Clicker Android Native Application

Summer Internship - 2013

Submitted in fulfilment of internship project

By

Clicker Android Native Application Team

Under the Guidance of

Prof. D. B. Phatak

Department of Computer Science and Engineering

Indian Institute of Technology, Bombay

Acknowledgment

We, the summer interns of the team Clicker Software, are overwhelmed in all humbleness and gratefulness to acknowledge our deep gratitude to all those who have helped us put our ideas to perfection and have assigned tasks, well above the level of simplicity and into something concrete and unique.

We, whole heartedly thank **Padma Shri Deepak B Phatak** for having faith in us, selecting us to be a part of his valuable project and for constantly motivating us to do better.

We are very thankful to our mentors Mr. Rajesh Kushalkar, Mr.Rajavel , Ms. Dipti Ghosalkar, Mr.Chetan Jaiswal, Mr.Varun, Mr.Gobinath, Mr.Hitesh and Mr.Sachin for their valuable suggestions. They were and are always there to show us the right track when in trouble. With help of their brilliant guidance and encouragement, we all were able to complete our tasks properly and were up to the mark in all the tasks assigned. During the process, we got a chance to see the stronger side of our technical and non-technical aspects and also strengthen our concepts. Here by, we gladly consider ourselves to be the most fortunate batch of interns.

We would also like to thank Mr.Dileep for making our stay here comfortable and for all his administrative help. Without his mention our project would always be incomplete.

Last but not the least, we whole heartedly thank all our other colleagues working in different projects under Prof. D.B Phatak for helping us evolve better with their advice.

Team Members

1. Ajitha -- MIT, Anna University, Chennai

2. Gagan Kumar -- RGUIIIT, Basar

3. Law kumar -- ISM, Dhanabad

4. Pruthvi Gopal -- Nagarjuna University, Vijayawada

5. Rabart Kurrey -- NITK, Surathkal

6. Srilatha -- RGUIIIT, Basar

7. Swetha Pullur -- RGUIIIT, Basar

MENTORS

Mr.Rajavel: A great person who guided for all kinds of problems.

Mr. Tushar Kambli: Our guide for all problems

Mr.Rajesh Khusalkar: A very patient person who supervised all our project work.

Mr.Chetan: A very helpful person and he helped a lot in android coding and to understand the previous code.

Mr.Varun: A very helpful person and he helped a lot in android coding.

Ms Dipti Ghosalkar: Helped a lot in understanding the previous code and debugging process.

Mr. Harshavardan: Helped a lot in understanding the previous code and debugging process at server side.

Mr. Gobinath Mani: Showed a keen interest in our project work and helped us in networking problems.

Abstract

Clicker Software – is a MHRD sponsored project aimed to simplify interaction between students and Instructors in many colleges across India. It is a technology aimed to mainly reduce the manual labour; one which does same for both students and teachers. It is an affordable, interactive software utility for course registration, creating and conducting quizzes, generating reports, polling and clarifying doubts raised by a student precisely and in lesser time. It helps the instructor in analysing quizzes, reports etc. with ease.

The software part is a computational and analytical tool running on an instructor machine which uses MySQL for database storage, JAVA, JSP, Java-Script, J-query etc. for designing and utilizing the REST-Technology based model . The application is built on android platform. Client-side we are managing things at maximum level so that average load on the server will be less, compared to previous clicker versions.

Contents

Chapter 1

Introduction

Clicker is a student response system being developed at IIT Bombay as part of the National Mission on Education through ICT (NMEICT), MHRD.

The system can be used to capture responses from a live audience in real time. During a classroom session, each of the participants is provided an Aakash Tablet, which is associated with a unique ID. It can be used in a classroom to teach the students effectively and to evaluate performance of the students by keeping records of quizzes, generating reports and solving doubts. Clicker system provides the facility to the instructor to interact with students by presenting question, collecting answers of individual students, and evaluating the understanding level of each student in that session.

Some faculty refer to this as just in-time teaching making adjustments or adaptations to their course as needed in any given situation. Personal response technology works well both for large lecture halls and for smaller classes. Regardless of the class size, the instructor is always able to gauge student understanding due to the immediate feedback the technology provides. Results from class responses can be saved and then displayed for semester-long analyses that may inform subsequent curriculum development.

We developed a Native Android Application such that student can access the server without using the browser. There will not be persistent connection between client and server so that load on server is reduced.

1.1 Purpose

Native android application development for this software has provided an efficient and easier way to extract and manage the software. It does not require any installation and hence is user friendly. The main purpose of this native android application is to reduce the load on server and provide scalability for a considerable number of clients.

We use a robust database which is very flexible and easy to handle. Hence the required data can be manipulated and managed easily. It also provides an efficient way of learning, as the instructor can evaluate the understanding level of each student and help them to develop their overall concepts. It allows students to participate by submitting responses to questions and viewing immediate feedback.

The main purpose in making this document is to describe all the features and its working to make any enhancements in future. To increase the students interest in the classroom session and also increase their understanding.

1.2 Scope

Web based clicker had persistent connection between server and client and also used browser. So there is heavy load on server and also only few clients can be connected to a server according to router capacity. Now in native android application server-client connection is established only during request and response which reduces server and router load and enables scalable and robust system which can support many clients.

In ClickerV4, a simple device which has android as Operating System can do the job. Hence any person sitting at some place can access it and run it with single installation of ClickerV4 apk. This system allows for active participation by all students and provides immediate feedback to the instructor and the students can inform about their confusion or misunderstandings of the material being presented. Native android clickers represent an easy-to-adopt technology that can enhance the learning experience and help the student to have a better understanding of the courses in a friendly manner. Native android development of clicker, by virtue of its simplicity can be of great application in a classroom session and its use can also be increased in different colleges.

1.3 Definitions, Abbreviations

1	XML	e**X**tensible **M**arkup **L**anguage
2	REST	Representational State Transfer Model
3	JSP	Java Server Pages.
4	J2EE	Java 2 Enterprise Edition.
5	SQL	Structured Query Language.
6	IDE	Integrated Development Environment
7	GUI	Graphical User Interface.

8	UML	Unified Modelling Language.
9	ID	Identification number
10	Clicker	A system which is used to collect response from audience
11	HTTP	Hypertext Transfer Protocol.
12	JRE	Java run time environment
13	HTML	Hyper Text Markup Language
14	JS	JavaScript
15	JSON	JavaScript Object Notation
16	LOC	Line of code
17	JAX-RS	Java API for RESTful Web Services

1.4 Motivation

The motivation behind developing clicker was mainly to make interaction between students and professors in a classroom more efficient and less time consuming.

- To increase and manage student interaction in the classroom;
- To assess students understanding of a subject and misunderstanding of any content in a lecture;
- To guide thinking and review the learning materials;
- To create an interface which will allow a large number of students to participate and learn in a single classroom session and also create an interface which will allow students from remote center to participate in the classroom session conducted at some other place.

It allows to take quizzes of different patterns (single answer correct, Multiple answer correct,True/False, numeric, etc.) and a unique feature of raise hand (i.e. student clicks the raise hand button on clicker and his/her ID and the doubt is visible to the instructor, he will clarify it in the class. Clickers give the faculty the ability to fine-tune their instruction based on student response. Regardless of the class size, the instructor is always able to gauge student understanding due to the immediate feedback the technology provides. Results from class responses can be saved and then displayed in a format and can be saved for semester –long analyses that may inform subsequent curriculum development.

As far as internship work modules go the motivation was to add extra features and furnish the few software codes already written. The feature of Quiz conduction was added to evaluate student mastery of content and to identify concepts that are proving difficult for students to grasp. For students, they provide a quick way to validate their own learning, helping them identify areas that need improvement. The feature of Polling used to gauge student opinion on controversial or sensitive issues. It can be used to catalyse debate and discussion, turning a passive lecture into an interactive exchange. The Raise Hand module can be used when a student is having some doubt or confusion.

Chapter 2

Objective

The main purpose of this project is to not have persistent connection, so that number Of students accessing will increase, while this can be done by using rest technology where the connection gets terminated after sending response and also to reduce the load on server by maintaining all the student details including sessions, saved at client side. Along with these synchronising the server and client activities is to be taken care.

Instructor Module:

The aim of designing the module was to provide admin the facility to add, edit, fetch and delete the information about the department, course, instructor and student. Also provide instructors and students the login.

Raise Hand:

This module aimed to help the student in case he/she is having some doubt or confusion; it is just the raising hand in the classroom environment. The instructor can see the list of all the students who has pressed the raised hand button along with their doubts which can be discussed in the class.

Polling:

It aimed at getting the views of the students on some sensitive and controversial issues and helps the instructor in making a decision.

Create Quiz:

This module is for conducting quiz in the classroom session, this will provide the facility to get the question from the database according to the course selected by the instructor and also they can create the quiz as required by the instructor.

Chapter 3

Design Considerations

3.1 Software and Libraries

JAX-RS with Jersey

In order to simplify development of RESTful Web services and their clients in Java, a standard and portable JAX-RS API has been designed It support exposing the data in a variety of representation media types and abstract away the low-level details of the client-server communication. JAX-RS uses annotations to define the REST relevance of Java classes. *Jersey* is the reference implementation for this specification. Jersey contains basically a REST server and a REST client. On the server side Jersey uses a servlet which scans predefined classes to identify RESTful resources. This servlet analyzes the incoming HTTP request and selects the correct class and method to respond to this request. This selection is based on annotations in the class and methods. A REST web application consists therefore out of data classes (resources) and services. These two types are typically maintained in different packages as the Jersey servlet will be instructed via the ***web.xml*** to scan certain packages for data classes. JAX-RS supports the creation of XML and JSON via the Java Architecture for XML Binding (JAXB).

JAX-RS annotation

@PATH(give_path)

Sets the path to base URL + /give_path. The base URL is based on application name, the servlet and the URL pattern from the web.xml" configuration file.

@POST

Indicates that the following method will answer to a HTTP POST request

@GET

Indicates that the following method will answer to a HTTP GET request

@Produces (MediaType.APPLICATION_JSON)

This defines which MIME type is delivered by a method annotated with @GET

@Consumes(MediaType.APPLICATION_FORM_URLENCODED)

This defines which MIME type is consumed by this method.

@PathParam

Used to inject values from the URL into a method parameter.

Various JAX-RS annotation used in ClickerV4 project

@Path("quiz")

This is the root path ,used in rest web resource page.

@GET
@Path("/course/{id}/{mac}")
@Produces(MediaType.APPLICATION_JSON)

This is used for returning courselist of a student Whose Student_id is “**id”** and mac address is “**mac**”.

@GET
@Path("/sample/{courseid}")
@Produces(MediaType.APPLICATION_JSON)

This is used for retrieving quiz details in json form to student who requested for quiz with a specific courseid.

@POST
@Path("/question")
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
@Produces(MediaType.APPLICATION_JSON)

This is used for getting quiz response in json form of a student with a specific courseid.

```
@GET
@Path("/result/{courseid}/{studentid}")
@Produces(MediaType.APPLICATION_JSON)
```

Above annotation is used for giving all past result details of quizzes for a particular courseid.

```
 @GET
 @Path("/poll/{courseid}")
 @Produces(MediaType.APPLICATION_JSON)
```

This is used for retrieving poll questions details in json form to student with a specific courseid.

```
@POST
@Path("/poll")
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
@Produces(MediaType.APPLICATION_JSON)
```

This is used for getting poll response in json form of a student with a specific courseid.

```
@POST
 @Path("/raisequestion")
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
 @Produces(MediaType.APPLICATION_JSON)
```

This is used for getting raised questions from student to instructor with a specific courseid.

3.2 Technology Used

The technology used over here is REST technology.

3.2.1 Restful Web Services:

REST is not the system's architecture in itself, but it's a simply a set of constraints if applied on existing web services that lead to a Restful architecture. There are no. of constraints that if any web services follows then, it is a restful.

These constraints are:-

1. It must be a **client-server system**.

2.Second, but most important point, is that it has to be **Stateless.**it means each request should be independent of others or previous requests. Each request from client to server must contain all of the information necessary to understand the request, and cannot take advantage of any stored context on the server. There are **no permanent connection** b/n clients and servers here. **Session state is therefore kept entirely on the client.**

Advantage of Stateless:-

Scalability is improved because not having to store state between requests, allows the server component to quickly free resources, and further simplifies implementation because the server doesn’t have to manage resource usage across requests.

Disadvantage of Stateless:-

1. Decrease network performance by increasing the repetitive data (per-interaction overhead) sent in a series of requests, since that data cannot be left on the server in a shared context.

2. Reduces the server’s control over consistent application behaviour.

3. It has to support a **caching system**. Cache constraints require that the data within a response to a request be implicitly or explicitly labelled as cacheable or non-cacheable. If a response is cacheable, then a client cache is given the right to reuse that response data for later, equivalent requests.

4.It has to be **uniformly accessible**—each resource must have a unique address and a valid point of access. The uniform interface constraint defines the interface between clients and servers. The four guiding principles of the uniform interface are:

- Identification of resources.
- Manipulation of resources through representations.
- Self-descriptive messages.
- hypermedia as the engine of application state
- It has to be layered—it must support scalability.
- It should provide code on demand. Servers are able to temporarily extend or customize the functionality of a client by transferring logic to it that it can execute.

These constraints only define how data is transferred b/w Components, we can use any networking architectures available.it doesn't need any new networking protocols. Static websites are following rest constraints, but dynamic applications are not stateless, as servers require tracking users through container sessions or client-side cookie schemes. Therefore, we can say that the dynamic web is not normally built on the REST-like architectural style.

Abstractions that makes a System: Restful

1.Resource:-

Resource is anything that is addressable over the Web. By addressable, we mean resources that can be accessed and transferred between clients and servers Since we are using HTTP to communicate, we can transfer any kind of Information (text, xml, json, flash movies) that can be passed between clients and servers. Content-type of these responses is mentioned in the Response header of HTTP.

2. Representation:-

The representation of resources is what is sent back and forth between clients and servers. A representation is a temporal state of the actual data located in some storage device at the time of a request. In general terms, it's a binary stream together with its metadata that describes how the stream is to be consumed by either the client or the server. Different clients are able to consume different representations of the same resource. Therefore, a representation can take various forms, such as an image, a text file, or an XML stream or a JSON stream, but has to be available through the same URI.

3. URI (Uniform Resource Identifier) :-

In a Restful web service, it is a hyperlink to a resource, and it's the only means for clients and servers to exchange representations. In a Restful system, the URI is not meant to change over time as the architecture's implementation is what manages the services, locates the resources, negotiates the representations, and then sends back responses with the requested resources.

Different Means to Exchange Data Representation b/w Clients & Servers

Developing Restful web services is similar to what we've been doing up to this point with our web applications. However, the fundamental difference between modern and traditional web application development is how we think of the actions taken on our data abstractions. Restful development is rooted in the concept of nouns (exchange of resources) while others development is rooted in the concept of verbs (remote actions taken on data) that is RPC-like service.so, for Remote Procedure Call there are millions of actions , but for restful, we have only specific actions that we can take upon resources—**Create, Retrieve, Update, and Delete(CRUD).**

Map of CRUD action to the HTTP methods:-

Data action of HTTP protocol equivalent:

CREATE	-	POST
RETRIEVE	-	GET
UPDATE	-	PUT
DELETE	-	DELETE

In their simplest form, Restful web services are networked applications that manipulate the state of resources. Resource manipulation means resource creation, retrieval, update, and deletion.

Descriptions of Restful technology used in clickerV4 project:-

1. Register Jersey as the servlet dispatcher for REST requests

In our web.xml file, we want to direct all requests to a servlet called Jersey REST Service. We will also tell this service where to find the resources that we want to make available to our android client app. The parameter "com.sun.jersey.config.property.package" defines in which package jersey will look for the web service classes. This property point to our resources classes. Here restresource page are having rest services class. In the servlet-mapping section, we create a global url-pattern, which essentially says that any request that goes to **/rest/** will attempt to be mapped to the appropriate methods.

web.xml

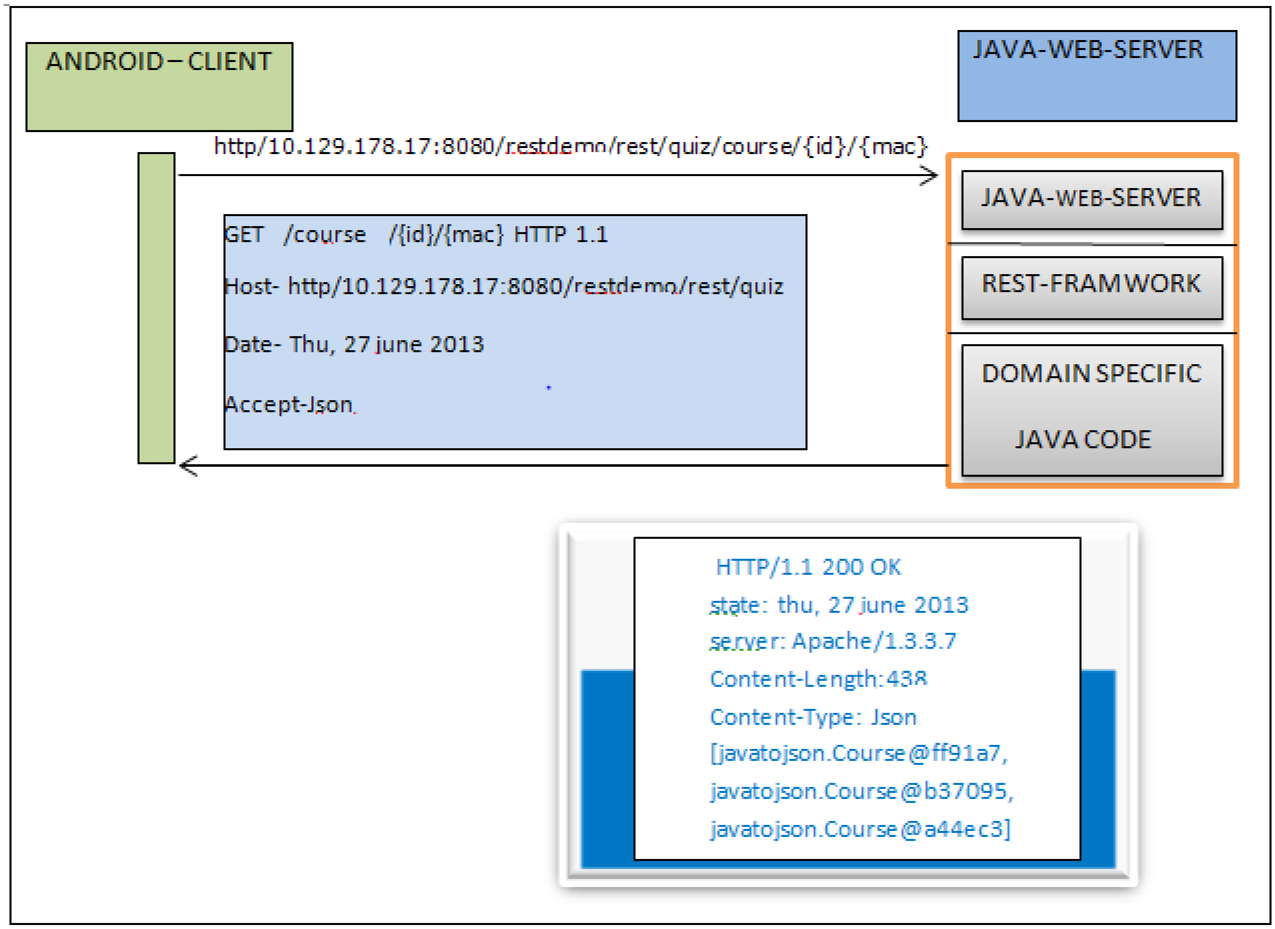
```
<servlet>
<servlet-name>Jersey REST Service</servlet-name>
<servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
<init-param>
<param-name>com.sun.jersey.config.property.packages</param-name>
<param-value>restresource</param-value>
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>Jersey REST Service</servlet-name>
<url-pattern> /rest/* </url-pattern>
</servlet-mapping>
```

Suppose server ip :-10.129.178.17
Base URI for accessing web-services- http://**10.129.178.17:8080/restdemo/rest/quiz**

MIME or representation format used for restful services- **JSON**

2. Different GET and POST methods used for communication between android clients and server in REST web Resource page:-

a) Get methods for Retrieving Json representation of courselist and total no. of student registers with respective course:-
Json representation used :- [javatojson.Course@ff91a7, javatojson.Course@b37095, javatojson.Course@a44ec3]



```
GET   /course   /{id}/{mac} HTTP 1.1
Host- http/10.129.178.17:8080/restdemo/rest/quiz
Date- Thu, 27 june 2013
Accept-Json
```

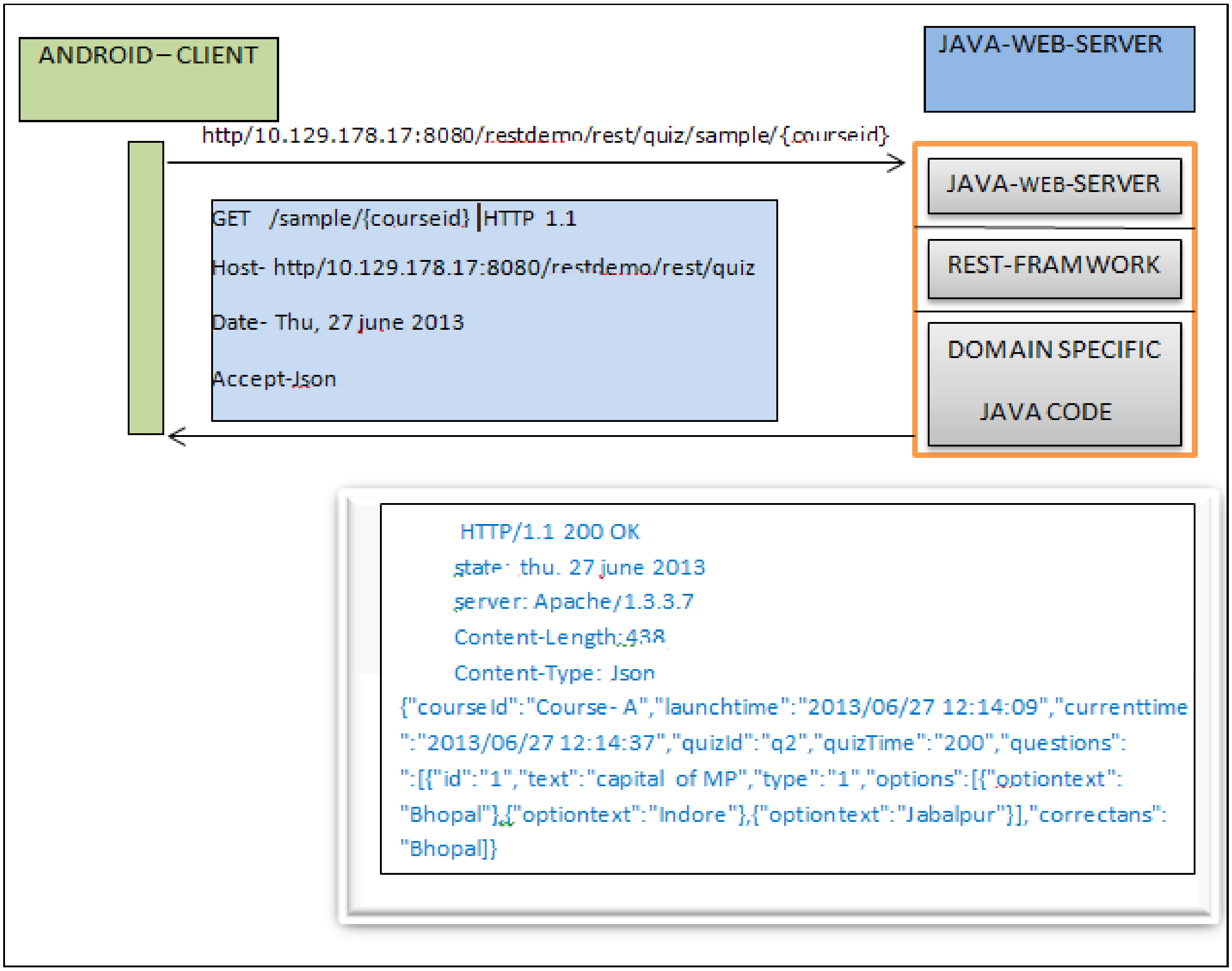
```
HTTP/1.1 200 OK
state: thu, 27 june 2013
server: Apache/1.3.3.7
Content-Length:438
Content-Type: Json
[javatojson.Course@ff91a7,
javatojson.Course@b37095,
javatojson.Course@a44ec3]
```

All the messages between client and server are standard HTTP protocol calls. For every retrieve action, we send a GET request and we get an HTTP response back with the payload of the response being the representation of the resource or, if there is a failure, a corresponding HTTP error code.

b) Get methods for Retrieving Json representation of quiz details with respect to corresponding courseid:-

Json Representaion for quiz-

{"courseId":"Course-A","launchtime":"2013/06/27 12:14:09","currenttime":"2013/06/27 12:14:37","quizId":"q2","quizTime":"200","questions":[{"id":"1","text":"capitalofMP","type":"1","options":[{"optiontext":"Bhopal"},{"optiontext":"Indore"}, {"optiontext":"Jabalpur"}],"correctans":"Bhopal]}

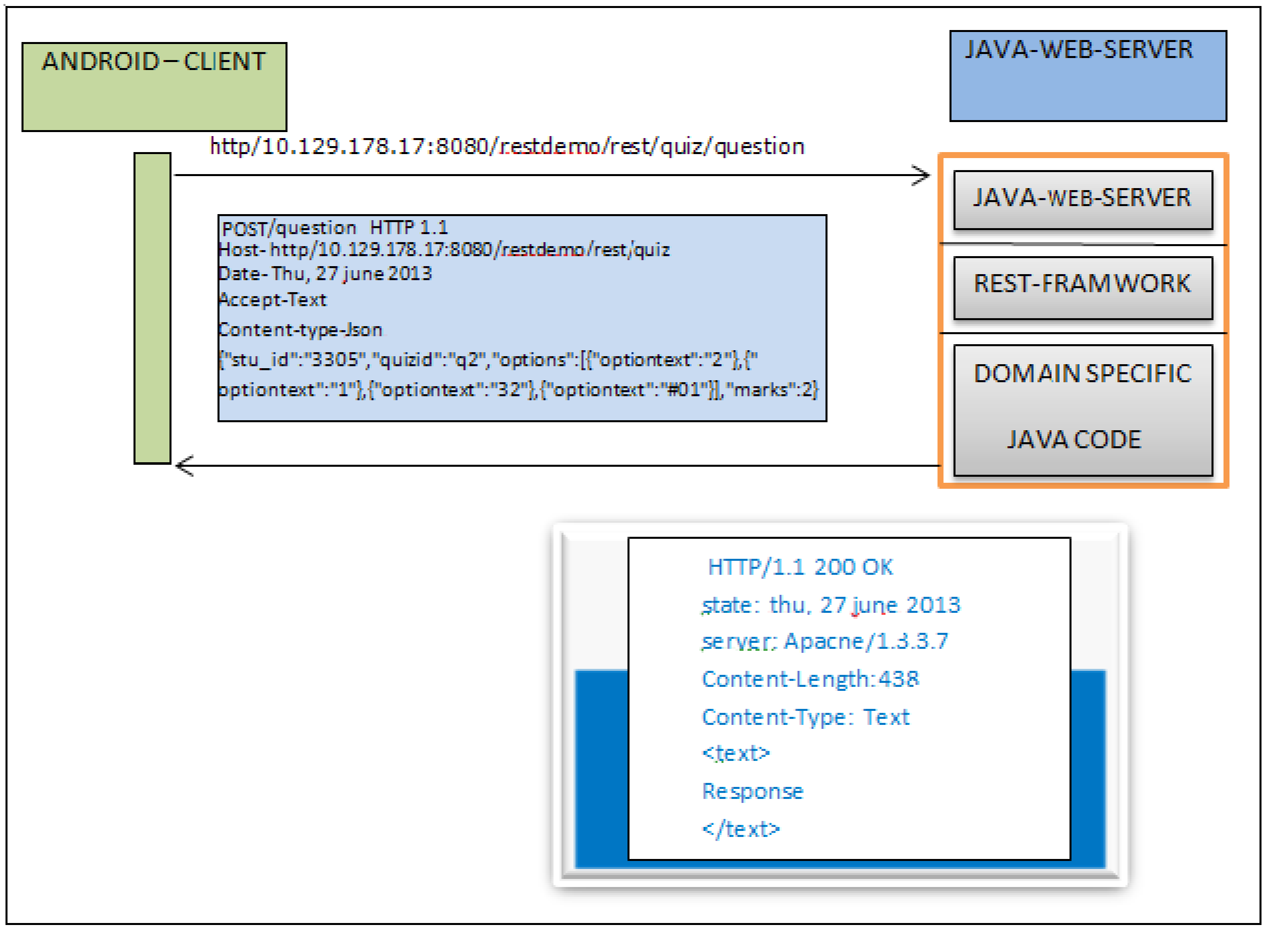


```
GET   /sample/{courseid} HTTP 1.1

Host- http/10.129.178.17:8080/restdemo/rest/quiz

Date- Thu, 27 june 2013

Accept-Json
```

```
        HTTP/1.1 200 OK
       state: thu. 27 june 2013
       server: Apache/1.3.3.7
       Content-Length:438
       Content-Type: Json
{"courseId":"Course- A","launchtime":"2013/06/27 12:14:09","currenttime
":"2013/06/27 12:14:37","quizId":"q2","quizTime":"200","questions":
":[{"id":"1","text":"capital of MP","type":"1","options":[{"optiontext":
"Bhopal"},{"optiontext":"Indore"},{"optiontext":"Jabalpur"}],"correctans":
"Bhopal]}
```

C) POST methods for Getting Json representation of quiz response of each individual student with respect to corresponding courseid:-

Consumes Json representation-
{"stu_id":"3305","quizid":"q2","options":[{"optiontext":"2"},{"optiontext":"1"},{"optiontext":"32"},{"optiontext":"#01"}],"marks":2}

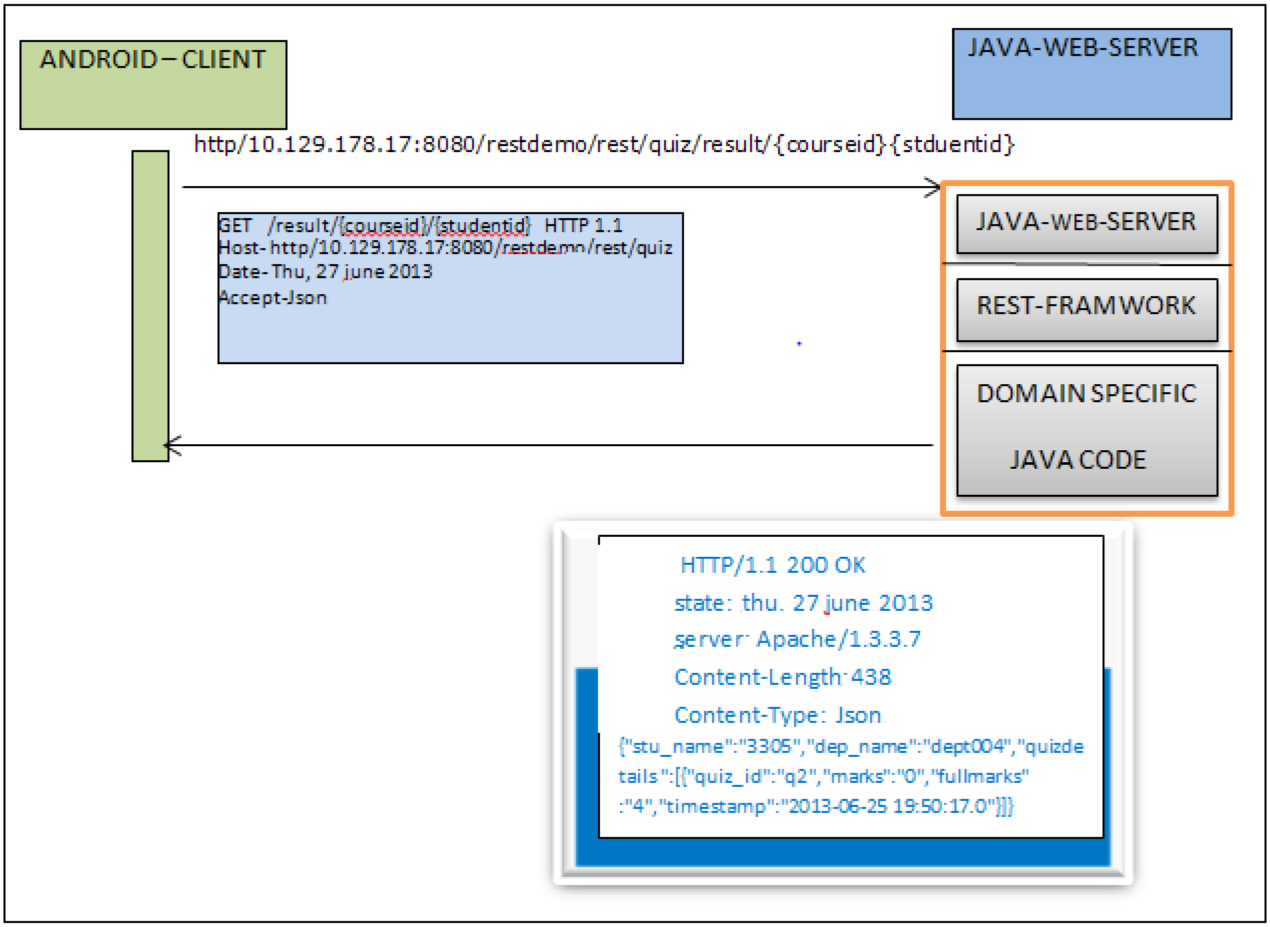


```
POST/question  HTTP 1.1
Host- http/10.129.178.17:8080/restdemo/rest/quiz
Date- Thu, 27 june 2013
Accept-Text
Content-type-Json
{"stu_id":"3305","quizid":"q2","options":[{"optiontext":"2"},{"
optiontext":"1"},{"optiontext":"32"},{"optiontext":"#01"}],"marks":2}
```

```
 HTTP/1.1 200 OK
state: thu, 27 june 2013
server: Apacne/1.3.3.7
Content-Length:438
Content-Type: Text
<text>
Response
</text>
```

d) Get methods for Retrieving Json representation of all past Result of quizzes given by a particular student for a specific courseid from server to Android client :-

Json representation-
{"stu_name":"3305","dep_name":"dept004","quizdetails":
[{"quiz_id":"q2","marks":"0","fullmarks":"4","timestamp":"2013-06-25 19:50:17.0"}]}

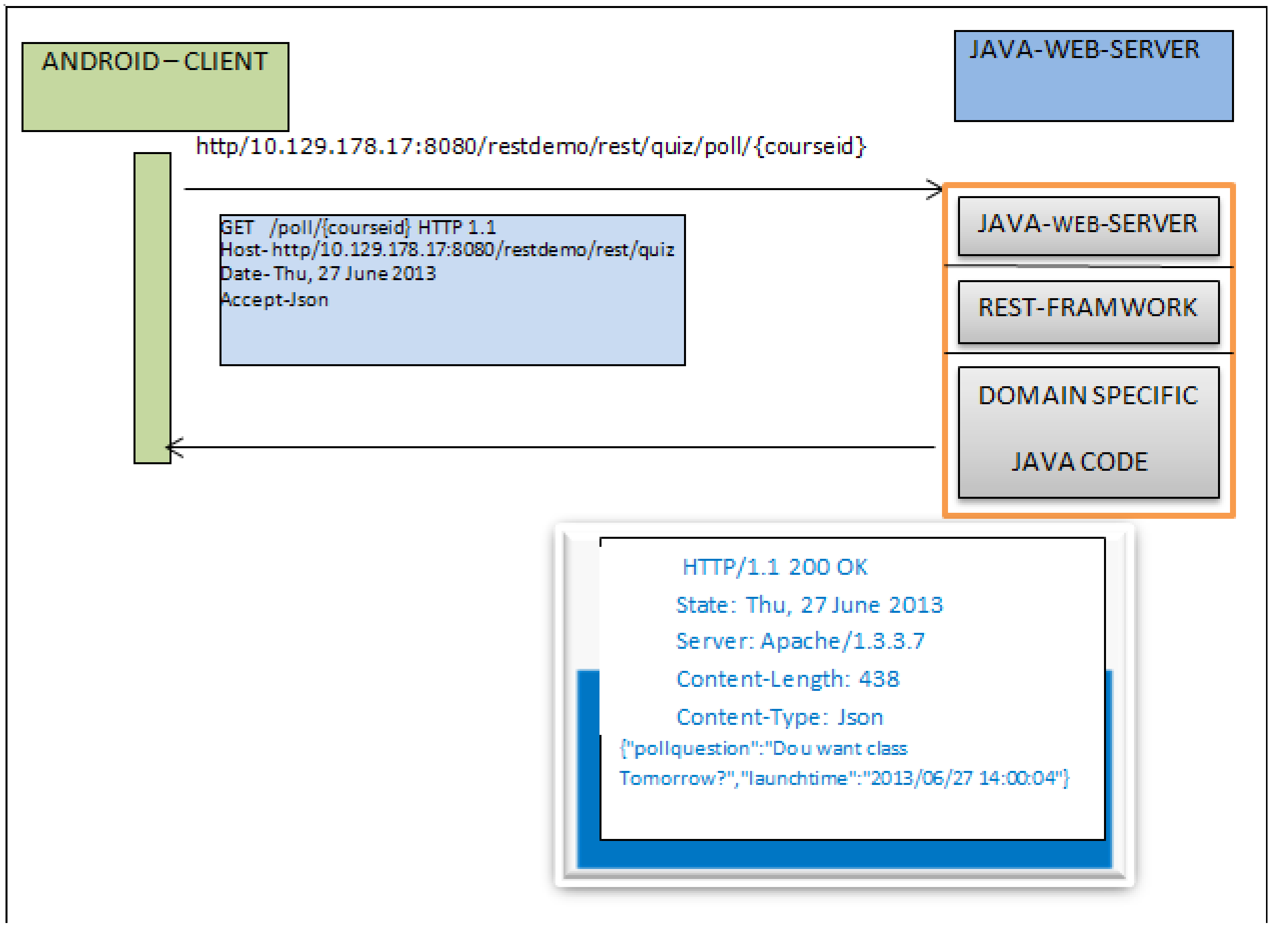


```
GET   /result/{courseid}/{studentid}   HTTP 1.1
Host- http/10.129.178.17:8080/restdemo/rest/quiz
Date- Thu, 27 june 2013
Accept-Json
```

```
HTTP/1.1 200 OK
state: thu. 27 june 2013
server: Apache/1.3.3.7
Content-Length: 438
Content-Type: Json
{"stu_name":"3305","dep_name":"dept004","quizde
tails":[{"quiz_id":"q2","marks":"0","fullmarks"
:"4","timestamp":"2013-06-25 19:50:17.0"}]}
```

e) Get methods for Retrieving Json representation of Poll question details with respect to corresponding courseid:-

Json Representation-
{"pollquestion":"Do u want class Tomorrow?","launchtime":"2013/06/27 14:00:04"}

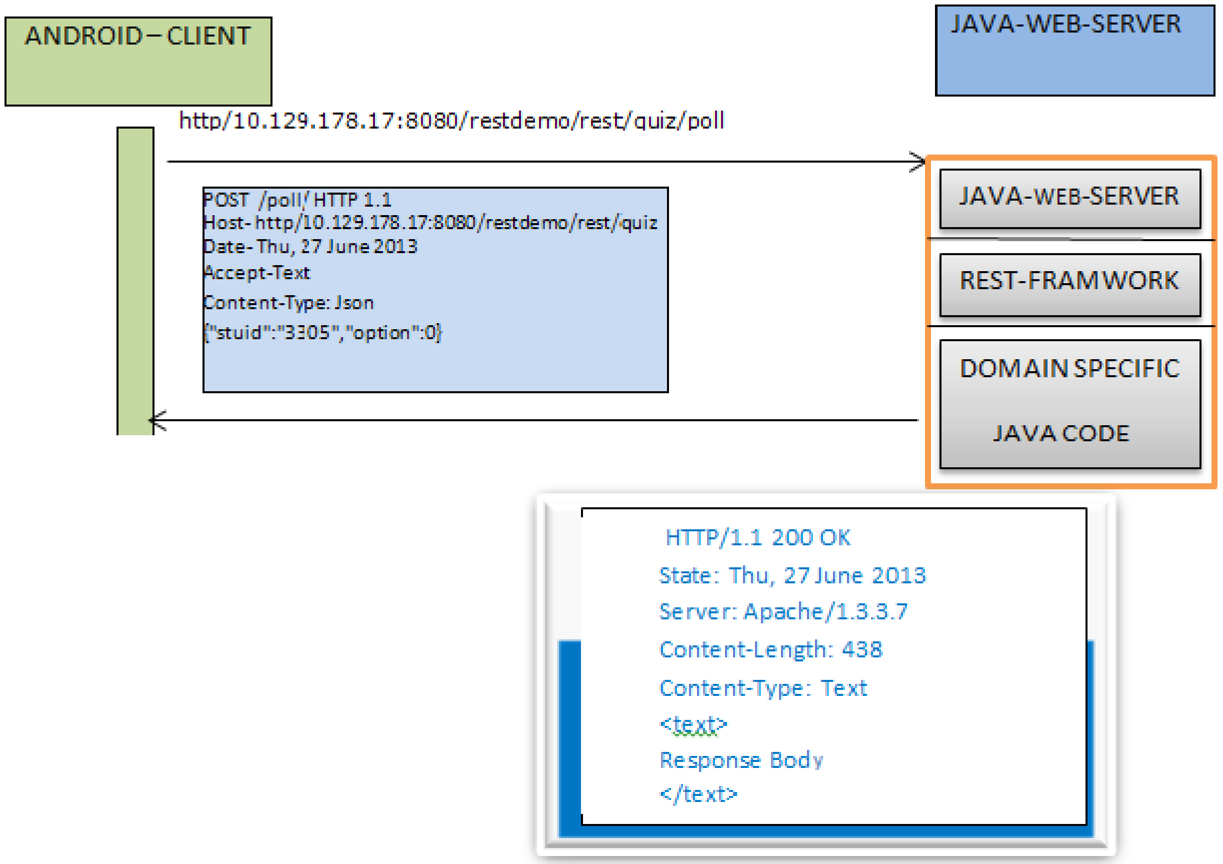


```
GET   /poll/{courseid} HTTP 1.1
Host- http/10.129.178.17:8080/restdemo/rest/quiz
Date- Thu, 27 June 2013
Accept-Json
```

```
HTTP/1.1 200 OK
State: Thu, 27 June 2013
Server: Apache/1.3.3.7
Content-Length: 438
Content-Type: Json
{"pollquestion":"Do u want class
Tomorrow?","launchtime":"2013/06/27 14:00:04"}
```

f) POST methods for Getting Json representation of POLL response of each individual student with respect to corresponding courseid:-

Consumes Json representation-
{"stuid":"3305","option":0}

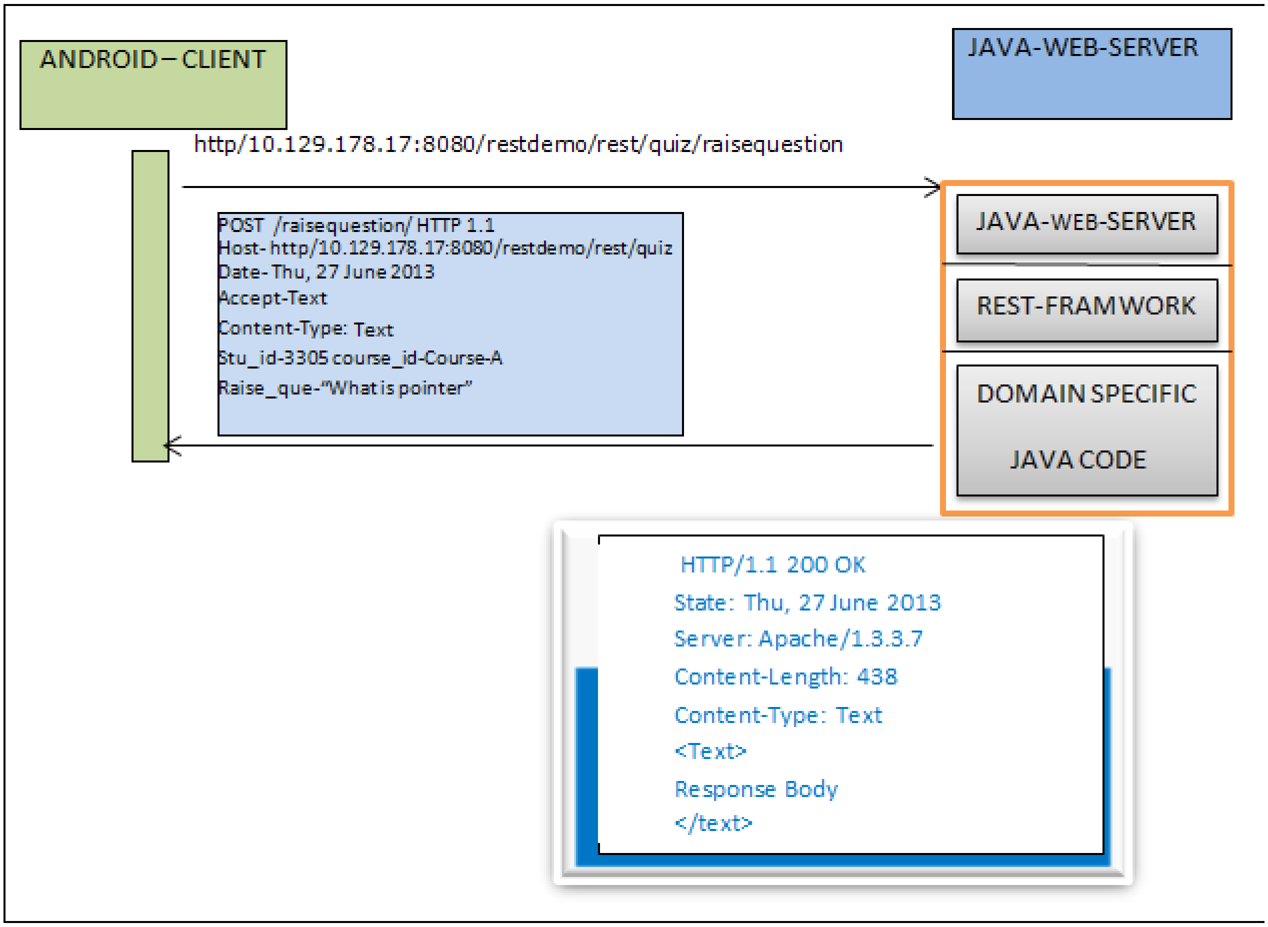


```
POST /poll/ HTTP 1.1
Host- http/10.129.178.17:8080/restdemo/rest/quiz
Date- Thu, 27 June 2013
Accept-Text
Content-Type: Json
{"stuid":"3305","option":0}
```

```
HTTP/1.1 200 OK
State: Thu, 27 June 2013
Server: Apache/1.3.3.7
Content-Length: 438
Content-Type: Text
<text>
Response Body
</text>
```

g) POST methods for Getting Text-representation of Raise question of a student with respect to corresponding courseid:-

Consume Text Representation-
Stu_id- **3305**
Course_id- **course-A**
Raise_qu-**"what is pointer?"**



```
POST  /raisequestion/ HTTP 1.1
Host- http/10.129.178.17:8080/restdemo/rest/quiz
Date- Thu, 27 June 2013
Accept-Text
Content-Type: Text
Stu_id-3305 course_id-Course-A
Raise_que-"What is pointer"
```

```
HTTP/1.1 200 OK
State: Thu, 27 June 2013
Server: Apache/1.3.3.7
Content-Length: 438
Content-Type: Text
<Text>
Response Body
</text>
```

3.2.2 HTML

It is the main markup language for displaying web pages and other information that can be displayed in a web browser

3.2.3 JavaScript

It is a prototype-based scripting language that is dynamic, weakly typed and has first-clas functions and is mainly used for validation etc.

3.2.4 JSP

JSP provide excellent server side scripting support for creating database driven web applications JSP enable the developers to directly insert java code into jsp file, this makes the development process very simple and its maintenance also becomes very easy.

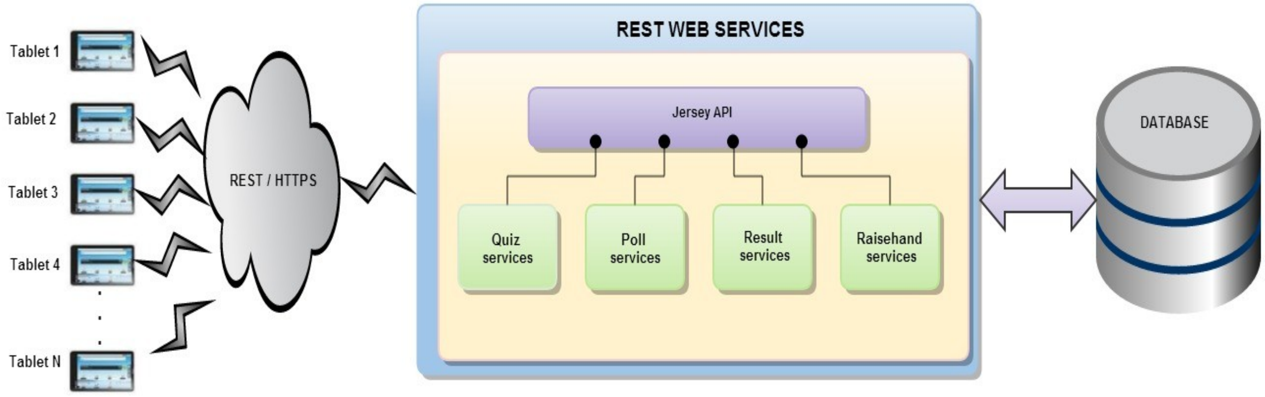
3.2.5 MYSQL

It is the world's most used open source relational database management that runs as a server providing multi-user access to a number of databases.

Json: json stands for **J**ava**S**cript **O**bject **N**otation. JSON is syntax for storing and exchanging text information between server and client,much like XML.JSON is smaller than XML, and faster and easier to parse.

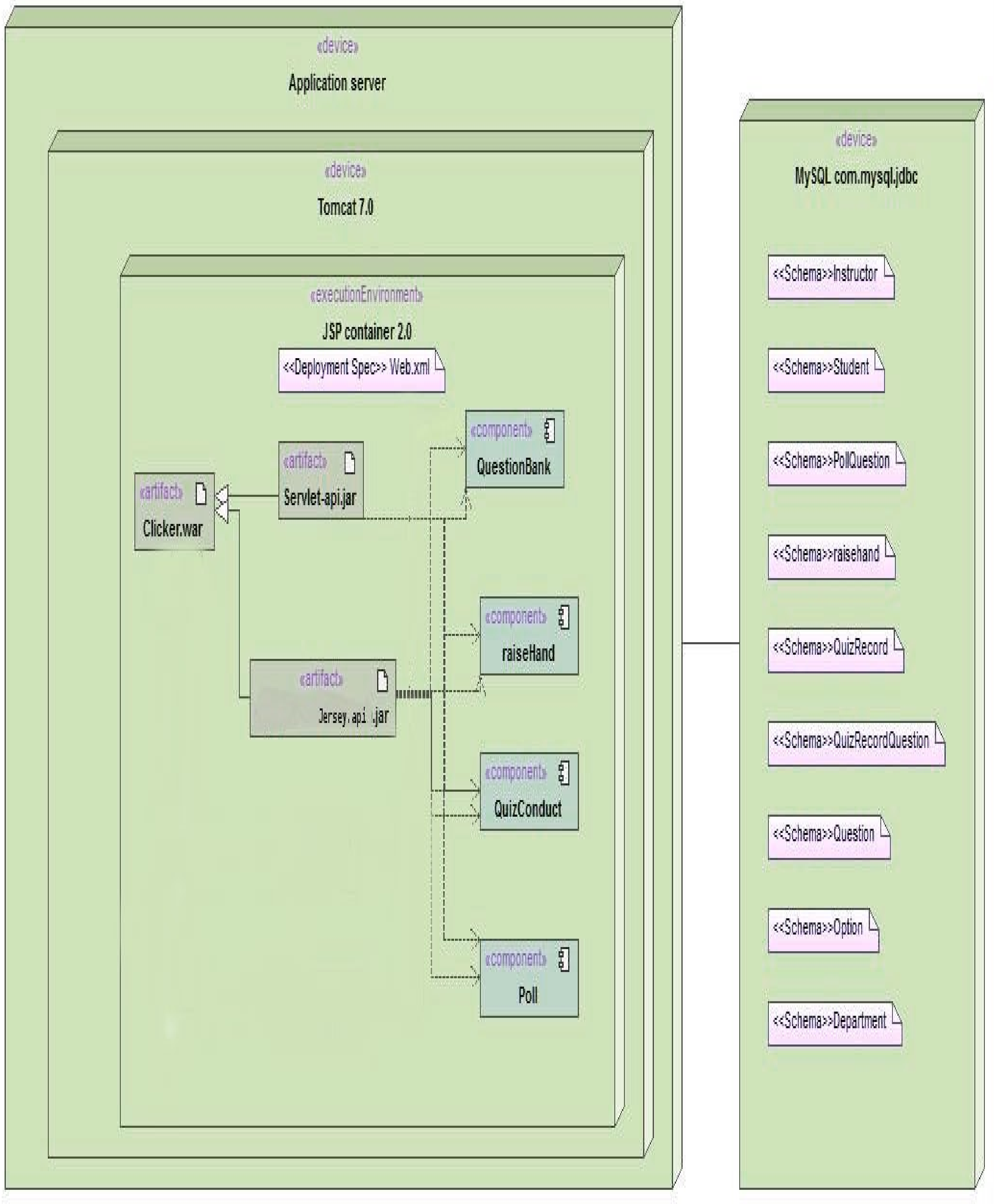
Chapter4

Architecture

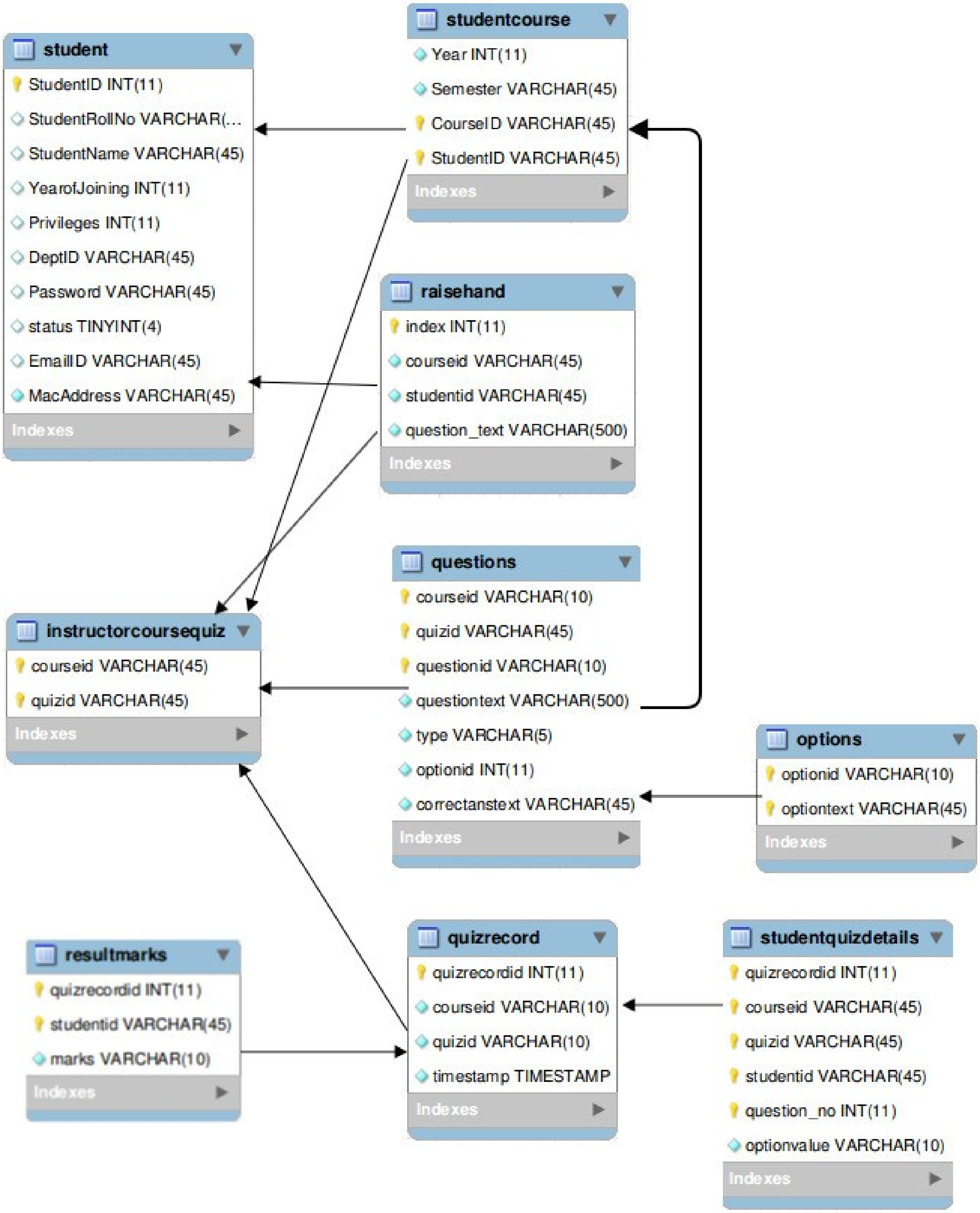


Chapter 5: Design And Implementation

5.1 Deployment Diagram

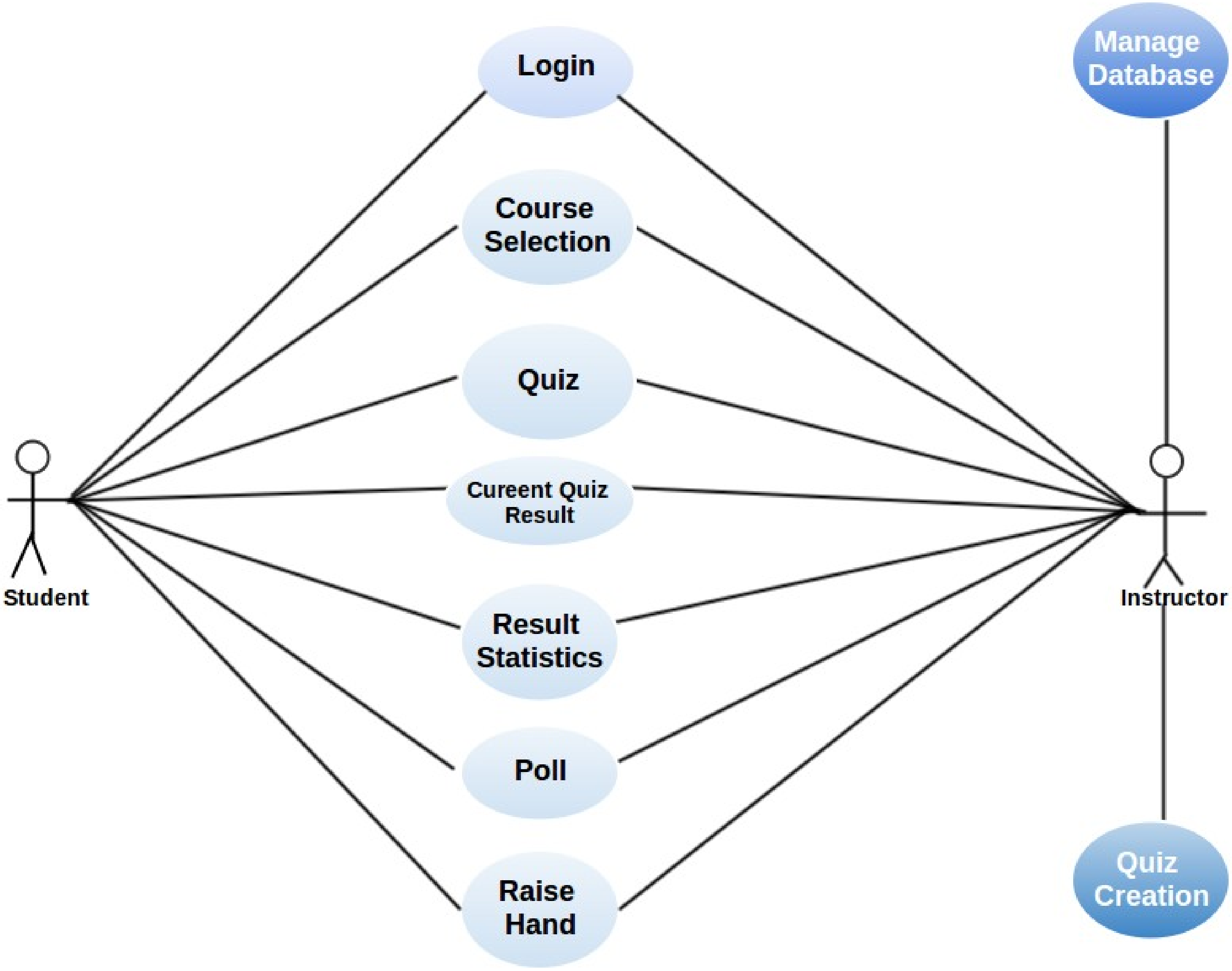


5.2 Schema Diagram

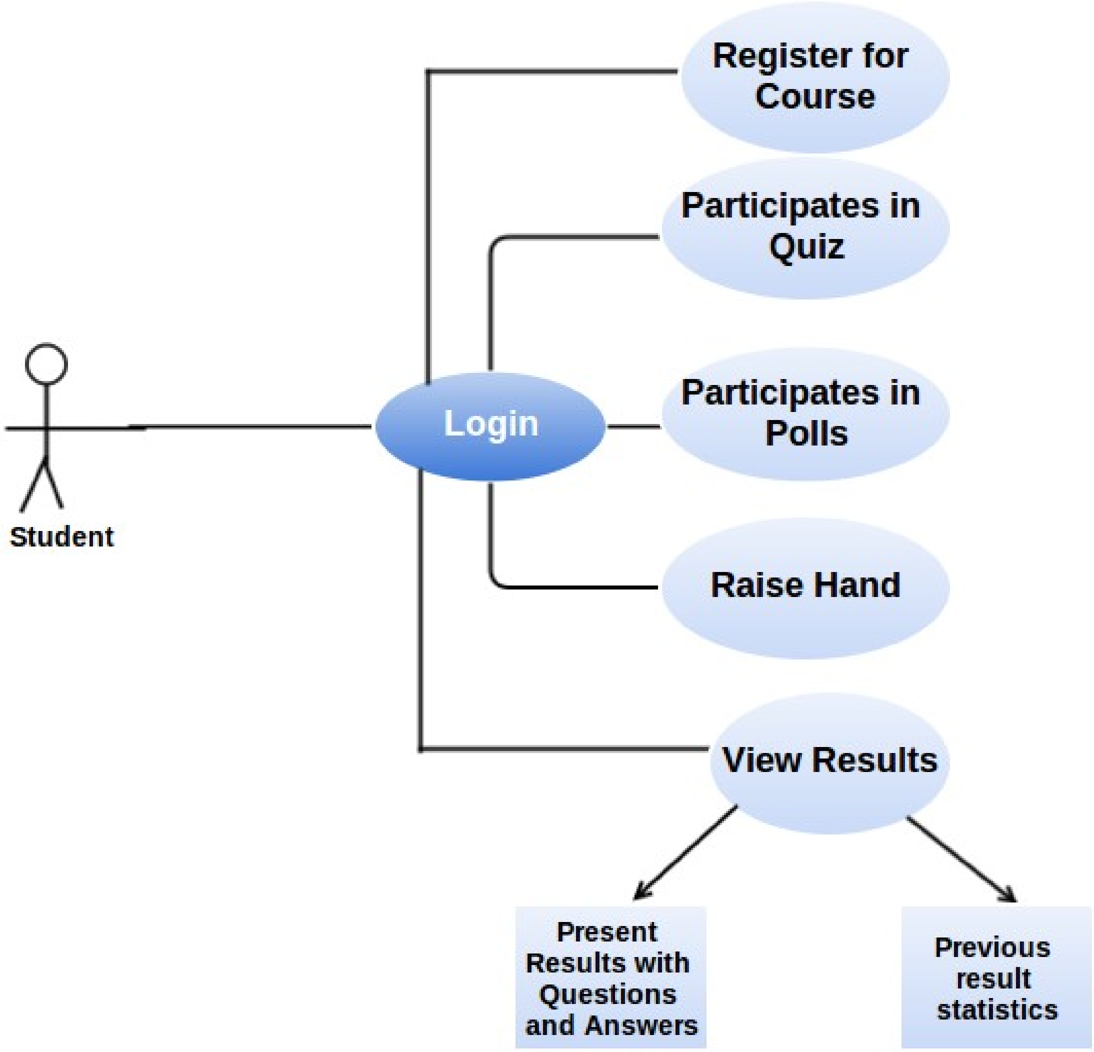


5.3 Use Case Diagrams

5.3.1 Overall

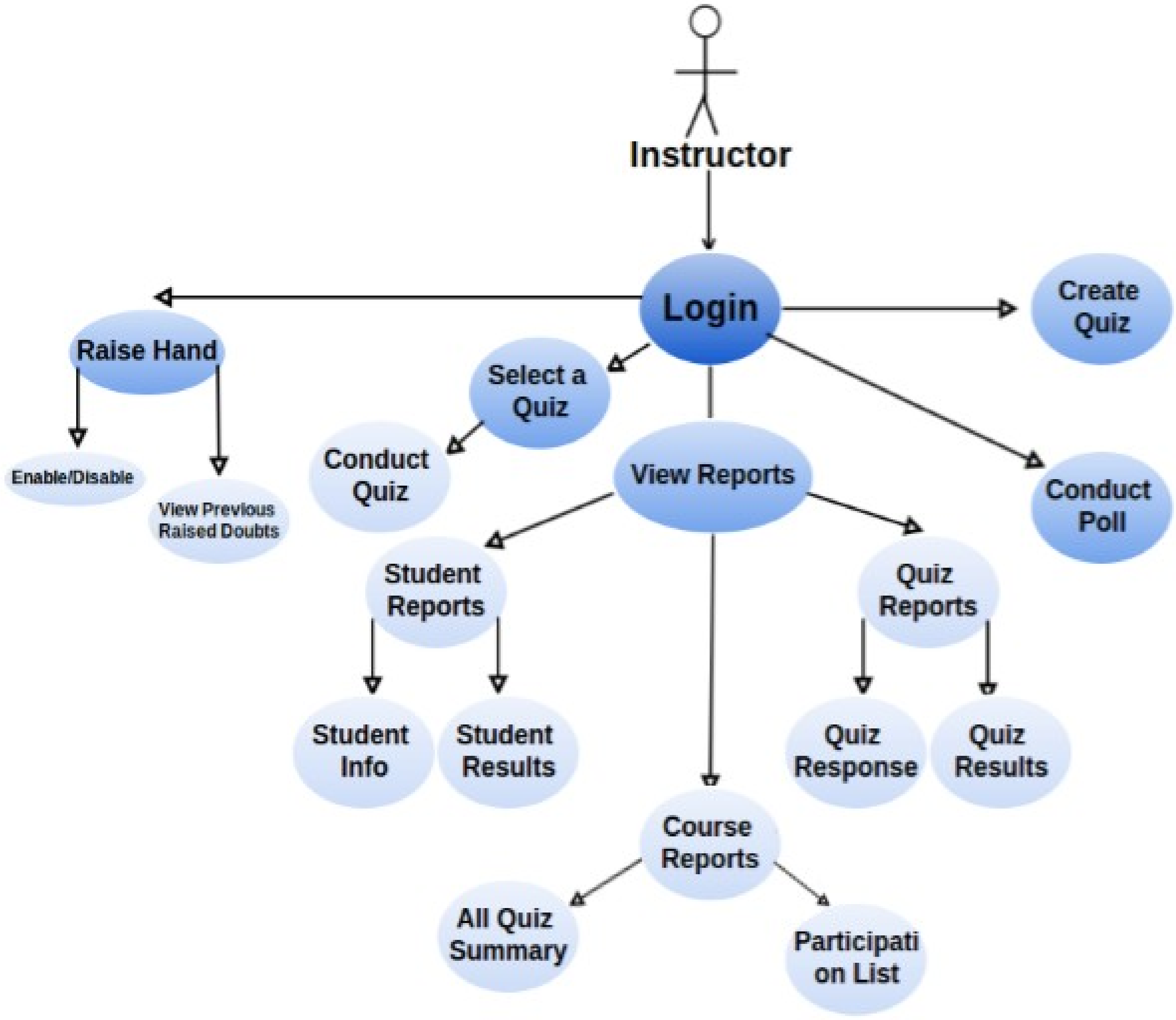


5.3.2 Student


Register for Course
Participates in Quiz
Login
Participates in Polls
Raise Hand
View Results
Student
Present Results with Questions and Answers
Previous result statistics

Use Case	**Descriptions**
Login	The Student must login with a valid studentid and server URL
SelectCourse	The Student hast to select the course for which he would like do quiz, poll and raisehand
Participate in Quiz	The Student can participate in the quiz launched by the Instructor
Raise Hand	The Student can raise a doubt throught raisehand module
View Reports	The Student can view reports related to his/her previous results
Poll	The Student can participate in poll by giving his opinion
View Results	The Student can see his marks and correct answers of the present quiz

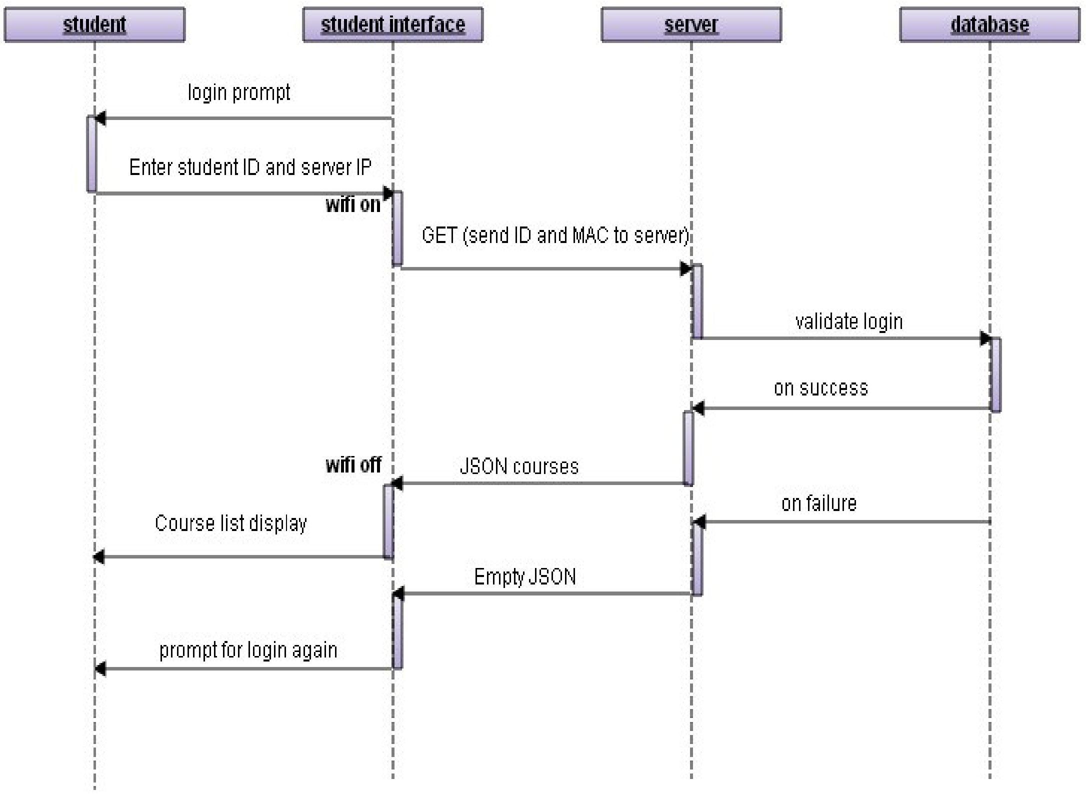
5.3.3 Instructor



Use Case	Descriptions
Login	Instructor need to make sure that server is started.
Courseselection	The Instructor can make the list of courses through database
Conduct Quiz	The Instructor can conduct a quiz selected among available quizzes
Create Quiz	The Instructor can create a quiz from the questions available in database
View Reports	The Instructor can view/download various reports pertaining to students, quizzes or courses
Student Reports	The Instructor can view reports related to Students
Student Info	The Instructor can view report containing information about a particular student
Student Result	The Instructor can view report containing the results of a particular student
Course Reports	The Instructor can view reports related to his course
AllQuiz Summary	The Instructor can view the summary of all the quizzes for his course in a single report
Participant List	The Instructor can view all the participants enrolled in his course through database
Quiz Reports	The Instructor can generate reports related to the existing quizzes for his course
Quiz Details	The Instructor can view the details about all the questions in one particular quiz
Quiz Response	The Instructor can view the responses of students for an already conducted quiz
Quiz Results	The Instructor can view the results of students for an already conducted quiz
Conduct Poll	The Instructor can conduct a poll in the form of a yes/no question and view the response of students for that poll
Raise Hand	The Instructor can view/save doubts raised by students while raise hand module is enabled
Enable/Disable Raise Hand	The Instructor can enable or disable the raise hand module
View Previously Raised Doubts	The Instructor can view the doubts raised by students at a later time
Reply to doubts	The Instructor can reply to the saved doubts physicall or through mail at a later time

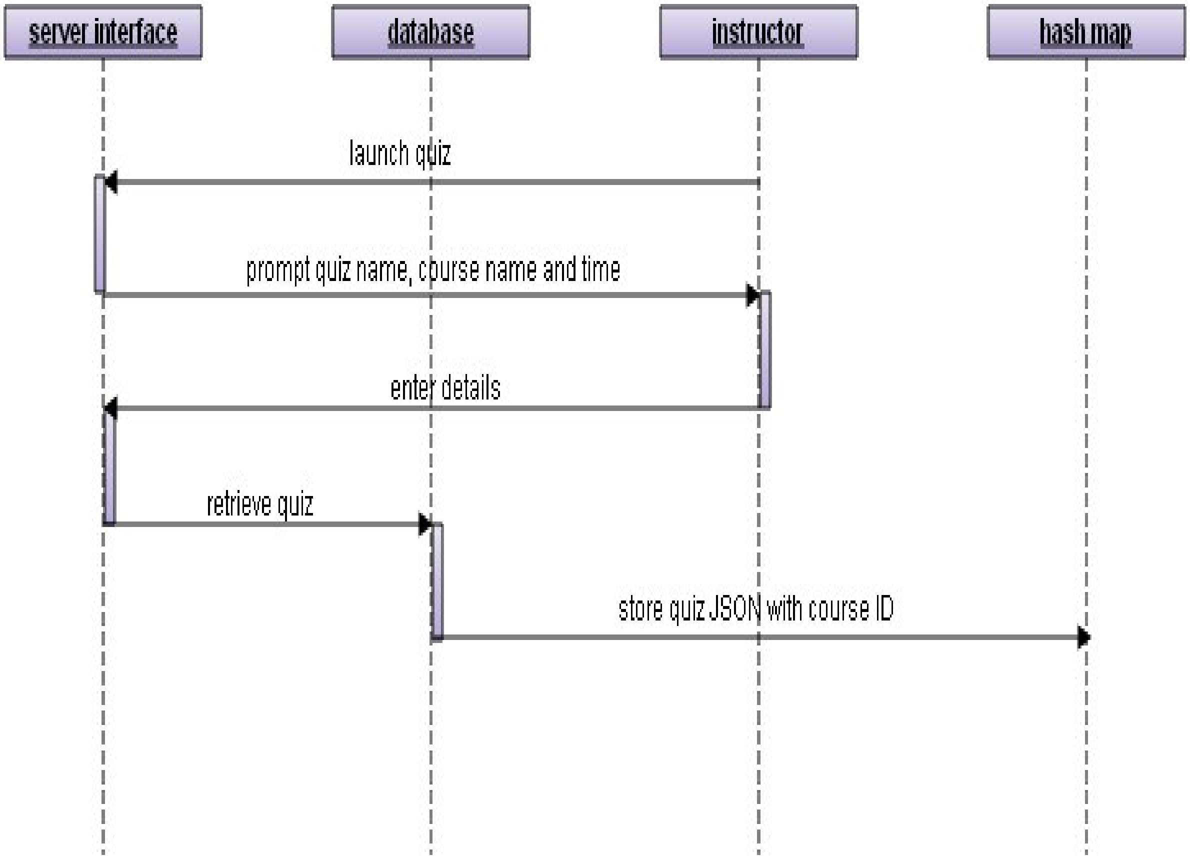
5.4 Sequence Diagram

5.4.1 Login



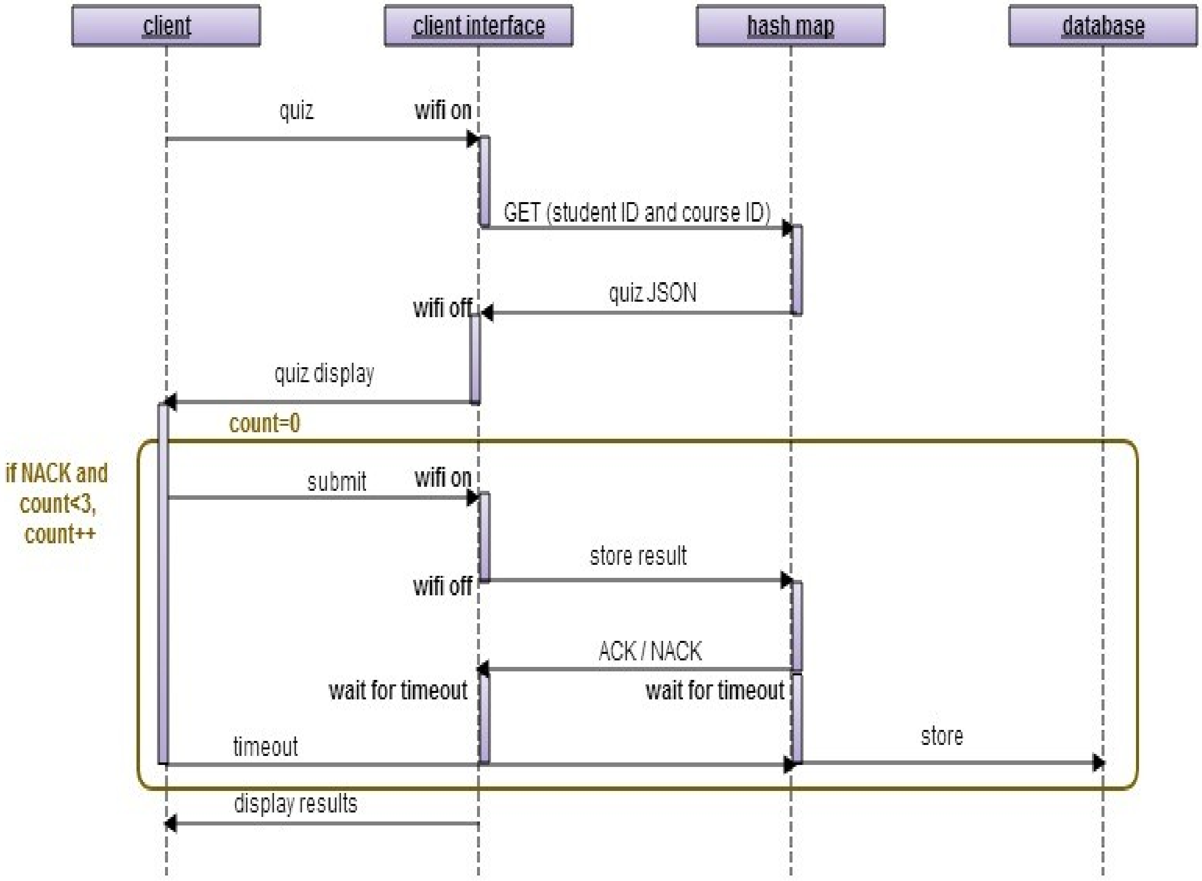
Sequence Diagram of login

5.4.2 Quiz launch



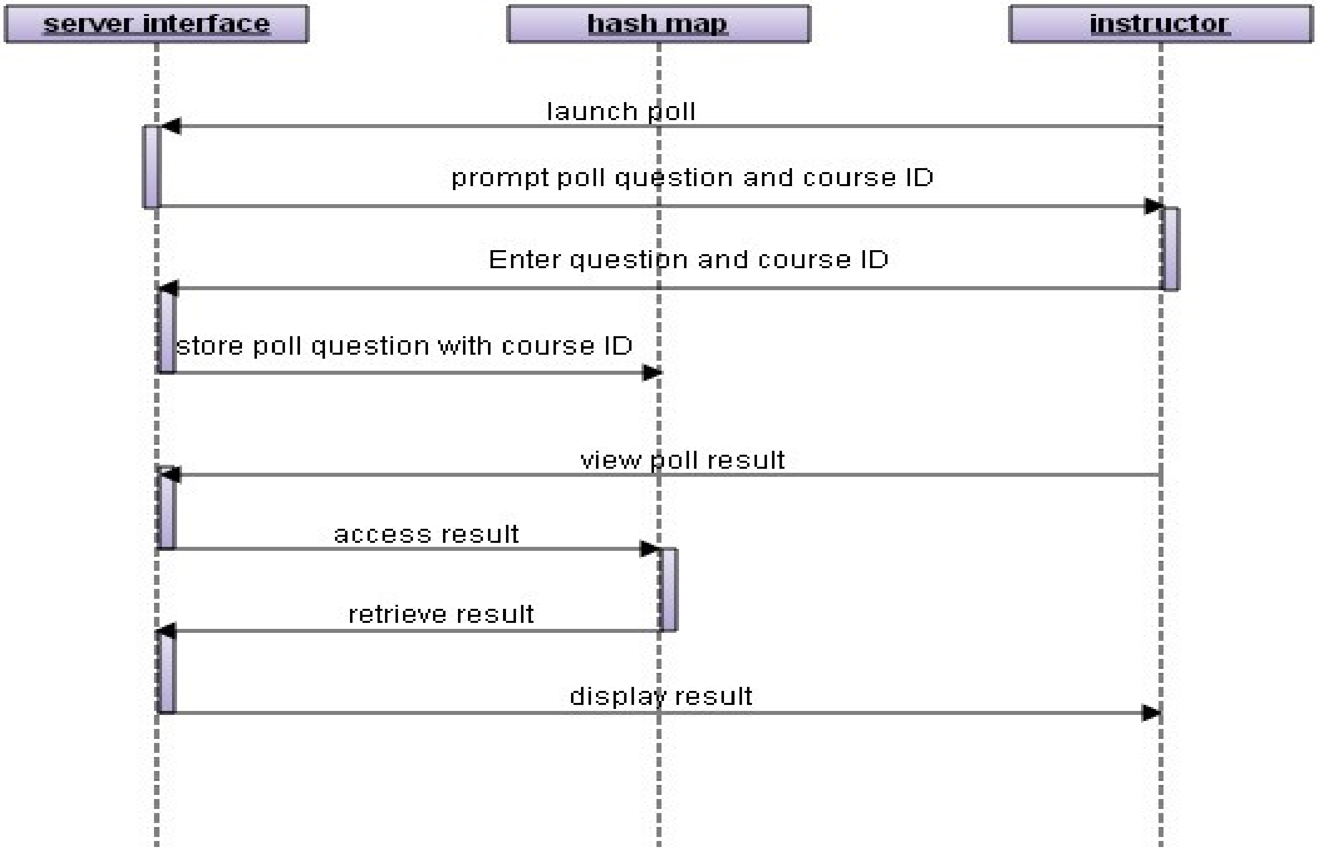
Sequence diagram of quiz launch

5.4.3 Quiz Access



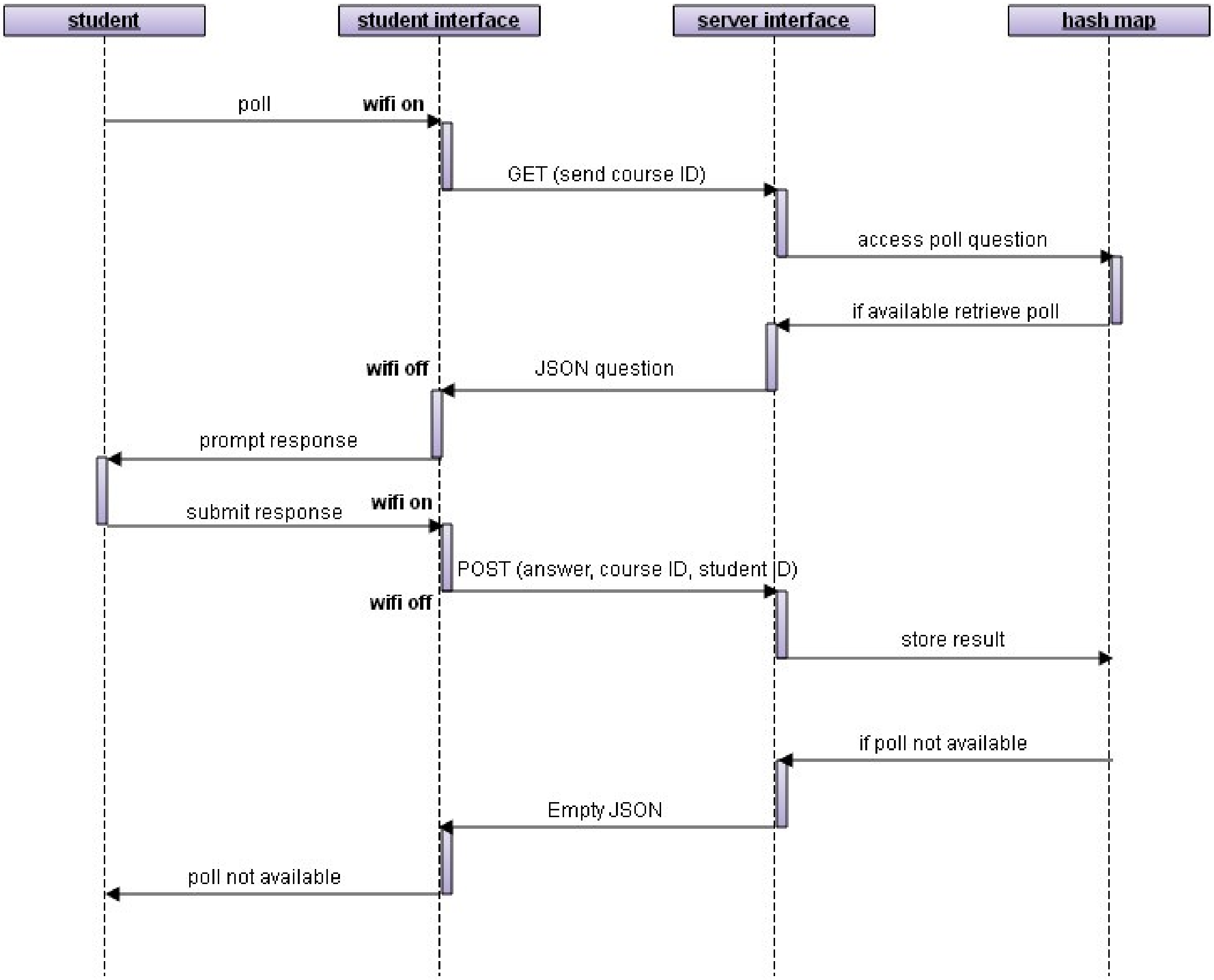
Sequence Diagram of Quiz access

5.4.4 Poll Launch



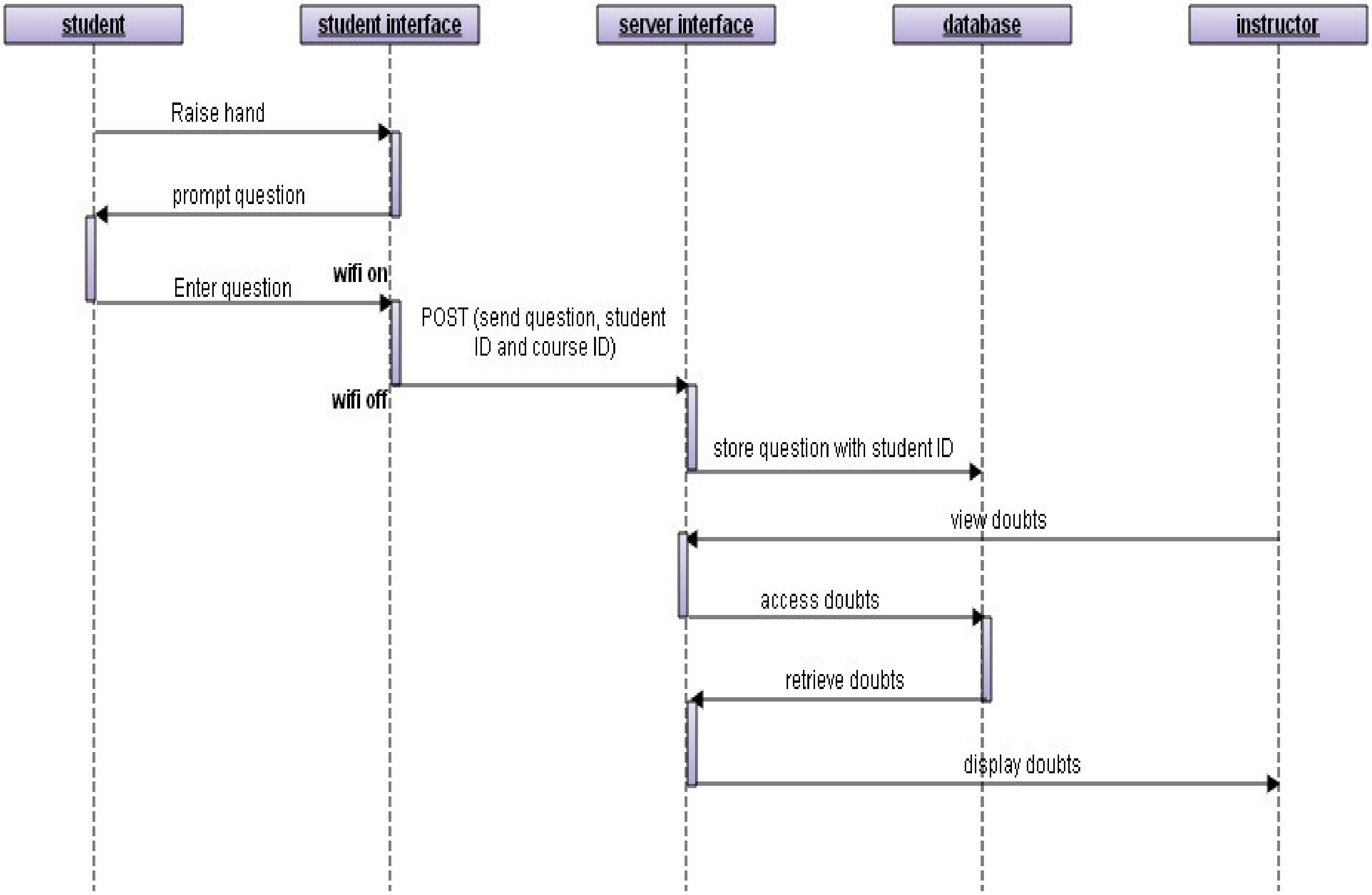
Sequence Diagram of Poll Launch

5.4.5 Poll Access



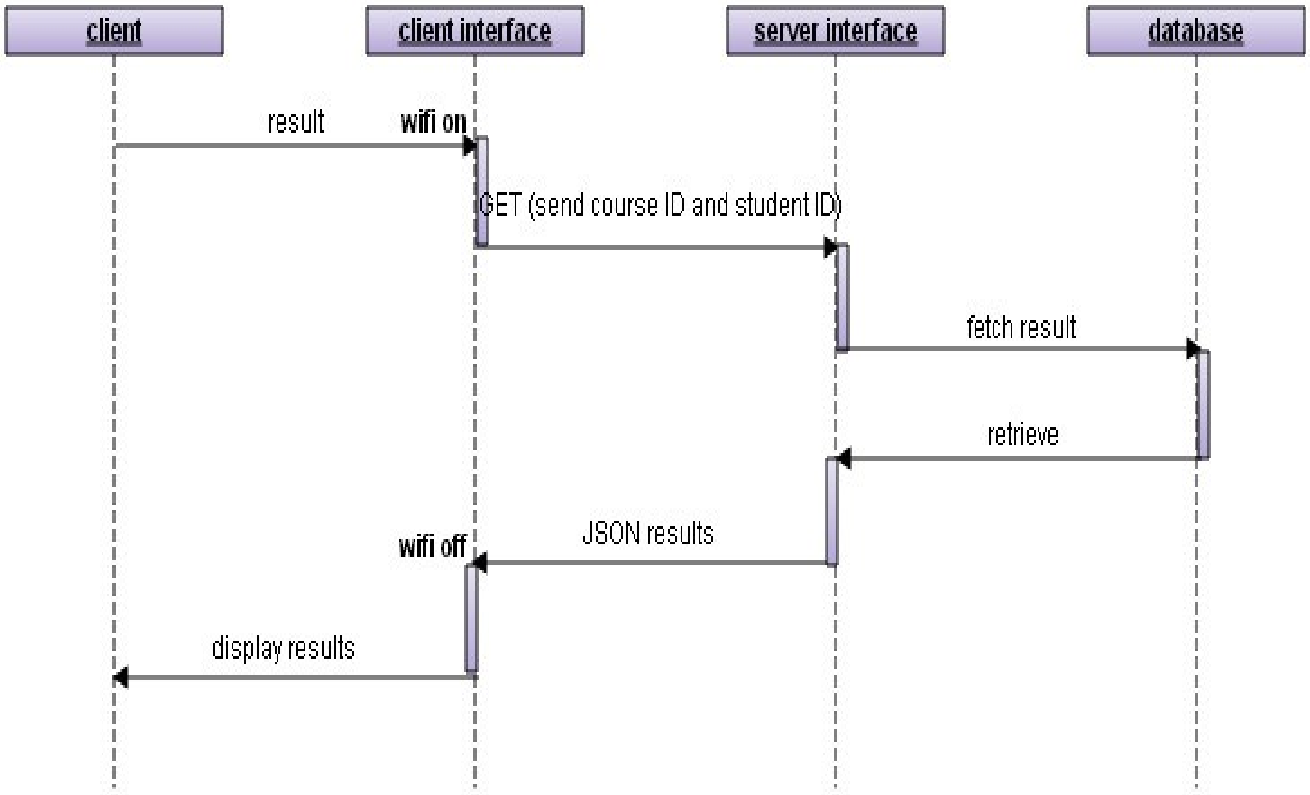
Sequence diagram of Poll Access

5.4.6 Raise Hand



Sequence Diagram of Raise Hand

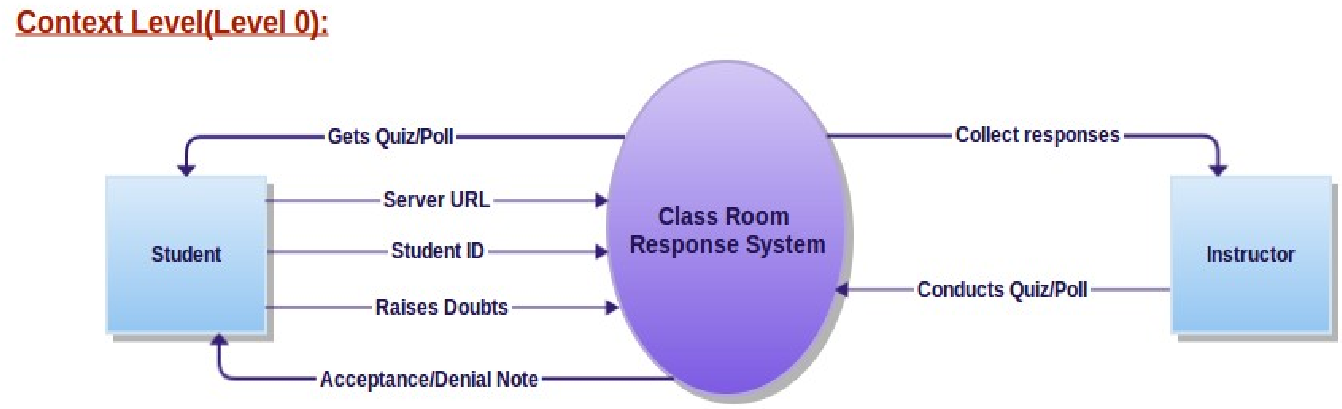
5.4.7 Result



Sequence Diagram of Result

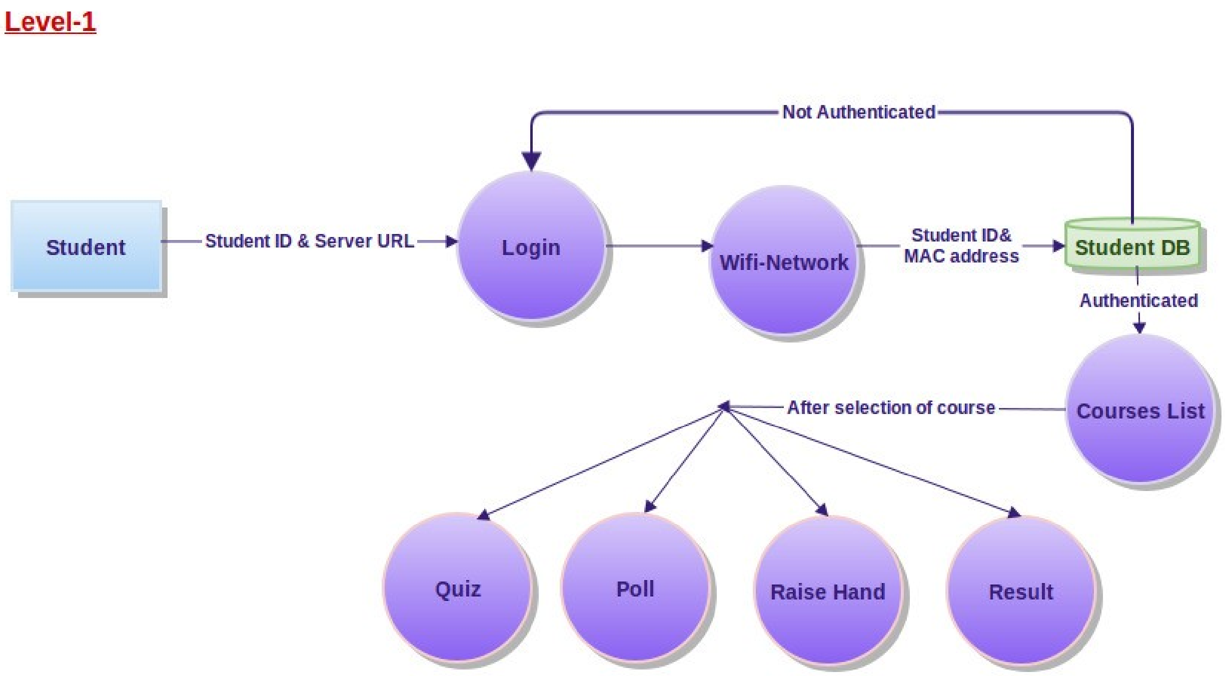
5.5 Data Flow Diagrams

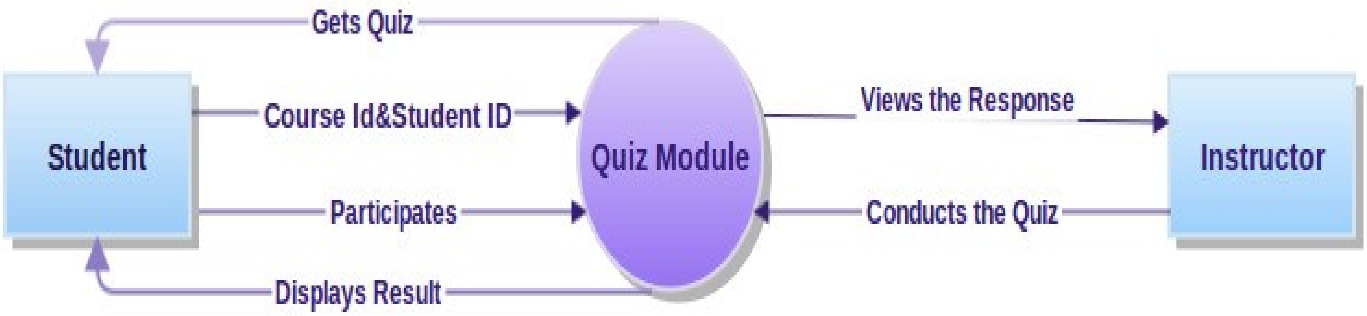
5.5.1 Level 0

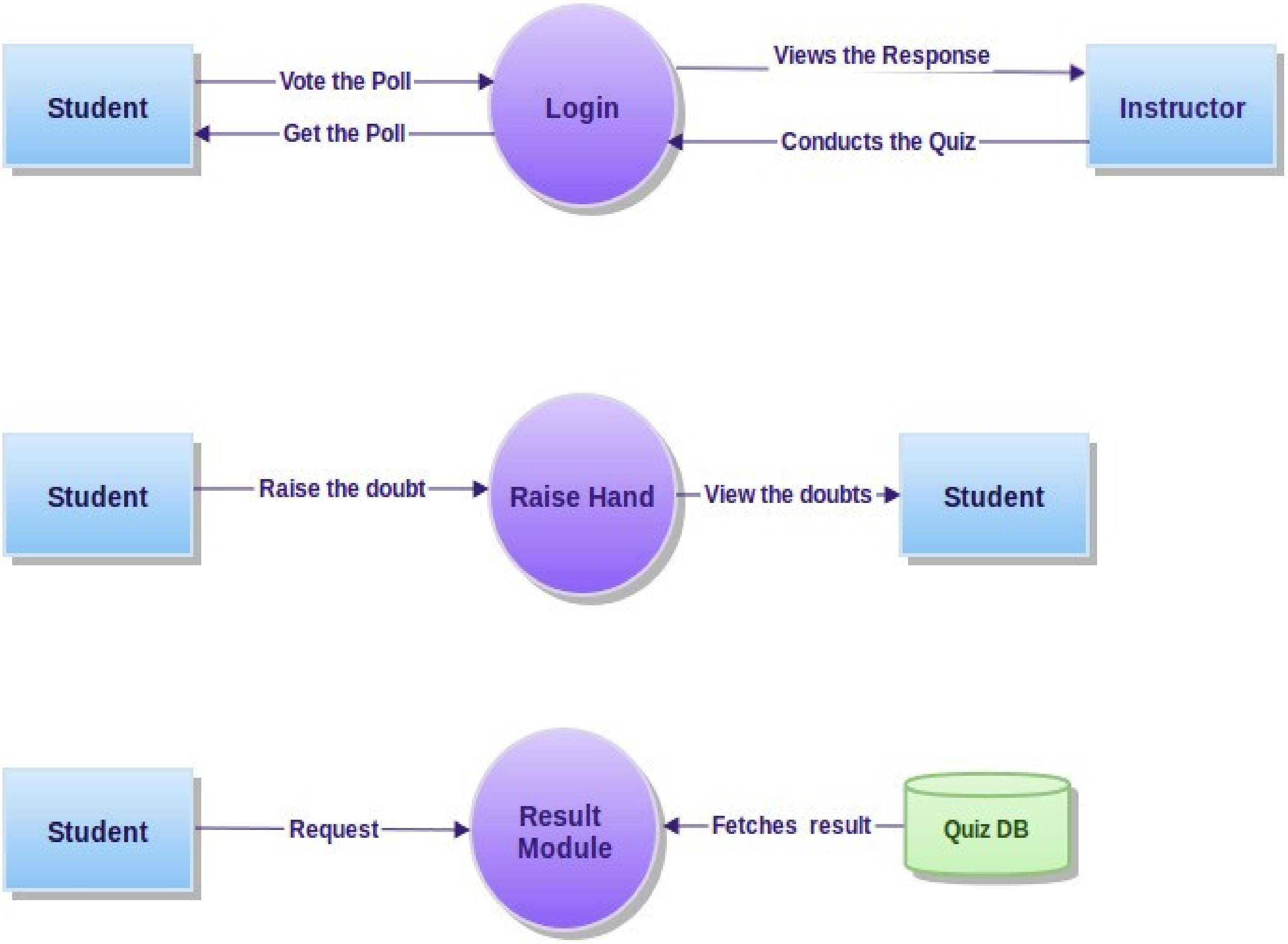


A simple representation of the whole system. It gives the brief idea of how the data is being processed between Student and Instructor through Class Room Response System. The processing is indicated by arrow marks with process name on it.

5.5.2 Level 1

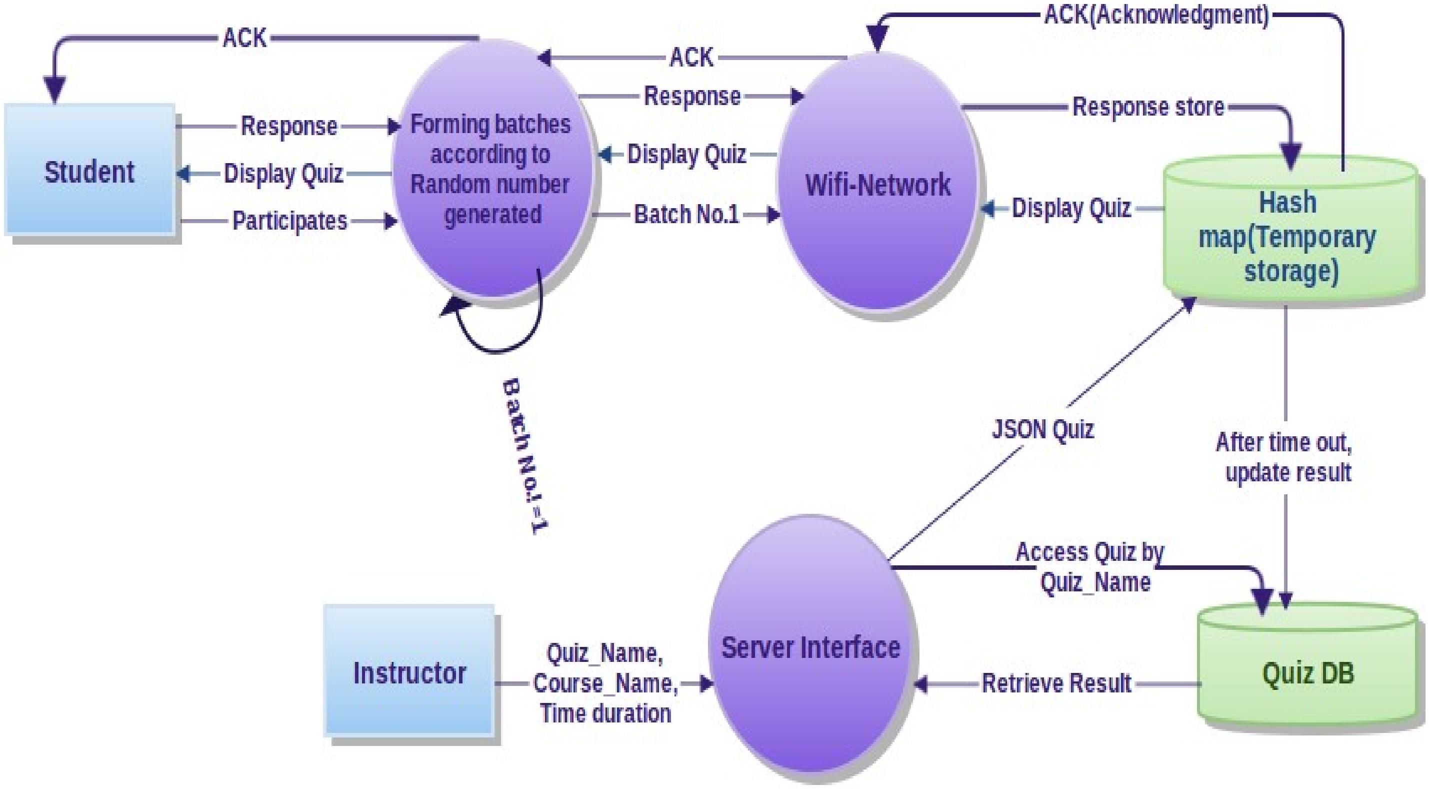

Level-1
Not Authenticated
Student
Student ID & Server URL
Login
Wifi-Network
Student ID&
MAC address
Student DB
Authenticated
Courses List
After selection of course
Quiz
Poll
Raise Hand
Result


Gets Quiz
Student
Course Id&Student ID
Participates
Displays Result
Quiz Module
Views the Response
Conducts the Quiz
Instructor


Student
Vote the Poll
Get the Poll
Login
Views the Response
Conducts the Quiz
Instructor
Student
Raise the doubt
Raise Hand
View the doubts
Student
Student
Request
Result Module
Fetches result
Quiz DB

5.5.3 Level 2

Level-2:(Quiz)



Chapter 6

Detailed Description

This time Clicker Version 4 has been implemented on android platform Exclusively for Aakash Tablet. Now we are describing all basic modules of Clicker V4 with more detail in following paragraph.

To use this application student has to be enrolled in any of the course. When first time student starts his application he will put his STUDENT_ID and URL to Login. Then along with studnetID ,mac address of tablet would be sent to server and will be saved in database.

Now whenever student wants to Login once again he can simply put his student Id and url to login.Next page the list of Course for which student has been registered will come and accordingly he can give Quiz,Polling,Raise Hand kind of functionality.

Before starting application make sure that your tablet had been connected to same Wifi which you are going to use while Login and Switch off Wifi.

6.1 LOGIN:-

Student is allowed to enter his student id and url in the Text fields then press connect button. After pressing Connect button You can observe that Wifi is being switching ON and getting connected to latest connected/used Wifi Automatically.

After student connects to server and get authenticated Client will receive one json object having list of courses available. For particular student, Wifi will be automatically get switched off after that and no more Connection to server is needed. If the verification fails then the user is informed of the wrong Student ID and Server URL.

6.2 SelecCourse:-

Successful login will jump to next activity called SelectCourse with listing all the available courses. Student are allowed to select any one of the course and then jump to next activity having all main functionality which is present in Home,Wifi is not required to be enable meanwhile moving from SelectCourse Activity to Home Page.

6.3 Home:-

This Activity is very important having main four functionalities which are Quiz,Poll,Raise Hand,Results.

Each of these functionality except raise hand will need Wifi enabled and data from server in json formate. Raise hand will need wifi enabled only at the time of submitting doubt.

6.4 Quiz:-

In order to conduct the quiz, the instructor needs to select the quiz name from the dropdown list which contains all the names of the quiz associated with that course.

Once he/she selects the quiz then he needs to set the time for the quiz and then launch the quiz URL

After he launched the quiz a page with countdowntimer will be started, within that time whatever may be the number of responses that will be stored in database.

As soon as Quiz has been launched by Instructor student will be able to take up quiz and submit. Quiz button would be disabled when once it has been clicked by student.

Now Quiz has been launched student try to access Quiz Wifi will switch ON and fetch Question in json formate. All the question will be random in order when it is being displayed in tablet along with different order of options. So No one will be able to cheat during test.

After downloading quiz wifi is no more needed and will get switched off. Then Only at the time of manually submition or submittion after timer finishes will lead to Switching ON of Wifi.

Meanwhile Quiz is going on even if Tablet's orientation is changed will not affect application and back button has been disabled So you can not even go to back. Only When response is submitted and result will be displayed to the client.

6.5 Poll:-

This class again will need to talk to the server So previously mentioned functions will be called. This method will work only if Poll has been conducted by instructor side.

The instructor enters the poll question and presses the Conduct Poll button. The students see the poll question and are allowed to give the response within the specified time limit. No student is allowed to give response more than once. The instructor can see the responses given by the student and the results are also stored in the database for the future reference.

6.6 Raise Hand:-

Firstly the instructor needs to enable the raise hand so that the student can post his/her query. Whenever any student posts any query the colour of the side bar of the instructor window changes. The instructor can see the doubts and has the option of saving or deleting the doubts. The instructor can later on see the saved doubts and can respond to the student by mail.

Raise Hand class mainly focuses on asking doubt by student to instructor through writing question and send it to instructor.

6.7 Result:-

This class will allow student to see their previously attempted quiz and their results. Here also communication with server has to be done along with automatic switching OFF and ON of Wifi.

Chapter 7

Technical Details

7.1 Instructor Module

FILE	LOC	Description
Courseidinstructor.jsp	17	Instructor will enter Course ID, for which he is going quiz, poll and raisehand
Home.jsp	22	Instructor need to choose whether he is conducting quiz, poll or raisedquestions
Instructorquiz.jsp	60	Gives dropdown list of quizes available, time for the quiz can be given here
Showquiz.jsp	85	It will display the countdowntimer, course and quiz details of the selected quiz to the Instructor.
Responsestore.jsp	122	It will display the student response about the quiz like how many of them are requested for quiz, downloaded the quiz, submitted the quiz. It will give statistics of each question ,with how many number of students selected option:A, B, C, etc.,
Poll.jsp	23	Instructor will enter the poll question and launch the poll.
Showpollquestion.jsp	59	Display the poll question and countdown timer for the polled question.
Pollresponseshow.jsp	52	This will give information about number of students participated in poll, and give option details.
Showpastraise question.jsp	52	It will display all past raised questions of students in that particular course
Calljson.java	79	Take the question details for a particular quiz from question table of database. And stores into concurrenthashmap.
Global.java	13	Different concurrenthashmap for storing quiz details, poll questions details, and both student poll and quiz response

		details.
Quiz.java	95	It is a pojo class having GET and SET methods for finding quiz details like courseid, launchtime, quizid, quizduration etc.,
Question.java	84	It is a pojo class having various GET and SET methods for describing question details as id, text, correct answer and type .
Option.java	29	It is a pojo class having GET and SET methods for opetion text.
Course.java	28	It is a Pojo class having GET and SET methods for giving information of number of students registered for a particular course.
Quizresult.java	44	It is a Pojo class having GET and SET methods for giving past quiz results of students with including marks obtained, timestamp of the quiz, and out of marks of the quiz.
Resulmarks.java	38	It is a Pojo class having GET and SET methods , used for giving information about students for queries of results.
Coursedatabase.java	118	It is a database page. It gives course list of the student for which student has registered. And also gives total number of students registered for that course
Instructorcountstudent.java	43	It is a database page. It wil give total number of students registered for a particular course.
Savequizrecord.java	42	It is a databse page. Used for storing timestamp, courseid, quizid of conducted quiz.
Getnoofquestionoption.java	60	It is a database page. Used for fetching quiz details like total number of questions with number of options of the conducted quiz
Databasestoreresponse.java	286	It is a database page. Used for storing student response about quiz and collects the marks details.
Marksdatabase.java	81	It is a database page. Used for retrieving past quiz details which includes marks, student details.
Raisehandstoredatabase.java	44	It is a database page. It is for storing raised question of students in databse.
Getresponsefromhash.java	55	This is for storing response of polled question in concurrenthashmap.
Restquizresource.java	247	This is RESTWEB Resource page having all the GET and POST methods that act as interface between android client and server.

Total Number of Pages: 26
Total Lines of Code: 1826

Description of classes and methods that are used at server side:
Database package contains classes and methods as below:

1. Coursedatabase class

   data(String studentid, String mac): This method is used to authenticate student login.

2.Databasestoreresponse class

   2.1 saveQuizresponse(String courseid, int a[][], int b[]): This method is used to save all students quiz responses

   2.2 insertIndivifualStudentRecord(Connection con, String courseid, int a[][], String studentjsonresponse): This method is used to store individual student record.

   2.3 insertQuizRecordQuestionData(Connection con, String courseid, String quizid, String current_sid, int j, String response, String LQRI) : This method is used to store latest quiz records.

3. Getnoofquestionoption class

   getno(String courseid, String quizid): This method is used to store courseid and quiz id in database.

4. Instructorstudentcount class

   studentcount(String courseid): This method is used to store courseid for registered students in database.

5. Marksdatabase class

   getmarks(String courseid, String studentid): This method is used to get marks of corresponding studentid of that courseid.

6. Raisehandstoredatabase class

   saveraisequestiontext(String courseid, String studentid, String raise_qu): This method is used to store raised questions by students.

7. Savequizrecord class

   savelatestquizrecord(String courseid, String quizid, String timestamp): This method is used to store latest quiz details into database.

Javatojson package contains classes:

1. calltojson class:

   calljson(String courseid, String launchtime, String currenttime, String quizid, String minutes, String seconds): This method is used to store quiz json by instructor into hashmap.

2. callpolljson class

   callpolljson(String courseid, String pollquestion, String launchtime): This method is used to store pollquestion into hashmap by instructor.

3. course, option, question, quiz, pollque, quizresult, resultmarks: All these classes are "POJO" classes.

4. Global POJO class:

   It contains all the hashmap variables.

7.2 Student Module

FILE	LOC	Description
Login.java	339	It will connect to the server on clicking on connect button. After connect if we enter valid information then this allow us to move to course selection page.
Selectcourse.java	123	It will connect to the server to retrieve the number of courses from course table of database.
Home.java	518	It will connect to the server to get quiz, poll, result and to raise doubts.
Quiz.java	1799	It will connect to the server to get quiz that is available with that particular course.
Poll.java	399	It will conncet to the server to get poll question that is posed by the Instructor.
Help.java	101	It will give an introduction about each and every module of the application.
About.java	47	It gives description about the Clicker V4.
RaiseHand.java	273	It will connect to the server to post the questions raised by students to intimate the instructor.

Result.java	250	It will automatically connect to the server to get the result of the present quiz after completion of the quiz. The result is instantaneous.
Resultstats.java	393	It will connect to the server to get the all possible result of quizes that student has appeared for.
Logout.java	11	It is a page which will get away us from the application.

Total number of classes: 11
Total lines of code: 4253

Total lines of code in the entire project: 6079

Description of main classes and methods that are used in this application:

Some Classes and Methods are common for most Activities. Those methods we are going to describe below:-

These methods and classes are used whenever communication with server is required.

1.*WebServiceTask:* Runs a Web method and stores the result in a package variable or a file.This class can not be inherited.

2.*onPreExecute():*Can be called when a task is about to execute. This event is raised by a task or a container immediately before it runs.

3.*doInBackground()*: works on a background thread.This method can call publicProgress(Progress...) to publish updates on the UI thread.

4.*onPostExecute()*:Runs on the UI thread after doInBackground(Params...). The specified result is the value returned by doInBackground(Params...).

This method won't be invoked if the task was cancelled.

5.*getHttpParams():* Acess all entries of HttpParameters

6.*doResponse()* : Hold the XML response from Web Service

7.*inputStreamToString()* : Reads all contents of InputStream to a String.

8.*handleResponse()* : This method processes the response message.

Login Class:

1.*Connect()*:-This method is being called when Connect button has been called. This will intent list of courses in next activity called is SelectCourse

SelecCourse Class:

1.*onClick()*:- This method would be called when particular course will be selected and then intent to next activity named Home along with student name, selected Course. It do not need Communication with the server.

Home class:

1.*callserver():-* This function is used for switching ON wifi and obtain IP address of previously connected wifi.

2.*retrieveSampleData()*:-This method is called when Quiz button is pressed internally this methods will call CallSever() method to switch ON wifi and Obtain IP address and fetch Quiz and display.

3.*logout()*:- Just to Logout from application

4.*raise()*:-This method will be called to write some doubt and send to instructor

5.*help()*:- Help will give brief detail of all functionality in paragraphs.

6.*resultstats()*:-Previously given test can be viewed and need communication with server.

7.*poll()*:-Here instructor can put some Question for which he/she wants opinion of the class.

Quiz Class:

1.*shuffleArray()*:-To Shuffle the array having 0.....Last Question-1

2.*swap()*:-To swap the two values but here it is being used during randomization.

3.*onBackPressed()*:-This is inbuilt system function of android which invoke when backbutton is pressed.

4.*visibility_groups()*:-total no of question we are dividing into some groups having 5 questions in each.

5.*hide_top()*:-This will take care of button with question being displayed at top for each group.

6.*show_all()*:-This will make visible all top buttons visible.

7.*prev()*:-This button will decrease the counter variable by -1 and would allow switch to previous question.

8.*next()*:-This button will increase the counter variable by +1 and would allow switch to next question.

9.*onRadioButtonClicked()*:-This method will be invoked whenever any of the radio button is clicked.

10.*onCheckClicked()*:-This method will be invoked whenever any of the checked button is clicked.

11.*make_all_visible()*:-this will make all options visible before a new type of question is switched.

12.*Text_ok()*:-This method will make sure that fill in the blank question has been answered and set the Textfield.

13.*print_result()*:-This function will print the result and will call JsonResponse function.

14.*jsonResponse()*:-This function will generate response in json formate and try to send in response formate to server.

15.*display_Quiz()*:-This method will properly display the Question.

16.*make()*:-this will make all top buttons of same color Except first one. Whenever you want to switch different group will make first question Color as highlighted to make sure that student will come to know he is in currently first question of that group.

17.*click1()*:-This is for group1 of questions which are on bottom.

18.*click2()*:-This is for group2 of questions which are on bottom.

19.*click3()*:-This is for group3 of questions which are on bottom.

20.*click4()*:-This is for group4 of questions which are on bottom.

21.*top()*:-This will be called whenever anyone of the top button is called. It will check which button has been clicked and based on that it will change the color of that particular button.

22.*CountDownTimer()*:- to show timer in client Side.

Poll:

1.*sendpoll()*:- To send response of student to instructor in json formate with student selected option.

2.*CountDownTimer()*:-The Counter will keep running till it get finishes.

Raise Hand:

1.*message()*:-This is the only separate method has been used and to be called when somebody wants to send doubt to instructor. This function will internally need communication with server So that it will use previously stated code in starting.

Chapter 8

Future Work and Conclusion

The basic aim of the project was to pave way for a user-friendly and an interactive session in the classroom for a better future. The task has been successfully completed by the team and all the above mentioned features and characteristics have been incorporated in the project. The future work regarding this project can be:

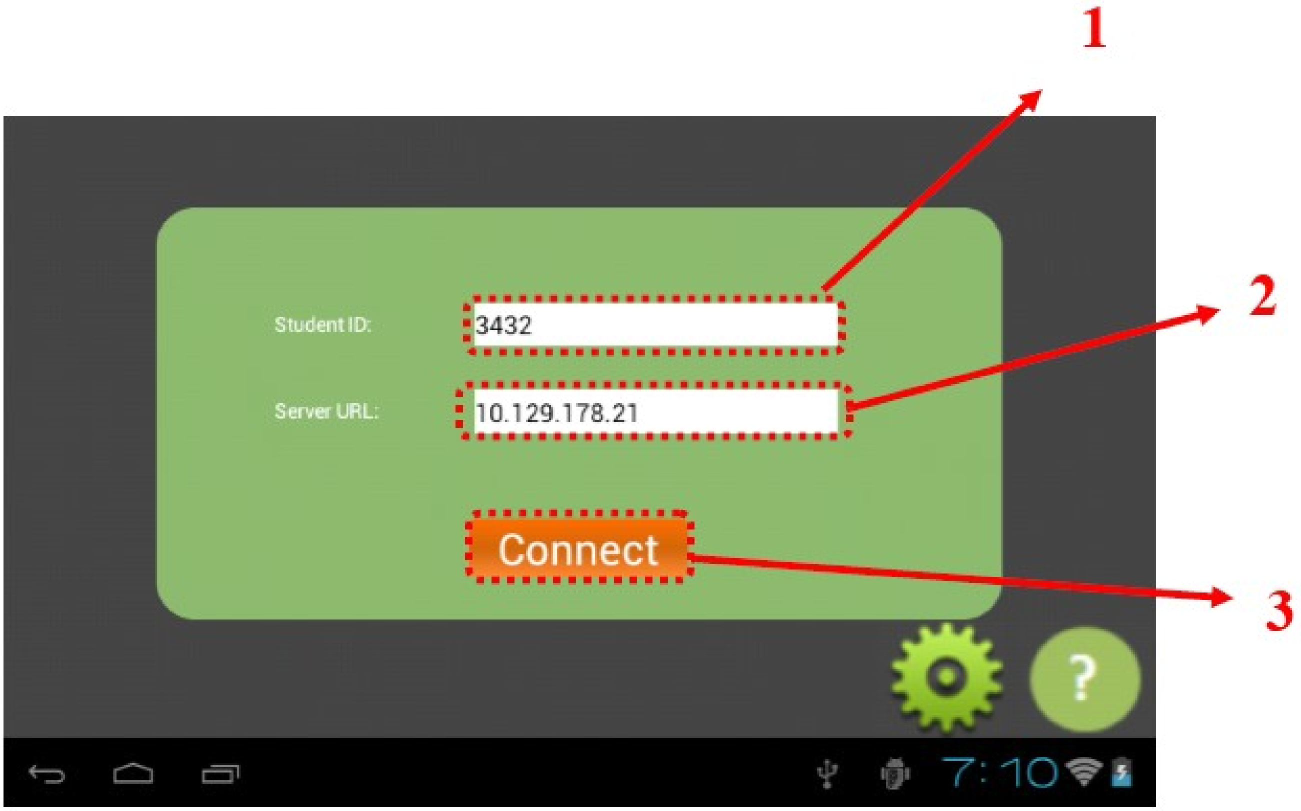
- Add images with Questions for better understanding of questions
- Polling should have editable options and a feature to add more choices when needed, also students can optionally give their views in brief with votes on same topic.
- Enable students to know their relative performance after each quiz.
- Teacher should able to view a Students performance for entire semester, overall performance as well as performance in each quiz
- Lastly, if the Instructor is physically unable to attend any lecture, he can conduct the lecture remotely or have pre-planned activity for students, which would be conducted automatically in absence of Instructor.
- Tutorials can be integrated into the application where in the student can browse through the subject whenever required.
- Use GPS for giving quiz such that, no two near students gets the same order.
- To implement batching algorithm so that access point will be more.

Chapter 9

APPENDIX

9.1 APPENDIX-A(User Manual)

Login:

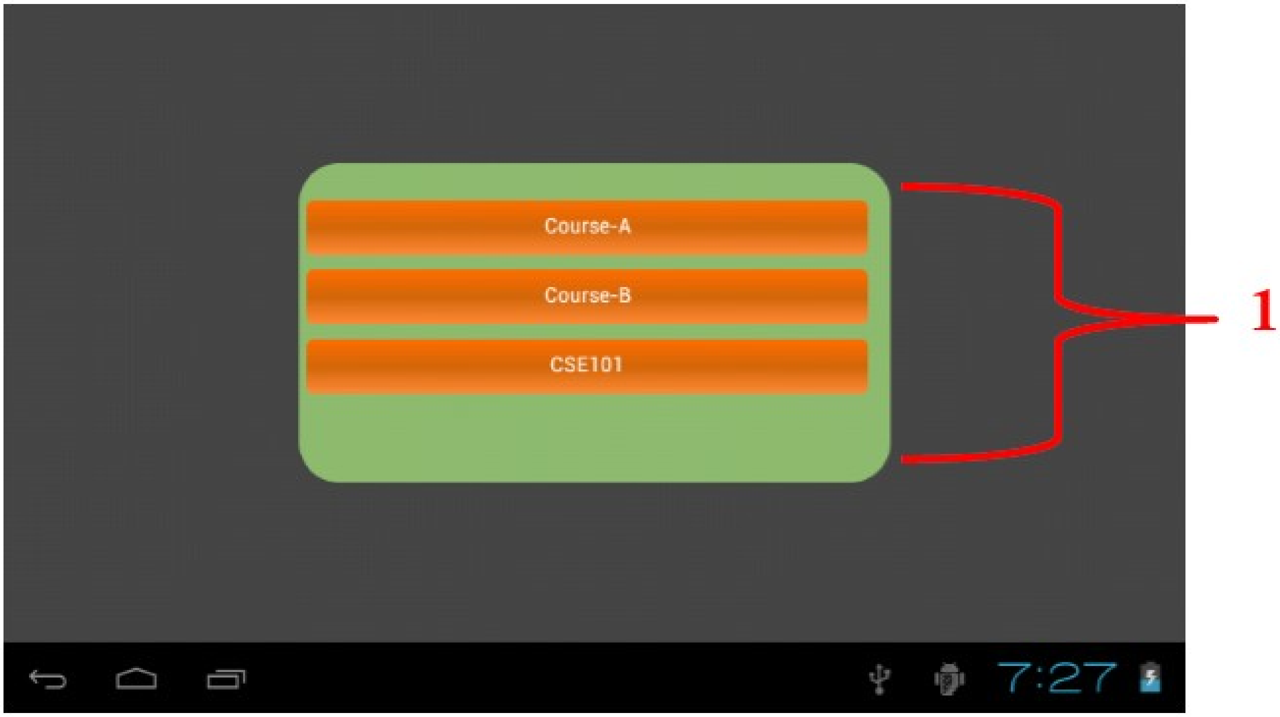


Student ID: 3432

Server URL: 10.129.178.21

1. Enter Student ID
2.Enter Server URL i.e, IP Of your instructor
3.Click on connect to proceed further

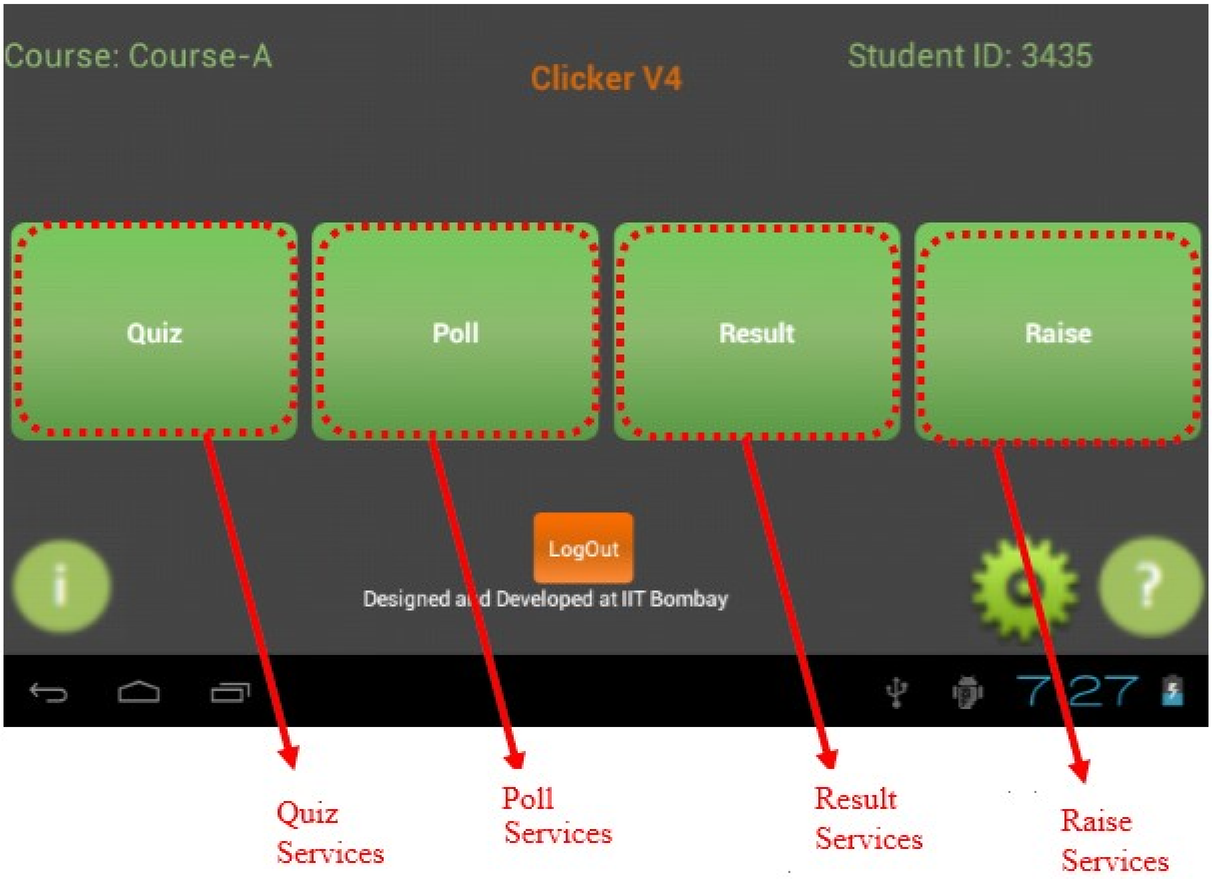
Registered Course list:



1. Registered course list

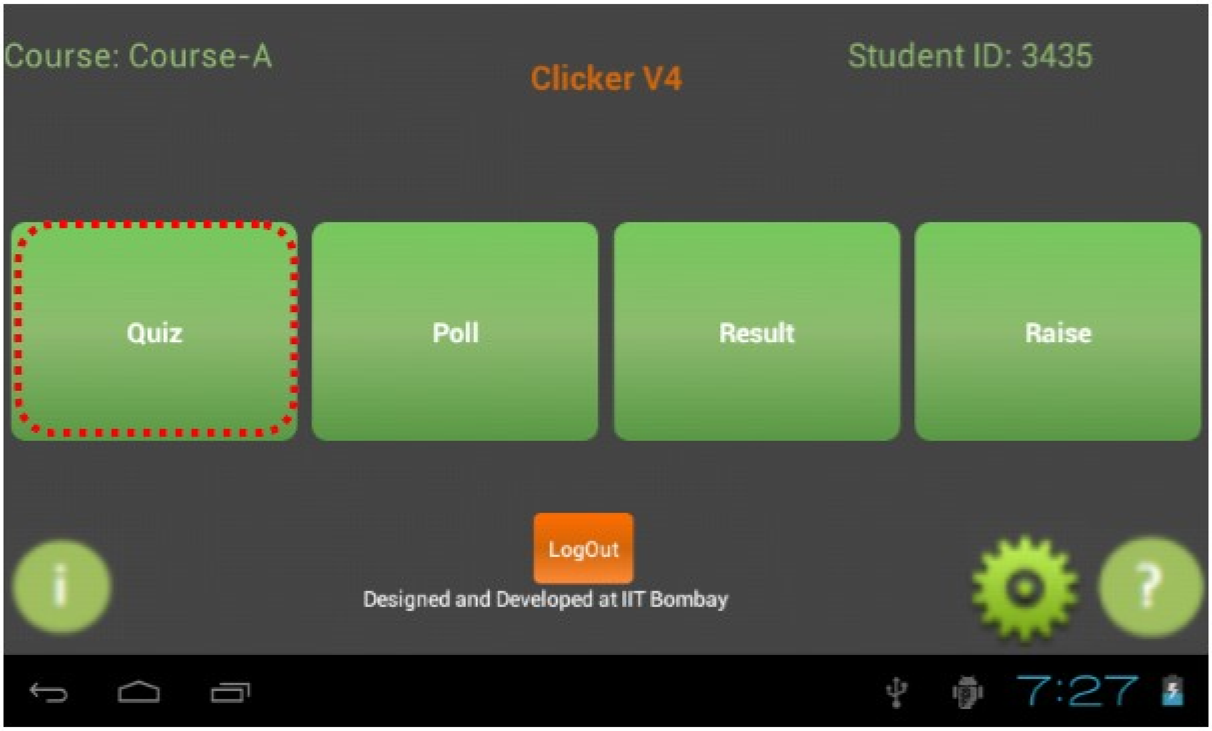
After clicking on connect, Tablet connects to wifi and requests server for course list and after receiving course list, wifi gets off and course list is displayed as like the above page.

Home page:



After selecting course from course list, it displays home page like this. It contains all the possible ways to communicate with the instructor like quiz, poll and raisehand. If you want to do the quiz then click on Quiz. After that we will get pages like below.

Quiz module:

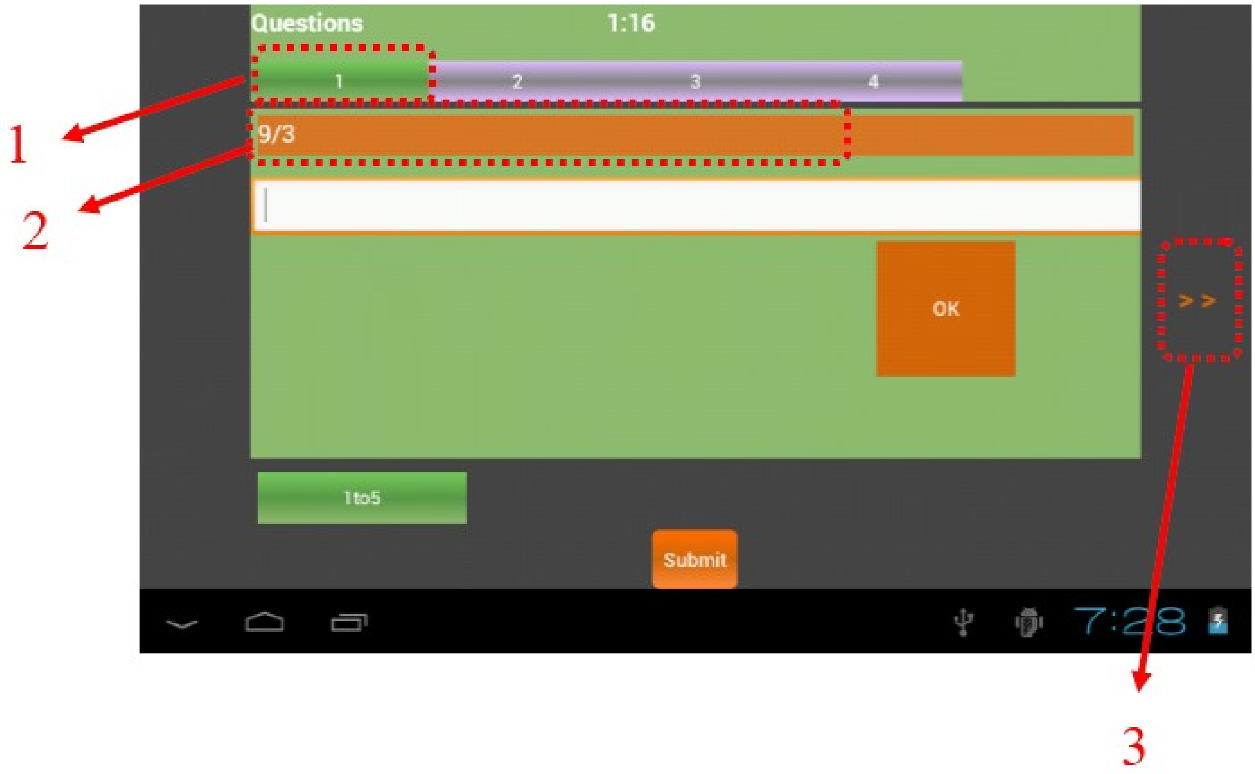


Course: Course-A

Student ID: 3435

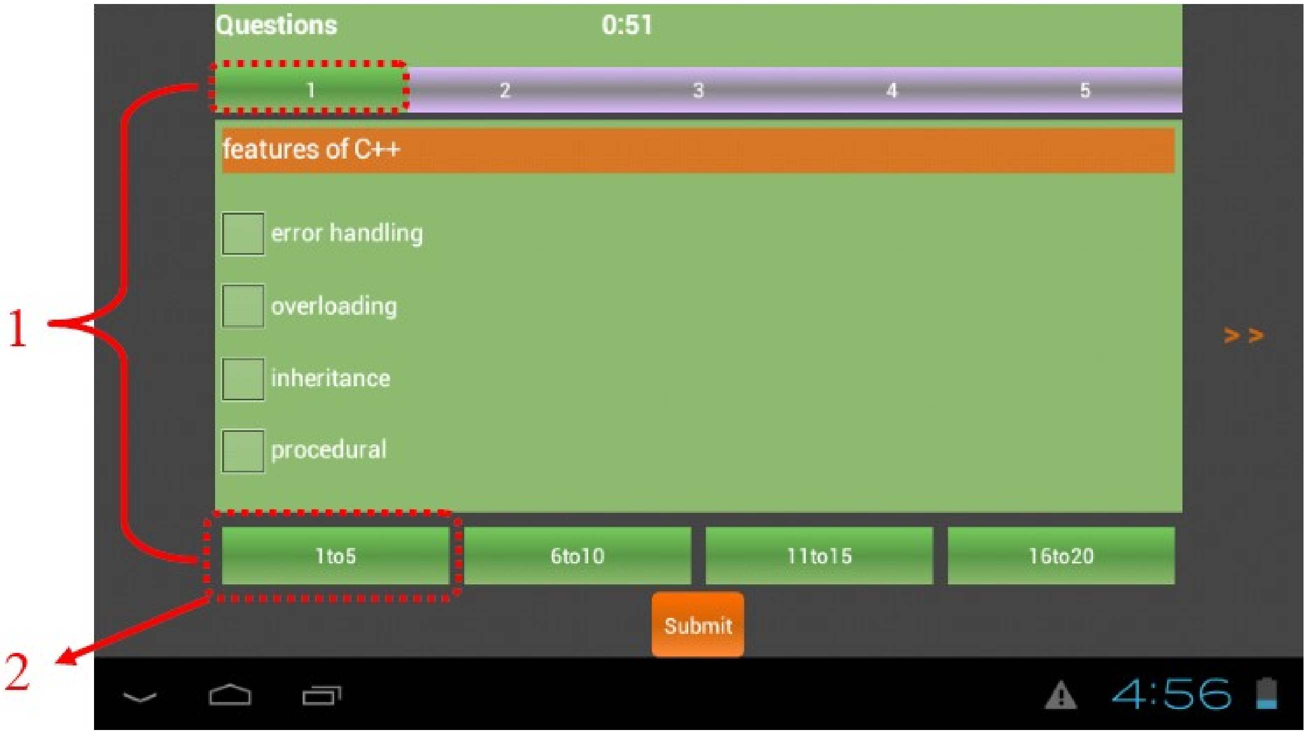
After selecting quiz, we get fetching data, where inbackground tablet on’s wifi and connects to server for getting quiz json request. After receiving json file, Tablet displays quiz. In quiz we have different types of questions like:

Numeric type:



1. Present question number
2. Numerical question text
3. Navigation bar

Multiple type:



features of C++

□ error handling

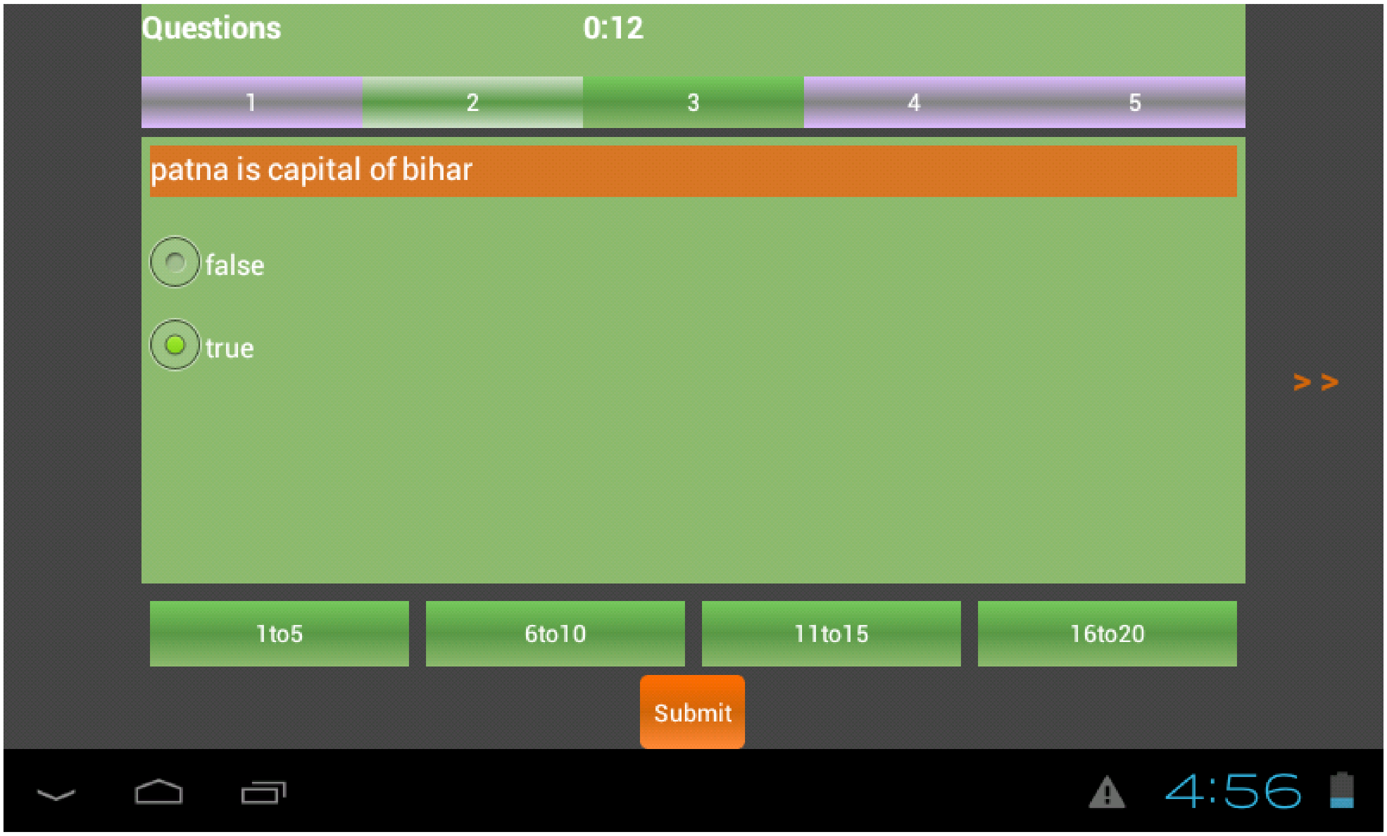
□ overloading

□ inheritance

□ procedural

1. First question of the first block that is ranging from 1 to 5
2. First block containing 1 to 5 questions

True or false:

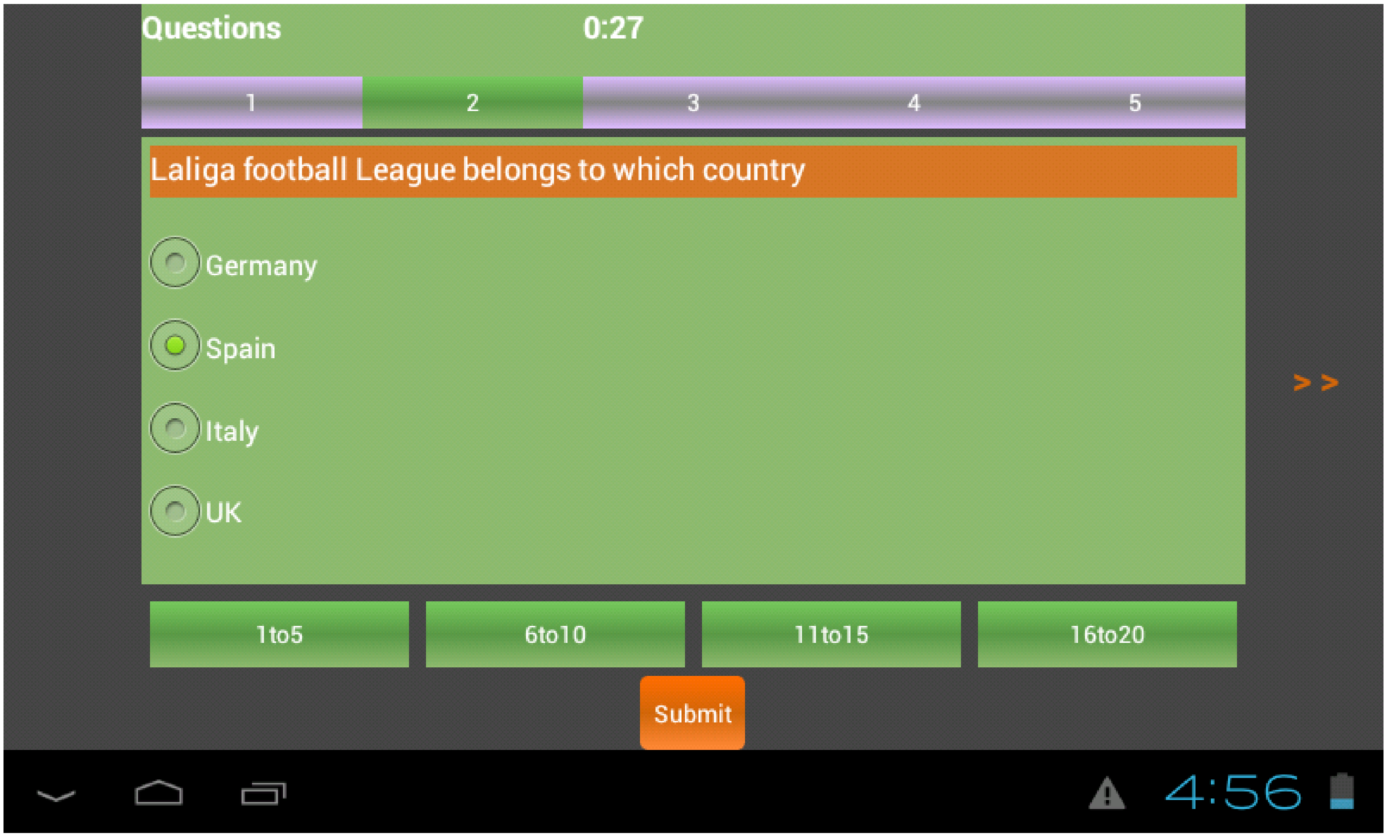


patna is capital of bihar

○ false

● true

Multiple choice



Laliga football League belongs to which country

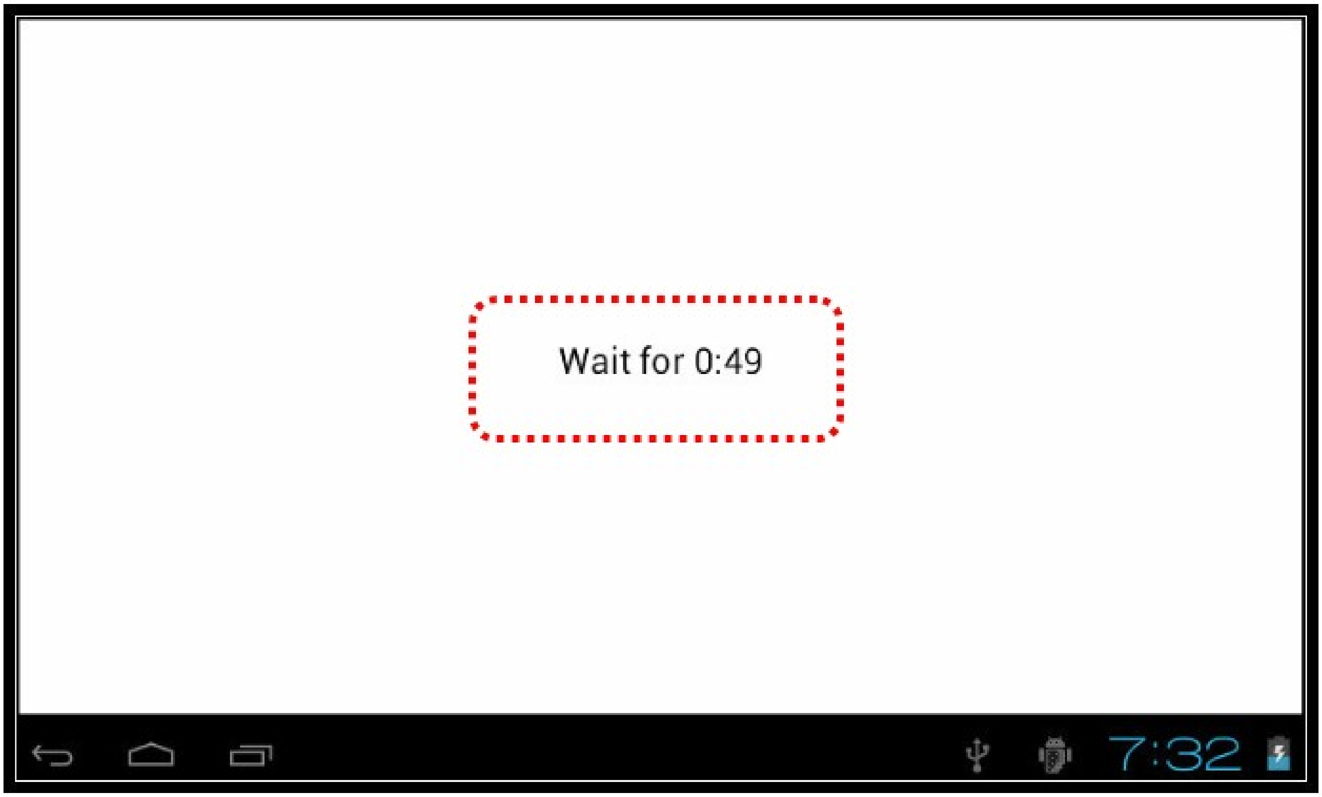
○ Germany

● Spain

○ Italy

○ UK

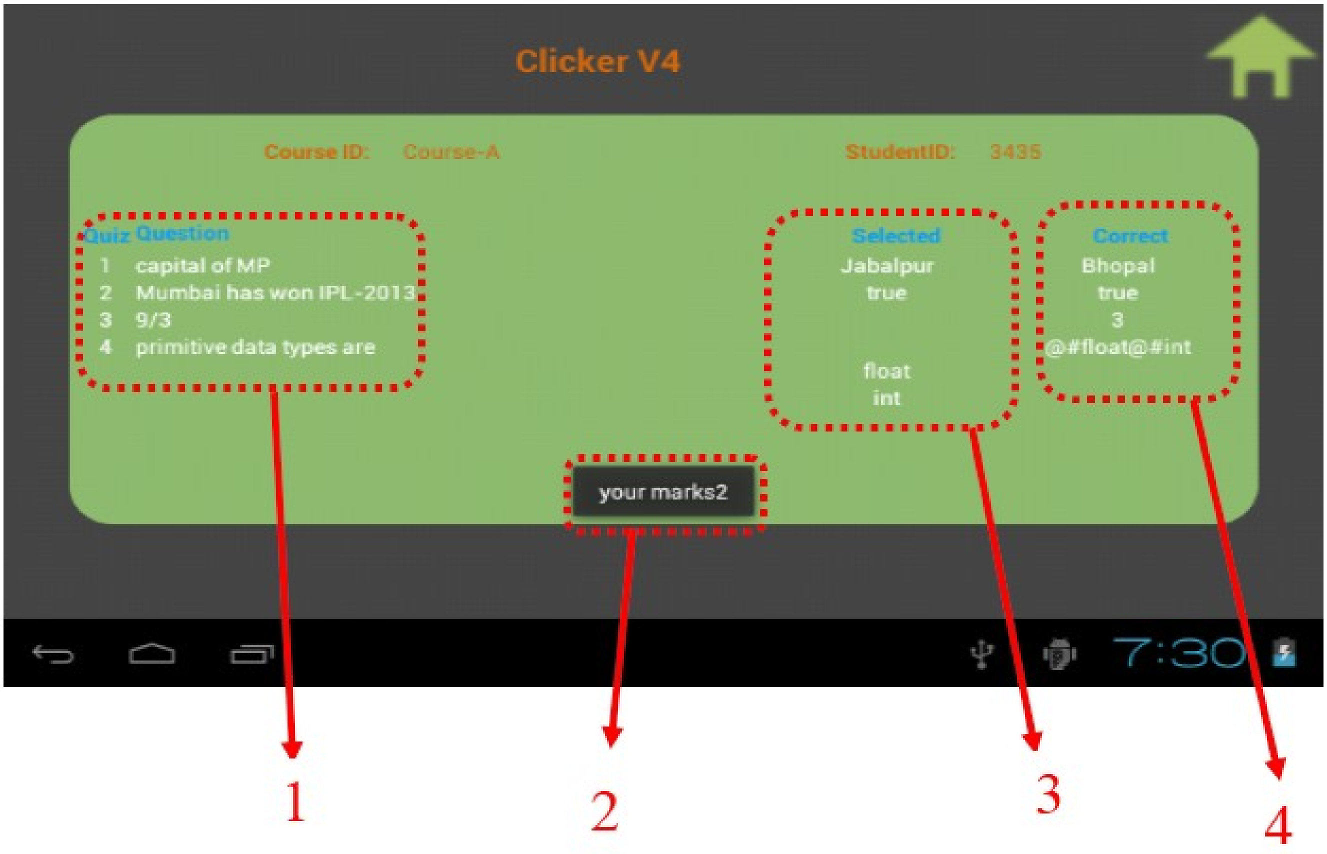
After submitting we get waiting time for all to submit, in order to synchronize and it is displayed as follow:



Wait for 0:49

While submitting, Tablet turns on wifi, connects to server, it sends the responses and waits for acknowledgment. If acknowledgment is received then instant result will be displayed.

Instantaneous result



Clicker V4

Course ID: Course-A

StudentID: 3435

Quiz Question

1 capital of MP

2 Mumbai has won IPL-2013

3 9/3

4 primitive data types are

Selected

Jabalpur

true

float

int

Correct

Bhopal

true

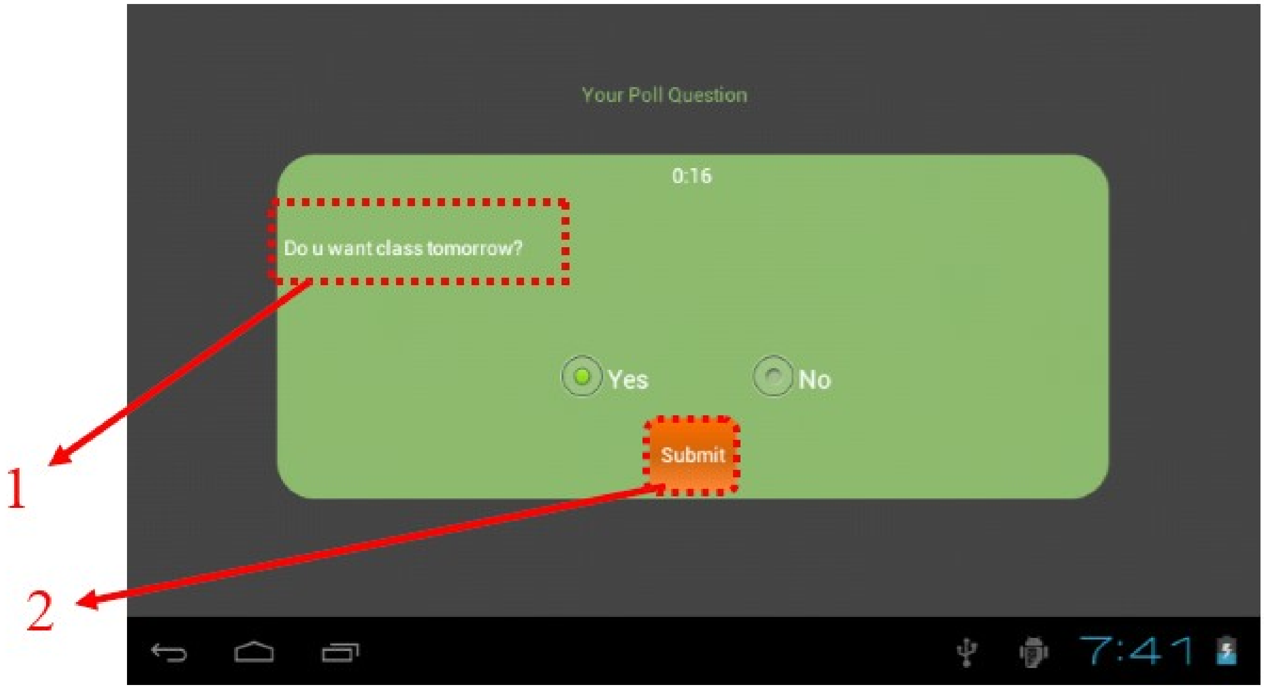
3

@#float@#int

your marks2

1. Quiz question that has sent by server
2. Marks that you have secured in the quiz
3. Your selected options/choices
4. Correct answers for the respective questions

If you want to poll after completion of quiz, then click on home button of result page, then it will moves to home page, as we have seen previously. Now, click on poll button which results in a page like this:



Your Poll Question

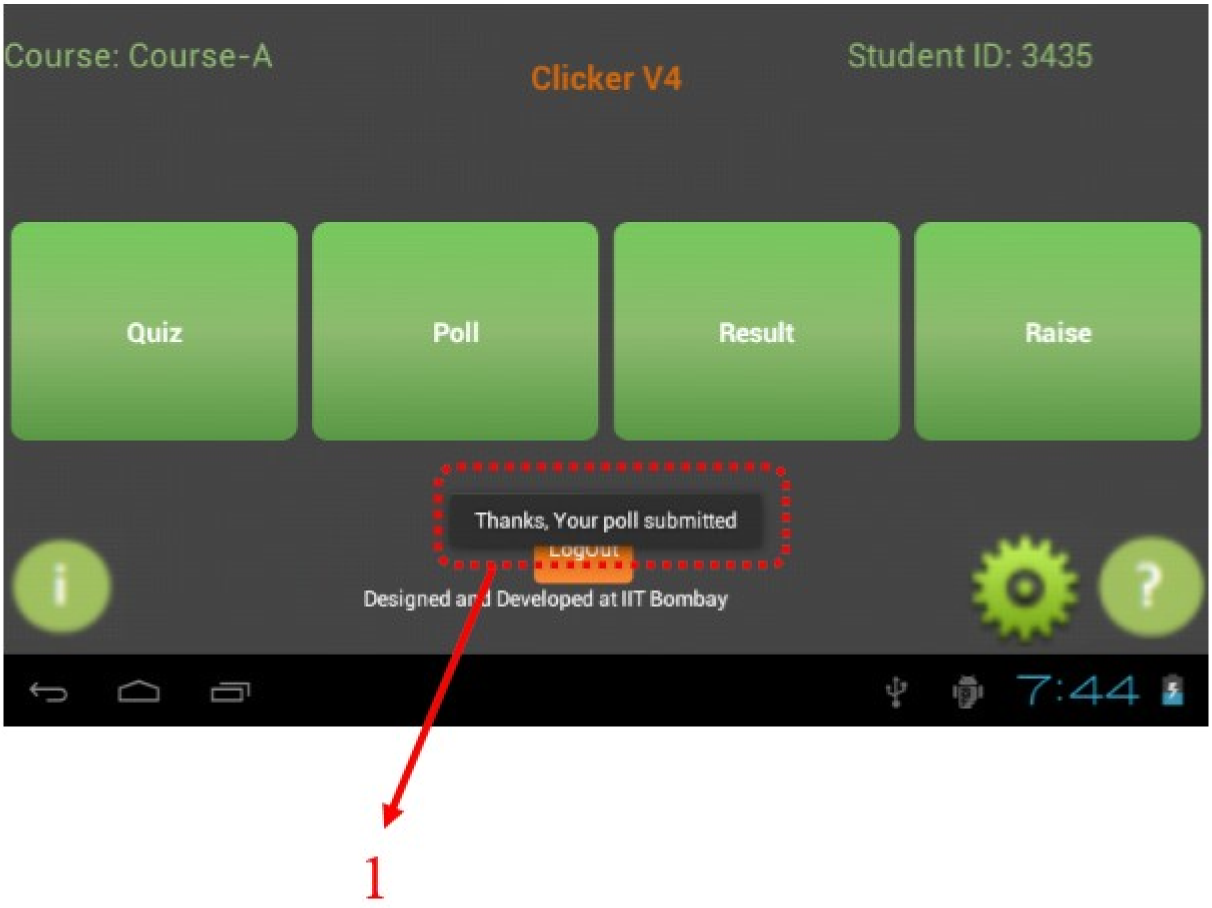
0:16

Do u want class tomorrow?

● Yes ○ No

1. Polled question of Instructor
2. Click on Submit buttton to submit your pollquestion to instructor.

After selection of poll, Poll page will be displayed with poll question and timer running(Countdown Timer), by selecting true or false you can give your opinion about the polled question. And click on Submit button to submit the poll. On submission you will get a page like this.



Course: Course-A

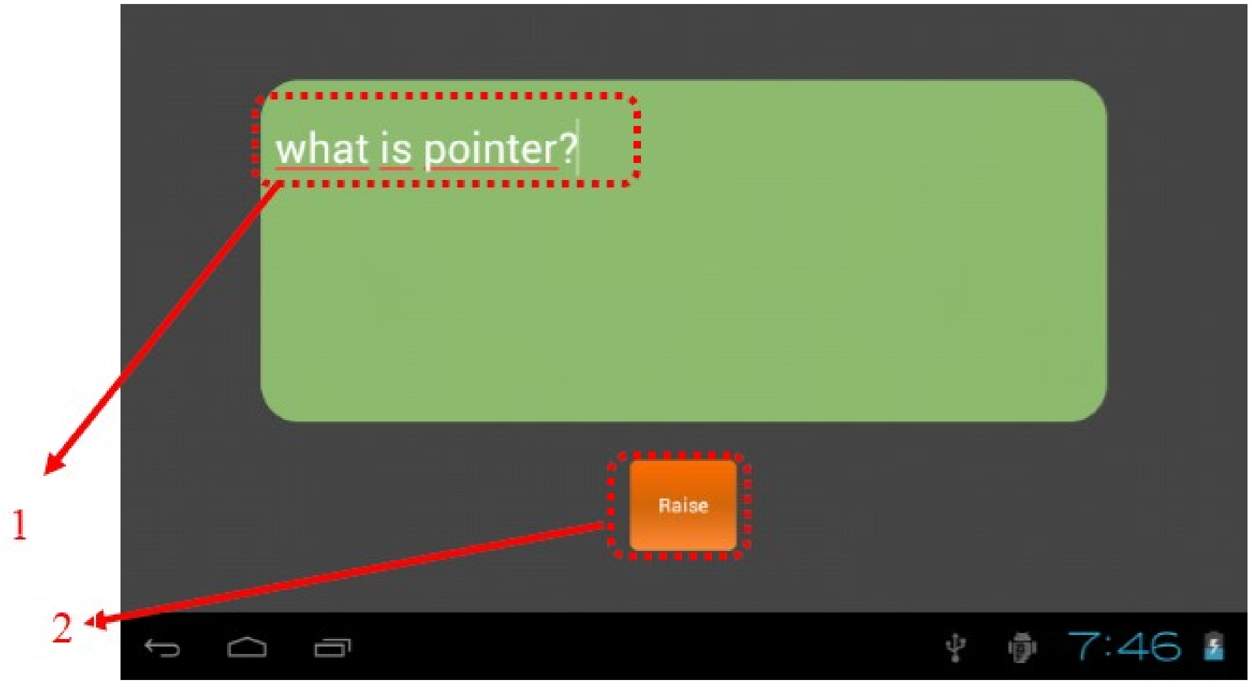
Clicker V4

Student ID: 3435

Thanks, Your poll submitted

1. Intimating you that your poll is submitted successfully.

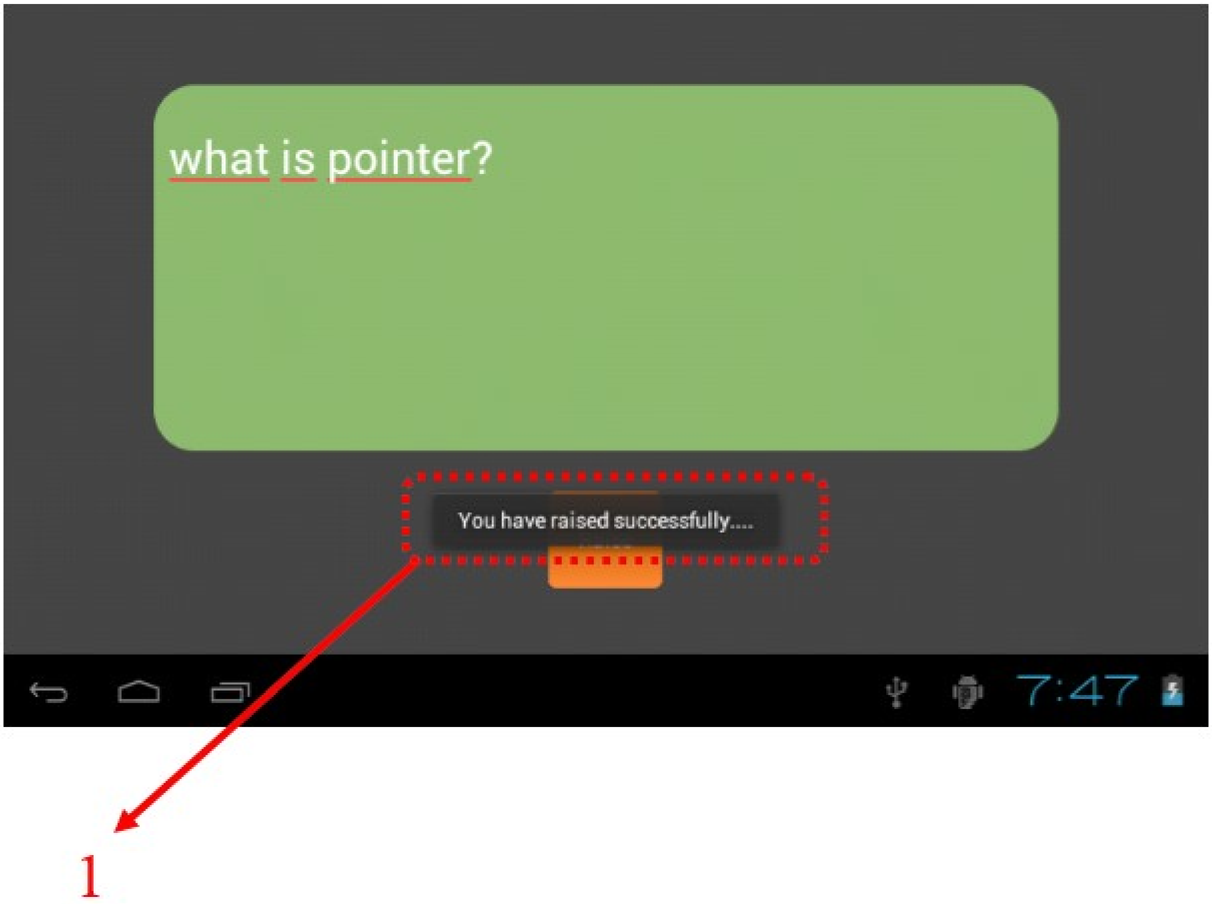
As like poll, you can move to raisehand page. The page will be displayed like this.
RaiseHand Page:



what is pointer?

1. The question raised by student
2. Click on Raise button to submit your doubt regarding the course.

On clicking on raise button to submit the raised question, you will get a page like this.

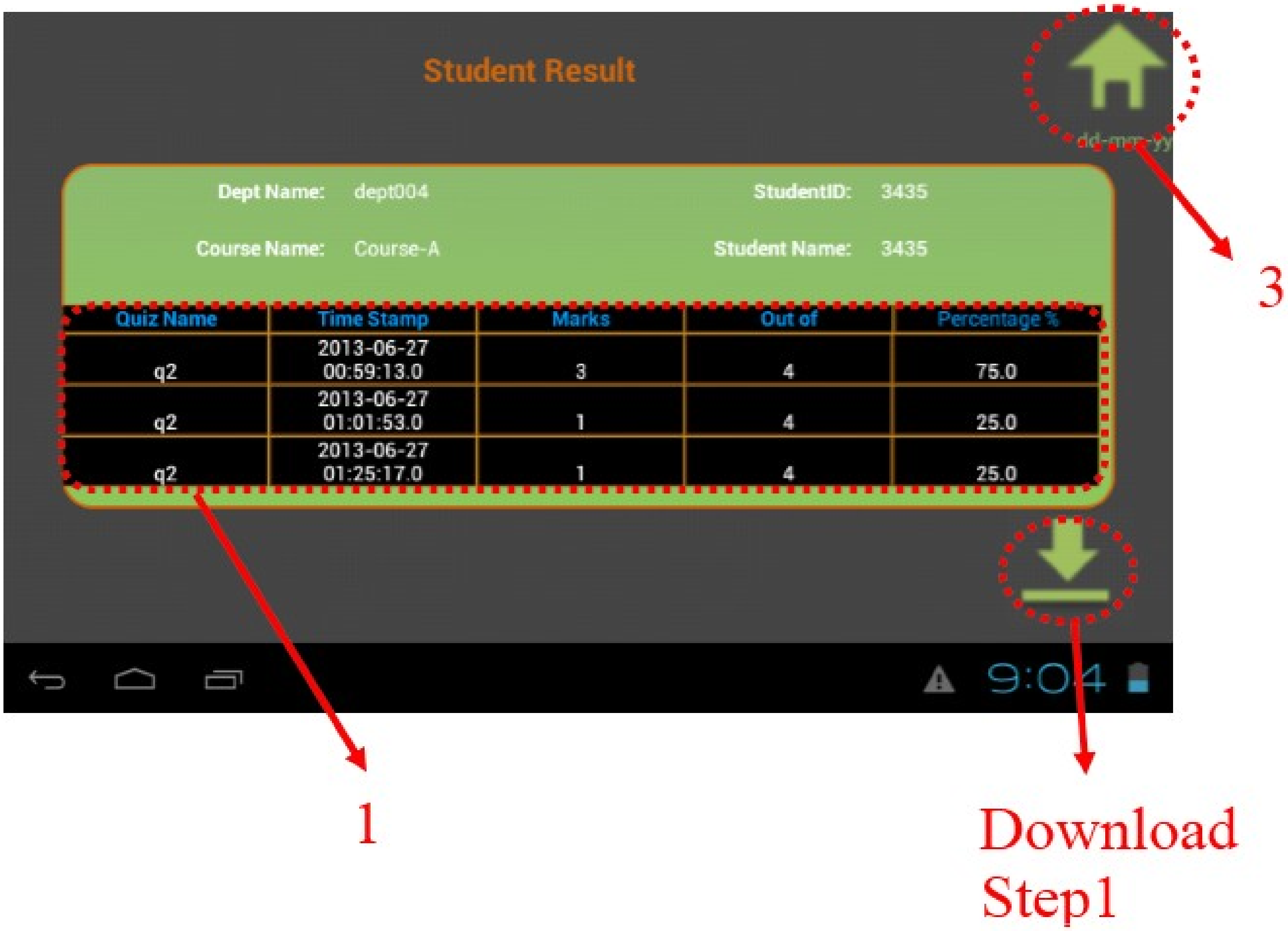


what is pointer?

You have raised successfully....

1. Intimating the student that their doubt is submitted to the instructor successfully.

If you want to see previous results of quizzes that you have appeared for, click on Result button of home page. The Result page will be like this:



Student Result

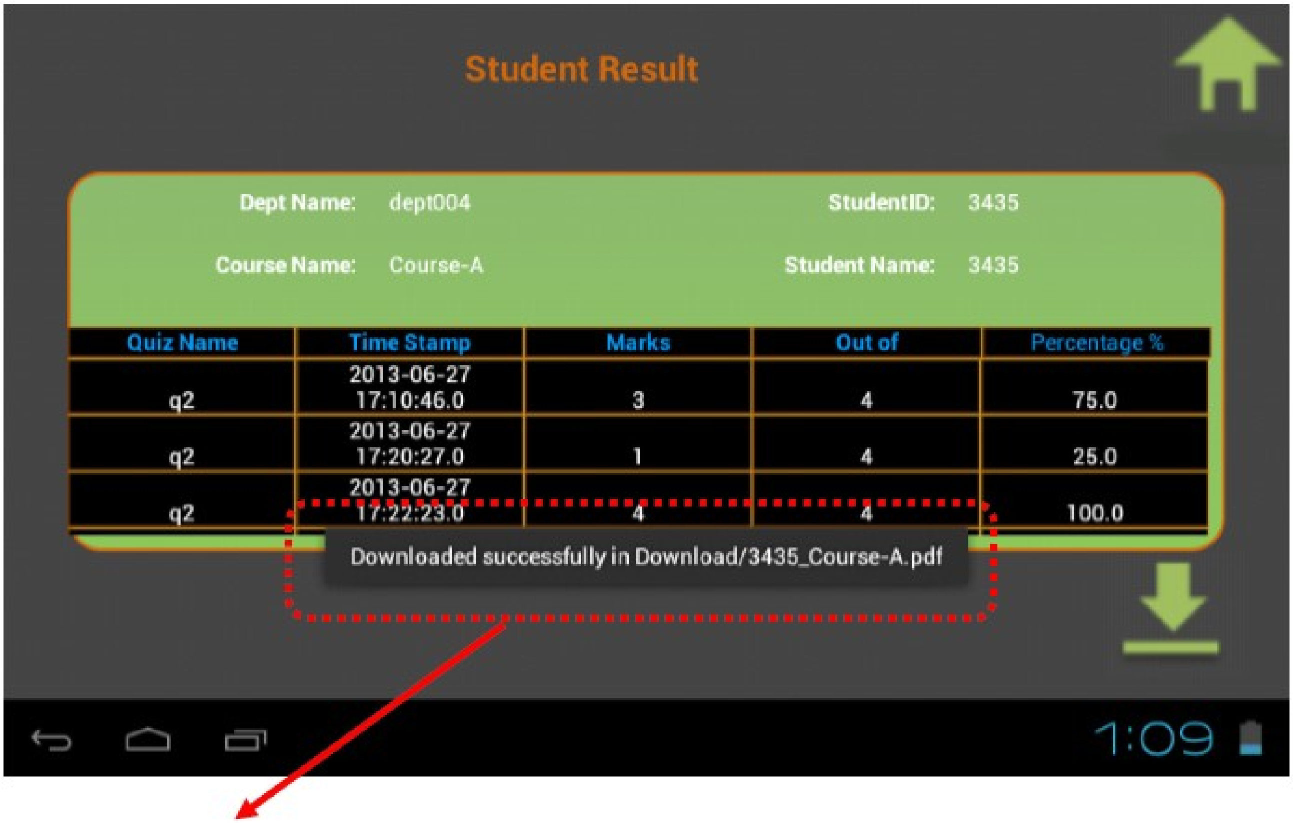
Dept Name: dept004 StudentID: 3435

Course Name: Course-A Student Name: 3435

Quiz Name	Time Stamp	Marks	Out of	Percentage %
q2	2013-06-27 00:59:13.0	3	4	75.0
q2	2013-06-27 01:01:53.0	1	4	25.0
q2	2013-06-27 01:25:17.0	1	4	25.0

1. Previous quiz details in particular course
2. Dowload icon
3. To go to home click on home icon

On clicking on download button, we will get a page like this



Student Result

Dept Name: dept004

StudentID: 3435

Course Name: Course-A

Student Name: 3435

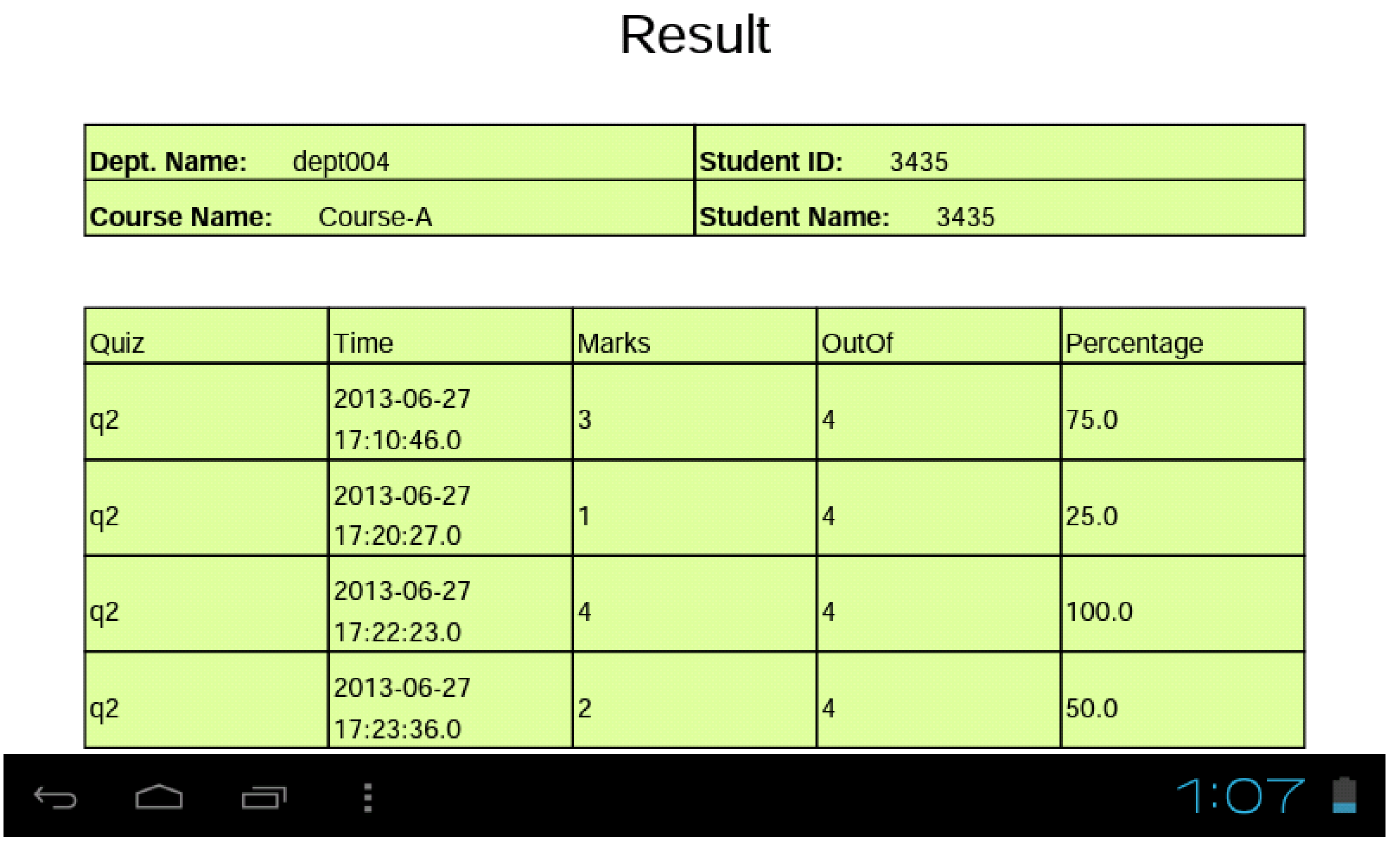
Quiz Name	Time Stamp	Marks	Out of	Percentage %
q2	2013-06-27 17:10:46.0	3	4	75.0
q2	2013-06-27 17:20:27.0	1	4	25.0
q2	2013-06-27 17:22:23.0	4	4	100.0

Downloaded successfully in Download/3435_Course-A.pdf

Download Step2

Intimating the student about the download location and download is done successfully.

The downloaded pdf will be saved in downloads of filemanager. And the result will be saved in this format.

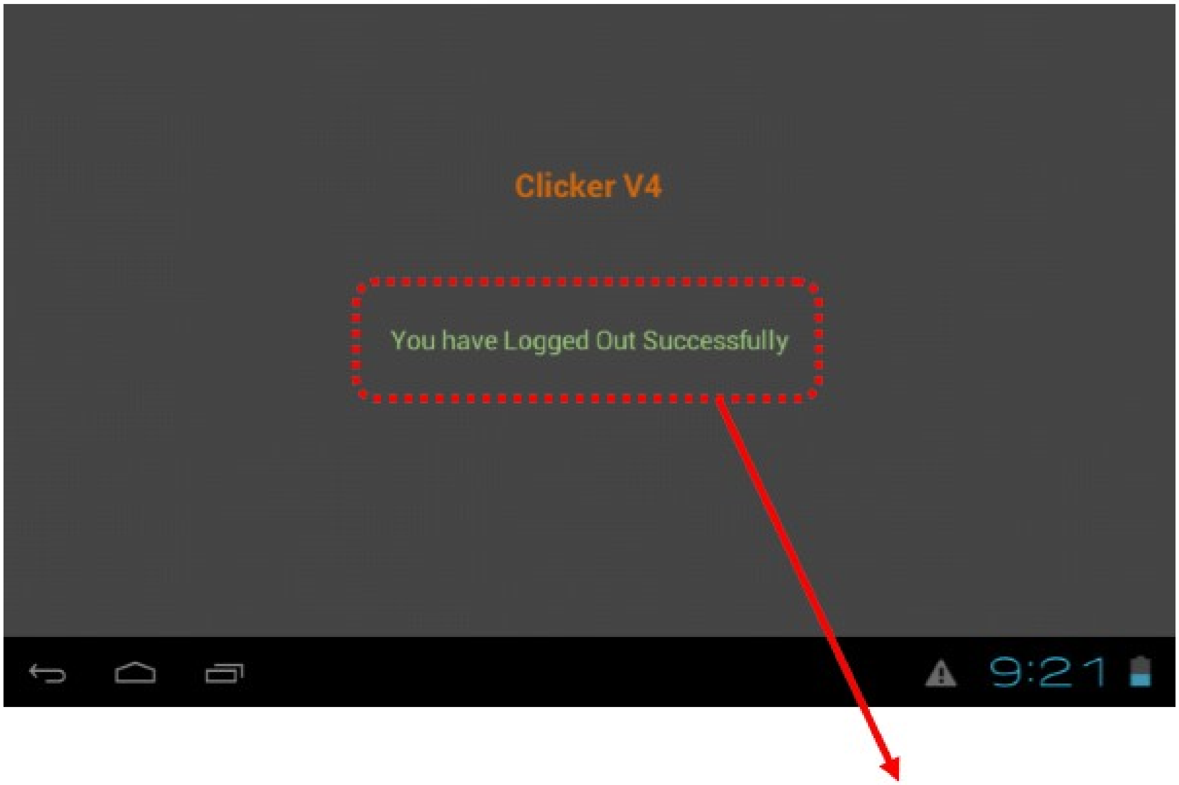


Result

Dept. Name: dept004	**Student ID:** 3435
Course Name: Course-A	**Student Name:** 3435

Quiz	Time	Marks	OutOf	Percentage
q2	2013-06-27 17:10:46.0	3	4	75.0
q2	2013-06-27 17:20:27.0	1	4	25.0
q2	2013-06-27 17:22:23.0	4	4	100.0
q2	2013-06-27 17:23:36.0	2	4	50.0

Logout Page:



Clicker V4

You have Logged Out Successfully

Message displayed after pressing “LOGOUT”

If you want to come out of the application, you click on logout button of home page.

9.2 APPENDIX-B(FAQ):

Frequently asked Questions

Q.) From where you can get description of ResT technology used in our projects?

You can get proper ideas about rest architecture from books *RESTful Java Web Services*, witten by Jose Sandoval and RESTful Web Services By Leonard Richardson, sam Ruby or you can always find number of links on google for any topics related to Restful web services technology.

Q.) Have you included any external jars files in your project?

Yes, you have to import the jersey-jars that you can downloaded from http://jersey.java.net into the WEB-INF/lib folder. Right click that folder (WebContent/WEB-INF/lib) and select Import.Now that the jars are in the WEB-INF/lib folder, you will need to configure the project to include these jars in its build path.

Q.) Where to find the backup of the database created by admin?

The backup of the database is stored in the backup folder inside the main project folder with the name containing the current date (eg if the backup is created on 2/7/2013 then the name of the file will be Dump20130702.sql).

Q.) Where to find all the work done by summer interns 2012?

You should contact Mr.Rajesh Kushalkar, Mr.Chetan, Mr.Rajavel, Ms. Dipti for the same. You will find them in Clicker lab in KRESIT building.

Q.) What to do in case of any doubts?

If there are some specific doubts which you want to discuss with previous interns, you can take contact numbers from Mr.Bikas Chhatri from 4rth Floor office in KRESIT.

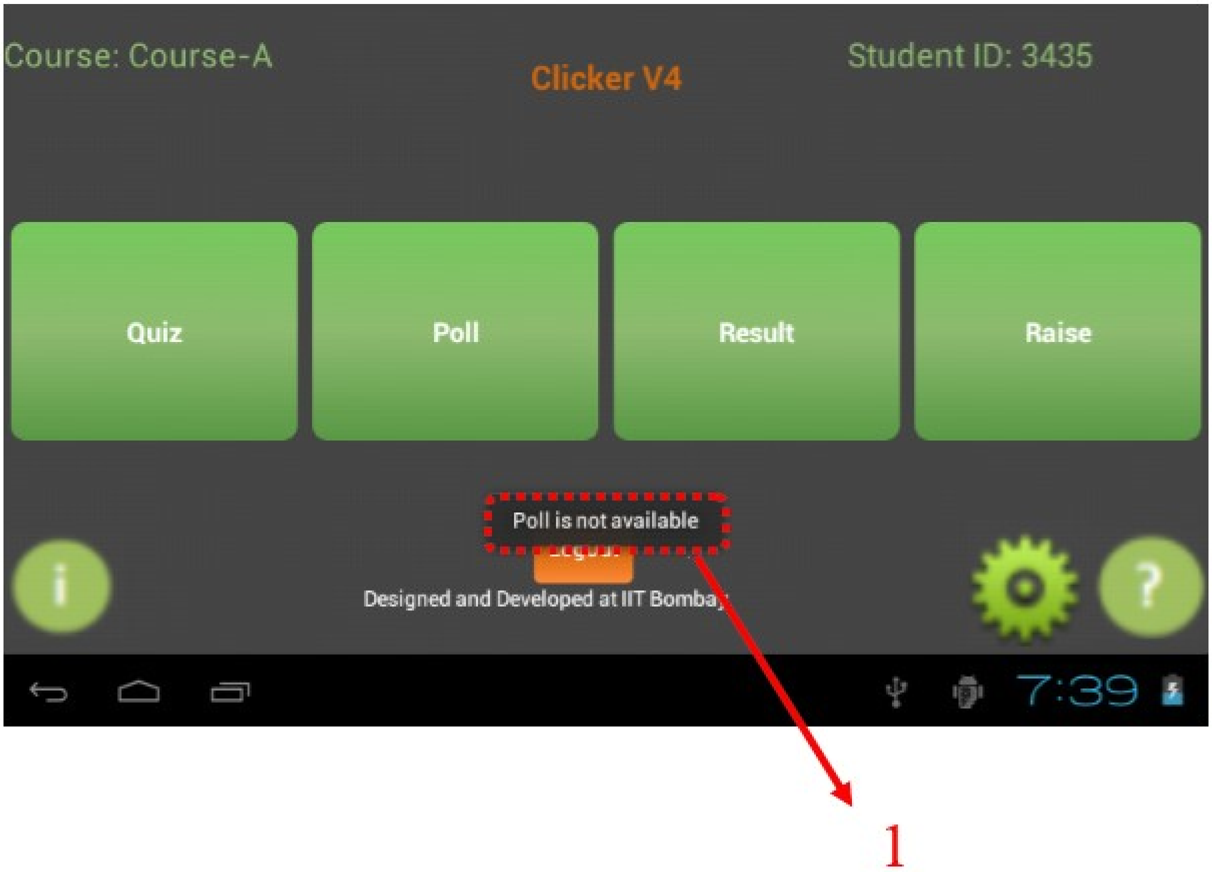
Q.) How to use this apk?

Go through the user manual.

9.3 APPENDIX-C(Test Cases)

Test case 1:

Test Case Name: Unit Testing
Description: If the student enters a wrong student id which already exists then a message will be displayed that student id is invalid.
Output: A dialog box that will show “student id invalid”

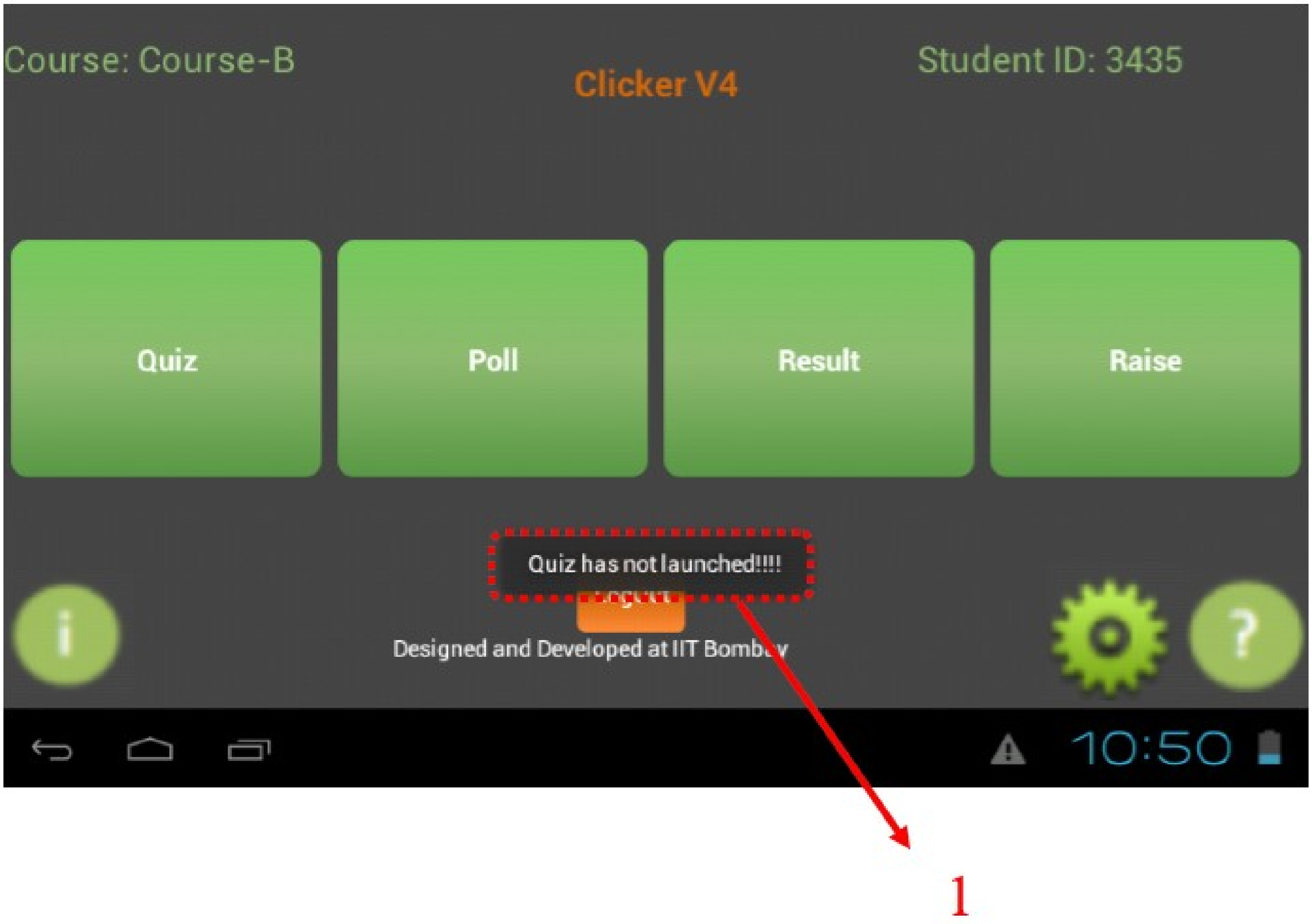


Poll is not available

1. If you enter wrong studentID then you will get a message like this.

Test case2:

Test Case Name: Unit Testing
Description: If the student enters wrong server ip addresss then it will display a toast that server is not found.
Output: A dialog box that will show “server not found”.



Course: Course-B

Clicker V4

Student ID: 3435

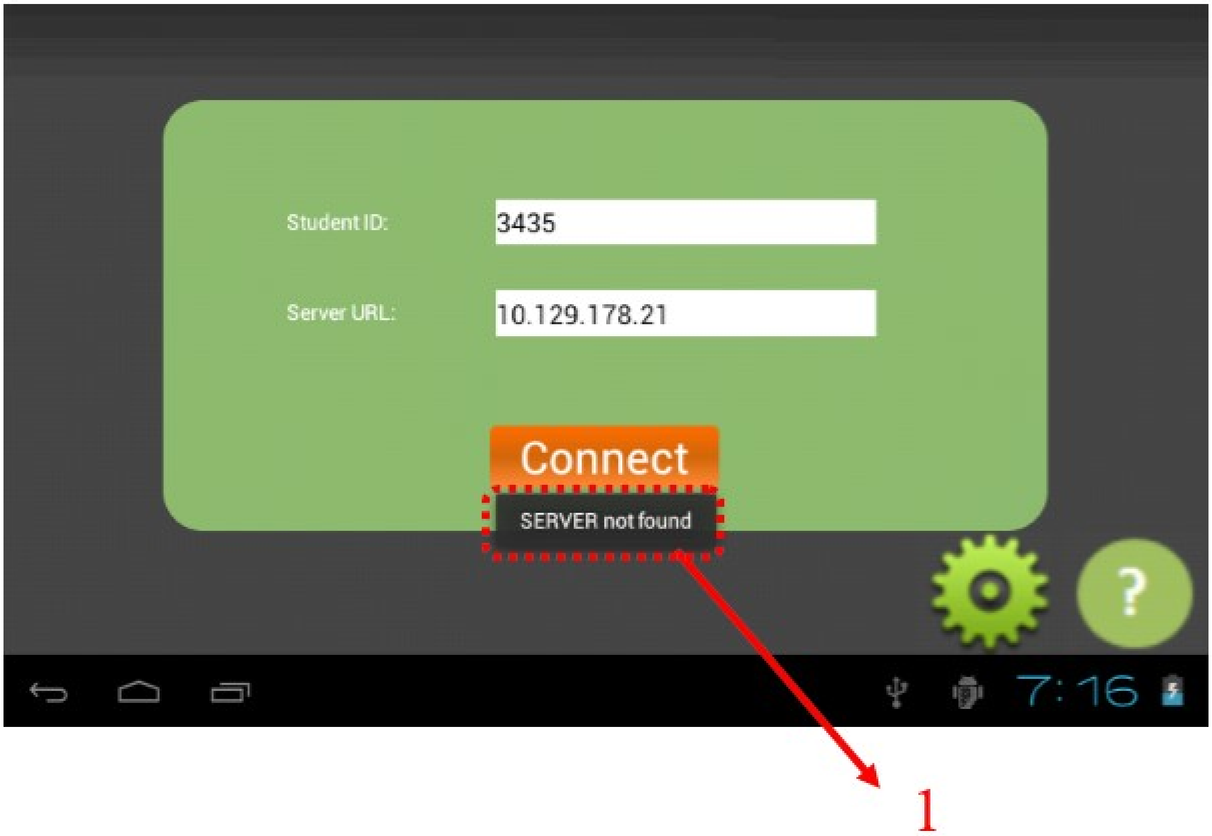
Quiz has not launched!!!!

Designed and Developed at IIT Bombay

1. If server is not available or wifi is not in range then you will get a message like this.

Test case 3:

Test Case Name: Unit Testing
Description: If the student enters quiz when quiz is not launched then it will display a toast that quiz is not launched.
Output: A dialog box that will show “quiz not launched”.



Student ID: 3435

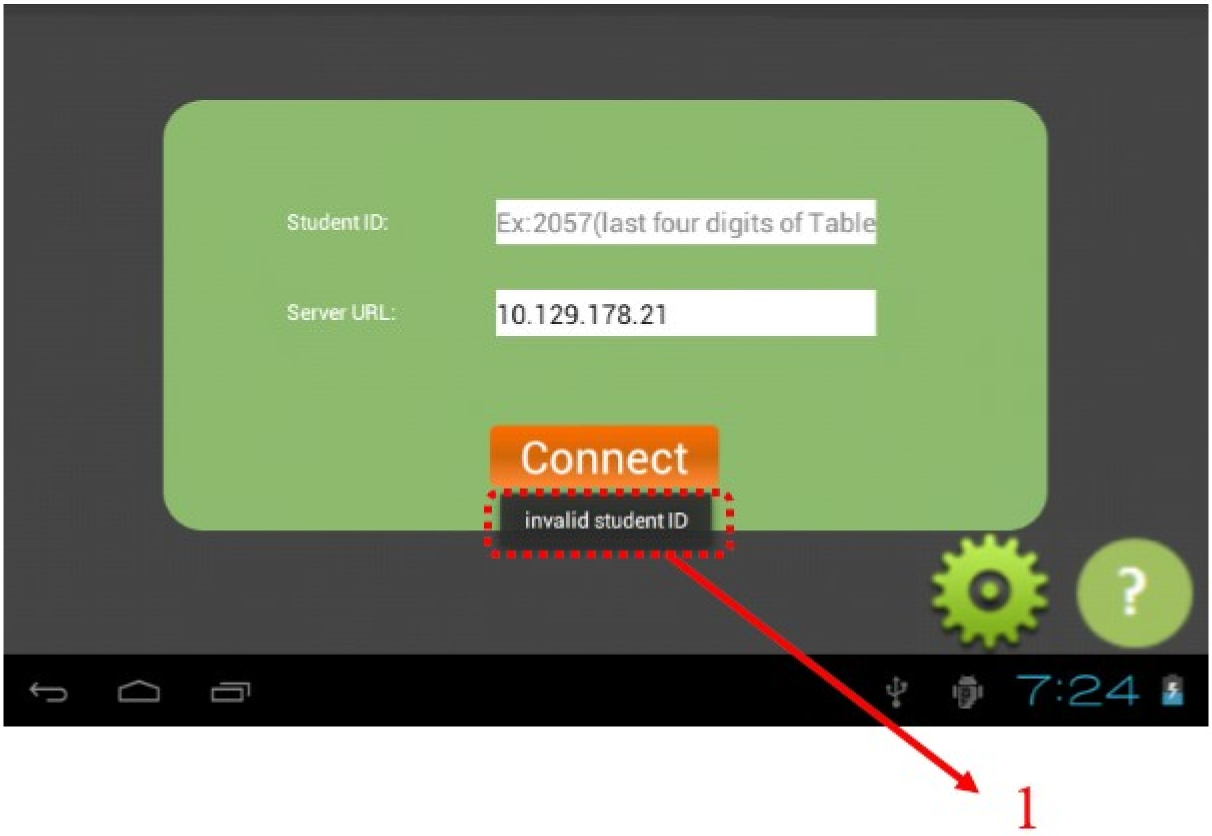
Server URL: 10.129.178.21

SERVER not found

1. If you click on quiz when the quiz is not launched by instructor you will get this message.

Test case 4:

Test Case Name: Unit Testing
Description: If the student enters poll when poll is not launched then it will display a toast that poll is not launched.
Output: A dialog box that will show “poll not launched”.



Student ID: Ex:2057(last four digits of Table

Server URL: 10.129.178.21

invalid student ID

1. If the instructor is not launched the poll then you will get this message.

Statistics of Tests conducted at ASL lab on two different apks:

1) Throughout Wifi-APK

Quiz	Number of Tablets	Number of user Downloaded	Number of Tablet Response	Remarks
1	55	55	54	One student response is not sent.
2	55	55	55	Good
3	55	55	55	Server got all responses. But in one tablet the page is not displayed.
4	55	55	55	Good
5	55	55	55	Good
6	55	55	55	Good
7	74	77	74	Two tablets request submitted twice because they killed the application and again requested for quiz.
8	75	75	75	Good
9	74	74	74	Good
10	75	75	74	Response is send to server,but apk remains in quiz page while the arrangement of students is scattered.
Server and client Response time: Quiz Download : maximum 4 sec average : 1.5 sec Response Time : maximum 4 sec average :1.5 sec				

2) **Wifi ON/OFF:**

Quiz	Number of Tablets	Number of user Downloaded	Number of Tablet Response	Remarks
1	37	37	37	Latency time of worst case is 5sec
2	50	50	50	Latency time of worst case is 20sec
3	50	50	50	Latency time of worst case is 25sec
4	65	65	64	One tablet had Wifi connection problem
5	70	67	66	2-tablets took more time, 1 tablet hanged in between
6	66	66	64	1-tablet response was sent after time out, In 1 tab result was not submitted.
7	75	75	73	In 2 tablets response was not submitted.

9.4 APPENDIX-D(Challenges & Solutions)

Client- Side:

- Automatic connection and disconnection of Wifi while communicating with server**(Login,Quiz,Poll,Raise Hand,Result)**.
- Giving Batch No. based on random number generated for reducing router load.
- Displaying multiple type(fill in the blank,multiple answer,objective questions) of questions and options randomly(different order) in **Quiz Module.**
- **Implementing REST** technology for maintaining independent requests(No sessions can be formed).
- Designing GUI to support multiple screen types.
- **Time synchronization** of server and client.
- Downloading **Result statistics**(previous quizes attempted) in **pdf** format.
- Avoiding multiple attempts for the same quiz from a single student.
- Displaying **instantaneous result**, selected option along with question and correct option .

Server-Side:

- Maintaining results details of individual student in Database.
- Getting **current time stamp** at time of quiz request.
- Handling **multiple Instructors** and Students **simultaneously**.
- **Evaluation** of Response for multiple choice questions.
- Keeping track of each student response and total counts of students responding in **concurrent-hash-map**.
- Maintaining count of students for particular option in Quiz&Poll.

Chapter 10

Bibliography

www.stackoverflow.com
www.developer.android.com
www.w3schools.com
www.avilyne.com
www.itextpdf.com
www.mkyong.com
www.ibm.com/developerworks/library
www.utilities-online.info
docs.oracle.com/javase/6/docs/api/java/util/ConcurrentHashMap.html