YILDIZ TECHNICAL UNIVERTSITY DEPARTMENT OF COMPUTER ENGINEERING

HOMEWORK

Course Name:	Introduction to Mobile Programming
Course Group:	Group 1
Instructor Name:	Assistant Prof. Dr. M. Amaç Güvensan
Student ID:	11011027
Student Name and Surname:	Mustafa Berkay Mutlu
Delivery Date of the Assignment:	25.11.2015

General Overview of the Application

In this project, I developed a Movie Quiz game on Android platform. Each player is registering using his/her name and day of birth. Then a game menu is displayed which consists of two buttons, “Play” and “Scores”. Player can either play the quiz or look at the scores of the previous players played on that phone. If player choses to play the quiz, after quiz is finished, player’s score, player’s answers and true answers to each question is displayed. If player wants to see the all scores of previous players (and their games) he/she can click Scores button on the main menu and look at the scores. SQLite database is used to store player, question and player answers information.

Technical Details

Activities Used

RegisterActivity

This Activity is Launcher Activity and it is used to register players to the database. The important part of the registration is, player is not registered to the database in this Activity. Instead, the player information is passed to the *MenuActivity* and then to the *QuizActivity*. The main reason of this is players just want to look at the scores and not play with the game. It was the best option to do, because I don’t want my database to full of players who didn’t played any games.

QuizActivity

This is the Activity for playing the game. It has couple of *TextView*’s for displaying the question, and number of the question as well as buttons for answering the question. This Activity also will not reset its View values when the orientation of the phone is changed. After all the questions are answered by player, this Activity opens the *GameFinishedActivity* for displaying the score of the player and actual answers of the questions.

GameFinishedActivity

This is the Activity for displaying the score of the player, answers of the player and actual answers of the questions. This list is built with *ListView* with a custom Adapter. You can see it’s XML in the “layout” folder named as “list_item_answers.xml”

MenuAcitivity

This Activity is used for displaying a game menu. After user is registered, this Activity opens and user have choose if he/she either wants to play the game or just want to see the scores of previous games. If he/she choses to play, then *QuizActivity* opens and game starts. If he/she choses to see the scores of previous games then *ScoresActivity* opens.

ScoresActivity

This Activity is used for displaying the scores of the previous games. Scores are listed in decreasing order. So higher score will be at the top. On each line there is a rank (1^{st}, 2^{nd} etc.), player's name and player's score. This list is built with ListView with a custom Adapter. You can see it's XML in the "*layout*" folder named as "*list_item_scoreboard.xml*"

XML Files

activity_game_finished.xml

This XML file holds the layout design for the GameFinishedActivity. It has 2 *TextView* for displaying the score of the player and one *ListView* for displaying the both actual answers and player answers. *LinearLayout* is used in this layout as main layout.

activity_menu.xml

This XML file holds the layout design for the *MenuActivity*. It has one *TextView* for application title and two Buttons. Buttons are for playing and displaying the scores of previous games. *LinearLayout* is used in this layout as main layout.

activity_quiz.xml

This XML file holds the layout design for the *QuizActivity*. This layout has lots of *TextView*'s for displaying player name, question, current question number and total question number. It also has two Button's (True and False buttons) for answering the question. *RelativeLayout* is used in this layout as main layout.

activity_register.xml

This XML file holds the layout design for the *RegisterActivity*. This is the first layout users will see. Users have to register in this Activity using their name and birthday information in order to play the

game. *LinearLayout* is used in this layout as main layout. To get the name from users an *EditText* is used and for the birthday, a *DatePicker* Dialog Fragment is used. Players will see their picked dates on the bottom of their names.

activity_scoreboard.xml

This XML file holds the layout design for the *ScoreboardActivity*. This is the layout for displaying the scores of previous games. It has one *TextView* for title and one *ListView* for scores. *LinearLayout* is used in this layout as main layout. You can read the details about the *ListView* in the "list_item_scoreboard.xml" and "list_view_header_scoreboard.xml" XML files.

list_item_answers.xml

This is the layout file for inflating the *ListView* in the *GameFinishedActivity*. Each row has a question number on the left side, the question itself on the center top and a one-line information about whether the player answered this question correctly or not in the center bottom. The information message mentioned lastly is also colored green or red depending on the correctness of the player's answer. *RelativeLayout* is used in this layout as main layout

list_item_scoreboard.xml

This is the layout file for inflating the *ListView* in the *ScoreboardActivity*. It has player's rank on the side, player's name and age on the center and the score on the right side. *RelativeLayout* is used in this layout as main layout. It also has a Header so that users can easily understand which value is which. You can see the details of the Header in the "list_view_header_scoreboard.xml" XML file.

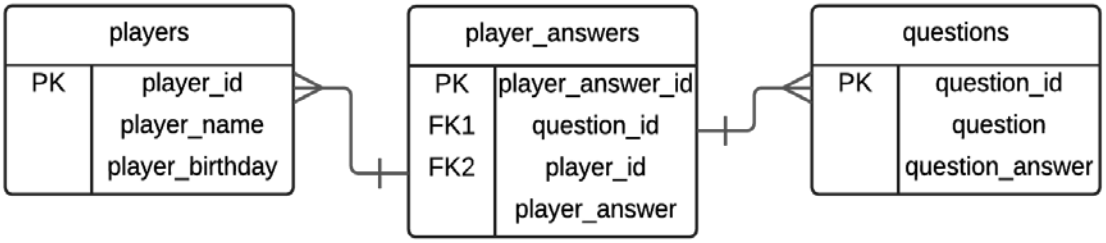
list_view_header_scoreboard.xml

This is the layout file for the Header of the scoreboard *ListView*. In order to make *ListView* easy to understand, this Header layout is used. It only has *TextView*'s inside, representing the columns below them. Starts with rank, name, age and lastly score. *LinearLayout* is used in this layout as main layout.

Database

Requirement of storing system for having lots of questions, players and answers, a database fits better than a file or SharedPreferences. Therefore SQLite database is used to keep player information, questions, and answers of players. With this database design, a player can play more than one game and get different scores with the same player information, as it should be.

The database has three tables and those are “players”, “questions” and “player_answers”.



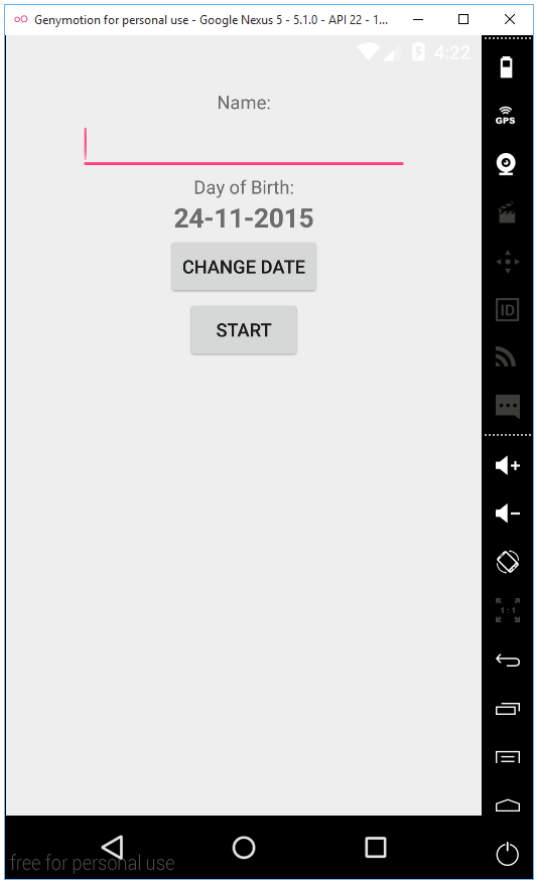
You can see the table names in the *DatabaseContract* class and all the database functions in the *LocalDatabaseHandler* class. Also, *LocalDatabaseHandler* is a Singleton class, so every *getInstance(Context)* function call will return the same object.

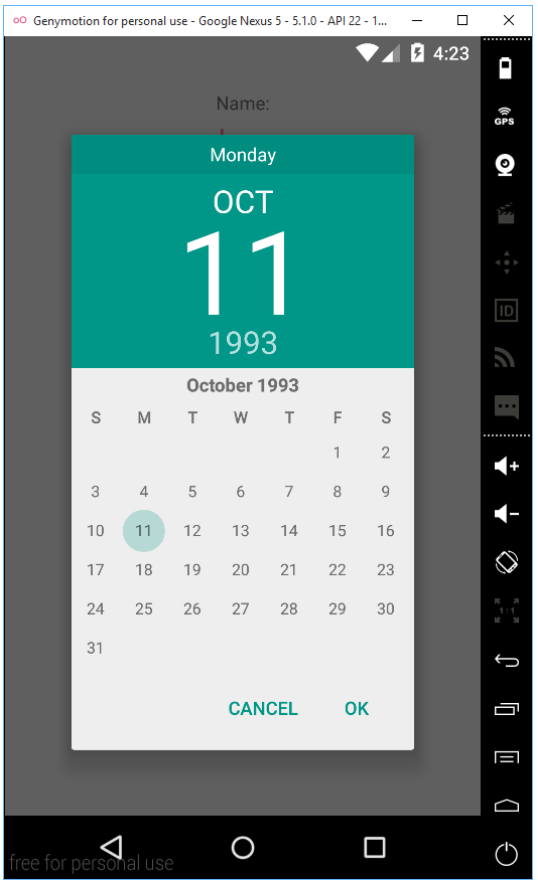
Questions are initialized in the *LocalDatabaseHandler* class. There are lot of methods for basic database operations especially for selecting and inserting. Those method’s method names are self-explanatory and all of them have their own Javadoc. For example *AddPlayer(Player player), AddQuestion(SQLiteDatabase db, Question question)* and *AddPlayerAnswer (PlayerAnswer playerAnswer)* methods are used to insert entries to tables. *GetAllQuestions(),GetPlayer(long playerID), GetQuestion(long questionID), GetPlayerAnswers(long playerID)* methods are used for select operations.

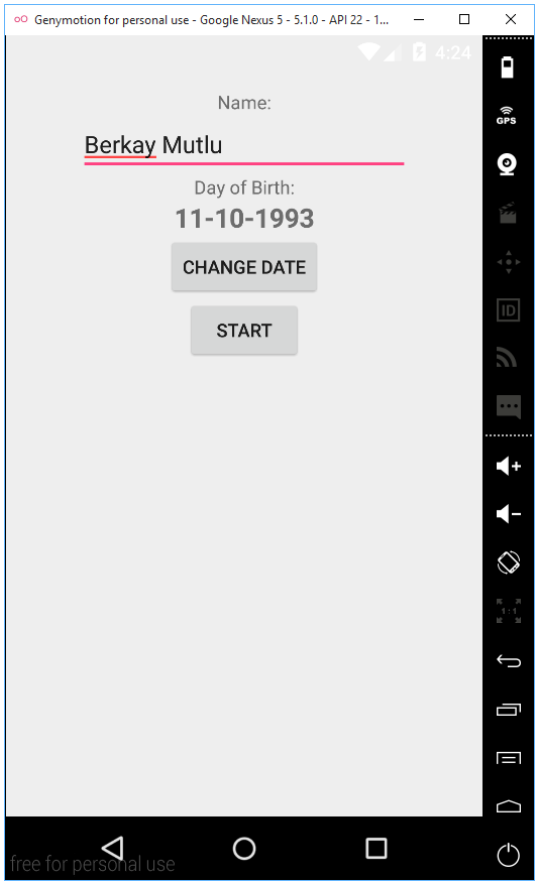
Models

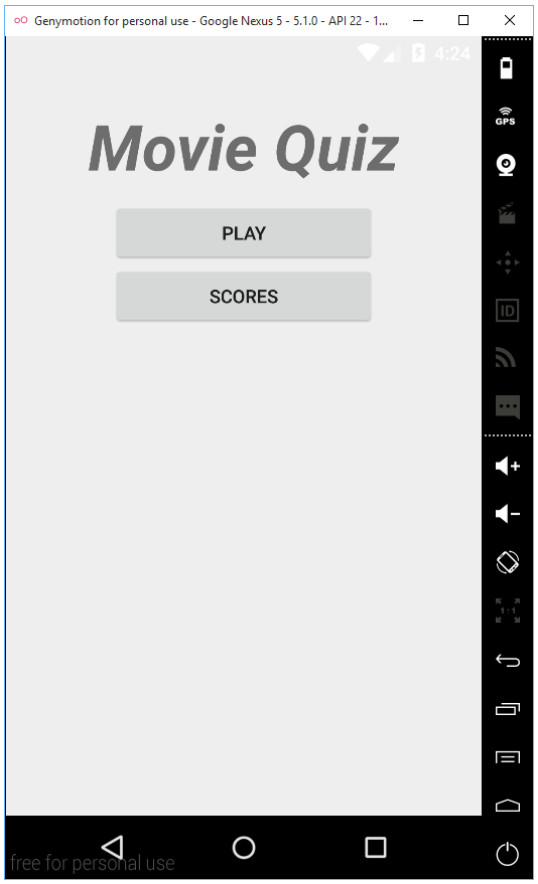
Each entity in the database is represented as a Java class. “players” table’s entities represented as *Player* class, “player_answers” is represented as *PlayerAnswer* class and “questions” table is represented as *Question* class. Each class in Java has its own properties just like in the database tables so that we can do basic operations on the database and return a Java object.

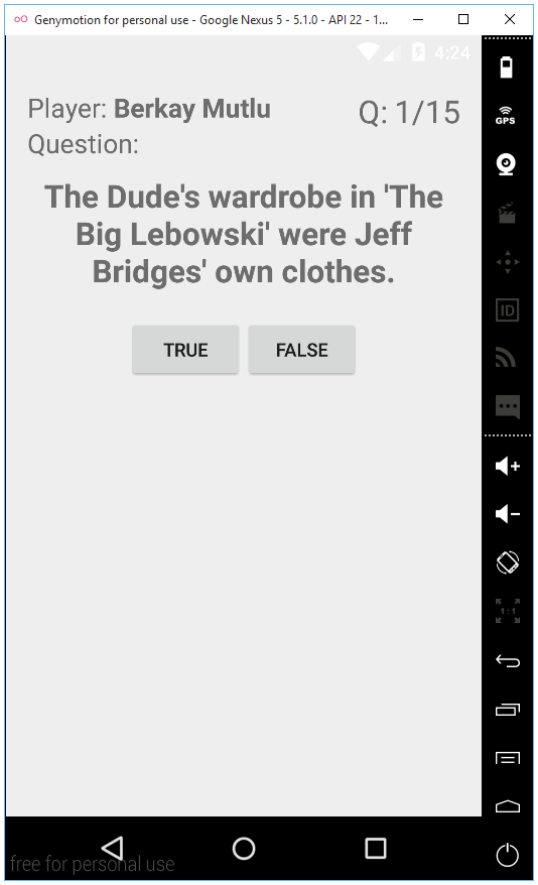
Screenshots

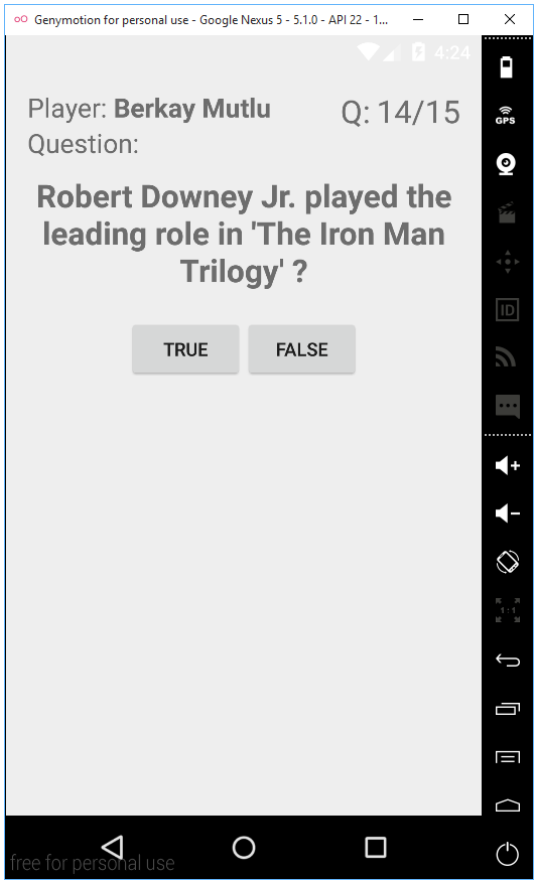


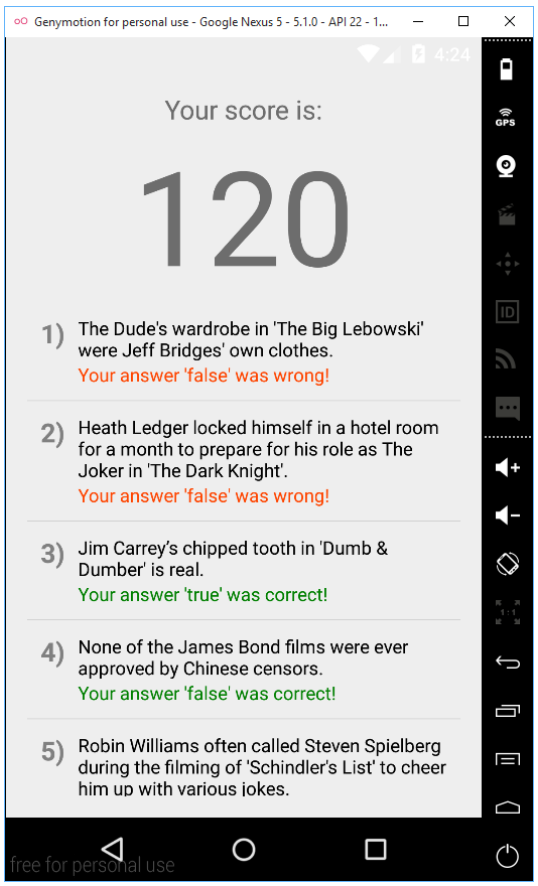


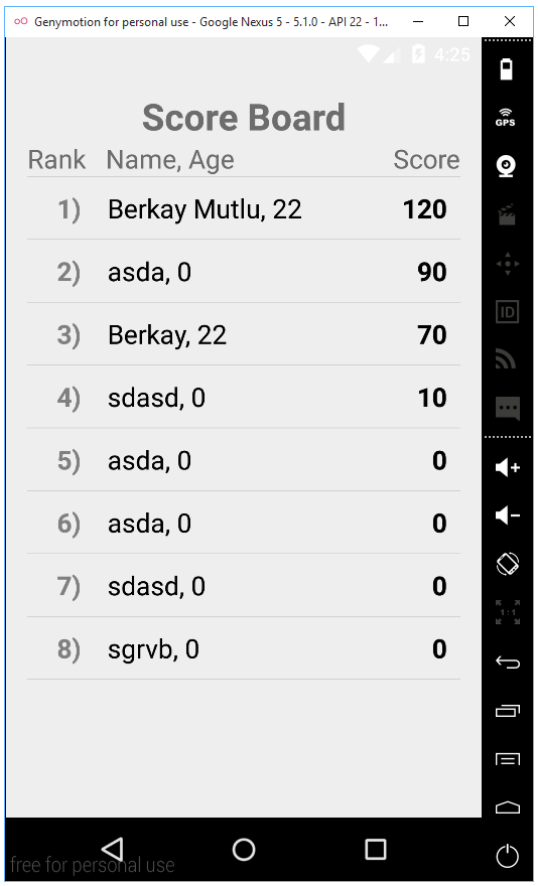





Genymotion for personal use - Google Nexus 5 - 5.1.0 - API 22 - 1...
4:24
Player: Berkay Mutlu
Q: 1/15
Question:
The Dude's wardrobe in 'The Big Lebowski' were Jeff Bridges' own clothes.
TRUE
FALSE
free for personal use


Genymotion for personal use - Google Nexus 5 - 5.1.0 - API 22 - 1...
4:24
Player: Berkay Mutlu
Q: 14/15
Question:
Robert Downey Jr. played the leading role in 'The Iron Man Trilogy' ?
TRUE
FALSE
free for personal use


Genymotion for personal use - Google Nexus 5 - 5.1.0 - API 22 - 1...
4:24
Your score is:
120
1) The Dude's wardrobe in 'The Big Lebowski' were Jeff Bridges' own clothes.
Your answer 'false' was wrong!
2) Heath Ledger locked himself in a hotel room for a month to prepare for his role as The Joker in 'The Dark Knight'.
Your answer 'false' was wrong!
3) Jim Carrey's chipped tooth in 'Dumb & Dumber' is real.
Your answer 'true' was correct!
4) None of the James Bond films were ever approved by Chinese censors.
Your answer 'false' was correct!
5) Robin Williams often called Steven Spielberg during the filming of 'Schindler's List' to cheer him up with various jokes.
free for personal use


Genymotion for personal use - Google Nexus 5 - 5.1.0 - API 22 - 1...
4:25
Score Board
Rank Name, Age Score
1) Berkay Mutlu, 22 120
2) asda, 0 90
3) Berkay, 22 70
4) sdasd, 0 10
5) asda, 0 0
6) asda, 0 0
7) sdasd, 0 0
8) sgrvb, 0 0
free for personal use