**Unit Test**

**Member DAO:**

**UTC-01:** testGetMember() : List<Member>

**Description**

This test case tests whether the method can get the all Member objects in system or not. Test testGetMember method by getting list of member information in the database.

**Test Function**

public List<Member> getMember()

**Test Data**

\*member1 and member2 are in database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Name** | **Id** | **Email** | **Password** |
| Member1 | 1 | "se542115004@vr.camt.info" | "1234" |
| Member2 | 2 | "se552115065@vr.camt.onfo" | "5544" |

**Test Cases**

|  |  |  |
| --- | --- | --- |
| **Case** | **Description** | **Expected Result** |
| **Type: List of Member** |
| 1 | Test getting all of Member. | { member1 and member2 } |

**UTC-02:** testGetMemberByEmail() : Member List<Member>

**Description**

This test case test whether the method can get the right Member object by valid email and cannot get Member object by invalid email. Test testGetMemberByEmail method by getting member information in the database by using member email.

**Test Function**

Public Member getMemberByEmail(email : string)

**Test Data**

\*member1 and member2 are in database.

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Id** | **Email** |
| Member1 | 1 | "se542115004@vr.camt.info" |
| Member2 | 2 | "se552115065@vr.camt.onfo" |

**Test Cases**

|  |  |  |
| --- | --- | --- |
| **Case** | **Description** | **Expected Result** |
| **Type: String Email** |
| 1 | Test getting Member by inputting correct email. | { email1,email2 } |

**UTC-03:** testAddMemberByEmail() : Member

**Description**

This test case tests whether the method can add Member object to system or not. Test testAddMemberByEmail method by inputting member information in the database by using member email.

**Test Function**

Public Member addMember(member : Member)

**Test Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Name** | **Id** | **Name** | **Email** |
| Member1 | 1 | "Krittika Yothawuth" | "se542115004@vr.camt.info" |

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected Results** |
| **Type: User object** | **Type: member object** |
| 1 | Test with all parameters have input the data | Member1 | Can add |
| 2 | Test with input duplicate data | Member1 | Can’t add |
| 3 | Test with empty data | new member() | Can’t add |
| 4 | Test with null | Null | Can’t add |

**Admin DAO:**

**UTC-04**: testGetAdmin() : List<Admin>

**Description**

This test case tests whether the method can get the all Admin objects in system or not. Test testGetMember method by getting list of admin information in the database.

**Test Function**

Public List<Admin>getAdmin()

**Test Data**

\*admin1 and admin2 are in database.

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Id** | **Email** |
| Admin1 | 1 | "se542115004@vr.camt.info" |
| Admin2 | 2 | "se552115065@vr.camt.onfo" |

**Test** **Cases**

|  |  |  |
| --- | --- | --- |
| **Case** | **Description** | **Expected Result** |
| **Type: List of Admin** |
| 1 | Test getting all of Admin. | { admin1 } |

**UTC-05**: testGetAdminByEmail() : Admin

**Description**

This test case test whether the method can get the right Admin object by valid email and cannot get Admin object by invalid email. Test testGetAdminByEmail method by getting admin in formation in the database by using admin email.

**Test Function**

Public Admin getAdminByEmail(email : string)

**Test Data**

\*admin1 and admin2 are in database.

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Id** | **Email** |
| Admin1 | 1 | "se542115004@vr.camt.info" |
| Admin2 | 2 | "se552115065@vr.camt.onfo" |

**Test Cases**

|  |  |  |
| --- | --- | --- |
| **Case** | **Description** | **Expected Result** |
| **Type: String Email** |
| 1 | Test getting Member by inputting correct email. | { email1.eamil2 } |

**UTC-06**: testAddAdminByEmail() : Admin

**Description**

This test case tests whether the method can add Member object to system or not. Test testAddAdminByEmail method by inputting member information in the database by using admin email.

**Test Function**

Public Admin addAdmin(Admin : Admin)

**Test Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Name** | **Id** | **Name** | **Email** |
| Admin1 | 1 | "Krittika Yothawuth" | "se542115004@vr.camt.info" |

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected Results** |
| **Type: User object** | **Type: admin object** |
| 1 | Test with all parameters have input the data | Admin1 | Can add |
| 2 | Test with input duplicate data | Admin1 | Can’t add |
| 3 | Test with empty data | new admin() | Can’t add |
| 4 | Test with null | Null | Can’t add |

**Message DAO:**

**UTC-07**: testGetMessage() : List<Message>

**Description**

This test case tests whether the method can get the all Message in system or not. Test testGetMessage method by getting full message

**Test Function**

Public List<Message> getMessage()

**Test Data**

\*message1 and message2 are in database.

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Id** | **Message** |
| Message1 | 1 | "xxxxxxxxxxx" |
| Message2 | 2 | "zzzzzzzzzzzz" |

**Test Cases**

|  |  |  |
| --- | --- | --- |
| **Case** | **Description** | **Expected Result** |
| **Type: String Email** |
| 1 | Test getting Message all of message. | { message1,message2 } |

**UTC-08**: testAddMessage() : Message

**Description**

This test case tests whether the method can add Message object to system or not. Test testAddMessage method by adding message into the database.

**Test Function**

Public Message addMessage(Message message)

**Test Data**

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Id** | **Message** |
| Message1 | 1 | "xxxxxxxxxxx" |

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected Results** |
| **Type: message object** | **Type: message object** |
| 1 | Test with all parameters have input the data | Message1 | Can add |
| 2 | Test with input duplicate data | Message1 | Can’t add |
| 3 | Test with empty data | new message() | Can’t add |
| 4 | Test with null | Null | Can’t add |

**UTC-09**: testSendMessage() : Message

**Description**

This test case tests whether the method can send Message object to system or not. Test testSendMessage method by sending message object to the system

**Test Function**

public Message sendMessage(Message message)

**Test Data**

\*message1 in database

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Id** | **Message** |
| Message1 | 1 | "xxxxxxxxxxx" |

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected Results** |
| **Type: User object** | **Type: message** |
| 1 | Test with all parameters have input the data | Message1 | Can send |
| 2 | Test with input duplicate data | Message1 | Can’t send |
| 3 | Test with empty data | Null | Can’t send |
| 4 | Test with null | Null | Can’t send |

**History DAO:**

**UTC-10**: testGetHistory() : List<History>

**Description**

This test case tests whether the method can get the all History in system or not. Test testGetHistory method by getting list of history information in the database.

**Test Function**

Public List<History> getHistory()

**Test Data**

\* history1 , history2 , history3 and history4 is in database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object Name** | **Admin ID** | **Time** | **Date** | **Status** |
| History1 | 1 | "08:30:00" | "2015-06-13" | 0 |
| History2 | 2 | "08:45:00" | "2015-08-06" | 1 |
| History3 | 3 | "09:00:00" | "2012-09-25" | 0 |
| History4 | 4 | "09:15:00" | "2000-01-31" | 1 |

**Test Cases**

|  |  |  |
| --- | --- | --- |
| **Case** | **Description** | **Expected Result** |
| **Type: List of History object** |
| 1 | Test getting list of history | { history1 , history2 , history3 and history4 } |

**UTC-11**: testUpdateHistory() : History

**Description**

This test case tests whether the method can update the History in system or not. Test testUpdateHirtory method by inputting history information to the database.

**Test Function**

Public History updateHistory(history : History)

**Test Data**

\* history1 , history2 , history3 and history4 is in databases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object Name** | **Admin ID** | **Time** | **Date** | **Status** |
| History1 | 1 | "08:30:00" | "2015-06-14" | 0 |
| History2 | 2 | "08:30:00" | "2015-06-14" | 0 |
| History3 | 3 | null | "2015-06-14" | 0 |
| History4 | 4 | "08:30:00" | null | 0 |

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected Results** |
| **Type: History object** | **Type: history** |
| 1 | Test with all parameters have input the AdminID, Time, Date and Status | History1 | Can update |
| 2 | Test with input duplicate data | History1 | Can’t update |
| 3 | Test with null time | History3 | Can’t update |
| 4 | Test with null date | History4 | Can’t update |
| 5 | Test with empty data | new History() | Can’t update |
| 6 | Test with null | Null | Can’t update |

**QA DAO:**

**UTC-12**: testGetQuestion() : List<QA>

**Description**

This test case tests whether the method can get the all Question in system or not. Test testGetQuestion method by getting list of question information in the database.

**Test Function**

Public List<QA> getQuestion()

**Test Data**

\* question1 , question2 , question3 are in database

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Id** | **Question** |
| Question1 | 1 | "Q1Q1Q1" |
| Question2 | 2 | "Q2Q2Q2" |
| Question3 | 3 | “Q3Q3Q3” |

**Test Cases**

|  |  |  |
| --- | --- | --- |
| **Case** | **Description** | **Expected Result** |
| **Type: List of question object** |
| 1 | Test getting list of question | { question1 , question2 , question3} |

**UTC-13:** testGetQuestionById() : Question

**Description**

This test case test whether the method can get the right Question object by valid questionId and cannot get Question object by invalid. Test testGetQuestionById method by getting question information in the database by using question id.

**Test Function**

Public Question getQuestionByld(questionId: long)

**Test Data**

\* question1 , question2 , question3 and question4 is in database

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Id** | **Question** |
| Question1 | 1 | "Q1Q1Q1" |
| Question2 | 2 | "Q2Q2Q2" |
| Question3 | 3 | “Q3Q3Q3” |

**Test Cases**

|  |  |  |
| --- | --- | --- |
| **Case** | **Description** | **Expected Result** |
| **Type: long question** |
| 1 | Test getting Question by inputting correct id. | { id1,id2,id3 } |

**UTC-14:** testAddQuestion(): Question

**Description**

This test case tests whether the method can add Question object to system or not. Test testAddQuestion method by inputting question into the database.

**Test Function**

Public Question addQuestion(question : Question)

**Test Data**

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Id** | **Question** |
| Question1 | 1 | "Q1Q1Q1" |

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected Results** |
| **Type: Question object** | **Type: question** |
| 1 | Test with all parameters have input the data | Question1 | Can add |
| 2 | Test with input duplicate data | Question1 | Can’t add |
| 3 | Test with empty data | new question() | Can’t add |
| 4 | Test with null | Null | Can’t add |

**UTC-15:** testAddAnswer (): Answer

**Description**

This test case tests whether the method can add Answer object to system or not. Test testAddAnswer method by inputting answer into the database

**Test Function**

Public Answer addAnswer(answer : Answer)

**Test Data**

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Id** | **Question** |
| Answer1 | 1 | "Q1Q1Q1" |

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected Results** |
| **Type: Question object** | **Type: question** |
| 1 | Test with all parameters have input the data | Question1 | Can add |
| 2 | Test with input duplicate data | Question1 | Can’t add |
| 3 | Test with empty data | new question() | Can’t add |
| 4 | Test with null | Null | Can’t add |

**UTC-16:** testGetUserQuestion() : List<Question>

**Description**

This test case tests whether the method can get the all User Question in system or not. Test testGetUserQuestion method by getting list of user question in the database.

**Test Function**

Public List<Question> getUserQuestion()

**Test Data**

\* userQuestion1 , userQuestion2 , userQuestion3 are in database

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Id** | **Question** |
| userQuestion1 | 1 | "uQ1uQ1uQ1" |
| userQuestion2 | 2 | "uQ2uQ2uQ2" |
| userQuestion3 | 3 | “uQ3uQ3uQ3” |

**Test Cases**

|  |  |  |
| --- | --- | --- |
| **Case** | **Description** | **Expected Result** |
| **Type: List of user question object** |
| 1 | Test getting list of userQuestion | {userQuestion1 , userQuestion2 , userQuestion3} |

**Activity DAO:**

**UTC-17:** testGetActivity() : List<Activity>

**Description**

This test case tests whether the method can get the all Activity in system or not. Test testGetActivity method by getting list of activity information in the database.

**Test Function**

Public List<Activity> getActivity()

**Test Data**

\* activity1 , activity2 , activity3 and activity4 are in database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object Name** | **Title** | **Description** | **Date** | **Status** |
| Activity1 | A | "AAAA" | "2015-06-13" | 0 |
| Activity2 | B | "BBBB" | "2015-08-06" | 1 |
| Activity3 | C | "CCCC" | "2012-09-25" | 0 |
| Activity4 | D | "DDDD" | "2000-01-31" | 1 |

**Test Cases**

|  |  |  |
| --- | --- | --- |
| **Case** | **Description** | **Expected Result** |
| **Type: List of History object** |
| 1 | Test getting list of activity | { acivity1 , activity2 , activity3 and activity4 } |

**UTC-18:** testGetActivityByType() : Activity

**Description**

This test case test whether the method can get the right Activity object by valid activityType and cannot get Activity object by invalid activityType. Test testGetActivityByType method by getting activity information into the database by using activity type.

**Test Function**

Public Activity getActivityByType(Type: string)

**Test Data**

\* activity1 , activity2 , activity3 and activity4 are in database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object Name** | **Title** | **Description** | **Date** | **Type** |
| Activity1 | A | "AAAA" | "2015-06-13" | Annual |
| Activity2 | B | "BBBB" | "2015-08-06" | Annual |
| Activity3 | C | "CCCC" | "2012-09-25" | Annual |
| Activity4 | D | "DDDD" | "2000-01-31" | Other |

**Test Cases**

|  |  |  |
| --- | --- | --- |
| **Case** | **Description** | **Expected Result** |
| **Type: string** |
| 1 | Test getting Activity by inputting correct type. | { annual, annual, annual, other } |

**UTC- 19:** testAddActivity() : Acivity

**Description**

This test case tests whether the method can add Activity object to system or not. Test testAddActivity method by adding activity information into the database.

**Test Function**

Public Activity addActivity(Activity activity)

**Test Data**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object Name** | **Title** | **Description** | **Date** | **Type** |
| Activity5 | E | "EEEE" | "2015-07-10" | Other |

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected Results** |
| **Type: Activity object** | **Type: activity** |
| 1 | Test with all parameters have input the data | Activity5 | Can add |
| 2 | Test with input duplicate data | Activity5 | Can’t add |
| 3 | Test with empty data | new activity() | Can’t add |
| 4 | Test with null | Null | Can’t add |

**UTC-20:** testEditActivity() : Activity

**Description**

This test case tests whether the method can edit Activity object to system or not. Test testEditActivity method by editing activity information into the database.

**Test Function**

Public Activity editActivity(Activity activity)

**Test Data**

\* activity1 , activity2 , activity3 and activity4 are in database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object Name** | **Title** | **Description** | **Date** | **Type** |
| Activity1 | A | "AAAA" | "2015-06-13" | Annual |
| Activity2 | B | "BBBB" | "2015-08-06" | Annual |
| Activity3 | null | "CCCC" | "2012-09-25" | Annual |
| Activity4 | D | "DDDD" | "null" | Other |

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected Results** |
| **Type: Activity object** | **Type: activity** |
| 1 | Test with all parameters have input the Title, Description, Date and Type | Activity1 | Can edit |
| 2 | Test with input duplicate data | Activity1 | Can’t edit |
| 3 | Test with null title | Activity3 | Can’t edit |
| 4 | Test with null date | Activity4 | Can’t edit |
| 5 | Test with empty data | new activity() | Can’t edit |
| 6 | Test with null | Null | Can’t edit |

**UTC-21:** testDeleteActivity() : Activity

**Description**

This test case tests whether the method can delete Activity object to system or not. Test testDeleteActivity method by deleting activity information into the databas.

**Test Function**

Public Activity getDeleteActivity(Activity activity)

**Test Data**

\* activity1 , activity2 , activity3 and activity4 are in database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object Name** | **Title** | **Description** | **Date** | **Type** |
| Activity1 |  | “ “ | “ ” |  |
| Activity2 |  | "BBBB" | "2015-08-06" | Annual |
| Activity3 | C | "CCCC" | " " | Annual |
| Activity4 | D | "DDDD" | "2000-01-31" | Other |

**Test Case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected Results** |
| **Type: Activity object** | **Type: activity** |
| 1 | Test with all parameters have delete the Title, Description, Date and Type | Activity1 | Can test |
| 2 | Test with delete title | Activity2 | Can’t test |
| 3 | Test with delete date | Activity3 | Can’t test |
| 4 | Test with empty data | new activity() | Can’t test |
| 5 | Test with null | Null | Can’t test |

**Member Controller:**

**UTC-22:** testLogin() : Member

**Description**

Test testLogin method by getting list of user information in the database by using member email and member password.

**Test Function**

Public Member login(email : string, password : string)

**Test Data**

\*member1 in database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Name** | **Id** | **Email** | **Password** |
| Member1 | 1 | "se542115004@vr.camt.info" | "1234" |

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected Results** |
| **Type: User object** | **Type: member** |
| 1 | Test with all parameters have input the data | Member1 | Can login |
| 2 | Test with input duplicate data | Member1 | Can’t login |
| 3 | Test with empty data | new member1() | Can’t login |
| 4 | Test with null | Null | Can’t login |

**Admin DAO:**

**UTC-23**: testLogin() : Admin

**Description**

Test testLogin method by getting list of admin information in the database by using admin email and admin password.

**Test** **Function**

Public Admin login(email : string, password : string)

**Test** **Data**

\*admin1 in database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Name** | **Id** | **Email** | **Password** |
| Admin1 | 1 | "se542115004@vr.camt.info" | "1234" |

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected Results** |
| **Type: User object** | **Type: admin** |
| 1 | Test with all parameters have input the data | Admin1 | Can login |
| 2 | Test with input duplicate data | Admin1 | Can’t login |
| 3 | Test with empty data | new admin1() | Can’t login |
| 4 | Test with null | Null | Can’t login |