**Class diagram description**

**Controller package**

1. **Class name: UserController**

|  |
| --- |
| **UserController** |
| -userService : UserService |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | userService | Object of UserService, used for control feature about user. | **Type:** UserService |

1. **Class name: HistoryController**

|  |
| --- |
| **HistoryController** |
| -historyService : HistoryService |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | historyService | Object of HistoryService, used for control feature about history. | **Type:** HistoryService |

1. **Class name: ActivityController**

|  |
| --- |
| **ActivityController** |
| -activityService : ActivityService |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | activityService | Object of ActivityService, used for control feature about activity. | **Type:** ActivityService |

1. **Class name: QuestionController**

|  |
| --- |
| **QuestionController** |
| -questionService : QuestionService |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | questionService | Object of QuestionService, used for control feature about question and answer. | **Type:** QuestionService |

**Service package**

1. **Class name: UserServiceImpl**

|  |
| --- |
| **UserServiceImpl** |
| **-**userDAO : UserDAO |
| +addUser(User user): void  +getUser(User user): User |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | userDAO | Object of UserDAO, used for call feature about user. | **Type:** UserDAO |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | addUser | Method for create new user in the system. | **Return:** void  **Param:** User |
| 2 | getUser | Method for get user from database in the system. | **Return:** User  **Param:** User |

1. **Class name: HistoryServiceImpl**

|  |
| --- |
| **HistoryServiceImpl** |
| **-**historyDAO : HistoryDAO |
| +getHistory(): History  +editHistory(History history): void |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | historyDAO | Object of HistoryDAO, used for call feature about history. | **Type:** HistoryDAO |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | getHistory | Method for get history from database in the system. | **Return:** History |
| 2 | editHistory | Method for edit history in the system. | **Return:** void  **Param:** History |

1. **Class name: ActivityServiceImpl**

|  |
| --- |
| **ActivityServiceImpl** |
| **-**activityDAO : ActivtyDAO |
| +getActivity(): List<Activity>  +editActivity(long aid): void  +postActivity(Activity activity): void  +deleteActivity(long aid): void  +getActivityByID(long aid):Activity |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | activityDAO | Object of ActivityDAO, used for call feature about activity. | **Type:** ActivityDAO |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | getActivity | Method for get all activity from Database in the system. | **Return:** List<Activity> |
| 2 | editActivity | Method for edit activity in the system. | **Return:** void  **Param:** Long |
| 3 | postActivity | Method for post activity in the system. | **Return:** void  **Param:** Activity |
| 4 | deleteActivity | Method for get all activity from Database in the system. | **Return:** void  **Param:** Long |
| 5 | getActivityByID | Method for get activity from Database in the system. | **Return:** Activity  **Param:** Long |

1. **Class name: QuestionAnswerServiceImpl**

|  |
| --- |
| **QuestionAnswerServiceImpl** |
| **-**questionAnswerDAO : QuestionAnswerDAO |
| +DeleteQA(long qid): void  +answer(long qid, String answer): void  +getQuestion(): List<QuestionAnswer>  +askquestion(QuestionAnswer question): void |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | questionAnswerDAO | Object of QuestionAnswerDAO, used for call feature about question and answer. | **Type:** QuestionAnswerDAO |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | DeleteQA | Method for delete question and answer in the system. | **Return:** void |
| 2 | answer | Method for add answer in the system. | **Return:** void  **Param:** Long, String |
| 3 | getQuestion | Method for get all question and answer from Database in the system. | **Return:** List<QuestionAnswer> |
| 4 | askquestion | Method for create question in the system. | **Return:** void  **Param:** QuestionAnswer |

**Dao Package**

1. **Class name: UserDAOImpl**

|  |
| --- |
| **UserDAOImpl** |
| -userRepository : UserRepository |
| +addUser(User user) : void  +getUser(User user) : User |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | userRepository | Used for controller to database. | **Type:** UserRepository |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | addUser | Used for registration | **Return:** void  **Param:** User |
| 2 | getUser | Used for verify | **Return:** User  **Param:** User |

1. **Class name: HistoryDAOImpl**

|  |
| --- |
| **HistoryDAOImpl** |
| -historyRepository : HistoryRepository |
| +getHistory(): History  +editHistory(History history): void |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | historyRepository | Used for controller to database. | **Type:**HistoryRepository |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | getHistory | Used for retrieve history from database | **Return:** History |
| 2 | editHistory | Used for change information about history | **Return:** void  **Param:** History |

1. **Class name: ActivityDAOImpl**

|  |
| --- |
| **ActivityDAOImpl** |
| -activityRepository : ActivityRepository |
| +getactivity() : List<Activity> +editActivity(Activity activity) : void  +postActivity(Activity activity) : void  +deleteActivity(Activity activity) : void  +getActivityByID(long aid): Activity |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | activityRepository | Used for controller to database. | **Type:**ActivityRepository |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | getactivity | Used for retrieve list of activity in the database | **Return:** List<Activity> |
| 2 | editActivity | Used for change information about Activity | **Return:** void  **Param:** Activity |
| 3 | postActivity | Used for adding new Activity to the database. | **Return:** void  **Param:** Activity |
| 4 | deleteActivity | Used for delete activity | **Return:** void  **Param:** Activity |
| 5 | getActivityByID | Used for retrieve activity in the database | **Return:** Activity  **Param:** Long |

1. **Class name: QuestionAnswerDAOImpl**

|  |
| --- |
| **QuestionAnswerDAOImpl** |
| -questionAnswerRepository : QuestionRepository |
| +DeleteQA(QuestionAnswer questionAnswer) : void  +answer(long qid, String answer) : void  +getQuestion() : List<QuestionAnswer>  +askQuestion(QuestionAnswer question) : void |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | questionAnswerRepository | Used for controller to database. | **Type:**QuestionRepository |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | getQuestion | Used for retrieve list of question in the database | **Return:** List<QuestionAnswer> |
| 2 | DeleteQA | Used for delete Question. | **Return:** void  **Param:** QuestionAnswer |
| 3 | answer | Used for adding answer on selected question. | **Return:** void  **Param:** Long, String |
| 4 | askQuestion | Used for adding new question into database | **Return:** void  **Param:** QuestionAnswer |

**Entity package**

**Class name: User**

|  |
| --- |
| **User** |
| -userid :long  -username: string  -password : string  -email: string |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | userid | Store the id of the user | **Type:**long |
| 2 | username | Store the user name of the user | **Type:** String |
| 3 | password | Store the password of the user | **Type:** String |
| 4 | email | Store the email of the user | **Type:** String |

**Class name: Activity**

|  |
| --- |
| **Activity** |
| -aid :long  -aname: string  -adescription : string  -image: list<String>  -lang:char[2] |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | arid | Store the id of the activity | **Type:**long |
| 2 | aname | Store the name of the activity | **Type:** String |
| 3 | adescription | Store description of the activity | **Type:** String |
| 4 | image | Store the image url of the activity | **Type:** String |
| 5 | Lang | Store the language type of the activity | **Type** : array of char  char[2] |

**Class name: QuestionAnswer**

|  |
| --- |
| **QuestionAnswer** |
| -qid :long  -qname: string  -answer : string  -newQ : Boolean  -lang:char[2] |

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | qrid | Store the id of the question | **Type:**long |
| 2 | qname | Store the name of the question | **Type:** String |
| 3 | Answer | Store answer | **Type:** String |
| 4 | newQ | Store the image url of the question | **Type:** Boolean |
| 5 | Lang | Store the language type of the question | **Type** : array of char  char[2] |