|  |
| --- |
| Central Queensland University, Sydney |
| Mapping of design to implementation  Rotary Youth Drive Awareness Project |
| COIT20273: Software Design and Development Project |
|  |
| Lecturer/Tutor: Zakiullah Khan Course Coordinator: Dr. Lily Li |
| **Due date: 9/22/2017** |

Project Members:

Bijay Shahi (12021856)

Keshav Khadka (S0280624)

Pratik Shrestha (S0282733)

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **Document Name** | **Version** | **Revision Date** |
| 1. | Mapping of design to implementation | 1.0 | 2017-09-22 |

Table of Contents

[1. Software Architecture 3](#_Toc493843671)

[2. Layer Modelling 4](#_Toc493843672)

[3. Database Design 5](#_Toc493843673)

[4. Class modelling 6](#_Toc493843674)

[4.1. Class diagram of JPA Entities 20](#_Toc493843675)

[4.2. Class diagram of EJBs 21](#_Toc493843676)

[4.3. Class diagram of backing beans 22](#_Toc493843677)

[5. User Interface Design 23](#_Toc493843678)

[5.1. Quiz Main page’s 23](#_Toc493843679)

[5.2. Admin Interface 26](#_Toc493843680)

[6. Behaviour Modelling 30](#_Toc493843681)

[6.1. SD1 – Login 30](#_Toc493843682)

[6.2. SD2 – Create Quiz 30](#_Toc493843683)

[6.3. SD3 – Edit Quiz 30](#_Toc493843684)

[6.4. SD4 – Delete Quiz 30](#_Toc493843685)

[6.5. SD5 – List Quiz 30](#_Toc493843686)

[6.6. SD6 – View Quiz 30](#_Toc493843687)

[6.7. SD7 – Create Question 30](#_Toc493843688)

[6.8. SD8 – Edit Question 30](#_Toc493843689)

[6.9. SD 9 – Delete Question 30](#_Toc493843690)

[6.10. SD10 – List Question 30](#_Toc493843691)

[6.11. SD11 – View Question 30](#_Toc493843692)

[6.12. SD12 – Logout 30](#_Toc493843693)

[6.13. SD13 – Attempt Quiz 30](#_Toc493843694)

[7. Requirement Mapping 31](#_Toc493843695)

# 1. Software Architecture

The software architecture for Rotary Youth Driving Awareness application is not. The only change is in the database where Administrator table has been removed. Below is the software architecture:



**Figure 1. Software architecture of RYDA application**

# 2. Layer Modelling

Model, View, Controller (MVC) pattern has been implemented as designed.



**Figure 2. MVC design pattern for the RYDA application**

# 3. Database Design

The database design for the RYDA application is modelled in the figure 3 below. On the proposed document, Users table had a disjoint Administrator and Student. During the implementation, Administrator table was found not necessary as User can be treated as Administrator and Student can be different entity.



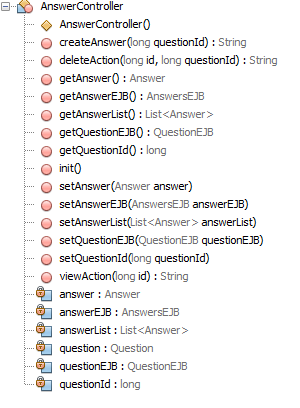
**Figure 3. Database design of RYDA application**

# 4. Class modelling

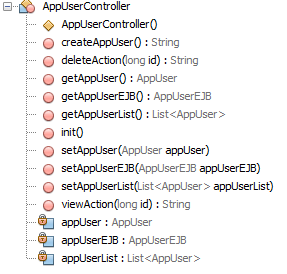
There are several changes that has been made in the class to meet RYDA application’s requirement. Below are the list of changes:

* AppUserController, AttemptQuizController, DashboardController, LoginController, SessionBean has been added.
* AppUserEJB and LoginEJB has been added
* AppUser entity has been added

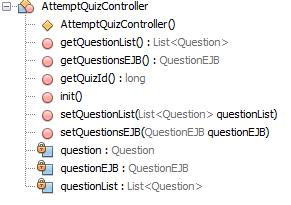
Below is the snapshot of the implemented classes:



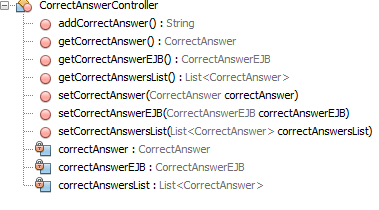
**Figure 4. AnswerController Class**



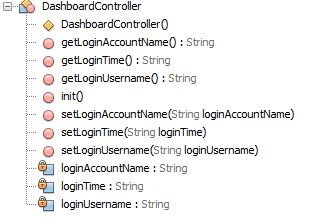
**Figure 5. AppUserController Class**



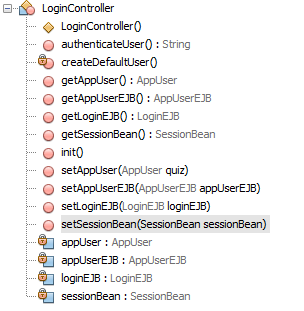
**Figure 6. AttemptQuizController Class**



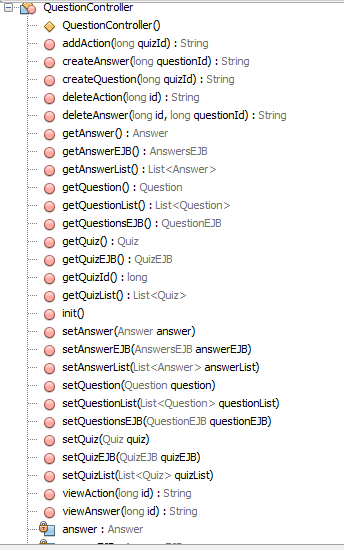
**Figure 7. CorrectAnswerController Class**



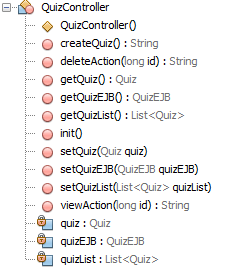
**Figure 8. DashboardController Class**



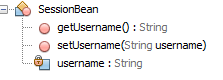
**Figure 9. LoginController Class**



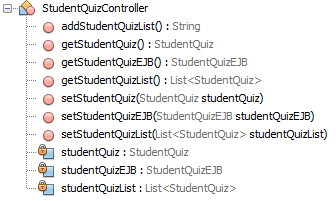
**Figure 10. QuestionController Class**



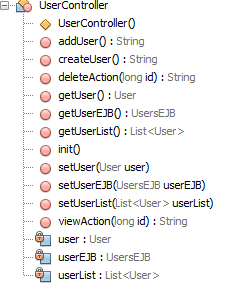
**Figure 11. QuizController Class**



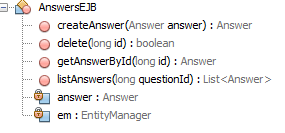
**Figure 12. SessionBean**



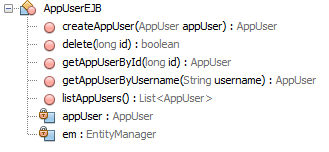
**Figure 13. StudentQuizController Class**



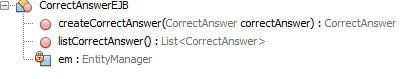
**Figure 14. UserController Class**



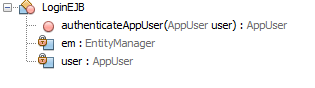
**Figure 15. AnswerEJB Class**



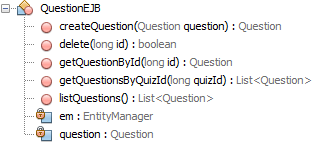
**Figure 16. AppUserEJB Class**



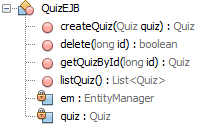
**Figure 17. CorrectAnswerEJB Class**



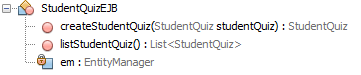
**Figure 18. LoginEJB Class**



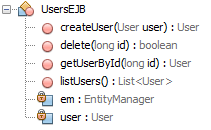
**Figure 19. QuestionEJB Class**



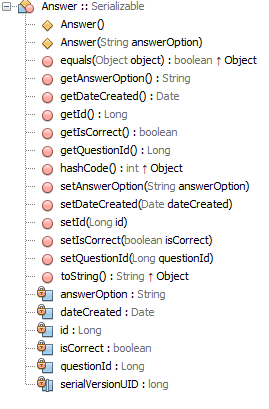
**Figure 20. QuizEJB Class**



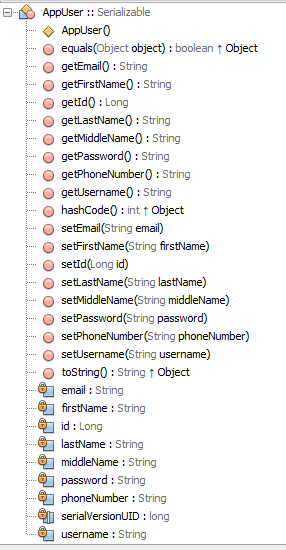
**Figure 21. StudentQuizEJB Class**



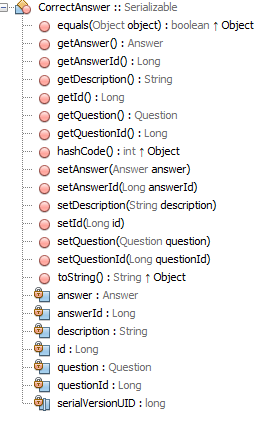
**Figure 22. UserEJB Class**



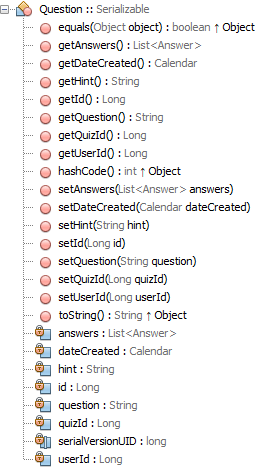
**Figure 22. Answer Class**



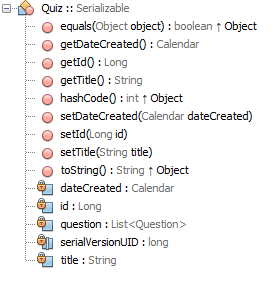
**Figure 23. AppUser Class**



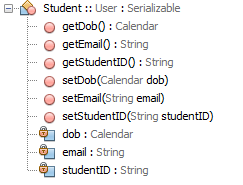
**Figure 23. CorrectAnswer Class**



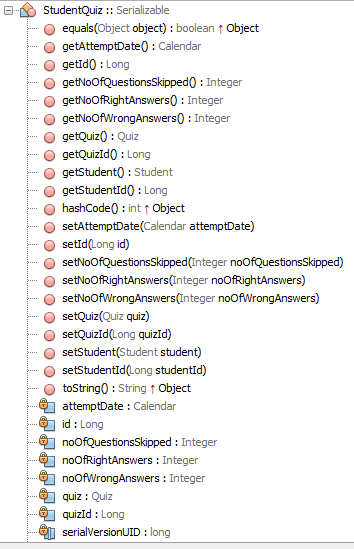
**Figure 24. Question Class**



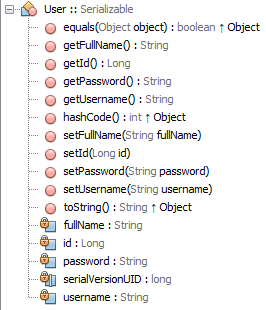
**Figure 25. Quiz Class**



**Figure 26. Student Class**

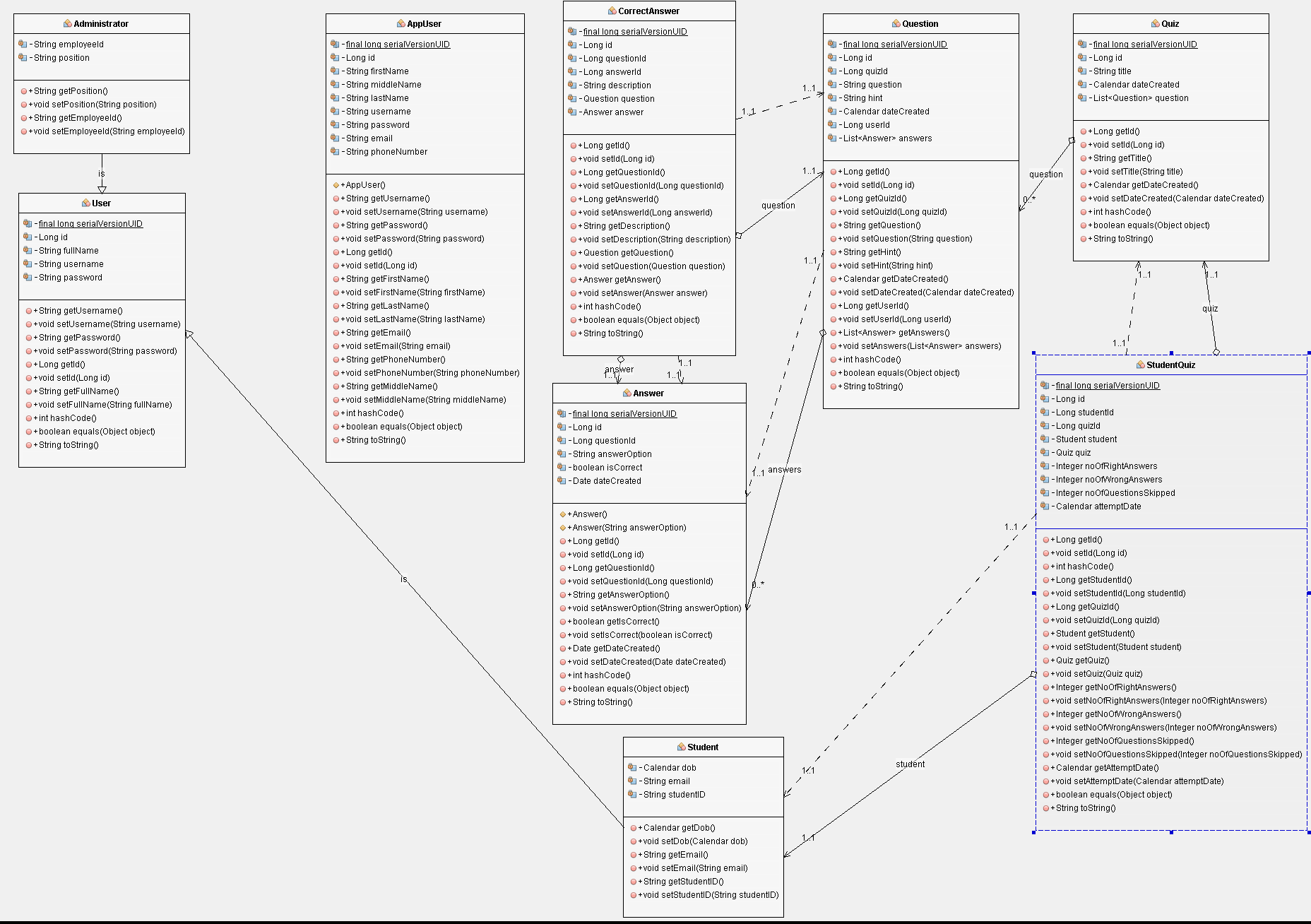


**Figure 27. StudentQuiz Class**



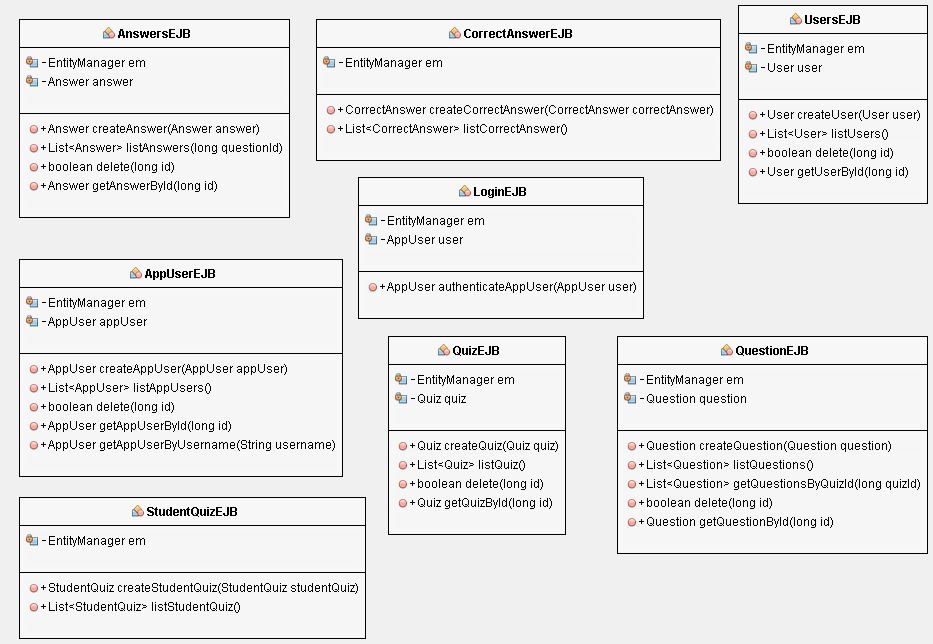
**Figure 28. User Class**

## 4.1. Class diagram of JPA Entities



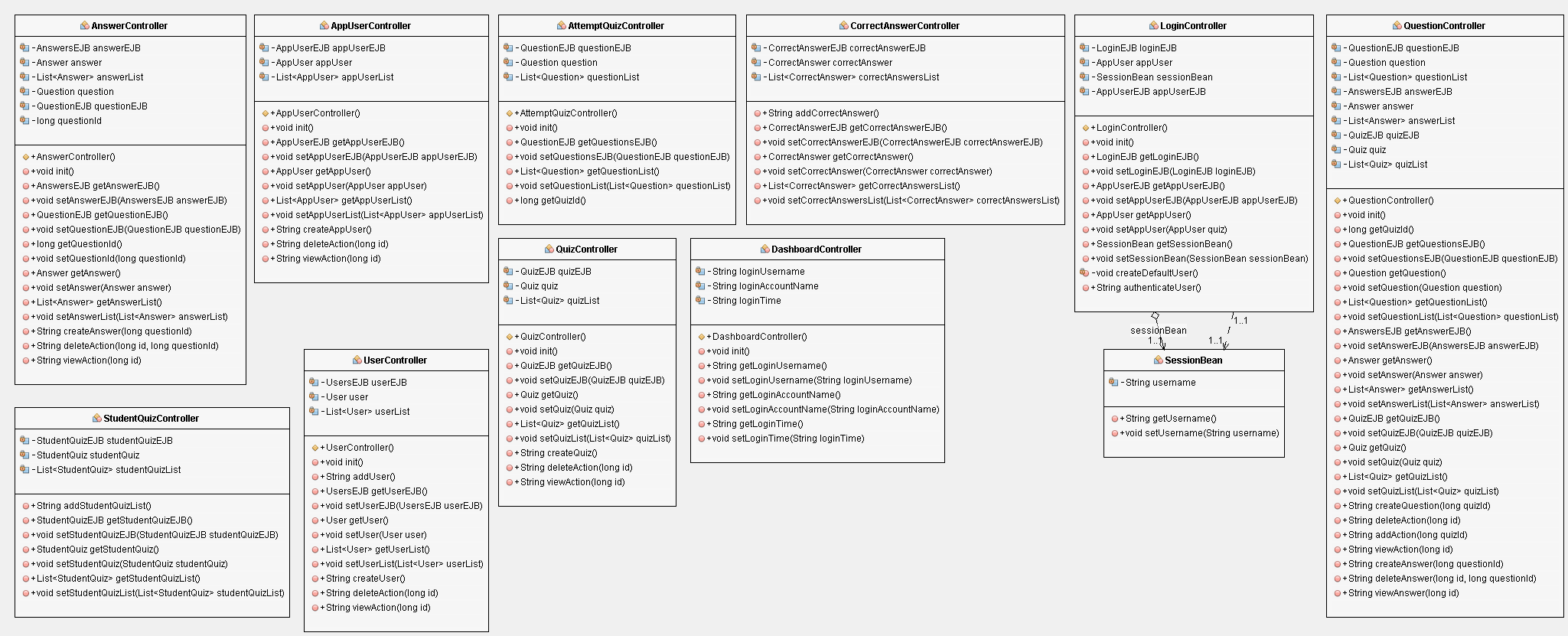
**Figure 29. Class diagram of JPA Entities**

## 4.2. Class diagram of EJBs



**Figure 30. Class diagram of EJBs**

## 4.3. Class diagram of backing beans



**Figure 31. Class diagram of backing beans**

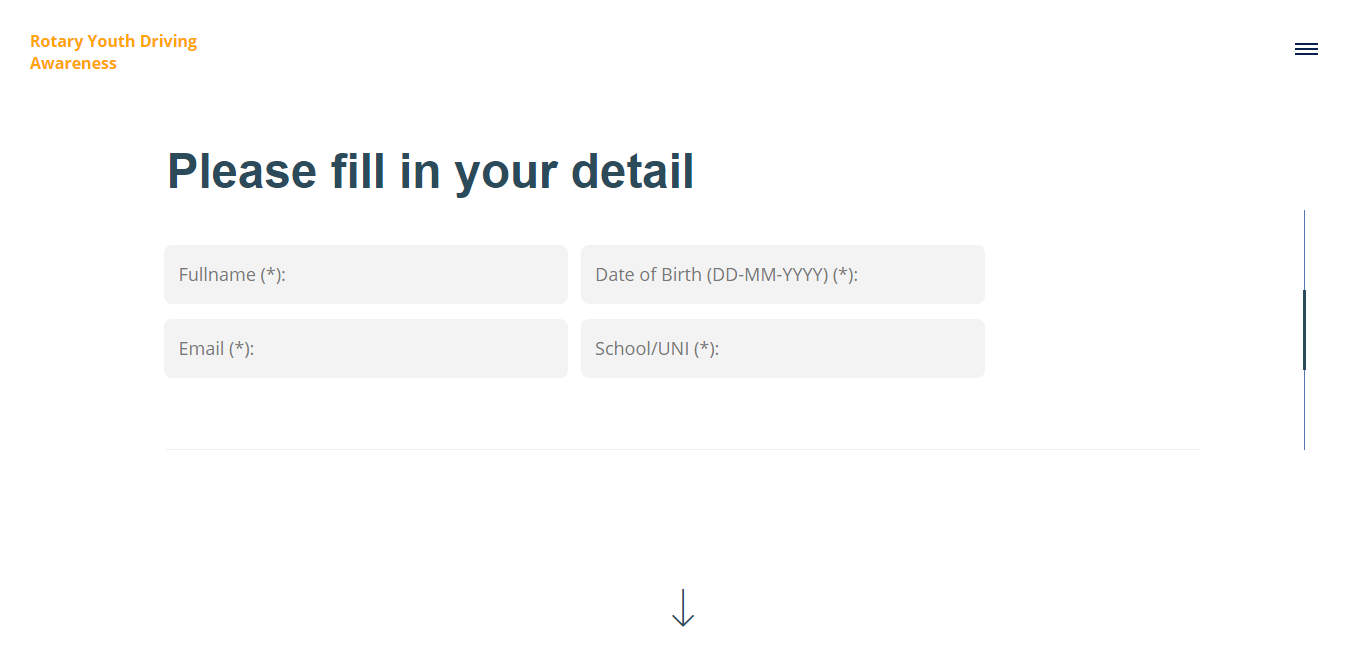
# 5. User Interface Design

Mock up views have been built to reflect the user interface design of the new Rotary Youth Driving Awareness Program.

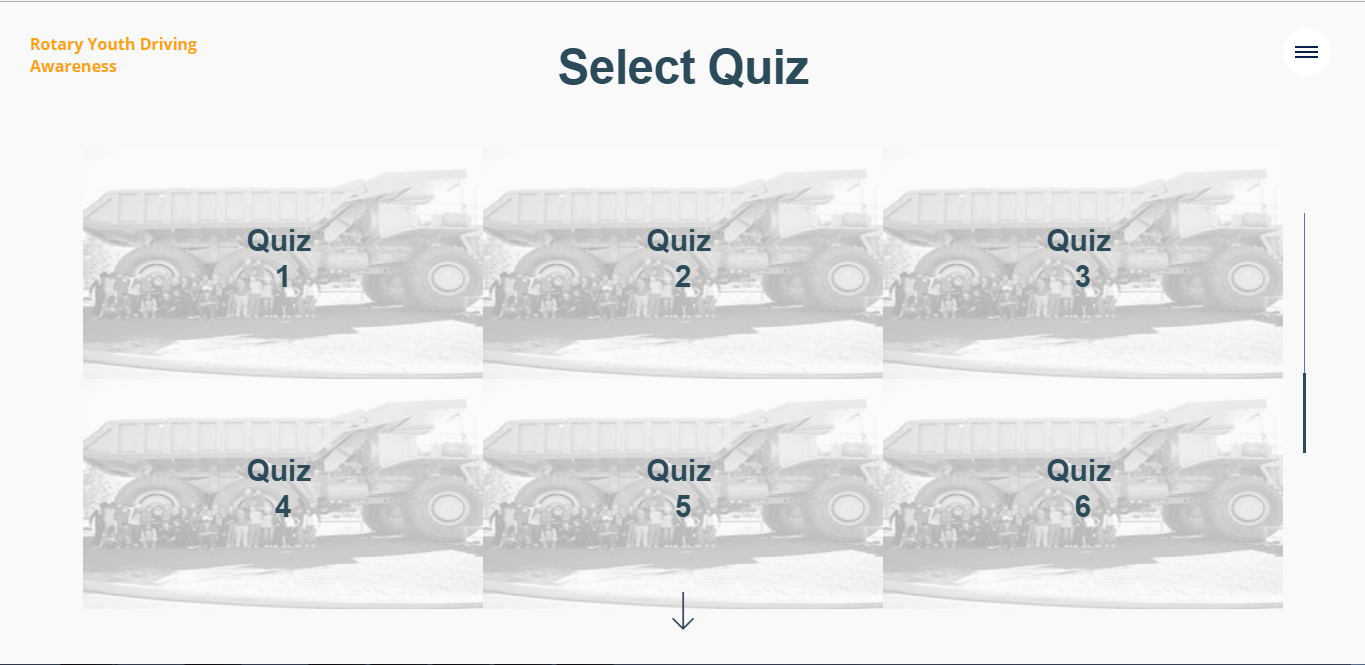
## 5.1. Quiz Main page’s



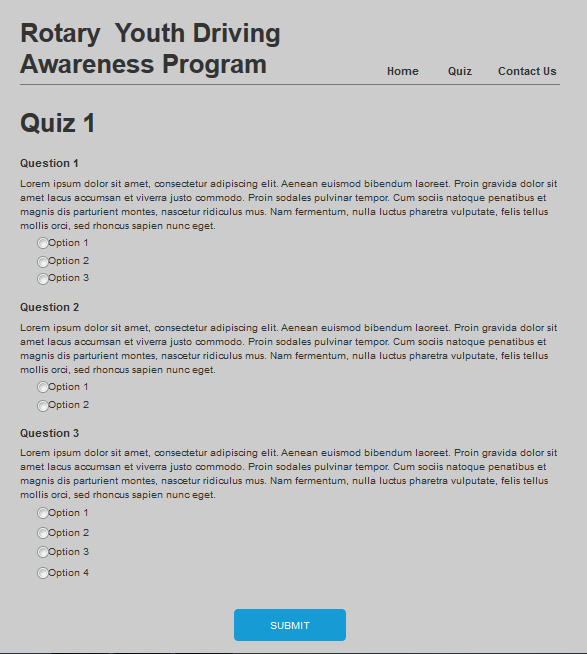
**Fig 32. Home Page**



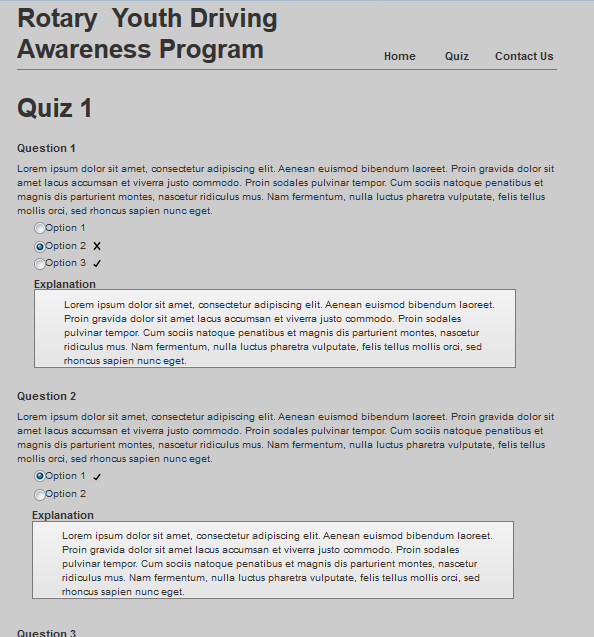
**Fig 33. Personal detail form**

****

**Fig 34. Quiz List**

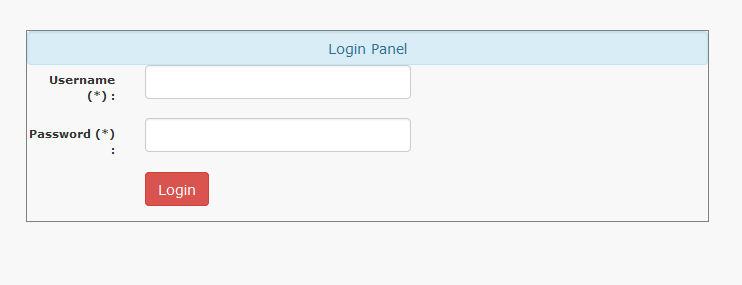


**Fig 35. Quiz**

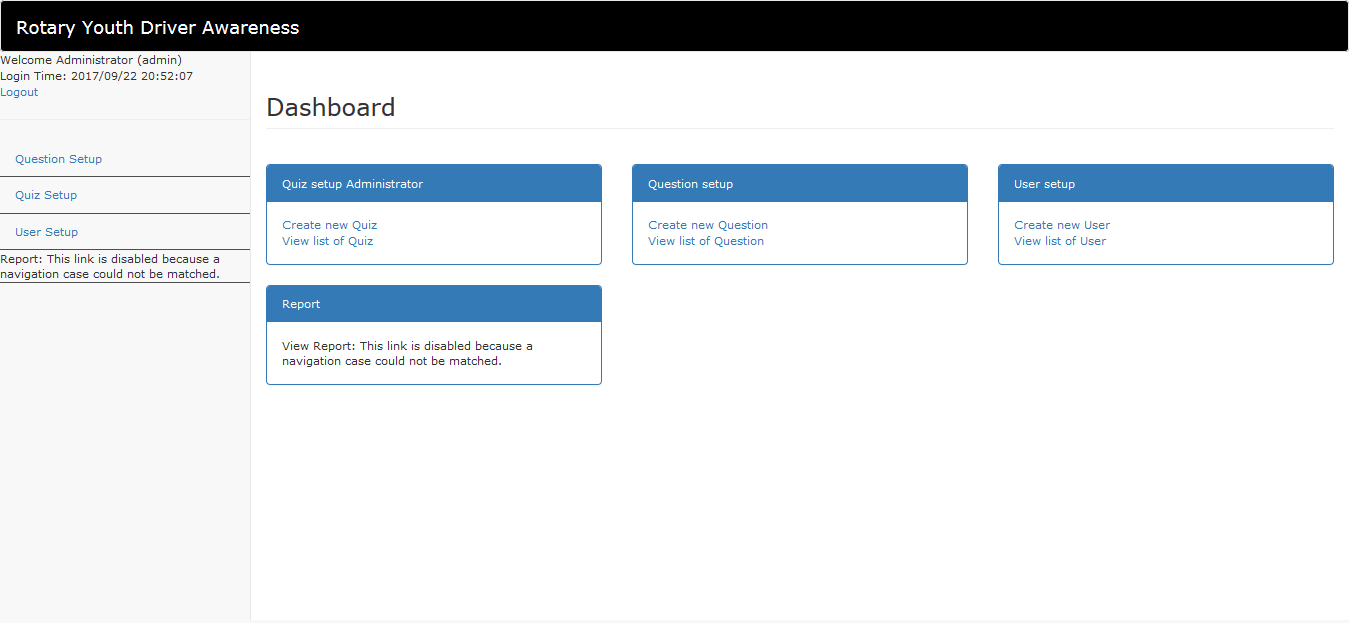


**Fig. 36. Quiz Answers**

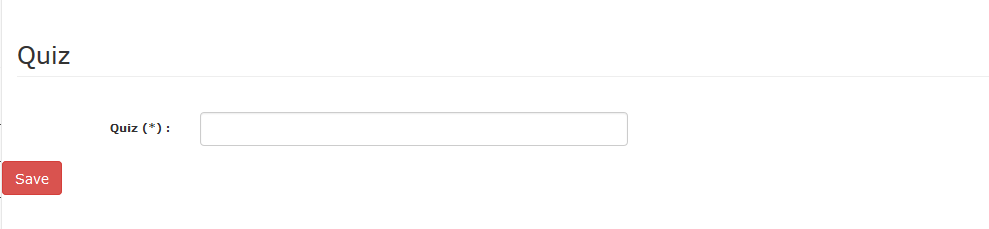
## 5.2. Admin Interface



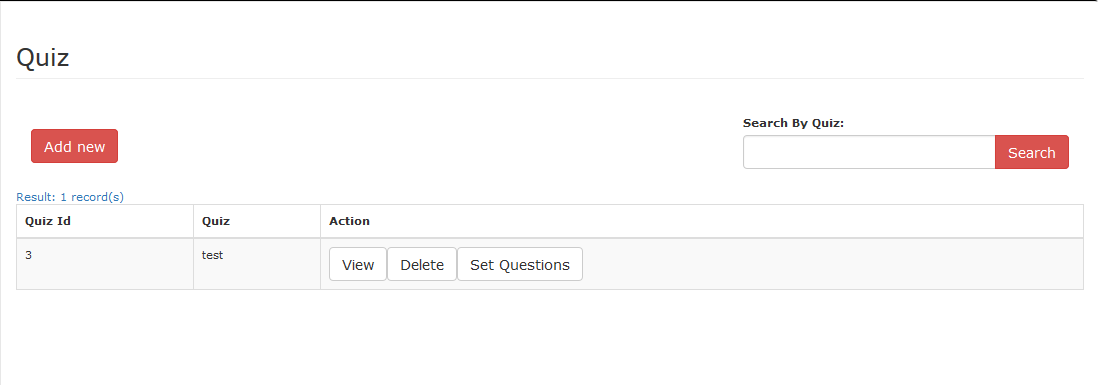
**Fig 37. Login Page**



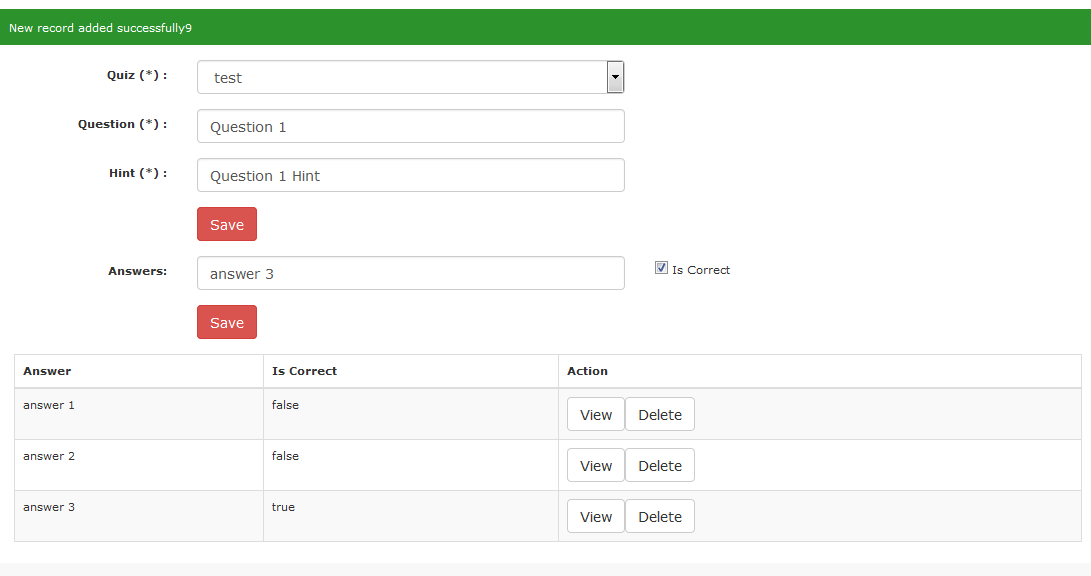
**Fig 38. Dashboard Menu**



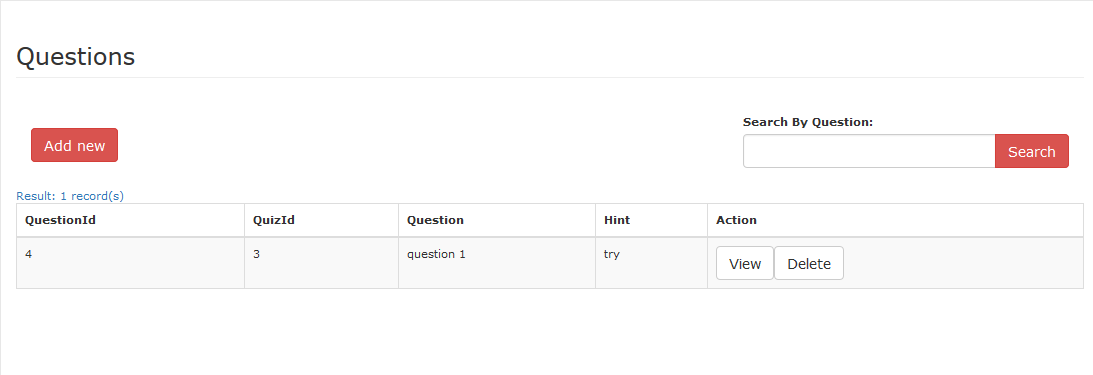
**Fig 39. Create Quiz**



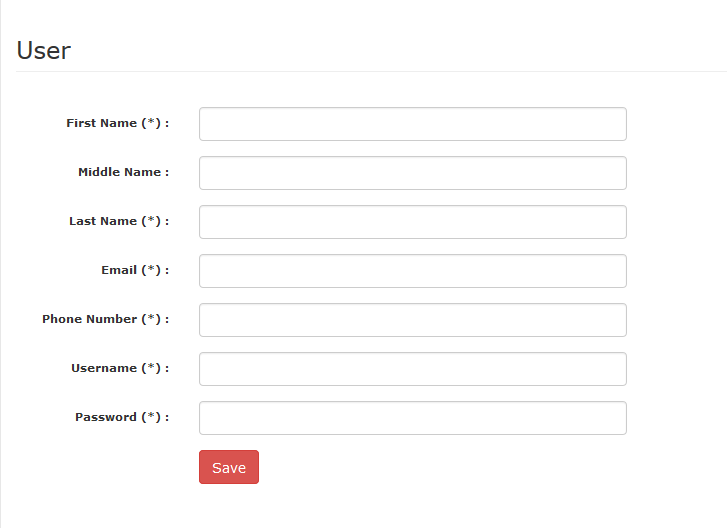
**Fig 40. Quiz List**



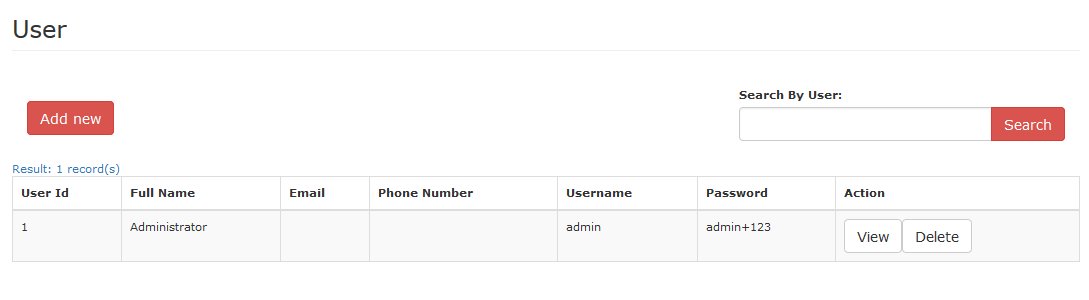
**Fig 41. Question and Answer setup**



**Fig 42. Question List**



**Fig 43. Create User**



**Fig 44. User’s List**

# 6. Behaviour Modelling

## 6.1. SD1 – Login

Instead of using the UserEJB, LoginEJB was created to verify the user credential.

## 6.2. SD2 – Create Quiz

As designed.

## 6.3. SD3 – Edit Quiz

As designed.

## 6.4. SD4 – Delete Quiz

As designed.

## 6.5. SD5 – List Quiz

As designed.

## 6.6. SD6 – View Quiz

As designed.

## 6.7. SD7 – Create Question

As designed.

## 6.8. SD8 – Edit Question

As designed.

## 6.9. SD 9 – Delete Question

As designed.

## 6.10. SD10 – List Question

As designed.

## 6.11. SD11 – View Question

As designed.

## 6.12. SD12 – Logout

As designed.

## 6.13. SD13 – Attempt Quiz

As designed.

# 7. Requirement Mapping

All functional and non-functional requirements are implemented as documented in the requirement specification document.