



# JavaFX Kviz igra

CS102

Objekti i apstrakcija podataka

## Projektna dokumentacija

14.06.2021

Student: Darko Tasevski, 4374

## 4 Sadržaj

1.	Predlog teme .....	3
2.	Opis funkcionalnosti .....	3
3.	Struktura aplikacije .....	4
4.	Korisničko uputstvo .....	7
4.1	Početna stranica .....	7
4.2	Stranica sa prikazom rezultata .....	8

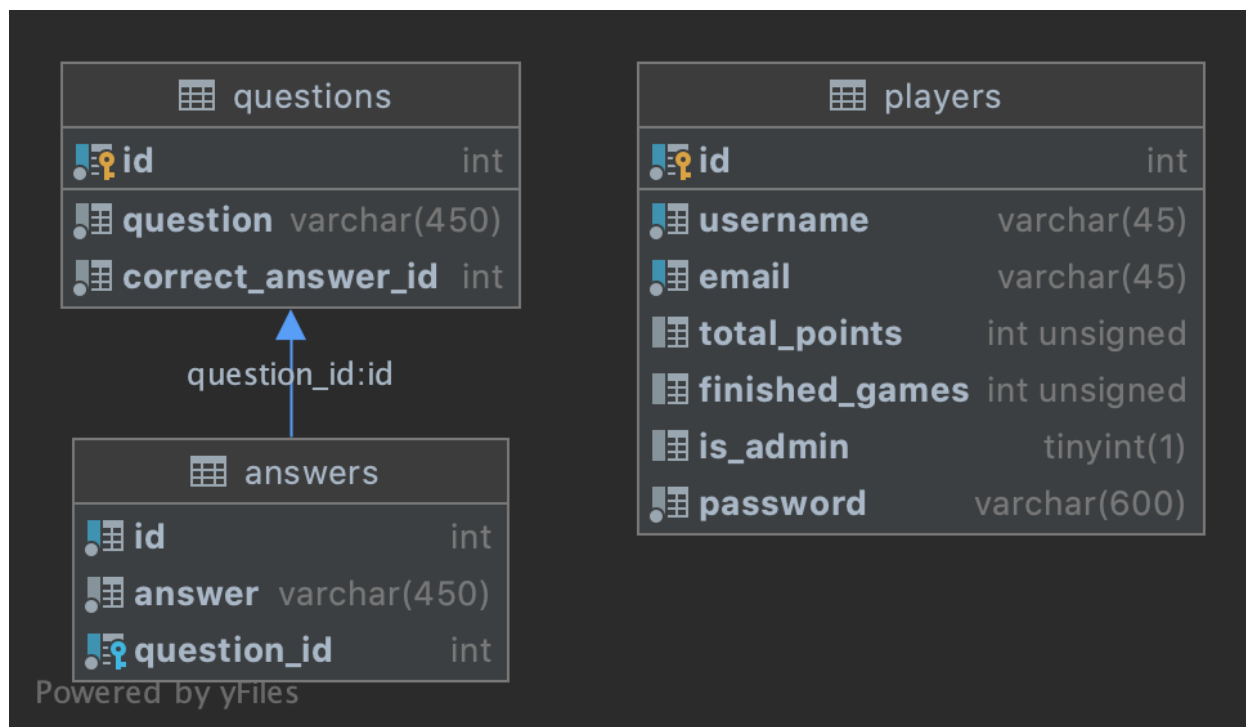
4.3	Stranica za login korisnika .....	9
4.4	Stranica za registraciju korisnika .....	11
4.5	Stranica igre .....	13
5.	Zaključak.....	17
6.	Literatura.....	18

# 1. Predlog teme

JavaFX igra, koja bi učitala podatke sa remote servera, sa opcijama za login za postojeće korisnike, i registraciju za nove. Korisnici mogu da budu regularni, i admin korisnici, ~~gde je razlika ta da admin korisnici mogu da ubacuju, brišu i menjaju pitanja u bazi podataka.~~ Aplikacija takodje može sadržati neke jednostavne animacije.

## 2. Opis funkcionalnosti

Aplikacija je napisana u Javi i koristi JavaFX frejmwork za generisanje korisničkog interfejsa. Aplikacija koristi Maven kao package manager. Baza podataka je hostovana na Digital Ocean serverima, i koristi MySQL. Na slici ispod možete videti dijagram tabela u bazi podataka.



Za komunikaciju sa bazom podataka se koristi Hibernate ORM. Za enkripciju i dekripciju korisničkih šifara se koristi jasypt biblioteka. Testovi su pisani uz pomoć JUnit i Mockito biblioteka.

Osnovna funkcionalnost aplikacije je imulacija kviz igre sa jednostavnim pitanjima koja imaju po tri ponudjena odgovora, od kojih je samo jedan tačan. Korisnik može pokrenuti igru i dok nije autentikovao, ali se rezultati upisuju u bazu podataka samo za autentikovane korisnike.

### 3. Struktura aplikacije

Aplikacija je bazirana na MVC šablonu programiranja za GUI aplikacije i strukturi projekta potrebnj za Maven package manager. Na slici ispod se može videti struktura aplikacije:



Direktorijum app u sebi sadrži glavnu klasu za pokretanje aplikacije, Main, i dva paketa: controllers i views. U controllers paketu se nalaze controller klase pomoću kojih kontrolišemo stanje korisničkog interfejsa, i to su:

- **AuthController** - Ova klasa funkcioniše kao singleton klasa, i njena odgovornost je praćenje auth stanja korisnika, tj. da li je korisnik ulogovan, ili nije ulogovan. U slučaju da se doda admin funkcionalnost, ova klasa bi takodje pratila akcije korisnika, sa ciljem da spreči neautorizovanu modifikaciju podataka. Ova klasa se instancira na samom početku rada aplikacije.

- `GameController` - Ova klasa kreira `Game` glasu i generiše listu pitanja i prati stanje igre.
- `PlayerController` - Ova klasa je zadužena za pripremu `Player` podataka za interakcijom sa bazom podataka. Koristi `PlayerDAOImpl` klasu kao servis za komunikaciju sa bazom podataka.
- `QuestionController` - Ova klasa je zadužena za pripremu `Questions` podataka i zahteva. Koristi `QuestionDAOImpl` klasu kao servis za komunikaciju sa bazom podataka.
- `ScreenController` - Ova klasa je zadužena za kontrolisanje i praćenje navigacije korisnika kroz aplikaciju. Ova klasa funkcioniše kao singleton klasa, i koristi `Stack` strukturu podataka za čuvanje istorije kretanja kroz aplikaciju.

U `views` paketu se nalaze klase koje su odgovorne za prikaz korisničkog interfejsa, i to su:

- `GameView` - Ova klasa je zadužena za prikaz UI igre.
- `HomeView` - Ova klasa je zadužena za prikaz UI početnog ekrana.
- `LoginView` - Ova klasa je zadužena za prikaz UI login ekrana i forme.
- `RegisterView` - Ova klasa je zadužena za prikaz UI registration ekrana i forme.
- `ResultsView` - Ova klasa je zadužena za prikaz UI za stranicu sa tabelom rezultata.

Direktorijum `DAO` sadrži "*Data Access Object*" klase, čija je svrha da obezbede interfejs za komunikaciju sa bazom podataka.

Direktorijum `entities` sadrži Hibernate entitete, klase mapirane prema tabelama u bazi podataka, u ovom slučaju to su `Player` klasa za `players` tabelu, `Answer` klasa za `answers` tabelu, i `Question` klasa za `questions` tabelu.

Direktorijum `enums`, sadrži enumeracije koje se koriste u aplikaciji, u ovom slučaju imamo samo `Views` enum, koji označava dostupne stranice u aplikaciji.

Direktorijum `exceptions` sadrži custom izuzetke, u ovom slučaju samo `AuthException`, koji se može desiti prilikom login ili registration procesa.

Direktorijum `interfaces`, sadrži interfejse koji se koriste u aplikaciji. Ovde imamo tri interfejsa:

- `PlayerDAO` - interfejs za `Player DAO` klasu
- `QuestionDAO` - interfejs za `Question DAO` klasu
- `BaseView` - interfejs za UI prikaze aplikacije

Direktorijum `models` sadrži ostale klase koje se koriste u projektu. Ovde imamo dve klase:

- `AnswerButton` - jednostavni wrapper oko `JavaFX Button` klase, koja dodaje `answerId` polje

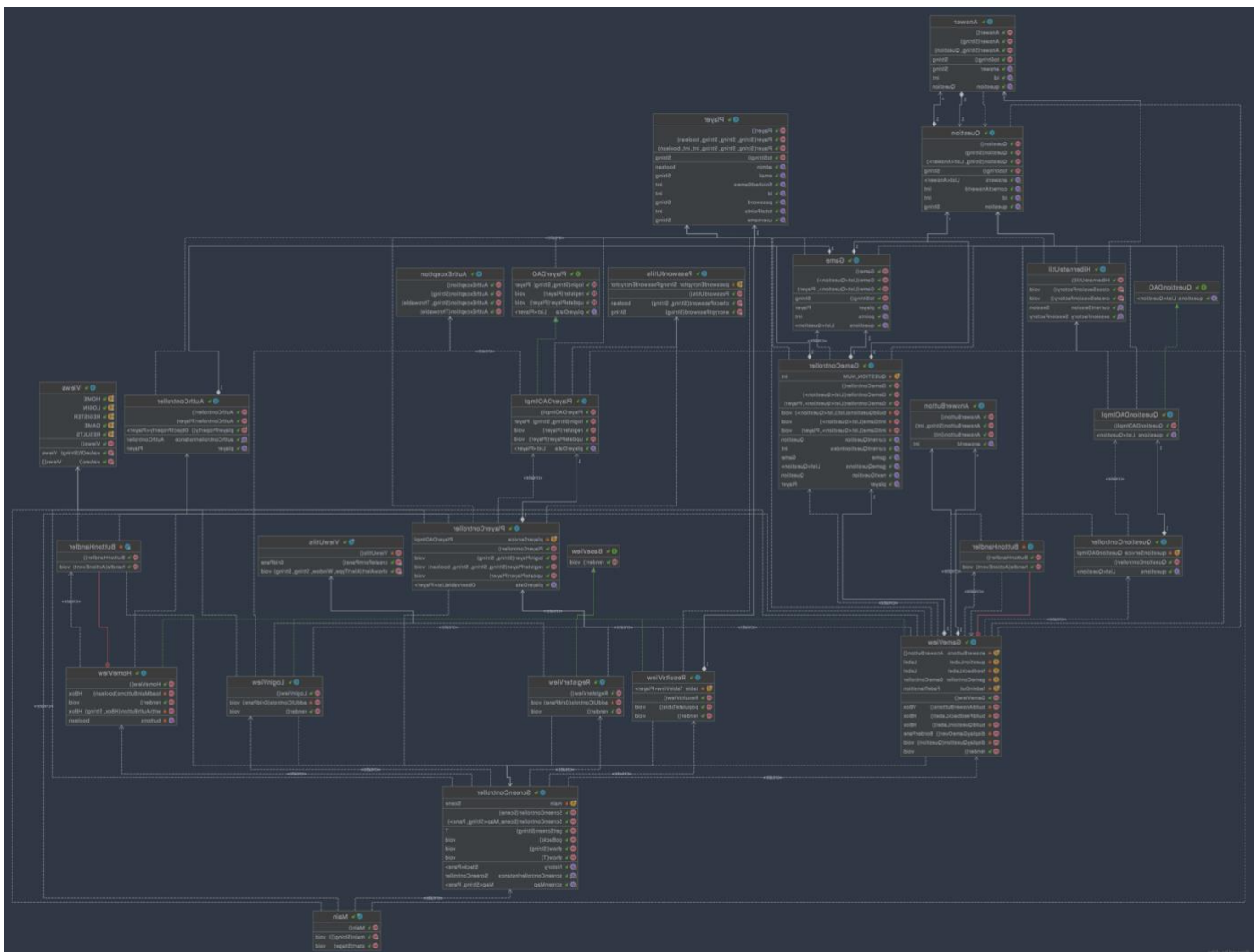
Diretorijum util sadrži statičke util klase:

- **HibernateUtils** - statička klasa sa metodama za rad sa Hibernate-om, kreiranje sessionFactory i *short-term* sesija za rad sa bazom. Ova klasa se instancira na početku rada aplikacije, i prekida konekciju pri zatvaranju aplikacije.
- **PasswordUtils** - statička klasa sa metodama za enkripciju i dekripciju korisničkih šifara
- **ViewUtils** - statička klasa sa nekoliko metoda koje se koriste u views paketu klasa

Direktorijum resources sadrži resurse aplikacije: logo, Hibernate config i CSS stilove.

Direktorijum test sadrži unit testove pisane uz pomoć JUnit biblioteke.

Na slici ispod možete videti UML dijagram klasa aplikacije:



Detaljniji dijagram se može videti u sklopu projekta.

## 4. Korisničko uputstvo

Korisnički interfejs aplikacije se sastoji iz pet delova (ekrana): početna stranica, stranica sa prikazom rezultata, stranica za login korisnika, stranica za registraciju korisnika i stranica igre.

### 4.1 Početna stranica

Početna stranica predstavlja glavnu stranicu aplikacije, odakle korisnik može da bira na koju će od ostalih ponudjenih stranica da ide.



Logo igre se učitava iz lokalnog fajl sistema. Dugmad "Quick Play" i "Sign in" se prikazuju samo u slučaju da korisnik nije autentikovao, u suprotnom prikazujemo "Play" i "Sign out" dugmad.



## 4.2 Stranica sa prikazom rezultata

Na ovoj stranici, autentikovani i neautentikovani korisnici mogu da vidi tabelu sa rezultatima svih korisnika koji su koristili aplikaciju i čiji su rezultati sačuvani u bazi podataka. Rezultati su sortirani u opadajućem redosledu po "Total Points" koloni.



Java kviz

Go Back

ID	Username	Email	Total Points	Finished Games
1	darkot	test@test.com	135	8
4	TestAcc3	test3@test.com	60	4
7	TestAcc6	test6@test.com	55	6
6	TestAcc5	test5@test.com	50	3
3	TestAcc2	test2@test.com	40	3
2	TestAcc	test1@test.com	20	2
8	TestAcc7	test7@test.com	20	3
10	TestAcc9	test9@test.com	15	2
11	TestAcc10	test10@test.com	15	1
9	TestAcc8	test8@test.com	10	2
5	TestAcc4	test4@test.com	0	0

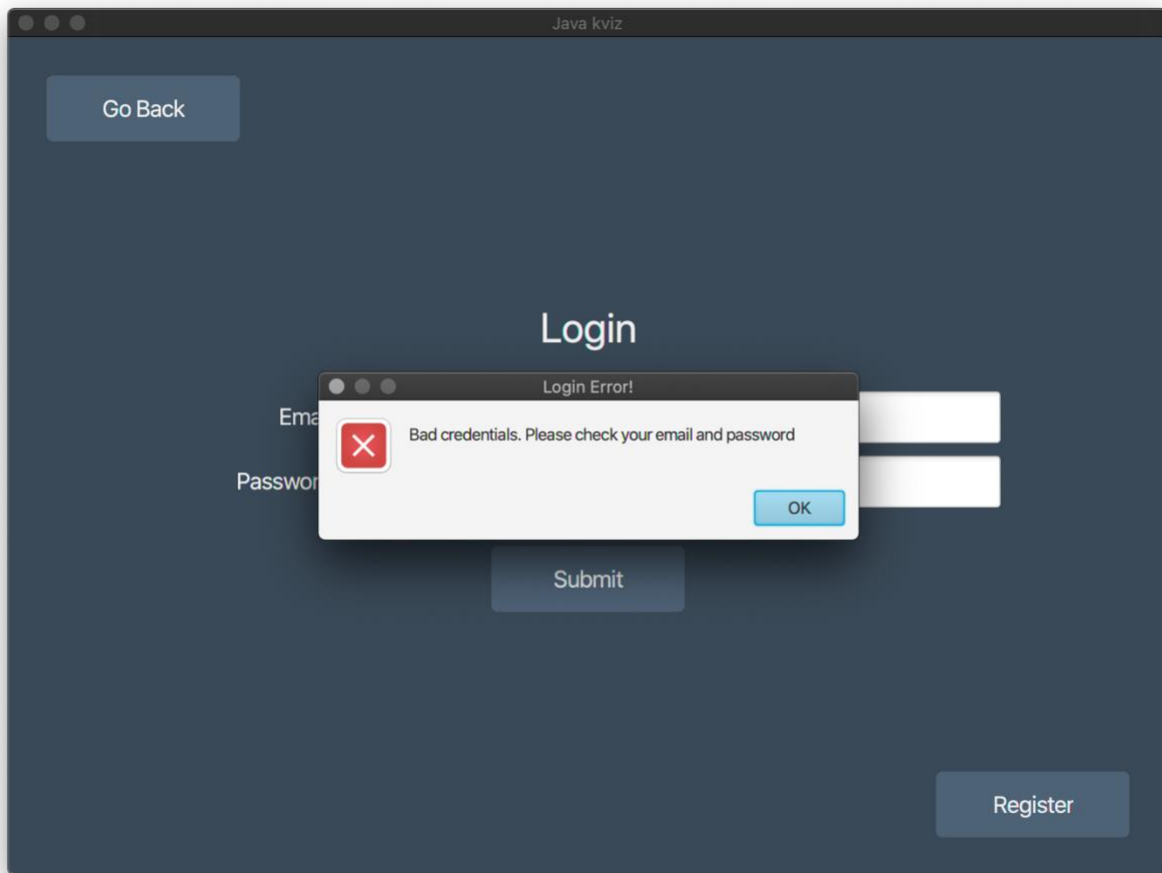
## 4.3 Stranica za login korisnika

Na ovoj stranici se nalazi forma preko koje korisnik može da se uloguje koristeći svoj email i šifru:

The image shows a window titled "Java kviz" with a dark blue background. In the top-left corner, there is a "Go Back" button. In the center, the word "Login" is displayed. Below it are two white input fields: the first is labeled "Email:" and the second is labeled "Password:". Below the password field is a "Submit" button. In the bottom-right corner, there is a "Register" button.

U slučaju da korisnik nema nalog, klikom na dugme u donjem desnom uglu može da ode na registration stranicu, a može se vratiti na početni ekran klikom na Go Back dugme.

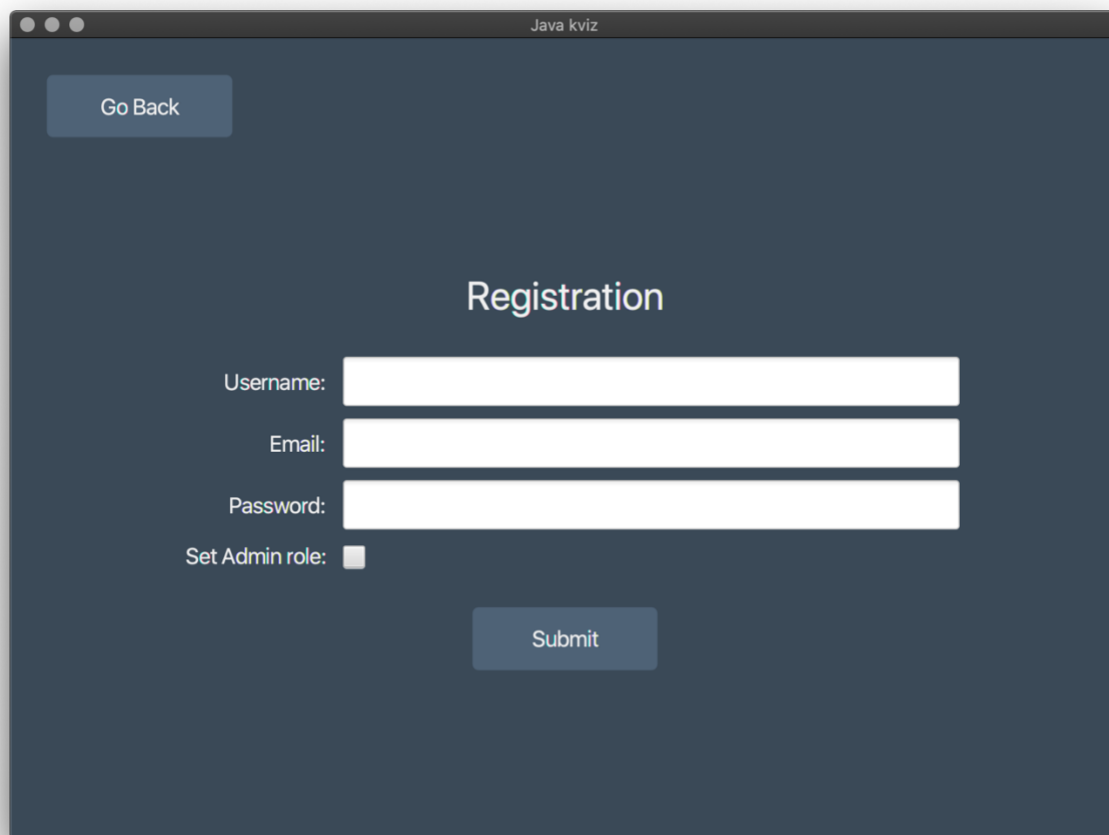
Ukoliko korisnik pokuša da se uloguje sa pogrešnom ili nepostojećom šifrom/email kombinacijom, prikazuje se Alert komponenta sa porukom o grešci:



Nakon uspešne prijave, korisnik je redirektovan na početnu stranicu aplikacije.

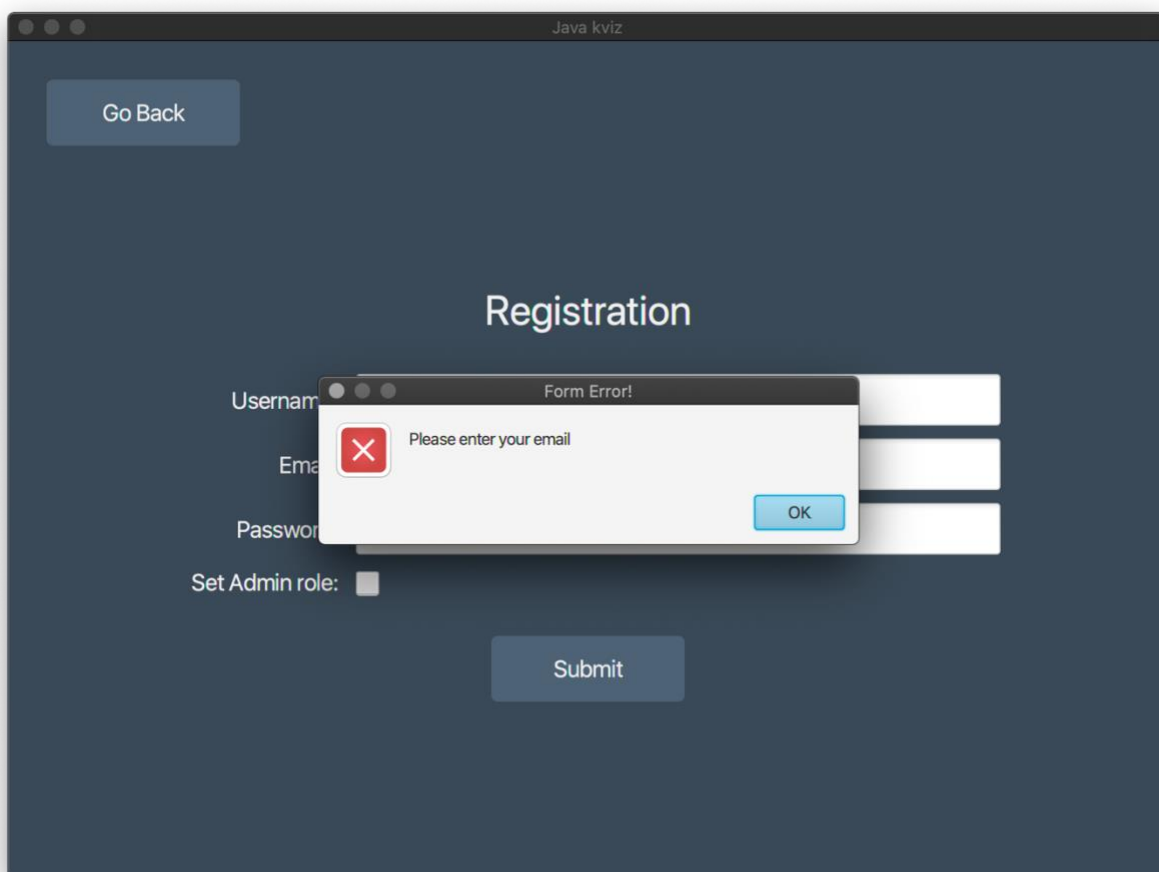
## 4.4 Stranica za registraciju korisnika

Na ovoj stranici, korisnik može registrovati novi nalog popunjavanjem svih polja u formi:



A screenshot of a web application window titled "Java kviz". The window has a dark blue background. In the top left corner, there is a "Go Back" button. In the center, the word "Registration" is displayed in a large, white, sans-serif font. Below this, there are three white input fields stacked vertically. The first field is labeled "Username:", the second "Email:", and the third "Password:". Below the input fields, there is a label "Set Admin role:" followed by a small, empty square checkbox. At the bottom center, there is a "Submit" button.

U slučaju da korisnik pokuša da pošalje formu bez nekog polja popunjenog, dobija Alert sa porukom o grešci:

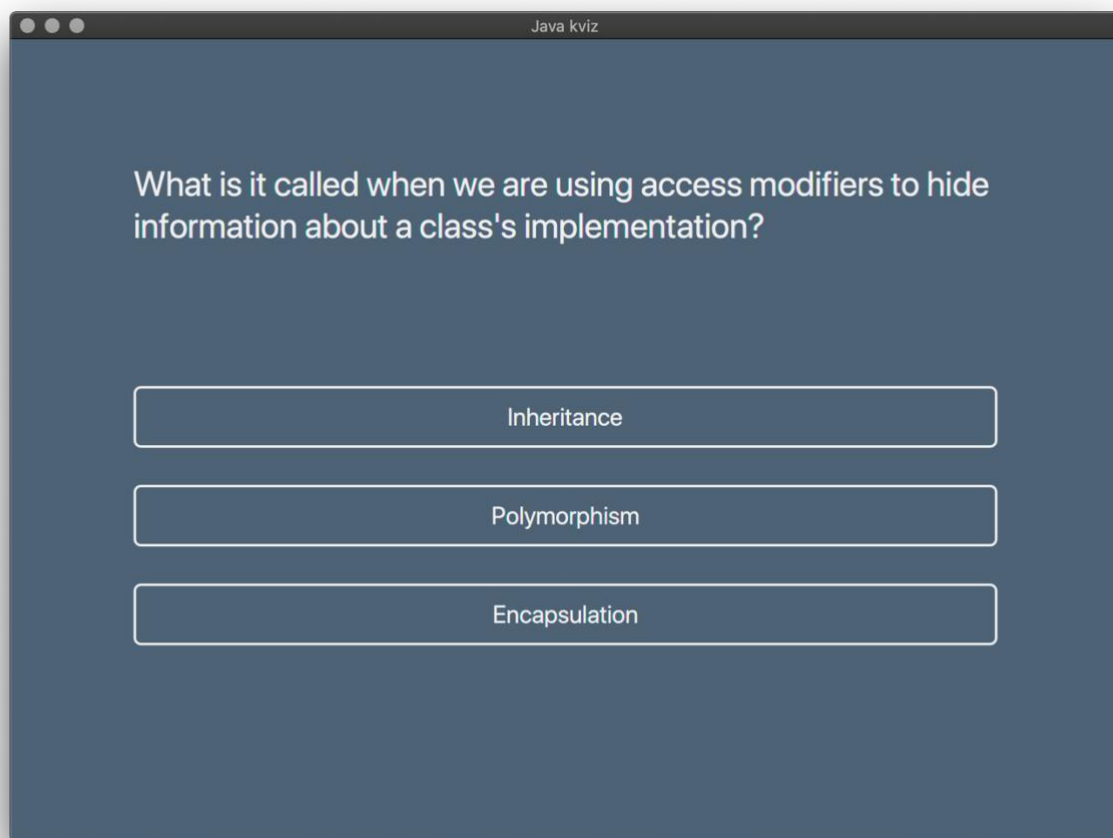


A screenshot of the same "Registration" form as above, but with an alert dialog box open in the foreground. The alert dialog has a title bar that says "Form Error!". It features a red square icon with a white "X" on the left. To the right of the icon, the text "Please enter your email" is displayed. In the bottom right corner of the dialog, there is a blue button labeled "OK". The background form is partially obscured by the dialog box.

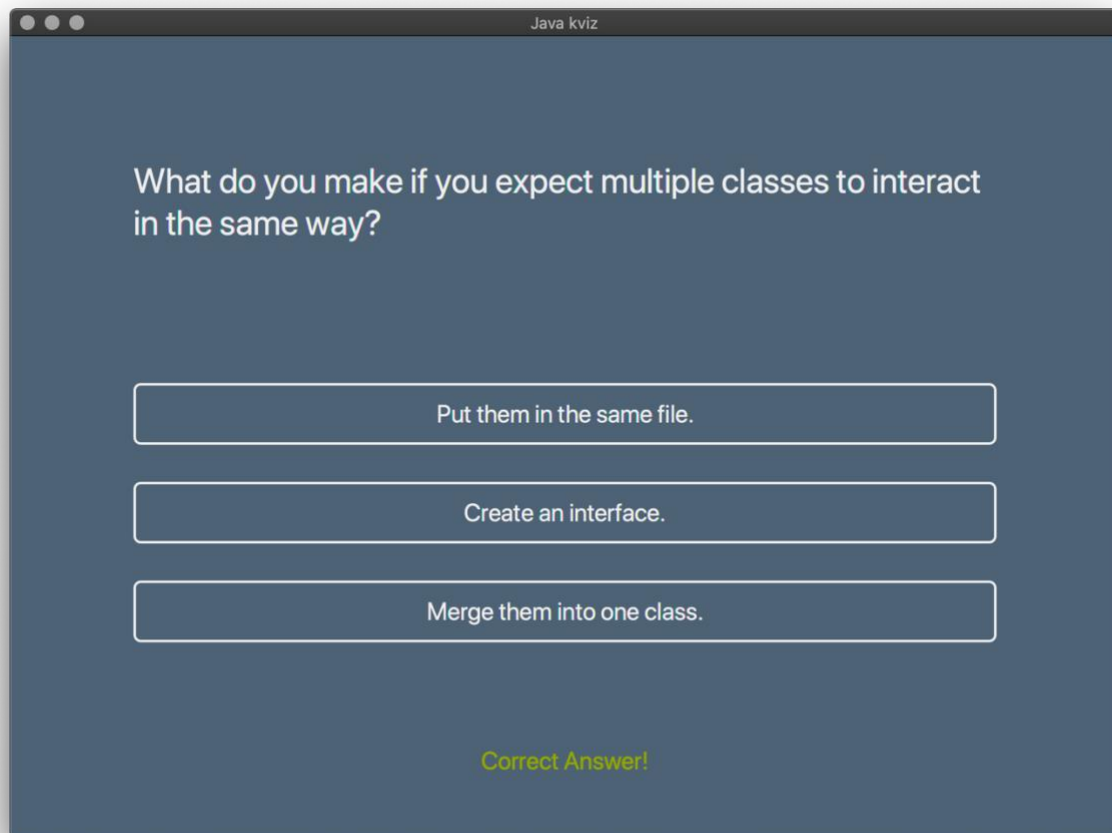
Nakon uspešne registracije, korisnik je redirektovan na početnu stranicu aplikacije.

## 4.5 Stranica igre

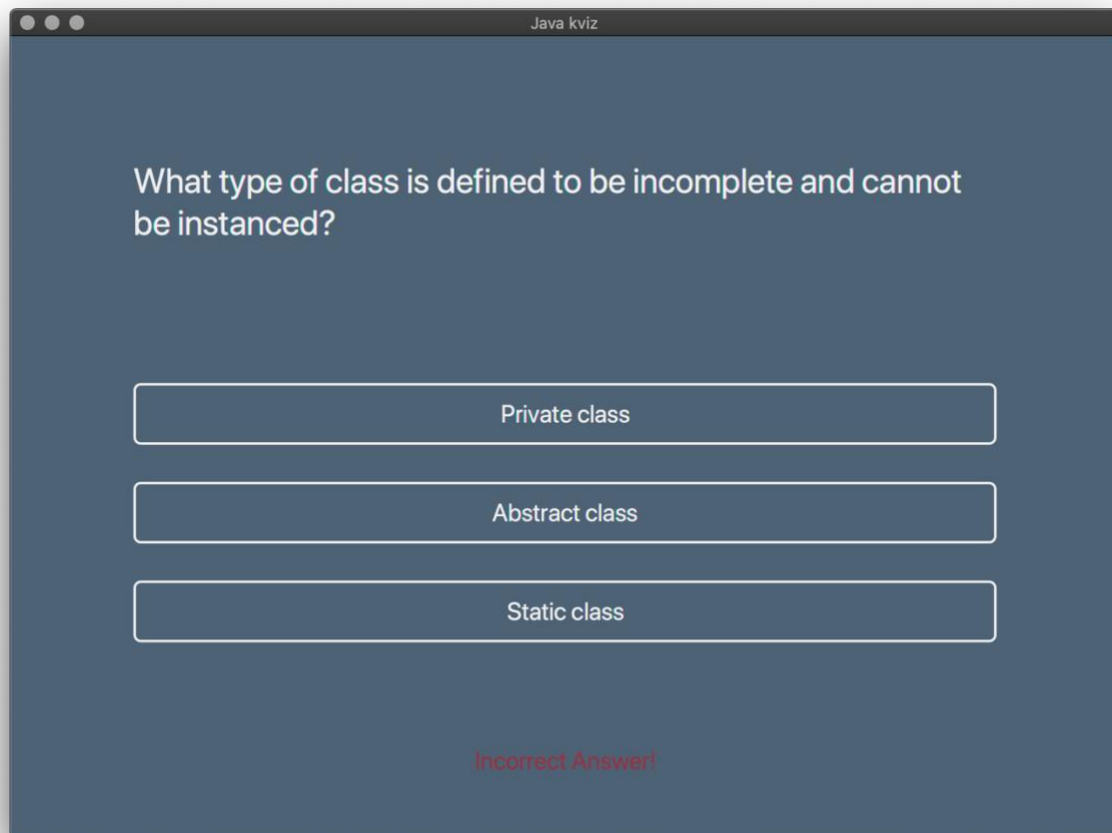
Na ovoj stranici korisnik aplikacije ulazi u proces igre, gde se nakon svakog odgovora prikazuje novo pitanje sa povratnom porukom da li je dati odgovor na pitanje bio tačan ili ne:



U slučaju da je odgovor na pitanje bio tačan, na dnu ekrana se pojavljuje poruka, koja nestaje nakon jedne sekunde uz pomoć FadeTransition animacije:

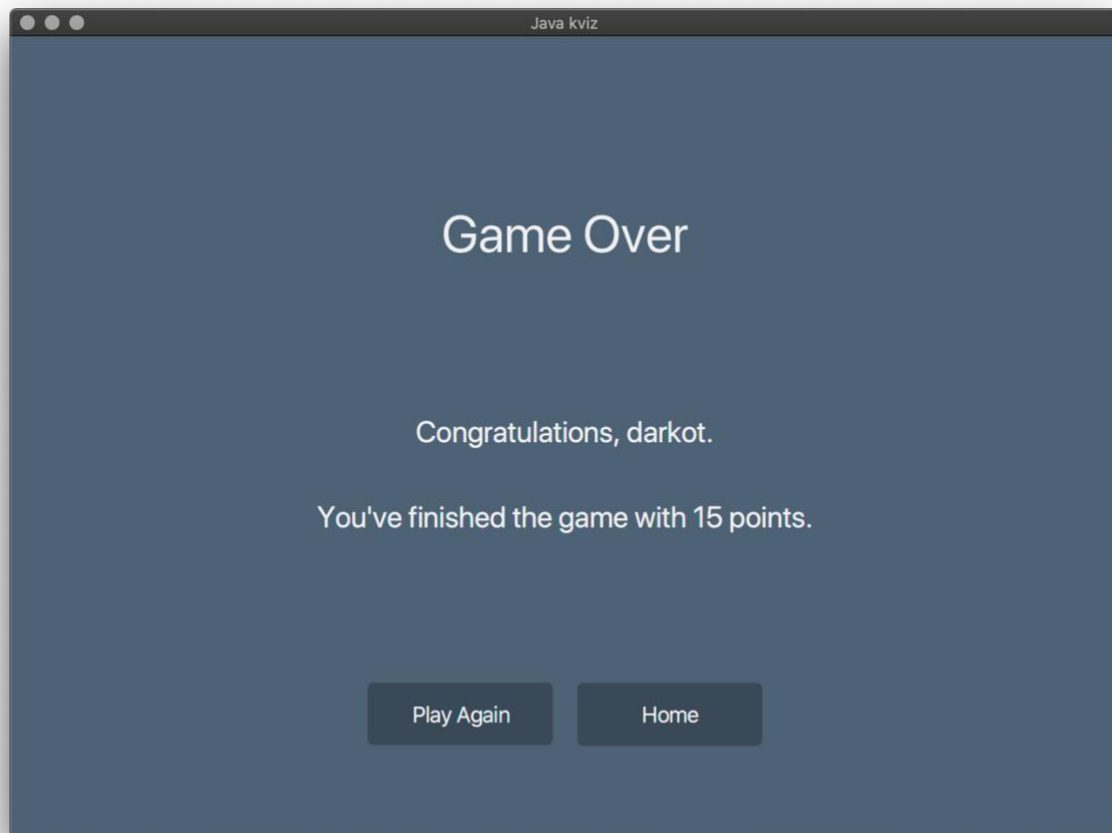


U slučaju da odgovor nije bio tačan, tada se pojavljuje malo drugačija poruka:



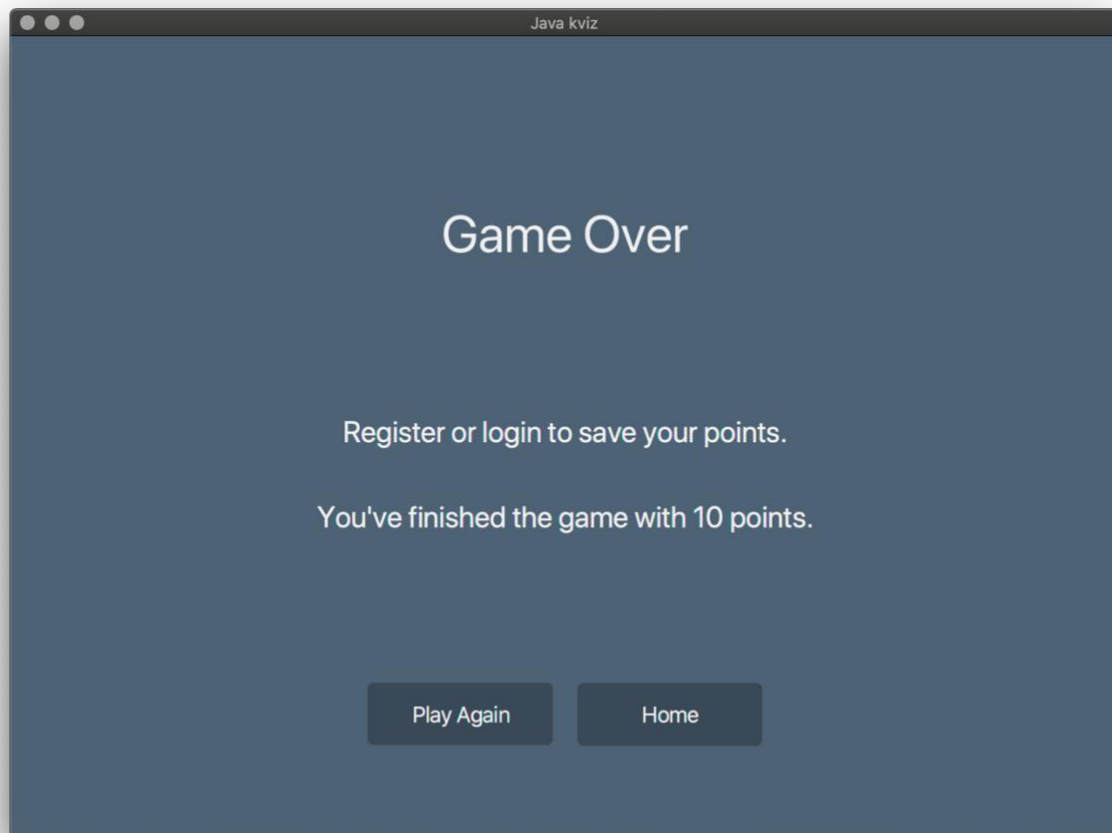
Svaka instanca igre sadrži 5 pitanja sa maksimalno 25 poena koji se mogu osvojiti.

Ukoliko je korisnik ulogovan, na završnom ekranu nakon igre, dobiće poruku o broju osvojenih poena i opciju da pokrene igru ponovo ili da ode na početni ekran aplikacije:



U slučaju da je neulogovani korisnik završio igru, nudi im se ista opcija, ali se prikazuje i poruka da je potrebno ulogovati se ukoliko žele da sačuvaju osvojene poene:





## 5. Zaključak

Ovaj projekat obuhvata nekoliko bitnih delova Java razvoja, a to su rad sa korisničkim interfejsom uz pomoć JavaFX, rad sa bazom podataka, Hibernate ORM i korišćenje Maven package menadžera. Kao aplikacija sa ciljem da pokaže naučeno iz CS102 predmeta, ona pokriva većinu naučenih stvari. Jedina stvar koja nedostaje je rad sa više niti, ali u ovom slučaju, nije bilo potrebe za pisanjem multi-thread koda, sa obzirom da Hibernate u pozadini vodi računa kako se izvršavaju operacije vezane za bazu podataka.

Meni lično, ova aplikacija je bila prva kompleksnija aplikacija pisana u Java programskom jeziku, sa JavaFX frejmworkom za razvoj grafičkog interfejsa, i Hibernate ORM-om za rad sa bazom podataka. Pored dosta stvari koje sam naučio tokom rada na njoj, takodje je ostalo dosta stvari koje nisam imao prilike da upotrebim u kodu, i za koje se nadam da ću moći upotrebiti u nekom od narednih projekata.

Neke stvari koje se mogu unaprediti u aplikaciji:

- CRUD operacije za pitanja i odgovore, koje mogu da izvršavaju admin korisnici
- Više animacija
- Veći izbor pitanja i odgovora

- PostgreSQL umesto MySQL-a
- Criteria API umesto HQL-a

## 6. Literatura

- Oracle, JavaFX CSS Reference Guide, [openjfx.io](https://openjfx.io), 2008
- Vladimir Milićević, CS102 - Objekti i apstrakcija podataka, Univerzitet Metropolitan, 2020
- icons8, quiz icons, [icons8.com](https://icons8.com)
- Red Hat, Mapping Entities, [docs.jboss.org](https://docs.jboss.org), 2014
- Steven Haines, Java persistence with JPA and Hibernate, Part 1: Entities and relationships, [infoworld.com](https://infoworld.com), 2019
- Thorben Janssen, Hibernate Tips: How to map an entity to multiple tables, [thorben-janssen.com](https://thorben-janssen.com), 2018
- StackOverflow, Inserting Data in Multiple Tables in Hibernate, [stackoverflow.com](https://stackoverflow.com), 2017
- Andy Till, JUnit Rule for JavaFX Controller Testing, [andrewtill.blogspot.com](https://andrewtill.blogspot.com), 2012
- Java CodeHouse, Mockito Tutorial (A comprehensive guide with examples), [javacodehouse.com](https://javacodehouse.com), 2017
- Ed Eden-Rump, How to style a TableView in JavaFX, [edencoding.com](https://edencoding.com), 2020
- baker6romeo, Singleton Class in Java, [geeksforgeeks.org](https://geeksforgeeks.org), 2020
- Sai Dandem, Gradients In JavaFX, [blog.e-zest.com](https://blog.e-zest.com), 2012
- Vlad Mihalcea, The best way to map a @OneToMany relationship with JPA and Hibernate, [vladmihalcea.com](https://vladmihalcea.com), 2021
- Eugen (Baeldung), @JoinColumn Annotation Explained, [baeldung.com](https://baeldung.com), 2020
- Eugen (Baeldung), Hibernate One to Many Annotation Tutorial, [baeldung.com](https://baeldung.com), 2020
- mkyong, Hibernate – Many-to-Many example – join table + extra column (Annotation), [mkyong.com](https://mkyong.com), 2011
- Arvind Rai, Hibernate Bidirectional Mapping Example with @JoinTable Annotation, [concretepage.com](https://concretepage.com), 2015