**Unit Test**

**Member Controller:**

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

**Description**

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**

Test testGetMemberByEmail method by getting member information in the database by using member email.

**Test Function**

Public 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**

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** |
| 1 | Test with all parameters have input the data | Member1 | True |
| 2 | Test with input duplicate data | Member1 | False |
| 3 | Test with empty data | new member1() | False |
| 4 | Test with null | Null | False |

**UTC-04:** 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 | True |
| 2 | Test with input duplicate data | Member1 | False |
| 3 | Test with empty data | new member1() | False |
| 4 | Test with null | Null | False |

**Admin Controller:**

**UTC-05**: testGetAdmin() : List<Admin>

**Description**

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-06**: testGetAdminByEmail() : Admin

**Description**

Test testGetAdminByEmail method by getting admin information in the database by using admin email.

**Test Function**

Public 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-07**: testAddAdminByEmail() : Admin

**Description**

Test testAddAdminByEmail method by inputting member information in the database by using admin email.

**Test Function**

Public Member 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** |
| 1 | Test with all parameters have input the data | Admin1 | True |
| 2 | Test with input duplicate data | Admin1 | False |
| 3 | Test with empty data | new admin1() | False |
| 4 | Test with null | Null | False |

**UTC-08**: 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 | True |
| 2 | Test with input duplicate data | Admin1 | False |
| 3 | Test with empty data | new admin1() | False |
| 4 | Test with null | Null | False |

**Message Controller:**

**UTC-09**: testGetMessage() : List<Message>

**Description**

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-10**: testAddMessage() : Message

**Description**

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: User object** | **Type: message** |
| 1 | Test with all parameters have input the data | Message1 | True |
| 2 | Test with input duplicate data | Message1 | False |
| 3 | Test with empty data | new message1() | False |
| 4 | Test with null | Null | False |

**UTC-11**: testSendMessage() : Message

**Description**

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 | True |
| 2 | Test with input duplicate data | Message1 | False |
| 3 | Test with empty data | null | False |
| 4 | Test with null | Null | False |

**History Controller:**

**UTC-12**: testGetHistory() : List<History>

**Description**

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-13**: testUpdateHistory() : History

**Description**

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 | True |
| 2 | Test with input duplicate data | History1 | False |
| 3 | Test with null time | History3 | False |
| 4 | Test with null date | History4 | False |
| 5 | Test with empty data | new History() | False |
| 6 | Test with null | Null | False |

**QA Controller:**

**UTC-14**: testGetQuestion() : List<QA>

**Description**

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-15:** testGetQuestionById() : Question

**Description**

Test testGetQuestionById method by getting question information in the database by using question id.

**Test Function**

Public 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-16:** testAddQuestion(): Question

**Description**

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 | True |
| 2 | Test with input duplicate data | Question1 | False |
| 3 | Test with empty data | new question1() | False |
| 4 | Test with null | Null | False |

**UTC-17:** testAddAnswer (): Answer

**Description**

Test testAddAnswer method by inputting answer into the database

**Test Function**

Public Question 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 | True |
| 2 | Test with input duplicate data | Question1 | False |
| 3 | Test with empty data | new question1() | False |
| 4 | Test with null | Null | False |

**UTC-18:** testGetUserQuestion() : List<Question>

**Description**

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 Controller:**

**UTC-19:** testGetActivity() : List<Activity>

**Description**

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-20:** testGetActivityByType() : Activity

**Description**

Test testGetActivityByType method by getting activity information into the database by using activity type.

**Test Function**

Public 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- 21:** testAddActivity() : Acivity

**Description**

Test testAddActivity method by adding activity information into the database.

**Test Function**

Public Message 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 | True |
| 2 | Test with input duplicate data | Activity5 | False |
| 3 | Test with empty data | new activity5 () | False |
| 4 | Test with null | Null | False |

**UTC-22:** testEditActivity() : Activity

**Description**

Test testEditActivity method by editing activity information into the database.

**Test Function**

Public Message 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 | True |
| 2 | Test with input duplicate data | Activity1 | False |
| 3 | Test with null title | Activity3 | False |
| 4 | Test with null date | Activity4 | False |
| 5 | Test with empty data | new activity() | False |
| 6 | Test with null | Null | False |

**UTC-23:** testDeleteActivity() : Activity

**Description**

Test testDeleteActivity method by deleting activity information into the databas.

**Test Function**

Public 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 | True |
| 2 | Test with delete title | Activity2 | False |
| 3 | Test with delete date | Activity3 | False |
| 4 | Test with empty data | new activity() | False |
| 5 | Test with null | Null | False |

**Member DAO:**

**UTC-1**

public Member getMemberByEmail(Email : String)

**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 Case**

**UTC-2**

public List<Member> getMember()

**Description** This test case tests whether the method can get the all Member objects in system or not.

**Test Case**

**UTC-3**

Public Member addMember(Member member)

**Descripton** This test case tests whether the method can add Member object to system or not.

**Test Case**

**Admin DAO:**

**UTC-4**

public Member getAdminByEmail(Email : String)

**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 Case**

**UTC-5**

public List<Admin> getAdmin()

**Description** This test case tests whether the method can get the all Admin objects in system or not.

**Test Case**

**UTC-6**

public Admin addAdmin(Admin admin)

**Descripton** This test case tests whether the method can add Member object to system or not.

**Test Case**

**Message DAO:**

**UTC-7**

public List<Message> getMessage()

**Description** This test case tests whether the method can get the all Message in system or not.

**Test Case**

**UTC-8**

public Message addMessage(Message massage)

**Description** This test case tests whether the method can add Message object to system or not.

**Test Case**

**UTC-9**

public Admin sendMessage(Message message)

**Descripton** This test case tests whether the method can send Message object to system or not.

**Test Case**

**History DAO:**

**UTC-10**

public List<History> getHistory()

**Description** This test case tests whether the method can get the all History in system or not.

**Test Case**

**UTC-11**

public History updateHistory(History history)

**Description** This test case tests whether the method can update the History in system or not.

**Test Case**

**QA DAO:**

**UTC-**

public List<QA> getQuestion()

**Description** This test case tests whether the method can get the all Question in system or not.

**Test Case**

**UTC-**public Question getQuestionByUserId(long QuestionId)  
**Description** This test case test whether the method can get the right Question object by valid questionId and cannot get Question object by invalid questionId.

**Test Case**

**UTC-**

public Question addQuestion(Question question)

**Description** This test case tests whether the method can add Question object to system or not. **Test Case**

**UTC-**public Answer addAnswer(Answer answer)  
**Description** This test case tests whether the method can add Answer object to system or not.  
**Test Case**

**UTC-**public List<Question> getUserQuestion(Email : string)  
**Description** This test case tests whether the method can get the all User Question in system or not.

**Test Case**

**Activity DAO:**

**UTC-**public List<Activity> getActivity() **Description** This test case tests whether the method can get the all Activity in system or not.

**Test Case**

**UTC-**

public Activity getActivityByType(Type : string)  
**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 Case**

**UTC-**

public Activity addActivity(Activity activity)  
**Description**  This test case tests whether the method can add Activity object to system or not.

**Test Case**

**UTC-**public Activity editActivity(Activity activity)

**Description** This test case tests whether the method can edit Activity object to system or not.

**Test Case**

**UTC-**

public Activity deleteActivity(Activity activity)  
**Description**  This test case tests whether the method can delete Activity object to system or not.

**Test Case**