers. Previous question papers are the most important source of questions as there is a high likelihood that few or more than a few questions will be repeated in the exam. These papers are accessed via webpages (and/or) other sources in the form of PDFs/word documents. This generation students are used to a fast-paced lifestyle. As the existing system is time consuming and it deals with patience, most of the students ignore the preparation for objective type questions. They might rather try their luck in objective type questions during the internal exams. Hence this affects the students overall performance in internal exams.

DISADVANTAGES:

1) It is difficult and time-consuming to find and download all question papers as they are located at various sites.

2) One cannot effectively from the question papers as questions and answers are located at different pages which leads to confusion.

We can overcome the flaws of the existing system by making use of the proposed system that is mentioned below.

2.2 PROPOSED SYSTEM:

The marks obtained in internal examinations play an important role in improving the percentage of students. According to JNTU curriculum, students are capable of scoring 25% of the overall marks scored in each semester, based on their performance in the internal i.e. Mid-Term examinations. In the internal examinations, each subject carries 25 marks, out of which 10 marks are allotted for subjective type questions, 5 marks are allotted for assignments and the rest of the 10 marks are allotted for the objective type questions. Students face many difficulties especially while preparing for the objective type questions as the existing system does not provide an effective way of learning. Hence, we were determined to come up with an idea that would solve all the above mentioned difficulties of students, which led us to propose a system which would likely take over the existing system. We strongly believe that technology can solve most the problems that deals with education. Hence we have decided to overcome the above mentioned problem by making use of one of the latest technologies that is widely used all over the world i.e. the Android technology. The system will be an Android application for learning JNTU bits. It will be a handy application for the users. The user shall be able to use this system whenever and wherever he/she wants. The user shall be able to prepare for different subjects objective type questions from one single source i.e. with the help of the proposed system (Android application). This system will be built with an intention to be highly beneficial for the users (especially for the students). We will build this system in such a manner that it would support most of the versions of Android operating system. Therefore this system is intended to provide an optimized way of learning the objective type questions to the users.

ADVANTAGES:

1) Easy to access JNTU bits which are updated regularly.

2) Scoring-system for students to learn effectively.

3) Saves time by allowing user to switch to multiple subjects within seconds

CHAPTER – 3
SOFTWARE REQUIREMENT SPECIFICATIONS

3.1 Overall Description

3.1.1 Product Perspective:

To provide users an optimized way of learning JNTU bits. This is done by providing the bits (Multiple Choice Questions and Fill in the Blanks) of various subjects in the application, which can be accessed easily by the users. The efficiency of learning is increased as this application can be operated in either of the two modes (learning mode and revision mode).

3.1.2 Product Functions:

The system is intended to perform the following functions:

• It provides branch wise objective type questions for the user.

• It also provides objective type questions based on the regulations.

• It allows the user to easily navigate through various subject’s objective type question papers based on his/her requirement.

• It acts as a platform for the users to learn the objective type questions effectively using different modes:

(1) Learning mode

(2) Revision mode.

• Learning mode: In this mode the system provides list of questions and when user selects a particular question then the system displays that question along with options where user is facilitated to select an option according to his choice and submit it then the system shows weather the user answered it correctly or not in this way it provides an active, comfortable and effective mechanism towards learning.

• Revision mode: In this mode the users are provided with all the list of questions along with answers to revise so that it does not waste time during exams.

• The system consists of a scoring system, which allows the user to self-assess.

3.1.3 User Documentation

The system will consist of in-built documentation which can be accessed via the help menu. The documentation will contain information about various functions of the system and how to utilize those functions in an effective manner.

External Interface Requirements

3.1.4 User Interfaces:

UI-1: This system display’s a welcome screen as an introduction for the first time for users.

UI-2: After that, the system navigates to a new screen where the users shall be able to set their batch details.

UI-3: Then the user shall be able to select the subject and the mid-term.

UI-4: Now the user shall be able to attempt a question.

UI-5: Then the user shall be able to view the correct answer.

UI-6: The system displays current mode (Learning, Revision) in the title bar.

UI-7: The system provides single space for choosing both modes and changing settings.

UI-8: The user shall be able to view help and about information.

UI-9: The system provides the users with a single screen where all the questions are displayed and whether it was answered correctly.

UI-10: Tabs to switch quickly both Multiple Choice questions and fill-in-the blanks.

UI-11: The user shall be able to exit the application.

3.1.5 Hardware Interfaces:

This system will be an Android application which is designed to run on phones and tablets. There are three types of buttons in Android. The system uses only the back button to facilitate navigation in the application. Android devices use full screens hence, our system will be configured to use touch and touch gestures such as swiping, flicking and so on.

3.1.6 Software Interfaces

Android operating system provides us with many pre-packaged frameworks to create things like menus, buttons, activities and access common functionality of mobile phones. This system will be using an SQL-based local database, SQLite, to store questions and their corresponding answers.

3.2 Operating Environment

This system is intended to be deployed on portable devices i.e. mobile devices, and tablets. The software executes on the Android operating system, version 4.0.3 and above. This will cover nearly 94.1% of all Android phones and tablets currently active worldwide. This data was collected by Google and has been updated on their website as of 01st June 2015.

3.2.1. Hardware Requirement

The minimum hardware requirements for android device are:
Processor Speed: 1 Ghz
RAM: 512 MB
Memory Storage: 1GB

3.2.2 Software Requirements

Operating System: Android 4.0.3 and above

3.3 FUNCTIONAL REQUIREMENTS:

The following are the three main scenarios of the application:

3.3.1 Learning Mode:

3.3.1.1 Description and Priority

Learning mode offers the user a chance to attempt a question before revealing the correct answer to the user. This gives the user feedback on whether their answer was correct or not.
Priority: HIGH

It is given high priority as it is an essential part of the system without which users cannot become better at scoring good marks in objective type questions in the internal examinations.

3.3.1.2 Functional Requirements

REQ-1: User should select the current batch.

REQ-2: User should select the current mid-term.

REQ-3: User must select appropriate subject.

REQ-4: System should verify that appropriate year and corresponding semester are selected. The system does not allow user to make erroneous selections.

REQ-5: System should verify that only subjects in that semester are displayed and selected.

REQ-6: User can select any question from a list of all the questions in that subject.

REQ-7: User can view and answer the questions.

REQ-8: System should be able to provide user with correct answer in a clear manner.

REQ-9: System should be able to provide option of loading the next Question.

3.3.2 Revision Mode

3.3.2.1 Description and Priority

In this mode, the user can browse through all the questions provided in the application. The correct option to the question is displayed to the user.

Priority: HIGH

Revision mode is high priority is a fast-paced mode designed to enable the users to quickly revise the answers before the exam.

3.3.2.2 Functional Requirements

REQ-1: User should select the current batch.

REQ-2: User should select the current mid-term.

REQ-3: User must select appropriate subject.

REQ-4: System should verify that appropriate year and corresponding semester are selected. The system does not allow user to make erroneous selections.

REQ-5: System should verify that only subjects in that semester are displayed and selected.

REQ-6: User can select any question from a list of all the questions in that subject.

REQ-7: System displays question and its correct answer.

REQ-8: System allows user to load the next question.

3.3.3 Scoring System

3.3.3.1 Description and Priority

The scoring system displays the total number of questions answered correctly and the score for each subject. It allows a user to assess their own progress while learning and allows them to compare their progress in different subjects.

Priority: HIGH

The user can measure their own performance. It is difficult to assess themselves and understand their progress without a scoring system.

3.3.3.2 Functional Requirements

REQ-1: System stores the total no. of questions answered by the user in an array.

REQ-2: System should check whether the user answered a question correctly or incorrectly.

REQ-3: System should assign a score based on the number of correctly answered questions.

REQ-4: System displays the score as a total and per subject basis.

REQ-5: The user can reset the score if needed. When the user resets the score the system displays attempted questions and their score as zero. The user can retake learning mode to start attempting the questions from the beginning.

3.4 NON-FUNCTIONAL REQUIREMENTS:

3.4.1 Performance Requirements

The system does not make use of any network capabilities. The system does not perform computationally expensive tasks. Hence the system will have a good speed while performing various tasks .This system requires sufficient memory storage capacity on the portable android device as it uses a database to store a large number of data.

3.4.2 Safety Requirements

There are not many safety requirements with this application, other than any normal hazards of a mobile device. One such hazard is user, using this system when they are driving a vehicle. And on the other hand the users are strictly suggested to use this system only for preparation purpose before the exam but not in the examination hall during the exam.

3.4.3 Security Requirements

A user cannot edit any question or answer stored in the database of the application. Android provides SQLite helper classes which is implemented in such a way such that only classes belonging to our system may access the database. Additionally, our system utilizes the built-in security in Android operating systems which provide security to the user's device through a password, pin or pattern.

3.4.4 Software Quality Attributes

a) Correctness: As the data of the system i.e. objective type questions and their respective answers are being collected from genuine sources like the JNTU University, websites, faculty etc. And it is arranged in an error-free manner. Hence the system ensures correctness to its users.

b) Flexibility: The system layout is in such a way that it can be accessed via both mobile and tablets.

c) Usability: This system can be used on-the-go by the users. It provides a high degree of usability.

d) Reliability: The user can completely rely on this system for effective learning of the objective type questions.

CHAPTER – 4

DESIGN DIAGRAMS

4.1 UML DIAGRAMS

4.1.1 Use case diagram

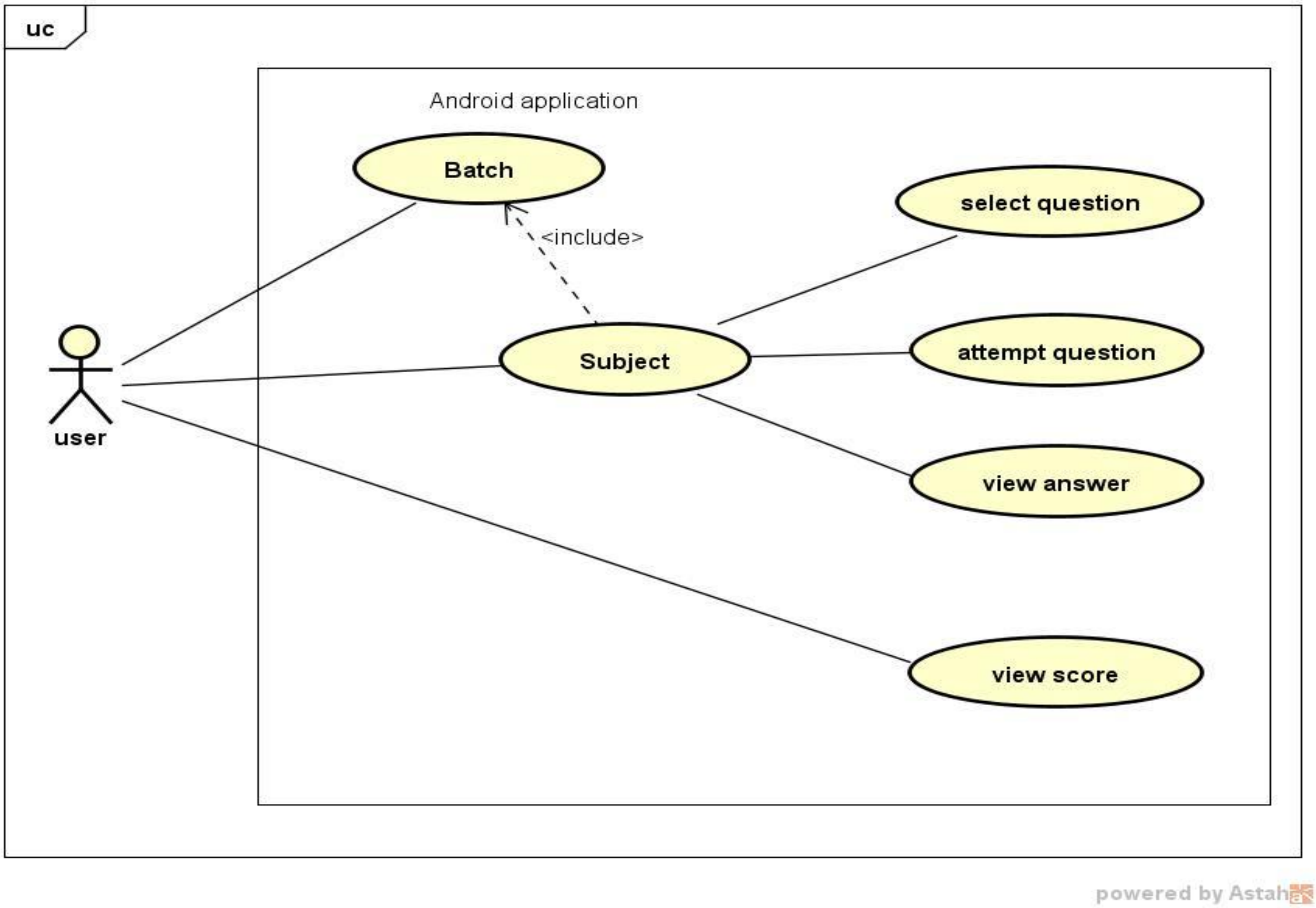


Fig 4.1.1.1 Use case diagram for Android application

Description

In the above use case diagram, the actor is the user. The system here is an Android application .The user can set the batch, the subject use case and batch use case have an include relationship between them .The subject use case is again associated with select question, attempt question and view answer use cases .The user is also associated with the view score use case.

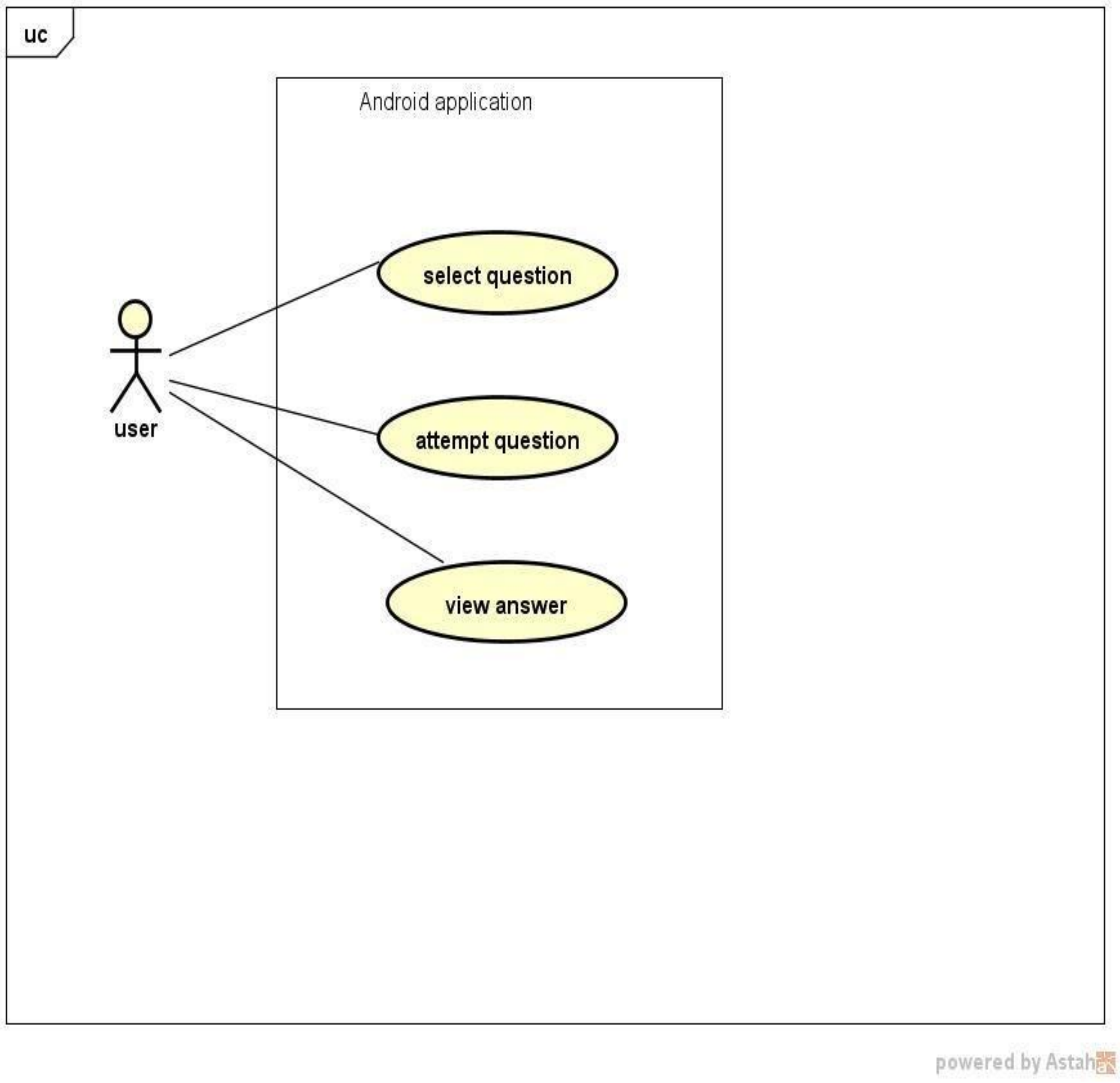


Fig 4.1.1.2 Use case diagram for learning mode

Description:

The system provides the user with two types of modes, one of the mode is learning mode and the other is revision mode. In learning mode the user will first select a question from a list of questions and after he attempts that question he is can view the correct answer.

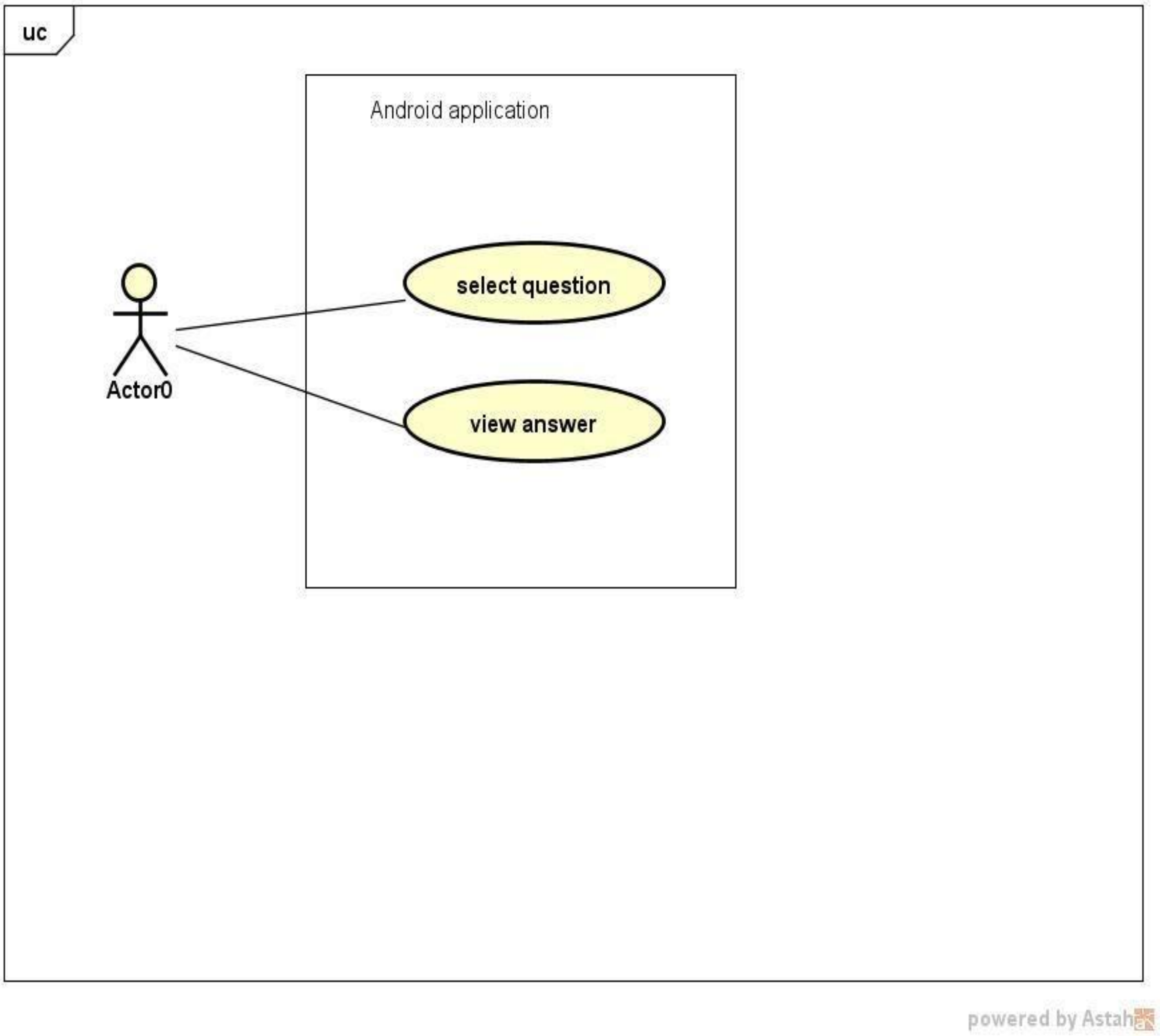


Fig 4.1.1.3 Use case diagram for revision mode

Description:

As mentioned earlier revision mode is one the mode provided to user by the system. The revision mode includes two use cases namely select question and view answer. In this mode the user will select a question from a list of questions and the answer is viewed along with the selected question. Unlike the learning mode, here the user need not attempt the question to view the correct answer.

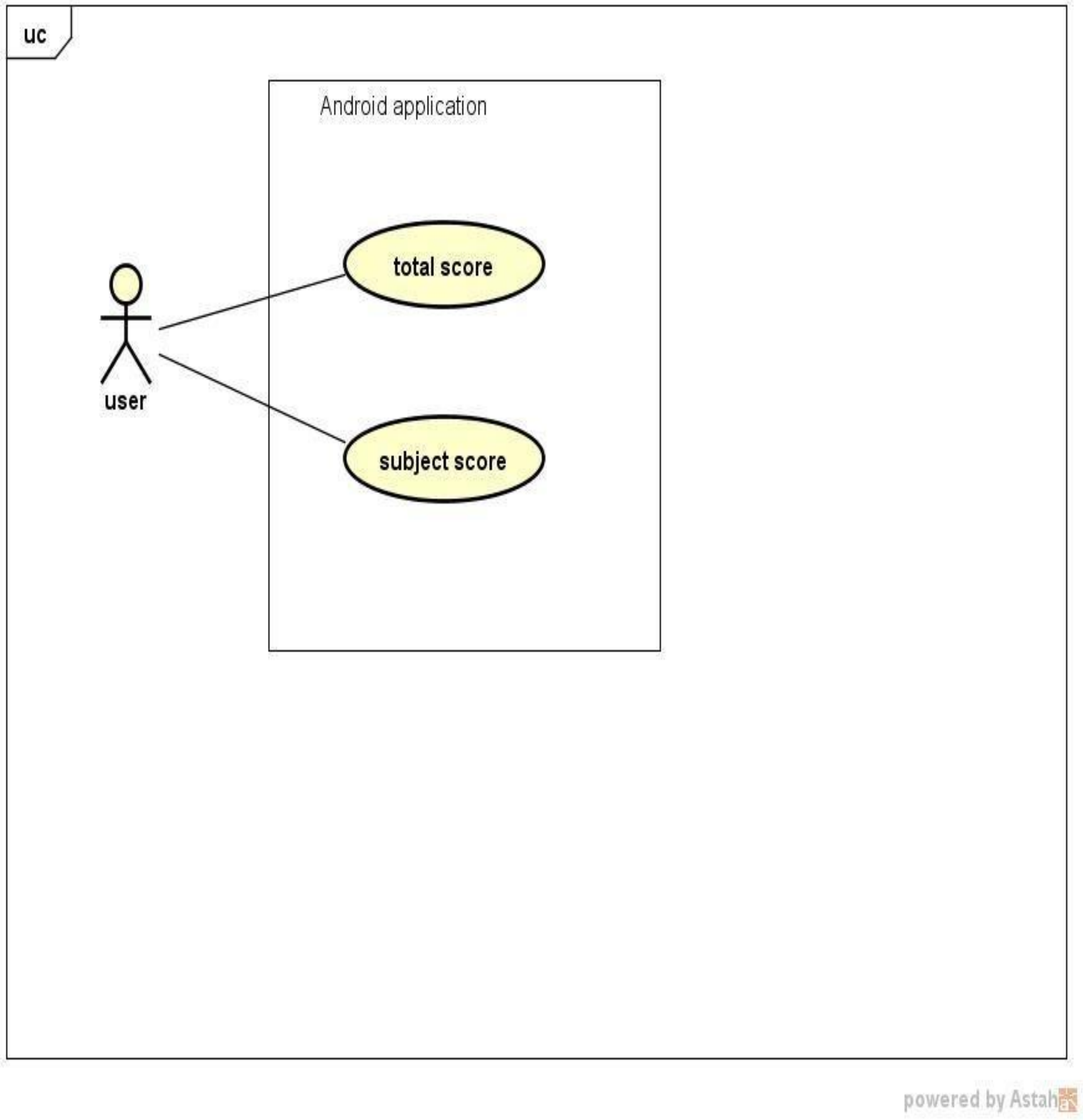


Fig 4.1.1.3 Use case diagram for scoring system

Description

In the scoring system, the user is associated with two use cases i.e. total score and subject score .The user can interact with the total score when he wants to know the overall score i.e. the total score the user obtained from each and every subject together. If the user wants to know the individual score he obtained in each subject, then he can interact with the subject score.

4.1.2 CLASS DIAGRAM

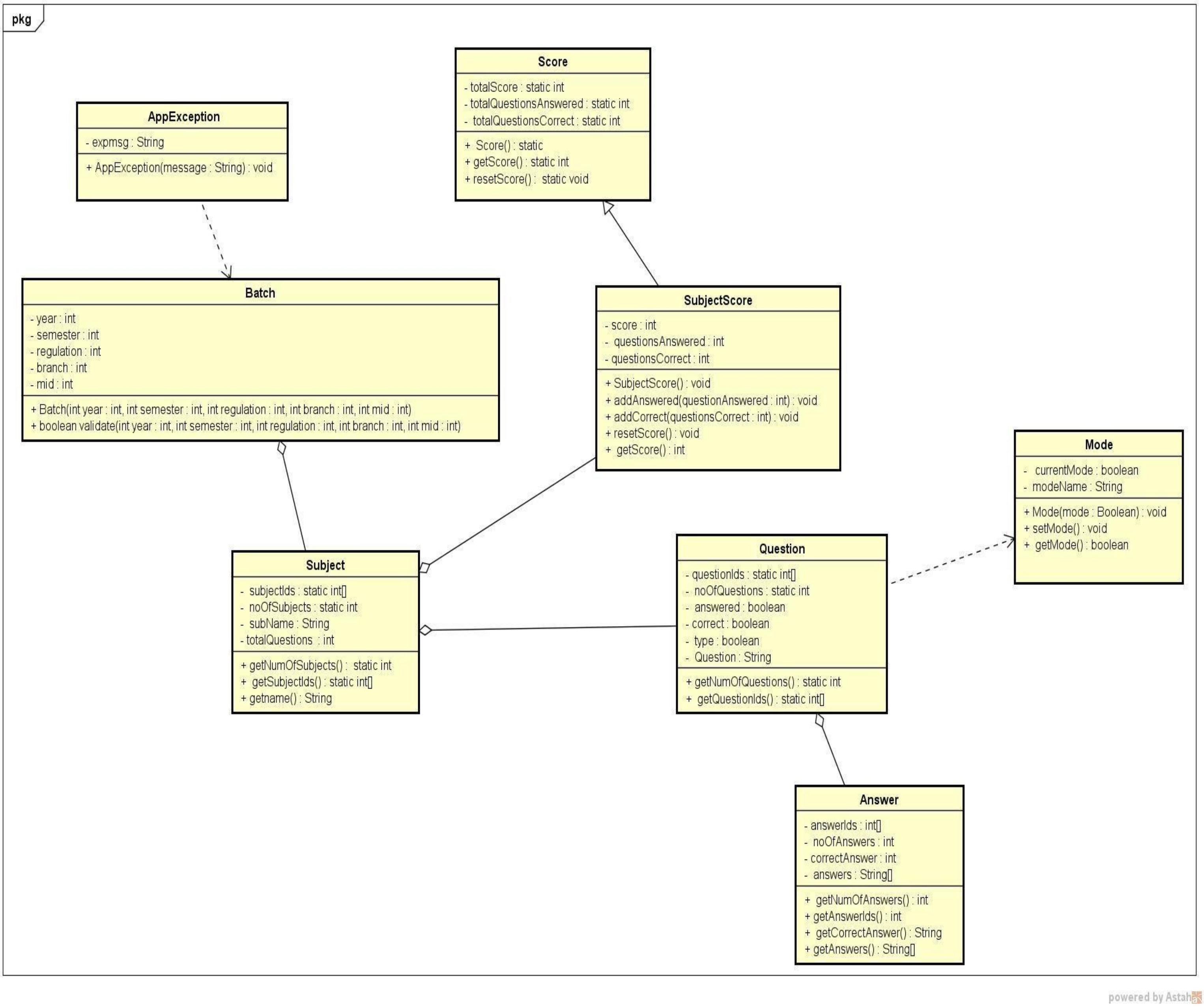

pkg
Score
- totalScore : static int
- totalQuestionsAnswered : static int
- totalQuestionsCorrect : static int
+ Score() : static
+ getScore() : static int
+ resetScore() : static void
AppException
- expmsg : String
+ AppException(message : String) : void
Batch
- year : int
- semester : int
- regulation : int
- branch : int
- mid : int
+ Batch(int year : int, int semester : int, int regulation : int, int branch : int, int mid : int)
+ boolean validate(int year : int, int semester : int, int regulation : int, int branch : int, int mid : int)
SubjectScore
- score : int
- questionsAnswered : int
- questionsCorrect : int
+ SubjectScore() : void
+ addAnswered(questionAnswered : int) : void
+ addCorrect(questionsCorrect : int) : void
+ resetScore() : void
+ getScore() : int
Mode
- currentMode : boolean
- modeName : String
+ Mode(mode : Boolean) : void
+ setMode() : void
+ getMode() : boolean
Subject
- subjectIds : static int[]
- noOfSubjects : static int
- subName : String
- totalQuestions : int
+ getNumOfSubjects() : static int
+ getSubjectIds() : static int[]
+ getname() : String
Question
- questionIds : static int[]
- noOfQuestions : static int
- answered : boolean
- correct : boolean
- type : boolean
- Question : String
+ getNumOfQuestions() : static int
+ getQuestionIds() : static int[]
Answer
- answerIds : int[]
- noOfAnswers : int
- correctAnswer : int
- answers : String[]
+ getNumOfAnswers() : int
+ getAnswerIds() : int
+ getCorrectAnswer() : String
+ getAnswers() : String[]
powered by Astah

Fig 4.1.2.1 Class diagram for an Android application for learning JNTU bits

Description:

AppException is a child of Exception class in Java. Similarly, SubjectScore inherits its attributes and methods from the Score class. From the class diagram it can be observed that Batch uses AppException and has one or more Subject. Subject has a SubjectScore and one or more Question. SubjectScore inherits from Score. Question class has one or more Answer and is using Mode.

4.1.3 SEQUENCE DIAGRAM

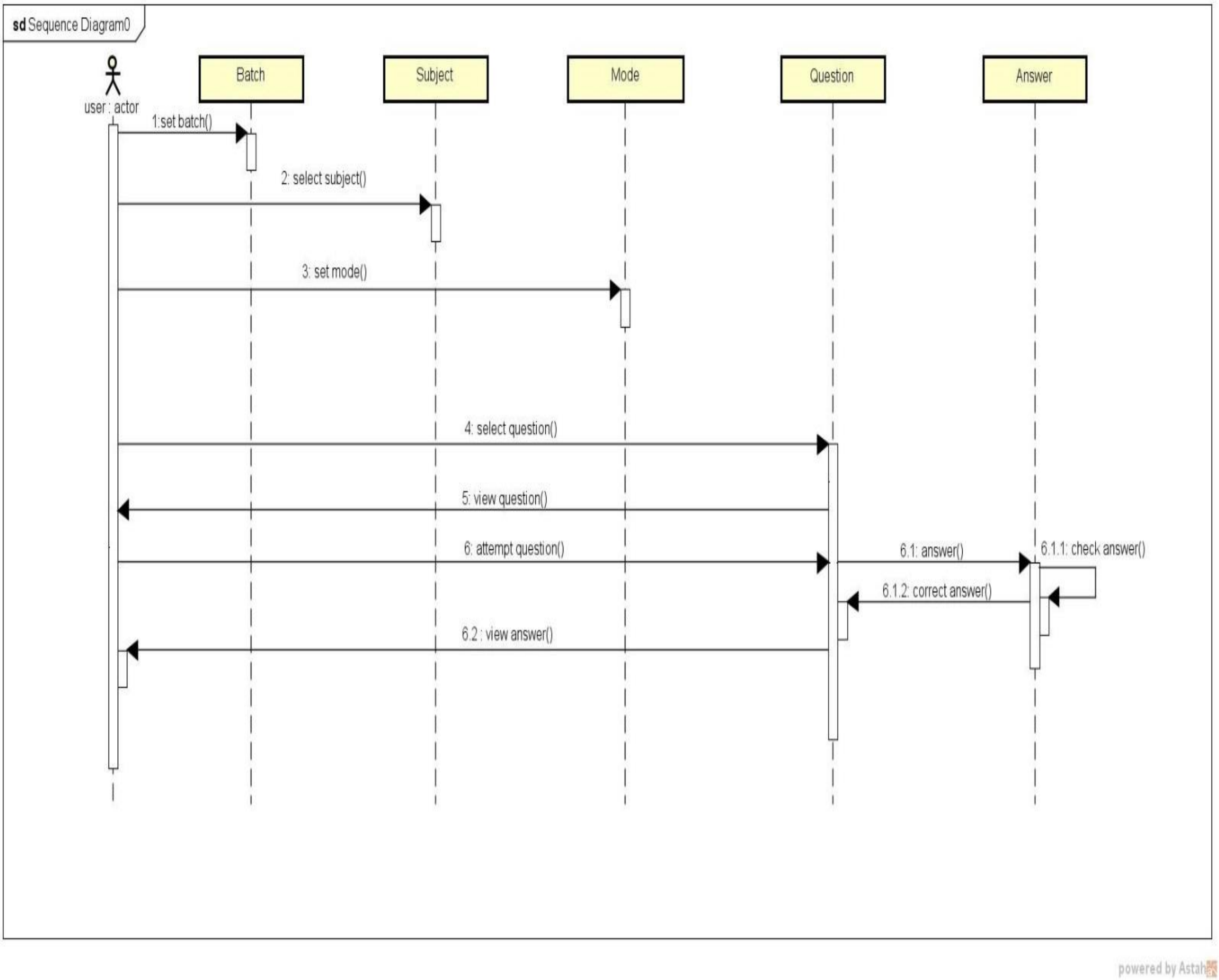


Fig 4.1.3.1 Sequence diagram for an Android application for learning JNTU bits

Description

In our system the user will first set the batch. After that the user will set the mode and then user will select a subject. Then user will select a particular question, then system views that selected question. Then the user attempts that question. Now the user can view the correct answer of that question.

CHAPTER-5

IMPLEMENTATION

5.1 Sample Code

Navigation Drawer and Subject List Activity

```
package in.momofactory.jntubits;
import android.content.Context;
import android.content.Intent;
import android.support.v4.view.MenuItemCompat;
import android.support.v7.app.ActionBar;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.support.v4.widget.DrawerLayout;
import android.widget.ArrayAdapter;
import android.widget.SpinnerAdapter;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity
        implements
NavigationDrawerFragment.NavigationDrawerCallbacks,
        OnItemSelectedListener {

    /**
     * Fragment managing the behaviors, interactions and
presentation of the navigation drawer.
     */
    private NavigationDrawerFragment mNavigationDrawerFragment;

    /**
     * Used to store the last screen title. For use in {@link
#restoreActionBar()}.
     */
    private CharSequence mTitle;
    private Spinner midSpinner;
    private AppSettings appSettings;

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

getSupportFragmentManager().findFragmentById(R.id.navigation_drawer)
;
        mTitle = getTitle();

        // Set up the drawer.
        mNavigationDrawerFragment.setUp(
                R.id.navigation_drawer,
                (DrawerLayout) findViewById(R.id.drawer_layout));
        appSettings = new AppSettings(getApplicationContext());
    }

    @Override
    public void onNavigationDrawerItemSelected(int position) {
        // update the main content by replacing fragments
        FragmentManager fragmentManager =
getSupportFragmentManager();
        fragmentManager.beginTransaction()
                .replace(R.id.container,
PlaceholderFragment.newInstance(position + 1))
                .commit();
    }

    public void onSectionAttached(int number) {
        switch (number) {
            case 1:
                mTitle = getString(R.string.title_section1);
                appSettings.setMode(AppSettings.LEARNING);
                break;
            case 2:
                mTitle = getString(R.string.title_section2);
                appSettings.setMode(AppSettings.REVISION);
                break;
            case 3:
                mTitle = getString(R.string.title_section3);
                break;
        }
    }

    public void restoreActionBar() {
        ActionBar actionBar = getSupportActionBar();
        actionBar.setDisplayShowTitleEnabled(true);
        actionBar.setTitle(mTitle);
    }


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        if (!mNavigationDrawerFragment.isDrawerOpen()) {
            // Only show items in the action bar relevant to this
screen
            // if the drawer is not showing. Otherwise, let the
drawer
            // decide what to show in the action bar.
            getMenuInflater().inflate(R.menu.main, menu);
            restoreActionBar();

            //midSpinner = (Spinner)
findViewById(R.id.action_bar_spinner);

            MenuItem item = menu.findItem(R.id.mid_spinner);
            midSpinner = (Spinner)
MenuItemCompat.getActionView(item);
            SpinnerAdapter mSpinnerAdapter =
ArrayAdapter.createFromResource(this.getSupportActionBar().getThemed
Context(),
                    R.array.mid_selection,
android.R.layout.simple_spinner_dropdown_item); //  create the
adapter from a StringArray
            midSpinner.setAdapter(mSpinnerAdapter); // set the
adapter
            midSpinner.setOnItemSelectedListener(this);
            return true;  }
        return super.onCreateOptionsMenu(menu);  }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so
long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_help) {
            Intent i = new Intent(this, HelpActivity.class);
            startActivity(i);
            return true;
        }

        if(id == R.id.action_about){
            Intent i = new Intent(this, AboutActivity.class);
            startActivity(i);
            return true;
        }
        if(id == R.id.action_score) {
            Intent i = new Intent(this, ScoreActivity.class);
            startActivity(i);
            return true;
        }
       return super.onOptionsItemSelected(item);
    }

    /**
     * A placeholder fragment containing a simple view.
     */
    public static class PlaceholderFragment extends Fragment {
        /**
         * The fragment argument representing the section number for
this
         * fragment.
```

```
          */
        private static final String ARG_SECTION_NUMBER =
"section_number";

        /**
         * Returns a new instance of this fragment for the given
section
         * number.
         */
        public static PlaceholderFragment newInstance(int
sectionNumber) {
            PlaceholderFragment fragment = new
PlaceholderFragment();
            Bundle args = new Bundle();
            args.putInt(ARG_SECTION_NUMBER, sectionNumber);
            fragment.setArguments(args);
            return fragment;
        }

        public PlaceholderFragment() { }
        @Override
        public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                                 Bundle savedInstanceState) {
            int sectionNumber =
getArguments().getInt(ARG_SECTION_NUMBER);
            int layoutId = R.layout.fragment_main;
            switch (sectionNumber) {
                case 1:
                    layoutId = R.layout.fragment_subjects;
                    break;
                case 2:
                    layoutId = R.layout.fragment_subjects;
                    break;
                case 3:
                    layoutId = R.layout.fragment_settings;
                    break;
            }
            View rootView = inflater.inflate(layoutId, container,
                    false);

            if(sectionNumber == 1 || sectionNumber == 2) {
                String[] subjects_list = {"Data Mining and Data
Warehousing", "Design Patterns", "Linux Programming", "Cloud
Computing", "Software Testing Methods", "Computer Graphics" };
                ArrayAdapter<String> adapter = new
ArrayAdapter<String>(getActivity(),
android.R.layout.simple_list_item_1, subjects_list);
                ListView subList = (ListView)
rootView.findViewById(R.id.subjectsListView);

                subList.setAdapter(adapter);
                subList.setOnItemClickListener(selectSubject);

          }
            if(sectionNumber == 3) {
                String[] settings_list = {"Change Batch"};
                ArrayAdapter<String> adapter = new 
ArrayAdapter<String>(getActivity(), 
android.R.layout.simple_list_item_1, settings_list);
                ListView settingsList = (ListView) 
rootView.findViewById(R.id.settingsListView);

                settingsList.setAdapter(adapter);
                settingsList.setOnItemClickListener(selectBatch);
            }
            return rootView;
        }

        public AdapterView.OnItemClickListener selectSubject = new 
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapter, View v, 
int position, long id) {
                Intent i = new Intent(getActivity(), 
QuestionsListActivity.class);
                startActivity(i);
            }};

        public AdapterView.OnItemClickListener selectBatch = new 
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapter, View v, 
int position, long id) {
                Intent i = new 
Intent(getActivity(),SelectOptions.class);
                startActivity(i);
            }
        }; @Override
        public void onAttach(Context context) {
            super.onAttach(context);
            ((MainActivity) context).onSectionAttached(
                    getArguments().getInt(ARG_SECTION_NUMBER));
        }
    }

    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int 
pos,
                               long id) {
        // TODO Auto-generated method stub
        String mid = (String) parent.getItemAtPosition(pos);
        Toast.makeText(MainActivity.this, mid, 
Toast.LENGTH_SHORT).show(); }
   @Override
    public void onNothingSelected(AdapterView<?> parent) {
        // TODO Auto-generated method stub

    }
```

CHAPTER-6
TESTING

6.1 Test Cases:

A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. The test cases for our project are listed as follows

Test case#:1

Test Case #ID	Test Description	Test case Steps	Test data	Expected output	Actual output	Result
TC_1	Set the batch by providing correct input values	In "batch" activity set the input values for regulation ,branch year and semester	regulation: r13 branch: CSE Year:3 Semester:2 and click on submit button	Navigates to "select a Subject" activity	Navigates to "select a Subject" activity	Pass
TC_2	Set the batch by providing incorrect input values	In "batch" activity set the input values for regulation, branch, year and semester	regulation: r13 branch: ECE Year: 1 Semester:2 and click on submit button	An error message appears : "please select correct year and semester"	An error message appears : "please select correct year and semester"	Pass

Test case#:2

Test Case #ID	Test Description	Test case Steps	Test data	Expected output	Actual output	Result
TC_3	Choose the required mid-term.	Select "Mid" present on the action bar of "select a Subject" activity. A drop down list appears with the following options: ○ Mid-1 ○ Mid-2 And now you can select the required Mid-term.		The selected mid-term is displayed on the action bar of "choose a Subject "activity	The selected mid-term is displayed on the action bar of "choose a Subject "activity	Pass
TC_4	Choose the required mode	Click on the navigation drawer icon present on the action bar of "choose a Subject" activity, the navigation drawer appears with the following options:		The selected mode is displayed on the action bar of "choose a Subject "activity	The selected mode is displayed on the action bar of "choose a Subject "activity	Pass

		○ Learning mode ○ Revision mode ○ Settings now you can select the required mode				

Test case#:3

Test Case #ID	Test Description	Test case Steps	Test data	Expected output	Actual output	Result
TC_5	Choose a subject	The "select a subject" activity consists of list of subjects. The required subject can be selected from that list		navigates to "Questions list" activity	navigates to "Questions list" activity	Pass
TC_6	Reset the batch	Click on the navigation drawer icon present on the action bar of "choose a Subject" activity, the navigation drawer appears with the following options: ○ Learning mode ○ Revision mode ○ Settings Click on settings, now the "settings" fragment is loaded. Now ,select the batch option and it navigates to "Batch" activity wherein the new input values of batch is set	regulation: r13 branch: CSE Year: 3 Semester:2 and click on submit button	The new values to batch is set successfully and it navigates to respective "choose a subject" activity	The new values to batch is set successfully and it navigates to respective "choose a subject" activity	Pass

Test case#:4

Test Case #ID	Test Description	Test case Steps	Test data	Expected output	Actual output	Result
TC_7	View the score	click on the action bar overflow menu present in the "choose a subject" activity, the following options appear: ○ Score ○ Help ○ About		The score is viewed successfully in the "Score" activity	The score is viewed successfully in the "Score" activity	Pass

		Select "Score" option. Now the "Score" fragment is loaded, wherein the score is viewed				
TC_8	View the help document	click on the action bar overflow menu present in the "choose a subject" activity ,the following options appear: ○ Score ○ Help ○ About Select "Help" option. Now the "Help" fragment is loaded, wherein the help documentation is viewed		The help documentation is viewed successfully in the "Help" activity	The help documentation is viewed successfully in the "Help" activity	Pass

Test case#:5

Test Case #ID	Test Description	Test case Steps	Test data	Expected output	Actual output	Result
TC_9	Select a multiple choice question (MCQ)	We can select multiple choice type of question by tapping on the "MULTIPLE CHOICE QUESTIONS" tab present in the "Questions List" activity. Now we can select a question from the list of multiple choice questions.		The selected question and its respective answer choices are loaded in the "Multiple Choice Questions" activity	The selected question and its respective answer choices are loaded in the "Multiple Choice Questions" activity	Pass
TC_10	Answer a multiple choice question (only for learning mode)	In "Multiple Choice Questions" fragment the question is and its answer choices are loaded .We can answer the question by tapping on the radio button of any one of the answer choices. (intended answer choice)		As soon as we tap on an answer choice, either of the following messages are displayed based on the selected answer choice "Correct!" or "Incorrect! Correct answer: X"	As soon as we tap on an answer choice, either of the following messages are displayed based on the selected answer choice "Correct!" or "Incorrect! Correct answer: X"	Pass

Test case#:6

Test Case #ID	Test Description	Test case Steps	Test data	Expected output	Actual output	Result
TC0011	go to next multiple choice question	After answering a Multiple choice question which is loaded in the "Multiple Choice Questions" activity, a "next" button appears below the question. Now click on it.		The next question and its respective answer choices are loaded in the "Multiple Choice Questions" activity	The next question and its respective answer choices are loaded in the "Multiple Choice Questions" activity	Pass
TC_12	View the correct answer choice for the multiple choice question, without attempting it (Revision mode)	Select the "Revision mode" present in the navigation drawer of the "choose a Subject" activity. And select a subject, it navigates to "questions lit" activity. Now tap on the "MUTIPLE CHOICE QUESTIONS" tab, and select a question under it. It navigates to "Multiple Choice Questions" activity.		In "Multiple Choice Questions" activity ,we can directly view the question and its respective correct answer choice	In "Multiple Choice Questions" activity ,we can directly view the question and its respective correct answer choice	Pass

Test case#:7

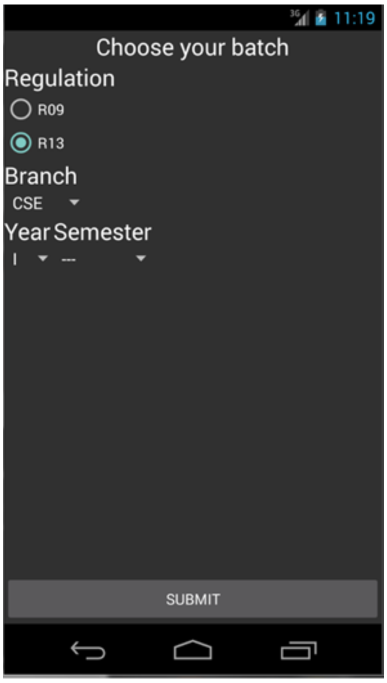
Test Case #ID	Test Description	Test case Steps	Test data	Expected output	Actual output	Result
TC0013	Select a fill in the blanks question	We can select multiple choice type of question by tapping on the "FILL IN THE BLANKS" tab present in the "Questions List" activity. Now we can select a question from the list of fill in the blanks questions.		The selected question and its respective blanks to answer are loaded in the "Fill in The Blanks" activity	The selected question and its respective blanks to answer are loaded in the "Fill in The Blanks" activity	Pass
TC0014	answer a fill in the blank question (only for learning mode)	The question and its respective blanks to answer are loaded in the "Fill in The Blanks" activity. We can answer the fill in the blank questions by manually entering the	1. xxxxxx 2. xxxxxx	The correct answers to the question is displayed along with either of the following messages based on	The correct answers to the question is displayed along with either of the following messages based on the	Pass

		answer in the blanks provided and click on submit button		the answer entered manually "Correct!" or "Incorrect!"	answer entered manually "Correct!" or "Incorrect!"	

Test case#:8

Test Case #ID	Test Description	Test case Steps	Test data	Expected output	Actual output	Result
TC_15	Go to next fill in the blank question	After answering the fill in the blank question which is loaded in the "Fill in The Blanks" activity, a "next" button appears below the correct answers. Click on it.		The next question and its respective blanks to answer are loaded in the "Fill in The Blanks" activity	The next question and its respective blanks to answer are loaded in the "Fill in The Blanks" activity	Pass
TC_16	View the correct answer for the Fill in the blank questions without attempting it (Revision mode)	Select the "Revision mode" present in the navigation drawer of the "choose a Subject" activity. And select a subject, it navigates to "questions lit" activity. Now tap on the "FILL IN THE BLANKS" tab, and select a question under it. It navigates to "Fill in The Blanks" activity.		In "Fill in The Blanks" activity, we can directly view the question and its respective correct answer (without manually answering the question)	In "Fill in The Blanks" activity, we can directly view the question and its respective correct answer (without manually answering the question)	Pass

SCREEN SHOTS



Choose your batch

Regulation

○ R09

● R13

Branch

CSE

Year Semester

I ---

Fig 7.1 Batch activity

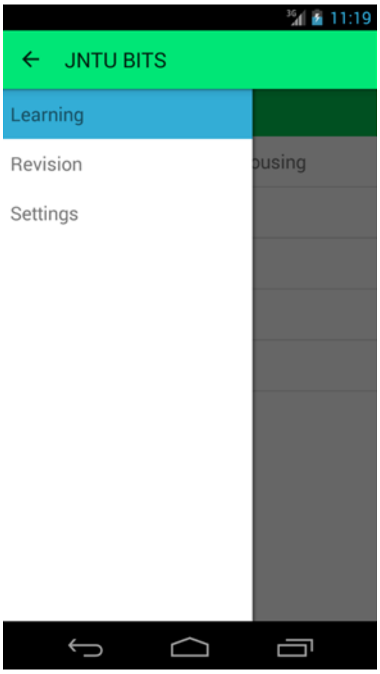
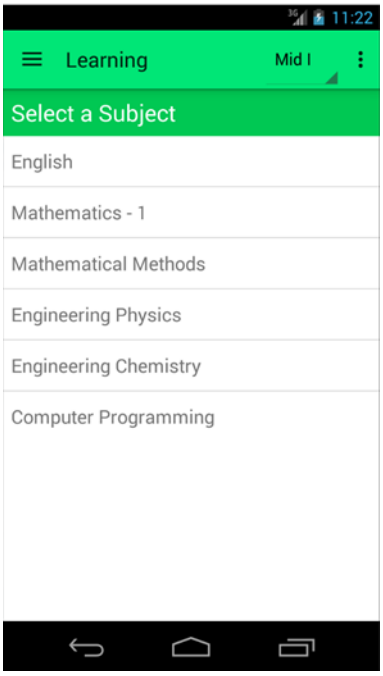


Fig 7.2 Navigation drawer window



Select a Subject

English

Mathematics - 1

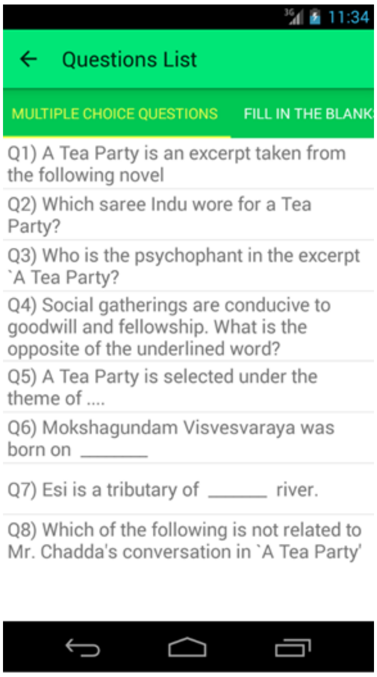
Mathematical Methods

Engineering Physics

Engineering Chemistry

Computer Programming

Fig 7.3 Select a Subject activity



Q1) A Tea Party is an excerpt taken from the following novel

Q2) Which saree Indu wore for a Tea Party?

Q3) Who is the psychophant in the excerpt `A Tea Party?

Q4) Social gatherings are conducive to goodwill and fellowship. What is the opposite of the underlined word?

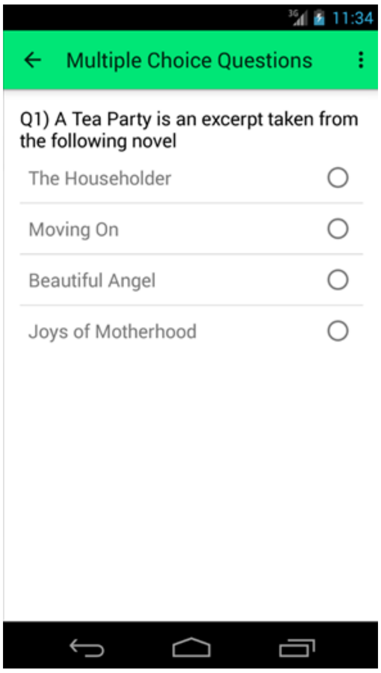
Q5) A Tea Party is selected under the theme of ....

Q6) Mokshagundam Visvesvaraya was born on _______

Q7) Esi is a tributary of ______ river.

Q8) Which of the following is not related to Mr. Chadda's conversation in `A Tea Party'

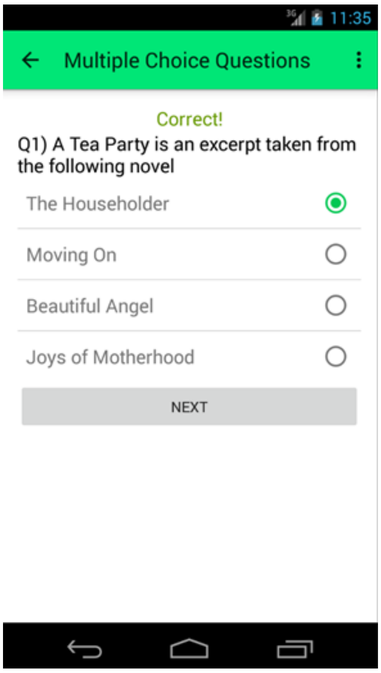
Fig 7.4 Questions list activity



Q1) A Tea Party is an excerpt taken from the following novel

- ○ The Householder
- ○ Moving On
- ○ Beautiful Angel
- ○ Joys of Motherhood

Fig 7.5 Multiple Choice Questions Activity

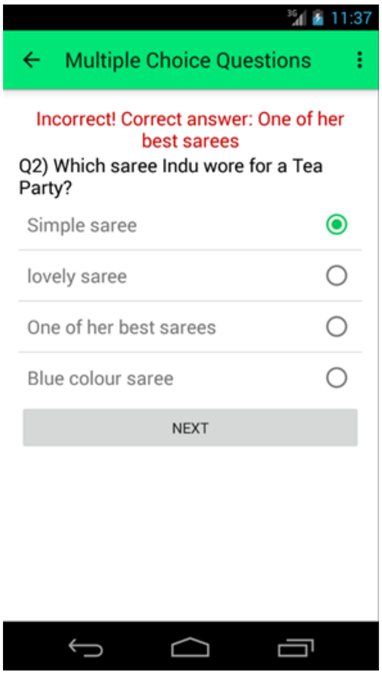


Correct!

Q1) A Tea Party is an excerpt taken from the following novel

- ● The Householder
- ○ Moving On
- ○ Beautiful Angel
- ○ Joys of Motherhood

Fig 7.6

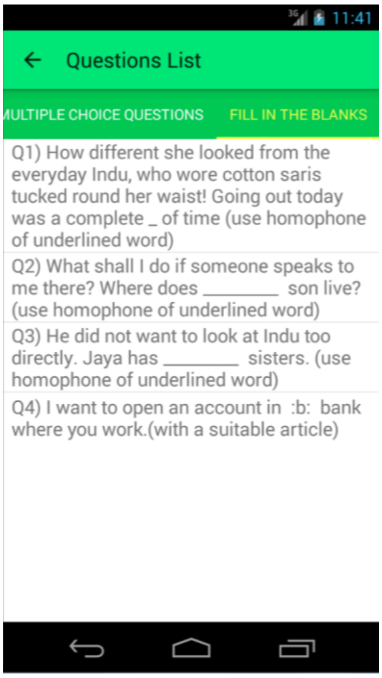


Incorrect! Correct answer: One of her best sarees

Q2) Which saree Indu wore for a Tea Party?

- Simple saree ●
- lovely saree ○
- One of her best sarees ○
- Blue colour saree ○

Fig 7.7



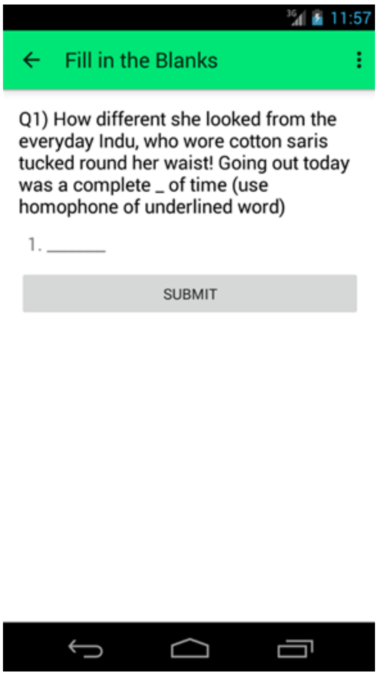
Q1) How different she looked from the everyday Indu, who wore cotton saris tucked round her waist! Going out today was a complete _ of time (use homophone of underlined word)

Q2) What shall I do if someone speaks to me there? Where does ________ son live? (use homophone of underlined word)

Q3) He did not want to look at Indu too directly. Jaya has ________ sisters. (use homophone of underlined word)

Q4) I want to open an account in :b: bank where you work.(with a suitable article)

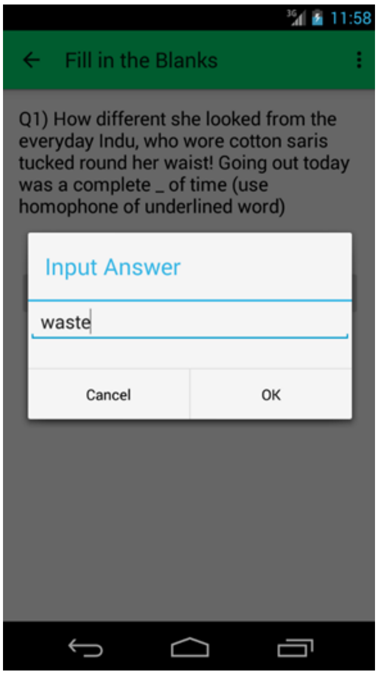
Fig 7.8



Q1) How different she looked from the everyday Indu, who wore cotton saris tucked round her waist! Going out today was a complete _ of time (use homophone of underlined word)

1. ______

Fig 7.9 Fill in the blanks activity

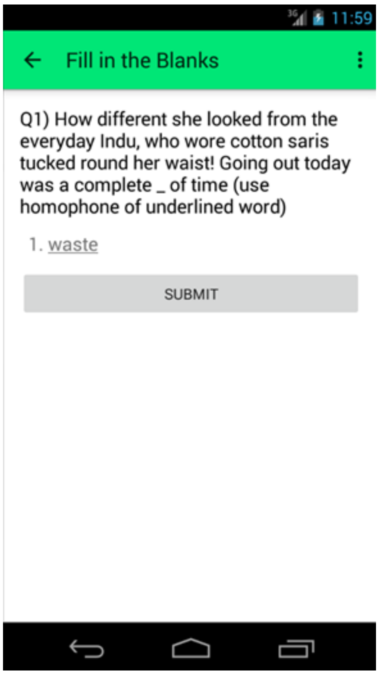


Q1) How different she looked from the everyday Indu, who wore cotton saris tucked round her waist! Going out today was a complete _ of time (use homophone of underlined word)

Input Answer

waste

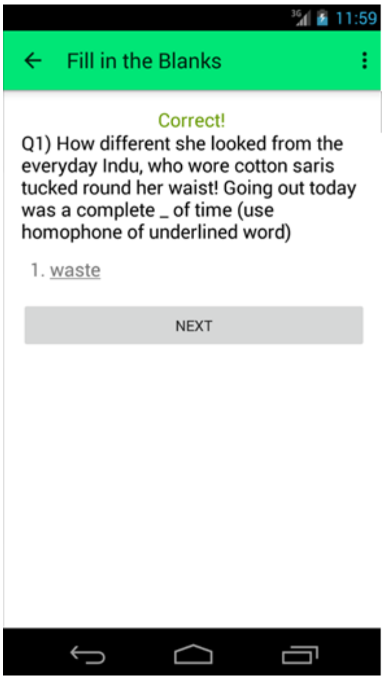
Fig 7.10



Q1) How different she looked from the everyday Indu, who wore cotton saris tucked round her waist! Going out today was a complete _ of time (use homophone of underlined word)

1. waste

Fig 7.11



Correct!

Q1) How different she looked from the everyday Indu, who wore cotton saris tucked round her waist! Going out today was a complete _ of time (use homophone of underlined word)

1. waste

Fig 7.12

8. CONCLUSION AND FUTURE SCOPE

This application is developed on Android .It is used to overcome the difficulties faced by the students while preparing for the bit paper of internal examination. It introduces a modern way of learning JNTU bits to its users. The list of questions and keys present in the application are collected from reliable sources like textbooks, downloaded from JNTU website etc… hence the content provided by application is reliable. The user interface of this application is simple, hence it can be operated with ease. This application allows the students (users) to learn the bits in an effective manner and hence this will help them to improve their overall aggregate.

In future, we need extend this application such that it can be used by students of all branches under JNTU i.e. EEE, Civil Engineering, Metallurgical Engineering etc. so that majority of students can be benefitted. We can also add additional features to this application like marking the most important questions, practise test with time limit etc…At present if we want to update this application, we need to uninstall it and again install the new version of the application. In future process can be made easier by making it as an online application. Efficient security methods should be provided to the application as it becomes online. The application should be able to run on the forth coming versions of Android.

BIBILIOGRAPHY

[1] Zigurd Medneiks, Laird Dormin, G. Blake Meike, Masumi Nakamura, Programming Android (Oreilly and Associate Series), "O'Reilly Media, Inc.", 2012.

[2] J.F. DilMarzio, Android a programmers guide, McGraw Hill Professional, 14-Aug-2008.

[3] Dave Smith, Jeff Friesen, Android Recipes: A Problem-Solution Approach (SpringerLink : Bucher) third edition, Apress, pg. 466-522, 2014.

[4] Neil Smyth, Android Studio Development Essentials, eBookFrenzy, pg. 233-486, 2015.

[5] Jason Morris, Android User Interface Development: Beginner's Guide, Packt Publishing Ltd, 2011.

[6] developver.android.com

[7] www.androidhive.info

[8] www.vogella.com

[9] www.androidcentral.com

[10]stackoverflow.com

[11] Sunny Kumar Aditya, Vikash Kumar Karn, Android SQLite Essentials, Packt Publishing Ltd, 2014.

[12]. Anders Goransson, Efficient Android Threading: Asynchronous Processing Techniques for Android Applications, "O'Reilly Media, Inc.", 2012.

[13] Jim Wilson, Creating Dynamic UI with Android Fragments, Packt Publishing Ltd, 2013.

[14] Joseph Annuzzi Jr., Lauren Darcey, Shane Conder, Introduction to Android Application Development: Android Essentials (Developer's Library), Addison-Wesley, 2013.

APPENDIX A: TOOLS AND TECHNOLOGY

This Android application consists of list of questions and keys. The pdf files of question papers and its keys are downloaded from the JNTU website. The data present in these files should be converted into a particular format from which we can generate the SQLite statements, which can be put into the database of the application. This work can be done manually, but it is iterative and time consuming process. Hence we decided to automate this process of conversion. Two python scripts are written that will complete this process in very less time. It is shown diagrammatically below:

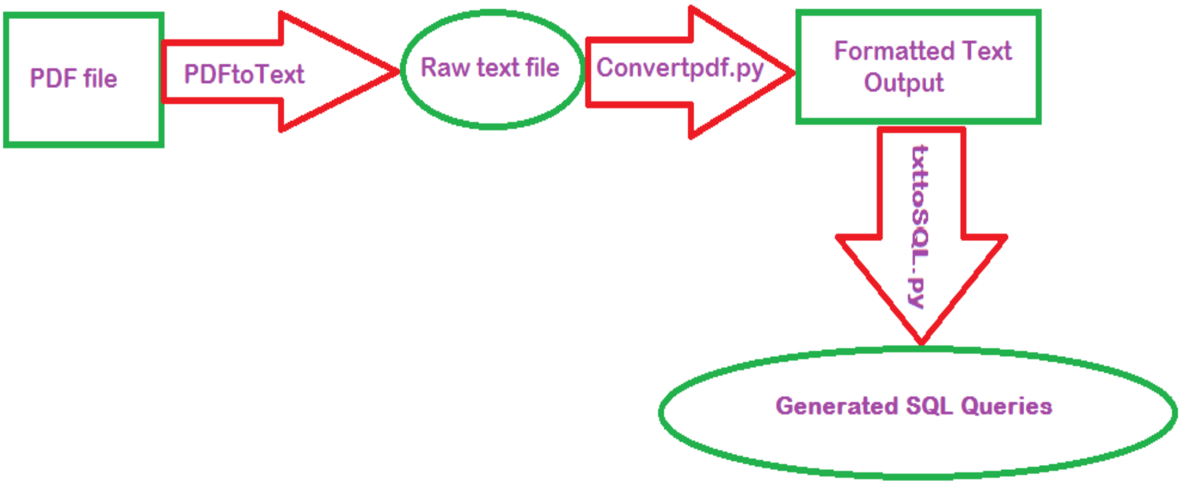


Fig 8.1 Process of Data Conversion

Auto pre-process:

```
#Rev 2.0
import re, sys
if len(sys.argv) != 2:
    print "Need one cmd line arg; filename"
    sys.exit()
fname = sys.argv[1]
try:
    f = open(fname, "r")
except IOError:
    print "Cannot open '"+ fname+ "'. Check filename spelling. Does
the file exist?"
```

```
    sys.exit()
num = "~@~"
brackets = "%#%"
option = "!$!"
cleanedString = ""
for line in contents:
    line = re.sub(r'^\s*[0-9]{1,2}[ \t]*\.[ \t]*', num, line, re.M)
    line = re.sub(r'\[\s*\]', brackets,  line)
    if re.match(r'^[a-dA-D]\)', line, re.M):
        line = re.sub(r'[a-dA-D]\)(?!(\s)*($|\n))', option, line)
    if re.match(r'^\([a-dA-D]\)', line, re.M):
        line = re.sub(r'[a-dA-D]\.(?!(\s)*($|\n))', option, line)
    line = re.sub(r'(_|\-){2,}', " :b: ", line)
cleanedString = cleanedString.replace(num, "\n")
cleanedString = cleanedString.replace(brackets, "\n")
cleanedString = cleanedString.replace(option, "\t")
```

Convert pdf:

```
#Rev 2.0
import os, re, sys
#Convert to text file
fname = ""
debug = False
if len(sys.argv) >= 2:
    fname = sys.argv[1]
    if len(sys.argv) > 2:
        if sys.argv[2] is "1":
            debug = True
```

#process the file (one set only)

```
     i = contents.index("1.")
     contents = contents[i:]
     contd = "##!@!##"
     o = "##ooo##"
     contents = re.sub(r'((c|C)on.*\d)',contd, contents)
     contents = re.sub(r'(-|~)+[oO]{3}(-|~)+', o, contents)
```

```
#Trimming loop
    for line in contents:
        if o in line or endFlag is True:
            break
        if "20." in line:
            endFlag = True
        if line.strip() is not empty:
#Making sure answers stay in one line
    temp = output.split('\n')
    #Start
    final = ""
    qFlag = False
    aFlag = False
    addbrackets = False
    qcount = 0

#for line in temp:
        if re.match(r'^11(.|\))', line, re.M):
        if re.match(r'^\s*[0-9]{1,2}\s*[\.|\)]\s*', line, re.M):

 #print "Question", line
            qFlag = True
            aFlag = False
        if re.match(r'^\(?(a|A)(\.|\))', line, re.M):
        if qcount > 1 or acount > 1:

#more than one line for question, shift the brackets to end of line
            final = re.sub(r'\[\s*\]\s*$', "" , final, re.M)
           addbrackets = True
            line = " " + line

#End
    if debug:
        txtName = "clean_" + txtName
    f = open(txtName, "w")
    #f.write(output)
    f.write(final)
```

```
f.close()
#Now cleaning the output
os.system("python  "  +  os.getcwd()+r'\autopreprocess.py  "'  +
txtName + '"')
```

We should make use of this tool (application) that is “pdftotext” to perform the above conversion process. pdftotext is an open source command-line utility for converting PDF files to plain text files —i.e. extracting text data from PDF-encapsulated files. It is freely available and included by default with many Linux distributions, and is also available for Windows as part of the Xpdf Windows port. Such text extraction is complicated as PDF files are internally built on page drawing primitives, meaning the boundaries between words and paragraphs often must be inferred based on their position on the page. This tool is downloaded from this site http://www.foolabs.com/xpdf/download.html. There are various other sites from which it can be downloaded.