

CS102**Fall 2016/2017**

Instructor:

Öznur Taştan Okan

Assistant:

Mohammed MobayenPro-
ject
Group**3/7****~Music Time~*****HI-5*****Group Members:**

Aslı ALPMAN 21501747

Cansu Canan CEYHAN 21400998

Giray Baha KEZER 21301937

Kenan ŞAHİN 21301469

Semih TEKER 21300964

Criteria	TA/Grader	Instructor
Presentation		
Overall		

Detailed Design Report**(First Draft)****25 November 2016****1. Introduction**

In this assignment, we are going to develop a music quiz application which is similar to Who Wants to Be a Millionaire [1]. Briefly, users will be able to test themselves in different categories which are composed of wide variety of songs. Score system will be similar to Who Wants to Be a Millionaire's score system which is composed of thresholds [1]. The more question user will correct, the more point user will get. Moreover, users are classified in different ranks by their scores.

2. System Overview**Quiz**

- **Properties:** half, twice, addtime
- **Methods:** useHalf, useTwice, useAddtime

BaseQuestion

- **Properties:** header, answer, options, mp3
- **Constructor:** BaseQuestion

LifeLines

- **Abstract Methods:** useHalf, useTwice, useAddTime

Account

- **Constructor :** Account
- **Methods :** logIn, signUp

Avatar

- **Properties :** avatar1, avatar2, avatar3
- **Methods :** setAvatar, avatarSelected

Game

- **Properties:** time
- **Constructor:** Game
- **Methods:** play, createQuiz, isQuizEnd, isAnswerCorrect, addScore, isTimeUp, giveUp

Database

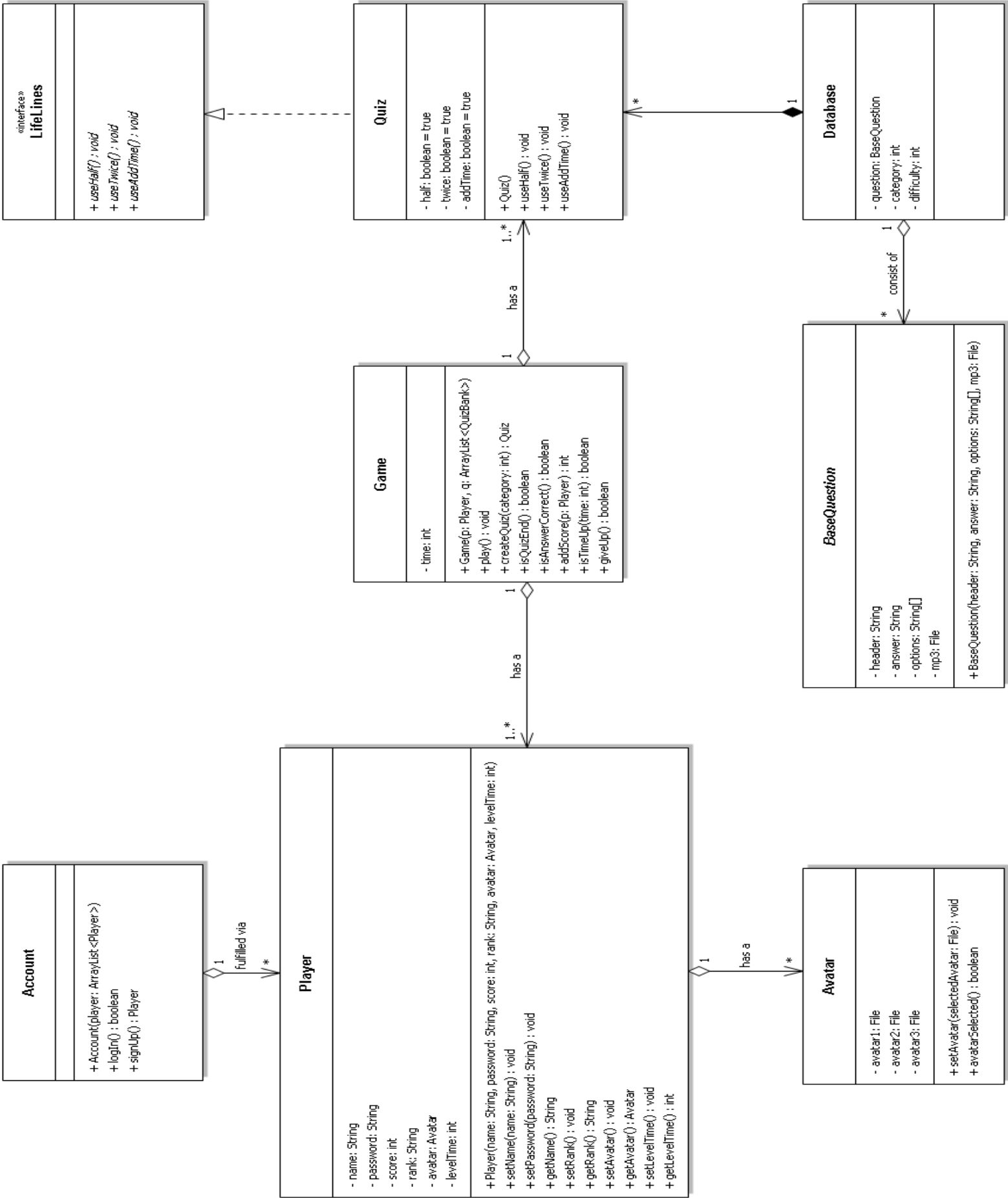
- **Properties:** question, category, difficulty

Player

- **Properties:** name, password, score, rank, avatar, level time
- **Constructor:** Player
- **Methods:** setName, setPassword, getName, getRank, setRank, setAvatar, getAvatar, setLevelTime, getLevelTime

Our program is going to be a desktop application. While developing this desktop application, databases and JavaFX will be used in order to simplify the design. Objects of BaseQuestion class which holds quiz questions as a whole and additional attributes such as difficulty and category of questions will be hold in database. Furthermore, JavaFX will be used to easily create and design user interfaces. To conclude, we will develop a desktop application by using databases and JavaFX.

3. Core Design Details



Quiz Class

useHalf():

A method which makes half, instance variable of Quiz class, false. Besides, it eliminates 2 wrong answers from options.

useTwice():

A method which makes twice, instance variable of Quiz class, false. In addition, it enable player to select 2 options.

useAddtime():

A method which adds time, instance variable of Quiz class, false. Besides, it increases time by default amount of time.

BaseQuestion Class

BaseQuestion():

A constructor which enables developer to create questions as a whole. It takes header of question, options of question, correct answer of question and mp3 file as parameters.

Account Class

Account():

A constructor which takes an array list composed of objects from player class.

login():

A method which checks that player has an account before or not. If has, than allows to log in the game.

signUp():

A method to create a new player account.

Avatar Class

setAvatar():

A method which sets the selected avatar from the given file.

avatarSelected():

A method to check which avatar is selected.

Game Class

play(p: Player, q: ArrayList<QuizBank>):

A method which starts the game and execute the necessary methods in it. To start the game firstly the Game needs to a Player and a Quiz consists of twelve questions. During the game, Questions are asked one by one to Player and each four questions the difficulty level of the questions is increased. The answers is checked by the isAnswerCorrect(). Each Quiz has three lifelines (Half, Twice, Add time) and player can use them once for each quiz. Each question has a period determined by the player and if the time is up, the quiz is finished. When the answer is correct the score is updated by the addScore() and when it is wrong, then the quiz is finished. Also, the player can give up the quiz whenever s/he wants via giveUp(). If the player gives all

correct answers to questions, quiz is finished. All the situations for ending, the score is shown as a table and the player obtains the threshold score.

createQuiz(category: int):

A method which creates new Quiz consist of questions which has the category (or random) selected by the user before the game start.

isQuizEnd():

A method which checks whether the all questions are answered or whether the quiz is interrupted by the user or whether the wrong answer is given.

isAnswerCorrect():

A method which checks the selected choice to control it is correct or not.

addScore(p: Player):

A method which adds the score to users' score table if the answer is correct.

isTimeUp(time: int):

A method which checks the time determined by the player is up or not. And if the time is up, it derives the quiz end.

giveUp():

A method which checks the 'Give Up' option is used or not. And if it is selected, then derives the quiz end.

Database

Database provides to keep information about questions. Each question has a BaseQuestion(header, answer, options, audio record) and a category (pop, rock, jazz, hip-hop, classic, country, reggae, speech, film or random) and a difficulty level (easy, middle or hard). When a quiz is created, game needs to a certain category selected by the player and holds twelve questions the same category [4]. By the way the difficulty level of the questions is increased each four questions so every quiz has four easy, four middle, four hard level questions. To create the random quiz for each game, database is very useful for holding all questions which is created by the programmers as a chart.

Player Class

Player():

A constructor which takes the name, password, score, rank, avatar, level time as parameters and according to the parameters enable program to create a player.

setName(name: String):

A method which sets the name of user who plays the game.

getName():

A method which returns the name of user.

setPassword(password: String):

A method which sets the password according to user preference.

setRank():

A method which sets the rank by looking the success of the user in the game.

getRank():

A method which returns the current rank of the user.

setAvatar():

A method which sets the avatar according to user preference.

getAvatar():

A method which returns the selected avatar.

setLevelTime():

A method which sets the difficulty level according to user's time preference.

getLevelTime():

A method which returns the selected level time.

4. Task Assignment

Task sharing will be as follows:

- Account, Avatar and Player classes are going to be written by Cansu Canan Ceyhan.
- Game and Database classes are going to be written by Kenan Şahin and Semih Teker.
- Quiz and BaseQuestion classes are going to be written by Aslı Alpman.
- GUI is going to be written together.
- Questions are going to be created by us and they are going to be done together.

5. References

- 1."Who Wants To Be A Millionaire Official Site ..." *Who Wants To Be A Millionaire*. N.p., n.d. Web. 21 Oct. 2016. official rules.
- 2."UML Design Pattern." UML Design Pattern. N.p., n.d. Web. 25 Nov. 2016.
- 3."Explanation of the UML Arrows." Oop - Explanation of the UML Arrows - Stack Overflow. N.p., n.d. Web. 25 Nov. 2016.
- 4."Free Java Course." Java For Complete Beginners - Databases. N.p., n.d. Web. 25 Nov. 2016.