CHAPTER 1

1. INTRODUCTION

1.1 Project overview

In this fast paced world, music is one of the thing that still become important to a lot of people. They always want to gain more access on music. Today, we see a lot of people use different type of music application. Music application that commonly used by people this day, usually only provide music player or music's lyrics while playing the song. These day, we only used application for user who enjoy music, so they mostly doesn't know anything about the chords in every music. Then, what about application for people who had more interest in music ?. That is why we would like to try bring more experienced for people who are interested with music chords.

This project is to create an Android-based application called ClarietEar. ClaritEar (/ˈklær.ə.tɪər/) is an educational and entertainment applications based on Android that are useful to train your hearing sensitivity in knowing the kind of tone or chord that is being played. With this application we hope people will gain more knowledge about music chords. Also we hope our application will be helping a lot of people on becoming new musician.

1.2 Project deliverables

During the course of this project, several documents and programs have to be produced and delivered.

- Preliminary Project Plan
- Software Project Management Plan
- Project Risk Management Plan (included in SPMP)
- Test Plan
- UML
- DD
- User Manual
- Final Document

1.3 Evolution of this document

This document will be updated as the project progresses. Update should be expected in the following sections.

a. ***References*** – updated as necessary

b. ***Definitions, acronyms, and abbreviations*** – updated as necessary.

c. ***Technical Process*** – will be revised as the requirements and design decisions become clearer.

d. ***Schedule*** – the schedule will be updated as the project progresses.

1.4 References

Several case studies used for the development of this project.

- http://www.nature.com/news/why-dissonant-music-strikes-the-wrong-chord-in-the-brain-1.11791
- http://listaka.com/top-10-most-difficult-musical-instruments/
- http://www.premierguitar.com/articles/19696-digging-deeper-how-many-chords-are-there

1.5 Definition, acronyms, and abbreviations

i. SPMP – Software Project Management Plan

ii. DD – Detailed Product

iii. UML – Unified Model Language

2. PROJECT ORGANIZATION

2.1 Process model

The project process will be based on Iterative Development Model, which is effective for dynamic software development, with the possibility to go back to the previous phase to dynamically improve and modify anything.

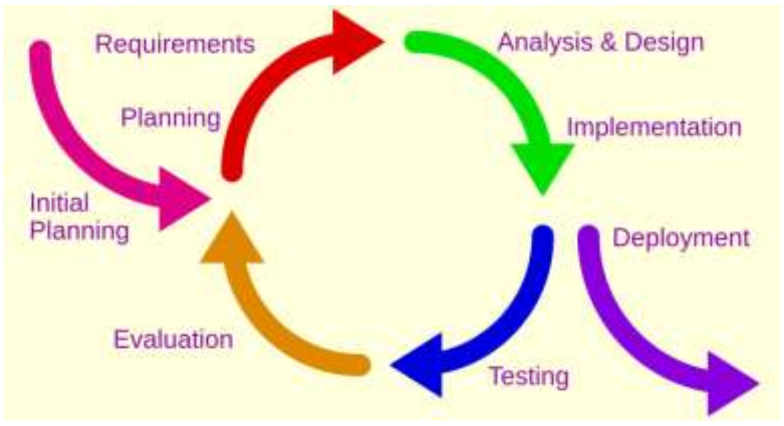


Figure 1 : Process Model

The project is divided in these phases:

1. Inception
2. Elaboration
3. Construction
4. Transition

Inception phase is the initial phase, which consist of initial planning of the project, and the planning of the whole project. Elaboration is the next phase, with the step of requirements analysis and the analysis of the project, with the initial design. The construction phase will be the longest phase in term of time, with the implementation and deployment step in it. The last phase, the transition phase, will be consisted of testing step and evaluation step, in order to validate the product and do the launching, while maintaining the product from the test results.

We will use Andorid Studio and UML tools to create the system model and the subsequent breakdown of the design. For this project, our group will be using Eclipse UML2 version Mars.

2.2 Organizational structure

Team members :

- Muhamad Harist Refian Anwar
- Andira Rozawati
- Pandu Wicaksono

Week/Deliverable	Team Leader	Deliverable Description
1	Muhamad Harist Refian A.	Project Plan
2	Andira Rozawati	Requirements Specification
3	Pandu Wicaksono	Analysis
4	Muhamad Harist Refian A.	Architecture Spec
5	Andira Rozawati	Component/Object Specification
6	Pandu Wicaksono	Source Code
7	Muhamad Harist Refian A.	Test Plan
8	Andira Rozawati	Final Deliverable

Table 1: Organizational Structure

This project employs three persons in which has their own main role and additional job. Main role is his/her position through this project and additional job is basically added in order to finish several technical things that may happen during doing this project. Also their additional job will accour when other team member needs help on doing their main job. There must be good communication between all employees during doing this task. As a team leader, he must be able to motivate and give solution when problem accour to another member and team leader also has to control the progress of the overall project.

2.3 Organizational boundaries and interfaces

Team Leader

Task: Perform all necessary activities to ensure that a task assigned to a team is performed well and on time. This includes but is not limited to:

• Planning and coordinating team activities;

• Providing feedback about team progress to the PM;

• Motivating team members;

• Chairing reviews of the items made by his team.

Team Member

Task: Perform all necessary activities to ensure that a task assigned to a team is performed well and on time. This includes but is not limited to:

• Assisting the Team Leader or Project Manager by signaling problems in an early stage;

• Executing plans made by the Team Leader and by the Project Manager;

• Keeping track of time spent on various tasks;

• Following procedures and plans.

2.4 Project responsibilities

Ultimately the entire project team is responsible for the successful delivery of the product. Team member assignments per deliverable according to expertise are :

1. Project Plan – Entire Team
2. Requirements Specification – Andira Rozawati
3. Analysis – Pandu Wicaksono
4. Architecture Spec – Muhamad Harist Refian Anwar
5. Component/Object Specification – Andira Rozawati
6. Source Code – Pandu Wicaksono
7. Test Plan – Muhamad Harist Refian Anwar
8. Final Deliverable – Entire Team

3. MANAGERIAL PROCESS

3.1 Management objectives and priorities

The objective of the project is to develop a mobile application system within allocated budget, time, and specified quality. The project is highly prioritized to help people in social life and also for our group’s project in Software Engineering course. The benefits will be further discussed in CBA (Cost benefit Analysis).

3.2 Assumptions, dependencies, and constraint

The project assumptions are as follows :

- Team of 3 resources
- Equipment and software availability
- Approval on Project
- Finding and Analizing The Software Content
- Programming The Software

The project dependencies are as follows

- The design of mobile application interface
- The content of it’s mobile application
- Programming using java
- Time
- Budget
- Man hours
- Availability of existing software

3.3 Risk management

This section mentions a number of possible risks for the project. Also, actions or measures are described to prevent or to reduce the risks. Four categories of risks are identified:

- Risks with respect to the work to be done;
- Risks with respect to the management;
- Risks with respect to the resources;
- Risks with respect to the customer.

The risk of each category will be explain below.

1. Risk With Respect to The Work to be done
   a. Programming
      Probability : High
      Prevention : Need to learn the programming language fastly

Corection : Finding more literature about the programming language our group use in the project. Finding similar code for reference. Asking seniors about our group difficulties in making the mobile application.

Impact : High

b. Time Shortage

Probility : High

Prevention : Care is taken to plan enough spare time.

Correction : When tasks are failed to be done or if tasks are finished earlier, then the planning schedule need to be adjust with the situation. If time shortage become severe, the content of the mobile application will be pinched.

Impact : High

c. Design

Probability : Low

Prevention : The design should be review very critically. We should consult the matter with the advisor or the customer it self about the design.

Correction : When errors in the design are noticed, we do some meeting to fixed the problem. Doing some research to used more appealing design for the customer.

Impact : Low

2. Risk With The Respect to Management

d. Sudden Absen of Member

Probability: Low

Prevention: There are very few things in which the presence of the member cannot be missed for a short period of time. Nevertheless the member will inform the other member of a planned period of absence in time so that the other member can prepare to take over.

Correction: By keeping the other member up-to-date on the project status we will have enough knowledge to take over in case of illness or absence of the member.

Impact: Low

3. Risk With The Respect to The Resources
    e. Unavailability of the advisor when needed

       Probability: Medium

       Prevention: Meetings with the advisor can be planned in advance and time has been reserved in their schedule for counseling the content our mobile application.

       Correction: A different appointment is made, or another expert is consulted.

       Impact: Medium

4. Risk With The Respect to The Customer
    f. The customer changed their mind

       Probability: High

       Prevention: It is obviously explained to the customer, that after he has accepted a version of the purpose and content of the project, it cannot be changed by the customer's wish only.

       Correction: If the customer changes his mind during the user requirement phase his new requirements can be incorporated.

       Impact: Low
    g. The Customer is not available when needed

       Probability: Medium

       Prevention: Meetings with the customer can be planned well in advance. The customer has been given room in his schedule for his Software Engineering related work.

       Correction: When the customer is not available, meetings may have to be rescheduled.

       Impact: Medium

5. Summary

    It is obvious that problems will occur during the project. To avoid problems the following rules should be followed by all team members:

    - Try to signal problems as early as possible and meeting with the member as soon as possible, so that action can be taken;

- Pay attention to communication and make sure everybody understands the things the same way;
- Focus on the agreed user requirements, which express the wishes of the customer;
- Minimize friction between people by helping and supporting each other;

3.4 Monitoring and Controlling Mechanism

Weekly Project Group Meetings

The project group meetings take place in Faculty of Engineering, University of Indonesia. Project group meetings usually take place on Thursday at 11 AM, although this time may be subject to change, e-mails or chats will be send about the time if it changes. These meetings are meant to inform each other of the progress made on various tasks. Before the meeting, all members read minutes of previous meeting. The member takes care of the agenda and presides the meeting.

Progress Meetings

These meetings are scheduled biweekly at 11:00 AM. On these meetings the member of the group project will meet together to discuss some problem or to inform the progress each member have done. Before progress meetings the following things need to be done:

- Write a progress report after the example of the previous reports;
- Read the minutes of the previous meeting;
- Deliver the report to the other member half an hour before the start of the first meeting on that day.

Project Log

Each of members (team leader and team member) should be filled their log after weekly internal meeting in order to be controlled by PM. The following table is used to show the reporting and communication plan for the project.

Information Communicated	From	To	Time Period
Status report	Project Team Leader	Member of The Project	Weekly
Status report	Member with Programming as their main job	Team Leader and Other Member	Weekly
Project Review	Group Project	Customer	Undefined / Flexible
Project Log	All Members	Advisor	Weekly

Table 2: Communication and Reporting Plan

3.5 Staff Plan

The following table contains contact information about the members of the Socializer project group:

Name	Email	Phone	Function
Muhamad Harist Refian Anwar	muhamad.harist@ui.ac.id	083895222109	Group Leader and Design
Pandu Wicaksono	pandu.wicaksono31@ui.ac.id	08174930665	Programming
Andira Rozawati	andira.rozawati@ui.ac.id	088210740164	Content and Managerial

Table 3: Staff Planning

4. TECHNICAL PROCESS

4.1 Methods, Tools and Techniques

In this project, we design using process model that has been inlcuded in section 2.1. The project will be implemented utilizing java programming languange, and tools such as Java JDK, Visio, Trello, Android Studio, Photoshop and Marvel will be utilized. The object oriented analysis technique will be used to successfully complete the project. ClaritEar is mobile application that train people's hearing sensitivity in knowing the kind of tone or chord that is being played. With this application we hope people will gain more knowledge about music chords. Also we hope our application could help people on becoming new musician.

4.2 Software Documentation

During the project, documents should conform to a number of aspects:

- Documents must be of good quality.

The standards all documents are required to meet are documented in with respect to style and with respect to content.

- Documents must be reviewed.

The manner in which document reviews by the advisor or by the group project themselves.

- The purpose of document reviews is to get docs of high quality.

The requirements which apply to the approval of documents are given to Software Engineering lecture.

4.3 User Documentation

Since the user of this software will have different perspectives from one to another,so the user documentation has to be very clear and easily understand by them. And that document should conform several aspects like: Have a good quality and must be reviewed. Also inside user documentation should explain user manual about how to use the software.

4.4 Infrastructur Plan

The hardware resources are all the member of the group project computers running windows 7 / 8 / 10 Operating System. Each of these computers should at least already installing the tools that helped making the project.

5. WORK ELEMENTS, SCHEDULE, AND BUDGETS

- The project is budgeted for 3 resources, and equipments needed to complete analysis, implementation, and the test of the application
- The project lead will be rotated for each phase out of 3 team members.
- The document for all phases will be revised in subsequent phases if applicable.

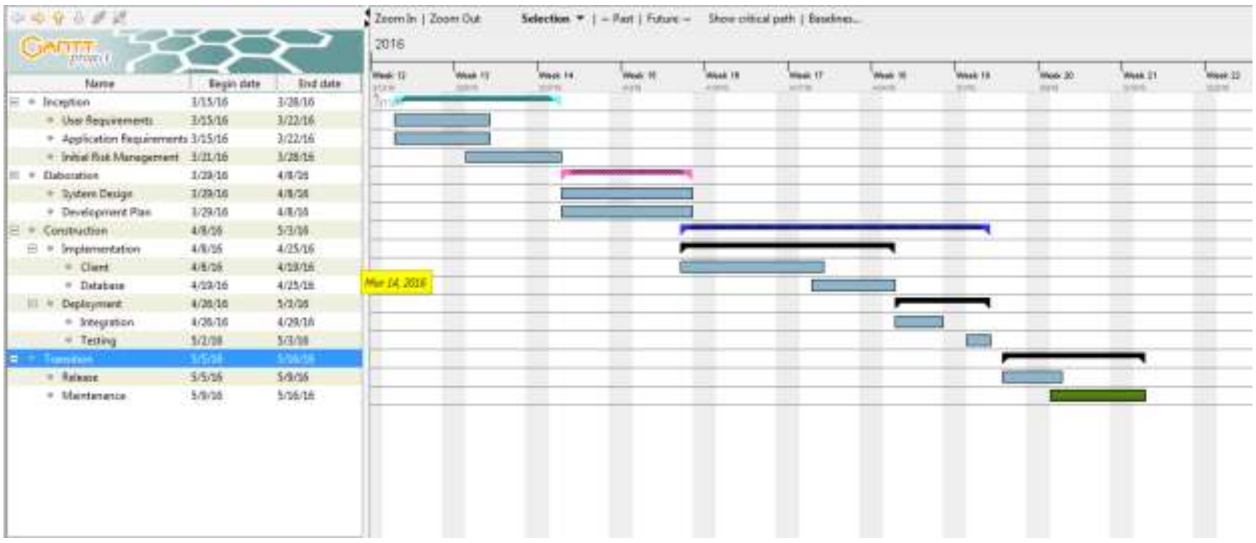
5.1 Schedule

The following is the schedule that we will do in our ClaritEar Mobile Application project :

- The team budget of 3 persons **x** 1,464 hours = 4,392 hours;
- The project deadline of May 24th 2016;
- The final presentation of June 03rd 2016;

5.2 Timeline

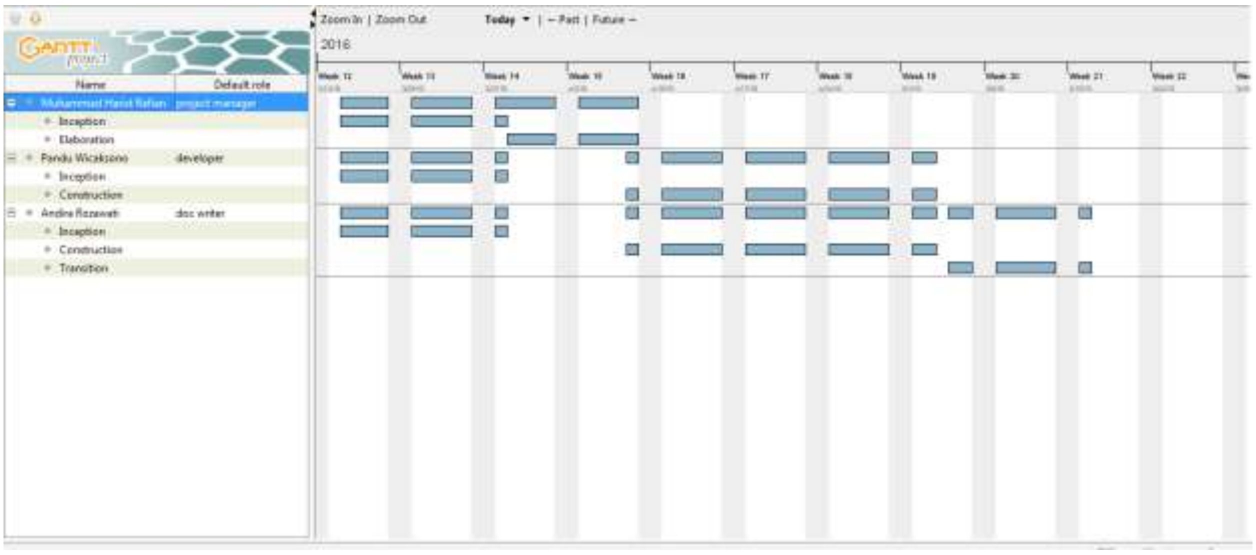
The following is our activities in the making of the application :



Name	Begin date	End date
Inception	3/15/16	3/28/16
User Requirements	3/15/16	3/22/16
Application Requirements	3/15/16	3/22/16
Initial Risk Management	3/21/16	3/28/16
Elaboration	3/29/16	4/8/16
System Design	3/29/16	4/8/16
Development Plan	3/29/16	4/8/16
Construction	4/8/16	5/3/16
Implementation	4/8/16	4/25/16
Client	4/8/16	4/19/16
Database	4/19/16	4/25/16
Deployment	4/26/16	5/3/16
Integration	4/26/16	4/29/16
Testing	5/2/16	5/3/16
Transition	5/5/16	5/16/16
Release	5/5/16	5/9/16
Maintenance	5/9/16	5/16/16

Zoom In | Zoom Out
Selection | ← Past | Future →
Show critical path | Baselines...
2016
Mar 14, 2016

Figure 2. Schedule



Name	Default role
Muhammad Hanif Rafian	project manager
Inception	
Elaboration	
Pandu Wicaksono	developer
Inception	
Construction	
Andina Rozawati	doc writer
Inception	
Construction	
Transition	

Figure 3. Resources Figure

5.3 Resources Requirement

The most important resource during the project is human resources. The software that is needed during the process of making this project are Adobe Flash or Construct 2, Photoshop, Adobe Illustrator and Corel Draw. During the project software development is required, for example programming language used in Adobe Flash and Construct 2. Other resources needed include development stations, a store where documents and information can be backed up, a printer, network connectivity, a working and meeting room with chairs. During the project software is required like text editor and compiler for any programming language.

6. ADDITIONAL COMPONENTS

6.1 Index

Index is not identified yet for the first draft of SPMP. Any changes are subjected soon.

6.2 Appendices

Appendices is not identified yet for the first draft of SPMP. Any changes are subjected soon.

CHAPTER 2

DESIGN

A. UML Diagram

To fully understand about ClaritEar project, we make some UML Diagram to help people know our project better in many perspectives. UML Diagram is basically used for designing our project's system. In our project, which is 'ClaritEar Mobile Application', we made some UML Diagram that can help people the architecture of 'ClaritEar Mobile Application', here are the diagram :

1. Use Case Diagram :

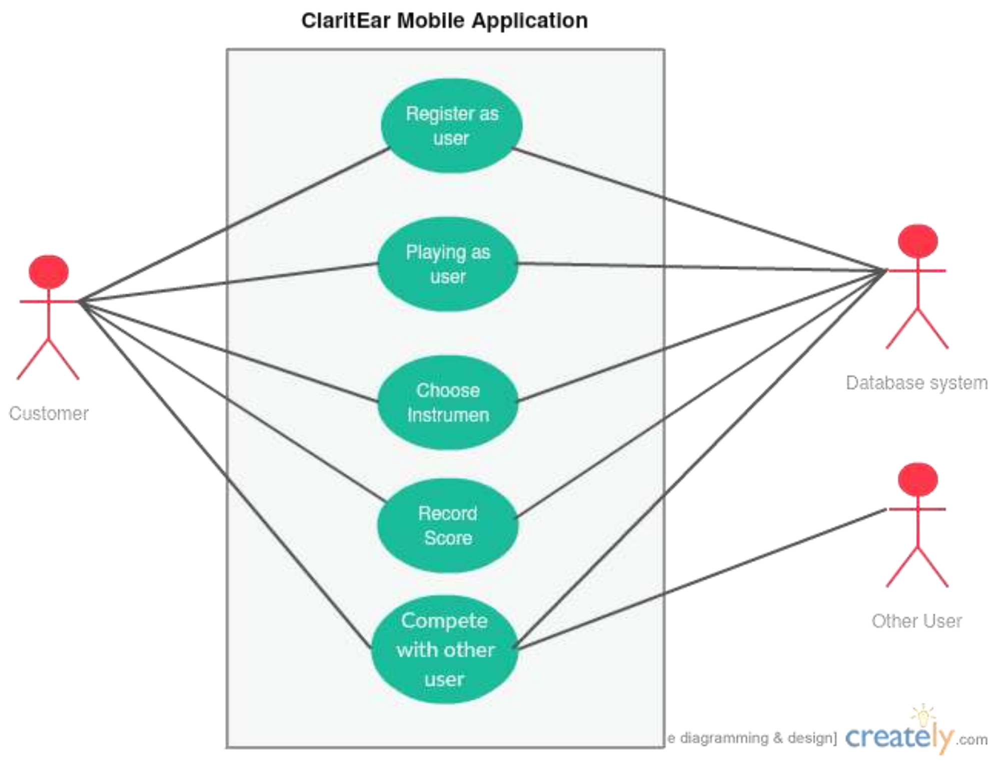


Figure 4. Use Case Diagram

Use case diagram is a graphic depiction of the intersection among the element of a system. A use case is methodology used in system analysis to identify, clarify, and organized system requirements. It can be used to describe the functionality of a system in horizontal way. In use case diagram, there are 4 major elements: the actors that the system you are describing interacts with, the system itself, the use cases or services that the system knows how to perform, and the last is the lines that represent relationship between these elements.

In use case diagram that we have made in figure 1, we think that user can register as ClaritEar mobile application's user and play the games by guessing the chord which the user already choose what instrument to play the games. The application also can record the high scores and update the scores to the database and compare it to other user's hisgh scores.

2. Activity Diagram :

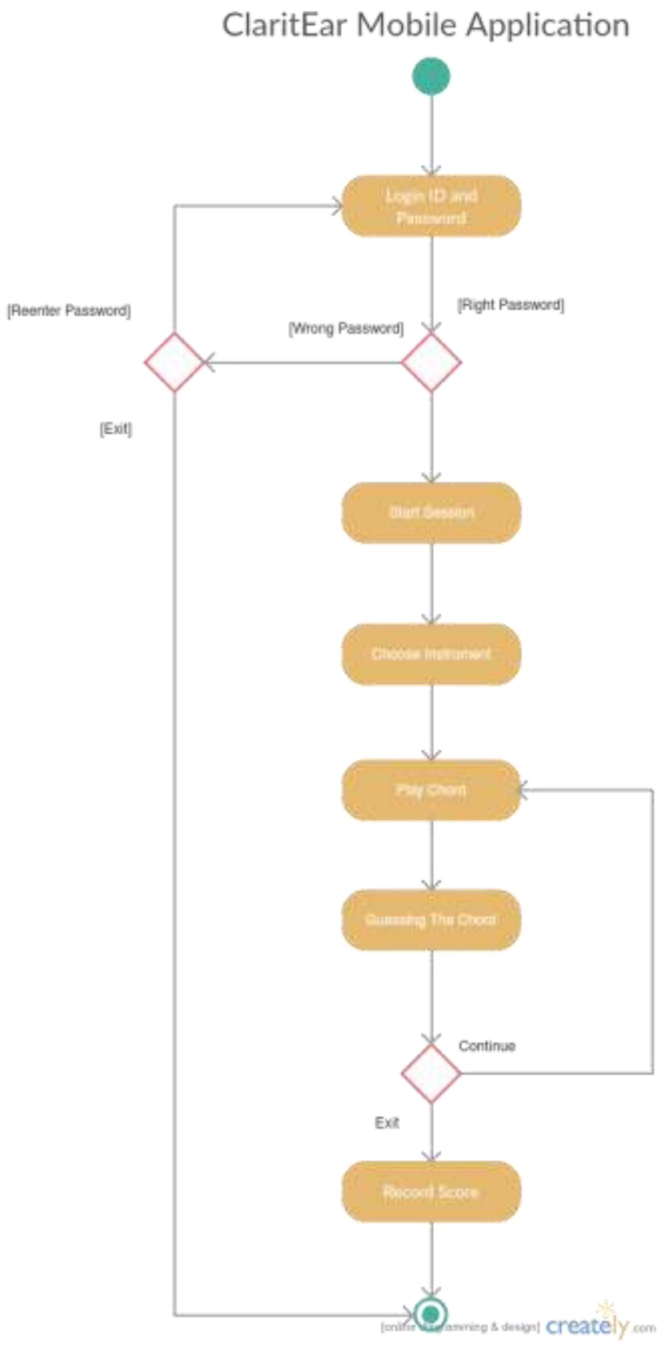


Figure 5. Activity Diagram

Activity diagram is UML behavior diagram which shows flow of control or object flow with emphasis on the sequence and conditions of the flow. The actions coordinated by activity models can be initiated because other actions finish executing, because objects and data become available, or because some events external to the flow occur. In activity diagram there are some following nodes and edges that typically drawn on it: activity, partition, action, object, control, activity edge.

3. Sequential Diagram :

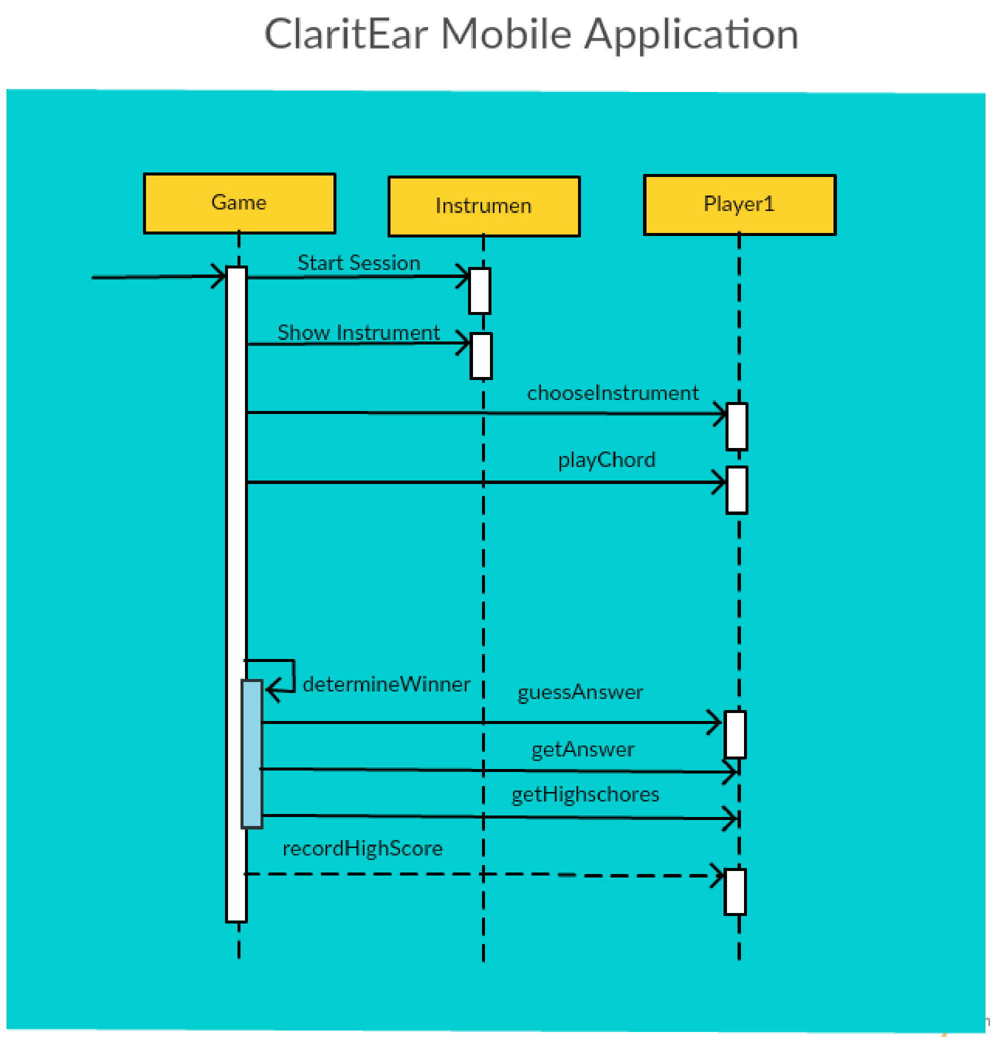


Figure 6. Sequencial Diagram

Sequential diagram is UML behavior diagram which shows flow of control or object flow with emphasis on the sequence and conditions of the flow. The actions coordinated by how the program will execute when user needed. Sequential diagram show the connection between the content in our project.

4. Class Diagram :

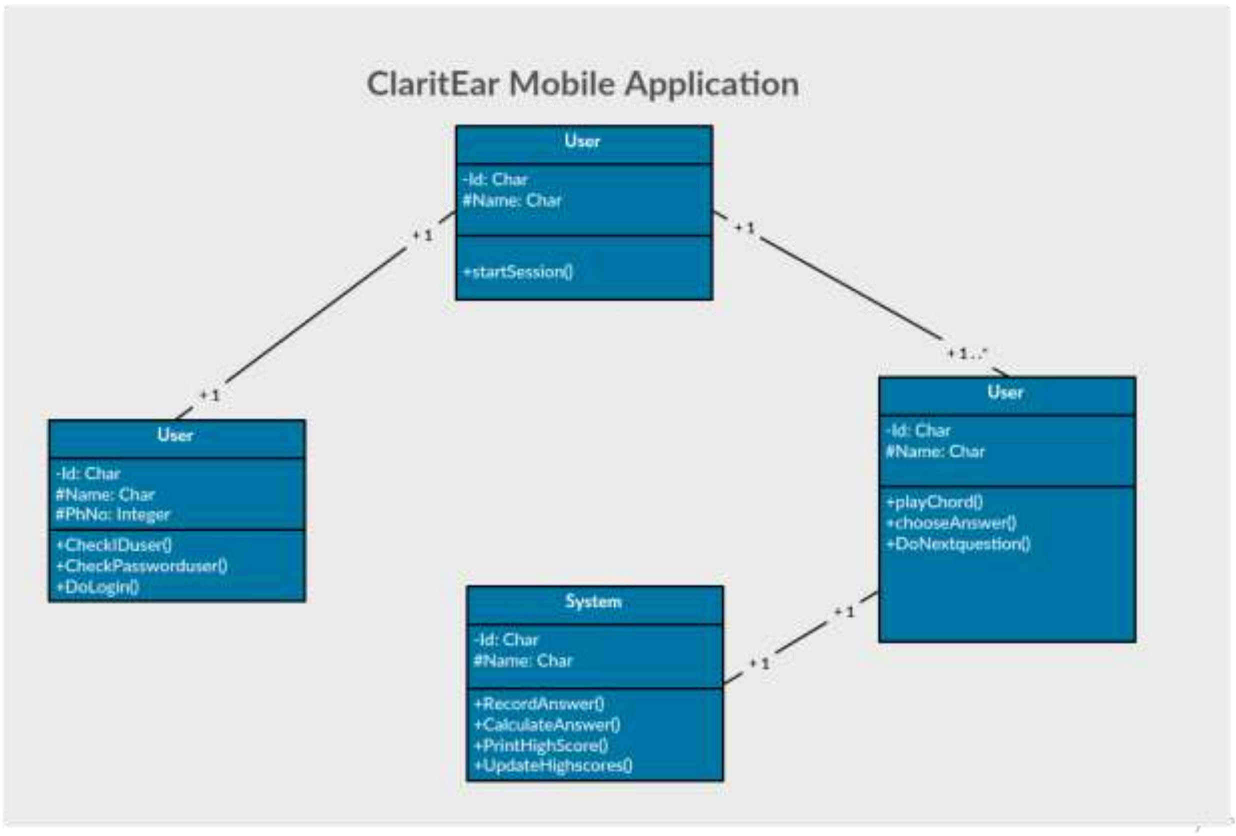


Figure 7. Class Diagram

The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction. The UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application but class diagram is a bit different. So it is the most popular UML diagram in the coder community.

5. Deployment Diagram :

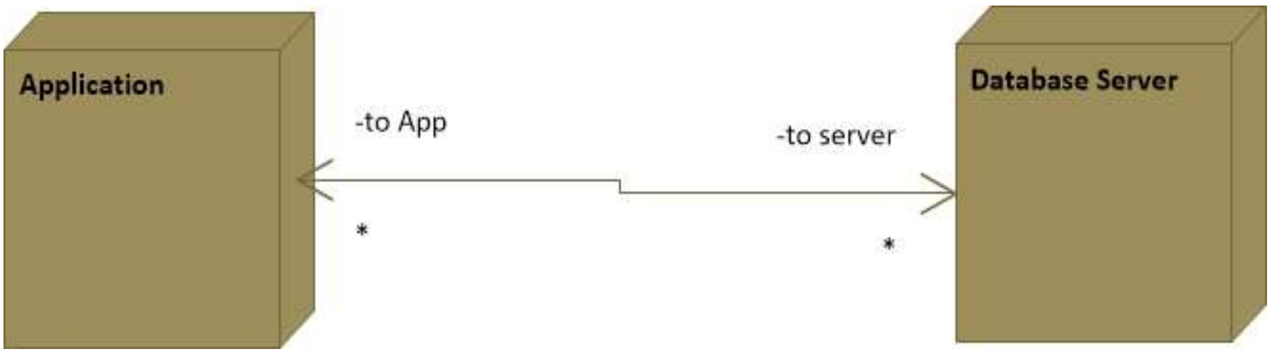


Figure 8. Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. In ClaritEar mobile application, the system will generate the selection situation from user to database system.

6. Package Diagram :

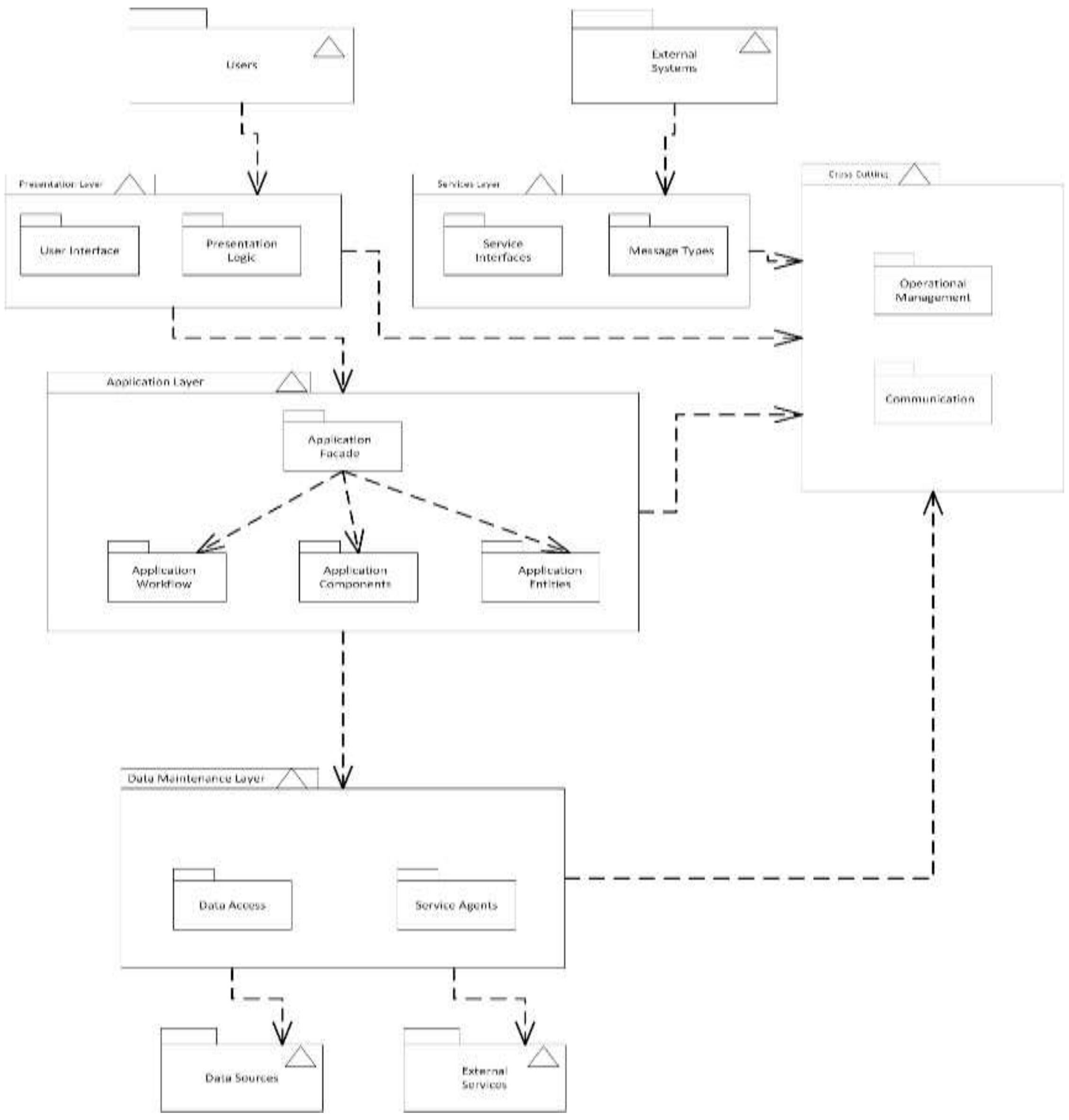


Figure 9. Package Diagram

Package diagram is UML structure diagram which shows packages and dependencies between the packages. This type of model diagram allow to show different views of a system, for example, as multi-layered application.

7. Component Diagram

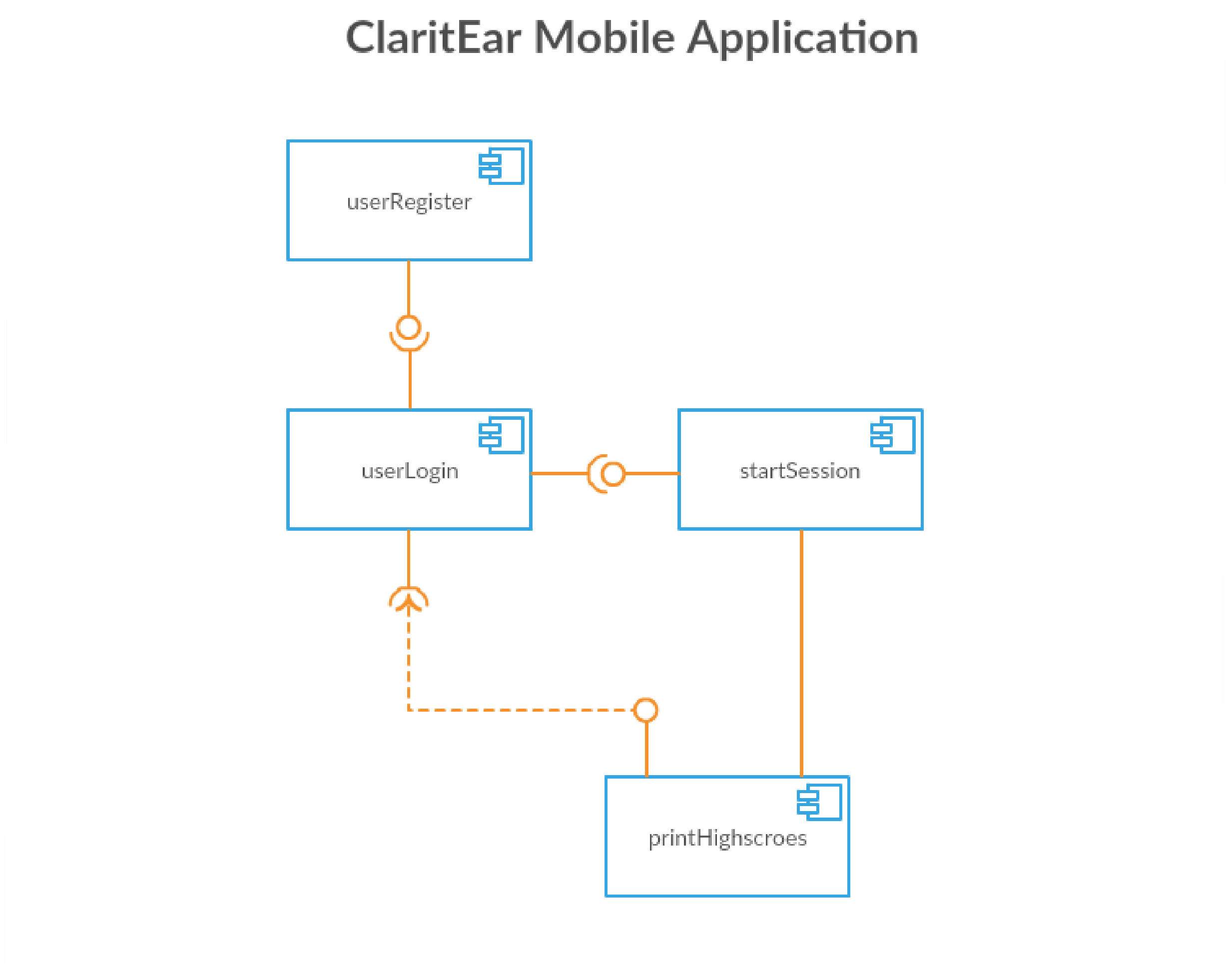


Figure 10. Component Diagram

The purpose of the component diagram can be summarized as:

1. Visualize the components of a system.
2. Construct executables by using forward and reverse engineering.
3. Describe the organization and relationships of the components.

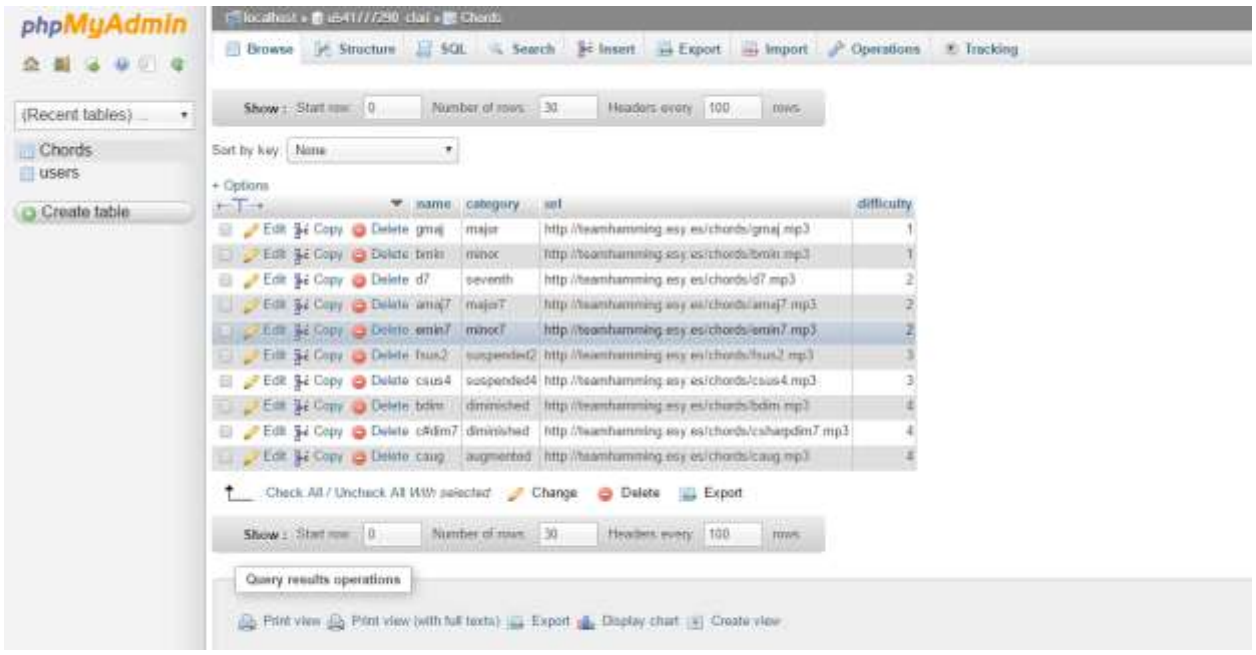
In ClaritEar mobile application, we have 3 package in the system. First we had interface package for user. So user will easily use the application. Second is get situation packaga, where user can select situation. Last one is database access to get sample question for situation that every user choose.

CHAPTER 3

IMPLEMENTATION AND SOURCE CODE

A. Programming and Database

To implement our project, we used Android Studio IDE to make the program and Graphical User Interface. For generate our session in choosing instrument we used database from idhostinger.com. idhostinger also provide with phpmyadmin and MySQL to do the query in selecting the right tabel in database. Our database will contain username and password for all user, music chords, and data of all highscores.

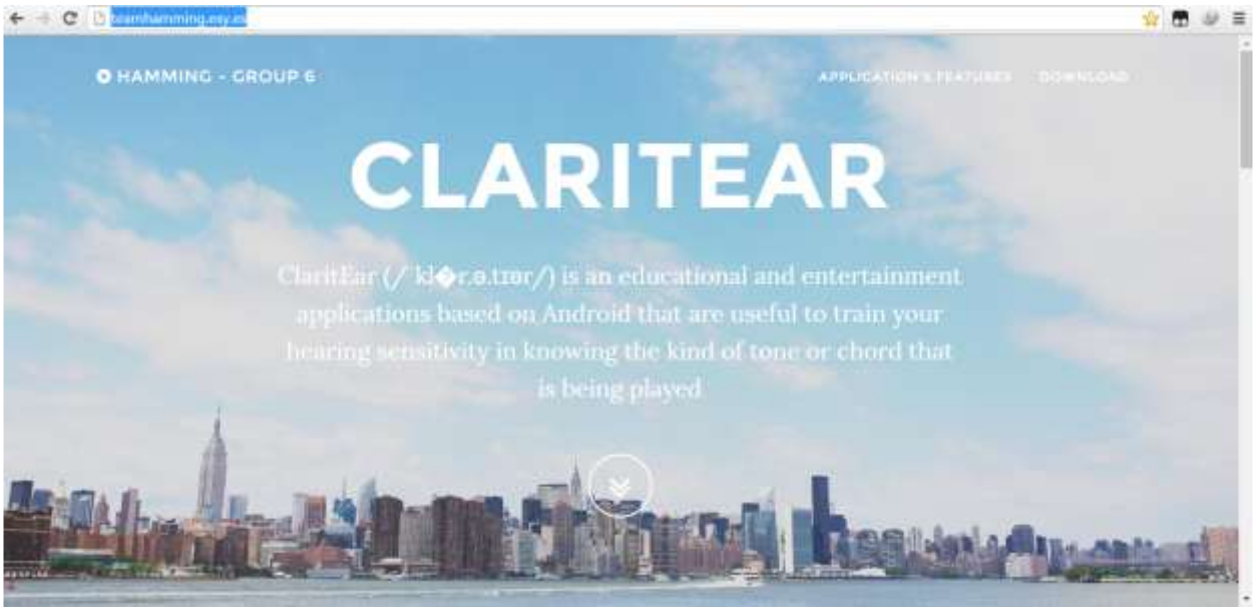


	name	category	url	difficulty
Edit Copy Delete	gmaj	major	http://teamhamming.esy.es/chords/gmaj.mp3	1
Edit Copy Delete	bmin	minor	http://teamhamming.esy.es/chords/bmin.mp3	1
Edit Copy Delete	d7	seventh	http://teamhamming.esy.es/chords/d7.mp3	2
Edit Copy Delete	amaj7	major7	http://teamhamming.esy.es/chords/amaj7.mp3	2
Edit Copy Delete	emin7	minor7	http://teamhamming.esy.es/chords/emin7.mp3	2
Edit Copy Delete	fsus2	suspended2	http://teamhamming.esy.es/chords/fsus2.mp3	3
Edit Copy Delete	csus4	suspended4	http://teamhamming.esy.es/chords/csus4.mp3	3
Edit Copy Delete	bdim	diminished	http://teamhamming.esy.es/chords/bdim.mp3	4
Edit Copy Delete	c#dim7	diminished	http://teamhamming.esy.es/chords/csharpdim7.mp3	4
Edit Copy Delete	caug	augmented	http://teamhamming.esy.es/chords/caug.mp3	4

Check All / Uncheck All With selected: Change Delete Export
Show : Start row: 0 Number of rows: 30 Headers every 100 rows
Query results operations
Print view
Print view (with full texts)
Export
Display chart
Create view

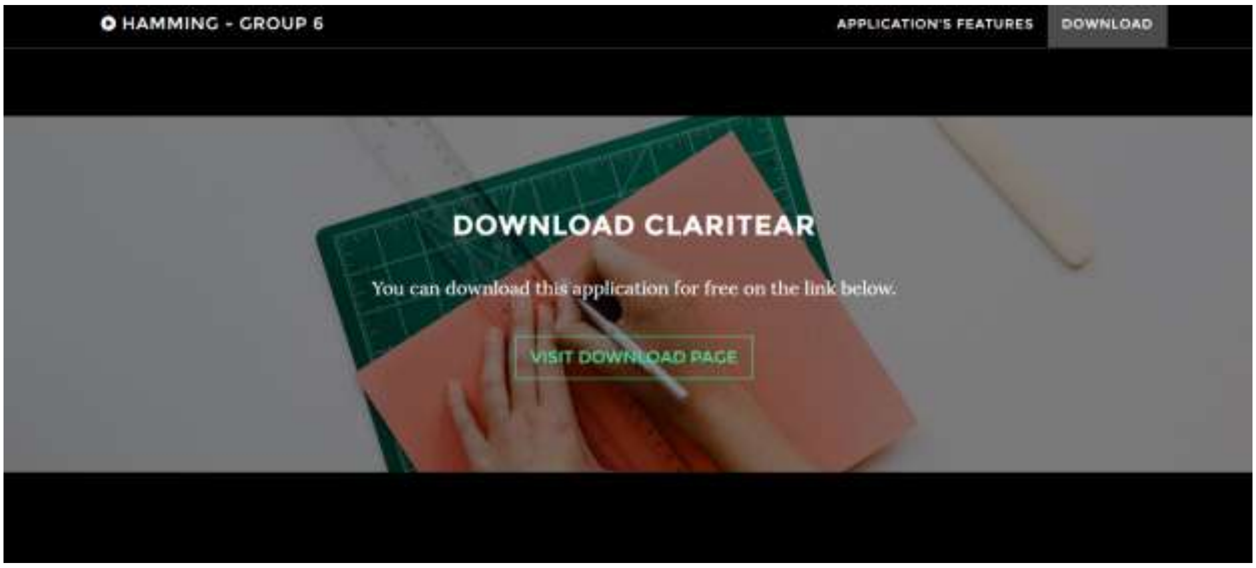
Figure 11. Database ClaritEar

We also provide our customer with a website so it will helps them to give feedback to our product. Also we provide a link for downloading our application in the website. We also give a contact to communicate with our team. The website is http://teamhamming.esy.es/.



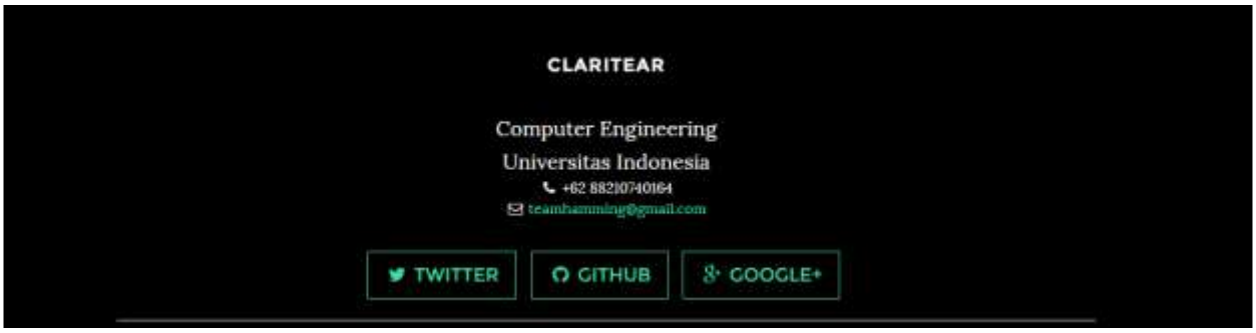
ClaritEar (/ˈkl�r.ə.tɪər/) is an educational and entertainment applications based on Android that are useful to train your hearing sensitivity in knowing the kind of tone or chord that is being played

Figure 12. Homepage of ClaritEar



You can download this application for free on the link below.

Figure 13. Donwlod Page



CLARITEAR

Computer Engineering
Universitas Indonesia
+62 88210740164
teamhamming@gmail.com

Figure 14. Contact Page

B. Source Code

- Code for Choosing Instrument

```
package com.hamming.claritear;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.ImageView;

/**
 * Class untuk memilih jenis instrument gitar atau piano
 */

public class ChooseInstrument extends AppCompatActivity {
    ImageView guitar, piano;

    /**
     * method override untuk mendeclare segala komponen pada file
xml ke file java
     * @param savedInstanceState
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_choose_instrument);
        guitar = (ImageView)findViewById(R.id.iv_guitar_icon);
        piano = (ImageView)findViewById(R.id.iv_piano_icon);
    }

    /**
     * method yang dipanggil saat tombol gitar dipilih yang akan
mengirim user ke playing
     * activty gitar
     * @param view
     */
    public void playGuitar(View view){
        Intent guitarIntent = new
Intent(ChooseInstrument.this,PlayingActivity.class);
        startActivity(guitarIntent);
        finish();
    }
    /**
     * method yang dipanggil saat tombol piano dipilih yang akan
mengirim user ke playing activty piano
     * @param v
     */
    public void playPiano(View v){
        Intent pianoIntent = new
Intent(ChooseInstrument.this,PlayingPianoActivity.class);
        startActivity(pianoIntent);
        finish();
    }

    /**
     * method override untuk kembali ke activity home saat tombol
back pada handphone ditekan
     */
    @Override
    public void onBackPressed() {
        Intent toHome = new Intent(ChooseInstrument.this,
HomeActivity.class);
        startActivity(toHome);
        finish();
    }
}
```

- Source code for calculating highscores

```
package com.hamming.claritear;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import com.kosalgeek.genasync12.AsyncResponse;
import com.kosalgeek.genasync12.PostResponseAsyncTask;

import java.util.HashMap;

/**
 * Class untuk menampilkan highscore baru yang diraih user
 */
public class HighscoreActivity extends AppCompatActivity {
    TextView newhighscore;
    Button share;
    Button noshare;
    private String highscore;
    private String sharedUser;

    /**
     * method override yang akan menampilkan hal-hal yang
dibutuhkan dan mendeclare componen pada
     * file xml ke file java
     * @param savedInstanceState
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_highscore);
        newhighscore = (TextView)
findViewById(R.id.tv_newhighscore);

        //Mengambil data username langsung dari loginInfo
        SharedPreferences sharedPrefLogin =
getSharedPreferences("loginInfo", Context
                .MODE_PRIVATE);
        sharedUser = sharedPrefLogin.getString("username", "");

        //Mengambil data dari database berdasarkan username
        HashMap postShared = new HashMap();
        postShared.put("username",sharedUser);

        //Mengambil data highscore dari database
        String urlhighscore =
"http://teamhamming.esy.es/FetchHighscore.php";
        PostResponseAsyncTask fetchHighScore = new
PostResponseAsyncTask(this, postShared, false, new AsyncResponse()
{
            @Override
            public void processFinish(String result3) {
                newhighscore.setText(result3);
                highscore = result3;
            }
        });
        fetchHighScore.execute(urlhighscore);
        share = (Button) findViewById(R.id.btn_share);
        noshare = (Button) findViewById(R.id.btn_noshare);
    }

    /**
     * method yang dipanggil apabila tombol share ditekan yang
mengirim user ke tab share melalui
     * media sosial
     * @param v
     */
    public void clickShare(View v){
        Intent shareIntent = new Intent();
        shareIntent.setAction(Intent.ACTION_SEND);
        shareIntent.putExtra(Intent.EXTRA_TEXT,sharedUser + " has
achieved new highscore of " + highscore
                + " point(s) in ClaritEar.\n\n" +
                "Go download ClaritEar and train your ear here
www.teamhamming.esy.es");
        shareIntent.setType("text/plain");
        shareIntent.createChooser(shareIntent, "Share via");
        startActivity(shareIntent);
    }

    /**
     * dipanggil apabila user tidak ingin meng-share yang akan
mengirim ke activity home
     * @param v
     */
    public void clickNoShare(View v){
        Intent backHome = new Intent(HighscoreActivity.this,
HomeActivity.class);
        startActivity(backHome);
        finish();
    }

    /**
     * override yang akan mengirim user ke home apabila tombol back
pada handphone ditekan
     */
    @Override
    public void onBackPressed() {
        Intent toHome = new Intent(HighscoreActivity.this,
HomeActivity.class);
        startActivity(toHome);
        finish();
    }
}
```

- Source code for homeactivity

```
package com.hamming.claritear;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import com.kosalgeek.genasync12.AsyncResponse;
import com.kosalgeek.genasync12.PostResponseAsyncTask;

import java.util.HashMap;

/**
 * Kelas HomeActivity
 */
public class HomeActivity extends AppCompatActivity {
    public static TextView detailusername, detailhighscore;
    Button start;

    /**
     * method override untuk mendeclare komponen pada xml ke java.
     * Juga dibutuhkan untuk menampilkan hal-hal yang perlu
ditempilkan seperti informasi user dll
     * @param savedInstanceState
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        detailusername =
(TextView)findViewById(R.id.tv_main_username);
        detailhighscore =
(TextView)findViewById(R.id.tv_main_highscore);
        start = (Button)findViewById(R.id.btn_main_start);

        //Mengambil data username langsung dari loginInfo
        SharedPreferences sharedPrefLogin =
getSharedPreferences("loginInfo", Context
                .MODE_PRIVATE);
        String sharedUser = sharedPrefLogin.getString("username",
"");
        detailusername.setText(sharedUser + "!");

        //Mengambil data dari database berdasarkan username
        HashMap postShared = new HashMap();
        postShared.put("username",sharedUser);

        //Mengambil data highscore dari database
        String urlhighscore =
"http://teamhamming.esy.es/FetchHighscore.php";
        PostResponseAsyncTask fetchHighScore = new
PostResponseAsyncTask(this, postShared, new AsyncResponse() {
            @Override
            public void processFinish(String result3) {
                detailhighscore.setText(result3);
            }
        });
        fetchHighScore.execute(urlhighscore);
    }

    /**
     * method startPlaying untuk mulai bermain jika tombol "start"
ditekan
     */
    public void startPlaying(View v){
        Intent startPlay = new Intent(HomeActivity.this,
ChooseInstrument.class);
        startActivity(startPlay);
        finish();
    }

    /**
     * method override onBackPressed untuk keluar saat tombol back
pada hardware ditekan
     */
    @Override
    public void onBackPressed() {
        super.onBackPressed();
        appExit();
    }

    /**
     * method yang akan menampilkan menu dropdown pada pojok kanan
atas layar
     * @param menu
     * @return
     */
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater menuinflater = getMenuInflater();
        menuinflater.inflate(R.menu.home_activity_menu, menu);
        return super.onCreateOptionsMenu(menu);
    }

    /**
     * method yang digunakan untuk menentukan apa yang dilakukan
pada setiap opsi yang tertera
     * pada menu dropdown
     * @param item
     * @return
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()){
            case (R.id.share):
                Intent shareIntent = new Intent();
                shareIntent.setAction(Intent.ACTION_SEND);
                shareIntent.putExtra(Intent.EXTRA_TEXT, "I am now
using ClaritEar to train my ear and musical ability\n\n" +
                        "Go get and download ClaritEar for free
here www.teamhamming.esy.es");
                shareIntent.setType("text/plain");
                shareIntent.createChooser(shareIntent, "Share
via");
                startActivity(shareIntent);
                break;
            case (R.id.leaderboard):
                Intent leaderIntent = new Intent(HomeActivity.this,
LeaderActivity.class);
                startActivity(leaderIntent);
                break;
            case(R.id.about):
                Uri uriUrl =
Uri.parse("http://www.teamhamming.esy.es");
                Intent launchBrowser = new
Intent(Intent.ACTION_VIEW, uriUrl);
                startActivity(launchBrowser);
                break;
            case(R.id.logout):
                SharedPreferences sharedPrefLogin =
getSharedPreferences("loginInfo", Context
                        .MODE_PRIVATE);
                //Menghapus memory loginInfo shared preferences
                SharedPreferences.Editor editor =
sharedPrefLogin.edit();
                editor.clear();
                editor.apply();

                //Kembali ke halaman Login
                Intent logout = new Intent(HomeActivity.this,
Login.class);
                startActivity(logout);
                finish();
                break;

            case (R.id.exit):
                appExit();
                break;
            default:
                return super.onOptionsItemSelected(item);
        }
        return true;
    }

    /**
     * method appExit yang akan dipanggil saat ingin keluar dari
app
     */
    public void appExit(){
        this.finish();
        Intent exit = new Intent(Intent.ACTION_MAIN);
        exit.addCategory(Intent.CATEGORY_HOME);
        exit.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(exit);
    }
}
```

- Source code for choosing leader from all user

```
package com.hamming.claritear;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.webkit.WebView;

/**
 * kelas untuk menampilan leaderboard
 */
public class LeaderActivity extends Activity {

    /**
     * method untuk menampilan textview yang berisi leaderboard
     * @param savedInstanceState
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_leader);
        String url = "http://teamhamming.esy.es/leader.php";
        WebView wvleader =
(WebView)findViewById(R.id.wv_leaderboard);
        wvleader.loadUrl(url);
    }
}
```

- Source code for login

```
package com.hamming.claritear;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.kosalgeek.genasync12.AsyncResponse;
import com.kosalgeek.genasync12.PostResponseAsyncTask;

import java.util.HashMap;

/**
 * Kelas Login untuk activity Login
 */
public class Login extends AppCompatActivity {
    EditText inputUsername, inputPassword;
    Button login;

    /**
     * Method onCreate untuk mendeklarasi button, edittext, dan
textview
     * @param savedInstanceState
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        inputUsername =
(EditText)findViewById(R.id.et_login_username);
        inputPassword =
(EditText)findViewById(R.id.et_login_password);
        login = (Button)findViewById(R.id.btn_login_login);
    }

    /**
     * Method login untuk login saat tombol login ditekan
     * @param v
     */
    public void login(View v){
        HashMap postDataLogin = new HashMap();
        postDataLogin.put("usernamelogin",
inputUsername.getText().toString());
        postDataLogin.put("passlogin",
inputPassword.getText().toString());


if(inputUsername.getText().toString().equals("")||inputPassword.get
Text().toString().equals("")){
            Toast.makeText(this, "One or more field(s) are empty",
Toast.LENGTH_LONG).show();
        }
        else{
            String urlLogin =
"http://teamhamming.esy.es/Login.php";
            PostResponseAsyncTask loginTask = new
PostResponseAsyncTask(this, postDataLogin, new AsyncResponse() {
                @Override
                public void processFinish(String result2) {
                    switch(result2){
                        case "true":
                            //Menyimpan info login berupa username
dengan SharedPreferance
                            SharedPreferences sharedPrefLogin =
getSharedPreferences("loginInfo", Context
                                    .MODE_PRIVATE);

                            SharedPreferences.Editor editor =
sharedPrefLogin.edit();
                            editor.putString("username",
inputUsername.getText().toString());
                            editor.apply();
                            Intent login = new
Intent(Login.this,HomeActivity.class);
                            startActivity(login);
                            finish();
                            break;

                        case "false":
                            Toast.makeText(Login.this, "Invalid
password or username", Toast.LENGTH_LONG)
                                    .show();
                            break;

                        default:
                            Toast.makeText(Login.this,"Connection
error. Please try again", Toast.LENGTH_LONG)
                                    .show();
                            break;
                    }

                }
            });
            loginTask.execute(urlLogin);
        }
    }

    /**
     * Method toRegister untuk pindah ke activity register saat
menekan text "Make new account"
     * @param v
     */
    public void toRegister(View v){
        Intent toRegister = new
Intent(Login.this,Registration.class);
        startActivity(toRegister);
        finish();
    }

    /**
     * Method onBackPressed untuk keluar dari program saat tombol
back ditekan
     */
@Override
    public void onBackPressed() {
        super.onBackPressed();
        this.finish();
        Intent exit = new Intent(Intent.ACTION_MAIN);
        exit.addCategory(Intent.CATEGORY_HOME);
        exit.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
```

```
            startActivity(exit);
        }
    }
```

- Source Code for playing

```
package com.hamming.claritear;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.os.SystemClock;
import android.os.Vibrator;
import android.support.v7.app.AppCompatActivity;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.kosalgeek.genasync12.AsyncResponse;
import com.kosalgeek.genasync12.PostResponseAsyncTask;

import java.util.HashMap;

/**
 * Class untuk bermain dengan instrument guitar
 */
public class PlayingActivity extends AppCompatActivity {
    TextView score;
    Button play;
    Button maj;
    Button min;
    Button seventh;
    Button min7;
    Button maj7;
    Button dim;
    Button dim7;
    Button aug;
    Button sus2;
    Button sus4;
    private String correctAns;
    public String chordURL;
    public String category;
    public int currentscore;
    public String highscore;

    /**
     * untuk mendeclare seluruh komponen pada xml
     * @param savedInstanceState
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_playing);
        score = (TextView) findViewById(R.id.tv_play_printscore);
        play = (Button) findViewById(R.id.btn_play_playChord);
        maj = (Button)findViewById(R.id.btn_maj);
        min = (Button)findViewById(R.id.btn_min);
        seventh = (Button)findViewById(R.id.btn_seventh);
        min7 = (Button)findViewById(R.id.btn_min7);
        maj7 = (Button)findViewById(R.id.btn_maj7);
        dim = (Button)findViewById(R.id.btn_dim);
        dim7 = (Button)findViewById(R.id.btn_dim7);
        aug = (Button)findViewById(R.id.btn_aug);
        sus2 = (Button)findViewById(R.id.btn_sus2);
        sus4 = (Button)findViewById(R.id.btn_sus4);

        setChordURLWhenTrue();
        resetScore();
        getHighScore();
    }

    /**
     * method untuk tombol playchord
     * menggunakan media plaer untuk memainkan sound yang diakses
dengan link url
     * @param v
     */
    public void playChord(View v){
        try{
            MediaPlayer chordPlayer = new MediaPlayer();
            Uri theUri = Uri.parse(chordURL);
            chordPlayer.setDataSource(getApplicationContext(),
theUri);
            chordPlayer.prepare();
            chordPlayer.start();
        } catch (Exception e){
            Toast.makeText(getBaseContext(),"Can't play chord",
Toast.LENGTH_LONG).show();
        }
    }

    /**
     * method untuk kembali ke home apabila ditekan back
     * namun dengan peringatan "yes" atau "no" karena apabila back
maka progress session akan hilang
     */
    @Override
    public void onBackPressed() {
        AlertDialog.Builder backalert = new
AlertDialog.Builder(this);
        backalert.setTitle("Quit Session");
        backalert.setMessage("Are you sure want to quit and lose
all your progresses?")
                .setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
which) {
                        backHome();
                        finish();
                    }
                })
                .setNegativeButton("No", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
which) {
                        dialog.dismiss();
                    }
                })
                .create();
        backalert.show();
    }

    /**
     * method untuk kembali ke home
     */
    public void backHome(){
        Intent backHome = new Intent(PlayingActivity.this,
HomeActivity.class);
        startActivity(backHome);
        resetScore();
    }

    /**
     * method untuk me-reset score menjadi 0
     * @return
     */
    public int resetScore(){
        return currentscore = 0;
    }

    /**
     * meng-set chord secara random dari database sesuai score saat
ini
     * semakin tinggi score saat ini maka chord yang harus ditebak
semakin sulit
     * @return
     */
    public String setChordURLWhenTrue(){
        if(currentscore <11){
            String urlChord =
"http://teamhamming.esy.es/difficulty1.php";
            PostResponseAsyncTask fetchChordURL = new
PostResponseAsyncTask(this, false, new AsyncResponse() {
                @Override
                public void processFinish(String url) {
                    chordURL = url;
                    getCorrectAnswer();
                }
            });
            fetchChordURL.execute(urlChord);
            return chordURL;
        }
        else if(currentscore >= 11 && currentscore < 21){
            String urlChord =
"http://teamhamming.esy.es/difficulty2.php";
```

```
            PostResponseAsyncTask fetchChordURL = new
PostResponseAsyncTask(this, false, new AsyncResponse() {
                @Override
                public void processFinish(String url) {
                    chordURL = url;
                    getCorrectAnswer();
                }
            });
            fetchChordURL.execute(urlChord);
            return chordURL;
        }
        else if(currentscore >= 21 && currentscore < 31 ){
            String urlChord =
"http://teamhamming.esy.es/difficulty3.php";
            PostResponseAsyncTask fetchChordURL = new
PostResponseAsyncTask(this, false, new AsyncResponse() {
                @Override
                public void processFinish(String url) {
                    chordURL = url;
                    getCorrectAnswer();
                }
            });
            fetchChordURL.execute(urlChord);
            return chordURL;
        }
        else if(currentscore >= 31 && currentscore < 41 ){
            String urlChord =
"http://teamhamming.esy.es/difficulty4.php";
            PostResponseAsyncTask fetchChordURL = new
PostResponseAsyncTask(this, false, new AsyncResponse() {
                @Override
                public void processFinish(String url) {
                    chordURL = url;
                    getCorrectAnswer();
                }
            });
            fetchChordURL.execute(urlChord);
            return chordURL;
        }
        else if(currentscore >= 41 && currentscore < 46 ){
            String urlChord =
"http://teamhamming.esy.es/difficulty5.php";
            PostResponseAsyncTask fetchChordURL = new
PostResponseAsyncTask(this, false, new AsyncResponse() {
                @Override
                public void processFinish(String url) {
                    chordURL = url;
                    getCorrectAnswer();
                }
            });
            fetchChordURL.execute(urlChord);
            return chordURL;
        }
        else{//currentscore >=46
            String urlChord =
"http://teamhamming.esy.es/difficulty6.php";
            PostResponseAsyncTask fetchChordURL = new
PostResponseAsyncTask(this, false, new AsyncResponse() {
                @Override
                public void processFinish(String url) {
                    chordURL = url;
                    getCorrectAnswer();
                }
            });
            fetchChordURL.execute(urlChord);
            return chordURL;
        }
    }

    /**
     * dipanggil saat user menekan tombol major
     * @param view
     */
    public void clickMajor(View view){
        checkAnswer("maj");
    }

    /**
     * dipanggil saat user menekan tombol minor
     * @param view
     */
    public void clickMinor(View view){
        checkAnswer("min");
    }

    /**
     * dipanggil saat user menekan tombol seventh
     * @param view
     */
    public void clickSeventh(View view){
        checkAnswer("seventh");
    }

    /**
     * dipanggil saat user menekan tombol major7
     * @param view
     */
    public void clickMajor7(View view){
        checkAnswer("maj7");
    }

    /**
     * dipanggil saat user menekan tombol minor7
     * @param view
     */
    public void clickMinor7(View view){
        checkAnswer("min7");
    }

    public void clickSus2(View view){
        checkAnswer("sus2");
    }

    /**
     * dipanggil saat user menekan tombol sus4
     * @param view
     */
    public void clickSus4(View view){
        checkAnswer("sus4");
    }

    /**
     * dipanggil saat user menekan tombol diminished
     * @param view
     */
    public void clickDim(View view){
        checkAnswer("dim");
    }

    /**
     * dipanggil saat user menekan tombol diminished7
     * @param view
     */
    public void clickDim7(View view){
        checkAnswer("dim7");
    }

    /**
     * dipanggil saat user menekan tombol augmented
     * @param view
     */
    public void clickAug(View view){
        checkAnswer("aug");
    }

    /**
     * Dipanggil saat user menjawab jawaban yang benar dengan
menampilkan pemberitahuan
     * dan menambah score saat ini +1
     */
    public void trueAnswer(){
        LayoutInflater inflaterTrue = getLayoutInflater();
        View layoutTrue =
inflaterTrue.inflate(R.layout.layout_toasttrue, (ViewGroup)
                findViewById(R.id.toast_true_root));

        Toast toastTrue = new Toast(getApplicationContext());
        toastTrue.setGravity(Gravity.CENTER|Gravity.TOP,0,250);
        toastTrue.setView(layoutTrue);
        toastTrue.setDuration(Toast.LENGTH_SHORT);
        toastTrue.show();

        currentscore = currentscore + 1;
        score.setText(Integer.toString(currentscore));
        setChordURLWhenTrue();
    }

    /**
     * method check answer yg dipanggil setiap tombol-chord ditekan
dengan parameter masing-masing
     * @param cat
     */
    public void checkAnswer(String cat){
        category = cat;
        HashMap postChordURL = new HashMap();
        postChordURL.put("chordURL", chordURL);
        postChordURL.put("category", category);
        String urlMatchAnswer =
"http://teamhamming.esy.es/MatchAnswer.php";
        PostResponseAsyncTask matchAnswer = new
PostResponseAsyncTask(this, postChordURL, false, new
AsyncResponse() {
            @Override
            public void processFinish(String result) {
                switch (result){
                    case "true":
                        trueAnswer();
                        break;

                    case "false":
                        LayoutInflater inflaterFalse =
getLayoutInflater();
                        View layoutFalse =
inflaterFalse.inflate(R.layout.layout_toastfalse, (ViewGroup)

findViewById(R.id.toast_false_root));

                        TextView correctans =
(TextView)layoutFalse.findViewById(R.id.correctanswer);
                        correctans.setText(correctAns);

                        Toast toastFalse = new
Toast(getApplicationContext());
                        toastFalse.setDuration(Toast.LENGTH_SHORT);
                        toastFalse.setView(layoutFalse);
                        toastFalse.setGravity(Gravity.CENTER |
Gravity.TOP, 0, 250);
                        toastFalse.show();

                        Vibrator v = (Vibrator)
getSystemService(Context.VIBRATOR_SERVICE);
                        v.vibrate(110);

                        if(currentscore >
Integer.valueOf(highscore)){
                            updateHighScore();
                            Intent highscore = new
Intent(PlayingActivity.this,HighscoreActivity.class);
                            startActivity(highscore);
                            finish();
                        }
                        else{
                            backHome();
                            finish();
                        }
                        break;

                    default:
                        Toast.makeText(PlayingActivity.this, "Error
occurs", Toast.LENGTH_LONG).show();
                        break;
                }
            }
        });
        matchAnswer.execute(urlMatchAnswer);
```

```
    }

    /**
     * untuk mengambil highscore dari database dari info logininfo
dan username
     */
    public void getHighScore(){
        SharedPreferences sharedPrefLogin =
getSharedPreferences("loginInfo", Context
                .MODE_PRIVATE);
        String sharedUsername =
sharedPrefLogin.getString("username", "");

        HashMap postShared = new HashMap();
        postShared.put("username",sharedUsername);

        String urlhighscore =
"http://teamhamming.esy.es/FetchHighscore.php";
        PostResponseAsyncTask getHighScore = new
PostResponseAsyncTask(this, postShared, new AsyncResponse() {
            @Override
            public void processFinish(String score) {
                highscore = score;
            }
        });
        getHighScore.execute(urlhighscore);
    }

    /**
     * untuk mengupdate highscore username bersangkutan di database
     */
    public void updateHighScore(){
        SharedPreferences sharedPrefLogin =
getSharedPreferences("loginInfo", Context
                .MODE_PRIVATE);
        String username = sharedPrefLogin.getString("username",
"");

        HashMap updateScore = new HashMap();
        updateScore.put("score", String.valueOf(currentscore));
        updateScore.put("username", username);

        String urlupdate =
"http://teamhamming.esy.es/UpdateHighScore.php";
        PostResponseAsyncTask updateHighScore = new
PostResponseAsyncTask(this, updateScore, new AsyncResponse() {
            @Override
            public void processFinish(String s) {

            }
        });
        updateHighScore.execute(urlupdate);
    }

    /**
     * untuk mendapatkan jawaban yg benar dari jawaban user yang
salah
     */
    public void getCorrectAnswer(){
        HashMap getAnswer = new HashMap();
        getAnswer.put("chordURL", chordURL);

        String urlGetAnswer =
"http://teamhamming.esy.es/getAnswer.php";
        PostResponseAsyncTask answer = new
PostResponseAsyncTask(this, getAnswer, false, new AsyncResponse() {
            @Override
            public void processFinish(String answer) {
                correctAns = answer;
            }
        });answer.execute(urlGetAnswer);
    }
}
```

- Source code for Instrument Piano

```
package com.hamming.claritear;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Vibrator;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.kosalgeek.genasync12.AsyncResponse;
import com.kosalgeek.genasync12.PostResponseAsyncTask;

import java.util.HashMap;

public class PlayingPianoActivity extends AppCompatActivity {

    TextView score;
    Button play;
    Button maj;
    Button min;
    Button seventh;
    Button min7;
    Button maj7;
    Button dim;
    Button dim7;
    Button aug;
    Button sus2;
    Button sus4;
    private String correctAns;
    public String chordURL;
    public String category;
    public int currentscore;
    public String highscore;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_playing_piano);
        score = (TextView) findViewById(R.id.tv_piano_printscore);
        play = (Button) findViewById(R.id.btn_piano_playChord);
        maj = (Button)findViewById(R.id.btn_piano_maj);
        min = (Button)findViewById(R.id.btn_piano_min);
        seventh = (Button)findViewById(R.id.btn_piano_seventh);
        min7 = (Button)findViewById(R.id.btn_piano_min7);
        maj7 = (Button)findViewById(R.id.btn_piano_maj7);
        dim = (Button)findViewById(R.id.btn_piano_dim);
        dim7 = (Button)findViewById(R.id.btn_piano_dim7);
        aug = (Button)findViewById(R.id.btn_piano_aug);
        sus2 = (Button)findViewById(R.id.btn_piano_sus2);
        sus4 = (Button)findViewById(R.id.btn_piano_sus4);

        setChordURLWhenTrue();
        resetScore();
        getHighScore();
    }

    public void playChord(View v){
        try{
            MediaPlayer chordPlayer = new MediaPlayer();
            Uri theUri = Uri.parse(chordURL);
            chordPlayer.setDataSource(getApplicationContext(),
theUri);
            chordPlayer.prepare();
            chordPlayer.start();
        } catch (Exception e){
            Toast.makeText(getBaseContext(), "Can't play chord",
Toast.LENGTH_LONG).show();
        }
    }

    @Override
    public void onBackPressed() {
        AlertDialog.Builder backalert = new
AlertDialog.Builder(this);
        backalert.setTitle("Quit Session");
        backalert.setMessage("Are you sure want to quit and lose
all your progresses?")
                .setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
which) {
                        backHome();
                        finish();
                    }
                })
                .setNegativeButton("No", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
which) {
                        dialog.dismiss();
                    }
                })
                .create();
        backalert.show();
    }

    public void backHome(){
        Intent backHome = new Intent(PlayingPianoActivity.this,
HomeActivity.class);
        startActivity(backHome);
        resetScore();
    }

    public int resetScore(){
        return currentscore = 0;
    }

    public String setChordURLWhenTrue(){
        if(currentscore <11){
            String urlChord =
"http://teamhamming.esy.es/pianodifficulty1.php";
            PostResponseAsyncTask fetchChordURL = new
PostResponseAsyncTask(this, false, new AsyncResponse() {
                @Override
                public void processFinish(String url) {
                    chordURL = url;
                    getCorrectAnswer();
                }
            });
            fetchChordURL.execute(urlChord);
            return chordURL;
        }
        else if(currentscore >= 11 && currentscore < 21){
            String urlChord =
"http://teamhamming.esy.es/pianodifficulty2.php";
            PostResponseAsyncTask fetchChordURL = new
PostResponseAsyncTask(this, false, new AsyncResponse() {
                @Override
                public void processFinish(String url) {
                    chordURL = url;
                    getCorrectAnswer();
                }
            });
            fetchChordURL.execute(urlChord);
            return chordURL;
        }
        else if(currentscore >= 21 && currentscore < 31 ){
            String urlChord =
"http://teamhamming.esy.es/pianodifficulty3.php";
            PostResponseAsyncTask fetchChordURL = new
PostResponseAsyncTask(this, false, new AsyncResponse() {
                @Override
                public void processFinish(String url) {
                    chordURL = url;
                    getCorrectAnswer();
                }
            });
            fetchChordURL.execute(urlChord);
            return chordURL;
        }
        else if(currentscore >= 31 && currentscore < 41 ){
            String urlChord =
"http://teamhamming.esy.es/pianodifficulty4.php";
            PostResponseAsyncTask fetchChordURL = new
PostResponseAsyncTask(this, false, new AsyncResponse() {
                @Override
                public void processFinish(String url) {
                    chordURL = url;
                    getCorrectAnswer();
                }
            });
            fetchChordURL.execute(urlChord);
            return chordURL;
        }
        else if(currentscore >= 41 && currentscore < 46 ){
            String urlChord =
"http://teamhamming.esy.es/pianodifficulty5.php";
            PostResponseAsyncTask fetchChordURL = new
PostResponseAsyncTask(this, false, new AsyncResponse() {
                @Override
                public void processFinish(String url) {
                    chordURL = url;
                    getCorrectAnswer();
                }
            });
            fetchChordURL.execute(urlChord);
            return chordURL;
        }
        else{//currentscore >=46
            String urlChord =
"http://teamhamming.esy.es/pianodifficulty6.php";
            PostResponseAsyncTask fetchChordURL = new
PostResponseAsyncTask(this, false, new AsyncResponse() {
                @Override
                public void processFinish(String url) {
                    chordURL = url;
                    getCorrectAnswer();
                }
            });
            fetchChordURL.execute(urlChord);
            return chordURL;
        }
    }

    public void clickMajor(View view){
        checkAnswer("maj");
    }

    public void clickMinor(View view){
        checkAnswer("min");
    }

    public void clickSeventh(View view){
        checkAnswer("seventh");
    }
    public void clickMajor7(View view){
        checkAnswer("maj7");
    }

    public void clickMinor7(View view){
        checkAnswer("min7");
    }

    public void clickSus2(View view){
        checkAnswer("sus2");
    }

    public void clickSus4(View view){
        checkAnswer("sus4");
    }

    public void clickDim(View view){
        checkAnswer("dim");
    }

    public void clickDim7(View view){
        checkAnswer("dim7");
    }

    public void clickAug(View view){
        checkAnswer("aug");
    }

    /**
     * Dipanggil saat user menjawab jawaban yang benar
     */
    public void trueAnswer(){
        LayoutInflater inflaterTrue = getLayoutInflater();
        View layoutTrue =
inflaterTrue.inflate(R.layout.layout_toasttrue, (ViewGroup)
                findViewById(R.id.toast_true_root));

        Toast toastTrue = new Toast(getApplicationContext());
        toastTrue.setGravity(Gravity.CENTER|Gravity.TOP,0,250);
        toastTrue.setView(layoutTrue);
        toastTrue.setDuration(Toast.LENGTH_SHORT);
        toastTrue.show();

        currentscore = currentscore + 1;
        score.setText(Integer.toString(currentscore));
        setChordURLWhenTrue();
    }

    public void checkAnswer(String cat){
        category = cat;
        HashMap postChordURL = new HashMap();
        postChordURL.put("chordURL", chordURL);
        postChordURL.put("category", category);

        String urlMatchAnswer =
"http://teamhamming.esy.es/PianoMatchAnswer.php";
```

```
        PostResponseAsyncTask matchAnswer = new
PostResponseAsyncTask(this, postChordURL, false, new
AsyncResponse() {
            @Override
            public void processFinish(String result) {
                switch (result){
                    case "true":
                        trueAnswer();
                        break;

                    case "false":
                        LayoutInflater inflaterFalse =
getLayoutInflater();
                        View layoutFalse =
inflaterFalse.inflate(R.layout.layout_toastfalse, (ViewGroup)

findViewById(R.id.toast_false_root));

                        TextView correctans =
(TextView)layoutFalse.findViewById(R.id.correctanswer);
                        correctans.setText(correctAns);

                        Toast toastFalse = new
Toast(getApplicationContext());
                        toastFalse.setDuration(Toast.LENGTH_SHORT);
                        toastFalse.setView(layoutFalse);
                        toastFalse.setGravity(Gravity.CENTER |
Gravity.TOP, 0, 250);
                        toastFalse.show();

                        Vibrator v = (Vibrator)
getSystemService(Context.VIBRATOR_SERVICE);
                        v.vibrate(110);

                        if(currentscore >
Integer.valueOf(highscore)){
                            updateHighScore();
                            Intent highscore = new
Intent(PlayingPianoActivity.this,HighscoreActivity.class);
                            startActivity(highscore);
                        }
                        else{
                            backHome();
                        }
                        finish();
                        break;

                    default:
                        Toast.makeText(PlayingPianoActivity.this,
"Error occurs", Toast.LENGTH_LONG).show();
                        break;
                }
            }
        });
        matchAnswer.execute(urlMatchAnswer);
    }

    public void getHighScore(){
        SharedPreferences sharedPrefLogin =
getSharedPreferences("loginInfo", Context
                .MODE_PRIVATE);
        String sharedUsername =
sharedPrefLogin.getString("username", "");

        HashMap postShared = new HashMap();
        postShared.put("username",sharedUsername);

        String urlhighscore =
"http://teamhamming.esy.es/FetchHighscore.php";
        PostResponseAsyncTask getHighScore = new
PostResponseAsyncTask(this, postShared, new AsyncResponse() {
            @Override
            public void processFinish(String score) {
                highscore = score;
            }
        });
        getHighScore.execute(urlhighscore);
    }

    public void updateHighScore(){
        SharedPreferences sharedPrefLogin =
getSharedPreferences("loginInfo", Context
                .MODE_PRIVATE);
        String username = sharedPrefLogin.getString("username",
"");

        HashMap updateScore = new HashMap();
        updateScore.put("score", String.valueOf(currentscore));
        updateScore.put("username", username);

        String urlupdate =
"http://teamhamming.esy.es/UpdateHighScore.php";
        PostResponseAsyncTask updateHighScore = new
PostResponseAsyncTask(this, updateScore, new AsyncResponse() {
            @Override
            public void processFinish(String s) {

            }
        });
        updateHighScore.execute(urlupdate);
    }

    public void getCorrectAnswer(){
        HashMap getAnswer = new HashMap();
        getAnswer.put("chordURL", chordURL);

        String urlGetAnswer =
"http://teamhamming.esy.es/PianoGetAnswer.php";
        PostResponseAsyncTask answer = new
PostResponseAsyncTask(this, getAnswer, false, new AsyncResponse() {
            @Override
            public void processFinish(String answer) {
                correctAns = answer;
            }
        });answer.execute(urlGetAnswer);
    }
}
```

- Source code for Registration

```
package com.hamming.claritear;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.kosalgeek.genasync12.AsyncResponse;
import com.kosalgeek.genasync12.PostResponseAsyncTask;

import java.util.HashMap;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

/**
 * Registration class for registration activity
 */
public class Registration extends AppCompatActivity {
    EditText username, email, password, rptpassword;
    Button register;

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        Intent toLogin = new Intent(Registration.this,Login.class);
        startActivity(toLogin);
        finish();
    }

    /**
     * Method onCreate untuk mendeklarasi fungsi button dan
editText di activity_registration.xml
     * @param savedInstanceState
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);

        username = (EditText)findViewById(R.id.et_regist_username);
        email = (EditText)findViewById(R.id.et_regist_email);
        password = (EditText)findViewById(R.id.et_regist_password);
        rptpassword =
(EditText)findViewById(R.id.et_regist_repeatPassword);
        register = (Button)findViewById(R.id.btn_regist_register);
    }

    /**
     * Method register untuk menentukan apa yang akan dilakukan
bila button register ditekan
     */
    public void register(View v){
        HashMap PostData = new HashMap();
        PostData.put("username",username.getText().toString());
        PostData.put("email",email.getText().toString());
        PostData.put("password",password.getText().toString());

        if(username.getText().toString().equals("") ||
email.getText().toString().equals("") ||
password.getText().toString().equals("") ||
rptpassword.getText().toString().equals("")){
            Toast.makeText(this,"One ore more field(s) are
empty",Toast.LENGTH_LONG).show();
        }
        else if(!validateUser(username.getText().toString())){
            username.setError("Username must contain at least 4
characters");
            username.requestFocus();
        }
        else if(!validateEmail(email.getText().toString())){
            email.setError("Invalid Email Address");
            email.requestFocus();
        }
        else if(!validatePassword(password.getText().toString())){
            password.setError("Password must contain at least 6
characters");
            password.requestFocus();
        }
        else
if(!password.getText().toString().equals(rptpassword.getText().toSt
ring())) {
            rptpassword.setError("Passwords don't match");
            rptpassword.requestFocus();
        }
        else{
            String urlRegister =
"http://teamhamming.esy.es/Register.php";
            PostResponseAsyncTask registerTask = new
PostResponseAsyncTask(this, PostData, new AsyncResponse() {
                @Override
                public void processFinish(String result) {

Toast.makeText(Registration.this,result,Toast.LENGTH_LONG).show();
                    if(result.equals("You are now registered")) {
                        Intent i = new Intent(Registration.this,
Login.class);
                        startActivity(i);
                        finish();
                    }
                    else {

Toast.makeText(Registration.this,"Connection error. Please try
again!",Toast.LENGTH_LONG)
                                .show();
                    }
                }
            });
            registerTask.execute(urlRegister);
        }
    }

    /**
```

```
     * Method validateUser untuk memvalidasi input username
     * @param stringusr
     * @return
     */
    private boolean validateUser(String stringusr) {
        //username harus lebih dari 4 karakter
        if(stringusr.length()<4){
            return false;
        }
        else{
            return true;
        }
    }

    /**
     * Method validatePassword untuk memvalidasi input input
password
     * @param stringpass
     * @return
     */
    private boolean validatePassword(String stringpass) {
        //password harus lebih dari 6 karakter
        if(stringpass.length()<6){
            return false;
        }
        else{
            return true;
        }
    }

    /**
     * Method validateEmail untuk memvalidasi input email
     * @param stringemail
     * @return
     */
    private boolean validateEmail(String stringemail) {
        //Menggunakan Regular Expression
        String emailPattern = "[a-zA-Z0-9\\+\\.\\_\\%\\-
\\+]{1,256}" +
                "\\@" +
                "[a-zA-Z0-9][a-zA-Z0-9\\-]{0,64}" +
                "(" +
                "\\." +
                "[a-zA-Z0-9][a-zA-Z0-9\\-]{0,25}" +
                ")+";
        Pattern pattern = Pattern.compile(emailPattern);
        Matcher matcher = pattern.matcher(stringemail);

        return matcher.matches();
    }
}
```

- Source code for splash screen

```
package com.hamming.claritear;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

/**
 * Class untuk splashscreen
 */
public class SplashActivity extends AppCompatActivity {

    /**
     * method override untuk mendeclare component pada xml
     * @param savedInstanceState
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);

        //menggunakan thread untuk mengatur sleep dan lainnya
        Thread timer = new Thread() {
            public void run() {
                try {
                    sleep(4500);

                    //Pengecekean untuk auto login
                    SharedPreferences sharedPrefLogin =
getSharedPreferences("loginInfo", Context
                            .MODE_PRIVATE);
                    //Jika memori masih menyimpan string data login
maka dari splashscreen akan langsung ke home
                    if (sharedPrefLogin.getString("username",
"").length() == 0) {
                        startActivity(new
Intent(SplashActivity.this, Login.class));
                    }
                    //Jika tidak maka dari splashscreen akan ke
login page
                    else {
                        startActivity(new
Intent(SplashActivity.this, HomeActivity.class));
                    }
                    finish();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        };
        timer.start();
    }
}
```

- Source code for Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.hamming.claritear" >

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/claritear"
        android:label="@string/app_name"
        android:supportsRtl="true" >
        <activity
            android:name=".Registration"
            android:configChanges="orientation"
            android:label="Registration"
            android:screenOrientation="portrait"
            android:theme="@style/Theme.AppCompat.NoActionBar" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>

                <category
android:name="android.intent.category.default" />
            </intent-filter>
        </activity>
        <activity
            android:name=".Login"
            android:configChanges="orientation"
            android:label="Login"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>

                <category
android:name="android.intent.category.default" />
            </intent-filter>
        </activity>
        <activity
            android:name=".HomeActivity"
            android:configChanges="orientation"
            android:screenOrientation="portrait"
            android:theme="@style/Theme.AppCompat" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>

                <category
android:name="android.intent.category.default" />
            </intent-filter>
        </activity>
        <activity
            android:name=".PlayingActivity"
            android:configChanges="orientation"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>

                <category
android:name="android.intent.category.default" />
            </intent-filter>
        </activity>
        <activity
            android:name=".SplashActivity"
            android:configChanges="orientation"
            android:screenOrientation="portrait"

android:theme="@style/Theme.AppCompat.Light.NoActionBar" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".HighscoreActivity"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>

                <category
android:name="android.intent.category.default" />
            </intent-filter>
        </activity>
        <activity
            android:name=".LeaderActivity"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>

                <category
android:name="android.intent.category.default" />
            </intent-filter>
        </activity>
        <activity
            android:name=".ChooseInstrument"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>

                <category
android:name="android.intent.category.default" />
            </intent-filter>
        </activity>
        <activity
            android:name=".PlayingPianoActivity"
            android:configChanges="orientation"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>
```

```
                <category
android:name="android.intent.category.default" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

CHAPTER 4
TEST PLAN

Introduction

This document is a high-level overview defining our testing strategy for the ClaritEar Mobile Application. Its objective is to communicate project-wide quality standards and procedures. It portrays a snapshot of the project as of the end of the planning phase. This document will address the different standards that will apply to the unit, integration and system testing of the specified application. We will utilize testing criteria under the white box, black box, and system-testing paradigm. This paradigm will include, but is not limited to, the testing criteria, methods, and test cases of the overall design. Throughout the testing process we will be applying the test documentation specifications described in the IEEE Standard 829-1983 for Software Test Documentation.

Team Interaction

The following describes the level of team interaction necessary to have a successful product.

- The Test Team is responsible for visualizing test cases and raising quality issues and concerns during meetings to address issues early enough in the development cycle.

- If an area is not acceptable for testing, the code complete date will be pushed out, giving the developers additional time to stabilize the area.

- Since the application interacts with a back-end system component, the Test Team will need to include a plan for integration testing. Integration testing must be executed successfully prior to system testing.

Test Objective

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. We will be testing a ClaritEar Application. There will be eight key functions used to manage our application: load, store, clear, search, insert, delete, list in ascending order, and list in descending order. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

Process Overview

The following represents the overall flow of the testing process:

1. Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.

2. Identify which particular test(s) will be used to test each module.

3. Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.

4. Identify the expected results for each test.

5. Document the test case configuration, test data, and expected results.

6. Perform the test(s).

7. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).

8. Successful unit testing is required before the unit is eligible for component integration/system testing.

9. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.

10. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

Testing Process

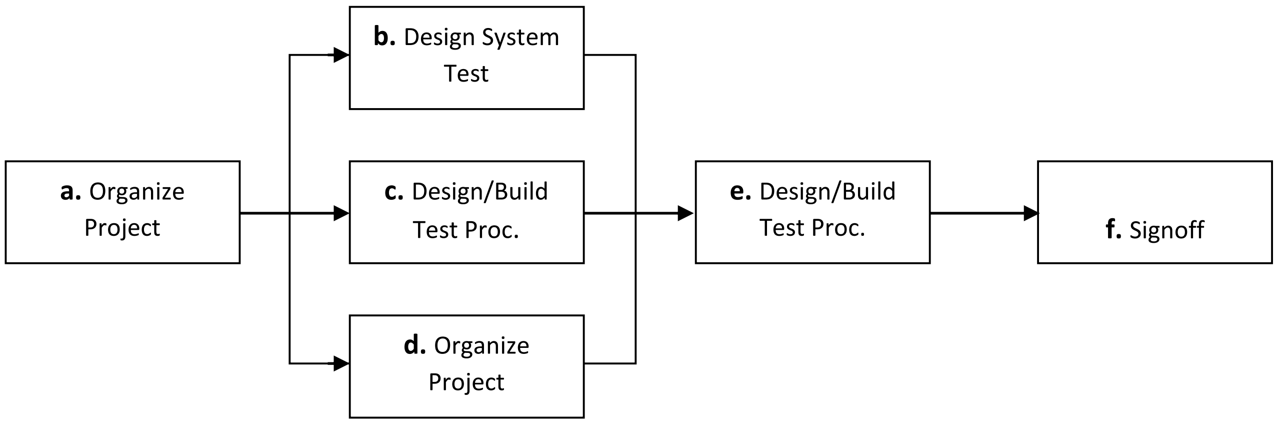


Figure 15. Test Process Flow

The diagram above outlines the Test Process approach that will be followed.

a. **Organize Project** involves creating a System Test Plan, Schedule & Test Approach, and assigning responsibilities.

b. **Design/Build System Test** involves identifying Test Cycles, Test Cases, Entrance & Exit Criteria, Expected Results, etc. In general, test conditions/expected results will be identified by the Test Team in conjunction with the Development Team. The Test Team will then identify Test Cases and the Data required. The Test conditions are derived from the Program Specifications Document.

c. **Design/Build Test Procedures** includes setting up procedures such as Error Management systems and Status reporting.

d. **Build Test Environment** includes requesting/building hardware, software and data set-ups.

e. **Execute System Tests** – The tests identified in the Design/Build Test Procedures will be executed. All results will be documented and Bug Report Forms filled out and given to the Development Team as necessary.

f. **Signoff** - Signoff happens when all pre-defined exit criteria have been achieved.

Testing Strategy

Tested By:		
Test Type		
Test Case Number		
Test Case Name		
Test Case Description		
Item(s) to be tested		
1		
2		
Specifications		
Input	**Expected Output/Result**	
Procedural Steps		
1		
2		
3		

4	
5	
6	
7	

The following outlines the types of testing that will be done for unit, integration, and system testing. While it includes what will be tested, the specific use cases that determine how the testing is done will be detailed in the Test Design Document. The template that will be used for designing use cases is shown in Figure 2.

Figure 2: Test Case Template

Unit Testing

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

White Box Testing

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

Black Box Testing

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

Equivalence Partitioning

In considering the inputs for our equivalence testing, the following types will be used:

- Legal input values – Test values within boundaries of the specification equivalence classes. This shall be input data the program expects and is programmed to transform into usable values.
- Illegal input values – Test equivalence classes outside the boundaries of the specification. This shall be input data the program may be presented, but that will not produce any meaningful output.

The equivalence partitioning technique is a test case selection technique in which the test designer examines the input space defined for the unit under test and seeks to find sets of input that are, or should be, processed identically. The following table represents our equivalence classes, both valid and invalid.

Input/Output Event	**Valid Equivalence Classes**	**Invalid Equivalence Classes**
Input maximum number of allowed values	25 values	> 25 values
Input biginteger	bigintegers between -9223372036854775808 and 9223372036854775808	bigInteger > 9223372036854775808 bigInteger < -9223372036854775808 Non-integers (characters) Non-integers (decimal values)
Load external file	Comma delimited file with only one value per line	No commas Multiple entries per line

	File exists	No file content File does not exist
Store external file	File exists	File does not exist

Boundary Value Testing

The acceptable range of values for this application was set by the development team. Due to the limitations of the GUI, the developers also limited the size of the input values to three digit integers. The valid and invalid ranges are shown below along with the corresponding valid and invalid boundary test values.

Acceptable Range: $-9223372036854775808 \leq x \leq 9223372036854775808$

Invalid Range: $-\infty < x < -9223372036854775808$ and $9223372036854775808 < x < +\infty$

Valid Boundary Tests:

Boundary_1: $x = -9223372036854775808$

Boundary_2: $x = 0$

Boundary_3: $x = 9223372036854775808$

Invalid Boundary Tests:

Boundary_4: $x = 1000000000000000000000$

Boundary_5: $x = -100000000000000000000$

Boundary_6: $x = -9999999999999999999$

Boundary_7: $x = 99999999999999999999$

Integration Testing

Incremental Testing

There are two primary modules that will need to be integrated: the Graphic User Interface module and the Shop Repository module (back-end). The two components, once integrated, will form the complete ClaritEar Mobile Application. The following describes these modules as well as the steps that will need to be taken to achieve complete integration. We will be employing an incremental testing strategy to complete the integration.

Module 1 - Graphic User Interface (GUI) Module

This module provides a simple GUI where the user can perform the different actions (functions). This module will be tested separate from the backend to check if each interface (e.g. insert button) is functioning properly, and in general, to test if the mouse-event actions are working properly. The testing will be performed by writing a stub for each element in the interface.

Module 2 – ClaritEar Mobile Backend Module

This module will be tested separate from the GUI by printing out the results to the Console. In testing this module we will follow the incremental testing method i.e. testing one function first and then keep adding additional function and test it again until all the required functions are tested.

When the GUI is combined with the backend module, we will have a complete ClaritEar Mobile application. To achieve complete integration of these two modules, we will test each element in the GUI by replacing the stubs with the appropriate function from the back end. The results will be displayed within the GUI instead of through the Console. In testing the combined modules, we will follow the incremental testing method. Each stub will be replaced one at a time and tested. This will be done until all stubs have been replaced by the appropriate functions from the backend.

System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case well focus only on function validation and performance. And in both cases we will use the black-box method of testing.

Function Validation Testing

The integrated "ClaritEar Mobile Application" will be tested based on the requirements to ensure that we built the right application. In doing this test, we will try to find the errors in the inputs and outputs

In addition, we will test:

- The interfaces to ensure they are functioning as desired (i.e. check if each interface is behaving as expected, specifically verifying the appropriate action is associated with each mouse_click event).
- The interaction between the GUI and the backend repository. In this case the data will be inserted and check if they are processed in the backend and give the expected output.

Performance testing

This test will be conducted to evaluate the fulfillment of a system with specified performance requirements. It will be done using black-box testing method. And this will be performed by:

- Storing the maximum data in the file and trying to insert, and observe how the application will perform when it is out of boundary.
- Deleting data and check if it follows the right sorting algorithm to sort the resulting data or output.
- Trying to store new data and check if it over writes the existing once.
- Trying to load the data while they are already loaded

Entry and Exit Criteria

This section describes the general criteria by which testing commences, temporarily stopped, resumed and completed within each testing phase. Different features/components may have slight variation of their criteria, in which case, those should be mentioned in the feature test plan. The testing phase also maps to the impact level definition when a defect is entered in the bug-tracking phase.

Unit Testing

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

Black Box Phase

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We will use Equivalence Partitioning and Boundary Value Analysis complexity metrics in order to quantifiably determine how many test cases needed to achieve maximum code coverage.

Black Box Entry Criteria

The Black Box Entry Criteria will rely on the component specification, and user interface requirements. Things that must be done on entry to the Black Box stage:

- All ClaritEar Mobile functions must either be coded or stubs created.
- The type of Black Box testing Methods will be determined upon entry. We will use Equivalency Partition, and Boundary Value Analysis.
- Equivalency Partition will include, Integer data types only, No Character data types accepted, each data field will be comma delimited, and there will be 1 value per line in the data file.

Black Box Exit Criteria

The Black Box Exit Criteria listed below explains what needs to be completed in-order to exit Black Box phase. To exit the Black Box phase 100% success rate must be achieved. Things that must be done upon exiting the Black Box stage:

- The Equivalence Classes will have been created for the valid and invalid input values. For our ClaritEar Mobile the input domain values for Equivalence Partitions will include Integer data types only, each data field will be delimited by a comma and carriage return, and one data value per line in the input data file.

- The invalid input domain values for Equivalence classes will include loading an empty input data file, entering a delimiter other that a comma, and entering more than one data value per line in the input data file.
- Boundary Value Analysis will have generated Test Cases based on the boundary values of Integer data type values. These Test Cases will test for values above and below the specified boundary values. For example, values that include infinity, negative infinity, zero, and decimal numbers.
- All code bugs that are exposed are corrected.

White Box Phase

The White Box criteria apply for purposes of focusing on internal program structure, and discover all internal program errors. Defects will be categorized and the quality of the product will be assessed.

White Box Entry Criteria

The White Box Entry Criteria will rely on the QA engineers verifying that the major features work alone but not necessarily in combination; exception handling will not be implemented. The design and human interface are stable. Things that must be done on entry to the White Box stage:

- All ClaritEar Mobile functions must be coded.
- Black Box Testing should be in its late stages.

After the White Box criteria have been met, the product enters the White Box stage. During White Box stage Development Engineering’s emphasis is on refining the product and fixing defects. Information Design’s emphasis is on developing product user documentation.

White Box Exit Criteria

ClaritEar Mobile Apps in the White Box stage should have a generally stable feel to it. White Box testing continues until the Black Box or next milestone criteria are met. To exit the White Box phase 100% success rate must be achieved. The following describes the state of the product upon exit from the White Box Stage:

- All ClaritEar Mobile functions are implemented, operational and tested.

- All Branch Testing test cases will be generated. The test cases will be generated from the Control Flow diagrams of all functions.
- All code bugs that are exposed are corrected.

Integration Test

There are two modules that will be integrated for Integration Testing. The two modules are The Graphic User Interface module and database(back-end). The two components will consist of a mixture of stubs, driver, and full function code. The following describes the entry and exit criteria for Integration testing.

Integration Test Entry Criteria

The Integration Test Entry Criteria will rely on both modules to be operational. Things that must be done on entry to the Integration Test stage:

- All ClaritEar Mobile function must either be coded and/or stubs created.
- The Graphical User Interface must either be coded and/or a driver and stubs must be created. The driver is implemented to facilitate test case input and output values.
- Interfaces and interactions between the Shop Module and the Graphical User Interface must be operational.
- Black Box Testing should either be in its late stages or completed.
- White Box Testing should have begun.

Integration Test Exit Criteria

The Integration Test Exit Criteria will rely on both modules to be operational. The ClaritEar Mobile design and human interface must be stable. To exit the Integration Testing phase 100% success rate must be achieved. Things that must be done on exit from the Integration Test stage:

- All code bugs that are exposed are corrected.
- The Shop Module and Graphical User Interface Module will interact together with complete accuracy, according to the System Specification Design. All discrepancies are corrected.
- Both Modules are ready for System Testing. Stubs and drivers are replaced with fully functional code.
- Black Box Testing is completed.
- White Box Testing should either be in its late stages or completed.

System Test

The System Test criteria apply for purposes of categorizing defects and the assessing the quality level of the product. All elements of the Shop Module and Graphical User Interface are meshed together and tested as a whole. System test focuses on functions and performance, reliability, instillation, behavior during special conditions, and stress testing.

System Test Entry Criteria

The Entrance Criteria specified by the Development Engineers, should be fulfilled before System Test can commence. In the event, that any criterion has not been achieved, the System Test may commence if both Development and Test Engineers are in full agreement that the risk is manageable.

- The Graphical User Interface and the ClaritEar Mobile back-end Module must be fully functional.
- All developed code must be unit tested. Unit and Link Testing must be completed and signed off by the development team.
- All test hardware and environments must be in place, and free for System test use.
- All Black Box testing must be complete and exposed bugs must be corrected.
- All White Box testing must be complete and exposed bugs must be corrected.
- Integration Testing must be complete and exposed bugs must be corrected
- The Graphical User Interface will be the method of interacting with the system, so the GUI will be tested thoroughly.
- Development and Test Engineers agree that Function Validation Testing will cover function performance, reliability, stress and load testing.

System Exit Criteria

The Exit Criteria must satisfy all the criteria listed below. This verifies that all elements of the project mesh properly. This is to make sure that all the system functions and performs according to the System Specification Document.

- All Function Validation Testing is 100 percent successful. Testing for all ClaritEar Mobile functions interact with complete accuracy.
- No degradation of System performance across different platforms of Windows operating system will be affected. (Windows 7 or above is acceptable)
- The Graphical User Interface performs to System Specification Requirements.
- All the Shop properties are expressed correctly through the Graphical User Interface.
- All input fields on the Graphical User Interface are working correctly.

- All high priority errors from System Testing must be fixed and tested.
- If any medium or low-priority errors are outstanding – the Development Engineers and Test manager must sign off the implementation risk as acceptable.

Bug Tracking/ Bug Process

During testing, the testing team members normally encounter behavior that goes against a specified or implied design requirement in the product. When this happens, we will document and reproduce the bugs for the developers.

Expectation of a bug:

- Keep track of what version of the application the bug is found
- Determine if bug has already been written up
- Indicate the steps to reproduce the bug – write enough details for others looking at the bug to be able to duplicate it; exclude unnecessary steps (i.e. If access point is irrelevant, be more general in your steps).
- Actual results – be specific on your findings.
- Expected results – how the product should behave based on the specified or implied requirements.
- Implications – How does the defect affect the quality of the product?

The following chart defines the impact levels to be used when entering bugs.

Impact	Definitions
1 – Fatal	**Test Stopper:** If you can't access a function and need the bug to be fixed immediately. The defect prevents QA from testing the feature area, sub-area or functionality of the feature.
2 – Serious	**Beta Stopper:** This is a bug that users would experience such as: data corruption, calculation errors, incorrect data, UE's and system crash on common user scenarios, significant QA risk, and major UI defects.
3 – Minor	**Live Release:** A bug that must be fixed before the product is officially completed, UE's or crashes, content, and UI and graphic changes required for release.

Various Roles in Bug Resolution

- **Author –** The person who wrote the bug; this will be someone on the QA team
- **Resolver –** Normally an Engineer assigned to a specific area of the application.
- **Verifier –** normally a QA Engineer responsible for testing the fix and closing the bug.

Bug Report Form

VHTN Software Development Problem Report #: ________

Program ______________________ Release ______________ Version ________

Report Type (1-6)_____	Severity(1-3)_____	Attachments (Y/N) ________
1 - Coding error	*1 - Fatal*	*If yes, describe:*
2 - Design issue	*2 - Serious*	______________________
3 - Suggestion	*3 - Minor*	______________________
4 - Documentation		
5 - Hardware		
6 - Query		

Problem Summary ____

Can you reproduce the problem? (Y/N) ________

Problem & how can it be reproduced?

____

____

____

Suggested fix (optional input)

____

____

____

Reported By: ______________________________ Date: ______________

Items below are for use only by the development team

Functional Area:____________________ Assigned To:____________________

Comments:

____

____

Status_____ Prioirity(1-5)_____

1 - open *2 - closed*

Resolution(1-9)_____ Resolution Version ________

1 - Pending	*4 - Deferred*	*7 - Withdrawn by reporter*
2 - Fixed	*5 - As designed*	*8 - Need more info*
3 - Irreproducible	*6 - Can't be fixed*	*9 - Disagree with suggestion*

Resolved By: ______________________________ Date: ______________

Tested By: ______________________________ Date: ______________

Treat as Deferred (Y/N)_____

Roles and Responsibilities

Development Team

Code Development Project Leader – Pandu Wicaksono

- Ensure Phase 1 is delivered to schedule and quality
- Ensure exit criteria are achieved prior to system test signoff
- Regularly review testing progress with test controller.
- Raise and manage issues/risks relating to project or outside test teams control.
- Review and sign off test approach, plans and schedule.

Testing Team

Test Planner / Controller – Muhammad Harist Refian

- Ensure Phase 1 is delivered to schedule and quality
- Produce high level and detailed test conditions
- Produce expected results
- Report progress at regular status reporting meetings
- Co-ordinate review and signoff of test conditions
- Manage individual test cycles and resolve tester queries/problems.

Lead Tester – Andira Rozawati

- Identify test data
- Execute test conditions and mark-off results
- Prepare software error reports
- Administrate error measurement system
- Ensure test systems outages/problems are reported immediately and followed up.
- Ensure entrance criteria are achieved prior to system test start.
- Ensure exit criteria are achieved prior to system test signoff.

Test Schedule

The section contains the overall project schedule. It discusses the phases and key milestones as they relate to quality assurance. It discusses the testing goals and standards that we'd like to achieve for each phase of testing that will be deployed, e.g., Usability Testing, Code Complete Acceptance, Beta Testing, Integration Testing, Regression Testing, System Testing.

The key dates for overall ClaritEar Mobile development and Testing are outlined below. For details on the schedule, refer to SPMP. For details on general Engineering QA deliverables, refer to the test plan document.

Milestones	**End Date**	**Notes**	**QA Deliverables/Roles**
Planning Phase	03/03/16	At this Milestone, the high level planning should be completed. Some of the deliverables are: Project Plan, Program function specifications.	High-level test planning activities, which include preliminary development of Master QA Plan (this document, QA schedule.
Design Phase	29/03/16	This is a feature-driven milestone where the requirements and initiatives are further defined and solutions are finalized. The deliverables for this phase are Program source code and other design related documents.	Development and Test engineers participate actively in feature design by inspecting and reviewing the requirements and design documents. As the design documents are completed, the test engineers are encouraged to start working on the Test Plan document and test design planning.
Code Complete -Infrastructure	30/04/16	This milestone is when all infrastructure development and functions should be complete. The testing team should have preformed unit & integration testing before checking the code into any build.	The Test Engineers should have completed or in the final stages of their preliminary Infrastructure Test Plan, test cases and other QA documents related to test execution for each feature or component such as test scenarios, expected results, data sets, test procedures, scripts and applicable testing tools.
Code Complete -Function	22/05/16	This milestone includes unit testing and code review of each function component prior to checking the code into the test phase. The deliverables include system-testing specification, Unit testing specifications, Integration plan.	The Test Engineers should have provided Code Complete Assessment Test to Development Engineer one week prior to Code Complete Review date. The Test Engineers should also have completed or in the final stages of their preliminary White Box Test Plan, test cases and other QA documents related to test execution for each feature or component such as test scenarios, expected results, data sets, test procedures, scripts and applicable testing tools.

Milestones	**End Date**	**Notes**	**QA Deliverables/Roles**
Beta Ready	31/05/16	This milestone represents that all features are ready for Beta release shutdown.	2 Weeks regression of ClaritEar Mobile features to Beta and preparation for Beta Shutdown.
Ship/Live	07/06/16	Product is out.	Any unfinished Testing documents should be complete.

Quality survey

The form

	Questions	Completely disagree	Disagree	Neutral	Agree	Completely agree
1	You can easily choose things in ClaritEar Mobile					
2	The app is recognizing your exact walk and direction					
3	The user interface is user-friendly					
4	The app does not crash too frequently					
5	You can check out and buy things in ClaritEar Mobile					

The tester will answer the following questions by choosing between the five (5) possibilities. These questions are about the tester's experience about Tomorrow-Shop, and should be answered honestly.

Deliverables

- Program function specifications
- Program source code
- Test plan document - this document should address testing objectives, criteria, standards, schedule and assignments, and testing tools.
  - Unit Testing Plan
  - Integration Plan
  - System Testing Plan
- Test Design Document
  - Unit white-box test design – covers white testing criteria, methods and test cases
  - Unit black-box test design – covers black-box testing criteria, methods and test cases
  - System test design – covers system test criteria, methods, and test cases, scripts.
- Test report document
  - Unit white-box test report – covers unit white box test results, problems, summary and analysis
  - Unit black-box test report – covers unit black box test results, problems, summary and analysis
  - System Test report – covers system test results, problems, summary and analysis

CHAPTER 5
QUALITY ASSURANCE

A. Purpose

The purpose of this Software Quality Assurance (SQA) Plan is to establish the goals, processes, and responsibilities required to implement effective quality assurance functions for the CLARITEAR Mobile Application project.

The CLARITEAR Mobile Application Software Quality Assurance Plan provides the framework necessary to ensure a consistent approach to software quality assurance throughout the project life cycle. It defines the approach that will be used by the SAM and Software Quality (SQ) personnel to monitor and assess software development processes and products to provide objective insight into the maturity and quality of the software.

B. Scope

This plan covers SQA activities throughout the phases of the CLARITEAR Mobile Application mission.

This project is to create an Android-based application called ClarietEar. ClaritEar (/ˈklær.ə.tıər/) is an educational and entertainment applications based on Android that are useful to train your hearing sensitivity in knowing the kind of tone or chord that is being played. With this application we hope people will gain more knowledge about music chords. Also we hope our application will be helping a lot of people on becoming new musician.

C. Reference Documents

The following documents were used or referenced in the development of this plan:

- IEEE STD 730-2002, IEEE Standard for Software Quality Assurance Plans
- CLARITEAR Mobile Application Preliminary Project Plan
- CLARITEAR Mobile Application Software Management Plan (or Product Plan)
- CLARITEAR Mobile Application Software Test Document

D. Management

This section describes the management organizational structure, its roles and responsibilities, and the software quality tasks to be performed.

Management Organization

CLARITEAR Mobile Application efforts are supported by personnel Relevant entities/roles that are of interest and applicable to this SQA Plan and the software assurance effort are described at a high level below.

Product Assessments

The following are typical product assessments that may be conducted by SQ personnel. See the SQ Activity Schedule for the planned assessments:

- Peer Review packages
- Document Reviews
- Software Management Plan
- Software Test Document
- Test results

Process Assessments

The following are typical process assessments that may be conducted by SQ personnel. See the SQ Activity Schedule for the planned assessments:

- Project Planning
- Project Monitoring and Control
- Measurement and Analysis
- System/Subsystem Reviews
- Peer Reviews
- Requirements Management
- Software Configuration Management and Configuration Audits (FCA/PCA)
- Test Management (Verification & Validation)

- Software Problem Reporting and Corrective Action
- Risk Management
- Supplier Agreement Management

Roles and Responsibility

This section describes the roles and responsibilities for each assurance person assigned to the CLARITEAR Mobile Application Project.

SAM

Responsibilities include, but are not limited to:

- Secure and manage SQ personnel resource levels
- Ensure that SQ personnel have office space and the appropriate tools to conduct SQ activities
- Provide general guidance and direction to the SQ personnel responsible for conducting software quality activities and assessments
- Escalate any noncompliances to project management

Software Quality Personnel

Responsibilities include, but are not limited to:

- Develop and maintain the project software quality assurance plan
- Generate and maintain a schedule of software quality assurance activities
- Conduct process and product assessments, as described within this plan, using objective criteria
- Interface with Safety, Reliability, and IV&V personnel on software assurance activities
- Identify and document noncompliances, observations, and risks from all software assurance related activities to the SAM
- Communicate results from assessments with relevant stakeholders
- Ensure resolution of noncompliances and escalate any issues that cannot be resolved within the project

- Identify lessons learned that could improve processes for future products
- Develop and maintain metrics

Safety Personnel

Responsibilities include, but are not limited to:

- Provide system software safety expertise to the SQ personnel and/or project personnel, as required
- Assist in the assessment of the various software development efforts in terms of meeting applicable software safety standards and requirements
- Assist in the resolution of any software safety related issues, concerns, and/or risks identified throughout the project life cycle
- Assist in the review of various life cycle related artifacts as they pertain to system software safety

For additional support information, reference the project's System Safety Plan.

[Note: make sure this information is covered in the System Safety Plan.]

Reliability Personnel

Responsibilities include, but are not limited to:

- Provide software reliability expertise to the SQ personnel and/or project personnel, as required. Assist in the assessment of the various software development efforts in terms of meeting applicable software reliability standards and requirements
- Assist in the resolution of any software reliability related issues, concerns, and/or risks identified throughout the life cycle
- Assist in the review of various life cycle related artifacts as they pertain to software reliability

Software Assurance Estimated Resources

Staffing to support software assurance activities must be balanced against various project characteristics and constraints, including cost, schedule, maturity level of the providers, criticality of the software being developed, return on investment, perceived risk, etc.

Documentation

Purpose

This section identifies the minimum documentation governing the requirements, development, verification, validation, and maintenance of software that falls within the scope of this software quality plan. Each document below shall be assessed (reviewed) by SQ personnel.

Minimum Documentation Requirement

- Quality Manual
- Software Assurance Plan
- Software Management Plan
- Configuration Management Plan
- Software Requirements Specification
- Risk Management Plan
- Software Safety Plan
- Test Plans (Verification and Validation)
- Software User's Guide
- Software Maintenance Plan
- Interface Control Document(s)
- Test Reports and Artifacts
- Software Version Description Document (VDD)
- Software Requirements Traceability Matrix
- Software Development Records
- Peer Review data packages

Software Reviews

Purpose

This section identifies the number and type of system/subsystem reviews and engineering peer reviews that will be supported by the SQ Personnel. The Software Management Plan (SMP), the project milestone chart, the project's Engineering Peer Review Plan, and the SQ Personnel resource levels determine the reviews that are supported.

Minimum Software Reviews

For each review, SQ will assess the review products to assure that review packages are being developed according to the specified criteria, the review content is complete, accurate, and of sufficient detail, and Requests for Action are captured, reviewed, and tracked to closure. In addition, SQ will assess the processes used to conduct the reviews to determine if appropriate personnel are in attendance, correct information is presented, entry and exit criteria are met, and appropriate documents are identified for update.

Test

SQ personnel will assure that the test management processes and products are being implemented per the Software Management Plan and /or Test Plan(s). This includes all types of testing of software system components as described in the test plan, specifically during integration testing (verification) and acceptance testing (validation).

SQ personnel will monitor testing efforts to assure that test schedules are adhered to and maintained to reflect an accurate progression of the testing activities. SQ will assure that tests are conducted using approved test procedures and appropriate test tools, and that test anomalies are identified, documented, addressed, and tracked to closure. In addition, SQ will assure that assumptions, constraints, and test results are accurately recorded to substantiate the requirements verification/validation status.

SQ personnel will review post-test execution related artifacts including test reports, test results, problem reports, updated requirements verification matrices, etc.

Problem Reporting and Corrective Action

SQ personnel generate, track, and trend assessment findings/nonconformances and observations in the centralized Software Quality Engineering Repository Database (SQERD), available via claritear.contact.us@gmail.com. Reference the SQ Assessment Process WI for details on tracking and trending of assessment findings and observations and the reporting escalation process.

Tools, Techniques and Methodologies

SQ personnel will require access to the following:

Software Quality Tools

- Microsoft Office tools (i.e., Word, Excel, and PowerPoint)
- Android Studio IDE
- Access to idhostinger.com
- Access to the Soft CLARITEAR Mobile Application ware Quality Engineer Reporting Database
- Adobe Photoshop

Project Tools

- CLARITEAR Mobile Application Server
- CLARITEAR Mobile Application SPMP
- CLARITEAR Mobile Application Database

Media Control

SQ deliverables will be documented in one of the following Microsoft software applications: Word, Excel, or PowerPoint. Deliverables will be in soft copy, unless specified otherwise.

Software Quality deliverables, work products, and data items shall be maintained in accordance with the Software Quality Assurance Data Management Plan. This plan provides information on the data item, data category, owner, location, collection frequency, and data retention period.

Supplier Control

SQ personnel will conduct off-site surveillance activities at supplier sites on software development activities. SQ personnel will conduct a baseline assessment of the supplier(s) Quality Management Systems (QMS) to ensure that the supplier(s) have quality processes in place. This initial assessment will help to scope the level of effort and follow-on activities in the area of software quality assurance.

Process and product assessments will be conducted and any findings will be reported and tracked to resolution.

Insight: The application can generate the situation that the user need to fetch the sample question based on the situation from CLARITEAR Mobile Application Database.

Oversight: The ClaritEar Mobile Application Software Quality Assurance Plan provides the framework necessary to ensure a consistent approach to software quality assurance throughout the project life cycle. It defines the approach that will be used by the SAM and Software Quality (SQ) personnel to monitor and assess software development processes and products to provide objective insight into the maturity and quality of the software.

For software that is to be developed, the supplier shall be required to prepare and implement an SQAP in accordance with this standard. Discuss the receipt and review of that plan and how SQ will use the deliverable.

Also state the methods to be employed to assure that the suppliers comply with the requirements of this standard. If software is to be developed under contract, then the procedures for contract review and update shall be described.

Record Collection, Maintenance, and Retention

SQ personnel will maintain records that document assessments performed on the project. Maintaining these records will provide objective evidence and traceability of assessments performed throughout the project's life cycle. Example records include the process and product assessments reports, completed checklists, the SQ Activity Schedule, metrics, weekly status reports, etc.

Training

SQ personnel shall have fundamental knowledge in the following areas/disciplines through prior experience, training, or certification in methodologies, processes, and standards:

- Software Quality Assurance:
- Audits and Reviews
- Risk Management
- Configuration Management
- Software Safety
- Contracts/Contractor Surveillance
- Project-specific Training
- ISD Software Engineering Discussions

It is the responsibility of the SQ personnel to acquire the necessary skills or knowledge in each of the above disciplines. An SQ Training log has been prepared that specifies the type of training and/or on-the-job experience that has been completed, along with the source of the training, and the date of completion.

Risk Management

SQ personnel will assess the project's risk management process against the The CLARITEAR Mobile Application Risk Management Plan and GPG 7120.4. SQ participates in mothly risk management meetings and reports any software risks to the SAM and the project's Risk Manager.

SQA Plan Change Procedure and History

SQ personnel are responsible for the maintenance of this plan. It is expected that this plan will be updated throughout the life cycle to reflect any changes in support levels and SQ activities. Proposed changes shall be submitted to the CLARITEAR Mobile Application Systems Assurance Manager (SAM), along with supportive material justifying the proposed change. Changes to this document require prior approval of the ClaritEar Mobile Application Project CCB Chairperson.

CHAPTER 6
USER MANUAL

GENERAL INFORMATION

The purpose of ClaritEar Mobile Application is to train people's hearing sensitivity in knowing the kind of tone or chord that is being played. With ClaritEar Mobile Application people could gain more knowledge about music chords. Also, this application could help people on becoming new musician.

System Overview

- Everyone can access overall application.
- The website and all its functionalities including the database for the ClaritEar Mobile Application will be hosted by IDHostinger and maintenance by team hamming.
- ClaritEar Mobile Application itself is a graphical user interface.
- As a user, they can choose the instrument that they wanted to play, and choose the correct answer based by the chord that was heard.
- The instrument's chord sound will be generated randomly on ClaritEar's database, and will be send to the application immediately.
- If a user experience some problems on the application, user can send comments and feedback to the Hamming team's e-mail through the ClaritEar Mobile Application and ClaritEar website.

Project References

- http://www.nature.com/news/why-dissonant-music-strikes-the-wrong-chord-in-the-brain-1.11791
- http://listaka.com/top-10-most-difficult-musical-instruments/
- http://www.premierguitar.com/articles/19696-digging-deeper-how-many-chords-are-there

Authorized Use Permission

All users have the ability to use the system functionalities of the mobile application. Access to the ClaritEar Mobile Application backend can only be accessed by the Hamming team.

Points of Contact

Information

Provide a list of the points of organizational contact (POCs) that may be needed by the document user for informational and troubleshooting purposes. Include type of contact, contact name, department, telephone number, and e-mail address (if applicable). Points of contact may include, but are not limited to, help desk POC, development/maintenance POC, and operations POC.

Muhamad Harist Refian Anwar (Team Leader and Designer):

-Email: haristrefiananwar9697@gmail.com.

Help Desk

Provide help desk information including responsible personnel phone numbers for emergency assistance.

Andira Rozawati (Team Member and Deliverables Control):

-Email: andira.rozawati@ui.ac.id.

Team Hamming :

-Email: claritear.contact.us@gmail.com

Organization of the Manual

1.0: General Information

This section contains general information about the ClaritEar Mobile Application, including the system overview, points of contact, project references, and acronyms and abbreviations

2.0: System Summary

This section is an overview of the functionality the ClaritEar Mobile Application contains in non-technical terminology.

3.0: Getting Started

This section contains the information to get started in using the ClaritEar Mobile Application.

4.0: Future Enhancements

This section contains the information about what could be added in the future to make the ClaritEar Mobile Application contain more functionality.

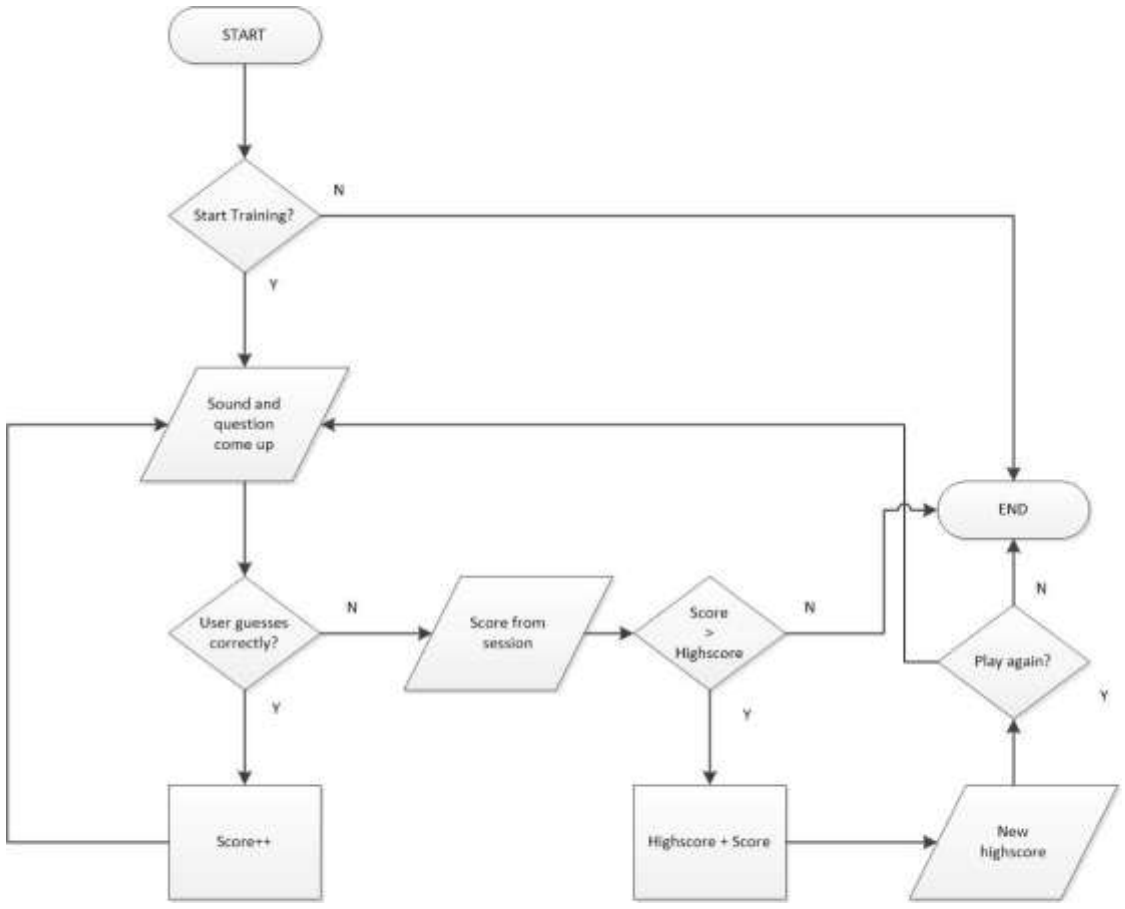
Acronyms and Abbreviations

ClaritEar – a mobile application that could train user's hearing sensitivity through the chord that is being played.

SYSTEM SUMMARY

System Configuration

Briefly describe and depict graphically the equipment, communications, and networks used by the system. Include the type of computer input and output devices.



User Access Levels

Administrator: The administrator for the ClaritEar mobile application shall be able to have access to ClaritEar database. Here, the administrator can remove or add different situation and sample question as needed. The administrator can delete situation and sample question from database.

User only get access to choose situation from the application. User also can get sample question from ClaritEar database based on the situation they choose by clicking generate button.

Contingencies and Alternate Modes of Operation

The PHP server is provided by idhostinger, thus they are the main caretakers of the system. If the system fails the information technology employees will do their best to get the server back up and running. Also idhostinger provide mysql database for generate input by user.

GETTING STARTED

First Display and Register Page

This is the first display when you open ClaritEar Mobile application. To access the application, users must register themselves (making new account). If the user already have an account, then the user can log in into the application.

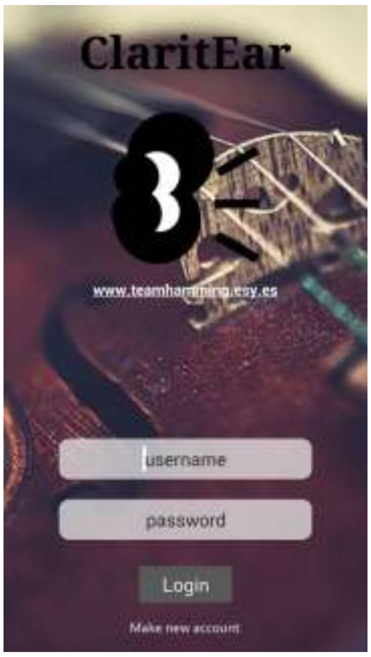


Fig. A. Main Page

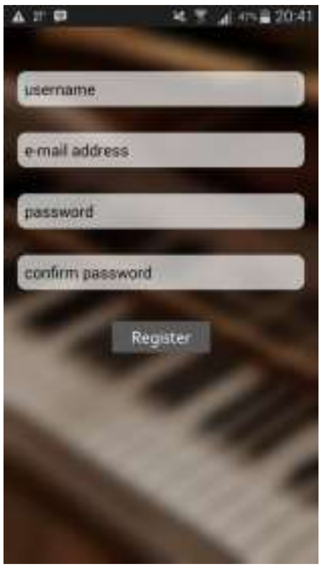
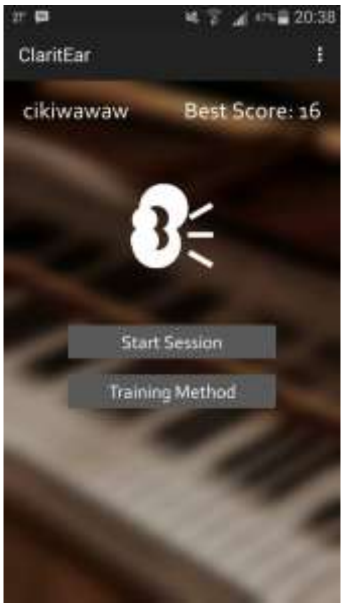


Fig. B. Register Page

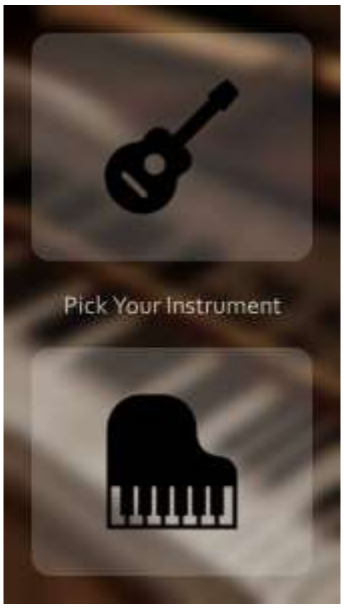
Main Page

After the user successfully enters into application using an account that has been made already, the user is now redirected to the main page, where users can start the tests session or conducting with training for advance.



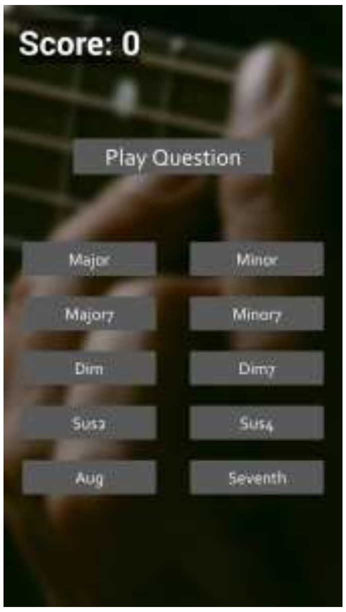
Choosing Instruments Page

In the choosing instruments page, users can choose one from many music instruments that they want to do the test. For now, we provide 2 music instruments, there are piano and guitar.



Test Page

After users choosing an instrument that they want to have a test with, now users redirected to the test page. In this page, users could press the “Play Question” button to hear the chord that is being played. If the users’s answer is correct , then the display will look like on the Fig. C. If the answer is wrong, then the display will look like on the Fig. D.



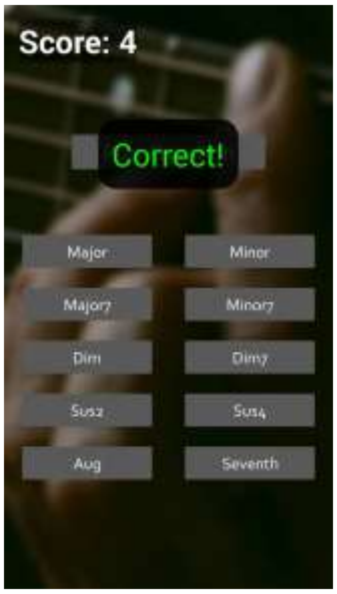


Fig. C. If the answer is correct.

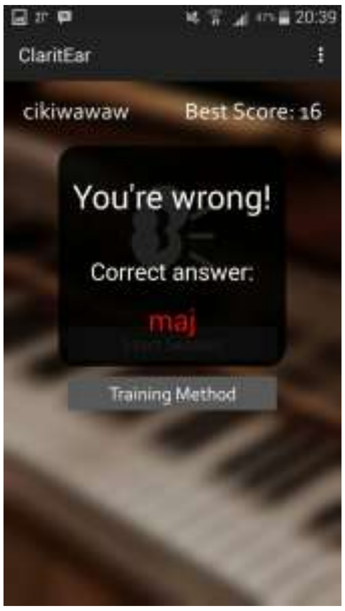


Fig. D. If the answer is wrong.

FUTURE ENHANCEMENTS

Future Functionality

Music instruments and music's chords will be added and deleted to and from the ClaritEar database via the Administrator backend.

Similar Systems – Additional Functionality

- Updates for components – Each components developer provides frequent updates that may want to be downloaded and installed at a future date.

- Contact Us – user can give feedback or comment after using our application via our application by clicking contact us button.

- Information Button – To understand ClaritEar mobile application, user need to access help page and about us page in the application by clicking the button on the top of the layout.

Maintenance Capabilities

Possibilities for site automation could possibly be presets for the addition or deletion of music instruments and chords. Any other automation or improvements could be implemented with the addition of other various ClaritEar mobile application compatible components.

CHAPTER 7
REFERENCES

- www.developer.android.com
- www.androidhive.info
- Grant, Kevin. 2013. Beginning Android Programming: Develop and Design. AmerikaSerikat : Peachpit Press.
- Noalan, Gofrey. 2004. Decompiling Java. Amerika Serikat : Apress.
- Noalan, Gofrey. 2014. Bulletproof Android: Practical Advice for Building Secure Apps. Amerika Serikat : Pearson Education Inc.
- Pressman, Roger S. Software Engineering - A Practitioner's Approach. Fifth edition. The McGraw-Hill companies, Inc.
- Kaner, C., Falk, J., Nguyen, H.-Q. Testing Computer Software. Wiley Computer Publishing, 1999.
- Stelle, James . 2013. The Android Developer&#39;s Cookbook: Building Applications with the Android SDK. Amerika Serikat : Pearson Education Inc. X. Li, K, Nie, S. Nikita, T. Jay Rubinstein, dan E. Atlas Les. Improved Perception of
- Music With a Harmonic Based Algortihm for Cochlear Implant. IEEE transaction neural system and rehabilitation, vol. 21, no 4. July 2013.
- K. H. Jung, J. H. Won, W. R. Drennan, E. Jameyson, G. Miyasaki, S. J. Norton, and J. T. Rubinstein, “Psychoacoustic performance and music and speech perception in prelingually deafened children with cochlear implants,” Audiol. Neurootol., vol. 17, no. 3, pp. 189–197, B. C. J. Moore, “Coding of sounds in the auditory system and its relevance to signal processing and coding in cochlear implants,” Otol.Neurotol., vol. 24, no. 2, pp. 243–254,
- R. Plomp, “Auditory psychophysics,” Annu. Rev. Psychol., vol. 26, pp. 207–232, 1975.
- Y. Y. Kong, A. Mullangi, J. Marozeau, and M. Epstein, “Temporaland spectral cues for musical timbre perception in electric hearing,” J.Speech Lang. Hear. Res., vol. 54, no. 3, pp. 981–994, 2011.