Android based Online Quiz Application

Submitted By

Sk. Imran Hossain Shoyeb

ID: 2011296004

A Project Submitted in the partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering
East West University, Dhaka-1212, Bangladesh
May 2015

ABSTRACT

Modern hand held devices such as smart phones have become increasingly powerful in recent years. However, there are some applications that allow users to flexibly execute tasks which are done by personal computer (PC), laptop etc. As mobile devices become more like PCs they will come to replace objects to accomplish necessary tasks. If any mobile applications has developed to mitigate administrative work as well as fulfill user (other than administrator) requirement, then task can be complete within the smart phone. Online Quiz application, which is developed for Android base platform falls into this category.

The prime objective of “Online Quiz Application” is to take quiz for any individuals through internet. Multiple choice questions (MCQ) will arrive for any interviewee with certain time limit for each quiz. All questions, answers and timer must be configure by an administrator and these administrative tasks including user creation can be done from “Online Quiz Application”. Besides, report will generate with score where administrator can check for interviewee’s result.

DECLARATION

I hereby, declare that all the work presented in this project is the outcome of the Project performed by me under the supervision of Md. Shamsujjoha, Lecturer, Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh. I also declare that neither it nor part of it has been submitted for the requirement of any degree or diploma or for any other purposes except for publications.

Countersigned

… … … … … … … … …

(Md. Shamsujjoha

Lecturer

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh)

Signature

… … … … … … … … … … …

Sk. Imran Hossain Shoyeb

LETTER OF ACCEPTANCE

The project entitled “Android Based Online Quiz Application” is submitted by Sk. Imran Hossain Shoyeb, Id: 2011-2-96-004 to the department of Computer Science and Engineering, East West University, Dhaka 1212, Bangladesh is accepted by the Department for the partial fulfillment of the requirements for the degree of MS in Computer Science and Engineering.

Dr. Shamim H. Ripon
Associate Professor
Chairperson
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh

Md. Shamsujjoha
Lecturer
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh

ACKNOWLEDGEMENT

I am truly grateful to Almighty Allah, whose blessings have always been enormous and who gave me the ability and strength to complete this project. I would like to dedicate my project to my parents, who enlightened me the value of education and always keep me on the right track. It is a great honor and pleasure for me to record this deep sense of gratitude and insightful indebtedness to my respected supervisor, Md. Shamsujjoha for his valuable contribution, constant guidance, intuitive advice, helpful criticism, valuable suggestions, commendable support, and also endless patience for the completion of this project work. I am very much grateful to him and feel proud to have worked with him because it was not possible for me to complete this work without his inspiring enthusiasm and encouragement.

ABBREVIATIONS AND ACRONYMS

OS	**O**perating **S**ystem
UI	**U**ser **I**nterface
MCQ	**M**ultiple **C**hoice **Q**uestion
HTTP	**H**yper **T**ext **T**ransfer **P**rotocol
RAM	**R**andom **A**ccess **M**emory
JDK	**J**ava **D**evelopment **K**it
APT	**A**nnotation-**P**rocessing **T**ool
SDK	**S**oftware **D**evelopmen**t K**it
API	**A**pplication **P**rogram **I**nterface
HAXM	**I**ntel **H**ardware **A**ccelerated **E**xecution **M**anager
HTML	**H**ypertext **M**arkup **L**anguage
CGI	**C**ommon **G**ateway **I**nterface
IDE	**I**ntegrated **D**evelopment **E**nvironment
EER	**E**nhanced **E**ntity **R**elationship

TABLE OF CONTENTS

Chapter 3:Implementation

Chapter 4:User Manual

Chapter 5: Conclusion and Future Works

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Overview

In today's world, Smart phones have changed our lives and have become an indispensable part of our lives because of its specialty to simplify our routine work and thereby saving our time. A Smartphone with an Android OS offers excellent functionality to the users offering a distinct experience. Android is a Linux based operating system and it was bought by Google in 2007.There are tons of application available and one of the prime reason for this vast number is android being an open source. On the otherhand, android based device like mobile, tab are very user friendly. A survey has done by "LightCastle Partners" research wing which indicates that though other operating system mobile users exist but the majority users are goes with android operating system [1].

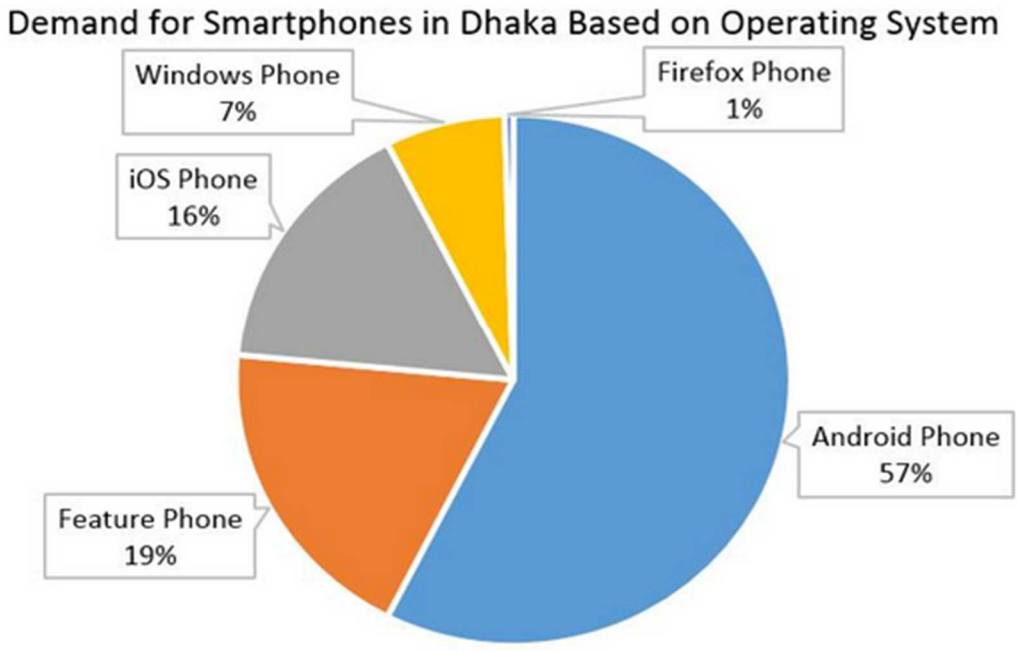

Figure 1.1: A survey result on a random sample of different age groups of mobile phone users in Dhaka, based on their preferred operating systems [1].

In this context, Project application is developed based on android platform. The name of application is define as **'Lets Quiz'**. Aims of this project is to develop an android platform supported Online Quiz application named "Lets Quiz". It is an online quiz application where user with admin privilege can do administrative task like add, delete and edit from application user interface (UI) and interviewee can participate for MCQ quiz with time limit. So, goals of this project to facilitate users to configure quizzes as well as giving quizzes with this android based smart phone. User friendly environment configuration is also another goal for this project application.

1.2 Objectives

The main objective of "Lets Quiz" is to facilitate a user friendly environment for all users and reduces the manual effort. In past days quiz is conducted manually but in further resolution of the technology we are able to generate the score and pose the queries automatically. The functional requirements include to create users that are going to participate in the quiz, automatic score and report generation and administrative tasks like add, delete, update for admin privilege users. In this application, all the permissions lies with the administrator i.e., specifying the details of the quiz with checking result will show to interviewee or not, addition of question and answers, marks for each question, Set timer for each quiz and generate report with score for each quiz.

1.3 Motivation

Currently most of the Examination like organizational recruitment, University class test are paper based, which costs time and resources. Questionnaire is developed, printed, and then collect data, entry, editing, cleaning, which time consuming and costly. Proposed application is the starting for avoid those circumstances which are been currently faced by any organization.

1.4 Book Organization

The Books has fiver chapters. Different chapters contain various table and figure which are relevant with this online Quiz project. The chapters are respectively Introduction, Proposed Model, Implementation, User Manual and conclusion. Introduction has described the objective of this online quiz application project. Second chapter, Proposed Model describe the model design for online quiz application. Third Chapter, Implementation describe the technical configuration which has done for this project. Fourth Chapter, User Manual describe the guideline for any UI user. Last chapter, Conclusion describe the summary of this literature. References and sample Code are attached at the end of this book.

1.5 Summary

Dramatic breakthroughs in processing power along with the number of extra features included in mobile devices have opened the doors to a wide range of commercial possibilities. In particular, most cell phones regularly include processors comparable to PCs and internet access from a few years ago. With all these added abilities, Online Quiz application is design for Android based system mobile.

Chapter 2

Proposed Model

2.1 Purpose of the project

This Project main purpose is to develop Online Quiz system named **'LETS QUIZ'.** The application (LETS QUIZ) will provide online based quiz with multiple choice question (MCQ). This quiz application will support android base operating system. With this application, users or any organization can perform actions like

- Administrative Task
- Interviewee Task

2.1.1 Administrative Task

Addition, Deletion and update for questions, answers are the main part of administrative task. Quizzes, Users also can add, delete and update from user interface (UI). As a result, the following tasks are define as administrative task.

- Log in to the application as administrative privilege
- Add, Delete, Edit/Update information for User
- Add, Delete, Edit/Update information for Quiz
- Add, Delete, Edit/Update Question and Answer
- View Result

2.1.2 Interviewee Task

Select any Quiz or subject which he /she wants to give Examination. One interviewee can give Quiz only for one time. Once a quiz has finished, it become

inactive to that user. Finally, Score can be shown considering the quiz has been taken from any individuals. As a result, the following tasks are define as interviewee task.

- Log in to the system as Interviewee privilege
- Select Desire Quiz
- Answer questions within set time (by admin)
- Finish Quiz

2.2 Client Server Architecture

Following figure 2, shows the client-server view for 'LETS QUIZ' online application. Administrator and Interviewee user login to the application from any android platform which is define as Client side. On the otherhand, application server or web server and Database server are define as server site. Application server communicates with the database where all the information are store. Client side and server side communicate between them with Http/Https.

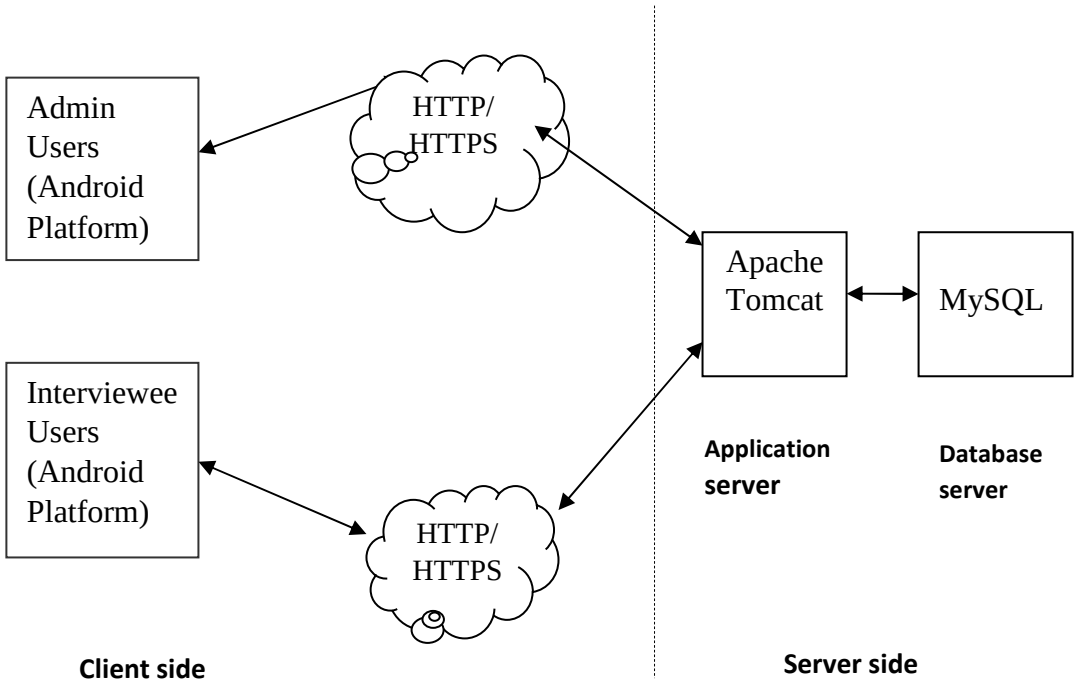

Figure 2.1: Client and Server Connectivity Diagram for 'LET'S QUIZ'

2.3 Flow Chart

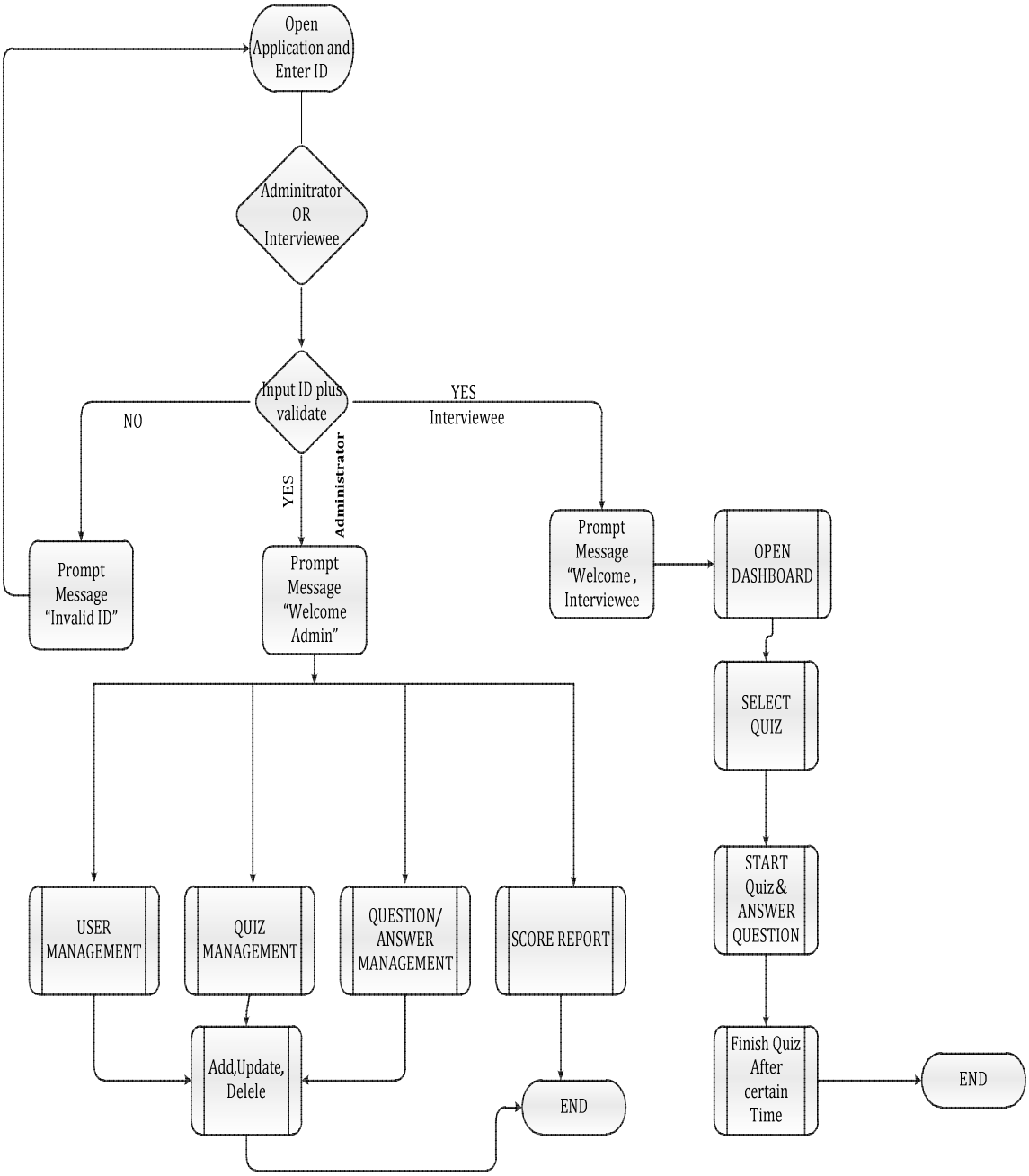

Figure 2.2: Flow Chart for 'LET'S QUIZ'

2.3.1 Description for flow chart

1. After login with username and password, Role for that particular user will verify from server database.
2. If user is administrator, following options shall be managed.
    - Quiz
    - Question and Answer
    - Report
    - Users

   Above four options can be add, edit and delete by administrator.

3. Score report will show to administrator in read only mode.
4. If user is interviewee, all active quiz list will enable for giving test.
5. Quiz will execute with certain time limit. Quiz will finish after that time or all question answered. .

2.4 Summary

Both administrative and others work can be execute through user interface. Add, delete, edit are belong to administrative work. Interviewee users can sit for any quiz. One quiz appear only one time to any interviewee user. Timer has set to each Quiz and Score will visible if it is set by administrator who create that quiz.

Chapter 3

Implementation

3.1 Technologies Used

- Microsoft Windows 7
- Java Development Kit
- Node.js
- Android SDK
- Intel Hardware Accelerated Execution Manager (HAXM)
- Titanium SDK
- MySQL Database
- Java Script
- PHP

3.1.1 Microsoft Windows 7

It should be mentioned that all tools and technology are installed for development work at windows 7 operating system 64 bit platform. Windows 7 is a personal computer operating system developed by Microsoft. It is a part of Windows NT family of operating systems. Development of Windows 7 started as early as 2006 under the codename "Blackcomb." Windows 7 was released to manufacturing on July 22, 2009, and became generally available on October 22, 2009, less than three years after the release of its predecessor, Windows Vista. Minimum hardware requirements for Windows 7 is given below [2].

Component	**Operating system architecture**	
	32-bit	**64-bit**
Processor	1 GHz IA-32 processor	1 GHz x86-64 processor
Memory (RAM)	1 GB	2 GB
Graphics card	DirectX 9 graphics processor with WDDM driver model 1.0 (Not absolutely necessary; only required for Aero)	
Free hard drive space	16 GB	20 GB
Optical drive	DVD-ROM drive (Only to install from DVD-ROM media)	

Table I: Windows 7 Minimum Hardware Requirement [2]

The maximum amount of RAM that Windows 7 supports varies depending on the product edition and on the processor architecture, as shown below figure

<table>
<tr><th rowspan="2">Edition</th><th colspan="2">Processor architecture</th></tr>
<tr><td>IA-32 (32-bit)</td><td>x64 (64-bit)</td></tr>
<tr><td>Ultimate</td><td rowspan="5">4 GB</td><td rowspan="3">192 GB</td></tr>
<tr><td>Enterprise</td></tr>
<tr><td>Professional</td></tr>
<tr><td>Home Premium</td><td>16 GB</td></tr>
<tr><td>Home Basic</td><td>8 GB</td></tr>
<tr><td>Starter</td><td>2 GB</td><td></td></tr>
</table>

Table II: Physical memory limits of Windows 7 [2]

3.1.2 Java Development Kit (JDK)

A Java Development Kit (JDK) is a program development environment for writing Java applets and applications. It consists of a runtime environment that "sits on top" of the operating system layer as well as the tools and programming that developers need to compile, debug, and run applets and applications written in the Java language[3]. JDK version 1.8.0_25 is used in this project.

The JDK has as its primary components a collection of programming tools, including [3]:

appletviewer – this tool can be used to run and debug Java applets without a web browser

apt – the annotation-processing tool

extcheck – a utility which can detect JAR-file conflicts

idlj – the IDL-to-Java compiler. This utility generates Java bindings from a given Java IDL file.

jabswitch – the Java Access Bridge. Exposes assistive technologies on Microsoft Windows systems.

java – the loader for Java applications. This tool is an interpreter and can interpret the class files generated by the javac compiler. Now a single launcher is used for both development and deployment. The old deployment launcher, jre, no longer comes with Sun JDK, and instead it has been replaced by this new java loader.

javac – the Java compiler, which converts source code into Java bytecode

javadoc – the documentation generator, which automatically generates documentation from source code comments

jar – the archiver, which packages related class libraries into a single JAR file. This tool also helps manage JAR files.

javafxpackager – tool to package and sign JavaFX applications

jarsigner – the jar signing and verification tool

javah – the C header and stub generator, used to write native methods

javap – the class file disassembler

javaws – the Java Web Start launcher for JNLP applications

JConsole – Java Monitoring and Management Console

jdb – the debugger

jhat – Java Heap Analysis Tool (experimental)

jinfo – This utility gets configuration information from a running Java process or crash dump. (Experimental)

jmap – This utility outputs the memory map for Java and can print shared object memory maps or heap memory details of a given process or core dump. (Experimental)

jmc – Java Mission Control

jps – Java Virtual Machine Process Status Tool lists the instrumented Hotspot Java Virtual Machines (JVMs) on the target system. (Experimental)

jrunscript – Java command-line script shell.

jstack – utility which prints Java stack traces of Java threads (experimental)

jstat – Java Virtual Machine statistics monitoring tool (experimental)

jstatd – jstat daemon (experimental)

keytool – tool for manipulating the keystore

pack200 – JAR compression tool

policytool – the policy creation and management tool, which can determine policy for a Java runtime, specifying which permissions are available for code from various sources

VisualVM – visual tool integrating several command-line JDK tools and lightweight performance and memory profiling capabilities

wsimport – generates portable JAX-WS artifacts for invoking a web service.

xjc – Part of the Java API for XML Binding (JAXB) API. It accepts an XML schema and generates Java classes.

3.1.3 Node.js

Node.js is an open source, cross-platform runtime environment for server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js provides an event-driven architecture and a non-blocking I/O API that optimizes an application's throughput and scalability.Node.js uses the Google V8 JavaScript engine to execute code, and a large percentage of the basic modules are written in JavaScript [4]. To run all Titanium Studio components, Node.js must be installed. Node.js version 0.10.33 is used for this project.

3.1.4 Android SDK

The Android SDK (software development kit) is a set of development tools used to develop applications for Android platform. The Android SDK includes the following:

- Required libraries
- Debugger
- An emulator
- Relevant documentation for the Android application program interfaces (APIs)
- Sample source code
- Tutorials for the Android OS

Every time Google releases a new version of Android, a corresponding SDK is also released. To be able to write programs with the latest features, developers must download and install each version’s SDK for the particular phone [5].

API Level is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform. The Android platform provides a framework API that applications can use to interact with the underlying Android system. The framework API consists of:

- A core set of packages and classes
- A set of XML elements and attributes for declaring a manifest file
- A set of XML elements and attributes for declaring and accessing resources
- A set of Intents
- A set of permissions that applications can request, as well as permission enforcements included in the system

The API Level identifier serves a key role in ensuring the best possible experience for users and application developers:

- It lets the Android platform describe the maximum framework API revision that it supports
- It lets applications describe the framework API revision that they require
- It lets the system negotiate the installation of applications on the user's device, such that version-incompatible applications are not installed.

The table below specifies the API Level supported by each version of the Android platform [6].

Platform Version	API Level	VERSION_CODE
Android 5.1	22	LOLLIPOP_MR1
Android 5.0	21	LOLLIPOP
Android 4.4W	20	KITKAT_WATCH
Android 4.4	19	KITKAT
Android 4.3	18	JELLY_BEAN_MR2
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1
Android 4.1, 4.1.1	16	JELLY_BEAN
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH
Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1
Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.1	2	BASE_1_1
Android 1.0	1	BASE

Table III: API Level supported by each version of the Android platform [6]

Applications can use a manifest element provided by the framework API — *<uses-sdk>* — to describe the minimum and maximum API Levels under which they are able to run, as well as the preferred API Level that they are designed to support. The element offers three key attributes:

- *android:minSdkVersion* — Specifies the minimum API Level on which the application is able to run. The default value is "1".
- *android:targetSdkVersion* — Specifies the API Level on which the application is designed to run. In some cases, this allows the application to use manifest elements or behaviors defined in the target API Level, rather than being restricted to using only those defined for the minimum API Level.

- *android:maxSdkVersion* — Specifies the maximum API Level on which the application is able to run .

3.1.5 Intel Hardware Accelerated Execution Manager (HAXM)

Intel(r) HAXM is the Intel® Hardware Accelerated Execution Manager is a hardware-assisted virtualization engine (hypervisor) that uses Intel Virtualization Technology (Intel(r) VT) to speed up Android app emulation on a host machine. In combination with Android x86 emulator images provided by Intel and the official Android SDK Manager, HAXM allows for faster Android emulation on Intel VT enabled systems. The Intel HAXM driver runs inside the emulator as well as on the host machine. It runs on various versions of Windows, Linux, and Mac OS [7].The following platforms are supported by the Intel HAXM.

- Windows 8 and 8.1 (32/64-bit)
- Windows 7 (32/64-bit)
- Windows Vista (32/64-bit)

3.1.6 Titanium SDK

The Titanium SDK helps to build native cross-platform mobile application using JavaScript and the Titanium API, which abstracts the native APIs of the mobile platforms. Titanium empowers to create immersive, full-featured applications, featuring over 80% code reuse across mobile apps. Appcelerator licenses Titanium under the Apache 2 license and is free for both personal and commercial use [8].Titanium SDK compatibility With Android SDK has described following table [Table IV]

Titanium SDK Version	Min Target Android/SDK Version (android:targetSdkVersion)	Max Target Android/SDK Version (android:targetSdkVersion)	Minimum Android/SDK Version (android:minSdkVersion)
4.0.0	5.0.x (API 21)	5.0.x (API 21)	TBD
3.4.1 - 3.5.0	4.0.x (API 14)*	5.0.x (API 21)**	2.3.x (API 10)
3.3.0 - 3.4.0	4.0.x (API 14)*	4.4.x (API 19)	2.3.x (API 10)
3.2.0 - 3.2.3	2.3.x (API 10)*	4.4.x (API 19)	2.3.x (API 10)
3.1.2 - 3.1.3	2.3.x (API 10)	4.3.x (API 18)	2.3.x (API 10)
3.1.1	2.3.x (API 10)	4.2.x (API 17)	2.3.x (API 10)
3.1.0	2.2 (API 8)	4.2.x (API 17)	2.2 (API 8)
2.1.2 - 3.0.2	2.2 (API 8)	4.1.x (API 16)	2.2 (API 8)
2.0 - 2.1.1	2.2 (API 8)	4.0.x (API 15)	2.2 (API 8)
1.8.x	2.2 (API 8)	3.x.x (API 11)	2.2 (API 8)
1.7.x	2.1 (API 7)	3.x.x (API 11)	2.1 (API 7)

Table IV: Titanium SDK supported by each version of the Android SDK [10]

The Titanium SDK tools comprise a set of Node.js-based utilities and supporting tools that work with the native SDK tool chains. The Titanium tools combine JavaScript source code, a JavaScript interpreter, and static assets into an application binary that will be installed to an emulator or mobile device. Studio will manage almost every aspect of this build chain, leaving you to focus on building your apps [9].

3.1.7 Java Script:

JavaScript is an interpreted programming or script language from Netscape. JavaScript is influenced by Java, the syntax is more similar to C and is based on ECMAScript, a scripting language developed by Sun Microsystems [11]. The JavaScript code can produce an error message before any information is actually transmitted to the server. JavaScript is a client-side scripting language, which means JavaScript functions can run after a webpage has loaded without communicating with the server. It can also be referenced in a separate .JS file.

3.1.8 PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP code can be simply mixed with HTML code, or it can be used in combination with various engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client [13].

3.2 Tools Used

3.2.1 Titanium Studio

Titanium studio is IDE with titanium SDK open-source framework that allows the creation of mobile apps on platforms including iOS, Android, Windows Phone, BlackBerry OS, and Tizen from a single JavaScript codebase, developed by Appcelerator [12]. System environment must meet the following requirements to run Titanium Studio:

- **Operating System**
  - A recent version of Windows, OS X or Ubuntu
- **Memory**
  - 2 GB RAM (available memory, rather than total memory)
- **Java Runtime**
  - Oracle JDK (no other brand of Java is suitable)
- **Node.js**
  - Required for the Titanium command-line tools like the CLI, Alloy and Node.ACS.

Titanium's unique trait among the various available cross-platform mobile solutions is that it creates truly native apps. This is in contrast to web-based solutions that deliver functionality via an enhanced web view. Titanium, not wanting to be limited by the native web view, has engaged in a much deeper integration with the underlying

platforms. This gives Titanium developers the ability to leverage native UI components and services, as well as near-native performance, features you won't find in other cross-platform mobile development solution. In addition, you'll have access to other features like:

- Platform specific APIs
- Location-based services
- Social sharing
- Rich multimedia
- Online and on-device data
- Extensibility

Studio is Appcelerator's free IDE (integrated development environment). It can use to write, test, and debug mobile applications. Studio also has integrated templates and sample applications to make it even easier to get started creating your own apps. In addition, Studio will help to manage Titanium SDK updates and module usage. Titanium exists as a bridge between the native operating system and app's code. The following graphic illustrates this architecture [12]:

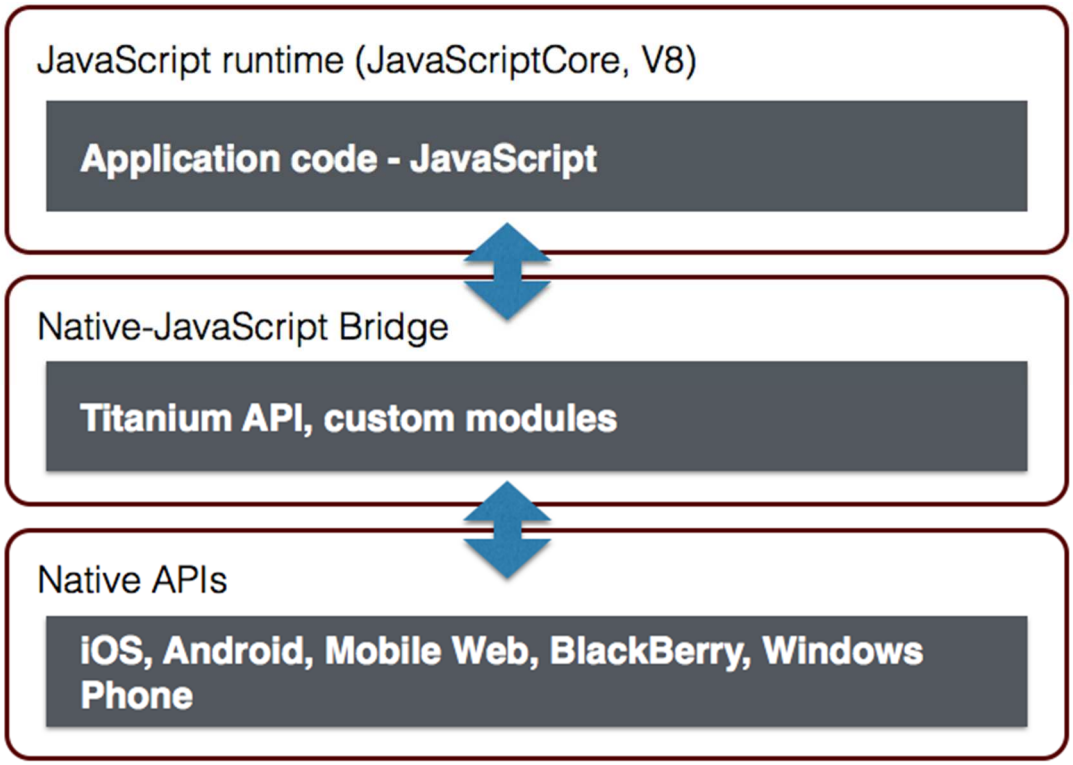

Figure 3.1: Titanium platform Existence between Code and Android OS [12]

At the bottom of the stack is the client operating system: Android, iOS, or the browser (for Mobile Web applications). At the top is desire app, built JavaScript. In between, is the Titanium SDK and the APIs it exposes. Application wrote in JavaScript, calling on the Titanium APIs to take actions like drawing buttons, opening windows, showing the camera, etc. The Titanium Bridge (part of the SDK) translates those calls into their native equivalents. In other words, when create a Titanium button, it's actually a proxy for a true native button. When you modify the Titanium button, say to change its label or add an event listener, Kroll applies corresponding changes to the native equivalent. When events occur in native-land, Kroll bubbles them up to your JavaScript code [l2].

3.3 System Design

3.3.1 Log in to the application

- Open the application
- Provide username and password. Username is define as the email address to concern user.
- After click on login button application communicate with server and verify whether the provided username and password is correct or not.
- If provided information found at server end and match with role ‘Admin’ or ‘Interviewee’ then open the desire portal. One role is define as ‘others’ for future expansion.
- If user is administrator the user, question and answer, Report, user button is enable.
- If user is interviewee, all quiz which are active will appear and he/she can select from those to start quiz.

Figure 3.2 describe the login mechanism design for this application.

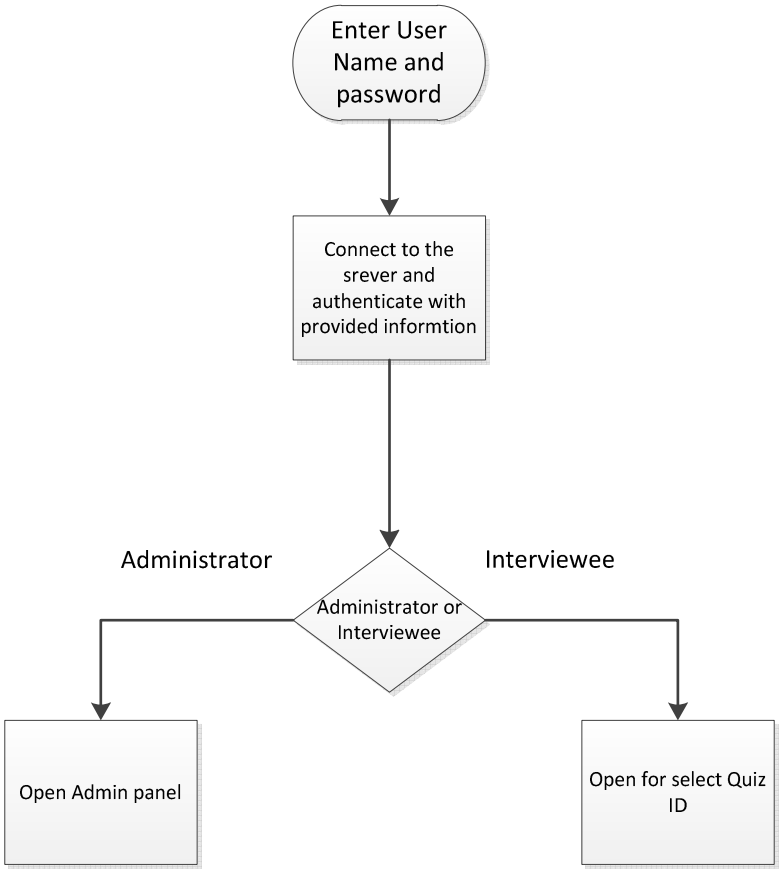

Figure 3.2: Login design

Password validation has configured with the following manner.

- should contain at least one digit
- should contain at least one lower case
- should contain at least one upper case
- should contain at least 6 from the mentioned characters

Email Validation has also configured.

3.3.2 Forget password

If any user forget his/her password then user should input username after send to server. If user found at database then system will generate a password and sent to concern user's email address.

This reset password will not be shown by anybody without the user (provided email to that user).

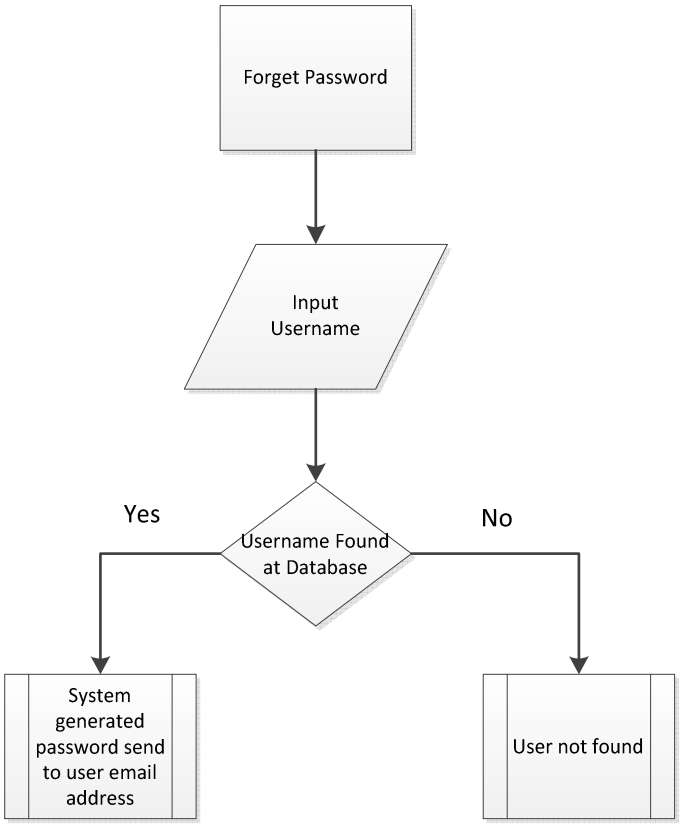

Figure 3.3: Forget Password System Design

3.3.3 Admin Panel

Administrator has privilege to access four options. They are

1. Quiz
2. Question and Answer
3. Report
4. User

1. Quiz

Administrator can add quiz with following four options. They are

- **Quiz Name**

  Name of the Quiz will define here.

- **Active or Inactive**

  Currently any specific quiz is active or not. If any quiz is inactive (update by administrator) then that quiz will not be shown to interviewee.

- **Show result or not**

  If show result option is mark with ‘yes’ then any individual interviewee will see score after finish Quiz. If ‘not’ mark, then can’t see the result.

- **Quiz Time**

  Total time for quiz can be set from here only by administrator.

Application will communicate with webserver API and connect with database to extract all quiz list and will show to administrator. Any quiz can be edit or delete from that list.

2. Question and Answers

Four multiple answer (max) can input by the administrator. Administrator can also set less than four multiple (i.e. two, three) answer for any question.

- **Question**

  Question will be written here.

- **Answers:**

  Four multiple answer (max) can input by the administrator. Administrator can also set less than four multiple (i.e. two, three) answer for any question.

- **Quiz Name:**

  Every active Quiz from database will be shown here which let administrator to choose any for add questions.

- **Order No:**

  Question order can be set here.

For Update or Delete any question, Administrator need to choose any quiz. All questions will visible to administrator for that particular quiz.

3. Report

Report will show for any particular quiz. Username of interviewee and score will be shown to administrator.

4. Users

Administrator can add quiz with following four options. They are

- **User Name**

  User name will be provided as the user's email address.

- **Role**

  Role will be define as 'Admin' or 'Interviewee'. If any user has 'Admin' role then he/she can set Application parameters. If any user has 'Interviewee' role then he/she can sit for any quiz.

- **Password**

  Set password for that particular user.

- **Confirm Password**

  Confirm password for that particular user.

After add any user, an automatic email to send to that user by system.

Application will communicate with webserver API and connect with database to extract all user list and will show to administrator. Any user can be edit or delete from that list.

Figure 3.4 described the system design of admin panel.

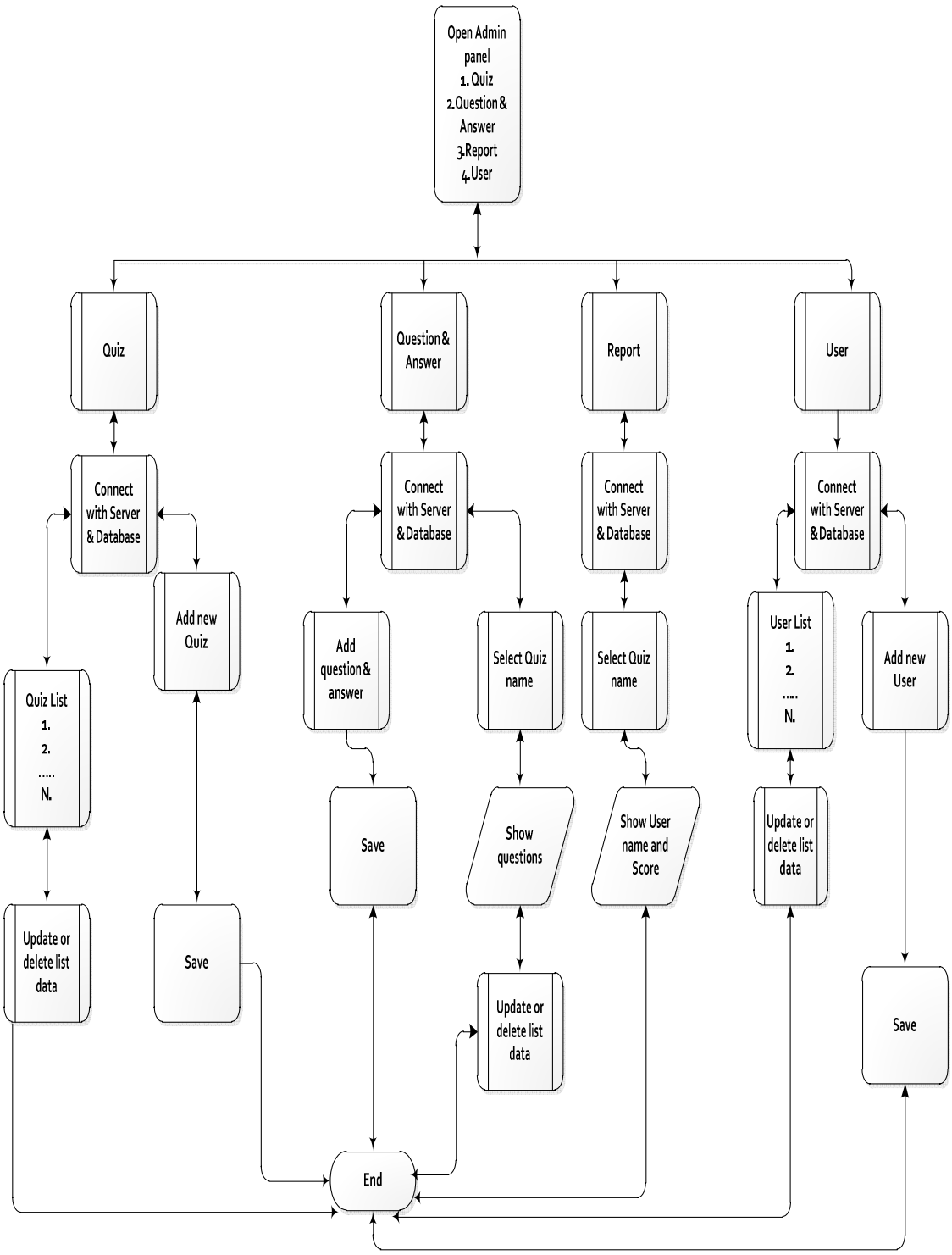

Figure 3.4: Admin Panel

3.3.4 Interviewee Panel

- Any candidate who have access to this application, will get the active quiz list after log in.
- Candidate will choose desire quiz to enter in the test.
- After finished any test, candidate can't not sit for that test for twice.
- Result (where M = Correct answers) will show (if its enable) after finish the quiz and will store to the database for specific candidates.

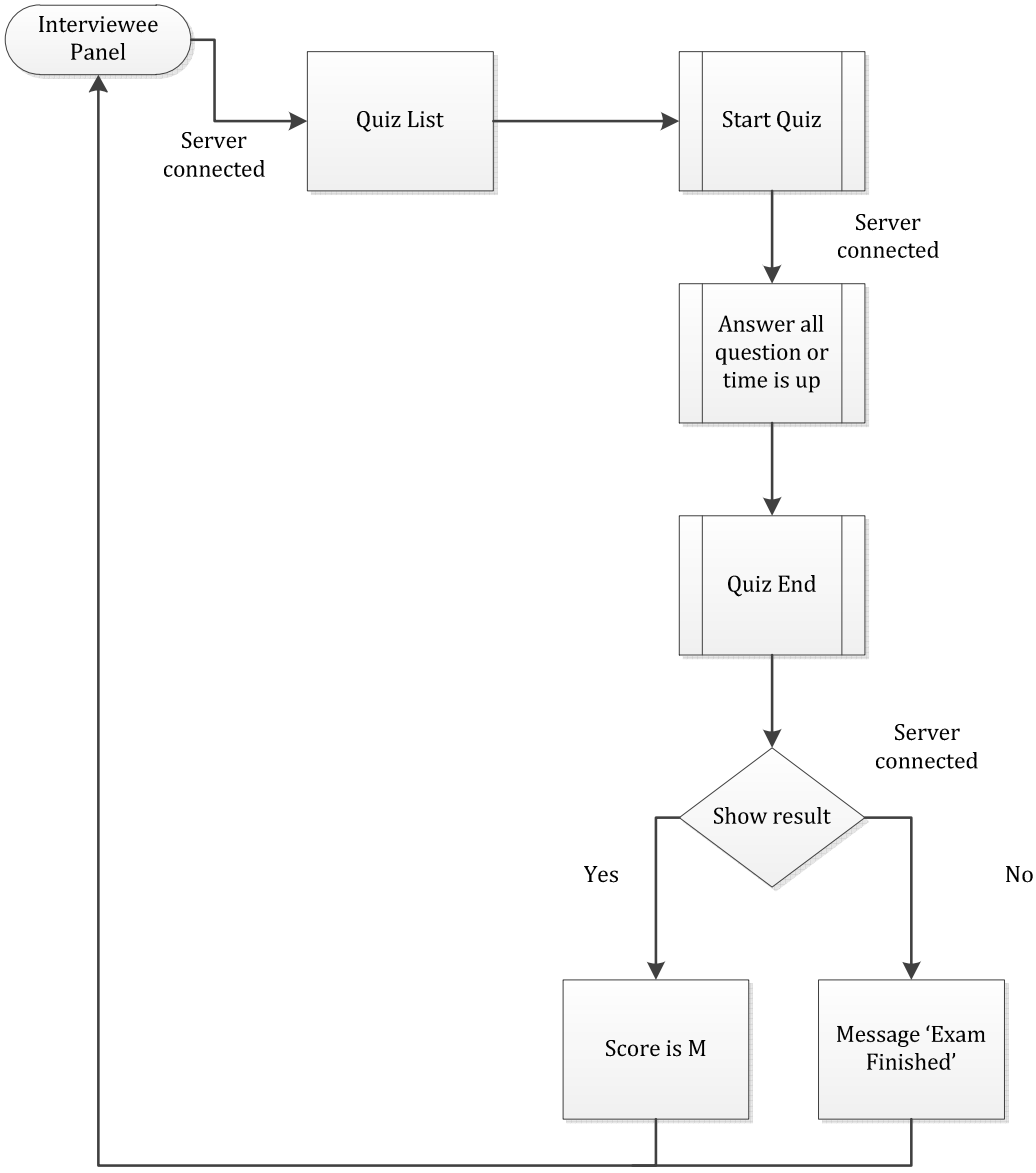

Figure 3.5: Interviewee Panel

3.4 Database Design

3.4.1 Table Schema

There are six tables configure for this application.

1. users
2. quiz
3. questions
4. answers
5. users_answers
6. user_quiz

users

Column	Type	Null
user_id	int(11)	No
email	varchar(255)	No
password	varchar(255)	No
role	varchar(100)	No

quiz

Column	Type	Null
quiz_id	int(11)	No
quiz_name	varchar(255)	No
active	int(11)	No
show result	int(11)	No
quiz_time	int(11)	No

questions

Column	Type	Null
ques_id	int(11)	No
question	varchar(255)	No
ques_order	int(11)	No
quiz_id	int(11)	No

answers

Column	Type	Null
ans_id	int(11)	No
ques_id	int(11)	No
answer	varchar(255)	No
is_correct	int(11)	No

users_answers

Column	Type	Null
uans_id	int(11)	No
ques_id	int(11)	No
quiz_id	int(11)	No
is_correct	int(11)	No
user_id	int(11)	No

user_quiz

Column	Type	Null
uquiz_id	int(11)	No
user_id	int(11)	No
quiz_id	int(11)	No
quiz_date	datetime	No

3.4.2 Table Relationship / EER Diagram

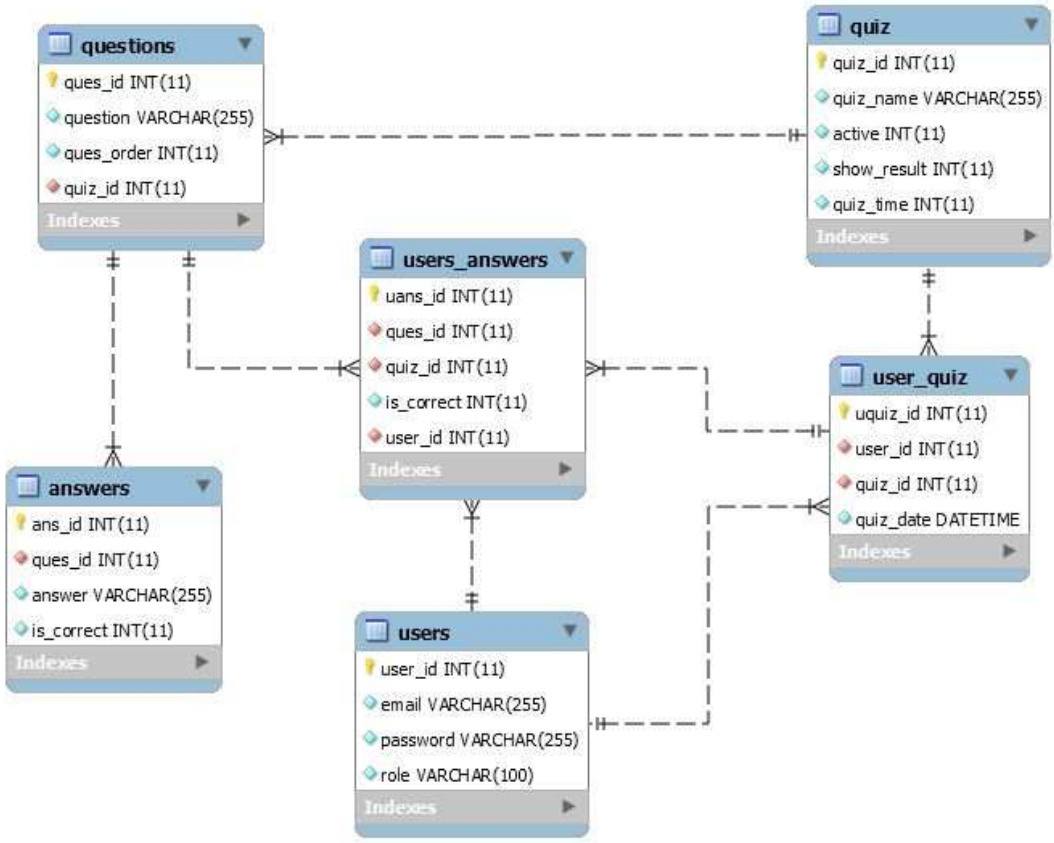

Figure 3.6: Database Table relationship

3.5 Summary

The Project is developed in Java Script dynamic programming language by using the “Titanium Studio” Integrated Development Environment (IDE). Titanium software development kit (SDK) and Android software development kit (SDK) integrated with Titanium studio to develop mobile applications on the Android platform. PHP (recursive acronym for PHP: Hypertext Preprocessor) is used for server side scripting which work as application program interface (API). MySQL Database is use for database configuration.

Chapter 4

User Manual

4.1 System Requirement

Device	Operating System (OS)	OS Version	RAM	Disk Space
Android Mobile/Tab	Android	4.0.x to higher	64 MB or higher	18 Mb

Table V: Minimum System Requirement

4.2 Prerequisite

- Internet Connection must active.
- Valid Username with password must require to use this application.

4.3 Installation

To install the application in android mobile device or tablet device user needed to run the setup package named as ‘letsquiz.apk’. This package can store both in android device memory and external memory card (micro SD card that connect with android device). Package need maximum 9 Mb space. After copy it to android device, following steps should be followed:

- Tap the folder where the ‘letsquiz.apk’ is store.
- Tap the application package ‘letsquiz.apk’.
- After selecting the package, application lets user to install
- Tap install
- Installation is done and application is ready for use.

4.4 Getting Started

Tap the icon to open the application.

4.4.1 Login

Login screen will appear. Enter username and password to the desire field. Hint text indicates where you want to input username and password. Tap the login button after type username and password.

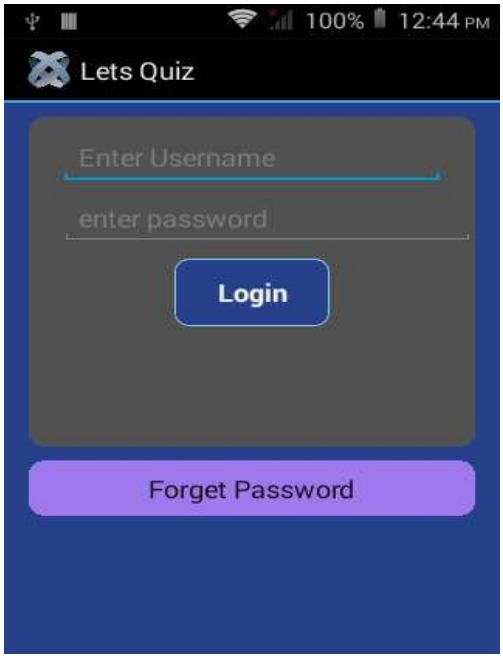

Figure 4.1: Login Form

4.4.2 Forget password

- If password has forgotten than tap the ‘forget password’ button.
- Another screen will appear where username need to input.
- After input username tap the button.
- Check the email address which one is tag with user ID. New password will sent to email.

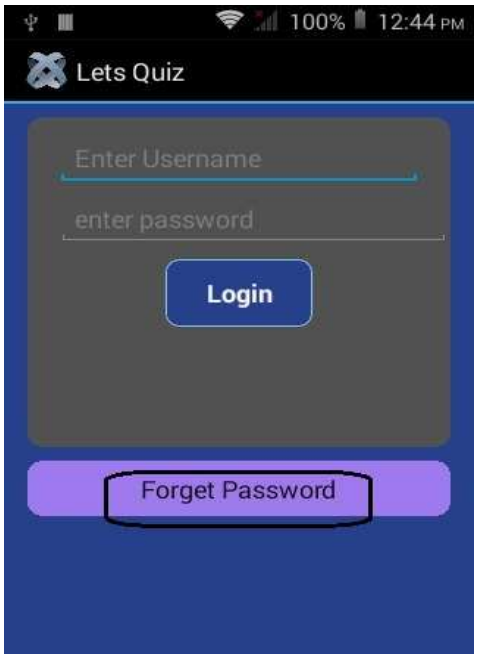

Figure 4.2: Forget Password

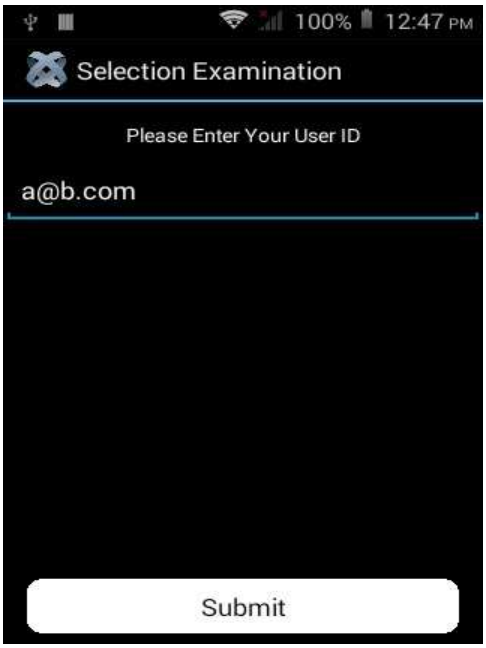

Figure 4.3: User ID Input

4.4.3Administrator

Figure4.4: Admin Panel Form

4.4.3.1 User Management

Tap on the ‘users’ button.

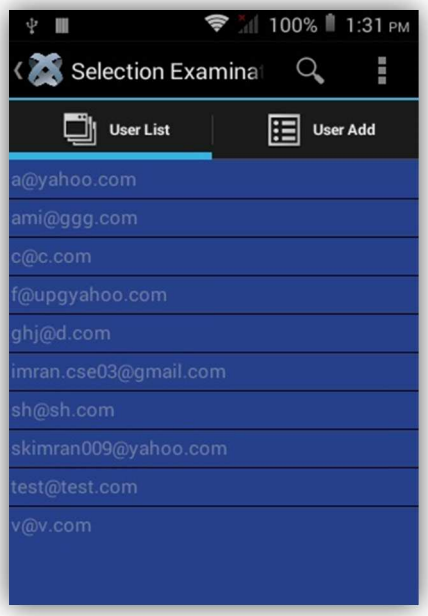

Figure 4.5: User List View

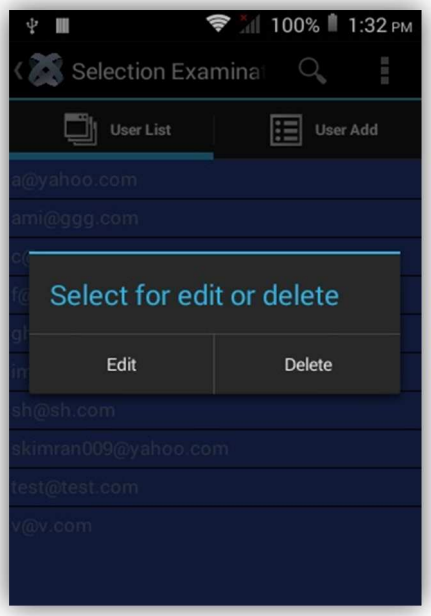

Figure 4.6: Selection for Update

For Edit and Delete

- Tap anyone row
- Confirmation message will show for edit or delete
- Press Edit for modify any row
- Press Delete for delete any Row

For Add User

- Tap ‘User Add’ Tab
- Fill up all field for user information
- Click on ‘Save’ Button

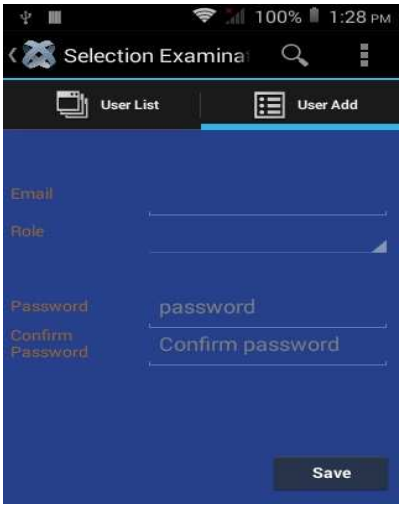

Figure 4.7: User Addition

4.4.3.2 Quiz Management

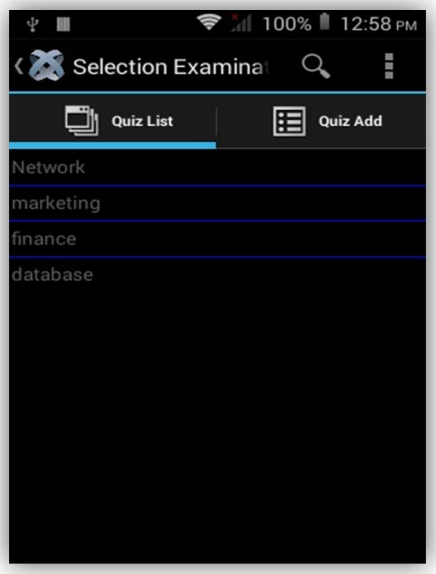

Figure 4.8: Quiz list

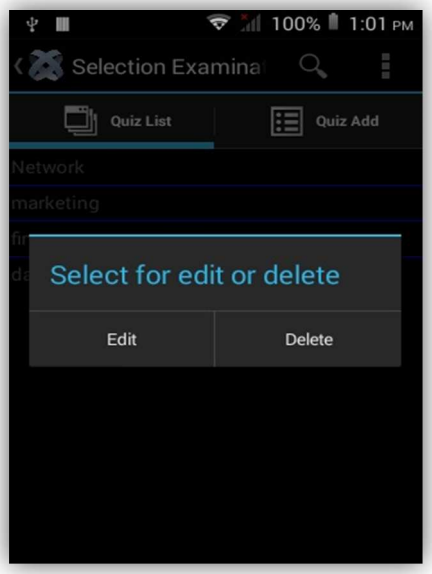

Figure 4.9: Quiz Edit/Delete

For Edit and Delete

- Tap ‘Quiz’ Button [Figure 4.4]
- Tap any row one row
- Confirmation message for edit or delete
- Press Edit for modify any row
- Press Delete for delete any Row
- Press ‘Update’

For Add Quiz

- Tap ‘QUIZ ADD’ Tab
- Fill up all field for user information
- Click on ‘Save’ Button

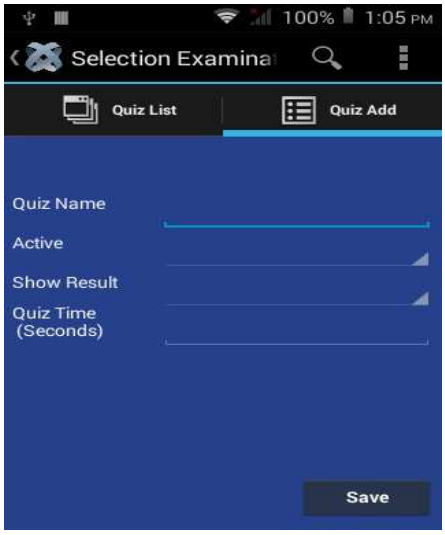

Figure 4.10: Quiz Add

4.4.3.3 Question and Answer Management

For Add Question and Answers

- Fill up the ‘question’ field for any question.
- Fill up ‘Option A’, ‘Option B’, ‘Option C’, ‘Option D’ for input answers respectively.
- Fill up the ‘check box’ which is right side of the answers for correct answer.
- Fill up ‘Quiz ID’ for which the question is belong to.

- Fill the ‘Question Order’ in which order the question will visible to interviewee.
- Tap ‘Save’ Button for store in server.

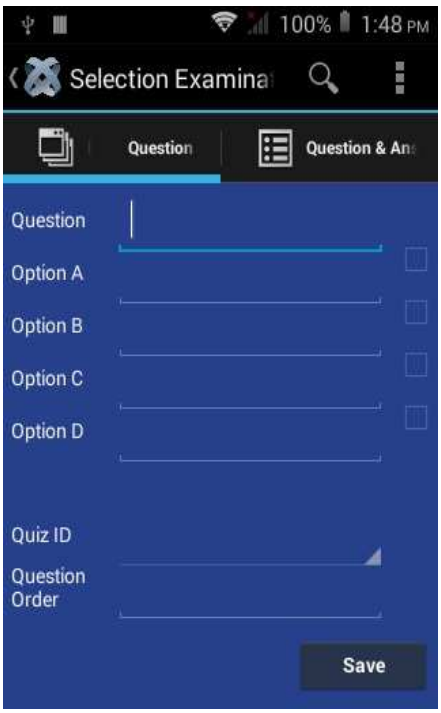

Figure 4.11: Question and Answer Addition

Edit and Delete

- Tap ‘Question and Answer list’ tab.
- Select quiz name from drop down list.
- Tap ‘Show’ button.
- Tap ‘Back to Previous’ Button if need to quiz name selection again.
- Select any question which one need to edit or delete
- Press Edit for modify any row
- Press Delete for delete any Row
- Press ‘Update’

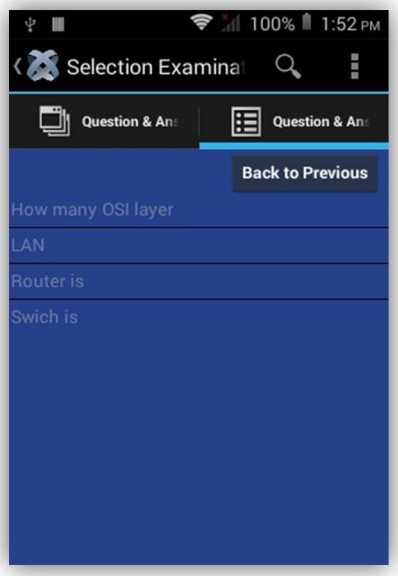

Figure 4.12: Quiz Selection

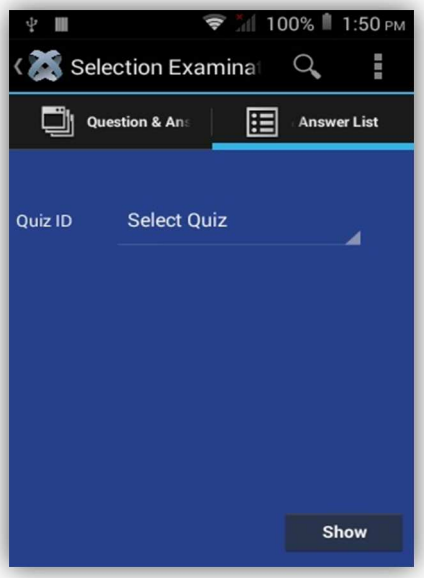

Figure 4.13: View Question

4.4.4 Report

- Tap ‘Report Button’
- Select any quiz name to show report
- Click on ‘Submit’ Button

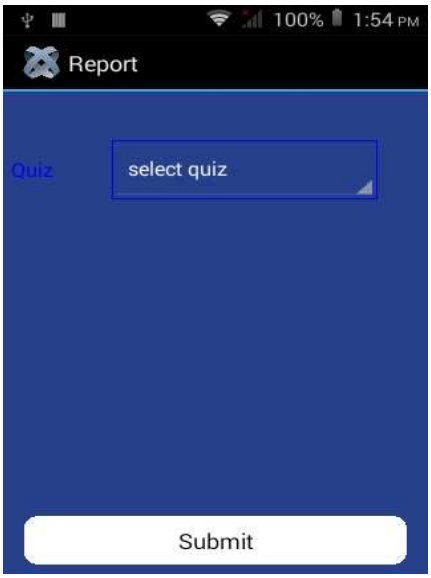

Figure 4.14: Select quiz for Report

4.4.5 Interviewee

- Select Quiz name from drop down list
- Tap on ‘Start’ button.
- Select checkbox for correct answer.
- If answer has given, tap on ‘Next’ button. Please note that if ‘Next’ button has pressed then the question at that time, will consider as answered.
- Tap ‘Skip’ Button if need to answer any question later.
- Time will indicate the remaining time for that particular quiz/test.

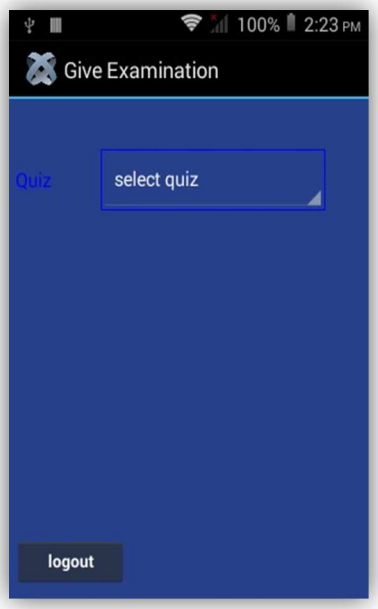

Figure 4.15: Select quiz

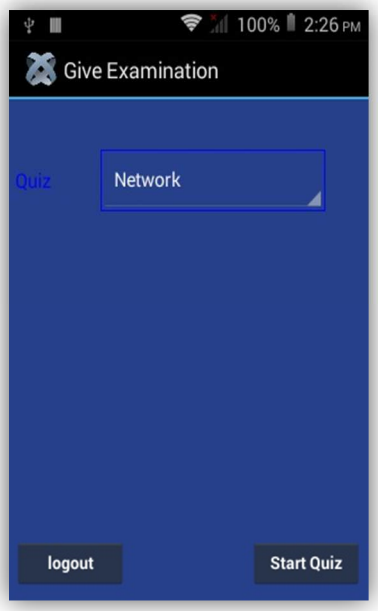

Figure 4.16: Start Quiz

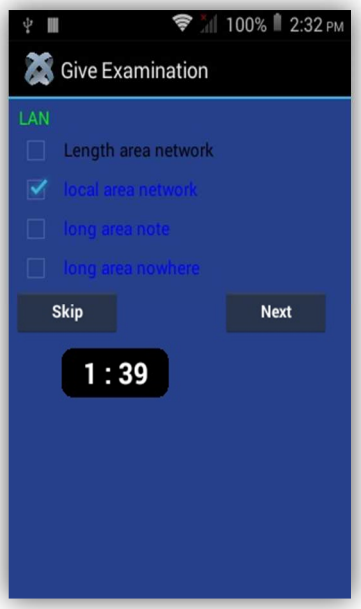

Figure 4.17: Quiz Continuation

4.5 Summary

System will mail User ID and password to concern user email. User can get the privilege according to role (Admin, Interviewee, others). Administrator should follow the steps that describe in Administrative section [4.4.3] for use online quiz application. Interviewee also follow the section Interviewee section [4.4.4]. User should change password after getting the first password using forget/reset password.

Chapter 5

Conclusion and Future Works

5.1 Summary of the literature

General purpose paper based exam system costs time, resources etc. Thus, online exam system gets much more popular. So there is necessary to change system. Online Quiz application will reduce manual intervention and brings flexibility to users. It's very easy to use from anywhere if internet is available and concern user has access to this application. Administrative task can also be done from UI which is very new for any mobile application considering online examination.

5.2 Future Work

Currently, 'Let's quiz' application is developed for Android operating system. As this is the staring of this kind of initiative, Majority for uses, Android is considered as first choice. Other platform like IOS, Windows, and Tizen etc. will consider as future development work. As Lets Quiz is developed with cross platform IDE (Titanium Studio), so few modification will help to build apps that support other operating system. Graphical enhancement will also a part of future planning.

REFERENCES

[1] Zuhayr Reaz. 'The Smartphone Industry in Bangladesh'. Google [online]. Available:http://lightcastlebd.com/blog/2015/01/market-insight-market smartphones-bangladesh

[2] Windows 7. Google [Online]. Available: http://en.wikipedia.org/wiki/Windows_7

[3] Java Development Kit. Wikipedia [Online].

Available: http://en.wikipedia.org/wiki/Java_Development_Kit

[4] Lambert M. Surhone, Mariam T. Tennoe, Susan F. Henssonow. 'Node.js'.

[5] James Steele, and Nelson To. Addison-Wesley, Upper Saddle River, NJ, (2010) 'The Android Developer's Cookbook: Building Applications with the Android SDK'
Available http://www.techopedia.com/definition/4220/android-sdk

[6] 'Developer guide for android'. Available [Online] http://developer.android.com/guide/topics/manifest/uses-sdk-element.html #ApiLevels

[7] Intel Developer Zone. 'Intel® Hardware Accelerated Execution Manager'. Google [Online]. Available: https://software.intel.com/en-us/android/articles/intel-hardware-accelerated-execution-manage

[8] Appecelerator Documentation. Google [Online]. Available on http://docs.appcelerator.com/titanium/3.0/#!/guide/Titanium_Compatibility_Matrixhttps://wiki.appcelerator.org/display/guides2/Titanium+Platform+Overview.

[9] Andrew K-Fox. 'Android Developer Tools'. November 2013. Android Distribution Methods: 'Docand' Reference Series 5/6 Nov 2013 - API19 - Volume 5, 1st edition. Publisher: CreateSpace Independent Publishing Platform

[10] Kevin Whinnery, last modified by Benjamin Hatfield on Apr 03, 2015. ‘Titanium SDK’. Available https://wiki.appcelerator.org/display/guides2/Titanium+SDK

[11] Douglas Crockford. ‘JavaScript: The Good Parts’. O'Reilly Media, Inc. ©2008, ISBN: 0596517742

[12] https://wiki.appcelerator.org/display/guides2/Titanium+Platform+Over view

[13] Stig Saether Bakken, Zeev Suraski, Egon Schmid. ‘PHP Manual’. iUniverse, ISBN: 0595132286

Annexure

Annexure Contain the Sample Code of online Quiz application for this project.

- **Sample Code**

```
/**
   *  This page is responsible for displaying login window
 *    Testing Environment:    CLI version 3.4.1, Titanium SDK version
3.5.0.GA,Targeting Android SDK: 16,Tested Device: Symphony E50,
 *   build: 3.4.1.201410281727 (c) Copyright 2012-2014 by Appcelerator,
Inc.  All rights reserved.
 */
var selectedID=0; // for update user,quiz with capture individual row
 loginid= 0; // for store logind id for store quiz taken user
//var data = [];  //picker array for edit


var winmain = Titanium.UI.createWindow({                    // create
window
    title:'Lets Quiz',
    backgroundColor: '#27408B',
    layout: 'vertical',
    exitOnClose: true
});

var view = Titanium.UI.createView                          //create view
 ({
   borderRadius: 10,
   backgroundColor:'#515151',  //cadet blue
   width:'90%',
   height:'60%',
   top : 10,
   color:"orange",
   layout: 'vertical'
});

winmain.add(view);                                 // add view to main window

var username = Titanium.UI.createTextField       // username textfield
({
    color : '#030303',
    top : 10,
    left : 20,
    right : 20,
    width : Ti.UI.FILL,
    height : 40,
    hintText : 'Enter Username',
    keyboardType : Titanium.UI.KEYBOARD_DEFAULT,
    returnKeyType : Titanium.UI.RETURNKEY_DEFAULT,
```

```
    //clearOnEdit: true,
    borderStyle : Titanium.UI.INPUT_BORDERSTYLE_ROUNDED
});

view.add(username);                              // add username textfield
to view

var password = Titanium.UI.createTextField      // create password
textfield
({
    hintText:'Enter password',
    color: '#030303',
    height:35,
    top:10,
    left:20,
    width : Ti.UI.FILL,
    passwordMask:true,
    //clearOnEdit: true,
    borderStyle:Titanium.UI.INPUT_BORDERSTYLE_ROUNDED,
});
view.add(password);                              // add password text field
to view

var loginBtn = Titanium.UI.createButton        //create login Button
({
   title: 'Login',
   top: 10,
   backgroundColor: '#27408B',
   backgroundSelectedColor: '#7FFFD4',
   width: 100,
   height: 50,
   borderRadius: 10,
   focusable : true,
   //selectedColor : 'green',
   borderColor: '#87CEEB', //skyblue
   font : {fontFamily : 'Garmond',fontWeight : 'bold',fontSize : 18}
});

view.add(loginBtn);                       // add login button to view



//username1="admin";
//password2="1aBcde";

loginBtn.addEventListener('click',password_validate);

function password_validate (){
//create function for password validation
   // var re = (/^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])[0-9a-zA-Z]{6,}$/);
    //var OK = re.exec(password.value);

   if (username.value == '' || password.value == '') {
         alert("username / password required");
   }
    else{

    //server coneection setup
         var ajax = Ti.Network.createHTTPClient();
```

```
        ajax.onerror = function(e){  //3
            alert('Error');
        };
        ajax.onload = function(){  //3
            Titanium.API.info(this.responseText);
            var data = this.responseText;   // receive reply data from php
files after query
            var jdata = JSON.parse(data); //parse and make array


            //-----------alert(loginid);----------//

            if(jdata.success){
                loginid= jdata.data.user_id;   //necessary for interviewee
                //alert(jdata.data.role);   //for check role value
                if (jdata.data.role == 'Admin'){
                         //rows=jdata.data;
                         alert('Welcome Admin');
                         var win2 = Ti.UI.createWindow({
                          backgroundColor:'#27408B',
                          layout: 'vertical',
                          exitOnClose: true,
                          title : 'Admin Panel',
                          url             : 'admin_panel.js'
                        });
                        win2.open();
                }
                 //alert(rows[0].email);
                else if(jdata.data.role == 'Interviewee'){
                        alert('Welcome Interviewee');
                        var Window = require('interviewee');
                        new Window().open({
                                activityEnterAnimation:
Ti.Android.R.anim.slide_in_left,
                                activityExitAnimation:
Ti.Android.R.anim.slide_out_right,
                                //modal:true
                        });
                }
            }
            else{
                alert(jdata.msg);
            }
            //alert(jdata.varname);

        };
        ajax.open('POST', 'http://www.skimranhossain.com/api/check-
user.php');
        ajax.send({
            'email':  username.value,
            'password' : password.value,
        });
        //server connection end

        //alert("Invalid Username/Password ");
       }
}
// Create a Button for forget password .
```

```
   var forget_password_button = Ti.UI.createButton({
       title : 'Forget Password',
       height : '10%',
       width : "90%",
       top: 10,
       bottom : "2%",
       backgroundColor :'#9F79EE',
       focusable : true,
       color:'black',
       borderRadius: 10

   });

   // Listen for click events.
   forget_password_button.addEventListener('click', function() {
           var Window = require('forget_password');
           new Window().open({
                                activityEnterAnimation:
   Ti.Android.R.anim.slide_in_left,
                                activityExitAnimation:
   Ti.Android.R.anim.slide_out_right,
                                    //modal:true
                            });
   });

   dialouge_about = Ti.UI.createAlertDialog({                        // For
   "About" menu for all pages except admin panel.js
          message : 'This is an online Examination Apps',
       });
```

```
/**
 * Admin Panel
 */
var myWin = Ti.UI.currentWindow;
/*
 * -------------------------Menu bar start---------------------------------
-
 */
myWin.activity.onCreateOptionsMenu = function(e) {
                        var menu = e.menu;
                        var logout = menu.add({title: 'logout',enabled: true,
itemId: '1',visible:true});
                        logout.addEventListener('click',function(){
                            // do the event handling here
                            //myWin.close();
                            dialouge.show();
                            //location.reload();
                        });
                        var about = menu.add({title: 'About',enabled: true,
itemId: '2',visible:true});
                        about.addEventListener('click',function(e){
                            // do the event handling here
                            dialouge_about.show();
                        });
```

```
};

var dialouge= Ti.UI.createAlertDialog({                        // For logout
        title: 'Sure to logout !',
        buttonNames: ['No','Yes'],
    });
dialouge.addEventListener('click',function(e){
            if(e.index == 0)
                  //no
                  dialouge.hide();
            else if (e.index == 1)
                  //yes
                  myWin.close();

});

var dialouge_about = Ti.UI.createAlertDialog({  // For About menu for all
pages except admin panel.js
       message : 'This is an online Examination Apps',
    });
/*
 * ------------------------Menu Bar End-------------------------------------
------------
 */
myWin.addEventListener('android:back',function(e) {
   dialouge.show();
});

//CREATE VIEW
var view = Titanium.UI.createView({
   borderRadius: 1,
   //backgroundColor:'0x0106001b',
   width:'auto',
   height:'auto',
   top : 10,
   //color:'black',
   backgroundColor:'#27408B',
   layout: 'vertical'
});

myWin.add(view);  //add view to current window

/*-----------------------------------------------Button creation for admin
panel start-------------------------------*/
var quiz_button = Titanium.UI.createButton({
    title : 'Quiz',
    width : Ti.UI.FILL,
    height : Ti.UI.SIZE,
    //backgroundcolor: 'AliceBlue',
    backgroundColor: '#B0C4DE',  //lavender
    color:'black',
    borderRadius : 1,
    font : {fontFamily : 'Garmond',fontWeight : 'bold',fontSize : 14},
    backgroundSelectedColor: '#7FFFD4',
    focusable: true,
    borderColor: '#87CEEB',  //skyblue
    borderRadius: 10,
    top : 10
    });
```

```
quiz_button.addEventListener('click',function(){
// redirect to quiz pasge
    //alert('quiz page entry');
    var Window = require('quiz');
    new Window().open({
                    activityEnterAnimation: Ti.Android.R.anim.slide_in_left,
                    activityExitAnimation: Ti.Android.R.anim.slide_out_right,
                    //modal:true
                    });
});

var question = Titanium.UI.createButton({
    title : 'Questions & Answer',
    width : Ti.UI.FILL,
    height : Ti.UI.SIZE,
    //backgroundcolor: 'AliceBlue',
    backgroundColor: '#B0C4DE',
    backgroundSelectedColor: '#7FFFD4',
    color:'black',
   // borderRadius : 1,
    font : {fontFamily : 'Garmond',fontWeight : 'bold',fontSize : 14},
    //left: 1,
    //right: 10,
    top : 10,
    borderRadius: 10
    //verticalAlign:'center',
    });

question.addEventListener('click',function(){  // redirect to question/answer
page
    var Window = require('question_answer');
    new Window().open({
                    activityEnterAnimation: Ti.Android.R.anim.slide_in_left,
                    activityExitAnimation: Ti.Android.R.anim.slide_out_right
                    });
});

var report = Titanium.UI.createButton({
    title : 'Reports',
    top : 10,
    width : Ti.UI.FILL,
    height : Ti.UI.SIZE,
    color:'black',
    //backgroundcolor: 'AliceBlue',
    backgroundColor: '#B0C4DE',
    backgroundSelectedColor: '#7FFFD4',
    borderRadius : 10,
    font : {fontFamily : 'Garmond',fontWeight : 'bold',fontSize : 14}
});

report.addEventListener('click',function(){                                //
redirect to
    //alert('do something');
    /*var Window = require('interviewee');
    new Window().open();*/
   var Window = require('report');
    new Window().open({
                    activityEnterAnimation: Ti.Android.R.anim.slide_in_left,
                    activityExitAnimation: Ti.Android.R.anim.slide_out_right
```

```
                    });
 });

var user = Titanium.UI.createButton({
    title : 'Users',
    top : 10,
    width : Ti.UI.FILL,
    height : Ti.UI.SIZE,
    color:'black',
    backgroundSelectedColor: '#7FFFD4',
    backgroundColor: '#B0C4DE',
    borderRadius : 10,
    font : {fontFamily : 'Garmond',fontWeight : 'bold',fontSize : 14}
    });
/*-----------------------------------------------Button creation finish for
Admin Panel-----------------------------------------*/

/*
 * user list view for administrator for "users" button
 */
function user_list(){
        var Window = require('users');
        new Window().open({
                    activityEnterAnimation: Ti.Android.R.anim.slide_in_left,
                    activityExitAnimation: Ti.Android.R.anim.slide_out_right
                        });
};

user.addEventListener('click',user_list); //user list view end for
administrator. Redirect to page user_list.js

// Adding buttons in view
view.add(quiz_button);
view.add(question);
view.add(report);
view.add(user);


/**
 * @Add Form
 */
function myWin () {

            var tabGroup = Titanium.UI.createTabGroup();   // create tab
group

            /*----------------------Action Bar back button----------------
---*/
            tabGroup.addEventListener('open', function(){
               tabGroup.activity.actionBar.onHomeIconItemSelected =
function()
                         { tabGroup.close();
                         };

               tabGroup.activity.actionBar.displayHomeAsUp = true;
            });

            /* -----------------Action Bar Back button End-----------*/
```

```
            var search = Ti.UI.Android.createSearchView({
//create searchview;
                    hintText: "User Search"
                });
            var dialouge= Ti.UI.createAlertDialog({          // For logout
                    title: 'Sure to logout !',
                    buttonNames: ['No','Yes'],
                });

            dialouge.addEventListener('click',function(e){
                if(e.index == 0)
                    //no
                    dialouge.hide();
                else if (e.index == 1){
                    //Ti.App.fireEvent('Selection Examination');
                    tabGroup.close();
                    var activity = Titanium.Android.currentActivity;
                   activity.finish();
                        };
                    //yes
                    //tabGroup.blur();

            });
/*
 * ----------------------Action Menu Bar start------------------------------
 */
            tabGroup.activity.onCreateOptionsMenu = function(e) {
                var menu = e.menu;

                var gotomainpage = menu.add({title: 'Main Menu',enabled:
true, itemId: '1',visible:true});
                gotomainpage.addEventListener('click',function(){
                    // do the event handling here
                    tabGroup.close();
                });

                var logout = menu.add({title: 'logout',enabled: true,
itemId: '1',visible:true});
                logout.addEventListener('click',function(){
                    // do the event handling here
                    dialouge.show();
                });

                var about = menu.add({title: 'About',enabled: true, itemId:
'2',visible:true});
                about.addEventListener('click',function(e){
                    // do the event handling here
                    dialouge_about.show();
                });


                var search_menu = menu.add({title: 'search',enabled:
true,actionView : search,
                icon: (Ti.Android.R.drawable.ic_menu_search ?
Ti.Android.R.drawable.ic_menu_search : "my_search.png"),
                showAsAction: Ti.Android.SHOW_AS_ACTION_IF_ROOM |
Ti.Android.SHOW_AS_ACTION_COLLAPSE_ACTION_VIEW,
                itemId: '1',visible:true});
```

```
                search.addEventListener('click',function(){
                    // do the event handling here
                    search.addEventListener('cancel', function(){
                                search.blur();
                    });

                });

            };

/*
 * ------------------------------Action Menu Bar End-------------------------
-
 */
// create base UI tab and root window
//
            var win1 = Titanium.UI.createWindow({
                title:'Tab 1',
                backgroundColor:'white'
            });

            var tab1 = Titanium.UI.createTab({
                icon:'KS_nav_views.png',
                title:'User List',
                window:win1
            });

            var scrollView = Ti.UI.createScrollView({
              contentWidth: 'auto',
              contentHeight: 'auto',
              //showVerticalScrollIndicator: true,
              //showHorizontalScrollIndicator: true,
              scrollType: 'horizontal',
              height: 'auto',
              width: 'auto'
            });

            var section = Ti.UI.createListSection();
            var listView = Ti.UI.createListView
            ({
              sections: [ section ],
              searchView: search,
              editing: true,
              caseInsensitiveSearch: true,
              pruneSectionsOnEdit : true,
              separatorColor: 'blue',
              width:'100%'

            });
                  //server coneection setup
         var ajax = Ti.Network.createHTTPClient();
        ajax.onerror = function(e){
            alert('Error');
        };
        ajax.onload = function(){
            Titanium.API.info(this.responseText);
            var data = this.responseText;
            var jdata = JSON.parse(data);
```

```
            if(jdata.success){
                //alert('success');   // alert if connection is success
                 rows=jdata.data;
               var dataArr = [];
                 for(i=0; i< rows.length; i++){
                     dataArr[i]={ properties: { title: rows[i].email,
canEdit: true, canMove: true, id: rows[i].user_id,color:'black',
                                searchableText :rows[i]}};
                     //dataArr.push({properties: {searchableText:
rows[i].email}});  //search text
                 }

                 console.log(dataArr);
                 section.setItems(dataArr);
            }
            else{
                alert(jdata.msg);
            }
            //alert(jdata.varname);

        };
        ajax.open('POST', 'http://www.skimranhossain.com/api/get-users.php');
        ajax.send();
        //server connection end
/*
 * For edit any user row start
 */


             listView.addEventListener('itemclick', function(e){    //for
update or edit user
                 var item = e.section.getItemAt(e.itemIndex);   //catch the
index number for user
                 var item_index_number = e.itemIndex; // capture item index
number for delete purpose
                 //alert(e.itemIndex);
                 //alert(e.items[e.itemIndex].properties.id);
                 //alert(item.properties.id);
                 //console.log(e);
                 selectedID=item.properties.id;   // catch the user ID
                   //e.section.updateItemAt(e.itemIndex,item);
                 var dialouge= Ti.UI.createAlertDialog({
                     title: 'Select for edit or delete',
                     buttonNames: ['Delete','Edit']
                 });

       dialouge.addEventListener('click',function(e){
                         if(e.index==0){
//Delete any user
                    var ajax = Ti.Network.createHTTPClient();
                    ajax.onerror = function(e){
                        alert('Error');
                    };
                                            // delete users
                     ajax.onload = function(){
                        Titanium.API.info(this.responseText);
                        var data = this.responseText;
                        var jdata = JSON.parse(data);
```

```
                        if(jdata.success){
                          section.deleteItemsAt(item_index_number,1);
                            alert('deleted');
                            //section.updateItemAt(a,item);
                         }
                        else
                            alert(jdata.msg);
                    };

                ajax.open('POST', 'http://www.skimranhossain.com/api/add-
user.php');
                ajax.send({
                    'delete_id': selectedID
                });

       }

       else if(e.index==1){
       //EDIT users coding start

           //------alert(selectedID);  // alert for check if the correct user
ID is showing or not
           var Window = require('users_edit');
           new Window().open({
                    activityEnterAnimation: Ti.Android.R.anim.slide_in_left,
                    activityExitAnimation: Ti.Android.R.anim.slide_out_right,
                    selectedID: selectedID
                 });
           //section.updateItemAt(item_index_number,item);
           /*
             *server call again for update user id at front interface
             */
            tab1.addEventListener('focus', function(){           //recal
server for refresh updated row
                        //server coneection setup
                                 var ajax = Ti.Network.createHTTPClient();
                                ajax.onerror = function(e){
                                    alert('Error');
                                };
                                ajax.onload = function(){
                                    Titanium.API.info(this.responseText);
                                    var data = this.responseText;
                                    var jdata = JSON.parse(data);

                                    if(jdata.success){
                                        //alert('success');   // alert if
connection is success
                                         rows=jdata.data;
                                       var dataArr = [];
                                         for(i=0; i< rows.length; i++){
                                             dataArr[i]={ properties: {
title: rows[i].email, canEdit: true, canMove: true, id: rows[i].user_id,

color:'black',searchableText :rows[i]}};
                                             //dataArr.push({properties:
{searchableText: rows[i].email}});  //search text
                                         }

                                         console.log(dataArr);
```

```
                                        section.setItems(dataArr);
                                    }
                                    else{
                                        alert(jdata.msg);
                                    }
                                    //alert(jdata.varname);

                                };
                                ajax.open('POST',
'http://www.skimranhossain.com/api/get-users.php');
                                ajax.send();
                                //server connection end
            });

            /*
             * server call again for update user id at front interface end
             */

                  }

      });

      dialouge.show();
      });

scrollView.add(listView);
win1.add(scrollView);

/*
 * ------------------User Add form start------------------
 */
// create controls tab and root window

            var win2 = Titanium.UI.createWindow({
                title:'Tab 2',
                backgroundColor:'#27408B'
            });
            var tab2 = Titanium.UI.createTab({
                icon:'KS_nav_ui.png',
                title:'User Add',
                window:win2
            });

            var emaillabel = Titanium.UI.createLabel({
                text : 'Email',
                height : 50,
                color : 'black',
                font : {
                    fontSize : 14
                },
                top : "10%", // vertical postion of label on the screen
w.r.t screen top
                left : "2%", // horizontal postion of label on the screen
w.r.t screen left
                width : 200 // width of the label
            });

            win2.add(emaillabel);
```

```
            var emailtext_field = Titanium.UI.createTextField({
                color : 'white',
                height : 50,
                top : "10%", // vertical postion of label on the screen
w.r.t screen top
                right : "2%", // horizontal postion of label on the screen
w.r.t screen left
                width : 200, // width of the field
                borderStyle : Titanium.UI.INPUT_BORDERSTYLE_ROUNDED
            });
            win2.add(emailtext_field);
            // add field to the current window
            var role_label = Titanium.UI.createLabel({
                text : 'Role',
                height : 50,
                color : 'black',
                font : {
                    fontSize : 14
                },
                top : "20%", // vertical postion of label on the screen
w.r.t screen top
                left : "2%", // horizontal postion of label on the screen
w.r.t screen left
                width : 200 // width of the label
            });

            win2.add(role_label);
                    /*-------------------------------------picker create for
ROLE-----------------*/
            var picker = Ti.UI.createPicker({
                height : 50,
                top : "20%", // vertical postion of label on the screen
w.r.t screen top
                right : "2%", // horizontal postion of label on the screen
w.r.t screen left
                width : 200, // width of the field
              //selectionIndicator: true,
            });
            var data = [];
            data[0]= Ti.UI.createPickerRow({title:''});
            data[1]= Ti.UI.createPickerRow({title:'Admin'});
            data[2]= Ti.UI.createPickerRow({title:'Interviewee'});
            data[3]= Ti.UI.createPickerRow({title:'Others'});
            picker.add(data);
            win2.add(picker);

            picker.addEventListener('change', function(e){
                picker.value= e.row.title;
            });

      /*----------------picker for ROLE END------------------*/

            var password_label = Titanium.UI.createLabel({
                text : 'Password',
                height : 50,
                color : 'black',
                font : {
                    fontSize : 14
                },
```

```
                top : "40%", // vertical postion of label on the screen
w.r.t screen top
                left : "2%", // horizontal postion of label on the screen
w.r.t screen left
                width : 200 // width of the label
            });

            win2.add(password_label);

            var password_text_field = Titanium.UI.createTextField({
                color : 'white',
               height : 50,
                top : "40%", // vertical postion of label on the screen
w.r.t screen top
                right : "2%", // horizontal postion of label on the screen
w.r.t screen left
                width : 200, // width of the field
                passwordMask:true,
                hintText:"password",
                borderStyle : Titanium.UI.INPUT_BORDERSTYLE_ROUNDED
            });
            win2.add(password_text_field);

            ////////////////////option 4//////////////////////////////

            var cpassword_label = Titanium.UI.createLabel({
                text : 'Confirm\nPassword',
                height : 50,
                color : 'black',
                font : {
                    fontSize : 14
                },
                top : "50%", // vertical postion of label on the screen
w.r.t screen top
                left : "2%", // horizontal postion of label on the screen
w.r.t screen left
                width : 200 // width of the label
            });

            win2.add(cpassword_label);

            var cpassword_text_field = Titanium.UI.createTextField({
                color : 'white',
                height : 50,
                top : "50%", // vertical postion of label on the screen
w.r.t screen top
                right : "2%", // horizontal postion of label on the screen
w.r.t screen left
                width : 200, // width of the field
                passwordMask:true,
                hintText:"Confirm password",
                borderStyle : Titanium.UI.INPUT_BORDERSTYLE_ROUNDED
            });
            win2.add(cpassword_text_field);

                                                    // Save button for all
entry ////
```

```
            var savebutton = Titanium.UI.createButton({
//create save button
                title : 'Save',
                color: 'white',
                right:'5dp',
                bottom:'10dp',
                height:'40dp',
                width:'100dp',
                font : {fontFamily : 'Arial',fontWeight : 'bold',fontSize :
14},
                backgroundcolor: 'black',
                //backgroundSelectedColor: '#7FFFD4'

            });
//save button create End


            savebutton.addEventListener('click',function(){
                var re = (/^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])[0-9a-zA-
Z]{6,}$/);   //PASSWORD VALIDATION
                var emailvalidation =
(/^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)*)|(\".+\"))@((\[[0-
9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-
Z]{2,}))$/);
                var OK = re.exec(password_text_field.value);
                var cOK= re.exec(cpassword_text_field.value);
                var emailfieldvalidation =
emailvalidation.exec(emailtext_field.value);
                if(password_text_field.value==''||
cpassword_text_field.value=='')
                    alert('Password blank');
                else if(!emailfieldvalidation)
                        alert('wrong email address');
                else if(!OK ||!cOK)
                   alert("invalid password format");
                else if (password_text_field.value !=
cpassword_text_field.value)
                         alert('Please type same password to both field');
                    // alert(picker.value);
                else{                                       //server coneection
setup
                    var ajax = Ti.Network.createHTTPClient();
                    ajax.onerror = function(e){
                        alert('Error');
                    };
                    ajax.onload = function(){
                        Titanium.API.info(this.responseText);
                        var data = this.responseText;
                        var jdata = JSON.parse(data);

                        if(jdata.success){
                             //rows=jdata.data;
                             var Cloud = require('ti.cloud');
                              Cloud.Emails.send({
                                       template : 'default',
                                       recipients : emailtext_field.value,
                                       usrname : emailtext_field.value,
                                       pword : password_text_field.value
                               }, function(e) {
```

```
                                        if (e.success) {
                                            alert('Mail Sent Successfully');
                                            // reset to original state
                                            emailtext_field.value = '';
                                            picker.value= '';
                                            password_text_field.value = '';
                                            cpassword_text_field.value = '';
                                        } else {
                                            alert('Error:\n' + ((e.error &&
e.message) || JSON.stringify(e)));
                                        }
                                    });
                            }
                            else{
                                alert(jdata.msg);
                            }
                            //alert(jdata.varname);

                        };
                        ajax.open('POST',
'http://www.skimranhossain.com/api/add-user.php');    //First open php file
                        ajax.send({
                            'email':  emailtext_field.value,
                            'password' : password_text_field.value,
                            'role': picker.value
                        });
                }
                });

                win2.add(savebutton);
                //  add tabs
                tabGroup.addTab(tab1);
                tabGroup.addTab(tab2);
                /*------------Event Listener to reset text field start-------*/
                tabGroup.addEventListener('focus', function(e)
                {
                    if (e.index == 0 && e.previousIndex != 0) {
                        //alert('refresh');
                        // reset to original state
                        emailtext_field.value = '';
                        password_text_field.value = '';
                        cpassword_text_field.value = '';
                    }
                });
                    /*-----------------Event Listener to reset text field End---
------------------*/

return tabGroup;
}
module.exports=myWin;



/**
 * @Edit
 */
function myWin (e) {
    var quiz_edit = Ti.UI.createWindow();
//console.log(e);
```

```
//var curtWind = Ti.UI.currentWindow;

       /*------------picker create for Quiz Active Query----------------*/
var picker = Ti.UI.createPicker({
    height : 50,
    top : "20%", // vertical postion of label on the screen w.r.t screen top
    right : "2%", // horizontal postion of label on the screen w.r.t screen
left
    width : 200, // width of the field
  //bottom:60,
  //width:150,
  //useSpinner : true
  borderStyle : Titanium.UI.INPUT_BORDERSTYLE_ROUNDED,
  selectionIndicator: true
});

/*
 * ------------------------show result picker------------------------------
 */
var picker_result = Ti.UI.createPicker({
    height : 50,
    top : "30%", // vertical postion of label on the screen w.r.t screen top
    right : "2%", // horizontal postion of label on the screen w.r.t screen
left
    width : 200, // width of the field
  //bottom:60,
  //width:150,
  //useSpinner : true
  borderStyle : Titanium.UI.INPUT_BORDERSTYLE_ROUNDED,
  selectionIndicator: true
});

//server coneection setup
         var ajax = Ti.Network.createHTTPClient();
        ajax.onerror = function(e){
            alert('Error');
        };
        ajax.onload = function(){
            Titanium.API.info(this.responseText);
            var data = this.responseText;
            var jdata = JSON.parse(data);
            console.log(jdata);
            if(jdata.success){
                 rows=jdata.data;
               var dataArr = [];
               var quiz_name=rows[0].quiz_name;  // email field data
               var quiz_active= rows[0].active;   //active quiz data
               var quiz_result=rows[0].show_result;  //show result or not
field data
               var quiz_time= rows[0].quiz_time;
               //alert(quiz_time);
               quiztext_field.value=quiz_name;
               quiz_time_text_field.value = quiz_time;
                       /*-------------------------------------pickerROW
create for active quiz start-----------------*/

                var data = [];
                data[0] = Ti.UI.createPickerRow({id:'1',title:'Yes'});
                data[1] = Ti.UI.createPickerRow({id:'0',title:'No'});
```

```
                picker.add(data);
                quiz_edit.add(picker); //add quiz active picker to window
                var selIndex=0;
                if(quiz_active== 1)
                    selIndex=0 ;
                else
                    selIndex=1;
                picker.setSelectedRow(0,selIndex);   //Selects a column's row
                picker.value = quiz_active;
          /*
           * -------------------Picker ROW create for Show result start------
------------------
           */
                var result = [];
                result[0] = Ti.UI.createPickerRow({id:'1',title:'Yes'});
                result[1] = Ti.UI.createPickerRow({id:'0',title:'No'});

                picker_result.add(result);
                quiz_edit.add(picker_result);  //add quiz show result picker
to window
                //picker_result.value(quiz_result);

                if(quiz_result== 1)
                    selIndex=0 ;
                else
                    selIndex=1;
                picker_result.setSelectedRow(0,selIndex);
                picker_result.value = quiz_result;

            }
            else{
                alert(jdata.msg);
            }
            //alert(jdata.varname);

        };
        ajax.open('POST', 'http://www.skimranhossain.com/api/get-
quizes.php');
        ajax.send({
            'quiz_id' : selectedID
        });
        //server connection end

picker.addEventListener('change', function(e){
                    picker.value= e.row.id;
                    //alert(picker.value);
                });

picker_result.addEventListener('change', function(e){
    picker_result.value= e.row.id;
});
//alert(picker.value);
//alert(selectedID);

/*
 * ------UPDATE QUIZ------------------
 */
```

```
var quizlabel = Titanium.UI.createLabel({
//Quiz name lable start
    text : 'Quiz Name',
    height : 50,
    color : 'white',
    font : {
        fontSize : 14
    },
    top : "10%", // vertical postion of label on the screen w.r.t screen top
    left : "2%", // horizontal postion of label on the screen w.r.t screen
left
    width : Ti.UI.SIZE // width of the label
});
//Quiz name lable End

quiz_edit.add(quizlabel);
// add label to the current window

var quiztext_field = Titanium.UI.createTextField({
    color : 'white',
    height : 50,
    top : "10%", // vertical postion of label on the screen w.r.t screen top
    right : "2%", // horizontal postion of label on the screen w.r.t screen
left
    width : 200, // width of the field
    borderStyle : Titanium.UI.INPUT_BORDERSTYLE_ROUNDED
});
quiz_edit.add(quiztext_field);
// add field to the current window

////////////// options A ////////
var quiz_activelabel = Titanium.UI.createLabel({
    text : 'Active',
    height : '50',
    color : 'white',
    font : {
        fontSize : 14
    },
    top : "20%", // vertical postion of label on the screen w.r.t screen top
    left : "2%", // horizontal postion of label on the screen w.r.t screen
left
    width : Ti.UI.SIZE // width of the label
});

quiz_edit.add(quiz_activelabel);

var quiz_result_label = Titanium.UI.createLabel({
    text : 'Show Result',
    height : '50',
    color : 'white',
    font : {
        fontSize : 14
    },
    top : "30%", // vertical postion of label on the screen w.r.t screen top
    left : "2%", // horizontal postion of label on the screen w.r.t screen
left
    width : Ti.UI.SIZE // width of the label
});
quiz_edit.add(quiz_result_label);
```

```
var quiz_time = Titanium.UI.createLabel({
    text : 'Quiz Time \n (Seconds)',
    height : '50',
    color : 'white',
    font : {
        fontSize : 14
    },
    top : "40%", // vertical postion of label on the screen w.r.t screen top
    left : "2%", // horizontal postion of label on the screen w.r.t screen
left
    width : Ti.UI.SIZE // width of the label
});
quiz_edit.add(quiz_time);
var quiz_time_text_field = Titanium.UI.createTextField({
    color : 'white',
    height : 50,
    top : "40%", // vertical postion of label on the screen w.r.t screen top
    right : "2%", // horizontal postion of label on the screen w.r.t screen
left
    width : 200, // width of the field
    borderStyle : Titanium.UI.INPUT_BORDERSTYLE_ROUNDED
});
quiz_edit.add(quiz_time_text_field);

var savebutton = Titanium.UI.createButton({
//create save button
    title : 'Update',
    color: 'white',
    right:'5dp',
    bottom:'10dp',
    height:'40dp',
    width:'100dp',
    //borderRadius : 10,
    font : {fontFamily : 'Arial',fontWeight : 'bold',fontSize : 14},
    backgroundcolor: 'black',
    backgroundSelectedColor: '#7FFFD4'
});
quiz_edit.add(savebutton);
savebutton.addEventListener('click',function(){
                                                                    //server coneection
setup
         var ajax = Ti.Network.createHTTPClient();
        ajax.onerror = function(e){
            alert('Error');
        };
        ajax.onload = function(){
            Titanium.API.info(this.responseText);
            var data = this.responseText;
            var jdata = JSON.parse(data);

            if(jdata.success){
                 //rows=jdata.data;
                 //quizwin.close();
                 //alert(rows[0].email);
                 //alert('success');
                 var dialouge_success= Ti.UI.createAlertDialog({
                     title: 'Successfully Updated',
                     buttonNames: ['OK'],
```

```
                    //buttonNames: ['Edit', 'Delete']
                });
                dialouge_success.show();
                dialouge_success.addEventListener('click',function(e){
                        if(e.index == 0)
                                quiz_edit.close();
                });
            }
            else{
                alert(jdata.msg);
            }
            //alert(jdata.varname);

        };
        ajax.open('POST', 'http://www.skimranhossain.com/api/add-quiz.php');
//need to change for quiz table
        ajax.send({
            'quiz_name':  quiztext_field.value,
            'active' : picker.value,
            'show_result' : picker_result.value,
            'quiz_time':quiz_time_text_field.value,
            'id': selectedID
        });
      //alert(picker.value);
//server connection end
});


    return quiz_edit;
}
module.exports=myWin;

/*
 * -------Form for Interviewee Exam start--Show result result---Exam
DashBoard-------applicable for interviewee----
 */
function myWin ()
         {
            var window_exam = Ti.UI.createWindow({
                 title:'Give Examination',
                 backgroundColor:'#27408B',  //aliceblue
            });
            var testLabels; //for 1st question label
            var answerLabels = []; //for answer label
            Switch = []; //for answer checkbox input from user
            questions = []; //global
            skippedQuestions = []; //global
            current_question = 0;
            var user_quiz;
            var quiz_id;
            var time;
            /*
            * --------------------clock start----------------------
            */
            var countDown = function(m, s, fn_tick, fn_end) {
                                return {
                                        total_sec : m * 60 + s,
                                        timer : this.timer,
                                        set : function(m, s) {
```

```
                                                this.total_sec =
parseInt(m) * 60 + parseInt(s);
                                                this.time = {
                                                      m : m,
                                                      s : s
                                                };
                                                return this;
                                          },
                                          start : function() {
                                                var self = this;
                                                this.timer =
setInterval(function() {
                                                if (self.total_sec) {
                                                      self.total_sec--;
                                                      self.time = {
                                                            m :
parseInt(self.total_sec / 60),
                                                            s :
(self.total_sec % 60)
                                                      };
                                                      fn_tick();
                                                }
                                                else {
                                                      self.stop();
                                                      fn_end();
                                                  }
                                                }, 1000);
                                                return this;
                                          },
                                          stop : function() {
                                                clearInterval(this.timer);
                                                this.time = {
                                                      m : 0,
                                                      s : 0
                                                };
                                                this.total_sec = 0;
                                                return this;
                                          }
                                    };
                              };

                  /*
                   * ------------------------Clock End----------------
                   */

            /*
             * Quiz label and picker creation for select quiz ID start
             */
            var quiz_idlabel = Titanium.UI.createLabel({
                text : 'Quiz',
                height : 50,
                color : 'black',
                font : {
                    fontSize : 16
                },
                top : "10%", // vertical postion of label on the screen w.r.t
screen top
                left : "2%", // horizontal postion of label on the screen
w.r.t screen left
```

```
            width : 200 // width of the label
        });
        window_exam.add(quiz_idlabel);


            //window_exam.add(startbutton);

        var picker = Ti.UI.createPicker({
                                    height : 50,
                                    top : "10%", // vertical postion of
label on the screen w.r.t screen top
                                    right : "12%", // horizontal postion
of label on the screen w.r.t screen left
                                    width : 200, // width of the field
                                    borderColor : 'blue',
                                    borderStyle :
Titanium.UI.INPUT_BORDERSTYLE_ROUNDED,
                                    selectionIndicator: true
                            });


        var ajax = Ti.Network.createHTTPClient();  //server connection
for getting quiz ID
        ajax.onerror = function(e){
              alert('Error');
          };

        ajax.onload = function(){
             Titanium.API.info(this.responseText);
             var data = this.responseText;
             var jdata = JSON.parse(data);
             if(jdata.success){
                                        var row_data = [];
                                        var rows=jdata.data;
                                        for(i=0;i < rows.length; i++){
                                              row_data[0]=
Ti.UI.createPickerRow({id:0,title:'select quiz'});
                                              row_data[i+1]=
Ti.UI.createPickerRow({id:rows[i].quiz_id, title:rows[i].quiz_name,});
                                          }
                                         picker.add(row_data);
                                         window_exam.add(picker);

picker.addEventListener('change', function(e){
                                              picker.value= e.row.id;

                                                    quiz_id =
picker.value;

   window_exam.add(startbutton

   //startbutton.visible = true;


   startbutton.show();

                                         });
                                    }
                else{
```

```
                        alert(jdata.msg);
                      }

            };
                  //alert(loginid);
               ajax.open('POST', 'http://www.skimranhossain.com/api/get-
quizes.php');
               ajax.send({
                        'user_id':loginid
                  });

                    var startbutton = Titanium.UI.createButton({
                                                //create exam start button

title : 'Start Quiz',

color: 'white',

right:'5dp',

bottom:'10dp',

height:'40dp',

width:'100dp',

//visible: false,

font : {fontFamily : 'Arial',fontWeight : 'bold',fontSize : 14},

// backgroundSelectedColor: '#7FFFD4',

backgroundcolor: 'black'

});


                    startbutton.addEventListener('click',function(e){
                              dialouge.show();
            });

            window_exam.addEventListener('focus', function(){
               logout = Titanium.UI.createButton({
                  //create exam logout button

title : 'logout',

color: 'white',

left:'5dp',

bottom:'10dp',

height:'40dp',

width:'100dp',

//visible: false,
```

```
font : {fontFamily : 'Arial',fontWeight : 'bold',fontSize : 14},

//backgroundSelectedColor: '#7FFFD4',

backgroundcolor: 'black'

});
              window_exam.add(logout);
               logout.addEventListener('click',function(){
                                 //alert('logout');
                                 window_exam.close();
                              //var activity = Titanium.Android.currentActivity;
                             //activity.finish();
                 });
                 });
            /*----------picker for quiz id get from server END----------*/

            var scrollView = Ti.UI.createScrollView({
             //All labels view
                                                  contentWidth: 'auto',
                                                  contentHeight: 'auto',

//showVerticalScrollIndicator: true,

//showHorizontalScrollIndicator: true,
                                                  //scrollType: 'horizontal',
                                                  layout : 'vertical',
                                                  scrollType : 'vertical',
                                                  height: 'auto',
                                                  width: 'auto',
                                                  visible: true,
                                                  //views: view,
                                                  //showPagingControl : true
                                        });
                     window_exam.add(scrollView);


              testLabels = Titanium.UI.createLabel({          // Question
label
                                                               //borderRadius:
35,

//backgroundcolor: 'blue',
                                                               left: 10,
                                                              // borderColor:
'white',
                                                               text:'',
                                                               color: 'green',

textAlign:'center',
                                                              // left : '10dp',
                                                               width:Ti.UI.SIZE,
                                                               height:
Ti.UI.SIZE,
                                                               top: 5,
                                                               //left:'2%',
                                                               touchEnabled:
true
```

```
                                    });
                scrollView.add(testLabels);


                var view1 = Ti.UI.createView({
                             width :Ti.UI.SIZE,
                                    height : Ti.UI.SIZE,
                                    //width :'auto',
                                    //height : 'auto',
                                    //top:testLabels.top+testLabels.height,
                                    left: 0,
                                    layout : 'horizontal'
                });
                scrollView.add(view1);

                Switch[0] = Ti.UI.createSwitch({
                             style: Ti.UI.Android.SWITCH_STYLE_CHECKBOX,
                             //textAlign:Ti.UI.TEXT_ALIGNMENT_CENTER,
                             //title:'Notifications',
                             //top: 10,
                             value:false,
                            // color: 'blue',
                            left: 10,
                            //right: 0,
                             visible: false,
                            // height: Titanium.UI.SIZE,
                            //verticalAlign : 10,
                           //TEXT_VERTICAL_ALIGNMENT_TOP:5 ,
                             //left : 2,
                             width : Ti.UI.SIZE,
                                      //textAlign : 'center',
                                      height : 'auto',
                             //width: Ti.UI.SIZE
                           });
             //top_label += answerLabels[0].top + answerLabels[0].height;
            // alert(top_label);
            view1.add(Switch[0]);

               answerLabels[0] = Titanium.UI.createLabel({

                                                            left : 10,
                                                            //right: 10,
                                                            text:'',
                                                            color: 'black',
                                                           // width: '70%',
                                                          // height: 200,
                                                           width:'80%',
                                                           height :
Ti.UI.SIZE,
                                                            //textAlign :
'left',
                                                            touchEnabled:
true
                                    });

                //scrollView.add(switch_view);

                         view1.add(answerLabels[0]);
```

```
            var view2 = Ti.UI.createView({
                        width : Ti.UI.SIZE,
                              height : Ti.UI.SIZE,
                              //top:view1.top+view1.height,
                              left: 0,
                              layout : 'horizontal'
            });
            scrollView.add(view2);

                           Switch[1] = Ti.UI.createSwitch({
                                                 style:
Ti.UI.Android.SWITCH_STYLE_CHECKBOX,

//textAlign:Ti.UI.TEXT_ALIGNMENT_CENTER,
                                                  //title:'Notifications',
                                                  //top: 10,
                                                  value:false,
                                                 // color: 'blue',
                                                 left: 10,
                                                 //right: 0,
                                                  visible: false,
                                                 // height:
Titanium.UI.SIZE,
                                                 //verticalAlign : 10,

//TEXT_VERTICAL_ALIGNMENT_TOP:5 ,
                                                  //left : 2,
                                                  width : Ti.UI.SIZE,
                                                           //textAlign :
'center',
                                                           height : 'auto',
                                                  //width: Ti.UI.SIZE
                                              });
                                  view2.add(Switch[1]);

            answerLabels[1] = Titanium.UI.createLabel({

                                                            left : 10,
                                                            text:'',
                                                            color: 'black',
                                                            width:'80%',
                                                          //  height: 200,
                                                          //width:Ti.UI.SIZE,
                                                          height:Ti.UI.SIZE,
                                                            textAlign :
'left',

                                                            touchEnabled:
true
                                         });

                  view2.add(answerLabels[1]);
         //scrollView.add(answerLabels[1]);
         //top_label += answerLabels[1].top + answerLabels[1].height;
         var view3 = Ti.UI.createView({
                        width : Ti.UI.SIZE,
                              height : Ti.UI.SIZE,
                              //top:view2.top+view2.height+5,
```

```
                                left: 0,
                                layout : 'horizontal'
            });
            scrollView.add(view3);

      Switch[2] = Ti.UI.createSwitch({
                                          style:
Ti.UI.Android.SWITCH_STYLE_CHECKBOX,

                                                   //title:'Notifications',
                                                   //top: 10,
                                                   value:false,
                                                  // color: 'blue',
                                                    left: 10,
                                                  //right: 0,
                                                   visible: false,
                                                  // height:
Titanium.UI.SIZE,
                                                  //verticalAlign : 10,

//TEXT_VERTICAL_ALIGNMENT_TOP:5 ,
                                                   //left : 2,
                                                   width : Ti.UI.SIZE,
                                                            //textAlign :
'center',
                                                            height :
Ti.UI.SIZE,
                                                   //width: Ti.UI.SIZE
                        });
        view3.add(Switch[2]);

         answerLabels[2] = Titanium.UI.createLabel({

                                                             left : 10,
                                                             text:'',
                                                             color: 'black',
                                                            width:'80%',
                                                             // height: 200,
                                                           //width:Ti.UI.SIZE,
                                                           height:Ti.UI.SIZE,
                                                             textAlign :
'left',

                                                             touchEnabled:
true
                                        });
         //scrollView.add(answerLabels[2]);
         view3.add(answerLabels[2]);

         //  top_label += answerLabels[2].top + answerLabels[2].height;
          var view4 = Ti.UI.createView({
                        width : Ti.UI.SIZE,
                                height : Ti.UI.SIZE,
                                //top:view3.top+view3.height+5,
                                left: 0,
                                layout : 'horizontal'
            });
            scrollView.add(view4);
```

```
              Switch[3] = Ti.UI.createSwitch({
                                          style:
Ti.UI.Android.SWITCH_STYLE_CHECKBOX,

//textAlign:Ti.UI.TEXT_ALIGNMENT_CENTER,
                                                  //title:'Notifications',
                                                  //top: 10,
                                                  value:false,
                                                 // color: 'blue',
                                                 left: 10,
                                                 //right: 0,
                                                  visible: false,
                                                 // height:
Titanium.UI.SIZE,
                                                 //verticalAlign : 10,

//TEXT_VERTICAL_ALIGNMENT_TOP:5 ,
                                                  //left : 2,
                                                  width : Ti.UI.SIZE,
                                                           //textAlign :
'center',
                                                           height :
Ti.UI.SIZE,
                                                  //width: Ti.UI.SIZE
                            });
                      view4.add(Switch[3]);

          answerLabels[3] = Titanium.UI.createLabel({
                                                                 //borderRadius:
35,

//backgroundcolor: 'AliceBlue',
                                                                   left : 10,
                                                                //  borderColor:
'white',
                                                                   //borderRadius :
10,
                                                                   text:'',
                                                                   color: 'black',
                                                                  // width :
Ti.UI.SIZE,
                                                                  width:'80%',
                                                                 // height: 200,
                                                                   height :
Ti.UI.SIZE,
                                                                   textAlign :
'left',
                                                                   touchEnabled:
true
                                          });
                    view4.add(answerLabels[3]);
/*
 * ---Switch [check Box] configuraiton for correct Answer End--
 */

              var view5 = Ti.UI.createView({
                            width : Ti.UI.SIZE,
                                  height : Ti.UI.SIZE,
```

```
                                //top:view3.top+view3.height+5,
                                left: 0,
                                layout : 'horizontal'
             });
             scrollView.add(view5);

                    var skipbutton = Titanium.UI.createButton({
//create question/ans skip button
                title : 'Skip',
                color: 'white',
                left:'5dp',
                bottom:'10dp',
                height:'40dp',
                width:'100dp',
                visible: false,
                font : {fontFamily : 'Arial',fontWeight : 'bold',fontSize :
14},
                backgroundcolor: 'black',
               // backgroundSelectedColor: '#7FFFD4'
            });
            view5.add(skipbutton);

              var nextbutton = Titanium.UI.createButton({
//create question/ans next button
                title : 'Next',
                color: 'white',
                //right:'5dp',
                left:'95dp',
                bottom:'10dp',
                height:'40dp',
                width:'100dp',
                visible: false,
                font : {fontFamily : 'Arial',fontWeight : 'bold',fontSize :
14},
                backgroundcolor: 'black',
                //backgroundSelectedColor: '#7FFFD4'
            });
            view5.add(nextbutton);

            var dialouge= Ti.UI.createAlertDialog({                        //
For exam begin confirmation
                    title: 'Quiz is about to Begin! Proceed?',
                    buttonNames: ['No','Yes'],
                });

            dialouge.addEventListener('click',function(e){
             if(e.index == 0)
                   //No
                   dialouge.hide();
             else if (e.index == 1){
                   //Yes                                        //Exam start
configuration
                    picker.hide();

                    quiz_idlabel.visible =  false;
                    startbutton.hide();
                   // startbutton.visible = false;
                    nextbutton.visible = true;
                    skipbutton.visible = true;
```

```
                    logout.visible = false;
                    window_exam.addEventListener('android:back',function(){

                                                    });


                            //-------server connection main 1---------
                     var ajax = Ti.Network.createHTTPClient();
                     ajax.onerror = function(e){
                        alert('Error');
                        };
                     ajax.onload = function(){
                        Titanium.API.info(this.responseText);
                        var data = this.responseText;
                        var jdata = JSON.parse(data);
                         if(jdata.success){   //success start
                            var row_data = [];
                            var rows=jdata.data;
                            questions = rows;
                            var display_lbl = Titanium.UI.createLabel({

                                                    height : 40,
                                                    width : 100,

                                                    left : '40dp',
                                                color : '#fff',
                                                borderRadius : 10,

        backgroundColor : '#000',
                                                font : {

        fontSize : 24,

        fontWeight : 'bold'
                                               },
                                                textAlign : 'center'
                                    });

                    view5.add(display_lbl);

                    var quiz_time= jdata.quiz.quiz_time;
                    user_quiz= jdata.user_quiz;


                            min = quiz_time / 60 ;
                            sec = quiz_time % 60 ;


                            var my_timer = new countDown(min, sec ,
                                                function()   {
                                                    //something here...
                                                    display_lbl.text =
my_timer.time.m + " : " + my_timer.time.s;
                                                },
                                                function() {
// when time is up
                                                    // something here...
                                                    alert('Time is up');
```

```
                                                    //my_timer.stop();
                                                   view1.hide();
                                                   view2.hide();
                                                   view3.hide();
                                                   view4.hide();
                                                   view5.hide();
                                                  testLabels.visible = false;
                                                quiz_idlabel.visible =  true;
                                                   picker.show();
                                                   logout.visible = true;

                                                   //server connection 2
                                                    var ajax =
Ti.Network.createHTTPClient();
                                          ajax.onerror = function(e){
                                              alert('Error');
                                          };
                                          ajax.onload = function(){

Titanium.API.info(this.responseText);
                                              var data = this.responseText;
                                              var jdata = JSON.parse(data);

                                               if(jdata.show_result){
                                                     // alert('success');
                                                     var result =
jdata.result;

                                                     var dialouge=
Ti.UI.createAlertDialog({
                     title: 'Your Score is: '+result,

                     buttonNames: ['ok'],

                 });

         dialouge.show();

         dialouge.addEventListener('click',function(e){

                                  if(e.index == 0)

                                      //no

                                      dialouge.hide();



                                });
                                               }
                                               else{
                                                      alert("Quiz Finished,
Thanks for your patience");
                                                      //alert(jdata.msg);
                                                  }
                                              };
```

```
                                        ajax.open('POST',
'http://www.skimranhossain.com/api/get-front-result.php');
                                        ajax.send({
                                                'quiz_id': user_quiz


                                              });


                                          }
                                    );

                                    my_timer.start();


                                //scrollView.visible = true;

                                var question=rows[0].question;
                                testLabels.text =  question;

                               // aSwitch.visible = true;
                               // bSwitch.visible = true;
                               // cSwitch.visible = true;
                               // dSwitch.visible = true;

                                for (j=0; j< 4; j++){      //answers label

                                            if(rows[0].answers.length > j){
                                                  answer =
rows[0].answers[j].answer;

                                                  answerLabels[j].text = answer;
                                                  Switch[j].visible = true;
                                             }
                                            else{
                                                answerLabels[j].visible = false;
                                                Switch[j].visible = false;
                                            }
                                       }
                               nextbutton.addEventListener('click',function(e){
//next click event

                                      // var  answer_id = 0;
checking ; declear varibale above four


                                                          var wrong=0;
      for(i=0;i< questions[current_question].answers.length; i++){
                                                           switchVal=0;

      if(Switch[i].value == true)

      switchVal=1;

                                                                        if(switchVal
!= questions[current_question].answers[i].is_correct)

      wrong=1;

                                                           }
```

```
                                                    var is_correct=1;
                                                    if(wrong==1)
                                                        is_correct=0;


                                        var ajax = Ti.Network.createHTTPClient();
                                        ajax.onerror = function(e){
                                            alert('Error');
                                        };
                                        ajax.onload = function(){
                                            Titanium.API.info(this.responseText);
                                            var data = this.responseText;
                                            var jdata = JSON.parse(data);
                                             if(jdata.success){
                                                   // alert('success');
                                             }
                                             else{
                                                    alert(jdata.msg);
                                                }
                                            };
                                            ajax.open('POST',
'http://www.skimranhossain.com/api/add-user-answer.php');
                                             ajax.send({
                                                    'quiz_id': user_quiz,
                                                    'user_id':loginid,
                                                    'ques_id' :
questions[current_question].ques_id,
                                                    'is_correct': is_correct
                                                    /*,


                                                    'option1_ans' : option1_ans,
                                                    'option2_ans': option2_ans,
                                                    'option3_ans':option3_ans,
                                                    'option4_ans':option4_ans*/
                                                    //'ans_id' : answer_id
                                                });
                                                /* console.log({
                                                    'quiz_id': quiz_id,
                                                    'user_id':loginid,
                                                    'ques_id' :
questions[current_question].ques_id,
                                                    'option1_ans' : option1_ans,
                                                    'option2_ans': option2_ans,
                                                    'option3_ans':option3_ans,
                                                    'option4_ans':option4_ans
                                                    //'ans_id' : answer_id
                                                });*/
                                                //
                                        // alert(answer_id);
                                         current_question = current_question+1;
                                         if(current_question == questions.length){
//end of question/exam end
                                            //go back quid picker page
                                            //alert('finish');
                                             //nextbutton.visible = false;
```

```
                                                    //skipbutton.visible =
false;
                                                    my_timer.stop();
                                                    view1.hide();
                                                    view2.hide();
                                                    view3.hide();
                                                    view4.hide();
                                                    view5.hide();
                                                    testLabels.visible =
false;
                                                    quiz_idlabel.visible =
true;
                                                    picker.show();
                                                    logout.visible = true;
                                                    //startbutton.show();

                                                     var ajax =
Ti.Network.createHTTPClient();  //server call for show result
                                                ajax.onerror = function(e){
                                                    alert('Error');
                                                };
                                                ajax.onload = function(){

Titanium.API.info(this.responseText);
                                                    var data = this.responseText;
                                                    var jdata = JSON.parse(data);

                                                     if(jdata.show_result){   //
result will show or not (checking)
                                                          // alert('success');
                                                          var result =
jdata.result;    //how may correct
                                                          //alert('Your Score is'
+result);
                                                          var dialouge=
Ti.UI.createAlertDialog({              // show result

                    title: 'Your Score is: '+result,

                    buttonNames: ['ok'],

                });

        dialouge.show();

        dialouge.addEventListener('click',function(e){

                                    if(e.index == 0)

                                        //no

                                        dialouge.hide();



                                });
                                                          }
                                                          else{
```

```
                                                    alert("Quiz Finished,
Thanks for your patience");
                                                    //alert(jdata.msg);
                                                }
                                            };
                                        ajax.open('POST',
'http://www.skimranhossain.com/api/get-front-result.php');
                                        ajax.send({
                                                    'quiz_id': user_quiz


                                                });




                                }
                                else{
                                        //switch refresh start
                                            //Switch[0].value==false;
                                            //Switch[1].value==false;
                                            //Switch[2].value==false;
                                            //Switch[3].value==false;
                                        question =
questions[current_question].question;
                                        testLabels.text =  question;
                                        for (j=0; j< 4; j++){
//answers label


if(questions[current_question].answers.length > j){

answerLabels[j].visible = true;

answer = questions[current_question].answers[j].answer;

answerLabels[j].text = answer;

Switch[j].visible = true;

Switch[j].value = false;  // refresh the switch button for next answers
                                                                }

                                                            else{

answerLabels[j].visible = false;

Switch[j].visible = false;
                                                                }

                                        }

                                }
```

PHP Form

```
<?php
set_time_limit(0);
ini_set("memory_limit","-1");
ini_set("max_execution_time","86400");

    /*if(!isset($_POST['api_key']) || $_POST['api_key'] != 'imran1234'){
        die ('Restricted Access!');
    }*/

    /*$data['varname']=$_POST['var'];
    $content = json_encode($data);
    header('Content-Type: application/json');
      echo $content;

    die;*/
    //$_POST=$_GET;

    include_once 'config.php';

    $data=array();
    $success=0;
    $msg='';

    //delete block
    if(isset($_POST['delete_id'])){
      mysql_query("DELETE FROM quiz WHERE quiz_id=".$_POST['delete_id']);  //
for delete quiz
      $success=1;
    }
    elseif(!empty($_POST['quiz_name'])){

      //edit block if id is passed
        if(isset($_POST['id'])){
             $sql="UPDATE quiz SET quiz_name='".$_POST['quiz_name']."',
active=".$_POST['active'].", show_result=".$_POST['show_result'].",
quiz_time=".$_POST['quiz_time']." WHERE quiz_id=".$_POST['id'];
                  if(mysql_query($sql)){
                      $user_result=mysql_query("SELECT * FROM quiz ORDER BY
quiz_id DESC");

                      while($row=mysql_fetch_array($user_result)){
                          $data[]=array("quiz_id" => $row['quiz_id'],
"quiz_name" => $row['quiz_name'], "active" => $row['active'], "show_result"
=> $row['show_result']);
                      }
                      $success=1;
                  }
                  else{
                      $msg=$sql; //'Error while updating record.';
                  }
        }else{
                          //add quiz

        $exists_res=mysql_query("SELECT COUNT(*) as cnt FROM quiz WHERE
quiz_name='".$_POST['quiz_name']."'");
        $exists_row=mysql_fetch_array($exists_res);
```

```
        if($exists_row['cnt'] > 0){
            $msg='Quiz already exists.';
        }
        else{
            $sql="INSERT INTO quiz
VALUES(NULL,'".$_POST['quiz_name']."','".$_POST['active']."','".$_POST['show_res
ult']."','".$_POST['quiz_time']."')";
            if(mysql_query($sql)){
                $user_result=mysql_query("SELECT * FROM quiz ORDER BY quiz_id
DESC");

                while($row=mysql_fetch_array($user_result)){
                    $data[]=array("quiz_id" => $row['quiz_id'], "quiz_name"
=> $row['quiz_name'], "active" => $row['active'], "show_result" =>
$row['show_result'], "quiz_time" => $row['quiz_time']);
                }
                $success=1;
            }
            else{
                $msg='Error while saving data.';
            }
        }
        }

    }
    else{
        $msg='Please complete all fields.';
    }

$results=array();
$results['data']=$data;
$results['success']=$success;
$results['msg']=$msg;

//$content = $_GET['callback']. '(' . json_encode($results). ')';
$content = json_encode($results);

header('Content-Type: application/json');
echo $content;


?>


<?php
set_time_limit(0);
ini_set("memory_limit","-1");
ini_set("max_execution_time","86400");

    /*if(!isset($_POST['api_key']) || $_POST['api_key'] != 'imran1234'){
        die ('Restricted Access!');
    }*/

    /*$data['varname']=$_POST['var'];
    $content = json_encode($data);
    header('Content-Type: application/json');
      echo $content;

    die;*/
```

```
    //$_POST=$_GET;

    include_once 'config.php';

    $data=array();
    $success=0;
    $msg='';


    if(isset($_POST[id])){
       $ques_id=$_POST['delete_id'];

       mysql_query("DELETE FROM questions WHERE ques_id=".$ques_id);
       mysql_query("DELETE FROM answers WHERE ques_id=".$ques_id);

       $success=1;
    }
    elseif(!empty($_POST['question']) && !empty($_POST['quiz_id'])  &&
!empty($_POST['ques_order'])){


       if(isset($_POST['id'])){
             $sql="UPDATE questions SET question='".$_POST['question']."',
ques_order=".$_POST['ques_order']."., quiz_id=".$_POST['quiz_id']." WHERE
ques_id=".$_POST['id'];

              if(mysql_query($sql)){
                  $ques_id=$_POST['id'];

                  mysql_query("DELETE FROM answers WHERE ques_id=".$ques_id);
// delete all answer for ques_id coming from UI to insert all answers again.

                  //$sql="INSERT INTO answers
VALUES(NULL,".$ques_id.",".$_POST['active'].")";

                  if($_POST['option1']){
                        //$is_correct=($_POST['answer_number'] == 1)? 1 :0 ;
                        mysql_query("INSERT INTO answers
VALUES(NULL,".$ques_id.",'".$_POST['option1']."',".$_POST['option1_ans'].")")
;
                  }

                 if($_POST['option2']){
                        //$is_correct=($_POST['answer_number'] == 2)? 1 :0 ;
                        mysql_query("INSERT INTO answers
VALUES(NULL,".$ques_id.",'".$_POST['option2']."',".$_POST['option2_ans'].")")
;
                  }

                  if($_POST['option3']){
                        //$is_correct=($_POST['answer_number'] == 3)? 1 :0 ;
                        mysql_query("INSERT INTO answers
VALUES(NULL,".$ques_id.",'".$_POST['option3']."',".$_POST['option3_ans'].")")
;
                  }

                  if($_POST['option4']){
                        //$is_correct=($_POST['answer_number'] == 4)? 1 :0 ;
```

```
                        mysql_query("INSERT INTO answers
VALUES(NULL,".$ques_id.",'".$_POST['option4']."',".$_POST['option4_ans'].")")
;
                    }

                    $success=1;
                }
                else{
                    $msg="Could not update";
                }

        }
        else{ // ADD BLOCK
               //check same question exist in same quiz or not.
               $exists_res=mysql_query("SELECT COUNT(*) as cnt FROM questions
WHERE question='".$_POST['question']."' AND quiz_id=".$_POST['quiz_id']);
                $exists_row=mysql_fetch_array($exists_res);

                if($exists_row['cnt'] > 0){
                    $msg='This question already added to this quiz.';
                }
                else{
                      $sql="INSERT INTO questions
VALUES(NULL,'".$_POST['question']."',".$_POST['ques_order'].",".$_POST['quiz_
id'].")";
                     if(mysql_query($sql)){
                         $ques_id=mysql_insert_id();

                          //$sql="INSERT INTO answers
VALUES(NULL,".$ques_id.",".$_POST['active'].")";

                        if($_POST['option1']){
                                //$is_correct=($_POST['answer_number'] == 1)?
1 :0 ;
                                mysql_query("INSERT INTO answers
VALUES(NULL,".$ques_id.",'".$_POST['option1']."',".$_POST['option1_ans'].")")
;
                         }

                        if($_POST['option2']){
                                //$is_correct=($_POST['answer_number'] == 2)?
1 :0 ;
                                mysql_query("INSERT INTO answers
VALUES(NULL,".$ques_id.",'".$_POST['option2']."',".$_POST['option2_ans'].")")
;
                         }

                         if($_POST['option3']){
                                //$is_correct=($_POST['answer_number'] == 3)?
1 :0 ;
                                mysql_query("INSERT INTO answers
VALUES(NULL,".$ques_id.",'".$_POST['option3']."',".$_POST['option3_ans'].")")
;
                         }

                         if($_POST['option4']){
                                //$is_correct=($_POST['answer_number'] == 4)?
1 :0 ;
```

```
                                mysql_query("INSERT INTO answers
VALUES(NULL,".$ques_id.",'".$_POST['option4']."',".$_POST['option4_ans'].")")
;
                            }

                            $success=1;
                        }
                        else{
                            $msg='Error while saving data.';
                        }
                    }
            }


        }
        else{
            $msg='Please complete all fields.';
        }

$results=array();
$results['data']=$data;
$results['success']=$success;
$results['msg']=$msg;

//$content = $_GET['callback']. '(' . json_encode($results). ')';
$content = json_encode($results);

header('Content-Type: application/json');
echo $content;


?>
```