|  |
| --- |
| Central Queensland University, Sydney |
| Design Document  Rotary Youth Drive Awareness Project |
| COIT20273: Software Design and Development Project |
|  |
| Lecturer/Tutor: Zakiullah Khan Course Coordinator: Dr. Lily Li |
| **Due date: 8/25/2017** |

Project Members:

Bijay Shahi (12021856)

Keshav Khadka (S0280624)

Pratik Shrestha (S0282733)

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **Document Name** | **Version** | **Revision Date** |
| 1. | Design Document | 1.0 | 2017-08-25 |

Table of Contents

[1. Software Architecture 1](#_Toc491451724)

[2. Layer Modelling 3](#_Toc491451725)

[3. Database Design 4](#_Toc491451726)

[4. Class modelling 5](#_Toc491451727)

[4.1. Class diagram of JPA Entities 6](#_Toc491451728)

[4.2. Class diagram of EJBs 7](#_Toc491451729)

[4.3. Class diagram of backing beans 8](#_Toc491451730)

[5. User Interface Design 9](#_Toc491451731)

[5.1. Quiz Main page’s 9](#_Toc491451732)

[5.2. Admin Interface 12](#_Toc491451733)

[6. Behaviour Modelling 17](#_Toc491451734)

[6.1. SD1 – Login 17](#_Toc491451735)

[6.2. SD2 – Create Quiz 17](#_Toc491451736)

[6.3. SD3 – Edit Quiz 18](#_Toc491451737)

[6.4. SD4 – Delete Quiz 18](#_Toc491451738)

[6.5. SD5 – List Quiz 18](#_Toc491451739)

[6.6. SD6 – View Quiz 19](#_Toc491451740)

[6.7. SD7 – Create Question 19](#_Toc491451741)

[6.8. SD8 – Edit Question 19](#_Toc491451742)

[6.9. SD 9 – Delete Question 20](#_Toc491451743)

[6.10. SD10 – List Question 20](#_Toc491451744)

[6.11. SD11 – View Question 20](#_Toc491451745)

[6.12. SD12 – Logout 21](#_Toc491451746)

[6.13. SD13 – Attempt Quiz 21](#_Toc491451747)

[7. Requirement Mapping 22](#_Toc491451748)

# 1. Software Architecture

The Rotary Youth Driver Awareness (RYDA) application is designed in three tier application layers, namely, presentation, business logic and database layers. The figure 1 below, is the conceptual model of the RYDA application, and presents the system structure, business processes, behaviour, framework and product technologies. As per the proposal, the application will use the derby database to store and retrieve business data for the application. The business model is designed to map the domain model of the system, and is achieved by persisting the business objects as entities in the database. The designed entities are persisted to the database and later managed by Java Persistent API (JPA) using Java Persistence Query Language (JPQL). Enterprise Java Beans (EJBs) in the business logic encapsulate business logic and take care of transactions and security for the application. The backing beans in the business logic perform business logic, delegate the EJBs, handle navigation between pages and hold data. The presentation layer handles the interaction with the users with the Java Server Faces (JSF) and produce HTMLs. The face servlet is the main Servlet for the application and can be configured by faces-config.xml descriptor file. The renderer is responsible for displaying components and translating user’s input into component property values, whereas converters convert component values to and from mark-up values and validator validates the value entered by the user. The users communicate to the application using web pages in the web browsers using Hyper Text Transfer Protocol (HTTP).



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

# 2. Layer Modelling

The Java Server Faces (JSF) framework encourages MVC design pattern and the design for the RYDA application adheres to the MVC design pattern to isolate business logic from the user interface. The following MVC design model in figure 2 presents the model layer to represent the data of the application, the view to present the user interface and the controller to manage communication between the layers.

The model layer creates and manages the data in the database for the application and the information for the users. The JSF framework model layer for the application consist of backing beans, EJBs and JPA entities. The view layer in the application is designed to present the graphical representation for a model. The JSF framework in the application uses XHTML pages as user interfaces in the view layer. In the application, the controller is designed to manipulate user interactions by collecting, converting and validating the data, invoke the business logics and generate the content for the view. The JSF framework for the controller in the application is the FacesServlet.



**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. The objects identified for the RYDA application are users, administrator, student, quiz, questions, answers, correct answer and the student quiz attempts. As per the identified objects, the entities for the application are mapped to relational database tables. The entities attributes along with the relationship between the entities are mapped in the figure 3 below.

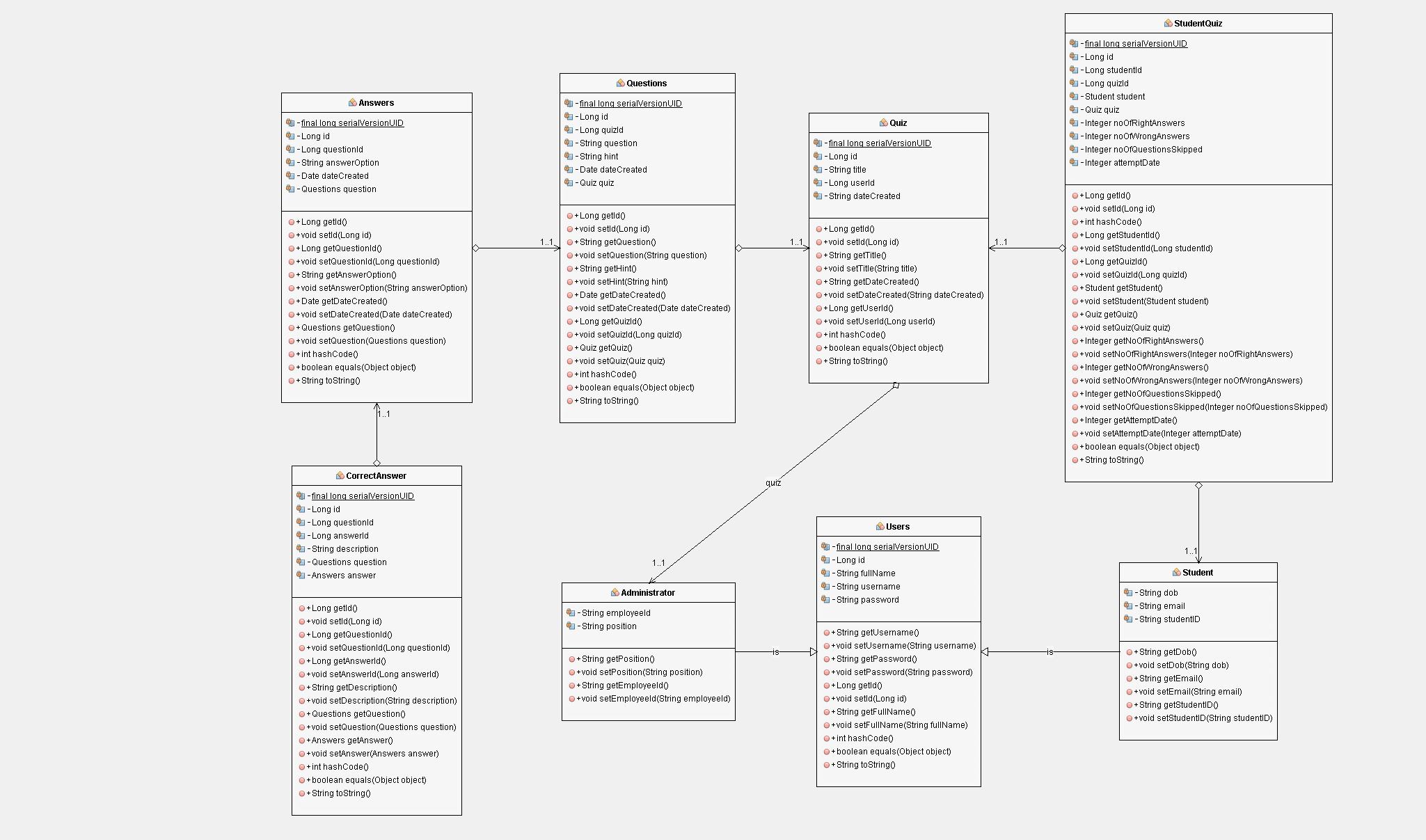


**Figure 3. Database design of RYDA application**

# 4. Class modelling

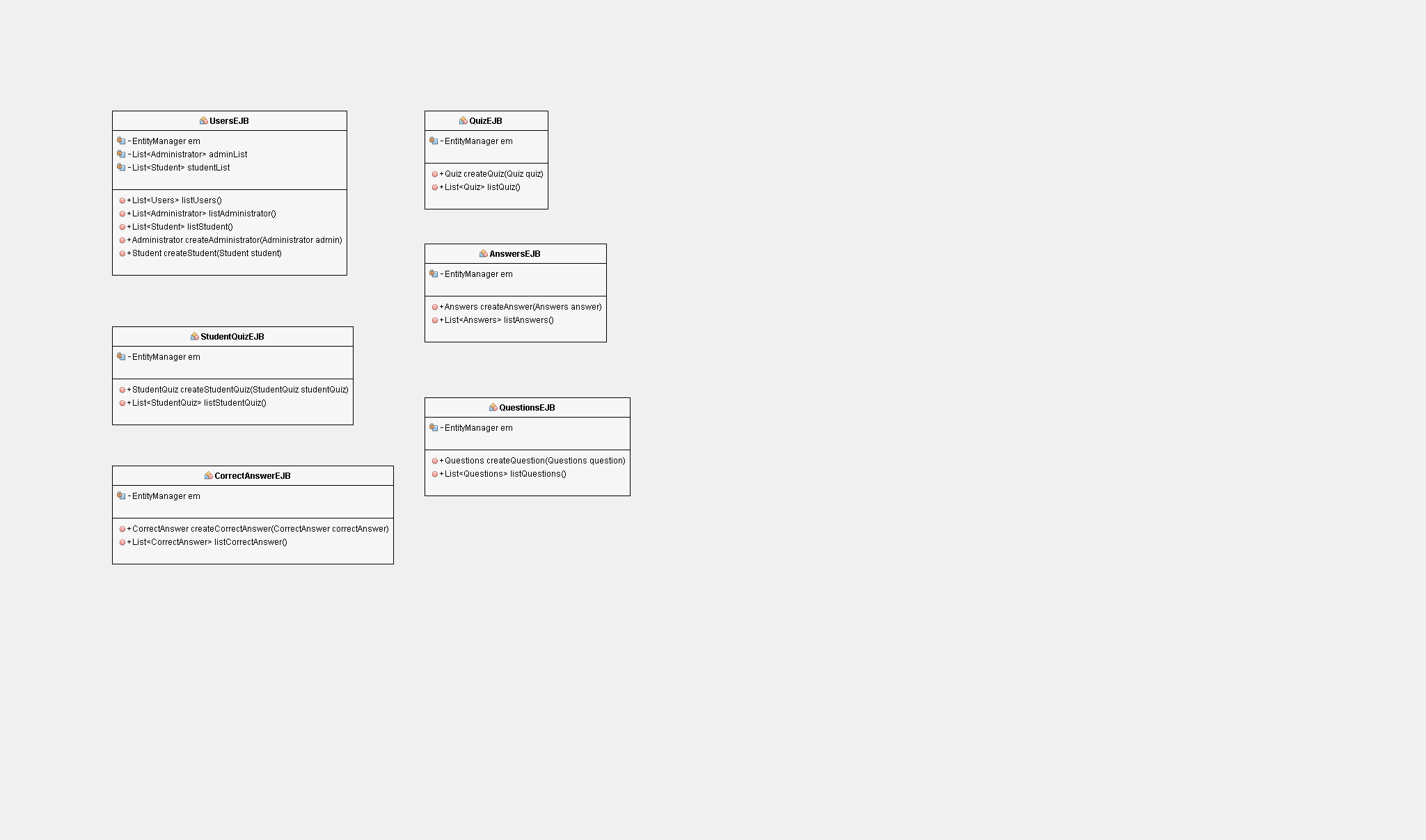
The following figure 4, 5 and 6 represent the class diagrams for the RYDA application. The class diagrams below are the model design for the application, which includes JPA entities, EJBs and the backing beans. As mentioned earlier the view layer for the application are the XHTML pages that uses JSF tags and components. The design for the view layer are generally the user interfaces which are mentioned later in the document. And the controller, as mentioned earlier, uses the Servlet which is an internal file and part of JSF framework. The Servlet can only be configured by using the external metadata through faces-config.xml descriptor. The following figure 4 represents the class diagram of the JPA entities, figure 5 represents the class diagram of EJBs and the figure 6 represents the class diagram of the backing beans for the model layer of the application.

## 4.1. Class diagram of JPA Entities



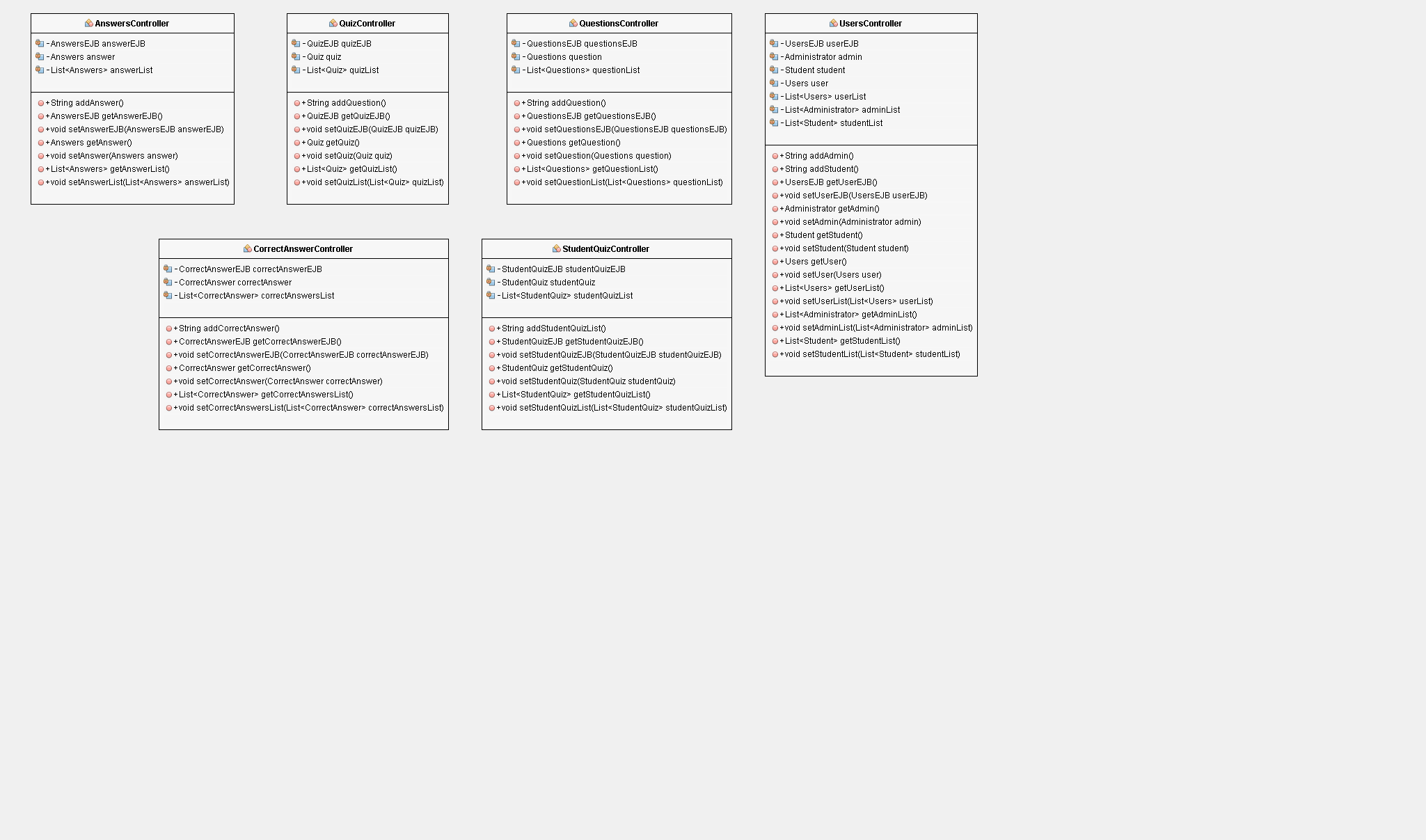
**Figure 4. Class diagram of JPA Entities**

## 4.2. Class diagram of EJBs



**Figure 5. Class diagram of EJBs**

## 4.3. Class diagram of backing beans

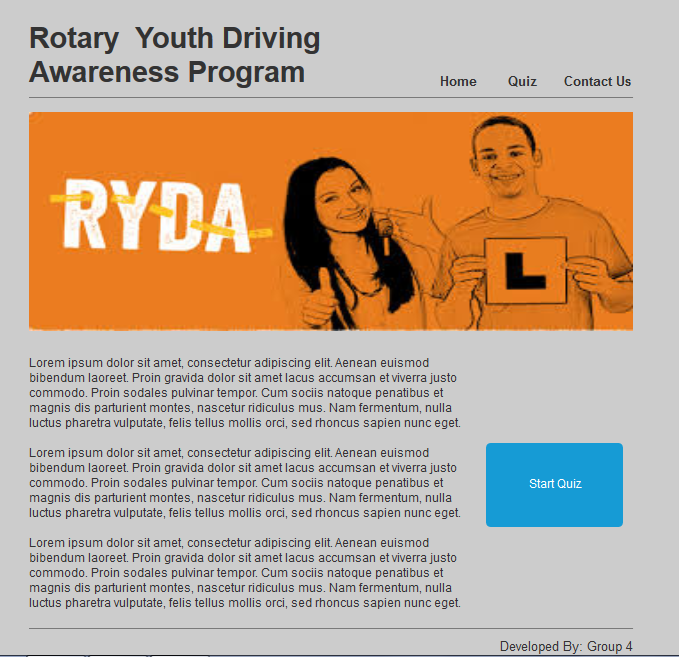


**Figure 6. Class diagram of backing beans**

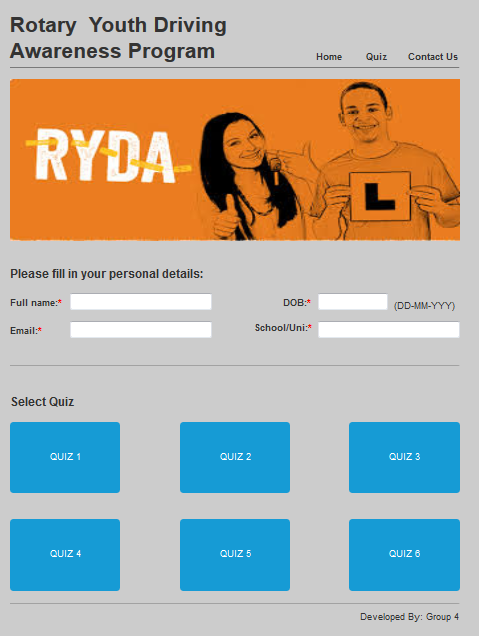
# 5. User Interface Design

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

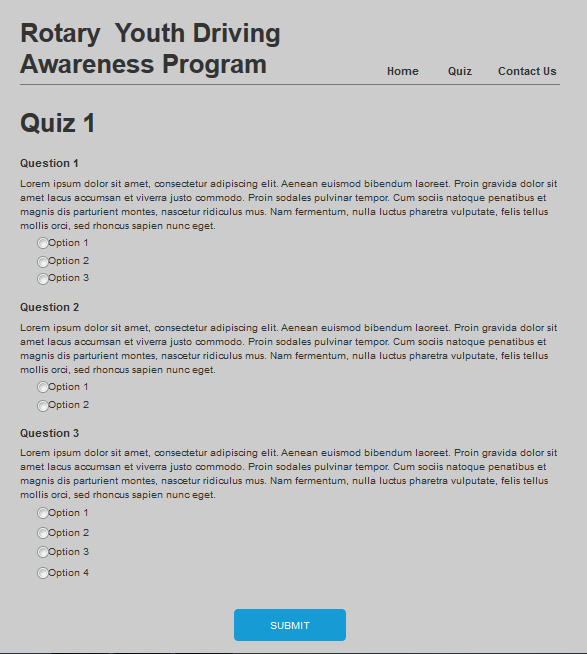
## 5.1. Quiz Main page’s



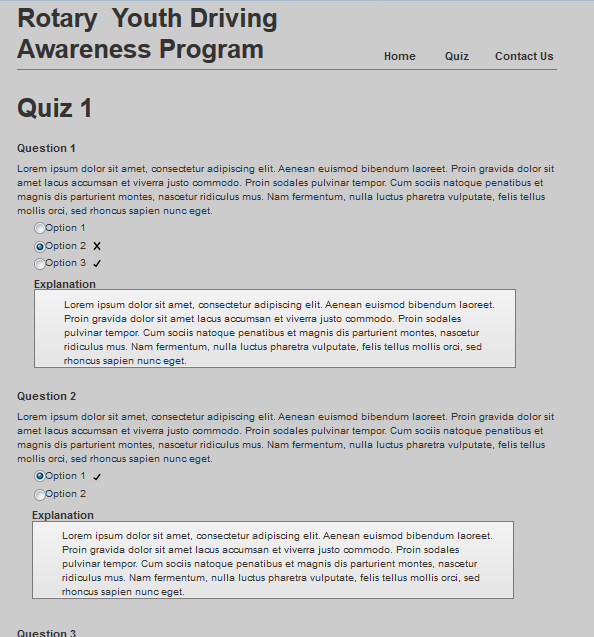
**Fig 7. Home Page**



**Fig 8. Personal detail and Quiz selection**

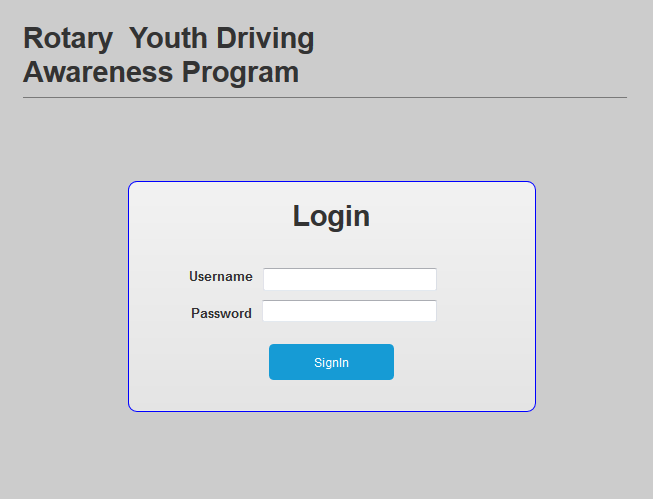


**Fig 9. Quiz**



**Fig. 10. Quiz Answers**

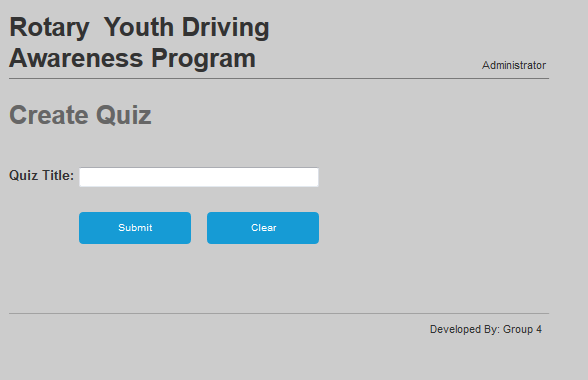
## 5.2. Admin Interface



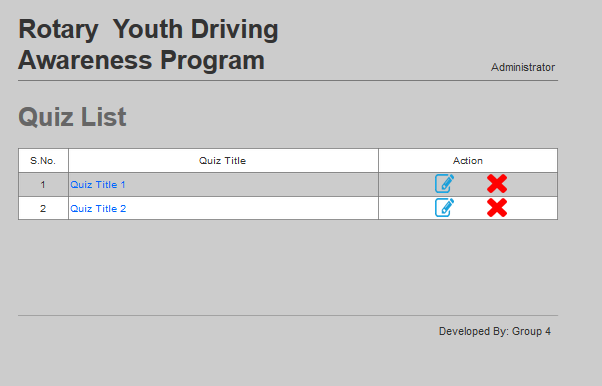
**Fig 11. Login Page**



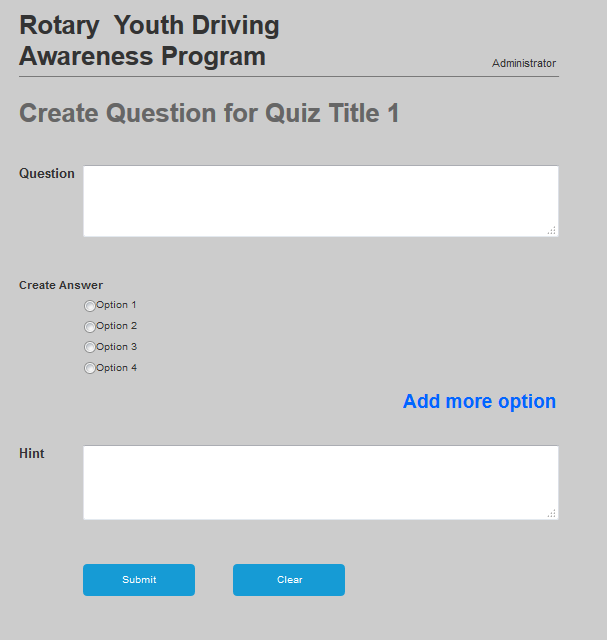
**Fig 12. Admin Menu**



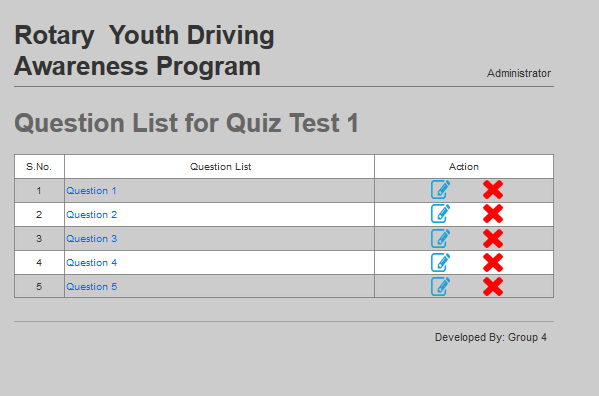
**Fig 13. Create Quiz**



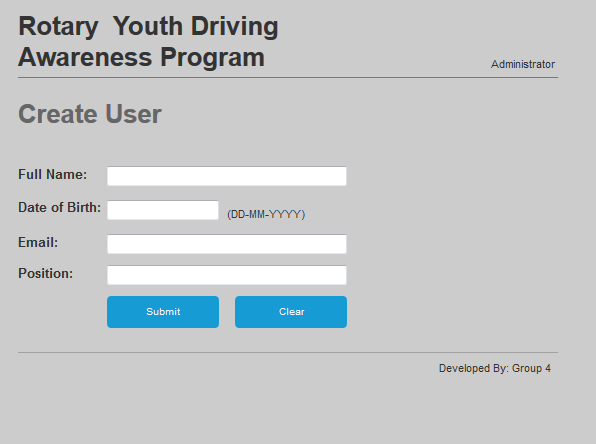
**Fig 14. Quiz List**



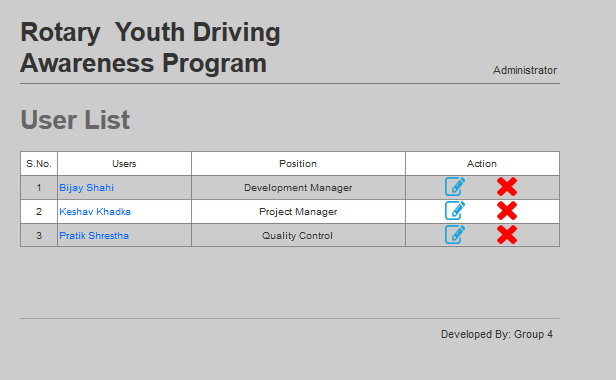
**Fig 15. Create Question**



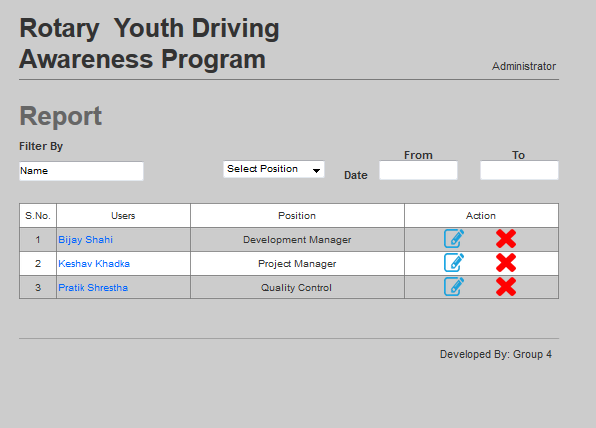
**Fig 16. Question List**



**Fig 17. Create User**



**Fig 18. User’s List**

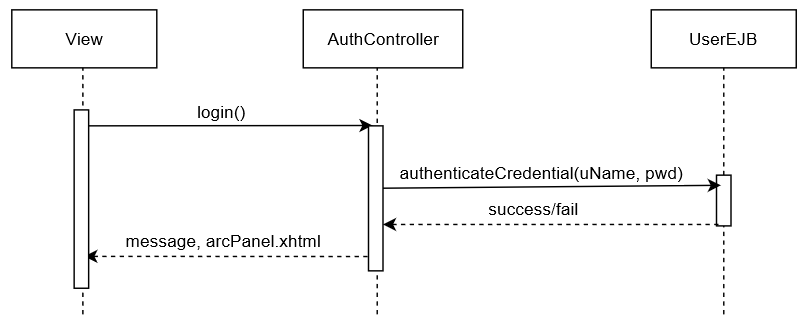


**Fig 19. Admin Report**

# 

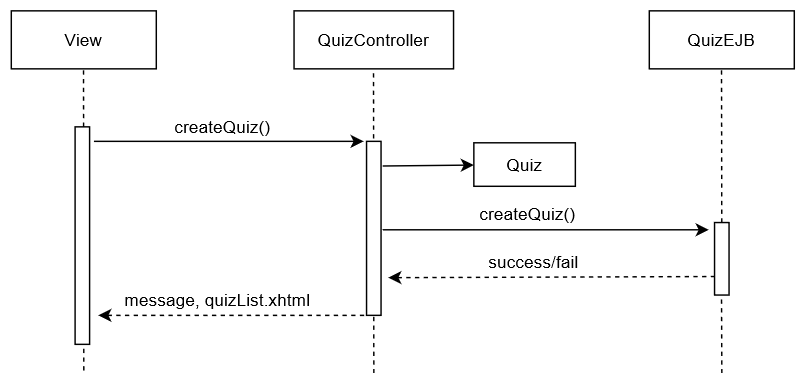
# 6. Behaviour Modelling

## 6.1. SD1 – Login



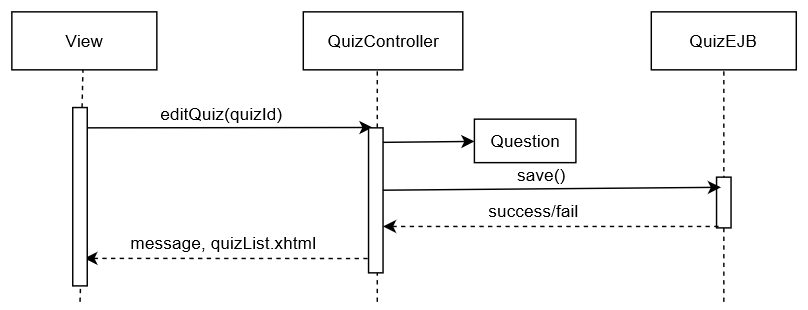
**Fig 20. SD1 – Login**

## 6.2. SD2 – Create Quiz



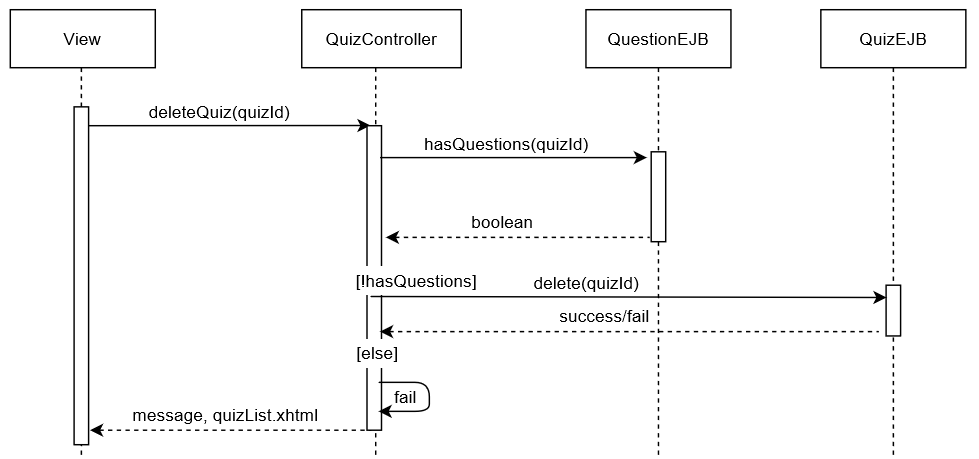
**Fig 21. SD2 – Create Quiz**

## 6.3. SD3 – Edit Quiz



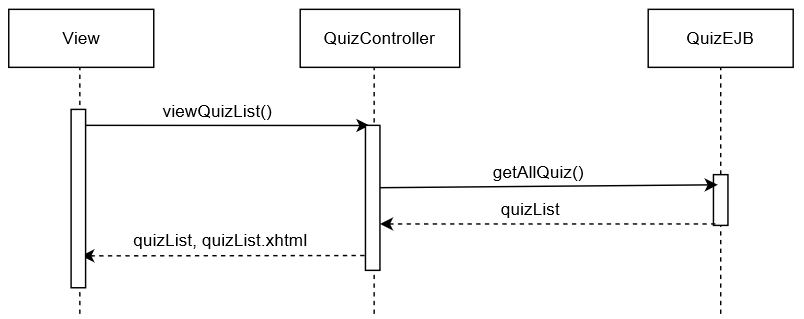
**Fig 22. SD3 – Edit Quiz**

## 6.4. SD4 – Delete Quiz



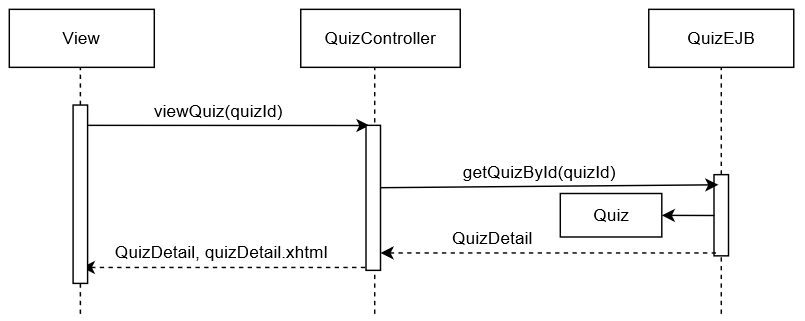
**Fig 23. SD4 – Delete Quiz**

## 6.5. SD5 – List Quiz



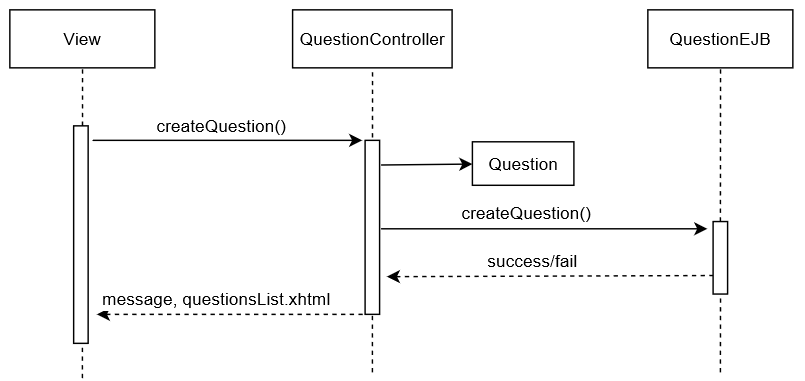
**Fig 24. SD5 – List Quiz**

## 6.6. SD6 – View Quiz



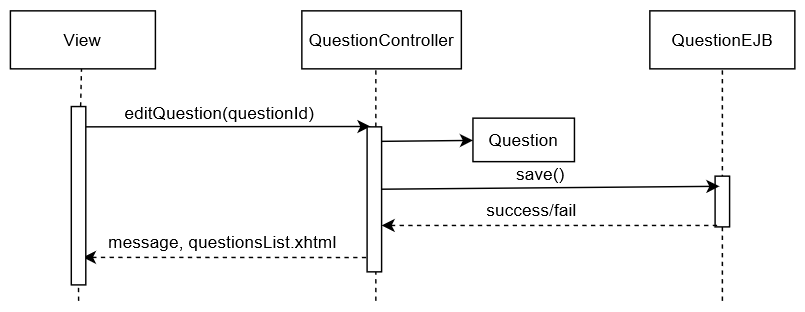
**Fig 25. SD6 – View Quiz**

## 6.7. SD7 – Create Question



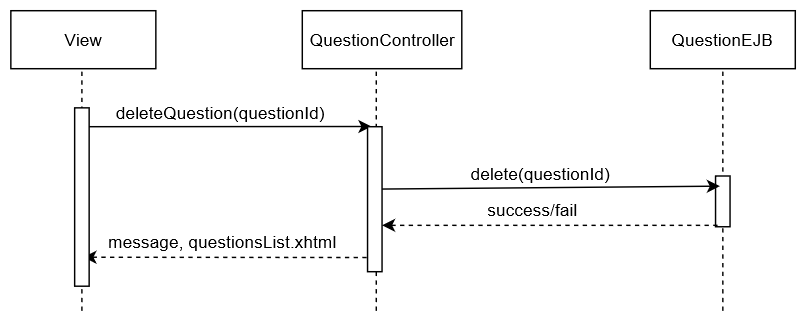
**Fig 26. SD7 – Create Question**

## 6.8. SD8 – Edit Question



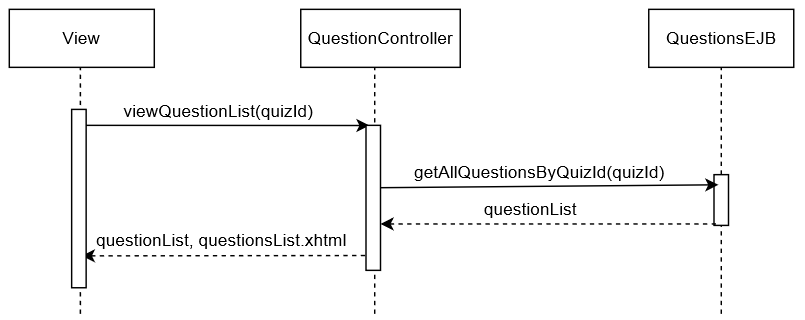
**Fig 27. SD8 – Edit Question**

## 6.9. SD 9 – Delete Question



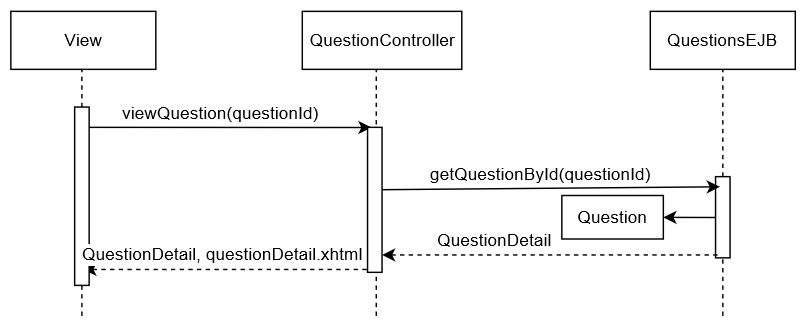
**Fig 28. SD9 – Delete Question**

## 6.10. SD10 – List Question



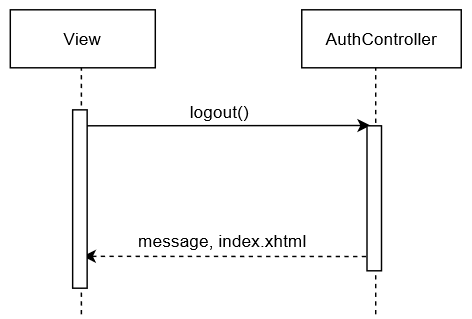
**Fig 29. SD10 - List Question**

## 6.11. SD11 – View Question



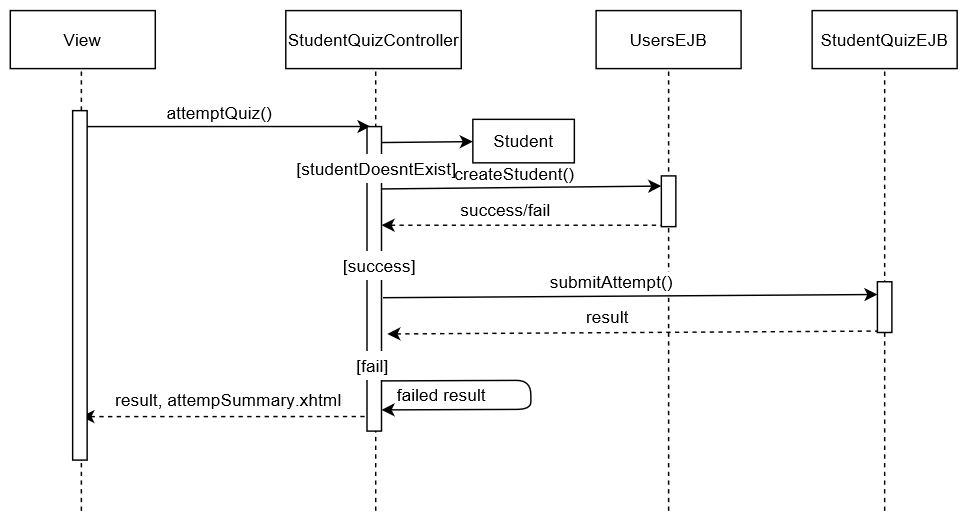
**Fig 30. SD11 - View Question**

## 6.12. SD12 – Logout



**Fig 31. SD12 – Logout**

## 6.13. SD13 – Attempt Quiz



**Fig 32. SD13 – Attempt Quiz**

# 7. Requirement Mapping

|  |  |
| --- | --- |
| **Use Case** | **Sequence Diagram** |
| Attempt Quiz | SD11 |
| View Quiz answers | Done through query |
| View Quiz result | Done through query |
| View explanation | Done through query |
| Login to the system | SD1 |
| Create/Update/Read/Delete quiz | SD2, SD3, SD4, SD5 |
| Explain quiz question | SD6 |
| Filter quiz results | Done through query |
| View student quiz history | Done through query |
| Log off the system | SD10 |

**Table 1. Requirement Mapping**