

DB Szenariobeschreibung

Hammann, Wilpert

June 2020

1 Introduction

Innerhalb dieser Szenario-Beschreibung soll beschrieben werden, wie der Anwendungsfall "Wissenstest spielen" ablaufen soll. Dabei wird besonders Wert auf die zu benutzenden Klassen und deren Erstellung gelegt.

2 Beschreibung

Spieler:

Um ein neues Objekt Spieler zu erstellen wird die Methode `interactiveGetOrCreatePlayer()` aufgerufen, wobei ein Dialog nach dem Namen des Spielers fragt. Sollte der Spieler schon in der Datenbank stehen, so wird er nicht neu erstellt

Fragen:

Mit der Methode `getQuestions(List<?> categories, int amountOfQuestionsForCategory);` werden die Fragen aus den gegebenen Kategorien erstellt. Die Kategorien wählt der Spieler aus (mindestens 2). Die Maximale Anzahl an Fragen pro Kategorie kann auch angegeben werden

Game:

Sobald ein Spieler-Objekt und die Fragen-Liste existieren, wird ein neues Game-Objekt erzeugt.

Spielen:

Mit dem Game-Objekt wird dann durch die Methode `interactivePlayGame(Object game);` das eigentliche Spiel gestartet

Persistenz:

Die Methode `persistGame(Object game);` kann ein gespieltes Spiel mit dem Spieler persistieren.

Spielen:

Wenn dem Spieler eine Frage gestellt wird, so bekommt er alle Antwortmöglichkeiten

angezeigt, wobei eine ausgewählt werden muss. Danach bekommt er sofort eine Auskunft über seine Antwort.

Auswertung:

Wenn mehrere Spieler das Spiel gespielt haben, so können diese miteinander verglichen werden.

3 Zusammenfassung

Aufzählung der benötigten neuen Klassen

- player
- game
- gameQuestion

Annotation von game

```
@Entity
@Table( schema = " hamwil ")
class Game
    @Id
    @GeneratedValue
    private int id
    @ManyToOne
    Player player;
    @Temporal(TemporalType.TIMESTAMP)
    Date starttime;
    @Temporal(TemporalType.TIMESTAMP)
    Date endtime;
    @OneToMany(mappedBy = "game")
    private List<GameQuestion> gameQuestions;
```

Annotation von Player

```
@Entity
@Table( schema = " hamwil " )
class Player
    @Unique
    @Id
    String name
    @OneToMany(mappedBy = "player")
    private List<Game> games;
```

Annotation von GameQuestion

```
@Entity
@Table(schema = "hamwil")
class GameQuestion
@Id
@GeneratedValue
private int id;
@ManyToOne
Question question;
@ManyToOne
Game game;
```