

Industrial Training (Android Bootcamp)

Project Name: Guessing Game

Group No 02

Group Members

S.no	Name	Roll No
1.	Tanbir Alam	05
2.	Saumo Banik	06
3.	Sayantan Barik	07
4.	Dipanjan Basu	08

Index

S.no	Title	Page no.
1.	About the Project	3
2.	Technologies Involved	4
3.	Methodologies Involved	4
4.	System Design	5
5.	Database Schema	8
6.	Screen-shot of the Running Project	9
7.	Future Scope	10

About the Project (Scope)

This project deals with an android-based application whose focus is to act as a guessing game. The app provides the users to select between two modes namely animals and flowers. Each of the modes presents the user with 25 images of animals/flowers each appearing in succession. Each image comes with four options out of which one is the correct name for the given flower or animal. Once a user presses an option, he/she can move on to the next picture. The project aims to alleviate boredom in users interested in a simple gaming structure and enjoy such genre of games.

Technologies Involved

The following technologies has been used to build the project:

- IDE Used: Android Studio 3.4.0
- Testing Device: Xiaomi Redmi Note 6 Pro (Android 9, API 28)
- Database Used: Realm Database
- Programming Language: JAVA

Methodologies Involved:

To build this project we followed the basic **Software Development Life-Cycle** model i.e. the Waterfall Model. Following are the steps followed:

- Feasibility Study: The first and foremost task was to find out if the project was feasible. The project was handed to us by our mentors and after going through the outcomes and aims of the project with them it was found that the project could be implemented.
- Requirement Gathering/Analysis: This phase was also done by the help of our mentors. Strict guidelines were provided dictating the bare minimum outcome that the project should achieve. Following those guidelines, the following were the requirements:
 1. The app should contain a login form where in the user can sign in using a username.
 2. There should be two modes to select from namely flower and animals.
 3. Each mode should have at least 25 images.

4. There should be a timer present which will determine the maximum time a user will get to guess one image.

5. There will also be concepts of lives. A user will lose if he/she gives multiple wrong answers which cross a certain limit.

6. Finally, the user can view their score and see where they stand among various other users.

7. There should be at least 10 results on the scoreboard.

In lieu of the requirements and the skills of the members the project was decided to be built on the Android Studio IDE using JAVA as the programming language and Realm as the database.

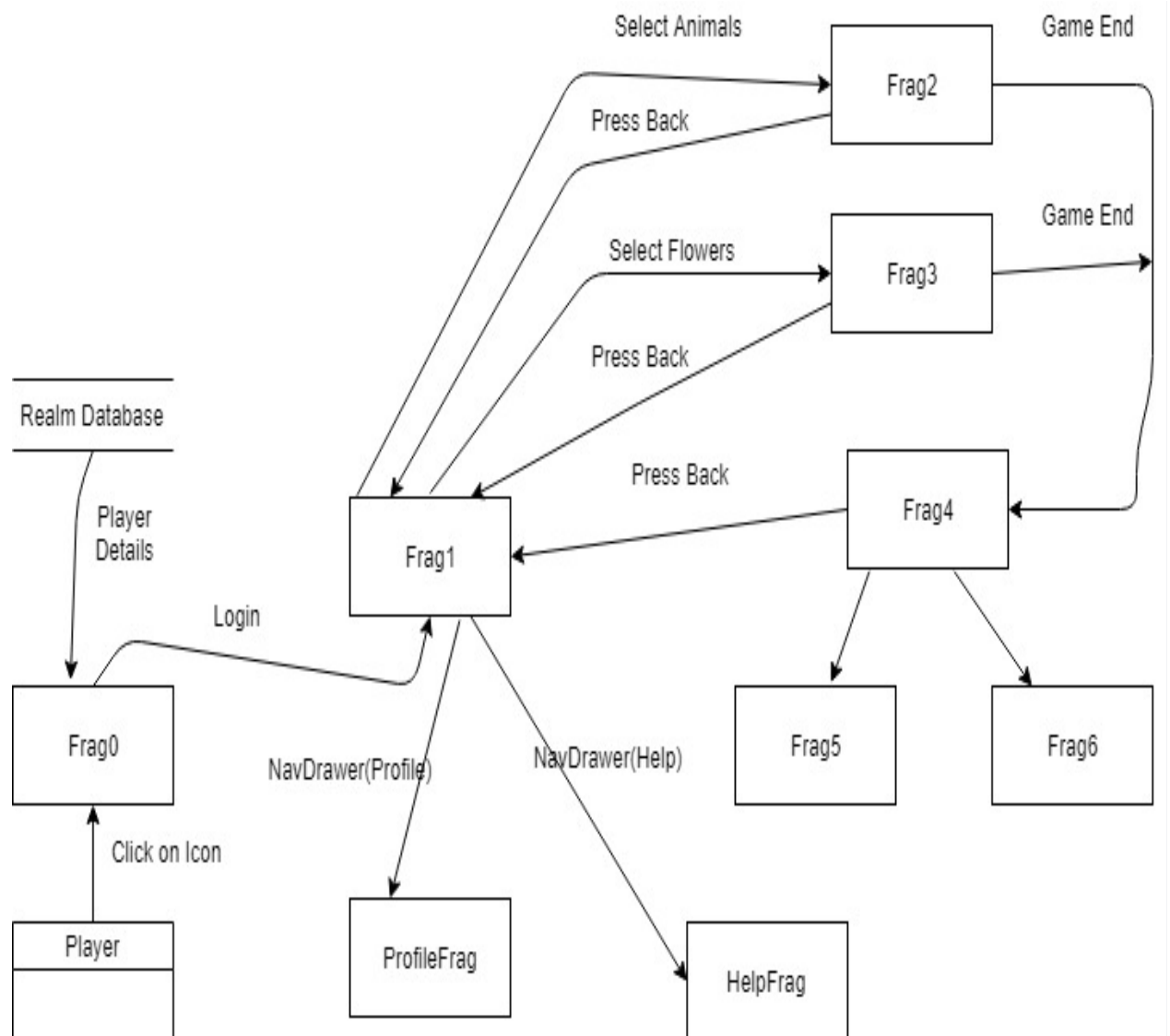
- Design: The basic layout and UI were designed before moving onto implementation.
- Coding/Implementation: The back-end part of the project was written on JAVA.
- Testing: Various tests were done using the device mentioned. The app was also tested on other mobile devices during various stages.
- Deployment: The app is finally ready to be submitted as a completed project.
- Maintenance: N/A (unless required by the mentors).

System Design

The Guess Game has one main activity and nine fragments working in its back end. Following are the descriptions and working of each of them.

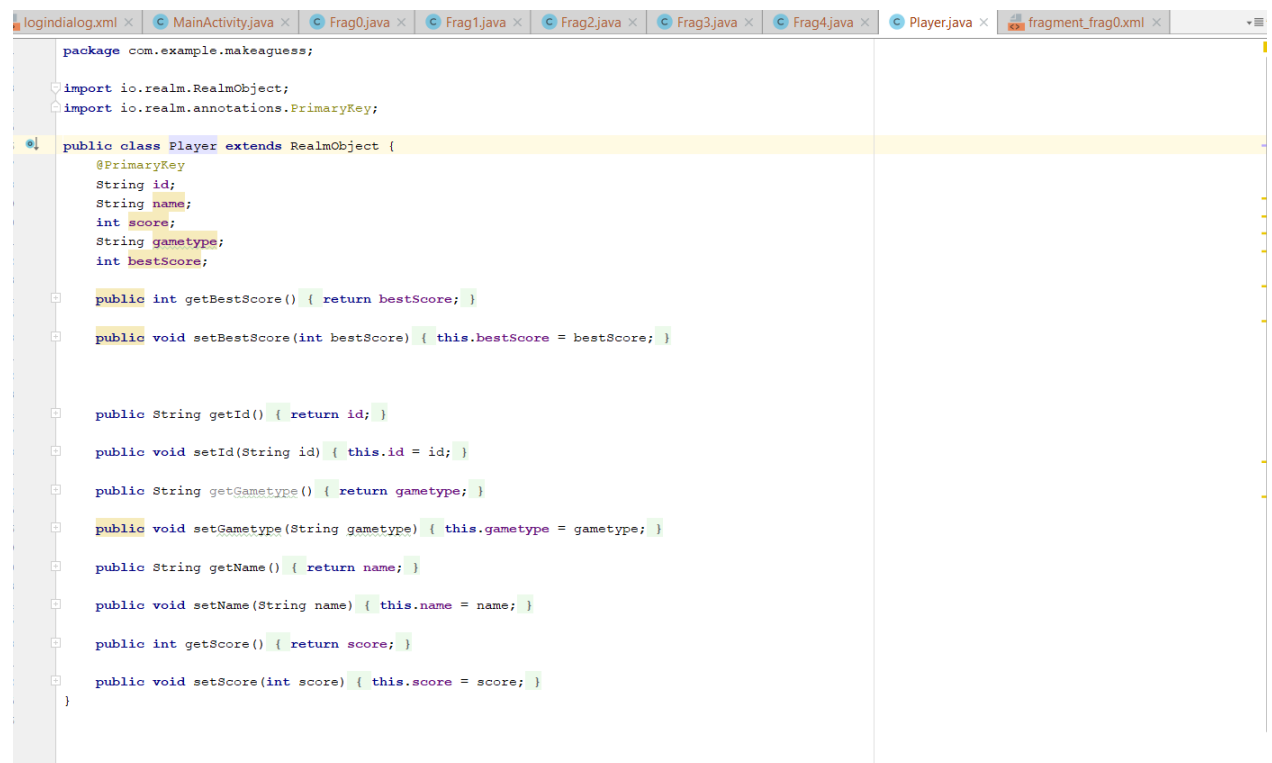
- **Main Activity:** This is used for holding all the fragments, one at a time, as required.
- **Frag0:** The initial fragment that is loaded when the user clicks on the game icon. This fragment loads a dialog fragment when clicked on, which prompts the user to enter a player name to enter.
- **Frag1:** The fragment which is loaded when the player has entered the game and is ready to play. Player can select between Animals and Flowers mode. This is also the fragment where the player can access the Navigation Drawer and go to the profile page, scoreboard etc.
- **Frag2:** The fragment which is loaded when the player selects animal mode. The layout of this fragment mainly consists of an image featuring an animal. Below the image are four options out of which only one is correct. The user can click on next to move on to the next animal. The user can see a timer of 5s running in the upper left corner. Once the timer runs out the user cannot guess the given animal anymore and will be provided with the correct answer. This procedure will continue until all the images are exhausted or the user loses their lives after attempting multiple wrong answers.
- **Frag3:** The fragment which is loaded when the player selects flowers mode. The layout of this fragment mainly consists of an image featuring a flower. Below the image are four options out of which only one is correct. The user can click on next to move on to the next flower. The user can see a timer of 5s running in the upper left corner. Once the timer runs out the user cannot guess the given animal anymore and will be provided with the correct answer. This procedure will continue until all the images are exhausted or the user loses their lives after attempting multiple wrong answers.
- **Frag4:** This contains a viewpager to be able to access two nested fragments.
- **Frag5:** This provides the player with the Animal scoreboard.
- **Frag6:** This provides the player with the Flower scoreboard.
- **ProfileFrag:** Displays the profile information of the player.

- **HelpFrag:** Gives user information about what to do in the game.



Database Schema

The following POJO class's member variables are the fields of the Realm database and the entire POJO class represents the database schema.



```
package com.example.makeaguess;

import io.realm.RealmObject;
import io.realm.annotations.PrimaryKey;

public class Player extends RealmObject {
    @PrimaryKey
    String id;
    String name;
    int score;
    String gametype;
    int bestScore;

    public int getBestScore() { return bestScore; }

    public void setBestScore(int bestScore) { this.bestScore = bestScore; }

    public String getId() { return id; }

    public void setId(String id) { this.id = id; }

    public String getGametype() { return gametype; }

    public void setGametype(String gametype) { this.gametype = gametype; }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

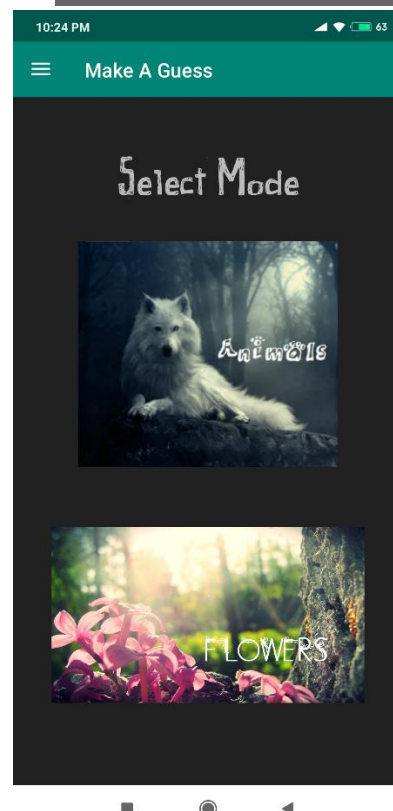
    public int getScore() { return score; }

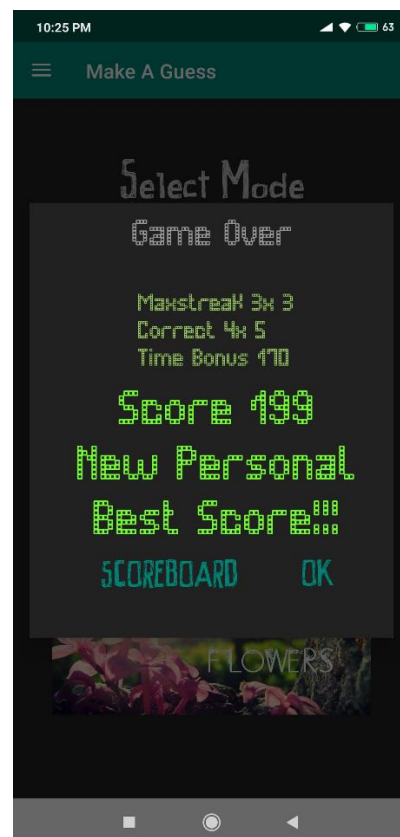
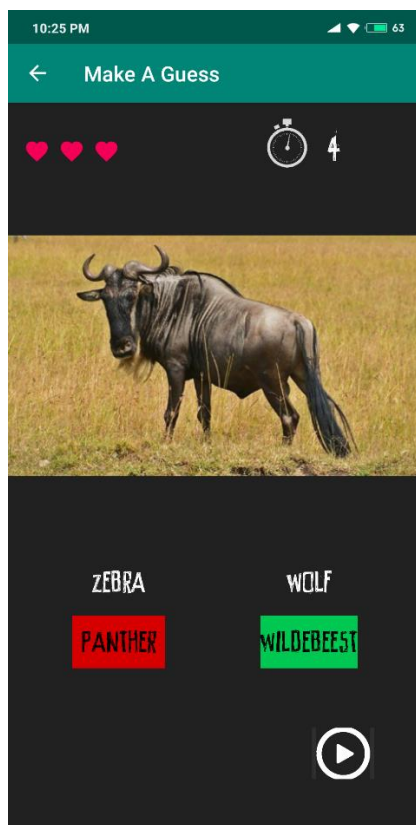
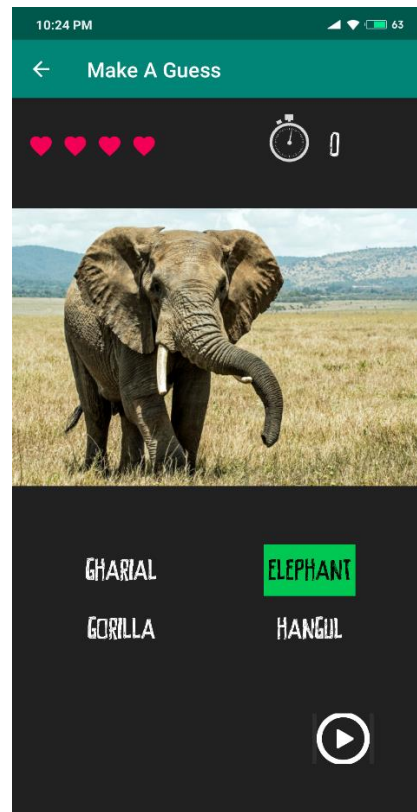
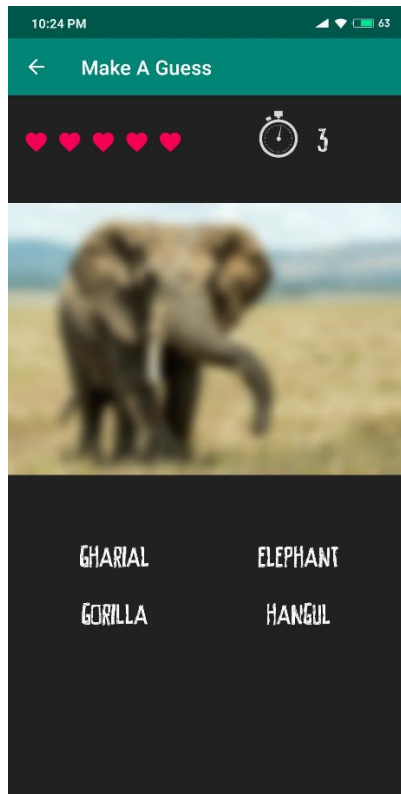
    public void setScore(int score) { this.score = score; }
}
```

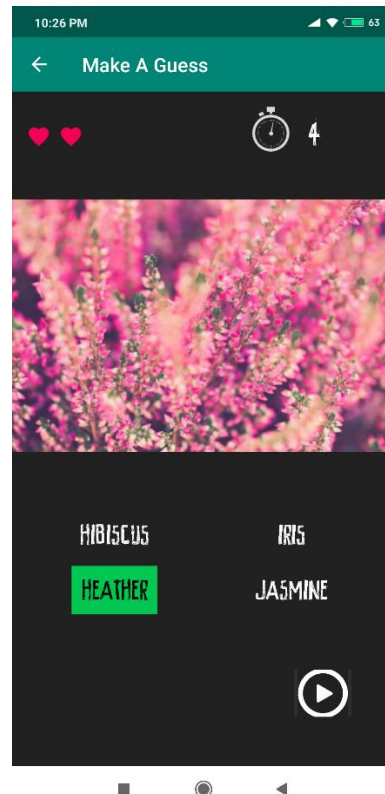
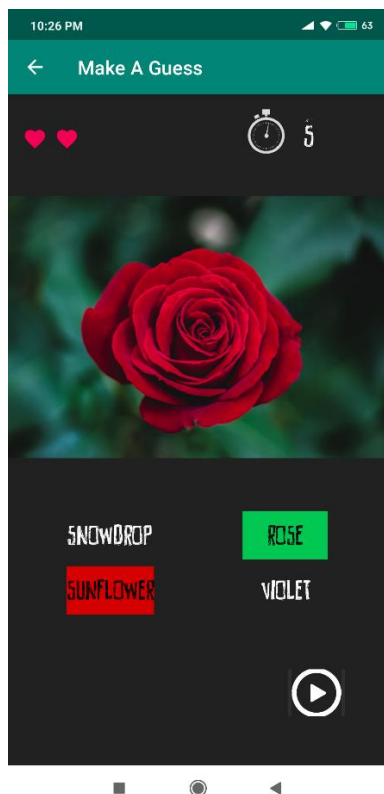
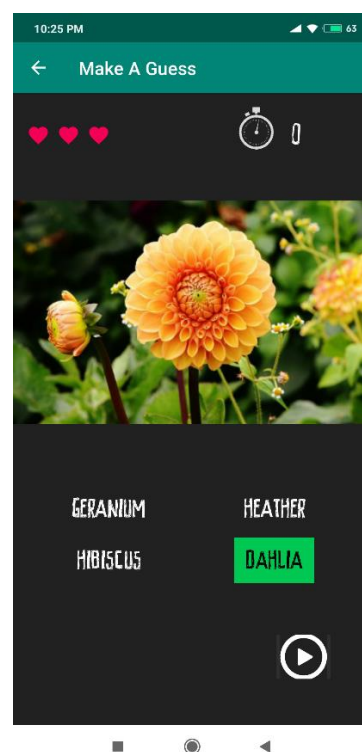
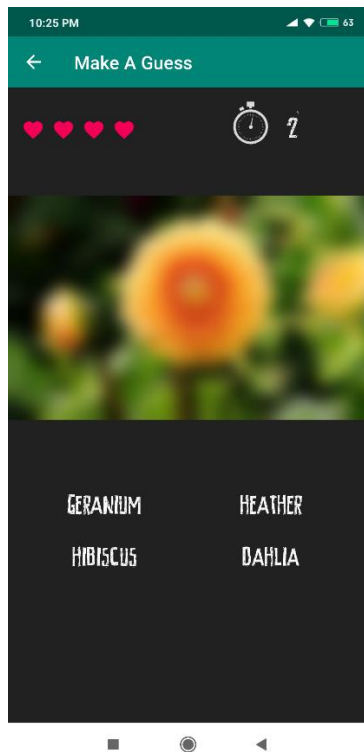
The fields of the database are:

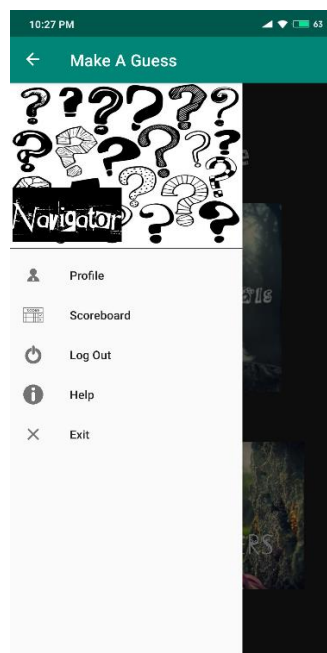
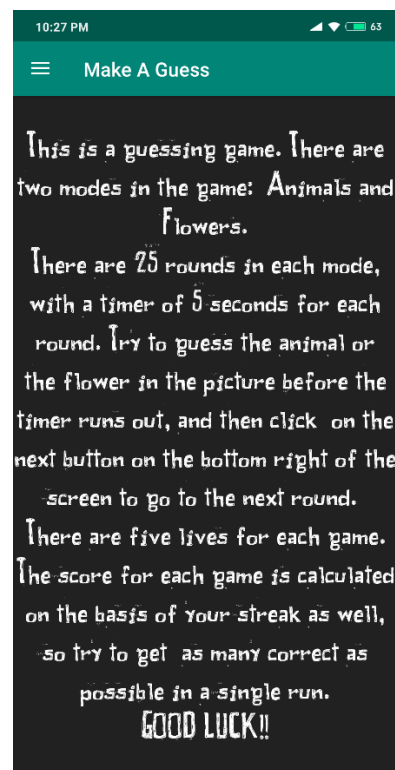
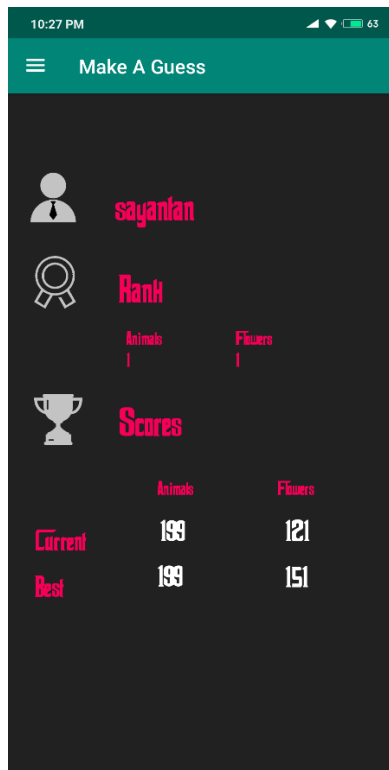
- id: created by concatenating the name and gametype
- name: entered by user on login
- score: user's score
- gametype: flowers or animals
- bestscore: user's personal best score

Screen-shots of the Running Project









Future Scope:

The project can be improved in certain ways. If the user gets an answer wrong then hint can be given which can help the user guess the answer. Also, for each user the correct answers given can be stored in the database. The user can then click on any of these to get a detailed description of the object viz. Animal/flower. To make it harder for the user to guess the object the images can be blurred just a little.