

Functional Extension to Clicker **Software**

Summer Internship 2011

Submitted in fulfillment of internship project

By

Clicker Software Team

Under the Guidance of
Prof. D. B. Phatak



Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai

Acknowledgment

We the summer interns team of Clicker Software are over helmed in all humbleness and gratefulness to acknowledge our depth to all those who have helped us to put our ideas and assigned work, well above the level of simplicity and into something concrete.

We all thank whole heartedly Prof. D.B. Phatak for selecting us a part of his valuable project, constantly motivating for doing better and showing complete confidence in our work.

We are very thankful to our mentors Mr.Tushar Kambli , Mr.Manjurelahi, Mr.Rajavel, Mr. Gobinath and the hardware team mentors for their valuable help. They were always there to show us the right track when we needed help. With help of their valuable suggestions, guidance and encouragement, we all were able to complete our tasks properly and with satisfaction. Also in the process, we learnt a lot other technical and non technical things from them and we consider ourselves to be very fortunate to have such mentors.

We would like to thank Mr.Bikas Chhatri for making our stay here as summer interns comfortable and for all his administrative help. Without him a lot of time would have gone in doing extra work rather than the main project work.

Finally we also like to thank all other colleagues working in different projects under Prof. Phatak for helping us at small problems as well as critical junctures.

Team Members

Dasa Anand Baskaran(NIT, Surathkal): Worked on Open office Integration and helped all in installing various softwares

Hardik Patel(S.V.NIT, Surat): Worked on Open office integration and ubuntu internal system details guide to all

Megha Tak(DA-IICT, Gandhinagar): Worked on Python Integration and did most of the compilation and presentation work of the group.

Mannepalli Praveen Kumar(K.L.College of Engg, Vijayawada): Worked on GUI installer

Dhwani Shah(BITS, Pilani, Goa Campus): Worked on Python Integration and took active interest in hardware learning

Setu Saxena(SGSITS, Indore): Worked on Web Based Clicker and most enthusiastic member of the group

Shiwani Srivastava(NIT, Jamshedpur): Worked on Web Based Clicker and a sincere student

Niraj Chiktey(IET-DAVV, Indore): Worked on Web Based Clicker and only one in group who came under the same internship for the 2nd time .

MENTORS:

Mr.Tushar Kambli: Our guide for all problems and a linux OS mastermind.

Mr.Manjurelahi: A very helpful person and he helped a lot in coding work.

Mr.Rajavel: A great person to learn and discuss all Java problems with.

Mr.Gobinath: Has helped us all in the architecture of the project on whole.

Mr.Rajesh, Ms Savita, Ms Sharmila, Mr.Rishi: The very patient hardware team guides who helped in the hardware problems we faced regularly.

Abstract

Clicker – is a MHRD sponsored project aimed to simplify interaction between students and professors in lakhs of colleges all across India. As all across the world people are working to make technologies that reduce manual labour, this is one which does same for students and teachers. It is an affordable, interactive hardware cum software utility for collecting attendance , conducting quizzes and clarifying student doubts in lesser time and helping the instructor in analysis of quizzes, etc easily and precisely.

Hardware has clickers and receivers designed using chips, convertors and various cables. Software is a computational and analytical tool running on instructor machine which uses Python for hardware software interaction, MySQL for database storage and Java , JSP etc for report generation and web based modeling.

Contents

1. Introduction	
1.1 Purpose.....	6
1.2 Scope.....	8
1.3 Definitions, Abbreviations.....	9
1.4 Motivation.....	10
2. Objectives.....	11
3. Design Considerations	
3.1 Assumptions and Dependencies.....	12
3.2 General Constraints.....	14
4. Architecture	
4.1 Clicker Desktop Application.....	16
4.2 Clicker Web Application.....	17
5. Design and Implementation of Python Serial Utility.....	19
6. Design and Implementation of GUI Installer.....	33
7. Design and Implementation of Open Office Integration.....	40
8. Design and Implementation of Web Based Application.....	53
9. Conclusion and Future Work.....	66
10. Technical Details and FAQ.....	67
11. Appendix A (Screenshots).....	70
12. Bibliography.....	85

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 a clicker device, which is associated with his roll number.

Clicker is developed at IIT Bombay and the summer interns clicker software team worked with 4 different features of this automation device.

We worked with developing Python Serial Utility for proper packet sending and retrieval across hardware and software, making a GUI installer so that installing the software does not need any technical knowledge, Open Office Integration such that the software can be run from Open Office slides directly rather than switching modes and Web Based Application such that instructor can access the utility directly from internet without downloading anything.

1.1 Purpose

1.1.1 Python Serial Utility :

The purpose to provide the following documentation is to explain the working and functionality of Python in development of the Clicker SRS. Python is an interpreted, interactive, object-oriented, extensible programming language. It lets one work quickly and integrates different systems more effectively. This advantage of Python to integrate systems is used by Clicker Team by using the language to communicate between the hardware clickers , receiver and the software running to generate relevant output and store information. Python Integration as a work module aimed at making the Python code for communication between hardware and software using Serial library mostly , making it more Object Oriented , Making a standalone utility that can be used for testing the receiver separately using keyboard (no hardware clickers needed) and testing the utility to function effectively. This document provides all information about the Python code , its need and interaction with other parts.

1.1.2 GUI Installer :

This document describes the need and functionality of a GUI installer for clicker remote software. And the complete working and design of the installer is clearly explained in this document. And the software which built this installer is briefly explained .

1.1.3 Open Office Integration :

The aim of the openoffice integration module is to implement all the features of openoffice software into the clicker desktop application & vice-versa i.e. implementing all features of clicker desktop application into openoffice software as addon. With this integration Clicker Desktop Application will allow user to use documents of type powerpoint, word, pdf etc.. to be displayed in the clicker software. Also it provides facilities of conducting poll, raisehand, quiz synchronized with these documents. The Addon for openoffice will also provide functionality mentioned above.

1.1.4 Web Based Application :

Web based development to this software will provide an efficient and easier way to extract and manage the software, it will not require any installation and hence is easy to use for the user. The main purpose for its web based development is to provide the user with a suitable environment which is easy to use and manage. It can be access from any remote centers and we have used python for the interaction with the Clicker Device. We use robust database which is very flexible and easy, hence data can be manipulated and managed easily.

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

1.2.1 Python Serial Utility:

Using Python 2.6.1 a utility has been developed as a part of Clicker project to receive data as packets from hardware to the serial port of the computer where the software runs, unpack those packets , read the information , send acknowledgment and start processes according to the request. It is now been developed a s a keyboard utility as well to simplify testing process and as the packets are sent and received they can be seen on “gtkterm” in Ubuntu and “Serial Port Monitor – TI” in Windows.

The code and modules needs change if in future the packets sent and received by hardware(data packet format given by hardware team) changes. Currently it includes processing of the following modules in clicker:

- ClickerConfiguration : must before any other usage
- ClickerAttendance : Making an XML file of students present in class(as per Clicker ID)
- ClickerQuiz : Collecting responses as per the time allotted by user(instructor) for 6 different types of questions.
- ClickerRaiseHand : To store clicker ID's that have queries : Using harware clickers only , not done by keyboard.

1.2.2 GUI Installer:

Installing software from source was mostly pain with Linux/Unix systems in the past. Even with proper software packages, the process of configuring, building and installing is an error-prone task, especially if you want to configure the software in a custom way, you have to look at your documentation. In addition its an uneasy task to keep track of installed files once more and more software packages are installed from source. This installer was developed to address the issues mentioned above. It allows for smooth installation and uninstallation of software from source.

1.2.3 Open Office Integration:

The software is basically handling all clicker hardware devices which will be held by the students. Instructor with a basic knowledge of openoffice will be creating quiz slides conducting quizzes and interacting with the student through the clickers and all the teaching meterials will be presented on the slides.

1.2.4 Web Based Application:

Desktop based version of clicker was dependent on the operating system and the specific softwares, and thus will not allow the clicker to run on all systems but Clicker Web Application is web based and hence does not require any specific Operating system and softwares, it just require a browser, from where we can access it through an url and will be more user friendly and easier to access and process.

Earlier Clicker versions require a specific version of Ubuntu and Netbeans to run on the desktop but this will not be the case in Clicker Web Application, a simple browser which are provided by every operating system will be enough for running in any system. Hence any person sitting at some place can access it and run it without any installation.

1.3 Definitions, Abbreviations

1. lod : Level of Difficulty
2. EJB : Enterprise Java Beans
3. XML : eXtensible Markup Language
4. OOP : Object Oriented Programming
5. LED : Light Emitting Diode
6. NOC : Number of Count
7. RC : Recycle Count
8. USB : Universal Serial Bus
9. OO : Object Oriented
10. ID : Identification number
11. GUI : Graphical user interface
12. JRE : Java run time environment
13. Install Jammer: A multi platform GUI installer.
14. Clicker: A system which is used to collect response from audience
15. Clicker Reciever: Receiver capture responses from a live audience(clicker) using device signal. Receiver using the USB port.

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. The motivation for the project was to provide an interface between student and teacher which help the instructor to conduct the classroom sessions more effectively. Also 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 centre to participate in the classroom session conducted at some other place. As it takes a long time of the class to take attendance or solve individual doubts in class this device was meant to take automated attendance, quizzes of different patterns (single answer correct, Multiple answer correct , T/F , numeric , etc) and a unique feature of raise hand(i.e. student clicks the raise hand button on clicker and his/her ID is stored in a file that's always visible in side screen in the software utility.

As far as internship work modules go the motivation was to add extra features and furnish the few software codes already written. Python Serial Utility was developed and enhanced for better understanding of packet flow and testing of receivers. It lead to automation f various manual works done earlier within python files. GUI installer was developed in order to address the issues like configuring, building and installing packages while installing clicker remote software in user's machine. This installer mainly focuses on dependency problems unlike the earlier modes installation.

Chapter 2

Objective:

The objective of the each assigned task is :

Python Serial Utility: A communication link is needed to transfer data and unpacking and reading information in it. And hence this utility is made and will always run in background process whenever any module of clicker runs. Without this interface the data won't get transferred from the hardware to the software that needs it. Python is the language used for the same. Now a receiver can collect attendance, quiz responses and raise hand from maximum 200 clickers having different IDs to distinguish students.

GUI Installer: The main objective behind this GUI installer is to install the clicker remote software in remote centers. And this installer should be able to install and uninstall the packages like JRE, python-serial as well. And one more thing is that the installer provides wizard based interface throughout the installation. This is so user friendly and easy to use. This installer completely solves the dependency problems. And this installer features an easy uninstallation tool which removes the desktop environment of software completely.

Open Office Integration: Clicker with openoffice creates an ease for instructor to use documents which are defined in well known formats like .ppt, .odp, .pdf. Student just need to know how to use that devices and instructor who is using clicker software just need know how to use openoffice software and have to have knowledge about all features like RaiseHand, SmartSlideShow, Creating Layout, Poll etc.. This software will aid the teacher in its teaching by introducing an ease of interaction as well as an ease to its teaching method by including regularly used openoffice documents with the clicker interaction features.

Web Based Application: The main purpose of this work is to create a web interface such that the instructor uses the complete software as it is without any installation and downloading. Server at IIT-B processes all information, stores it and sends result back to the remote centre. This is the technology which will enable to use this device even in remote locations in India where people don't have technical knowledge needed to run the software properly.

Chapter 3

Design Considerations:

3.1 Assumptions and Dependencies

3.1.1 Python Serial Utility:

Hardware Specifications

The receiver has 3 parts:

- 1.) The Serial Adaptor – To provide power (6V) and communicate with the python code.
- 2.) Voltage Convertor (Serial Interface) – To convert 6V to 3.3V: This will be integrated with main board in coming version.
- 3.) Main Chip Board – Has CC2511 chip on which message protocols are coded and an LED to show if the device works properly or not.
- 4.) The protocol used for communication is defined by the hardware team and are not according to industrial standards.

Assumptions:

- All files which are internally called (in total 14) should be in one package else they won't be accessed.
- LED in hardware blinks once the cables to it are connected properly. If it doesn't then there will be a problem.
- The power convertor takes power from the computer and converts it as per hardware need. This should be checked by multimeter if it gives a potential difference of 3.3V.
- The code runs using keyboard inputs from user (instructor) – the course ID , NOC , RC , No. of clickers(i.e. No. of students) , Quiz number , Quiz type , Question Number , Correct Answer. However for RaiseHand it will be checked with the hardware only as it runs as a constant process to detect Clicker ID's if students press Raisehand button for doubt.
- Check XML files once for clarity, if all ID's are configured there then it is correct. This is just so that a general person doesn't have to see the packet values using “gtkterm” or “Serial Port Monitor”.

3.1.2 GUI Installer:

- In order to run the clicker remote software, we should make a shell script which runs the jar file and this script should be included in Install Jammer while creating the application.
- The installer opens a console window which asks the root password in order to install the specified packages.
- After the installation is done, a desktop shortcut will be created which runs the clicker remote software by a double click.
- If we want to uninstall the clicker remote software, we can do it directly by clicking uninstall shortcut which is created in installation directory in home folder.
- Otherwise go to applications, you will find the clicker remote software application name and uninstall it and we can launch the software by doing the same.

3.1.4 Web Based Application:

The main software used by us to design the complete Clicker Web Application are as follows

- NetBeans IDE 6.9.1
- MySQL Administrator
- MySQL Query Browser
- Mozilla Firefox
- Apache Tomcat 6.0
- JasperSoft iReport Designer 4.0.2
- Python

All these software are Open Source software and are freely available and the Operating System used by us is Ubuntu 9.10, also we have ClickerV2 as the basis for designing the interface of the web pages.

Although we have designed it using these software but it does not depend upon these after completion, it will just use Python for getting XML files and one browser which is freely available in every operating system.

The dependencies of various modules are as follows:

- 1.) After login to the Clicker Web Application the instructor can work with other modules depending upon the requirement.
- 2.) During a classroom session every student has a clicker device and the instructor has the clicker receiver, and every clicker device is associated with a ClickerId, therefore first of all before starting a session in the classroom, the instructor tell every student to start the clicker device by pressing ST button and then the proper communication start between clicker device and clicker receiver.
- 3.) After starting of the communication all the things work properly and the instructor can perform the task required by him, and go through all the module.
- 4.) The communication occur by using the python which gets the XML files and the processing starts.

3.2 General Constraints:

3.2.1 Python Serial Utility :

- While taking attendance and quiz sometimes the hardware stops (i.e. the blinking LED stops or becomes constant red) . If so then restart the computer and connect power USB cable to hardware again.
- Raise Hand module will constantly run once started and you can't do any other process after that. So one needs to run Raise Hand module for testing in the end.
- Sometimes the code doesn't start execution due to unknown prblems showing error like “ser not defined / port not open” . Restart and reconnect all cables again to fix this.

3.2.2 GUI Installer:

- JRE installation is configured in such a way that the installer installs JRE if it was not installed earlier, otherwise it just skips the installation of JRE.
- Environment variables are added in the Install Jammer, there by the installation of JRE will be done in home folder by default.
- The installation directory will be created in the home folder by default.
- For Python installation, the working directory is Temp where one installation directory will be created during the installation and it will be deleted automatically after the installation is done.
- If JRE is already is there in the machine, then the welcome wizard shows like Java: 1 otherwise it shows like Java :0
- The environment variable for JRE is set as
JAVA_HOME: /home/jre1.6.0_22/

\$PATH:/home/jre1.6.0_22/bin

- JRE installation will be done home directory and python-serial package will be installed in root directories.
- While uninstallation, the installer opens a terminal which asks for root password to get it uninstalled.

3.2.3 Open Office Integration:

Clicker desktop application need jdk-jre-6, and open office libraries to be installed on machine. Also this entire integration is possible in ubuntu 9.04 or newer version. It is compatible with ubuntu 8.10 but some features does not works properly in it. In addon version of Clicker software machine needs to contain jdk-jre-6, openoffice-java-common or libreoffice-java-coomon and openoffice 3.3 and later versions and libreoffice should be installed on the user's machine.

Softwares needed:

- Openoffice3.3(or higher version) or Libreoffice
- Openjdk-6-headless
- Jre-6
- Openoffice-java-common or Libreoffice-java-coomon

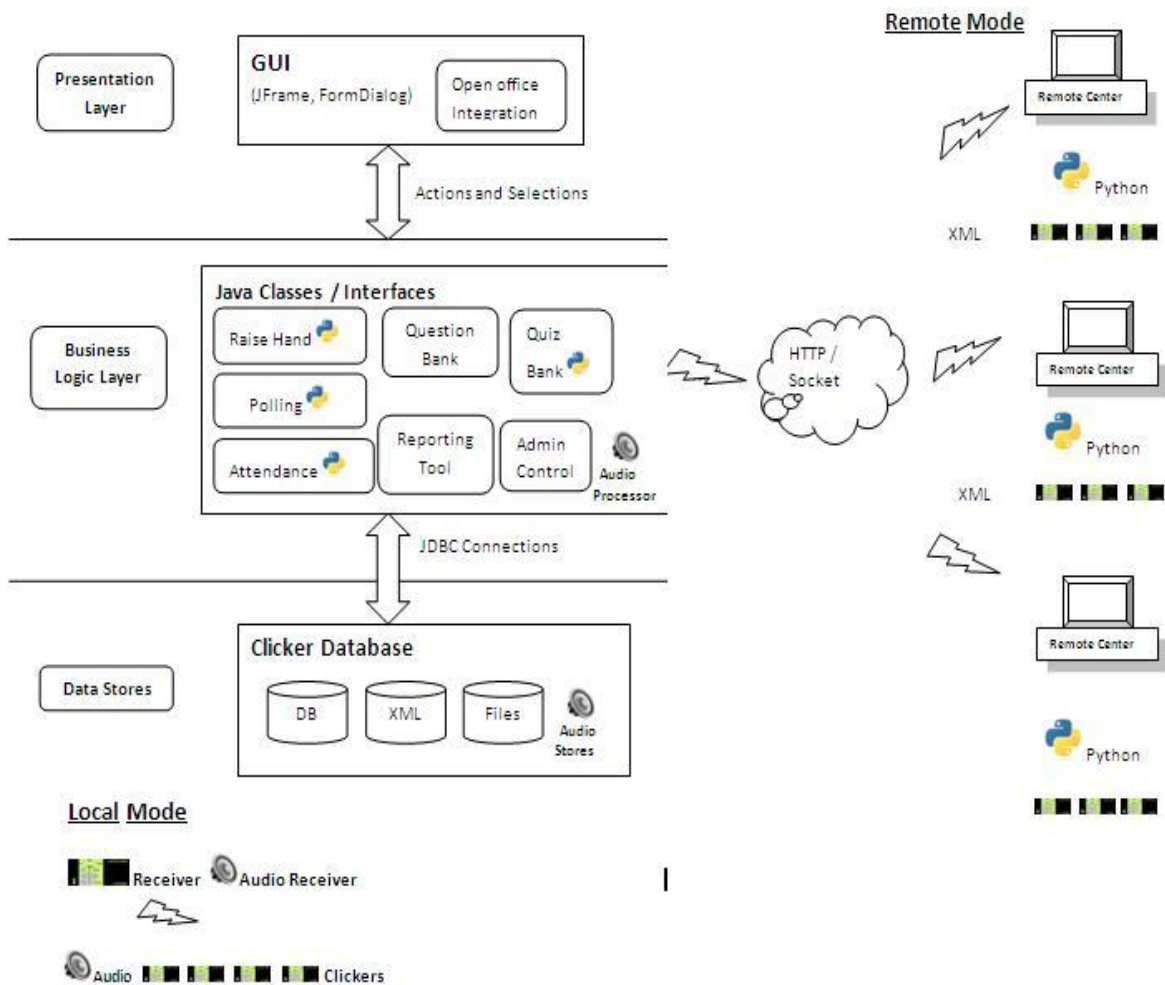
3.2.4 Web Based Application

- The Clicker Web Application has some General Constraints or limitation, which includes the following and these are necessary for successful processing of the software:
- It requires the clicker hardware i.e the Clicker Reciever and the Clicker device to communicate among themselves and for getting reponses from the students.
- It also require Python Software to get XML files and start communication.
- It require an internet Browser to fetch Clicker Web Application through url at remote centers.
- For using Clicker Web Application a classroom environment is required to get real time responses.

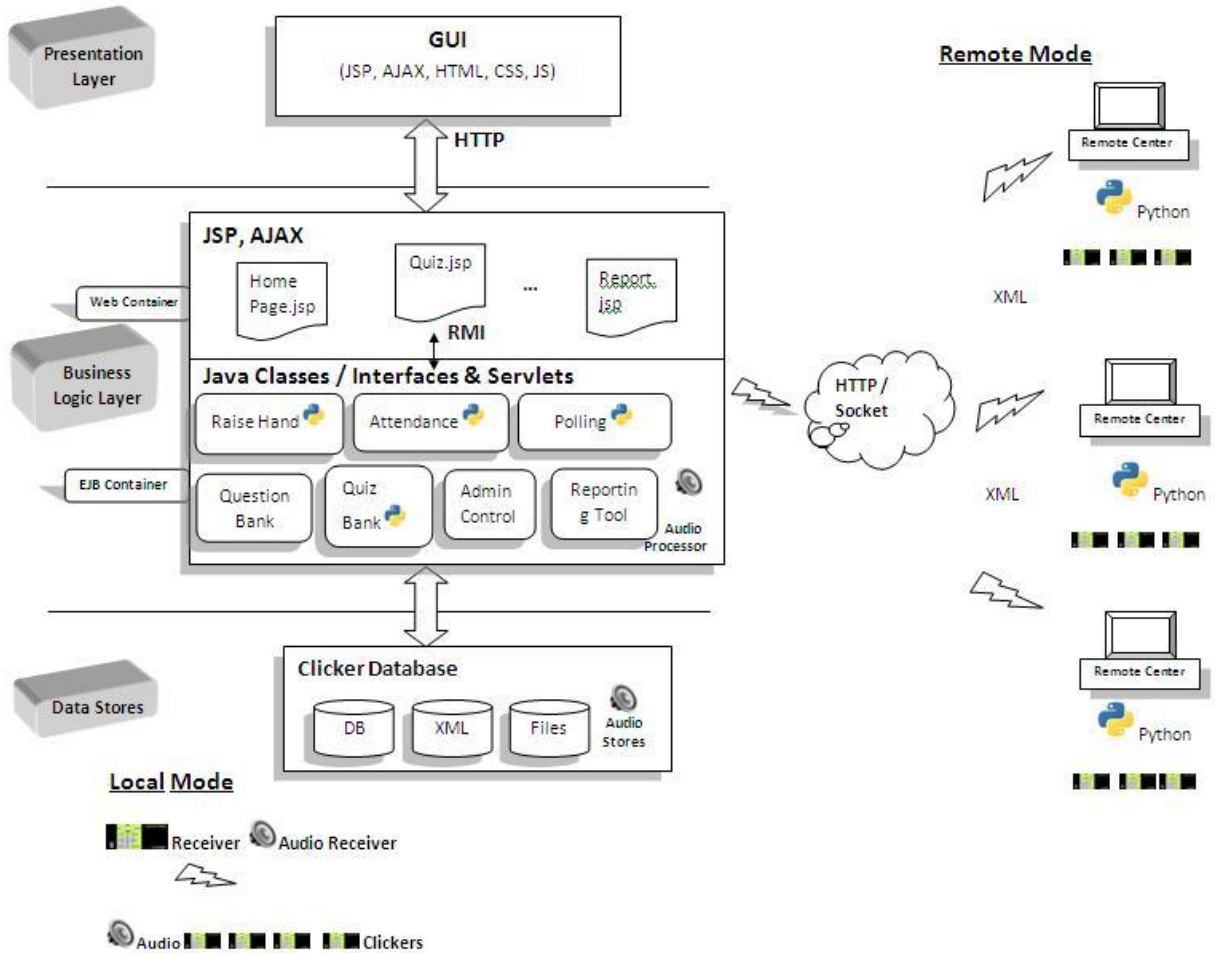
Chapter4

Architecture

4.1 Clicker Desktop Application



4.2 Clicker Web Application



When describing the architecture of the Clicker Web Application software, it has two mode namely Local mode and Remote mode, where the local mode can be viewed as a classroom environment and the remote mode are the remote centers where the utility program works and there are about 34-36 remote center of clicker in India from where the Clickers are managed and its main center is situated is situated at IIT Bombay from where it is managed.

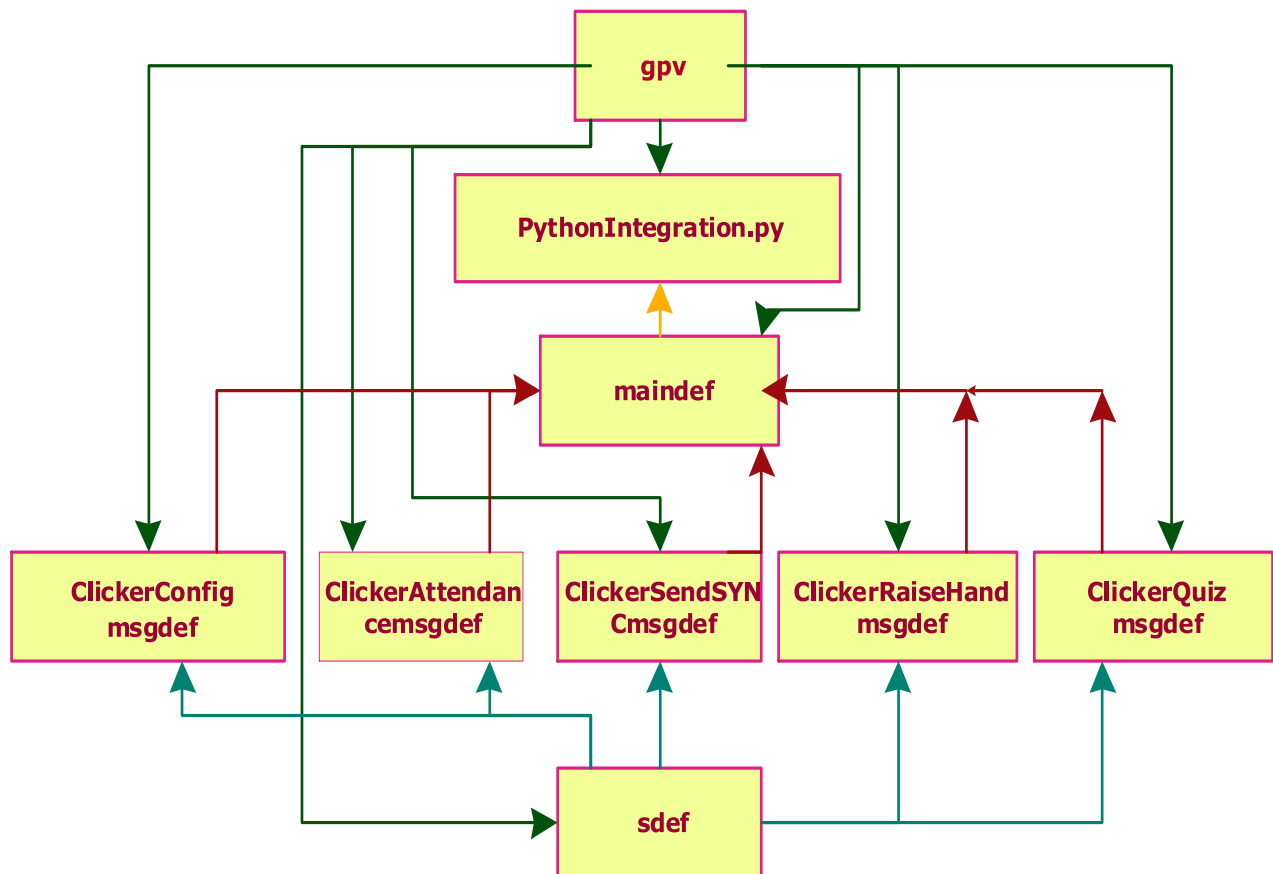
Now moving on to its interior architecture, the user interact with the GUI interfaces as shown in the Architectural Strategies section which are developed using HTML, JSP, Ajax, CSS and JavaScript and it is the presentation layer of the design which interact with the Business Logic layer through http which has the web container and EJB container, web container contains all the jsp and ajax pages while EJB container has the Java classes , interfaces and servlets which together forms the Business Logic layer and this interact with the clicker database which contains all the Xml and other files required and all these forms the Local mode and these system at this mode interact with the Remote mode through

HTTP/Socket and the remote center through python sends the Xml files and the connection get established and communication starts.

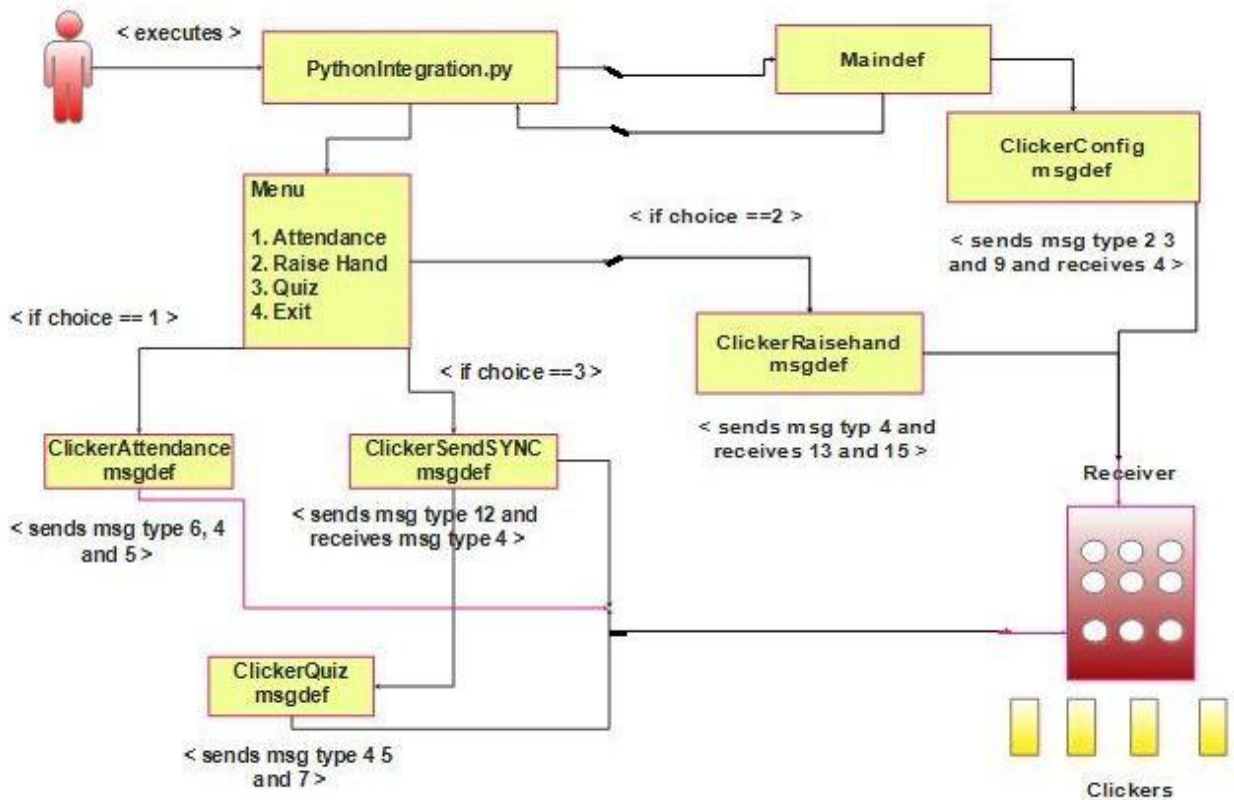
Chapter 5

Design and Implementation of Python Serial Utility

Code Flow Diagram



Usecase Diagram



Detailed Description

General:

- Compiled the modules as a single package having XML files , Shell script , text files and python modules. Hence it is now more OO format.
- Having a user(instructor) driven menu for choosing between attendance, quiz, raise hand, receiver flash after clicker configuration.
 - Documented the whole python process by commenting the functionalities in the code so that its easy for user to understand.
 - Generated XML file for the user inputs (Clicker ID , Attendance ,Quiz responses) and implemented an internal timer for the quiz and raise hand functionality.
- Testing the utility with maximum 200 clickers for configuration , attendance and quiz solving many problems in the way.

Details of each module and file in package:

- Open project in netbeans IDE 6.7.1
- Package location : /home/Desktop/Pythonintegration/src
- You can locate the package anywhere you wish and open it in netbeans.
- Following is the point algorithm of how the clicker works and the code is written and modularised as per this.

Clicker Function Algorithm:

1. Start
 2. Configure ClickerIDs to Receiver
 3. Collect Attendance (store attendance in xml file)
 4. Send SYNC signal
 5. Collect Quiz responses (any quiz type)
 6. Send Acknowledgment for Answer (store it in xml file)
 7. Raise Hand
 8. Collect Raise hand (store ID's in a text file)
 9. Stop
- Package contains 14 files and corresponding bytecode files for 9 python files(.pyc)

1.) **gpv.py** – This file imports all python libraries needed for execution . The libraries are : sys , os , serial , time , array, struct , threading , xml.dom.minidom.

2.) **sdef.py** – This file has all the secondary definitions, which are used by various modules. It imports gpv.py. These definitions include : open_ser_port() , getText() , intchksum() , onescomp() , twoscomp() , CHKSUMbitbinstr() , getClickerMessageTwoCompCHKSUM() , getNByteRepr() , StringtoASCIIArray() , ClickerACKchksumReturn() , make_xml_config() , getValueByFourByteHex() , ClickerXMLElement() , ClickerPacketsCountArray().

3.) **ClickerConfigmsgdef.py** – It contains all functions needed for packet sending and receiving during configuration and storing available clicker ID's in “clickerID.xml” . It imports sdef , gpv. It has functions like: ClickerMsgTyp2send() , ClickerResendMsgTyp2() , ClickerMsgTyp3send() , ClickerACKMsgTyp4rec() , ClickerACKMsgTyp4NOCRCrec() , ClickerNOCRCMsgTyp9send() .

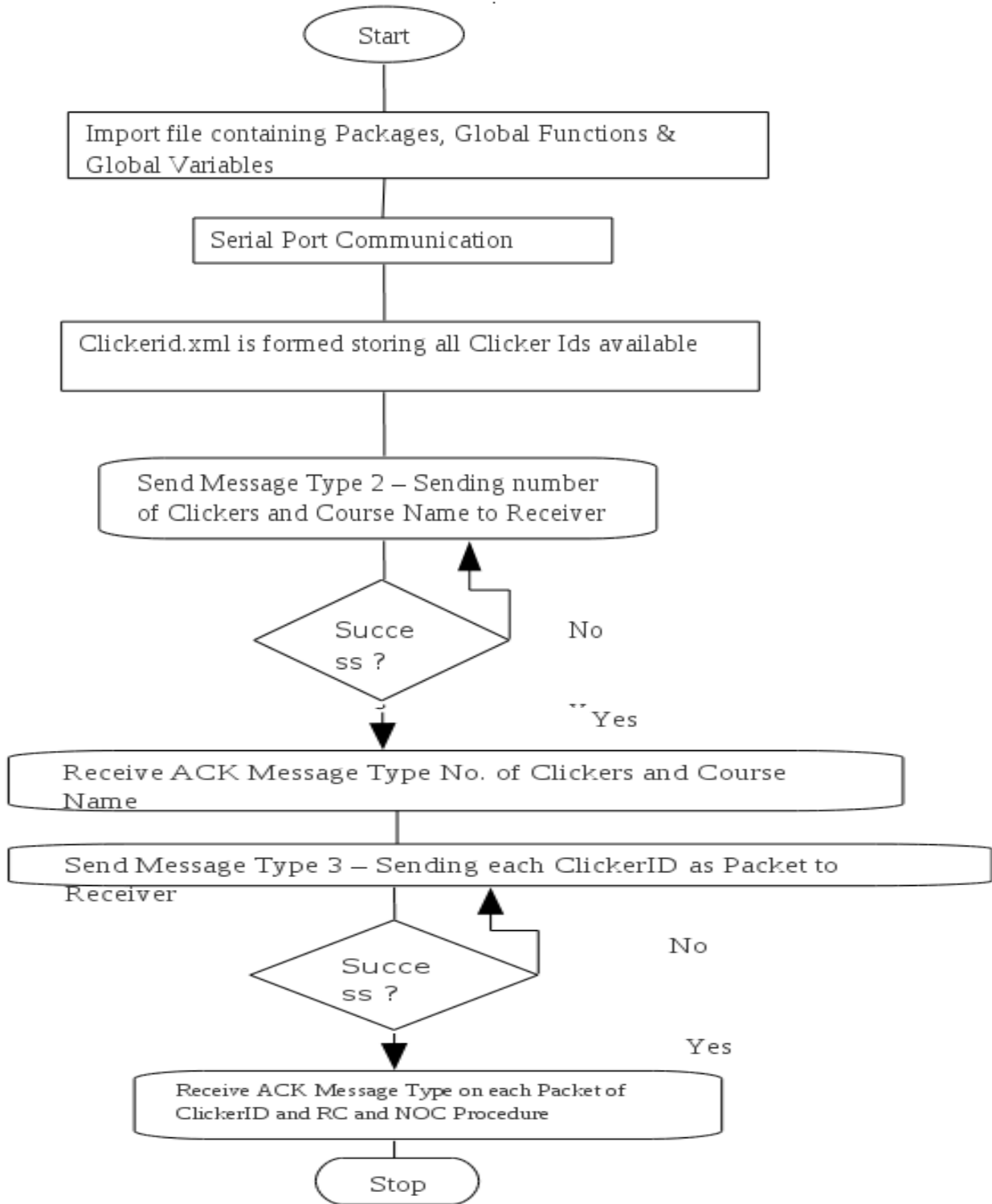
- 4.) **ClickerAttendancemsgdef.py** - It contains all functions needed for packet sending and receiving while collecting attendance of students and storing it in an XML file "clicker_xml.xml". It imports sdef, gpv. It has functions like: ClickerMsgTyp6send(), ClickerACKMsgTyp4send(), ClickerErrorMsgTyp5send(), Clickermsgtyp6Resend().
- 5.) **ClickerSendSYNCmsgdef.py** - It contains functions needed to initialize the receiver to collect quiz responses for 6 different types of quizzes which are:

Quiz type	Explanation
1	General - Single answer correct
2	Multiple - Multiple answer correct
3	Numeric answers(0-9)
4	Yes/No type
5	True/False type

It imports sdef, gpv and has functions like:
ClickerMsgTyp12Send(), ClickerACKMsgTyp4SYNCreco()

- 6.) **ClickerQuiz.py** - It has all the functions to receive responses of different types and send acknowledgment of correct answer or not. It imports sdef, gpv. It has functions like: ClickerMsgTyp7send(), ClickerACKMsgTyp4send(), ClickerErrorMsgTyp5send(), Clickermsgtyp7Resend().
- 7.) **ClickerRaiseHandmsgdef.py()** - It contains functions needed to continuously run Raise Hand module and keep storing ID's that press Raise Hand button in a text file. It imports sdef, gpv. It has functions like: ClickerACKMsgTyp4send(), ClickerMsgTyp13orMsgTyp15rec(), ClickerMsgTyp13rec().
- 8.) **Maindef.py** - This file contains functions which are actually calling the functions in the previous python files as per the algorithm shown.

Step1. Clicker Configuration Algorithm:



1. Start
2. Import other python files needed
3. Global Variables except in loop
4. Serial Port Communication
5. Clickerid.xml is generated as per number of clickers given by user(instructor)
6. Send Message Type 2 – Sending number of Clickers (Maximum 200) and Course Name (Maximum 6 Chars) to Receiver
7. If success then
Receive ACK Message Type for No. of Clickers and Course Name
Else
If error then
Send Message Type 2 again till receive
8. Send Message Type 3 – Sending each ClickerID as Packet to Receiver
9. If success then
Receive ACK Message Type on each Packet of ClickerID
NOC & RC Procedure
Else
If error then
Send Message Type 3 again till receive
10. Stop

Send Message Type 2 – Sending number of Clickers (Maximum 200) and Course Name (Maximum 6 Chars) to Receiver

SOF 1	SOF 2	HL	MSG TYPE	SRC ID	DST ID	Reserv ed	MSG LENGTH	PKT INDEX	MSG BODY	CHK
1	1	1	1	2	2	4	1	1	16	2
:)	29	2	0	0	0	8	0	No. of Clickers and Course Name	2'c

Example (Message Type 2):

If No. of Clickers is 8 and Hexadecimal Bit Value of Course Name is 31 32 33 34 35 36
3A 29 1D 02 00 00 00 00 00 00 00 00 08 00 00 10 31 32 33 34 35 36 00 00 00 00 00 00
00 2'c

Receive ACK Message Type for No. 4 of Clickers and Course Name

SOF 1	SOF 2	HL	MSG TYPE	SRC ID	DST ID	Reserved	MSG LENGTH	PKT INDEX	MSG BODY	CHK
1	1	1	1	2	2	4	1	1	16	2
:)	29	4	0	0	0	0	0	0	2'c

Example (Message Type 4):

3A 29 1D 04 00 00 00 00 00 00 00 00 00 00 00 16ZEROS 2'c

Send Message Type 3 – Sending each ClickerID as Packet to Receiver

SOF 1	SOF 2	HL	MSG TYPE	SRC ID	DST ID	Reserved	MSG LENGTH	PKT INDEX	MSG BODY	CHK
1	1	1	1	2	2	4	1	1	16	2
:)	29	3	0	0	0	16	1	2 Byte for each ClickerID	2'c

**Example (Message Type 3):
If Clicker IDs are 8**

3A 29 1D 03 00 00 00 00 00 00 00 00 08 01 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 2'c

Receive ACK Message Type on each Packet of ClickerID same as Message Type 4

NOC & RC Procedure

SOF 1	SOF 2	HL	MSG TYPE	SRC ID	DST ID	Reserved	MSG LENGTH	PKT INDEX	MSG BODY	CHK
1	1	1	1	2	2	4	1	1	16	2
:)	29	9	0	0	0	02	0	2 bits for NOC & RC	2'c

Example (Message Type 9):

3A 29 1D 09 00 00 00 00 00 00 00 02 00 01 01 (14 ZEROS) 2'c

Total Messages in this Step

Message Types	Status on Serial
2	Send
4	Receive
3	Send
4	Receive
9	Send
4	Receive

Step 2. Clicker Attendance

1. Global Variables except in loop
2. Serial Port Communication
 3. Generating Clicker_xml.xml with Clicker ID of all detected hardwares.
 3. Send Attendance Message Type 6
 4. Receive ACK Procedure & Error Procedure
 5. Receive Attendance Response Type 8 on each packet or Error Procedure
 6. Send ACK Procedure
 7. After success Prepare Clicker Response XML File
 8. Stop

Send Attendance Message Type 6

SOF 1	SOF 2	HL	MSG TYPE	SRC ID	DST ID	Reser ved	MSG LENGTH	PKT INDEX	MSG BODY	CHK
1	1	1	1	2	2	4	1	1	16	2
:)	29	06	0	0	0	0	0	0	2'C

Example (Message Type 6):

3A29 1D 06 0000 0000 00000000 00 0000000000000000 2'c

Receive Attendance Response Type 8 on each packet

Suppose Packet Index is 1
 Clickers are 1 to 8
 Response is Z

SOF 1	SOF 2	HL	MSG TYPE	SRC ID	DST ID	Reserved	MSG LENGTH	PKT INDEX	MSG BODY	CHK
1	1	1	1	2	2	4	1	1	16	2
:)	53	08	0	0	0	40	1	Clicker IDZCo untTi mer	2'c

Example (Message Type 8):

```
3A29 35 08 0000 0000 00000000 28 01 0001900000 0002900000 0003900000
0004900000 0005900000 0006900000 0007900000 0008900000 chksum
```

Total Messages in this Step

Message Types	Status on Serial
6	Send
4	Receive
8	Receive
5	Send
4	Send

Step 3. Send SYNC

- Global Variables except in loop
- Serial Port Communication
- Send Msg Type 12 as SYNC
- If success ACK
Else
Send Msg Type 12
- Stop

Send Msg Type 12 as SYNC

SOF 1	SOF 2	HL	MSG TYPE	SRC ID	DST ID	Reserved	MSG LENGTH	PKT INDEX	MSG BODY	CHK
1	1	1	1	2	2	4	1	1	16	2
:)	29	12	0	0000	0	10	0	2=Question Time 1=Question Number 1 = Question Type 1 = Date 1 = Month 2 = Year in YYYY 1 = Hour 1 = Minute	2'c

Example (Message Type 12): (1st 3 parameters given by user)

Suppose the contents of message body are,

Question Time =60 Seconds

Question Number =1

Question Type = 1

Date= 5

Month = 12

Year = 2010
 Hour = 10
 Minute = 35

3A29 1D 0C 0000 0000 00000000 0A 00 00 3C 01 01 05 12 07DA 10 35

Total Messages in this Step

Message Types	Status on Serial
12	Send
4	Receive

Step4. Collect Quiz Responses

1. Called after Send Sync
2. Global Variables except in loop
3. Serial Port Communication
4. Send Quiz Collect Response Request Message as Message Type 7
5. Receive ACK Otherwise send Message Type 7 again
6. Receive Message Type 8 as Quiz Response having certain conditions depending on Question Type on each Packet
7. Check Packet Index and Checksum of each Packet
8. If error, then send Error Message, Otherwise send ACK
9. After success Prepare Clicker Response XML File
10. Stop

Send Quiz Collect Response Request Message as Message Type 7

SOF1	SOF 2	HL	MSG TYPE	SRC ID	DST ID	Reserv ed	MSG LENGTH	PKT INDEX	MSG BODY	CHK
1	1	1	1	2	2	4	1	1	16	2
:)	29	07	0	0	0	0	0	0	2'C

Example (Message Type 7):

3A29 1D 07 0000 0000 00000000 00 0000000000000000 2'c

Receive Message Type 8 as Quiz Response having certain conditions depending on Question Type on each Packet

Types of Question = 4

NATURE OF QUESTIONS = GENERAL, MULTIPLE, NUMERIC, T/F

Suppose Packet Index is 1

CLICKERS ARE 1 TO 8

RESPONSES FOR PARTICULAR QUIZ = A

Now Depending on Question Type, Quiz Responses will be as follows:

Question Type 1 = General

SOF 1	SOF 2	HL	MSG TYPE	SRC ID	DST ID	RESERVED	MSG LENGTH	PKT INDEX	MSG BODY	CHK
1	1	1	1	2	2	4	1	1	40	2
:)	53	8	00	00	00000		1	CLICKERID+RESPONSE+COUNT+TIME	2'C

Example (Message Type 8):

3A29 35 08 0000 0000 00000000 28 01 0001410000 0002410000 0003410000
0004410000 0005410000 0006410000 0007410000 0008410000 CHKSUM

Question Type 2 = Multiple

RESPONSES FOR PARTICULAR QUIZ = A, B

SOF 1	SOF 2	HL	MSG TYPE	SRC ID	DST ID	RESERVED	MSG LENGTH	PKT INDEX	MSG BODY	CHK
1	1	1	1	2	2	4	1	1	80	2
:)	93	8	0	0	0	80	1	CLICKERID+ALL RESPONSES	

									ES + COUNTI MER	
--	--	--	--	--	--	--	--	--	-----------------------	--

Example (Message Type 8):

3A29 5D 08 0000 0000 00000000 50 01 00014142909090900000
0002414290909090000000003414290909090000000044142909090900000
0005414290909090000000006414290909090000000074142909090900000
00084142909090900000 CHKSUM

Question Type 3 = Numeric

RESPONSES FOR PARTICULAR QUIZ = ABCDEF

SOF 1	SOF 2	HL	MSG TYPE	SRC ID	DST ID	RESERVED	MSG LENGTH	PKT INDEX	MSG BODY	CHK
1	1	1	1	2	2	4	1	1	16	2
:)	93	08	0	0	0	80	1	CLICKERI D+ALL RESPONS ES + COUNTI MER	

Example (Message Type 8):

3A29 5D 08 0000 0000 00000000 50 01 00014142434445460000
00024142434445460000 00034142434445460000 00044142434445460000
00054142434445460000 00064142434445460000 00074142434445460000
00084142434445460000 checksum

Question Type 4 = True/False

RESPONSES FOR PARTICULAR QUIZ = 01 OR 00

SAME AS QUESTION TYPE 1

Question Type 5 = Yes/No

RESPONSES FOR PARTICULAR QUIZ = 01 OR 00

SAME AS QUESTION TYPE 1

TOTAL MESSAGES IN THIS STEP

Message Types	STATUS ON SERIAL
7	SEND
4	RECEIVE
8	RECEIVE
5	SEND
4	SEND

Step 5. Raise Hand (Student Doubt Identification)

1. Global Variables except in loop
2. Serial Port Communication
3. Receive Raise hand message type 13
4. Store clicker ID in text file
5. Send ACK message type 4
6. Stop

RECEIVE RAISE HAND MESSAGE TYPE 13

SOF1	SOF 2	HL	MSG TYPE	SRC ID	DST ID	RESERV ED	MSG LENGTH	PKT INDEX	MSG BODY	CHK
1	1	1	1	2	2	4	1	1	16	2
:)	29	13	0	0	0	0	0	0	2'C

Example (Message Type 13):

3A29 1D 0D 0000 0000 00000000 00 00 0000000000000000 CHKSUM

TOTAL MESSAGES IN THIS STEP

Message Types	STATUS ON SERIAL
13	RECEIVE
4	SEND

CHAPTER 6

DESIGN AND IMPLEMENTATION OF GUI INSTALLER

Manual installation of clicker remote software

Before developing this installer, there are two modes of installation for packages required for Clicker Remote software.

1. Using Internet(Online mode) with sudo commands or Synaptic Package Manager.
2. Using .deb package(Offline mode) direct installation.

4.1.1 Using Internet (Online Mode) with sudo Commands or Synaptic Package Manager

2. Only one Computer System is required for clicker remote software Installation
3. At this installation mode following conditions must be fulfilled.
4. A Computer System with Internet facility as on Ubuntu Operating System must be present
5. **Do not use Proxy Server Settings of Internet Facility.**
6. The Updated Ubuntu Operating System with updated packages & repositories must be present
7. Following basic command must be executed on Terminal like
8. `sudo apt-get install <Setup File Name>`

Basically this is very simplest method of installation. If we are familiar with sudo commands we can enter the following commands to install required packages.

9. By sudo Commands

Open the Terminal from Accessories from Applications Menu & enter commands as below,

- ✓ **For Java** : `sudo apt-get install sun-java6-jre`
- ✓ **For Python** : `sudo apt-get install python-serial`

10. By using Synaptic Package Manager

Any Terminal Window, Application and other window must not be opened, so close all Windows.

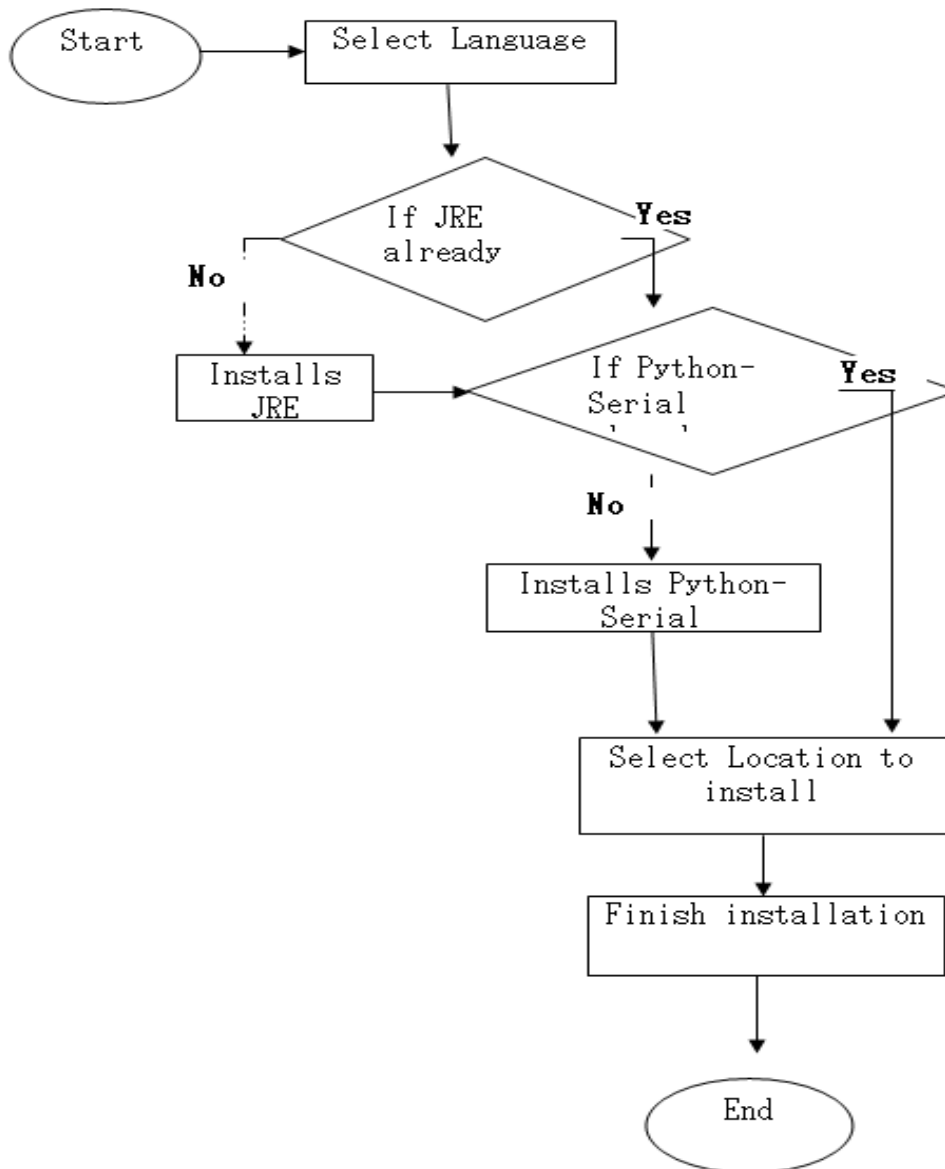
Follow below steps:

1. Open the **System** Menu from Top Panel
2. Select **Administration**
3. Click **Synaptic Package Manager**
4. Now Give your Password of root or user

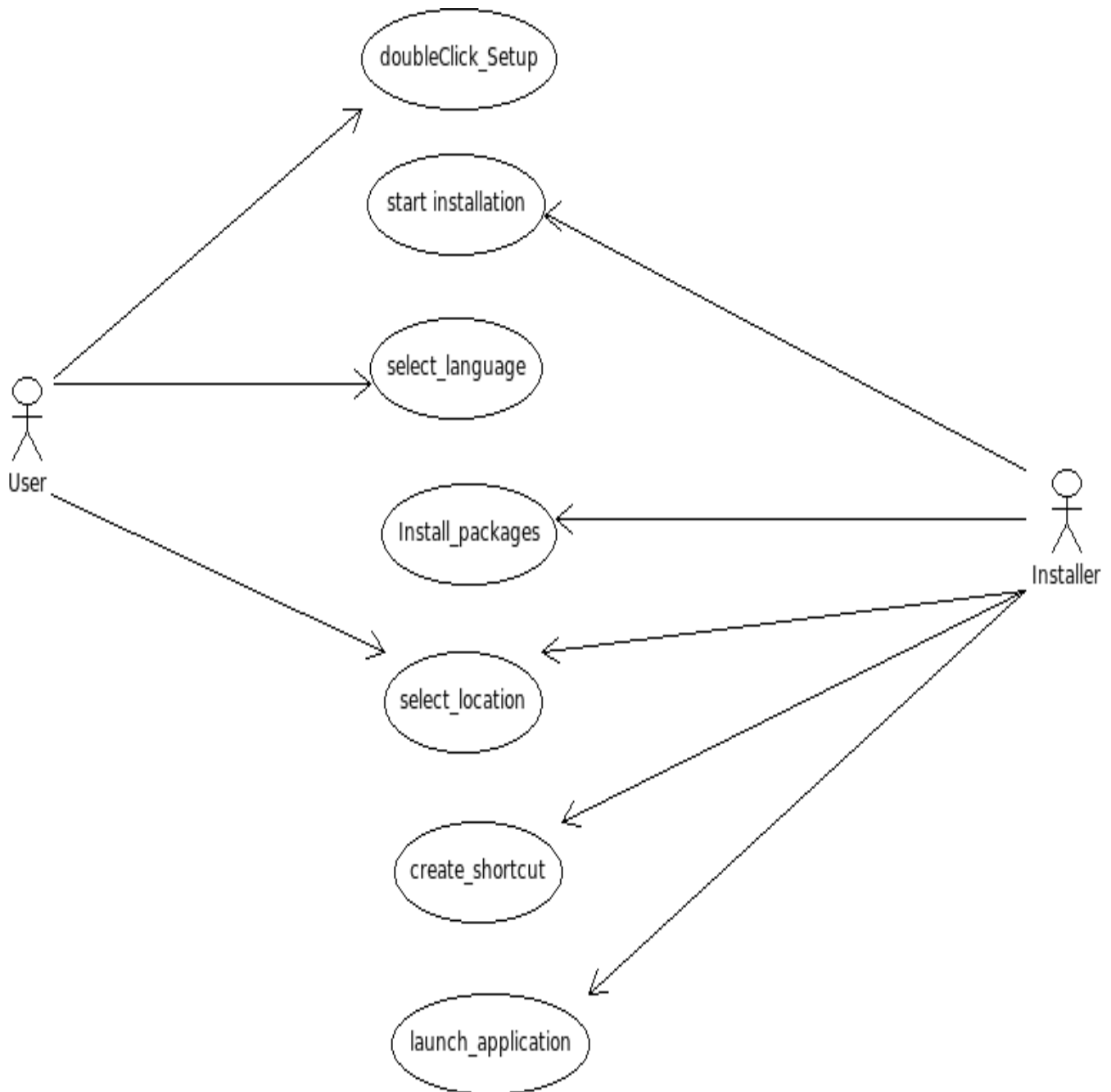
5. Once Password given, the Window of Synaptic Package Manager will open
6. Now Click on Search Tab or Search sub menu from Edit Menu directly or press key CTRL + F for searching required packages
7. Now give Search key as
8. For Java, **sun-java6-jre**
9. For Python, **python-serial**
10. Select option for Searching as Look in **Name**
11. Then click on **Search**
12. Respective application package will list out in Package Description List
13. Select required package & right click on that or select **Package** Menu from Menu Bar
14. Now click on Mark for Installation
15. Then click on **Apply** from Toolbar or click on **Apply Marked Changes** from **Edit** Menu or press **CTRL+P**
16. Further required installation packages for selected package will list out on another window then again click on **Apply** Button on same Window.
17. Hence required packages will be installed through GUI using Synaptic Package Manager.

In order to install clicker remote software in remote centers, we too need a GUI installer which makes the installation task very easier.

Flow Chart for GUI Installer



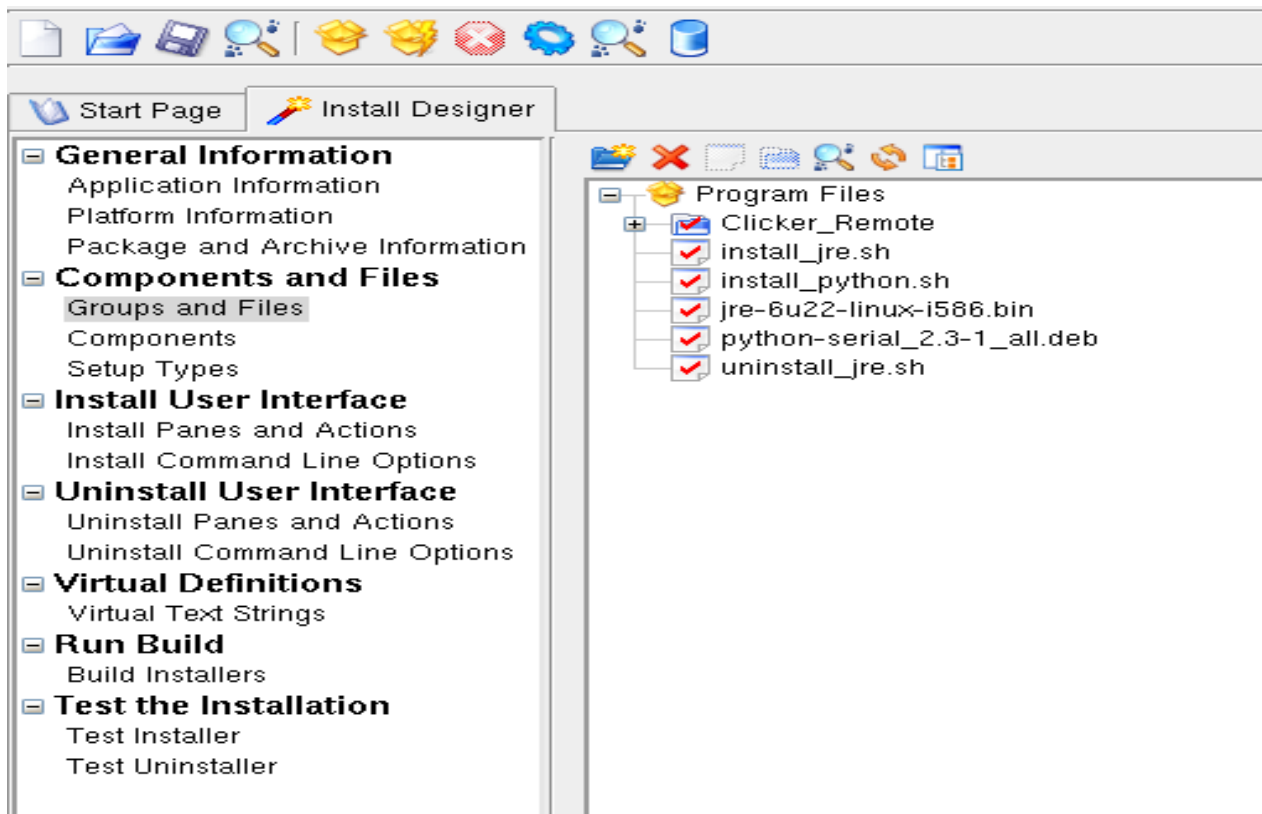
Use Case Diagram



Implementation

Including files

While making the GUI installer, after specification of all application details, we have to include program files as shown below. All program files, shell scripts required for the application should be included in groups and files. And we have to configure the details of every file in standard properties.

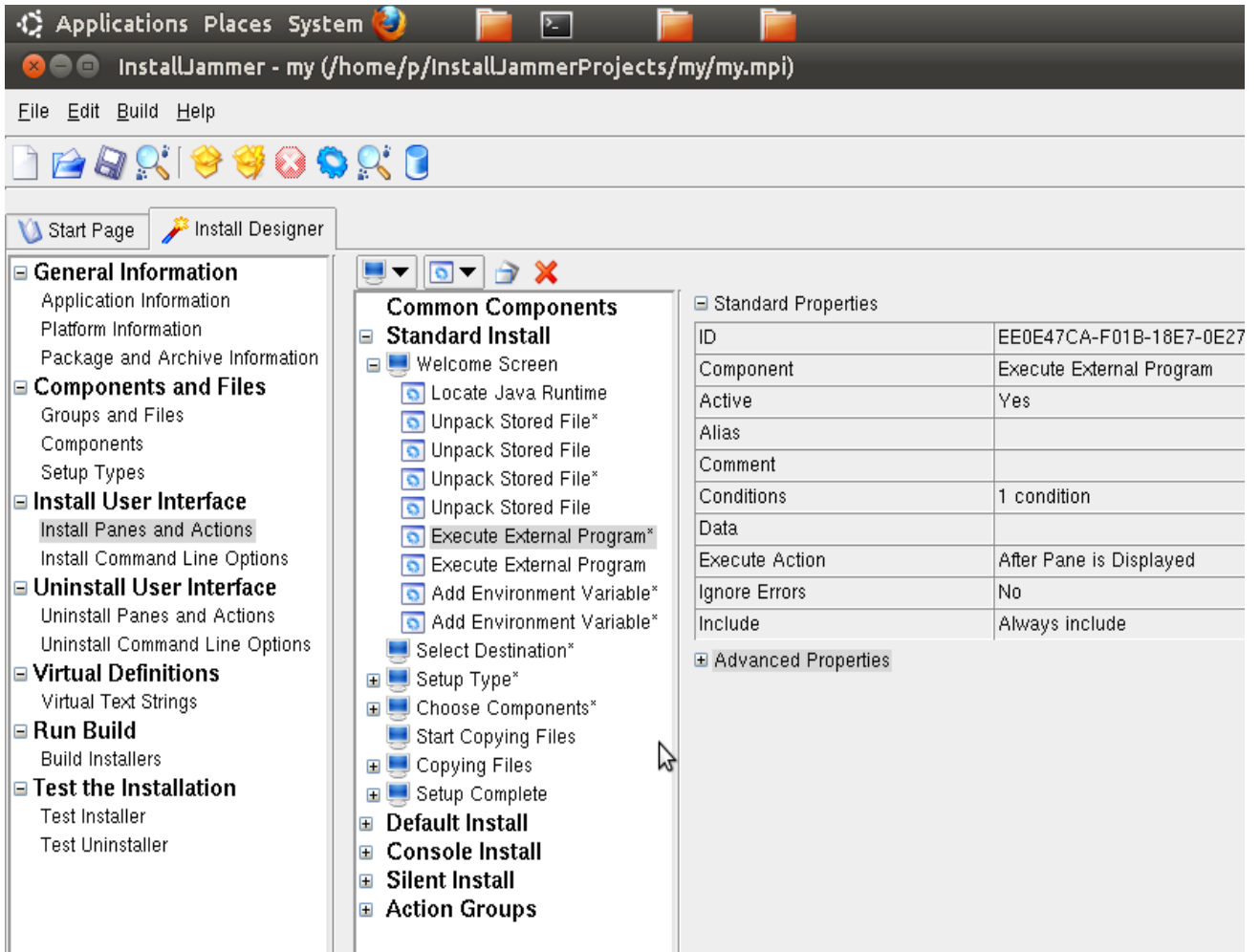


Designing user Interface

After inclusion of program files, we have to design the user interface.

- The first step in this process is unpacking the included program files in a certain directory
For clicker remote software, the unpacking is done for JRE and python files.
- The second step is configuring standard properties for execute external program actions.
- And add environment variables in order to install JRE in a specific location.

- Later add actions for environment variables.
- Those complete things were shown in the below screen shot.

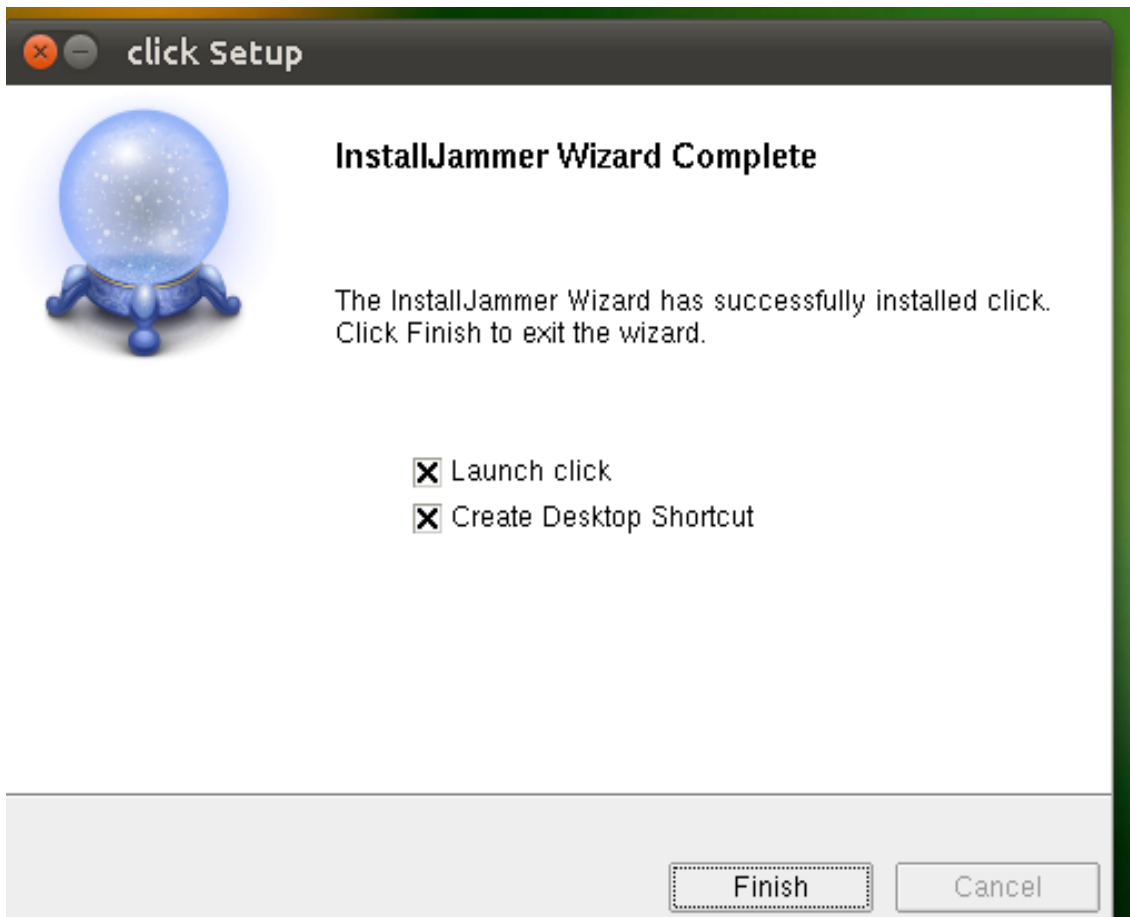


- Configure properties for uninstallation.
- The final step is test the installer in debugging mode and build it through run build option which is shown in the above screen shot and you will find a set up file in Install Jammer projects and run it, at the last step you will find a final wizard as shown in below screen shot.

Launching the application

After all the installation process is over, the Install Jammer shows the following wizard.

- After clicking the finish button, the installer creates a shortcut on the desktop and launches the application as well.
- If we don't want to launch the application, just uncheck launch click checkbox. Then the installer creates desktop shortcut only. The same condition will be applicable to



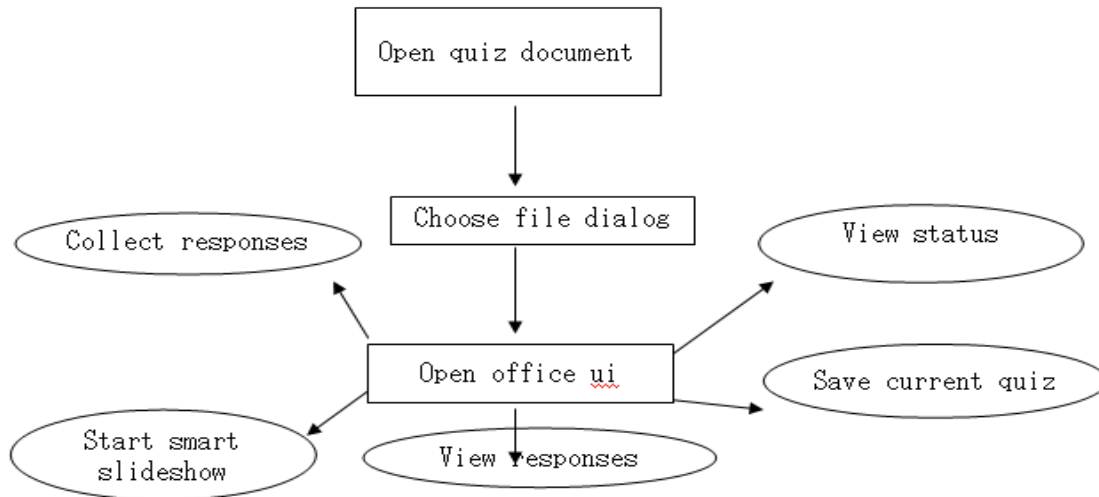
sho
rtcu
t
too.

Chapter 7

Design and Implementation of Open Office Integration:

Data Flow Diagrams

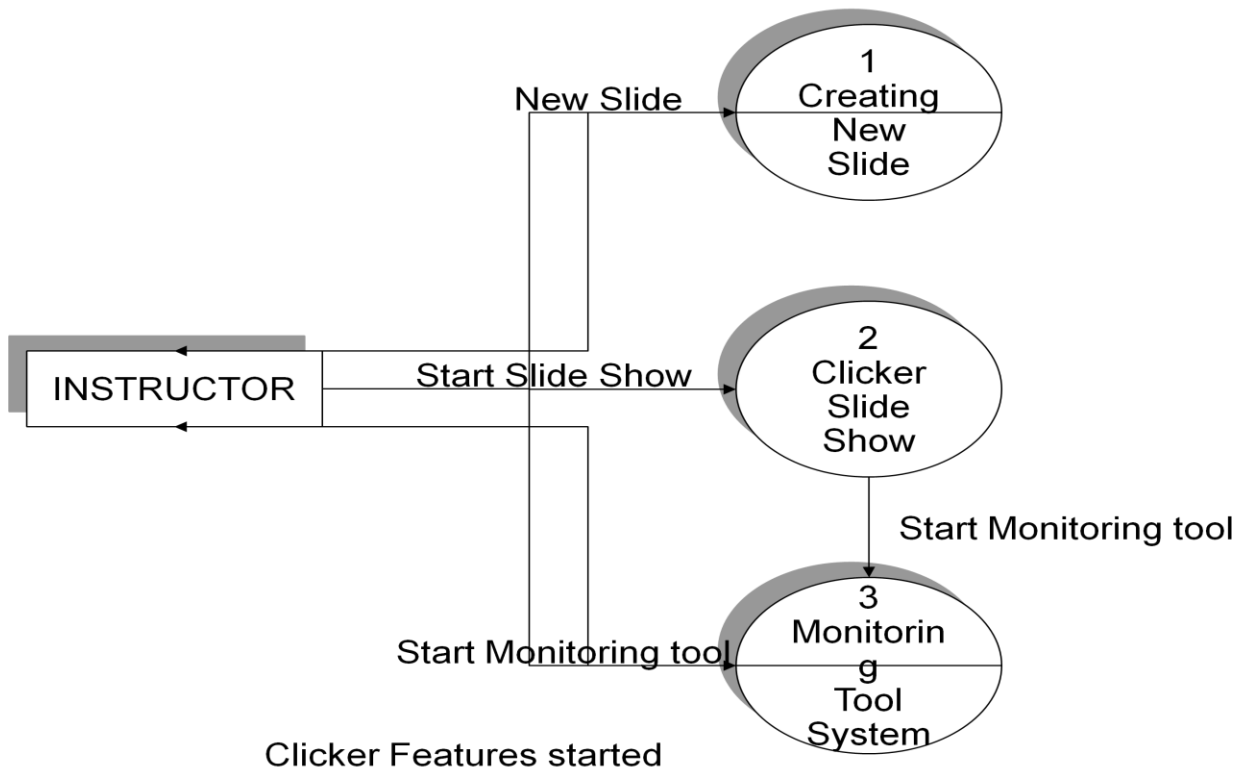
Clicker desktop application

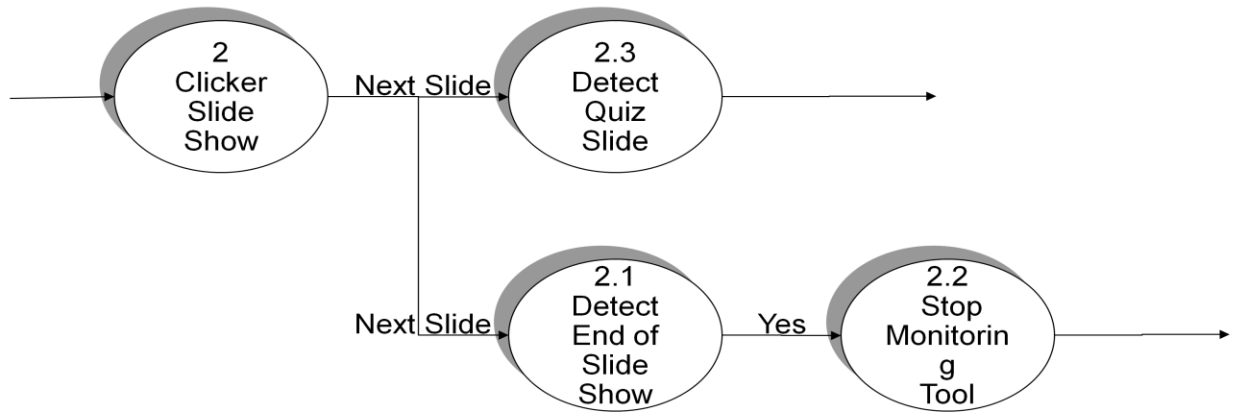
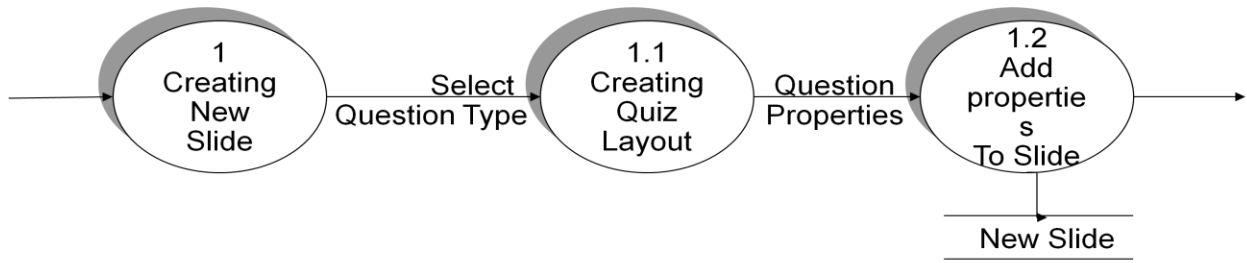


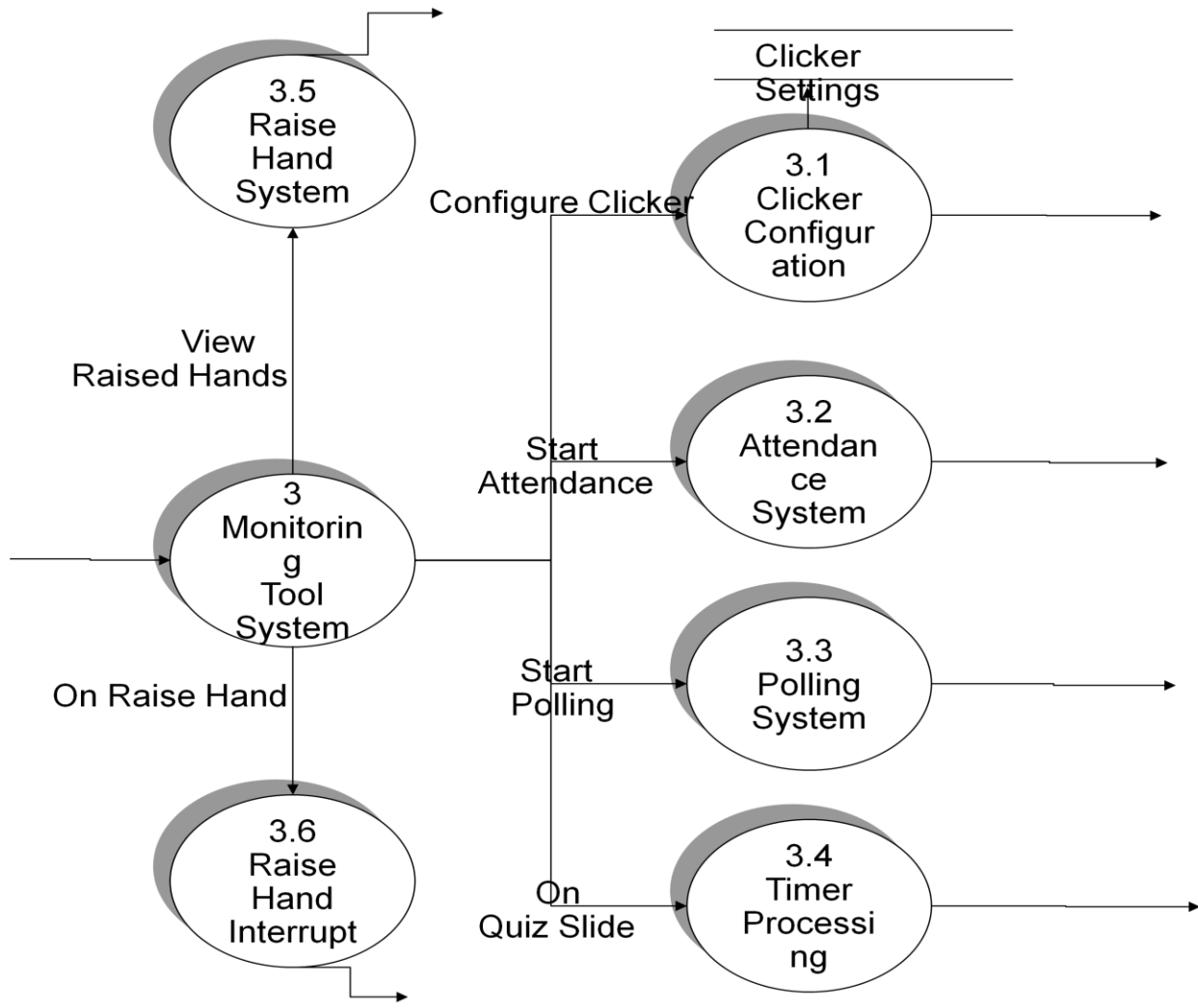
This is UI architecture for the clicker desktop application .All this functionalities are present in quiz module of clicker desktop .

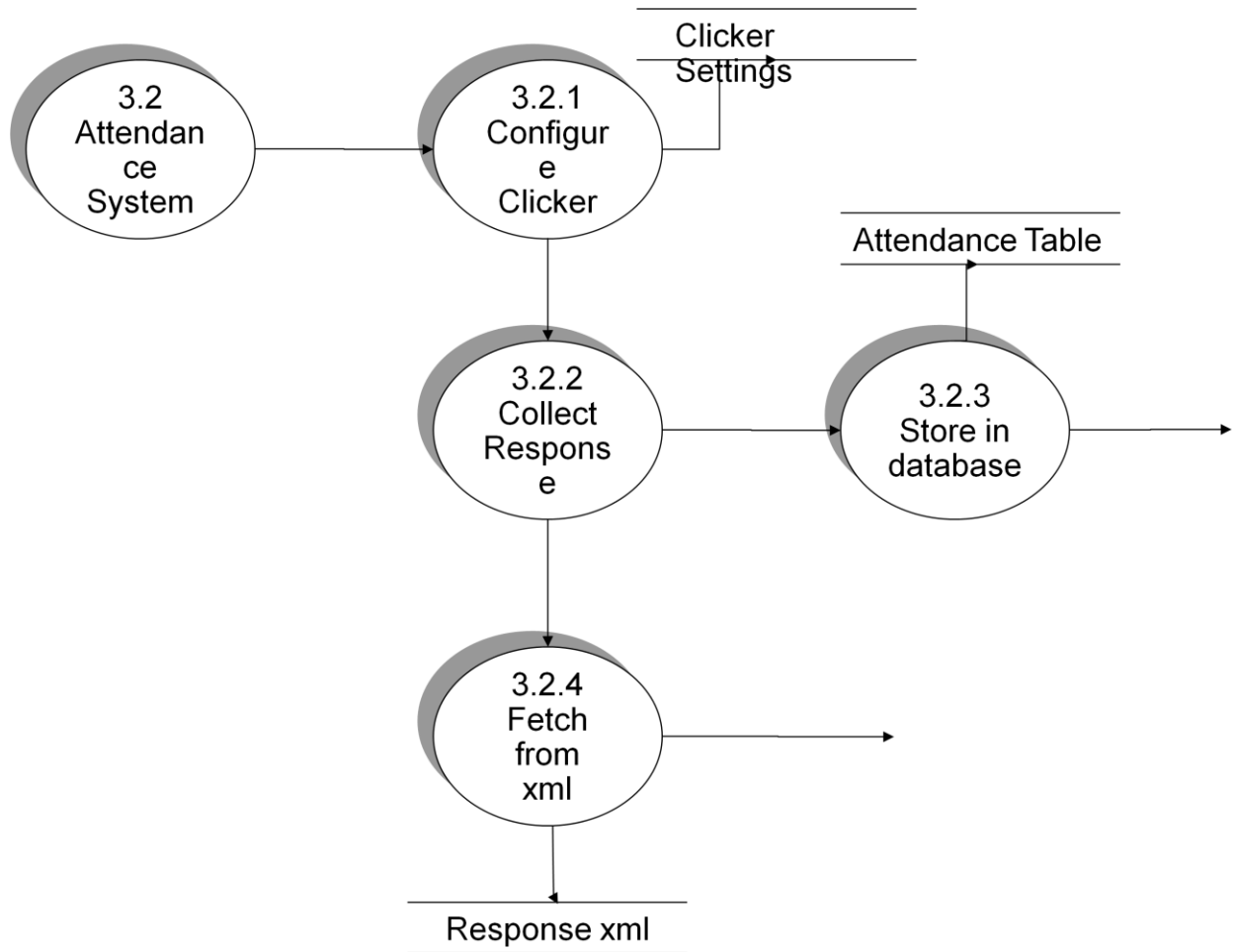
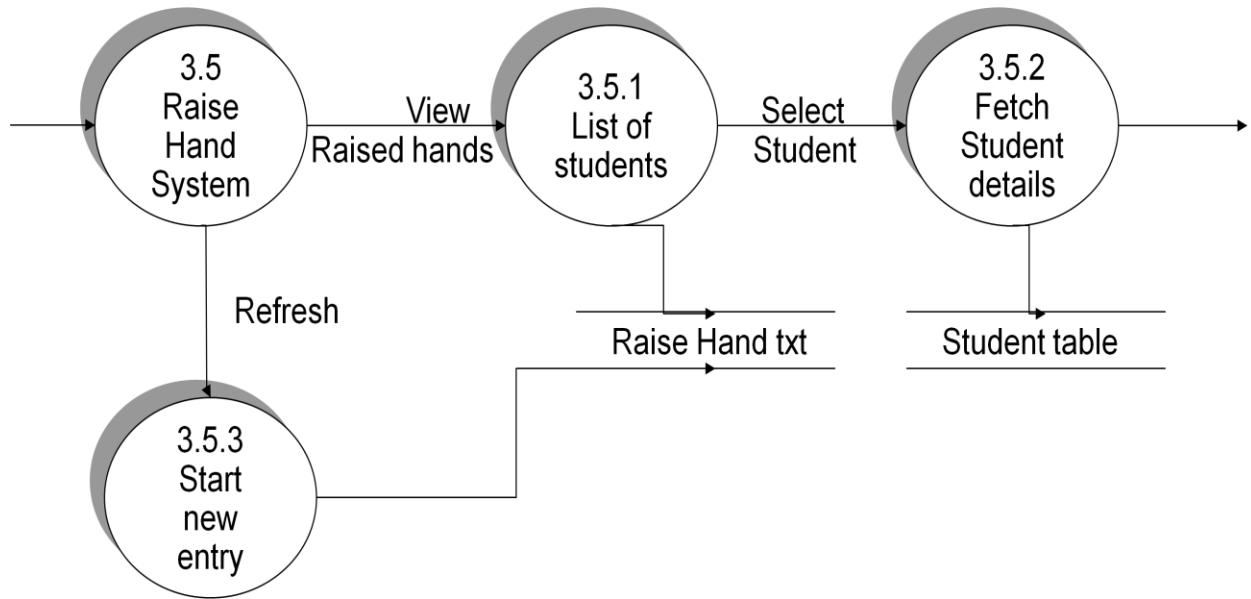
Clicker Add on

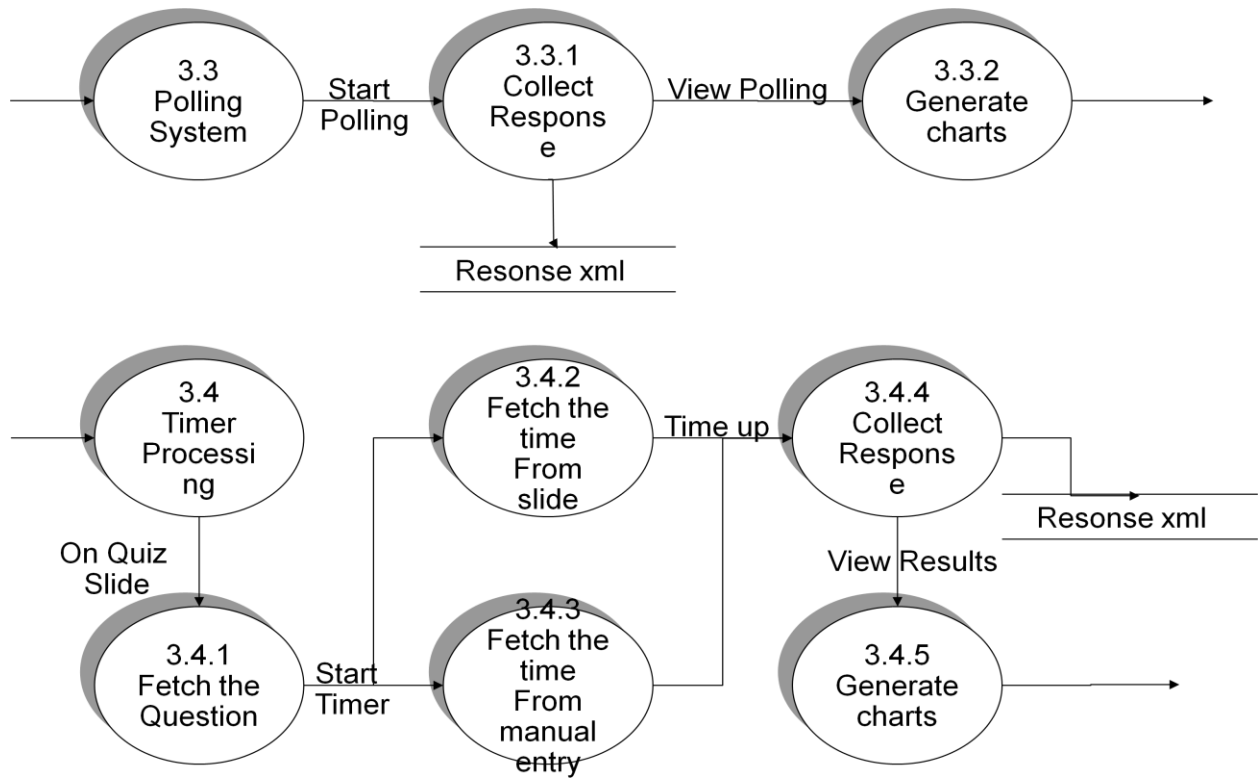
New Slide inserted with Question details



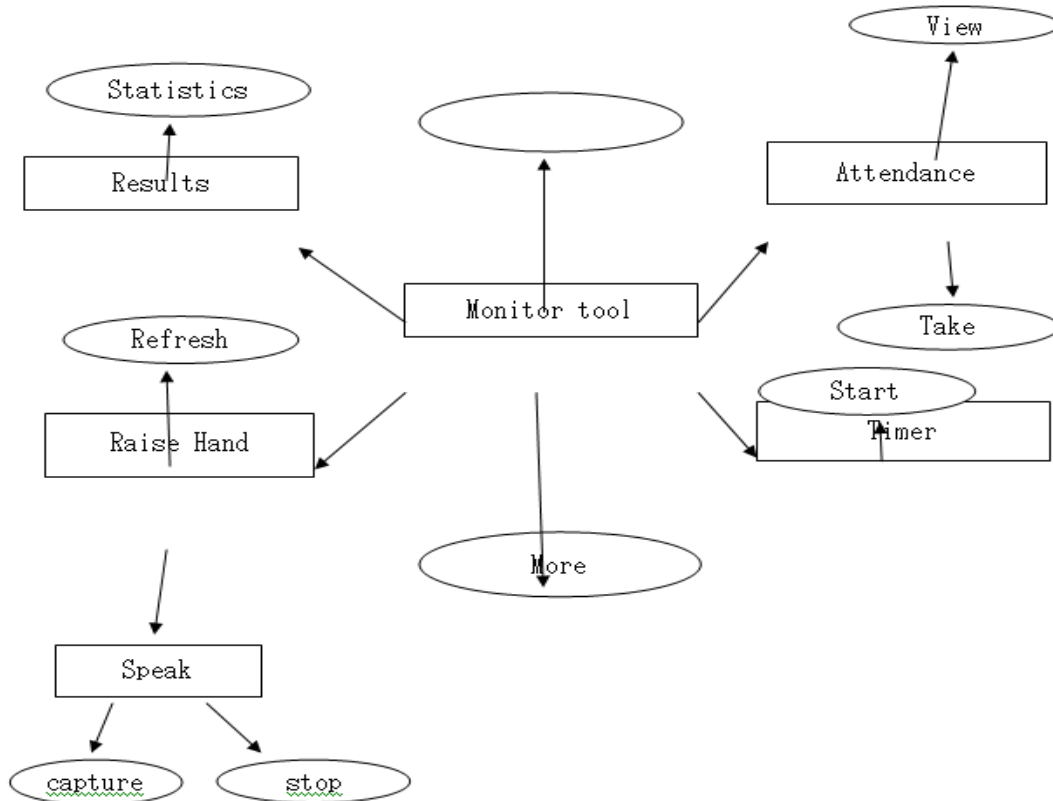








Monitor tool



This data flow diagram shows the features provided by the monitoring tool1

Detailed System Description

Description and Implementation:-

Clicker desktop application(Open office integration)

1. Collect Responses:-

This module is used for collecting responses from the students and starting the timer according to time given by the teacher. After the timer went off it will call the routine of collecting the responses from all the clicker devices which are present in the classroom. After collecting the responses it will compare the answers given by the students and prepare the chart indicating what percentage of people giving what answer.

Implementation:-

Whenever collect responses button is clicked it calls a routine for timer to be started manually. The timer runs for the given time after time is finished a python file is executed which will collect responses from all the devices which were configured. All the responses are stored into an XML file which will be read by the program to create a chart based on the responses. All the responses are also being stored in the database by calling the database routine.

2. Start Smart Slideshow:-

This module is new feature introduced into open office integration. It will create an ease for the teacher for conducting quiz with multiple questions in it. This module will start a process which automatically reads the value of time associated with current slide and set the timer to that value and after it collects the responses. Also it will read the data from the current slide in case user wants store it in the database. Also it can detect the correct answer attached with the slide and calculate the percentage of student giving the correct answer and then creates a chart for it.

Implementation:-

Whenever the smart slide show button is called a thread is started to handle the question slide in a smart way. All the question slide are assigned the properties like "QUIZ" ,"LOD" to get information about the slide. According to those properties the value of timer is set. After the timer is zero it will collect the responses from all the clicker devices and will create a chart which will show the responses. Then it will automatically goes to the next slide and set the timer according to the properties. It also

processes the slide according to the given properties.

3. View Responses:-

This module is used to see all the results collected from the different clicker devices present in the room in form of chats or statistics.

Implementation:-

Whenever the view responses button is clicked it will read the xml file which is carrying the responses from all the configured clicker devices and displays the each response with each clicker id as well the name of the student which is carrying that clicker device. This is done by fetching the name of student connected with the particular clicker id from database present on the system.

4. View Status:-

This will view the status of all reports of a particular student or a particular quiz conducted.

Implementation:-

This will create a report base on the students' responses collected over the time. All the data which are stored in the data base will be processed to create and overview of a students' performance status.

5. Save the current quiz:-

This module will invoke a process that will store all the question data which is present current slideshow into the database so that it can be used in future to create the quiz.

Implementation:-

Whenever the instructor will click on the save quiz button it will launch a process which will detect the path of the document being loaded and then it will scan through it for all the question slides. It will detect the content, type, and correct answer of the particular slide and then according to those it will enter the question in to the question table and all corresponding options in to the option so that it can be used in future for question bank.

6. Open quiz document:-

This module is used to launch open office or acrobat reader in our clicker desktop application. It will detect if the selected document is open office document or acrobat document and then open it in the desktop application for the future uses.

Implementation:-

Whenever the open quiz document button is clicked it opens a file dialogue box. It will check if the selected file is a valid document file or not. Then after the selected file is valid it will check for its extension and if the extension is compatible with open office then it will call the oobeans object of open office. If it is pdf then it will call pdf viewer of acrobat reader and then the selected document will be displayed in the clicker application frame.

Clicker Add On

1. Monitor tool:-

This module will start an application which will assist the instructor with the conducting quiz and poll during the session of slide show. This monitor tool will handle taking attendance, raise hand, storing questions, viewing results, voice communication, and collecting responses from the clicker devices.

Implementation:-

Whenever the monitor tool button is clicked it will create a java frame which will act as a java application. This application will be having all the features of clicker desktop application. It will also detect if the current slide is a quiz or not by checking its property. If it is a quiz slide then it will store its content for processing. It will also take the control of the current slide show.

2. Smart slide show:-

This module will start the slide show of the current document which can be controlled and handled by the monitoring tool. The slide show can be started from the beginning or from the current slide.

Implementation:-

Whenever the smart slide show button is clicked a routine is started and a slide show is started from the beginning or from the current slide according to the selected option. It will start a thread which will prepare a controller which will be then passed to the monitoring tool. If the current slide is a question slide then it would automatically launch the monitoring tool.

3. Configure clicker:-

This module will configure all the clickers i.e. register the expected devices for the particular course. It will create an xml file for all students in for that

particular course with the help of the database and also checks if all the hardware is connected properly or not.

Implementation:-

Whenever the configure clicker is clicked it will read from the xml files and prepare an input for the python file to check for all the devices one by one using the polling method. It will fetch the students expected for the selected course and prepare the system to check for only those clickerids.

4. New Quiz Slide:-

This will prepare the slide with the smart layout. The instructor must prepare the question in with help of this option. This will provide the facility to create all kind of questions which can be handled by the clicker device. It also provides the facility to associate correct answer, time and course id with the particular slide.

Implementation:-

Whenever the new quiz slide is clicked it will create an empty slide layout and pop up a dialogue box for user to set the properties according to the selected type of the question. It means if it is true or false question then it will set the options as true or false in the slide and also provide the space for questions. Option in the dialogue box will be assigning properties to the slide like correct answer, level of difficulty, department, credit and time to be given for this question.

5. Question Info:-

This module provides the facility to modify the current question slide and see the data for the current slide as well.

Implementation:-

This will pop up a dialogue box to change the properties assigned to the current slide. It will also be showing the properties for current slide.

6. Clicker Tool:-

The integration of clicker software is done to use all the features of clicker application into the open office software. This module contains all the facilities of clicker application. This module provides practically all the things which were there in clicker desktop application in the form of lists of menus in the open office.

Implementation:-

This option will be provided as a menubar item in the open office software. It will provide all the facilities mentioned above as its menu items.

Monitoring tool

1. More:-

This module is used for opening the provided modules like raise hand, attendance into the new window with some extra features which are not often used. It will detect which module's tab is currently selected and then open the particular window to be used.

Implementation:

This option will create a new java frame according to the tab currently selected. The newly created frames will be having some more functionality to handle the module. The newly created frames will be poll for results, Attendance for present and Raise hand frame for handling the raised hands.

2. Save:-

This module is use for saving the question and option of the current slide into the database. It also stores the correct answer, credit for the question, level of difficulty, time provided for that question.

Implementation:-

Whenever the instructor will click on the save quiz button it will launch a process which will detect the current running slide of the document being loaded and then it will scan through it for the question. It will detect the content, type, and correct answer of the particular slide and then according to those it will enter the question in to the question table and all corresponding options in to the option so that it can be used in future for question bank.

3. Raised hands:-

This tab will show the list of all the students who has pressed the raised hand button along with their names. It also provides the facilities to speak with that student.

Implementation:-

This tab will be showing all the clicker ids who has raised their hands with their names which are fetched from the database. Whenever a student press the raise hand button ,the monitor tool will start blinking. The check for raised hand is done by the constantly checking the raisedhand.txt file if it is modified then the monitor tool will start blinking. That is how instructor will know that someone has doubt and if he looks for it in the list then blinking will stop. The list of all students will be displayed in the monitor tool as well as raise hand frame.

4. **Timer:-**

This tab provides the facility to start the timer manually and automatically according to time set in to text boxes. It also shows the time left before collecting the responses from the clicker devices.

Implementation:-

This will check for the text box provided for the time value if nothing is set into those boxes then the value of time for that question will be set from the slide's property. If the value is set then value of time will be taken from the text box and new timer frame be launched showing the countdown timer for the question. After the time is up it will collect responses from the xml file created by the python process and creates the chart for the current question.

5. **Results:-**

This tab will show the list of conducted quizzes and also list of the charts prepared by the results collected from the responses for the particular slide.

Implementation:

All the charts for the current slides are stored in the database. This option will show the list of all the charts which was prepared during the session and their name will be the title of the slides which was containing the question.

6. **Attendance:-**

This tab will show the status of all the students there in the course. It will also configure all the clicker devices present in the classroom and take the attendance.

Implementation:-

This will do launch a thread which will launch both process configuring clicker according to the selected course in the list and taking the attendance for the students of that particular course.

It will first do the configuration and then check for the responses xml file. Whoever has responded properly will be counted as present and this status will be then stored into the data base.

7. **Speak:-**

This is the voice integration part of the clicker device. This will trigger the voice module for the selected clicker device and transfer the audio data received from its clicker device and transfer it to the speakers as well store it into the machine of the instructor.

Implementation:-

This is the voice integration part of the clicker device. Whenever instructor

click on the speak button it will launch a process which will detect the selected clicker id and then start taking the input from the receiver as target data line and redirect it to speaker as source data line. It also Store the data in to temporary buffer and then store the audio received from receiver into the file system whose path is stored into the database. It will store all the audio data of on session in one file as .wav format.

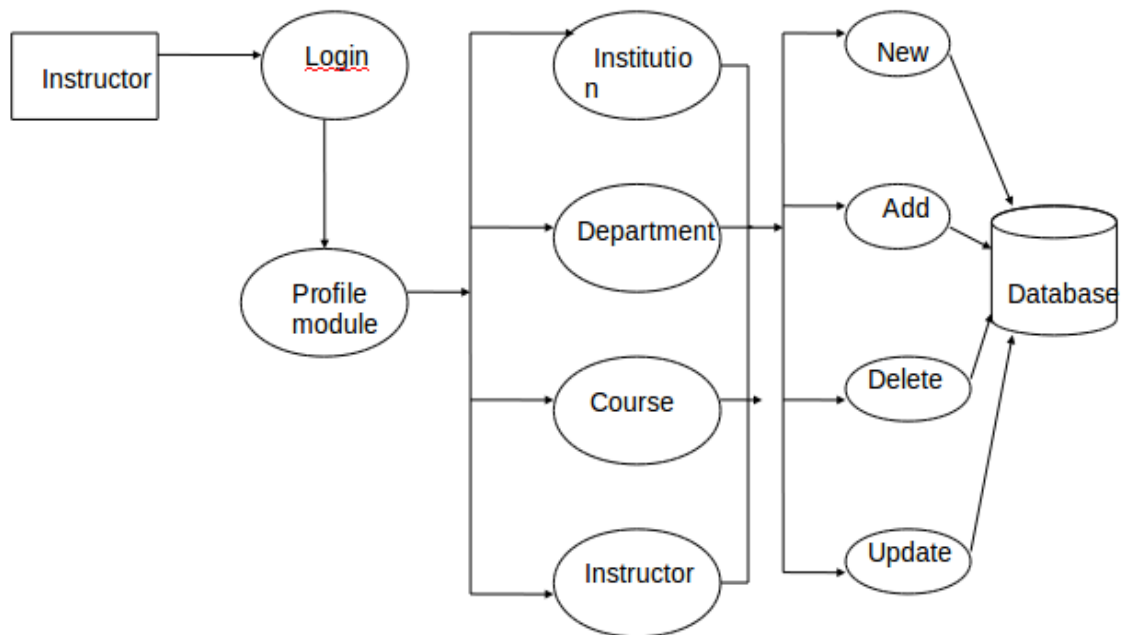
Chapter 8

Design and Implementation of Web Based Application

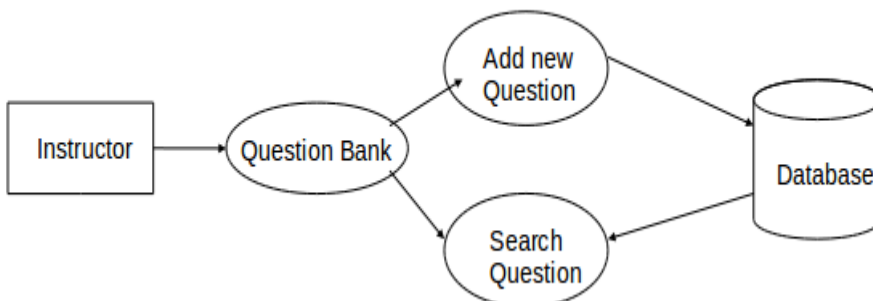
To design the whole software, we have used a basic theme of the page which was designed using CSS and is the base for the other module and through that page we have linked another page using Ajax to avoid reloading , hence in that page only the content of one div kept on changing and we get all the module like this. Now we describe the data flow diagram of different modules.

Data Flow Diagram

The Level 0 DFD is shown here and it explain the working of the Profile module



- The Level 0 DFD of the Question module is given here which explain its functionality.



4.2 User Interface

Now we start with the different module and describe its functionality and interface:

- **Question Bank:** The question Bank module of our project is designed for adding question and searching the specific question to conduct either quiz or for other classroom sessions. The interface is shown below and as can be seen from the interface that we can search a question by using a string in the database and the question can also be searched by specifying its lod, credit etc and also we can add a new question on clicking the button and also can get the question through a xml files. It also provide the facility to perform some editing in that Question.



- **Profile Module:** This module provide the login facility through a user name and a password and in this we add the new instructor and can fetch the information about the instructor and provide him the login .



Menu

Raise Hand

Poll

Attendance

Quiz Bank

Question Bank

View Report

Profile

Settings

Institution

Department

Course

Instructor

java.lang.ClassNotFoundException: com.mysql.jdbc.Driver

Institution ID	Institution Name	Address
----------------	------------------	---------

Institution Name

Address

New

Add

Delete

Update

Reset

- **Quiz Bank:**

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 as selected by the instructor and also they can create the quiz as required by the instructor

- **View Report :**

This module is used to view the report of the particular of the particular student depending upon its course and student id. Here we can view the report of any student from any institute.


- **Raise Hand:**

This module is used in realtime and used when a student is having some doubt or confusion, it is just the raising hand in the classroom environment. The interface associated with this is shown below:

The screenshot shows the 'Raise Hand' interface. At the top left is the IIT Bombay logo, followed by the text 'Indian Institute of Technology-Bombay'. On the left side, there is a dark red menu with the following items: 'Menu', 'Raise Hand', 'Poll', 'Attendance', 'Quiz Bank', 'Question Bank', 'View Report', 'Profile', and 'Settings'. The main area is titled 'Raise Hand Clickers' and contains a large empty box. To the right of this box are input fields for 'Name', 'ClickerID', 'StudentID', 'Course', and 'Department'. Below these fields is a 'Comment/Query' text area. At the bottom right, there are four buttons: 'Get Details', 'Refresh', 'Save', and 'Stop Raise Hand'.

- **Setting:**

It is GUI module representing Clicker Software Settings.and its interface is as shown below:


Indian Institute of Technology-Bombay

Menu

Raise Hand

Poll

Attendance

Quiz Bank

Question Bank

View Report

Profile

Settings

Clicker Database

ServerIP

Server Post

User Name

Password

Clicker Mode

Hardware Connection

Clicker Remote Code

Python Parameter

Windows USB Port

No. of Cycles

Retry Count

Clicker Remote Server

ServerIP

User Name

Password

Socket

Clicker Version

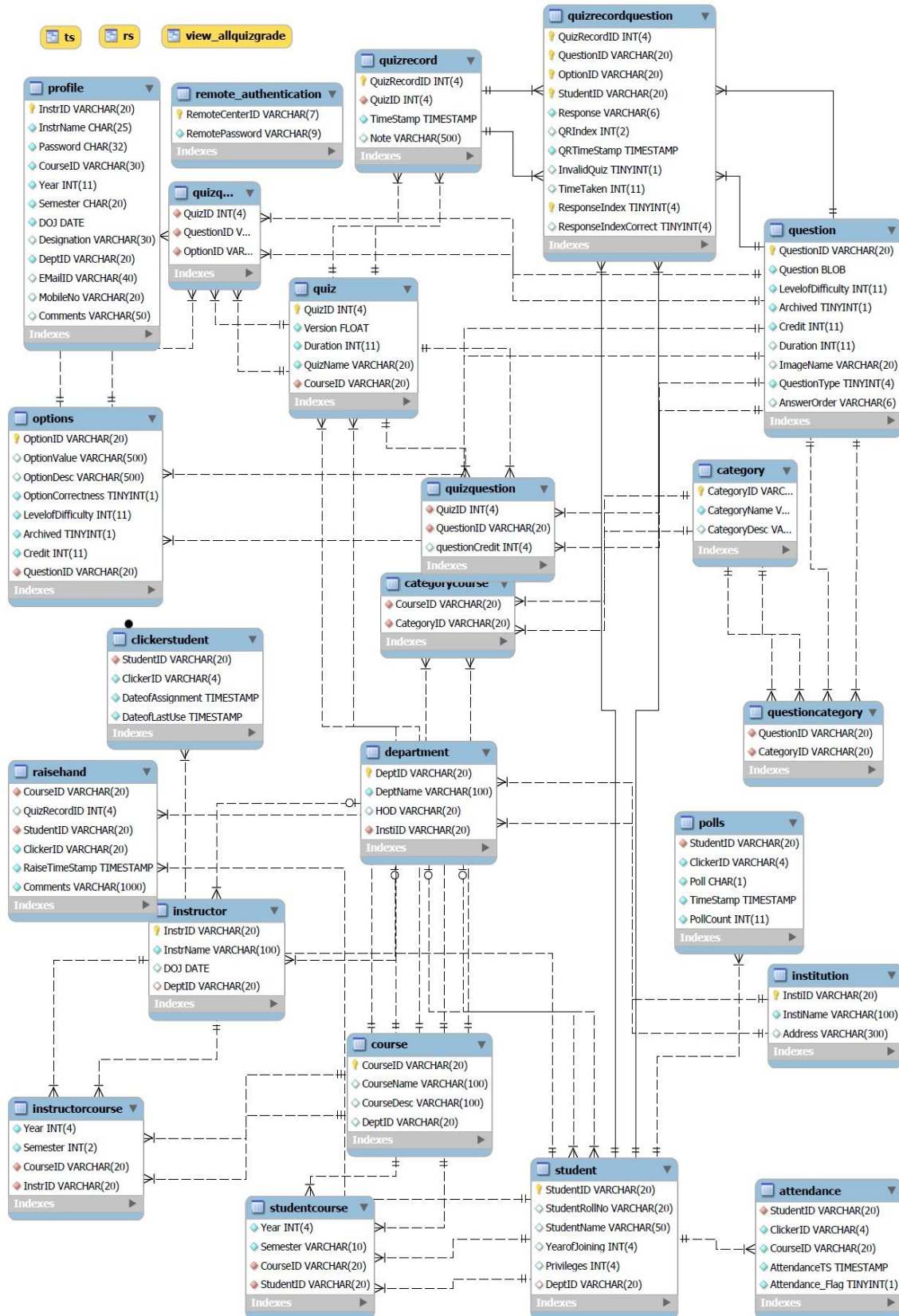
Version Update

Open Office Support

Open Office Support

ER Diagram

The ER Diagram associated with the Database of the Clicker software is given below, the database is kept on updating according to the requirement and it will be also get modified in future to make the database system suitable for the requirement of the user.



Now we begin with the detailed system description of the Clicker Web Application software system. As we know that it is divided into different modules and here we will describe in detail about these modules:

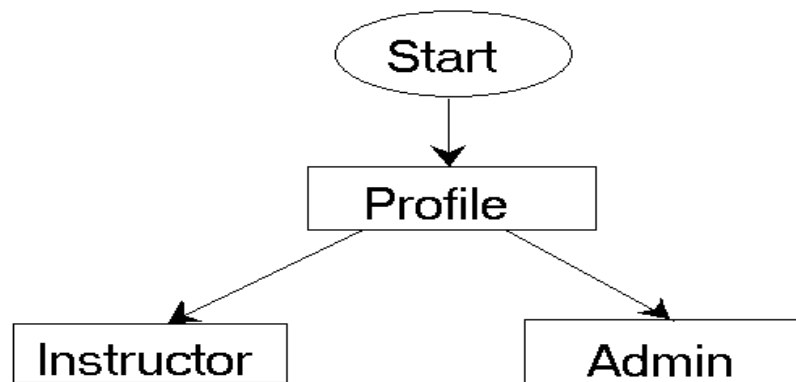
- **Profile**

For any instructor, registration is must to direct access centralized database, so Administrator Profile gives provision to register any Instructor in database with authenticated Username and password for further usage. This authentication will be valid forever for that particular Instructor to manipulate Clicker Software Database.

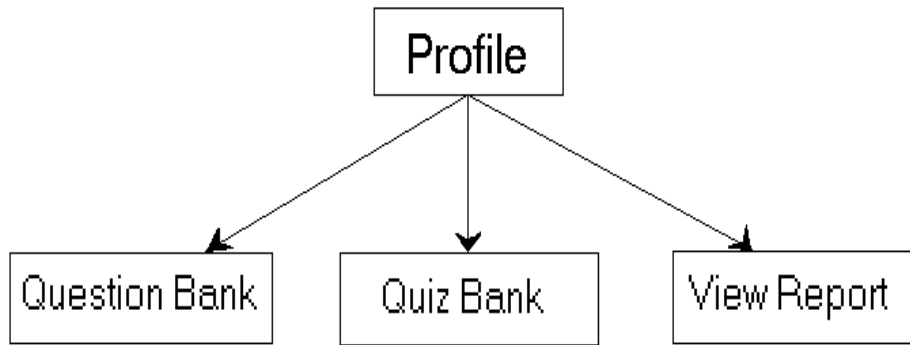
After successful login for Administrator Database Access, Instructor can use Clicker Database Admin Utility. This is very important Utility in order to do updates in Database. This is way by which an Instructor can direct access Database Contents and understand the status of Database.

Admin is Authenticated Instructor of Clicker Software System. There can be more than one Admin for Clicker Software System. The actions done by this Admin People is totally authenticated and valid according to rules for any Database Administrator. So rights related to changes of database include mostly Institution, Department, Course and Instructor. All manipulative changes related to database access such as Insert, Delete, Update, Show can be performed by efficient way

This will be the first module through which the user interact, here the user makes a login and then he will be allowed to go through other module. The flow chart is shown below:



And after the login in the profile the user can go through other module

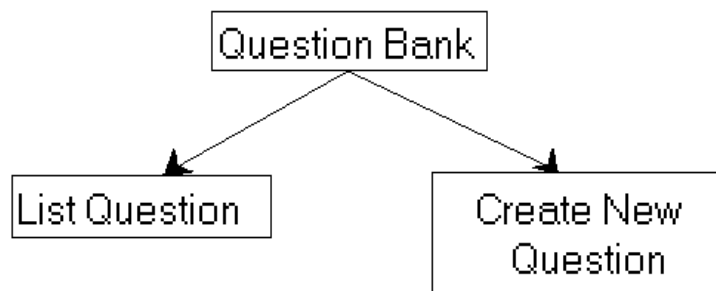


- **Question Bank**

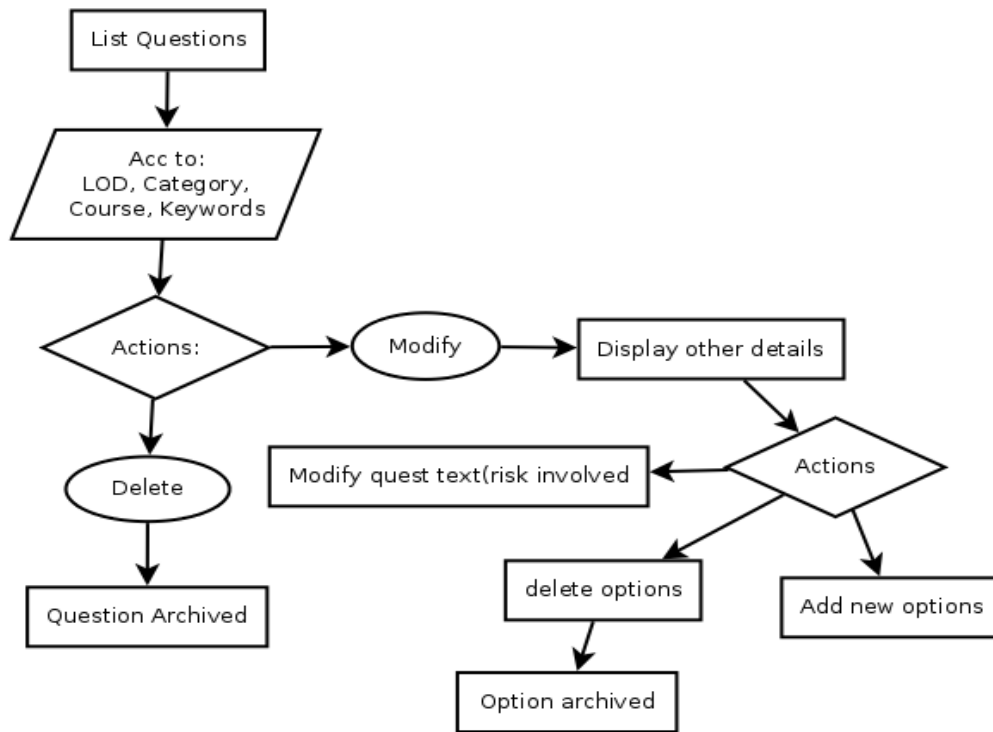
Similar to Quiz Bank this module is related to Questions only. You can create Question according to Course parameters as level of difficulty, Question type. Similar to Quiz, here also you can construct Questions by following types with minimum two to maximum six options as,

- 1) General Quiz having Single Answer
- 2) Multiple Answer Quiz having Multiple Answers
- 3) Numeric Quiz
- 4) True False Quiz
- 5) Yes No Quiz

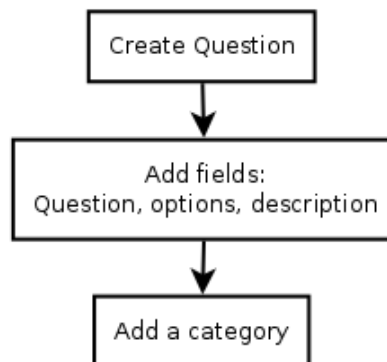
There is one more provision for adding Question in database is by using XML Template. We have Question Clicker Template, an isolated Utility which will prepare one xml file according to your Question data. You can use that XML file as input for creating Question, the flow chart is shown below:



The following chart shows the list Question part of the Question bank module which is used to search the particular question from the question bank:



The following chart shows the Create Question part of the Question Bank module which is used to add new question in the database which can be further retrieved from the database:



- **Quiz Bank**

As name suggest, this module is very much related to Quizzes. By using this module, Quizzes can be formed by selecting particular Questions and courses parameters. For Conduction of any quiz this primary module. The basic step behind any Quiz Conduction including open office integration also you will find here.

Quiz can be conducted by using software as by selecting it in Quiz Module. For particular Quiz List you have to select related necessary Course parameters. So

after getting Quiz selected the most important step is to configure all Clicker IDs to receiver. There is one button on Quiz Module it is Configure Clickers by Clicking on that it will configure all corresponding Clicker IDs to receiver according to selected Course.

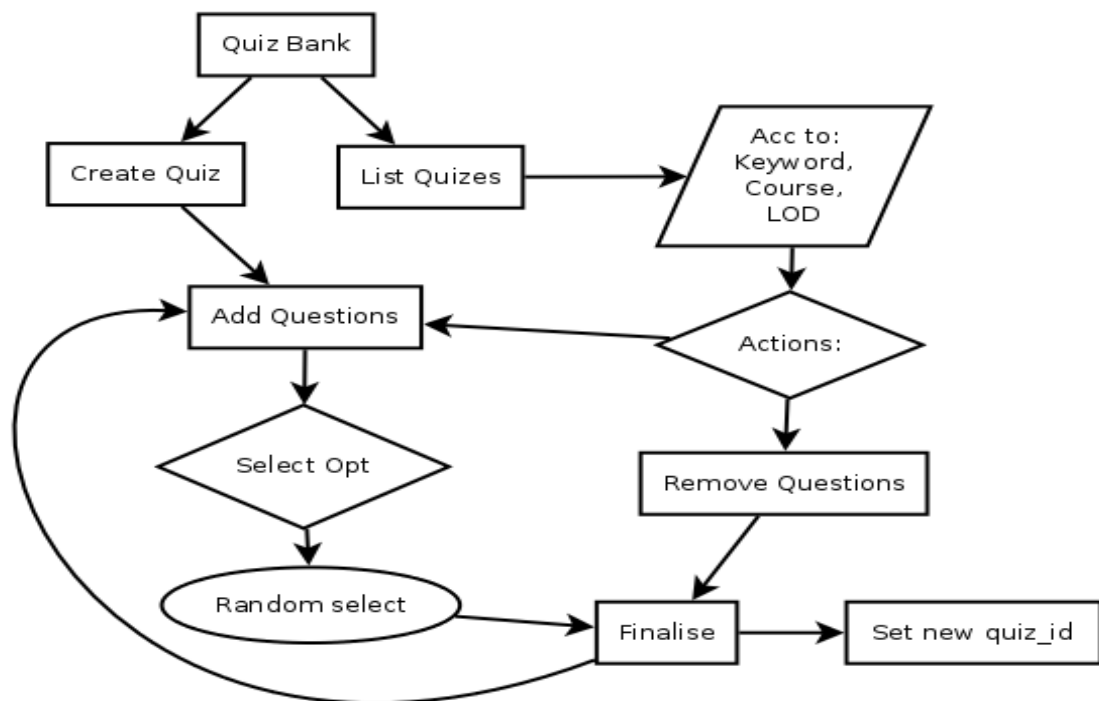
There are two modes of Quiz Conduction:

1) Automatic Mode:

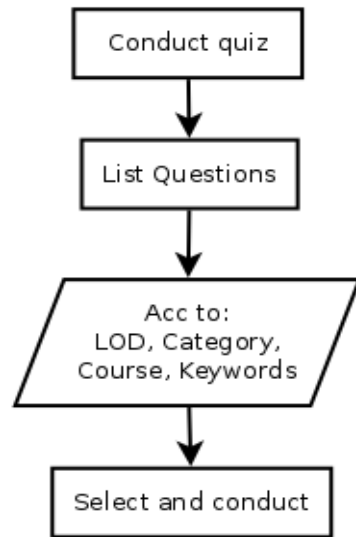
Here suppose multiple Questions carrying Quiz you want to conduct then you can give time and without any interruption, definite schedule conduction of Quiz can be done. At end of given time only you can see the Result.

2) Manual Mode:

Here in any Quiz Type you should click on Next Button to proceed, there is no automation behind any procedure. You should do every action manually. Meanwhile you can see result for Question also at runtime and the flow chart is shown below:



And the conduct quiz will occur as shown in the flow chart below:



- **View Report**

Based on database records, the following typical reports can be constructed according to their base types.

1. Based on Participants

- List of Participants for particular session
- Attendance Report
- Particular Quiz Report of Participants
- Question/Comment Report (Based on Raise Hand)

2. Based on Quizzes

Graphical Report of Quiz

Percentage Results of Participant

Quiz Result per Participant

Quiz Result per Participants

3. Based on Results and Participants

Graded Result per Participant

Graded Result per Participants

Percentage Result by Participants (Response Details)

Percentage Result by Participants (Response Details)

The main goal of View Report is to display all Jasper Report files on web. Jasper Report is basically the one of the leading open-source Java reporting libraries. It compiles .jrxml (XML source) to .jasper (compiled) files, which in turn can be transformed into several output types including PDF, HTML, CSV, and XLS.

- **Raise Hand**

It is Special Utility module designed for any Classroom or Workshop session to accept and to present the doubts, Queries, questions or comments raised by Clicker Raise Hand Facility. When the Instructor will start Raise Hand Mode of Software, Clicker IDs having queries will be listed out.

Here so by selecting particular Clicker ID and getting their all Details, later on by Verbal Communication Queries will be resolved. Along with this if you found any Query very important then that Query can be saved with related necessary parameters in database and further retrieved in future for analysis purpose.

- **Attendance:**

This module is used to take the attendance of the student. The student can make their presence by switch on the clicker by pressing RESET button and by pressing ST Button on clicker. In this the instructor can able to take attendance, see the attendance and can able to see the absence and presence. Before taking the attendance the instructor wants to select the course after selecting the course from the drop down list box. If the attendance has been taken successfully then the alert box will open. If there is any error occur in the hardware connection then the alert box will display not successful. After the attendance taken the instructor can view the attendance, presence and absence by pressing the corresponding button. The view attendance is used to display the attendance for the corresponding course which the instructor can select in the dropdown list box. View present and view absent is used to see the present and absents of student for the corresponding course.

- **Setting:** It consist all Settings parameters which are essential among Clicker Software Functionality. It is GUI module representing Clicker Software Settings.

- **Polling:** It is quite similar to the raise hand and used to take poll i.e vote from the students or others.

Conclusion and Future Work

To conclude with, all the assigned tasks have been successfully completed by the team and the above mentioned features are integrated with the project.

- Python Work: At later stages the maximum capacity of a receiver can be increased from 200 clickers and the industrial communication protocols need to be used.
- Installer: The installer should be such that you have options to install certain addons and leave others if you don't want them. Similarly there should be an option for uninstalling some packages and not necessarily all those got installed.
- Open Office: It can be made more attractive in design and the 2 screens that overlap for timer needs to be improved for betterment.
- Web Based Application: Socket / HTTP communication programming has to be done between client server for interaction.

Technical Details and FAQ:

Technical Details:

Python Serial Utility:

FILE	LOC	DESCRIPTION
gpv.py	15	Contains all the in-built python packages used for serial communication
PythonIntegration.py	118	Is the main python executable which imports all other function definitions
maindef.py	131	Contains the main definitions of ClickerConfig, ClickerAttendance, ClickerRaiseHand, ClickerSendSYNC, ClickerQuiz
sdef.py	93	Contains secondary definitions used for message transfer
ClickerConfigmsgdef	253	Contains the message definitions used for message transfer for configuring the receiver
ClickerAttendancemsgdef	93	Contains the message definitions used for message transfer for collecting attendance from the clickers
ClickerRaiseHandmsgdef	268	Contains the message definitions used for message transfer for collecting Raise Hand Response from Clicker
ClickerSendSYNCmsgdef	114	Contains the message definitions used for message transfer for syncing the receiver
ClickerQuizmsgdef	125	Contains the message definitions used for message transfer for collecting quiz response from Clickers

Total LOC = 1210

GUI Installer:

FILE (shell scripts)	DESCRIPTION
install_jre.sh	This file will be having a command which installs the JRE .
install_python.sh	This file will be having a command which installs python-serial.
jre-6u22-linux-i586.bin	This is the source package of JRE.
Python-serial_2.3-1_all.deb	This is the source package of Python-serial
uninstall_jre.sh	This file will be having a command to uninstall Python-serial.

install_jre.sh

This shell script installs the Java Run Environment in user's machine.

And that command it includes is “sudo ./jre-6u22-linux-i586.bin”.

install_python.sh

This shell script installs the python-serial in user's machine.

And the command it includes is “sudo dpkg -i python-serial_2.3-1_all.deb”.

uninstall_jre.sh

This shell script uninstalls JRE in user's machine.

And the command it includes is “sudo rm -rf /home/`whoami`/jre1.6.0_22”

Open Office Integration:

Code Details:

LANGUAGE	FILES	LINES OF CODE
Java	47	6552
XML	9	1213
SUM:	56	7765

FAQ:

Q.) Where to find all the work done by summer interns 2011?

You should contact Mr.Tushar Kambli, Mr.Manjurelahi, Mr.Rajavel for the same. You will find them in ASL 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 start the work?

As far as we as interns write it, just don't think to much , just start doing something , take help from Google and your mentors always. Keep discussing with friends as well.

Quiz (with timer and clicker ID 2 is pressing wrong answer)

```
Enter choice
>3
enter Question time
>10
enter Question No
>1
enter Question Type
>1
Send Correct?
>yes
Correct Answer
>b
Current using Serial Port is /dev/tty50
Sending SYNC as ['0x3a', '0x29', '0x1d', '0xc', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0xa', '0x0', '0x0', '0xa', '0x1', '0x1', '0x18', '0
x6', '0x7', '0xdb', '0xb', '0x28', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0xfe', '0x2b']
SOF1 of ACK for SYNC is = :
ACK Message Type 4 at Start ['0x3a']
SOF2 for ACK )
Message Type of ACK 4
ACK Message Type 4 as is ['0x3a', '0x29', '0x1d', '0x4', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '
0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0x0', '0xff', '0x7c']
Checksum for ACK by Serial 65404
Checksum for ACK by Python 65404
SYNC Sent to Receiver Successfully
10
9
8
7
6
5
4
Quiz Response is as follows:
Clicker Nos      Response      TimeTaken
1                X            0
2                E            5
3                X            0
4                X            0
5                X            0
6                X            0
7                X            0

45              X            0
46              X            0
47              X            0
48              X            0
49              X            0
50              X            0
Writing in XML File

1. Clicker Attendance
2. Raise hand module
3. SendSYNC
4. Exit
Enter choice
_
```

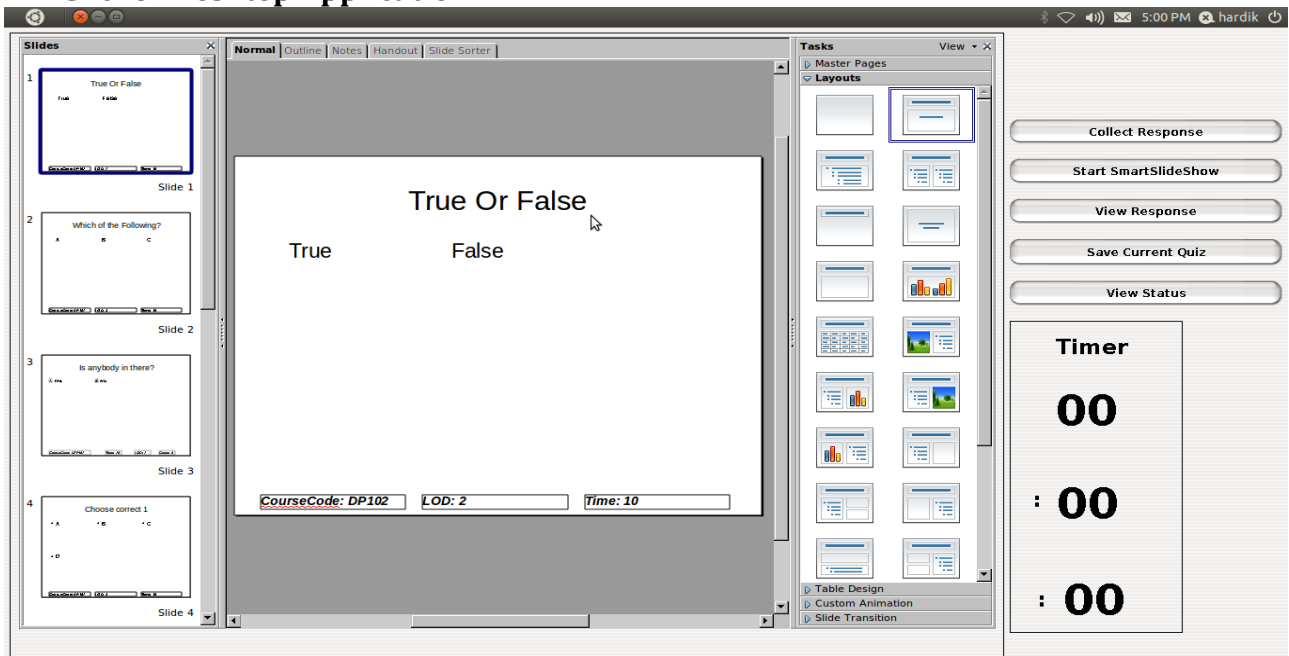

Open Office Integration:

User Interface Screenshots:-

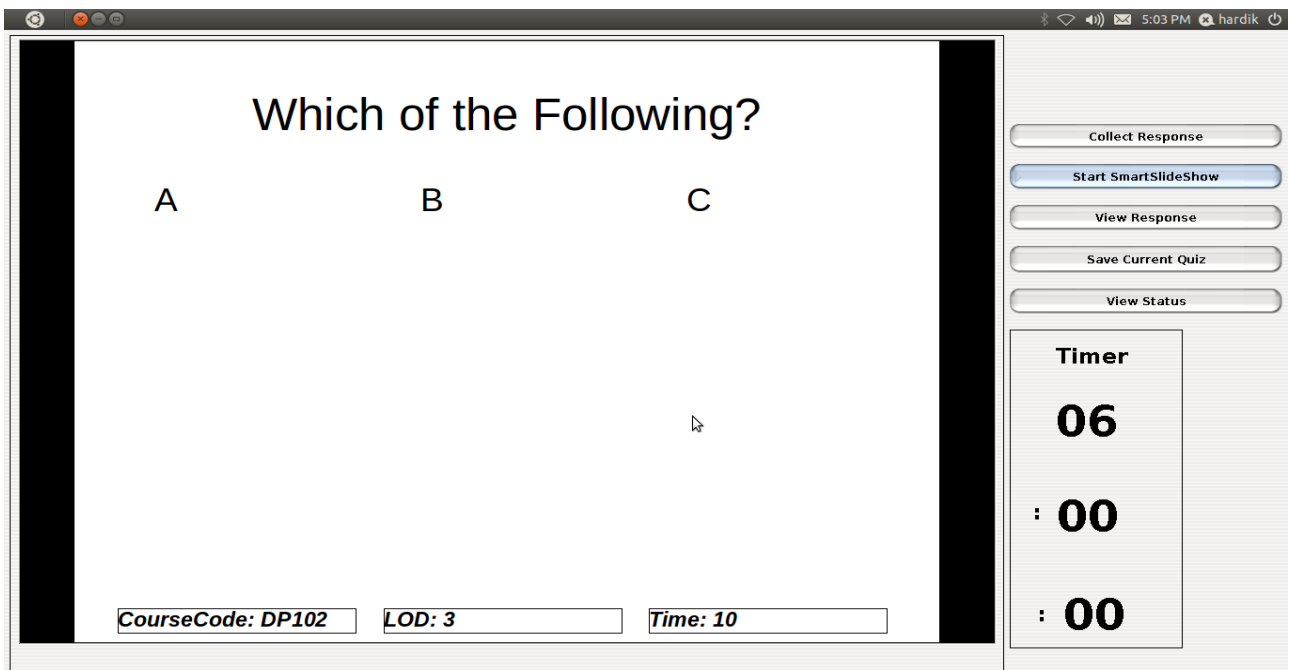
Clicker Settings

Clicker Database	Clicker Remote Server
Server IP: localhost	ServerIP: 10.129.50.129
Server Port: 3306	UserName: manjur
User Name: clicker_ui	Password: *****
Password: *****	Socket Port: 2004
Python Parameters	Clicker Mode
Window USB Port: 4	Hardware Connection: <input checked="" type="checkbox"/>
Number of Cycles: 1	Clicker Remote Mode: <input type="checkbox"/>
Retry Count: 1	CourseCode: DEP102
Save Cancel Reset	

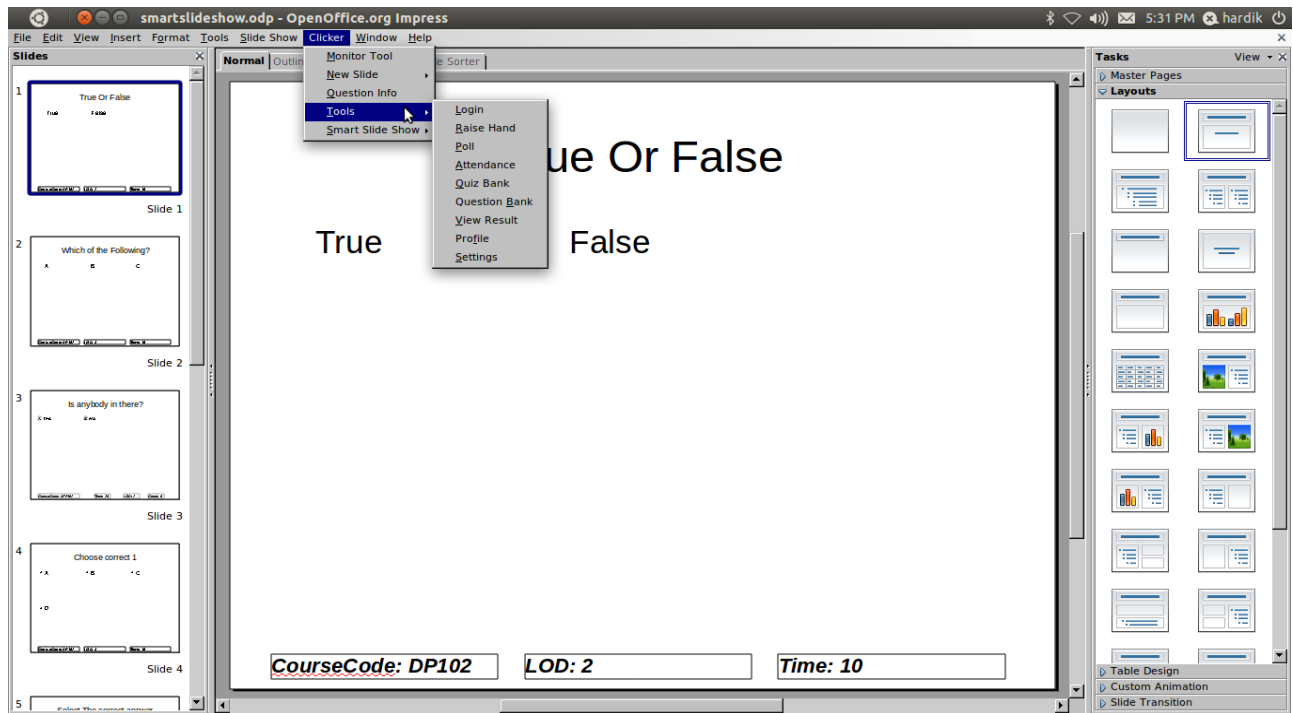
1. Clicker Desktop Application



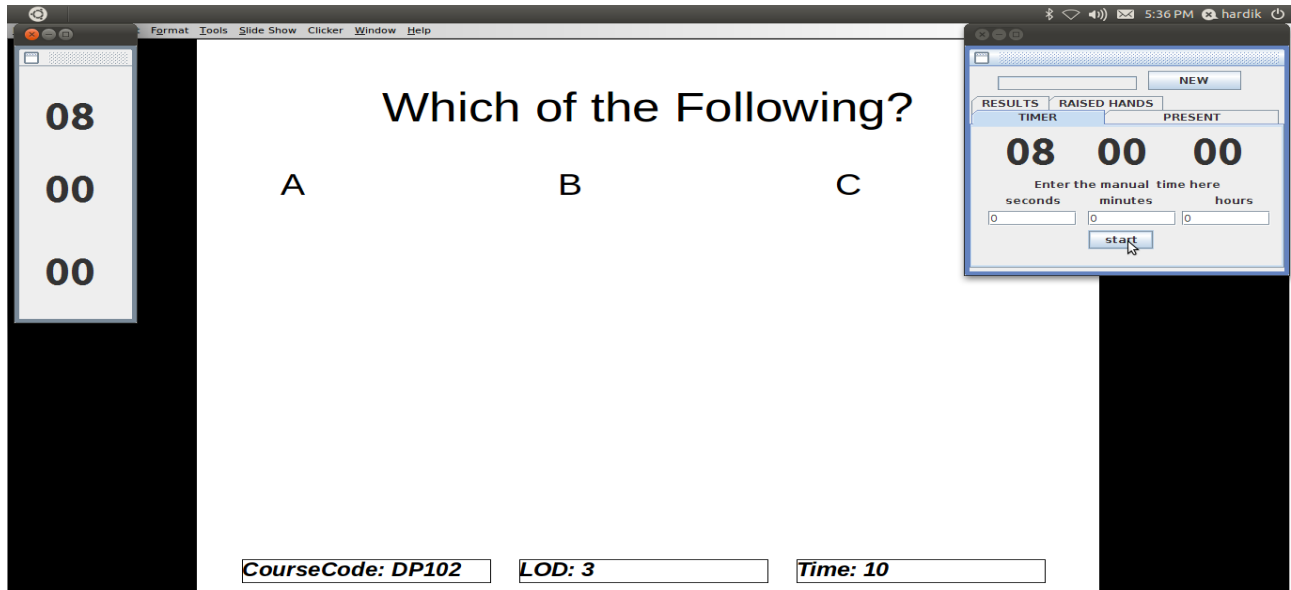
Openoffice integration in clicker desktop



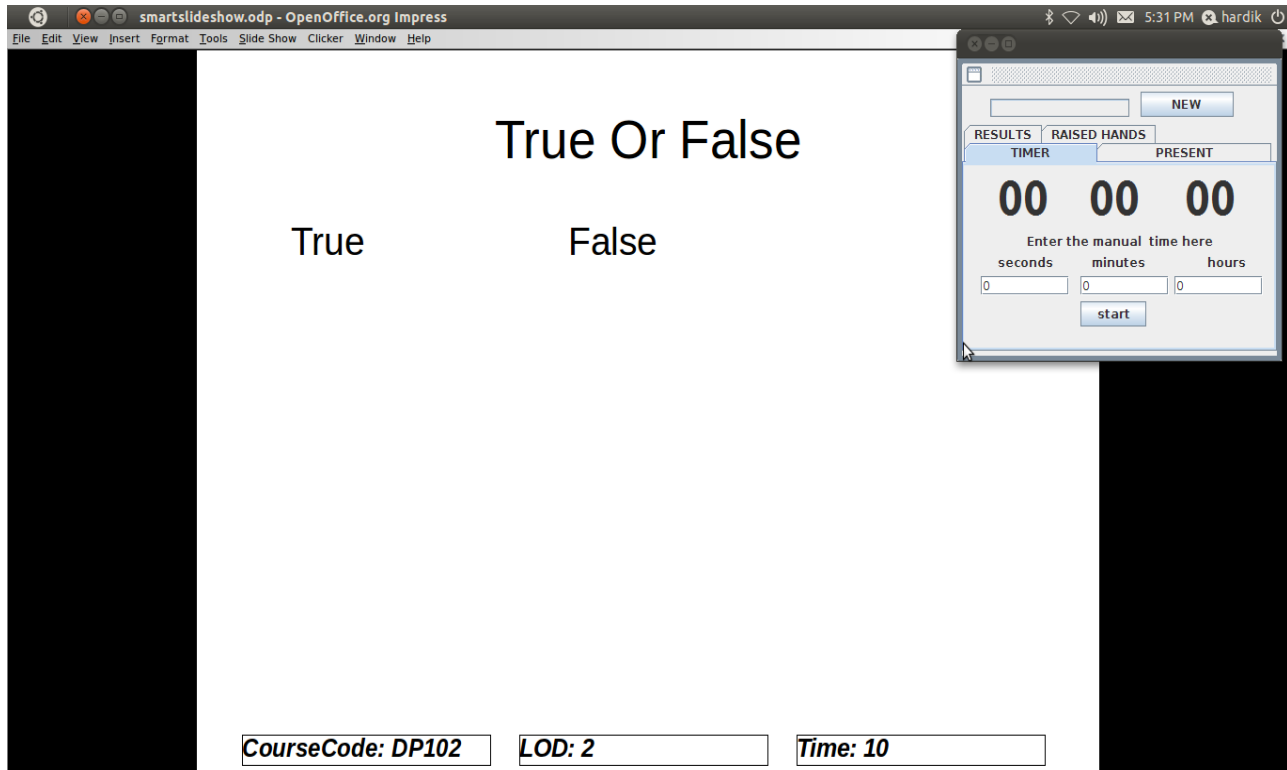
Timer in openoffice



2.Clicker Addon



monitor tool (1)to monitor student activity and timer for students(UI1)



The screenshot shows a presentation slide with a poll question: "Which of the Following?". Below the question are two options, B and C. A mouse cursor is pointing at option B. To the left of the slide is a window titled "Pie Chart" showing a pie chart with six segments labeled A through F. The segments are: A (red), B (blue), C (green), D (yellow), E (magenta), and F (cyan). A legend at the bottom of the pie chart window shows colored circles corresponding to labels A-F. At the bottom of the slide, there are three input fields: "CourseCode: DP102", "LOD: 3", and "Time: 10".

Charts for responses from student

Poll Frame

The screenshot shows a window titled "Poll". It has a table with two columns: "ClickerID" and "Poll". The table area is currently empty. To the right of the table is a vertical stack of buttons: "Bar Chart", "Pie Chart", "Refresh", "Save Polls", "Stop Polling", and "Close". A mouse cursor is pointing at the "Close" button.

Raised Hand Frame

Raise Hand

Raise Hands Clickers

Name:

Clicker ID:

Std ID:

Course ID:

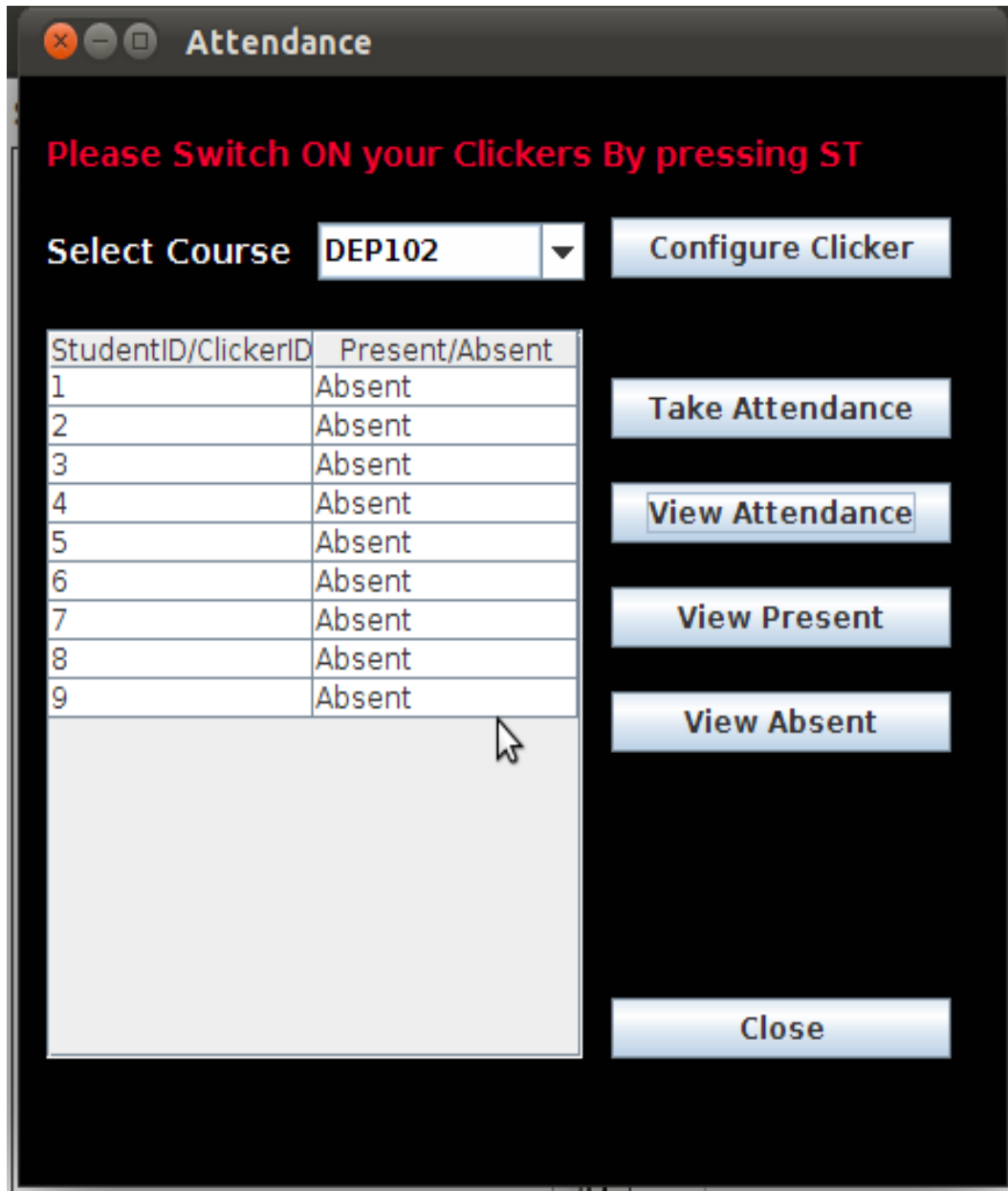
Department ID:

Comment/Query

Details **Refresh**

Save **Close**

Attendance Frame



The screenshot shows a software window titled "Attendance" with a dark background. At the top, there is a red instruction: "Please Switch ON your Clickers By pressing ST". Below this, there is a "Select Course" dropdown menu currently showing "DEP102" and a "Configure Clicker" button. A table with two columns, "StudentID/ClickerID" and "Present/Absent", contains nine rows, all of which are currently marked as "Absent". To the right of the table are five buttons: "Take Attendance", "View Attendance", "View Present", "View Absent", and "Close". A mouse cursor is visible over the bottom right corner of the table area.

Please Switch ON your Clickers By pressing ST

Select Course

StudentID/ClickerID	Present/Absent
1	Absent
2	Absent
3	Absent
4	Absent
5	Absent
6	Absent
7	Absent
8	Absent
9	Absent

Web Based Application:

Screenshots associated with Question bank:

Clicker V3

Menu

- Raise Hand
- Poll
- Attendance
- Quiz Bank
- Question Bank
- View Report
- Profile
- Settings

Add New Question

Question Text

Course
Category
LOD

Option

Explanation

Correct Yes No

LOD Credit

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools MySQL Enterprise Help

Back Next Execute Stop

SELECT * FROM question q LIMIT 0,1000

QuestionID	Question	LevelofDifficulty	Archived	Credit	Duration	ImageName	Qu
d_1	BLOB	1	0	5	60	NULL	1
d_10	BLOB	1	0	5	60	NULL	3
d_11	BLOB	1	0	15	60	NULL	2
d_12	BLOB	1	0	5	60	NULL	2
d_13	BLOB	1	0	5	60	NULL	1
d_2	BLOB	1	0	5	60	NULL	1
d_3	BLOB	1	0	5	60	NULL	1
d_4	BLOB	1	0	5	60	NULL	2
d_5	BLOB	1	0	5	60	NULL	3
d_6	BLOB	1	0	5	60	NULL	5
d_7	BLOB	1	0	5	60	NULL	4
d_8	BLOB	1	0	5	60	NULL	3
d_9	BLOB	1	0	5	60	NULL	3

13 rows fetched in 0:00.0673 Start Editing Apply Changes First Last Search

Query finished.

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools MySQL Enterprise Help

Back Next `SELECT * FROM quiz q LIMIT 0,1000` Execute Stop

QuizID	Version	Duration	QuizName	CourseID
1	1	30	3 Options	DEP102
2	1	30	4 Options	DEP102
3	1	30	5 Options	DEP102
4	1	30	6 Options	DEP102
5	1	30	MUL 5 Options	DEP102
6	1	30	MUL 6 Options	DEP102
7	1	30	Numeric	DEP102
8	1	30	TrueFalse	DEP102
9	1	30	YesNo	DEP102
10	1	30	MulOrder Quiz	DEP102
11	1	30	Mul Ques Quiz	DEP102
12	1	30	Mul Ques Quiz2	DEP102

12 rows fetched in 0:00.1295 Start Editing Apply Changes First Last Search

Query finished.

Schemata Bookmarks

- options
- polls
- profile
- question
- questioncatego
- quiz
- quizquestion
- quizquestionop

Syntax Functions Params Tr

- Data Definition Sta
- ALTER DATABAS
- ALTER TABLE Sy
- CREATE DATABA
- CREATE INDEX

Screenshots associated with Quiz Bank:

Indian Institute of Technology-Bombay

Clicker V3

Menu

- Raise Hand
- Poll
- Attendance
- Quiz Bank
- Question Bank
- View Report
- Profile
- Settings

Quiz ID

Quiz Time

Start Quiz

Find: last modified Previous Next Highlight all Match case Phrase not found

Screenshots associated with View Report:

The screenshot shows the top header with the IIT Bombay logo and name. On the left is a dark red menu with the following items: Menu, Raise Hand, Poll, Attendance, Quiz Bank, Question Bank, View Report, Profile, and Settings. The main content area is titled "Reports Login" in red. Below the title is a "Login" section with two input fields: "UserName" and "Password". At the bottom of the login section are two buttons: "Login" and "Cancel".

The screenshot shows the top header with the IIT Bombay logo and name. On the left is the same dark red menu as in the previous screenshot. The main content area features a row of five buttons: "Student Report", "Quiz Report", "Participant Rep", "Quiz/Student", and "Quiz". Below these buttons are three dropdown menus: "Select Institute" (with a placeholder "----Select Institute----"), "Department", and "Course". Underneath is a "Preview" section with two radio buttons: "Quiz" and "Student". At the bottom of the page is a large empty rectangular box.

Screenshots associated with Profile module:

The screenshot shows the Profile module interface for the Indian Institute of Technology-Bombay. On the left is a dark red menu with the following items: Menu, Raise Hand, Poll, Attendance, Quiz Bank, Question Bank, View Report, Profile, and Settings. At the top right, the IIT Bombay logo and name are displayed. Below the logo are four navigation buttons: Institution, Department, Course, and Instructor. A horizontal line separates this from the main content area. The main content area features a table with a Java error message above it: `java.lang.ClassNotFoundException: com.mysql.jdbc.Driver`. The table has four columns: Department ID, Department Name, Head of the Department, and Institution ID. Below the table are four form fields: Institute id, Department id, Dept Name, and Head Of the Department. At the bottom are five buttons: New, Add, Delete, Update, and Reset.

This screenshot shows the Profile module interface for the Indian Institute of Technology-Bombay, specifically the 'View Instructor' section. The layout is similar to the first screenshot, with the IIT Bombay logo and name at the top right, and navigation buttons (Institution, Department, Course, Instructor) below. The menu on the left is the same. The main content area has a section titled 'View Instructor' with a 'Select Institute' label and a 'Get Institute' button. Below this are labels for 'Department' and 'Course'. At the bottom are four buttons: View, Add New, Delete, and Update.

Menu

- Raise Hand
- Poll
- Attendance
- Quiz Bank
- Question Bank
- View Report
- Profile
- Settings

Institution
Department
Course
Instructor

Add Instructor

Select Institute

Department

Course

Name

Date of Join

Instructor ID



Indian Institute of Technology-Bombay

Menu

- Raise Hand
- Poll
- Attendance
- Quiz Bank
- Question Bank
- View Report
- Profile
- Settings

Profile

Admin Login

UserName

Password

Bibliography

- [1] www.python.org , Retrieved in July 2011
- [2] docs.python.org › *The Python Tutorial* , Retrieved in July 2011
- [3] <http://www.installjammer.com/support>, Retrieved in July 2011
- [4] <http://www.linux.com/archive/feed/60380> , Retrieved in July 2011
- [5] <http://sourceforge.net/projects/installjammer>, Retrieved in July 2011
- [6] <http://user.services.openoffice.org> , Retrieved in May and July 2011