

**Analysis and Design Document**

**Project Name**

Student Portal

**Project Title**

An application for student access to university-related resources

**Student details**

**Name:** Sergiu Tripon

**Phone:** +44 (0)786 904 7206

**E-mail:** [Sergiu.Tripon.1@city.ac.uk](mailto:Sergiu.Tripon.1@city.ac.uk)

**Supervisor details**

**Name:** Vladimir Stankovic

**E-mail:** v.stankovic@city.ac.uk

**Phone:** +44 (0)20 7040 0273

**Visit:** A229a, College Building

**Document details**

**Creation date:** 21st December 2014 **Last updated:** 8th January 2015

Contents

**Use case specification and diagrams1**

**Database scripts45**

**Entity relationship diagrams53**

**Class diagrams54**

[Use case specification & Use case diagram 1](#_Toc418343207)

# Use case specification & Use case diagram

**Account**

**Use case 1 (Account - Update account)**

**1. Brief description**

This use case describes how the user will use Student Portal to update their account.

**2. Actors**

2.1 User (account type: administrator, student, lecturer)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Account”

3. The user will click/tap on “Update account”

4. The system will load up an “Update account” form

5. The system loads up the existing personal details in specific labelled textboxes

6. The user will fill in one of the potentially empty textboxes or they will overwrite their already existing details

7. The user will click/tap the “Update” button

8. The system will update the user’s personal details.

9. The system will display a confirmation message.

10. The system will send a confirmation e-mail to the user

**5. Alternative flows**

None

**6. Key Scenarios**

None

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully updated their account.

**7.2 Failure condition**

None

**Use case 2 (Account - Change Password)**

**1. Brief description**

This use case describes how the user will use Student Portal to change their password.

**2. Actors**

2.1 User (account type: administrator, student, lecturer)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Account”

3. The user will click/tap on “Change Password”

4. The system will load up a “Change Password” form

5. The user will fill in the form

6. The user will click/tap the “Change” button

7. The system will update the user’s password

8. The system will display a confirmation message.

9. The system will send a confirmation e-mail to the user

**5. Alternative flows**

**5.1 Password doesn’t meet the character length requirement**

The system shall display an error message.

The use case resumes at step 3.

**5.2 Password doesn’t match the “minimum one capital, one lowercase, one number” requirement**

The system shall display an error message.

The use case resumes at step 3.

**5.3 Password and Password confirmation don’t match**

The system shall display an error message.

The use case resumes at step 3.

**6. Key Scenarios**

**6.1 Password doesn’t meet the character length requirement**

**6.2 Password doesn’t match the “minimum one capital, one lowercase, one number” requirement**

**6.3 Password and Password confirmation don’t match**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully changed their password.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 3 (Account - Pay course fees)**

**1. Brief description**

This use case describes how the user will use Student Portal to pay their course fees.

**2. Actors**

2.1 User (account type: student)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on Account

3. The user will click/tap on Pay course fees

4. If the user’s address doesn’t exist, the user will need to fill in the address before submitting

5. The system will also displays the amount the user has left to pay

6. The user will have the option to pay half of the fee amount by checking the “Pay half of the fee amount” checkbox

7. The user will click/tap the “Pay” button

8. The system will send data to PayPal

9. The system will redirect to the PayPal payments page.

10. The user will complete the payment on the PayPal page

11. The user will click on the link labelled “Return to test account's Test Store” to return to Student Portal

12. The system will redirect the user to a confirmation page displaying a confirmation message.

13. The system will send a confirmation e-mail to the user.

**5. Alternative flows**

**5.1 Cancel payment**

The system will redirect the user to a confirmation page displaying a confirmation message.

**6. Key Scenarios**

**6.1 Cancel payment**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully paid for their course fees.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 4 (Account - Delete account)**

**1. Brief description**

This use case describes how the user will use Student Portal to delete their account.

**2. Actors**

2.1 User (account type: administrator, student, lecturer)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Account”

3. The user will click/tap on “Delete account”

4. The user will click/tap the “Delete account” button

5. The system will display a prompt asking the user to confirm their action

6. The user will click on “Yes”

7. The system will delete the user’s account.

8. The system will display a confirmation message.

**5. Alternative flows**

None

**6. Key Scenarios**

None

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully deleted their account.

**7.2 Failure condition**

None

**Use case 5 (Account - Administrator - Create an account)**

**1. Brief description**

This use case describes how the user will use Student Portal to create an account.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Account”

3. The user will click/tap on “Create account”

4. The system will load up a “Create account” form

5. The user will fill in the form

6. The user will click on the “Create” button

7. The system will create the account

8. The system will display the message “The new account has been created successfully.”

**5. Alternative flows**

**5.1 Invalid e-mail address**

The system shall display an error message.

The use case resumes at step 3.

**5.2 E-mail address already exists**

The system shall display an error message.

The use case resumes at step 3.

**5.3 Password doesn’t meet the character length requirement**

The system shall display an error message.

The use case resumes at step 3.

**5.4 Password doesn’t match the “minimum one capital, one lowercase, one number” requirement**

The system shall display an error message.

The use case resumes at step 3.

**5.5 Password and Password confirmation don’t match**

The system shall display an error message.

The use case resumes at step 3.

**6. Key Scenarios**

**6.1 E-mail address already exists**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully created an account.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 6 (Account - Administrator - Update an account)**

**1. Brief description**

This use case describes how the administrator will use Student Portal to update an account.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Account”

3. The system will load up a list containing all the users within the system

4. The user will select a user

5. The user will click/tap on Update account

6. The system will load up an “Update account” form

7. The system will load up the existing details in specific labelled textboxes

8. The user will fill in one of the potentially empty textboxes or they will overwrite the already existing details

9. The user will click/tap the “Update” button

10. The system will update the account.

11. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 3.

**5.2 Invalid e-mail address**

The system shall display an error message.

The use case resumes at step 3.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**6.2 Invalid e-mail address**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully edited an account.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 7 (Account - Administrator - Change an account’s password)**

**1. Brief description**

This use case describes how the administrator will use Student Portal to change an account’s password.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Account”

3. The system loads up a list containing all the users within the system

4. The user will select a user

5. The user will click/tap on “Change password”

6. The system will load up a “Change an account’s password” form

7. The user will fill in the form

8. The user will click/tap the “Change password” button

9. The system will change the account’s password

10. The system will display the message “The account’s password has been changed successfully.”

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 3.

**5.2 Password doesn’t meet the character length requirement**

The system shall display an error message.

The use case resumes at step 3.

**5.3 Password doesn’t match the “minimum one capital, one lowercase, one number” requirement**

The system shall display an error message.

The use case resumes at step 3.

**5.4 Password and Password confirmation don’t match**

The system shall display an error message.

The use case resumes at step 3.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**6.2 Password doesn’t meet the character length requirement**

**6.3 Password doesn’t match the “minimum one capital, one lowercase, one number” requirement**

**6.4 Password and Password confirmation don’t match**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully changed an account’s password.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 8 (Account - Administrator - Delete an account)**

**1. Brief description**

This use case describes how the administrator will use Student Portal to delete an account.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an internet browser installed.

The user has an e-mail address.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Account”

3. The system will load up a list containing all the users within the system

4. The user will select a user

5. The user will click/tap on “Delete account”

6. The user will click/tap the “Delete account” button

7. The system will display a prompt asking the user to confirm their action

8. The user will click on “Yes”

9. The system will delete the account.

10. The system will redirect the user to a confirmation page displaying a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 3.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

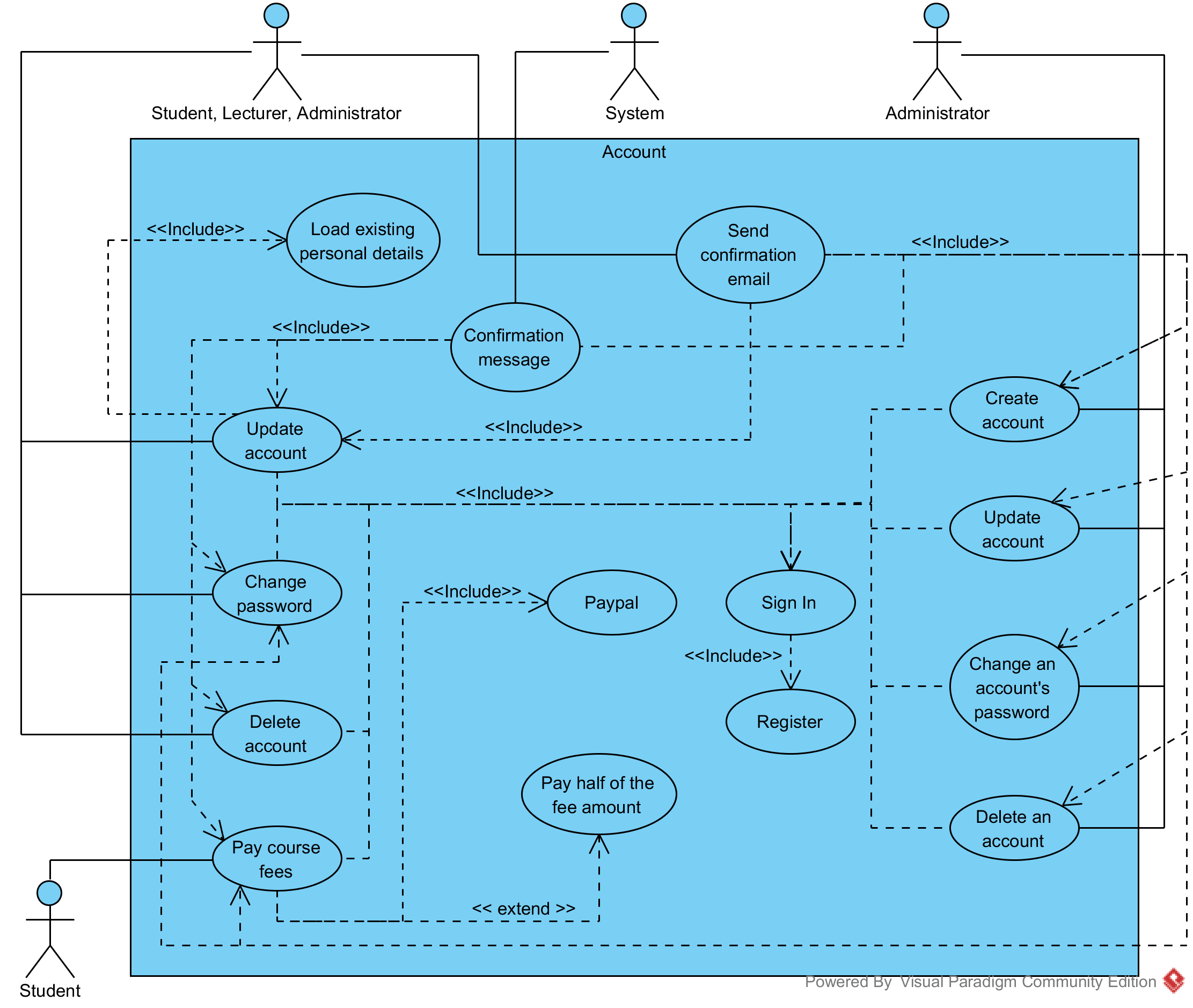
**7. Post-conditions**

**7.1 Successful completion**

The user has successfully deleted an account.

**7.2 Failure condition**

The system displays failure reasons accordingly.



**Calendar**

**Use case 1 (Calendar - Check calendar)**

**1. Brief description**

This use case describes how the user will use Student Portal to check their calendar.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Calendar”

3. The system will load up a calendar displaying already existing tasks

4. The user will check their due tasks

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully checked the calendar.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 2 (Calendar - Create a task)**

**1. Brief description**

This use case describes how the user will use Student Portal to create a task.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Calendar”

3. The user will click/tap the “Create task” button

4. The system will load up a “Create a task” form

5. The user will fill in the task details

6. The user will click/tap the “Create” button

7. The system will create the task

8. The system will display a confirmation message.

**5. Alternative flows**

**5.1 Task name already exists**

The system shall display an error message.

The use case resumes at step 3.

**6. Key Scenarios**

**6.1 Task name already exists**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully created a task.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 2 (Calendar - Check due tasks)**

**1. Brief description**

This use case describes how the user will use Student Portal to check their due tasks

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Calendar”

3. The user will click/tap to expand the “Due tasks” tab

3. The system will display a list of due tasks

4. The user will check their due tasks

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully checked their due tasks.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 3 (Calendar - Update a task)**

**1. Brief description**

This use case describes how the user will use Student Portal to update a task.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Calendar”

3. The system will load up already existing tasks which belong to the user that is currently signed in

4. The user will select a task

5. The user will click/tap the “Update task” button

6. The system will redirect to the “Update task” page and load up an “Update a task” form

7. The user will fill in or overwrite the task details

8. The user will click/tap the “Update” button

9. The system will update the task.

10. The system will display a confirmation message.

**5. Alternative flows**

**5.1 Task name already exists**

The system shall display an error message.

The use case resumes at step 2.

**5.2 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 Task name already exists**

**6.2 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully updated a task.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 4 (Calendar - Complete a task)**

**1. Brief description**

This use case describes how the user will use Student Portal to complete a task.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Calendar”

3. The system will load up already existing tasks which belong to the user that is currently signed in

4. The user will select a task

5. The user will click/tap the “Complete” task button

6. The system will request confirmation.

7. The user will click on Yes.

8. The system will set the task as completed.

9. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully completed a task.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 5 (Calendar - Check completed tasks)**

**1. Brief description**

This use case describes how the user will use Student Portal to check their completed tasks.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Calendar”

3. The user will click/tap to expand the “Completed tasks” tab

3. The system will display a list of completed tasks

4. The user will check their completed tasks

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully checked their completed tasks.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 6 (Calendar - Archive a task)**

**1. Brief description**

This use case describes how the user will use Student Portal to archive a task.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Calendar”

3. The system will load up already existing tasks which belong to the user that is currently signed in

4. The user will select a task

5. The user will click/tap the “Archive” button

6. The system will request confirmation.

7. The user will click on Yes.

8. The system will set the task as inactive.

9. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully archived a task.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 7 (Calendar - Check archived tasks)**

**1. Brief description**

This use case describes how the user will use Student Portal to check their archived tasks.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page.

2. The user will click/tap on “Calendar”.

3. The user will click/tap to expand the “Archived tasks” tab.

3. The system will display a list of archived tasks.

4. The user will check their archived tasks.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully checked their archived tasks.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 6 (Calendar - Restore a task)**

**1. Brief description**

This use case describes how the user will use Student Portal to restore a task.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Calendar”

3. The system will load up already archived tasks which belong to the user that is currently signed in

4. The user will select a task

5. The user will click/tap the “Restore” button.

6. The system will request confirmation.

7. The user will click on Yes.

8. The system will set the task as active.

9. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully restored a task.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 7 (Calendar - Delete a task)**

**1. Brief description**

This use case describes how the user will use Student Portal to delete a task.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Calendar”

3. The system will load up already existing tasks which belong to the user that is currently signed in

4. The user will select a task

5. The user will click/tap the “Delete” button

6. The system will request confirmation.

7. The user will click on Yes.

8. The system will delete the task.

9. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

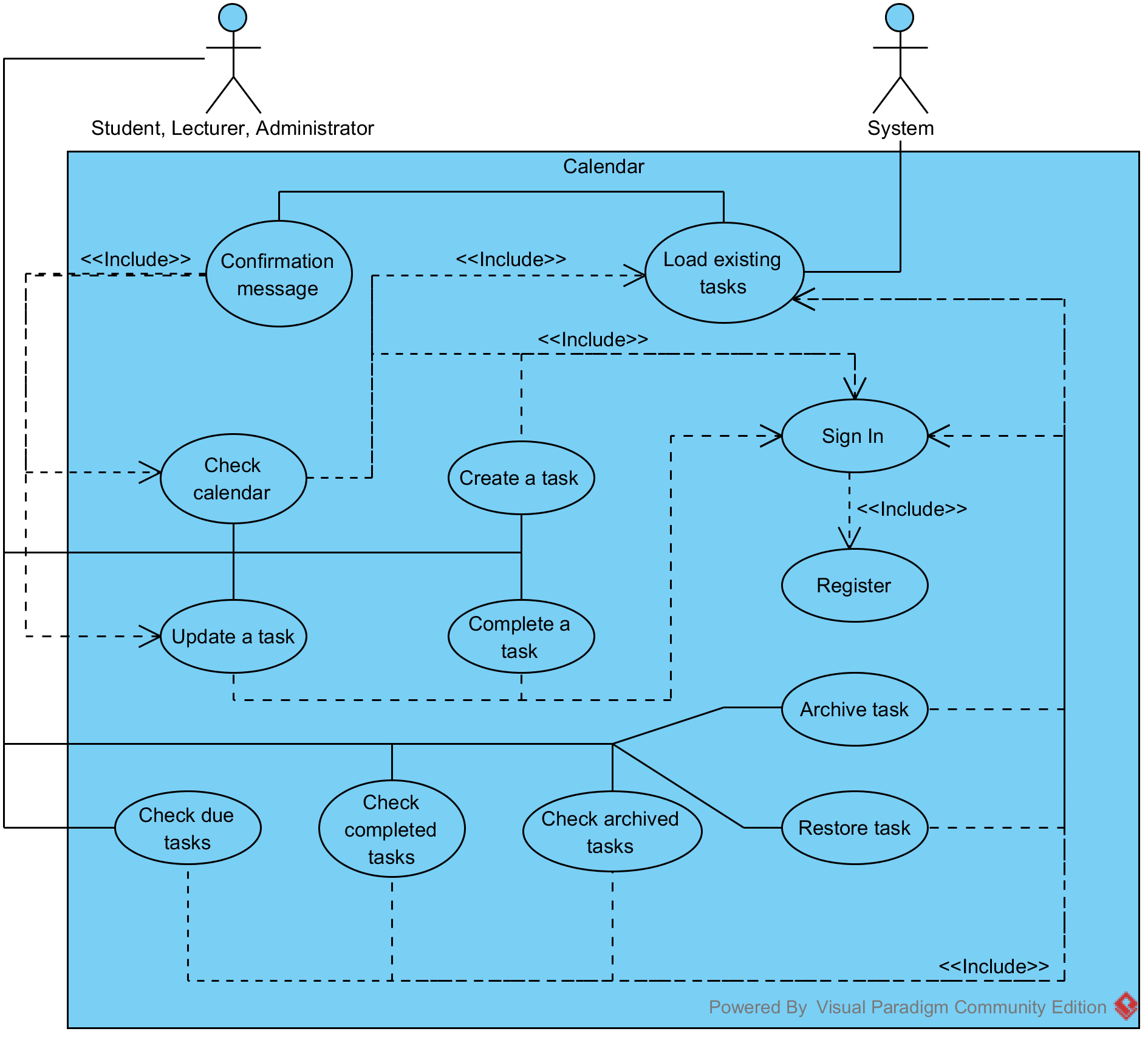
**7. Post-conditions**

**7.1 Successful completion**

The user has successfully deleted a task.

**7.2 Failure condition**

The system displays failure reasons accordingly.



**Events**

**Use case 1 (Events - Check events calendar)**

**1. Brief description**

This use case describes how the user will use Student Portal to check the events calendar.

**2. Actors**

2.1 User (account type: student, lecturer)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Events”

3. The system will load up a calendar displaying already existing events

4. The user will check the upcoming events

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully checked the events calendar.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 2 (Events - Book and Pay for an event)**

**1. Brief description**

This use case describes how the user will use Student Portal to book and pay for an event.

**2. Actors**

2.1 User (account type: student, lecturer)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Events”

3. The system will load up already existing events

4. The user will select an event

5. The user will click/tap the “Book event” button

6. The system will display a “Book an event” form

7. The user will fill in the form

8. The user will clicks/tap the “Book” button

9. The system will send the data to PayPal

10. The system will redirect to the PayPal payments page.

11. The user will complete the payment on the PayPal page

12. The user will click on the link labelled “Return to test account's Test Store” to return to Student Portal

13. The system will redirect the user to a confirmation page displaying a confirmation message.

14. The system will send a confirmation e-mail to the user

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**5.2 Cancel payment**

The system will redirect the user to a confirmation page displaying a confirmation message.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**6.2 Cancel payment**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully booked and paid for an event.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 3 (Events - Check booked events)**

**1. Brief description**

This use case describes how the user will use Student Portal to check their booked events.

**2. Actors**

2.1 User (account type: student, lecturer)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Events”

3. The user will click/tap to expand the “Booked events” tab

4. The system will display a list of booked events

5. The user will check their booked events

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully checked their booked events.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 4 (Events - Administrator - Create an event)**

**1. Brief description**

This use case describes how the user will use Student Portal to create an event.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Events”

3. The user will click/tap on “Create an event”

4. The system will display a “Create an event” form

5. The user will fill in the form

6. The user will clicks/tap the “Create” button

7. The system will check the data submitted and create the event

8. The system will display a confirmation message.

**5. Alternative flows**

None

**6. Key Scenarios**

None

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully created an event.

**7.2 Failure condition**

None

**Use case 5 (Events - Administrator - Update an event)**

**1. Brief description**

This use case describes how the user will use Student Portal to update an event.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Events”

3. The system will load up already existing events

4. The user will select an event

5. The user will click/tap on Update event

6. The system will redirect to the “Update event” page and load up an “Update event” form

7. The system loads up the existing details in specific labelled textboxes

8. The user will fill in one of the potentially empty textboxes or they will overwrite the already existing details

9. The user clicks/taps the “Update” button

10. The system updates the event details.

11. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully updated an event.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 6 (Events - Administrator - Deactivate an event)**

**1. Brief description**

This use case describes how the user will use Student Portal to deactivate an event.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Events”

3. The system will load up already existing events

4. The user will select an event

5. The user will click/tap the “Deactivate” button

6. The system will request confirmation.

7. The user will click on Yes.

8. The system will set the event as inactive.

9. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully deactivated an event.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 6 (Events - Administrator - Reactivate an event)**

**1. Brief description**

This use case describes how the user will use Student Portal to reactivate an event.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Events”

3. The system will load up already existing events

4. The user will select an event

5. The user will click/tap the “Reactivate” book

6. The system will request confirmation.

7. The user will click on Yes.

8. The system will set the event as active.

9. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully reactivated an event.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 20 (Events - Administrator - Delete an event)**

**1. Brief description**

This use case describes how the user will use Student Portal to delete an event.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Events”

3. The system will load up already existing events

4. The user will select an event

5. The user will click/tap the “Delete” button

6. The system will request confirmation.

7. The user will click on Yes.

8. The system will delete the event.

9. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

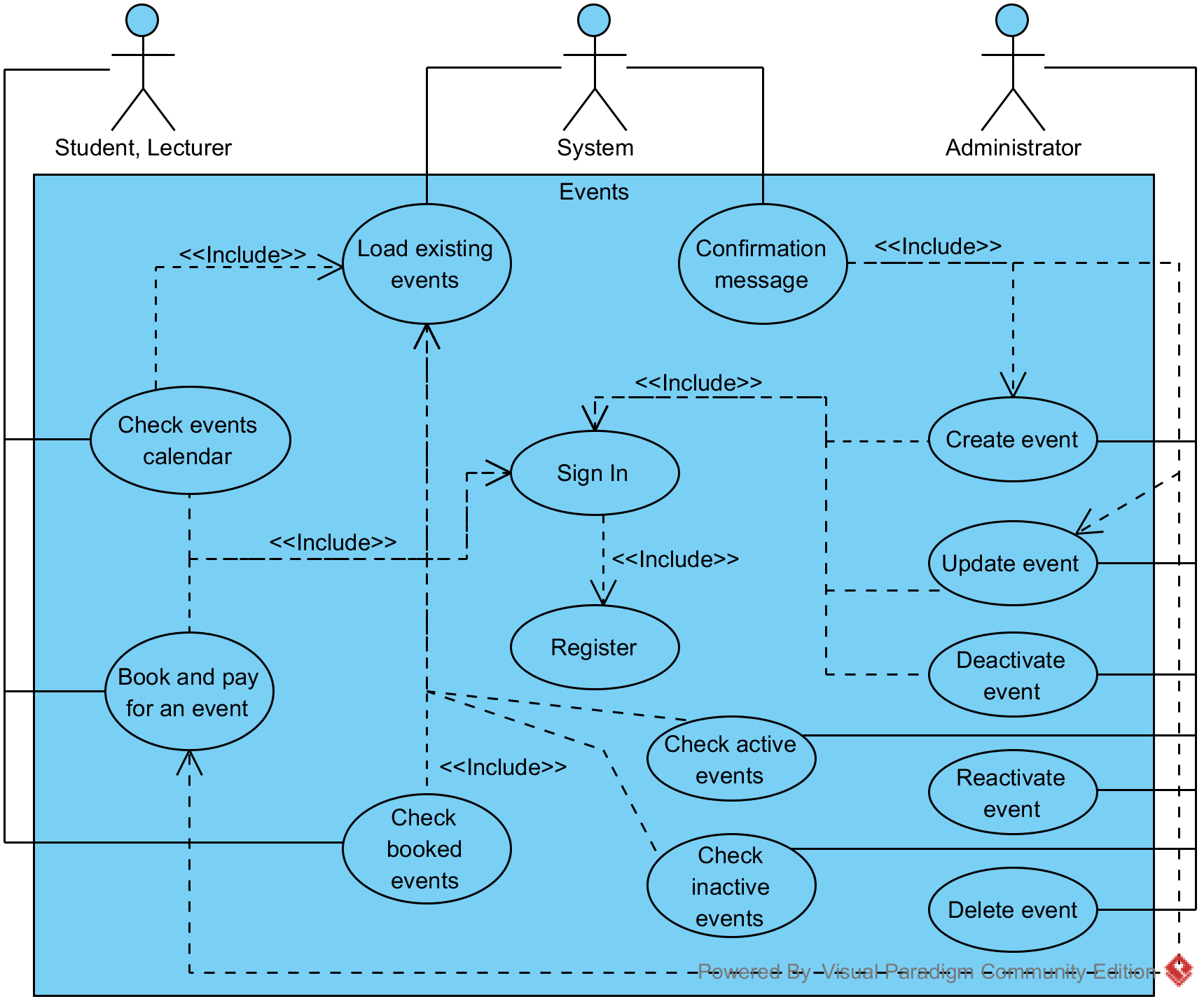
**7. Post-conditions**

**7.1 Successful completion**

The user has successfully deleted an event.

**7.2 Failure condition**

The system displays failure reasons accordingly.



**Use case 21 (Exams - Check exam timetable)**

**1. Brief description**

This use case describes how the user will use Student Portal to check their exam timetable.

**2. Actors**

2.1 User (account type: student)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Exams”

3. The user will click/tap on “Exam timetable”

4. The system will display a list of exams that the student is due to attend

5. The user will check the list to see more detailed information about their upcoming exams.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully checked the exam timetable.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 22 (Exams - Check exam results)**

**1. Brief description**

This use case describes how the user will use Student Portal to check exam results.

**2. Actors**

2.1 User (account type: student)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Exams”

3. The user will click/tap on “Exam results”

4. The system will display a list of exams that the user has already taken

5. The user will expand to see more detailed information about their exam results.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully checked their exam results.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 23 (Exams - Administrator - Create an exam)**

**1. Brief description**

This use case describes how the user will use Student Portal to create an exam.

**2. Actors**

2.1 User (account type: student)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Exams”

3. The user will click/tap on “Create an exam”

4. The system will load up a “Create an exam” form

5. The user will fill in the form

6. The user will click on the “Create” button

7. The system will create the exam

8. The system will display a confirmation message.

**5. Alternative flows**

None

**6. Key Scenarios**

None

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully added an exam.

**7.2 Failure condition**

None

**Use case 24 (Exams - Administrator - Update an exam)**

**1. Brief description**

This use case describes how the user will use Student Portal to update an exam.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Exams”

3. The system will load up already existing exams

4. The user will select an exam

5. The user will click/tap the “Update” button

6. The system will update the exam

7. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 3.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully updated an exam.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 25 (Exams - Administrator - Delete an exam)**

**1. Brief description**

This use case describes how the user will use Student Portal to delete an exam.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Exams”

4. The system will load up already existing exams

5. The user will select an exam

6. The user will click/tap the “Delete” button

7. The system will delete the exam

8. The system will display a confirmation message

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 5.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

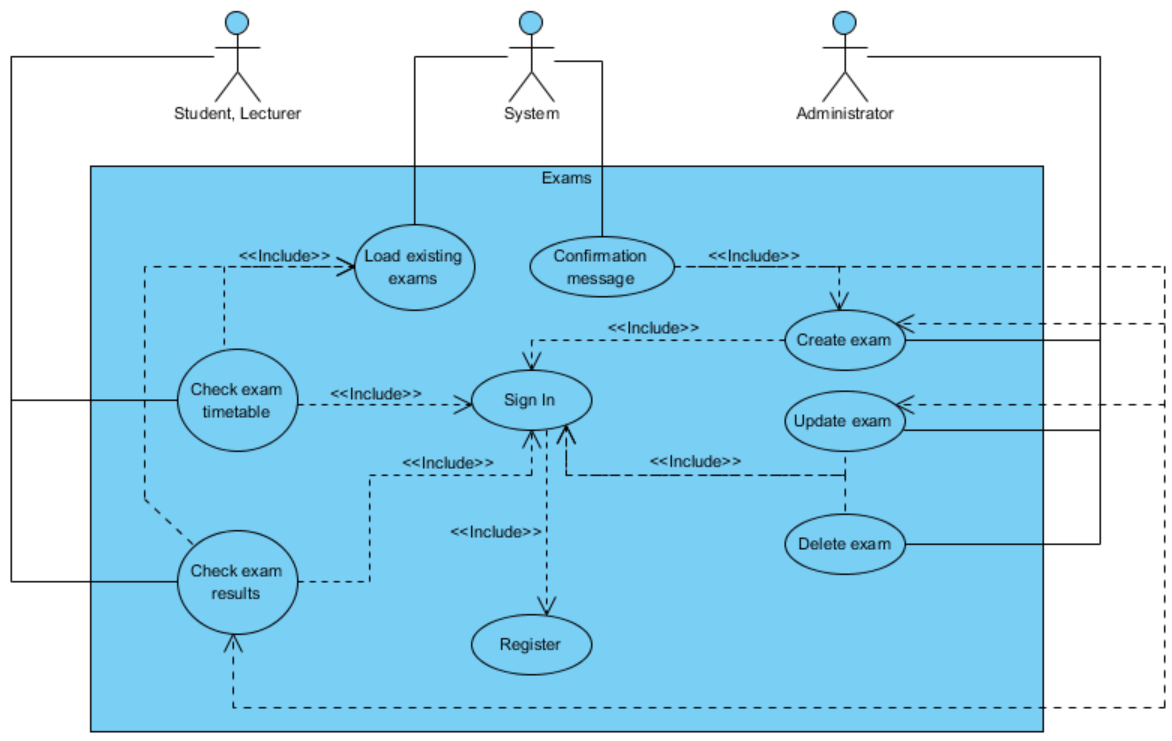
**7. Post-conditions**

**7.1 Successful completion**

The user has successfully deleted an exam.

**7.2 Failure condition**

The system displays failure reasons accordingly.



**Use case 27 (Feedback - Submit feedback on a lecture or tutorial)**

**1. Brief description**

This use case describes how the user will use Student Portal to submit feedback on a lecture or tutorial.

**2. Actors**

2.1 User (account type: student)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Feedback”

3. The system will load up lectures (that the currently signed in user undertakes)

4. The user will make a choice of lecture and click/tap on the “Submit feedback” button

5. The system will load up a “Submit lecture/tutorial feedback” form

6. The user will fill in the form

7. The system will submit the feedback for moderation

8. The system will send a copy of the feedback to the student

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully submitted feedback on a lecture or tutorial.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 28 (Feedback - View feedback on a lecture or tutorial)**

**1. Brief description**

This use case describes how the user will use Student Portal to view feedback on a lecture or tutorial.

**2. Actors**

2.1 User (account type: lecturer)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Feedback”

3. The system will load up lectures (that the currently signed in user is in charge of)

4. The user will make a choice of lecture and click/tap on the “View feedback” button

5. The system will load up any potential feedback

6. If there is no feedback, the system will display a confirmation message.

7. If there is feedback, the user will be able to view it and contact the user who submitted the feedback.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully viewed the feedback submitted on a lecture or tutorial.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 29 (Feedback - Administrator - Approve feedback)**

**1. Brief description**

This use case describes how the user will use Student Portal to approve submitted feedback.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Feedback”

3. The system will load up submitted feedback which is pending approval

4. The user will review the feedback

5. The user will click/tap the “Approve” button

6. The system will set the feedback as approved

7. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

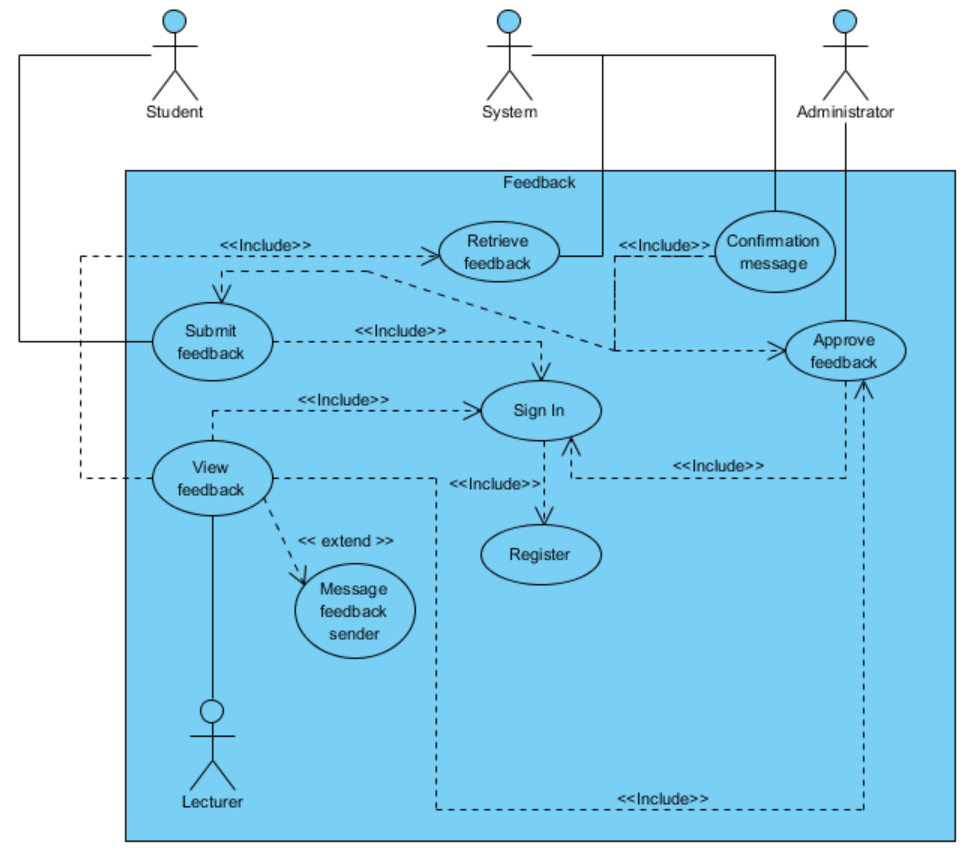
**7. Post-conditions**

**7.1 Successful completion**

The user has successfully approved the feedback.

**7.2 Failure condition**

The system displays failure reasons accordingly.



**Use case 30 (Library - Browse the book catalogue)**

**1. Brief description**

This use case describes how the user will use Student Portal to browse the library book catalogue.

**2. Actors**

2.1 User (account type: student)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system loads the “Overview” page

2. The user clicks/taps on “Library”

3. The user will click/tap on “Browse books”

4. The system will load up books (specifying the title, author, year, copies, format, availability etc.)

5. The user will browse through the books.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully browsed the book catalogue.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 31 (Library – Reserve a book)**

**1. Brief description**

This use case describes how the user will use Student Portal to browse the library book catalogue.

**2. Actors**

2.1 User (account type: student)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system loads the “Overview” page

2. The user clicks/taps on “Library”

3. The system will display a list of books (specifying the title, author, year, copies, format, availability etc.)

4. The table will have a search function where the user will be able to search for a book

5. The user will select a book

6. The user will click/tap the “Reserve” button

7. The system will display a “Reserve a book” form

8. The user will fill in the form

9. The system will reserve the book for the user and will update its availability

10. The system will display a confirmation message.

11. The system will send an e-mail confirmation to the student.

12. The system will send an e-mail confirmation to the library staff.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully reserved a book.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 32 (Library - Administrator - Create a book)**

**1. Brief description**

This use case describes how the user will use Student Portal to create a book.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Library”

3. The user will click/tap on “Create a book”

4. The system will load up a “Create a book” form

5. The user will fill in the form

6. The user will click on the “Create” button

7. The system will create the book

8. The system will display a confirmation message.

**5. Alternative flows**

None

**6. Key Scenarios**

None

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully created a book.

**7.2 Failure condition**

None

**Use case 33 (Library - Administrator - Update a book)**

**1. Brief description**

This use case describes how the user will use Student Portal to update a book.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Library”

3. The user will click/tap on “Update a book”

4. The system will load up a list of books that already exist in the system

5. The user will select a book

6. The user will click/tap the “Update” button

7. The system will update the book

8. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 5.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully updated a book.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 34 (Library - Administrator - Delete a book)**

**1. Brief description**

This use case describes how the user will use Student Portal to delete a book.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Library”

3. The system will load already existing books

4. The user will select a book

5. The user will click/tap the “Delete” button

6. The system will delete the book

7. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

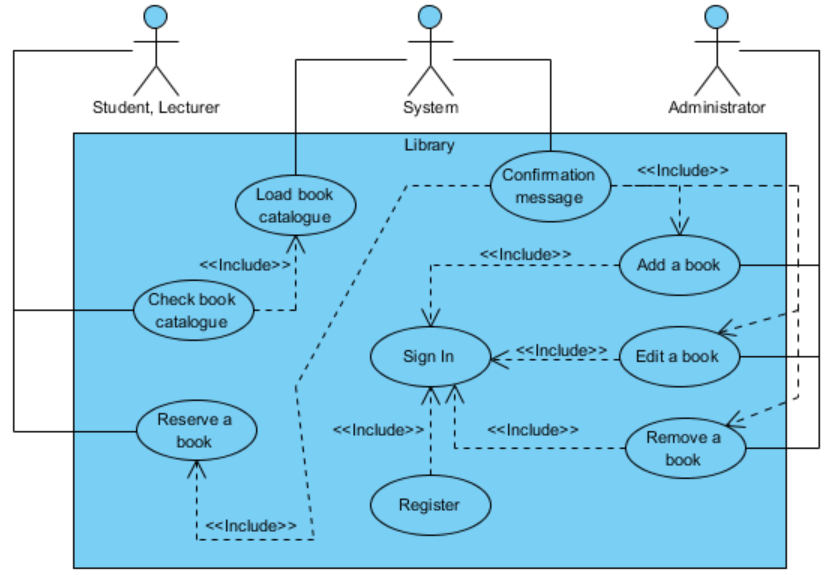
**7. Post-conditions**

**7.1 Successful completion**

The user has successfully deleted a book.

**7.2 Failure condition**

The system displays failure reasons accordingly.



**Use case 35 (Messenger - send a message to another user)**

**1. Brief description**

This use case describes how the user will use Student Portal to send a message to another user.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Messenger”

3. The system will load up a list of users

4. The user will make a choice of user and click/tap the “Message” button

5. The system will load up a “Message another user” form

6. The user will fill in the form

7. The system will submit the message

8. The system will display a confirmation message.

9. The system will send a copy of the e-mail to the sender

10. The system will send the message through an e-mail to the receiver

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 5.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has successfully sent a message to another user.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 36 (Messenger - Delete a message)**

**1. Brief description**

This use case describes how the user will use Student Portal to delete a message.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Messenger”

3. The system will load up a list of messages sent or received by the user

4. The user will select a message

5. The user will click/tap the “Delete” button

6. The system will delete the message

7. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 5.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

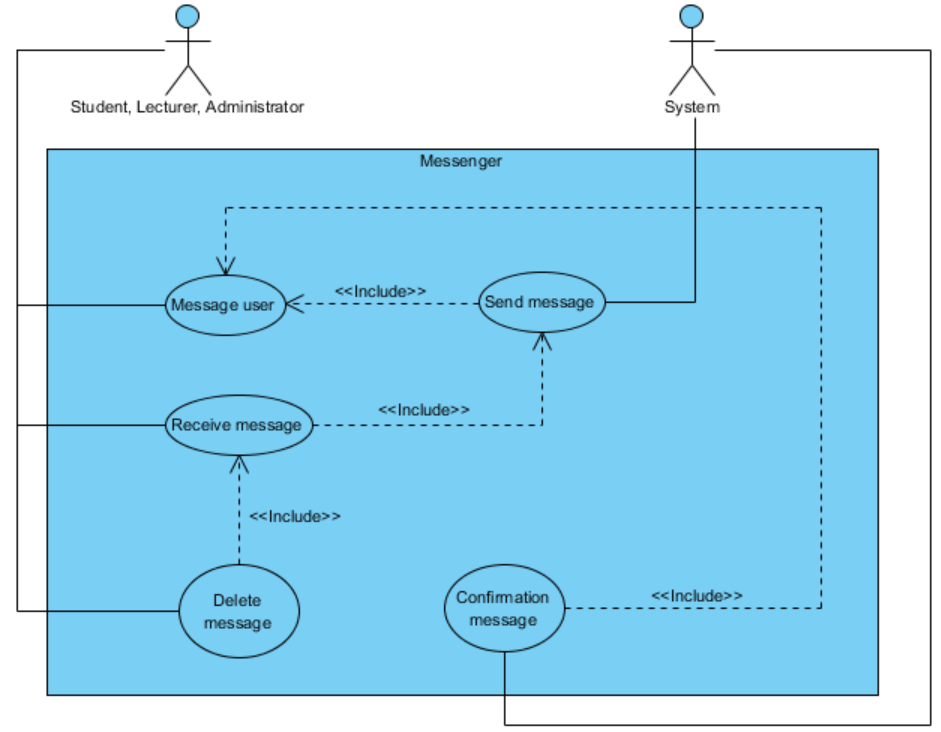
**7. Post-conditions**

**7.1 Successful completion**

The user has successfully deleted a message.

**7.2 Failure condition**

The system displays failure reasons accordingly.



**Use case 37 (Timetable - Check timetable)**

**1. Brief description**

This use case describes how the user will use Student Portal to check their university timetable.

**2. Actors**

2.1 User (account type: student)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system loads the “Overview” page

2. The user clicks/taps on “Timetable”

3. The system will load up a timetable that belongs to the currently signed in user

4. The user will check to see more detailed information about their upcoming lectures and tutorials. The user will be able to click on the “Moodle” button which will load up that specific module’s Moodle page in a new tab within the web browser

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has checked their timetable successfully.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 38 (Timetable - Administrator - Create a module/lecture/tutorial)**

**1. Brief description**

This use case describes how the user will use Student Portal to create a module/lecture/tutorial.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Timetable”

3. The user will click/tap on “Create a module/lecture/tutorial”

4. The system will load up a “Create a module/lecture/tutorial” form

5. The user will fill in the form

6. The user will click on the “Create” button

7. The system will create the module/lecture/tutorial

8. The system will refresh and display a confirmation message.

**5. Alternative flows**

None

**6. Key Scenarios**

None

**7. Post-conditions**

**7.1 Successful completion**

The user has created a module/lecture/tutorial successfully.

**7.2 Failure condition**

None

**Use case 39 (Timetable - Administrator - Update a module/lecture/tutorial)**

**1. Brief description**

This use case describes how the user will use Student Portal to update a module/lecture/tutorial.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Timetable”

3. The system will load up already existing modules/lecture/tutorial

4. The user will select a module/lecture/tutorial

5. The user will click/tap the “Update module/lecture/tutorial” button

6. The system loads up the existing details in specific labelled textboxes

7. The user will fill in one of the potentially empty textboxes or they will overwrite the already existing details

8. The user clicks/taps the “Update” button

9. The system will update the module/lecture/tutorial

10. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message.

The use case resumes at step 5.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

**7. Post-conditions**

**7.1 Successful completion**

The user has updated a module/lecture/tutorial successfully.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 40 (Timetable - Administrator - Delete a module/lecture/tutorial)**

**1. Brief description**

This use case describes how the user will use Student Portal to delete a module/lecture/tutorial.

**2. Actors**

2.1 User (account type: administrator)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Timetable”

3. The system will load up already existing modules/lecture/tutorial

4. The user will select a module/lecture/tutorial

5. The user will click/tap the “Delete module/lecture/tutorial” button

6. The system will delete the module/lecture/tutorial

7. The system will display a confirmation message.

**5. Alternative flows**

**5.1 System fails to retrieve data**

The system shall display an error message

The use case resumes at step 5.

**6. Key Scenarios**

**6.1 System fails to retrieve data**

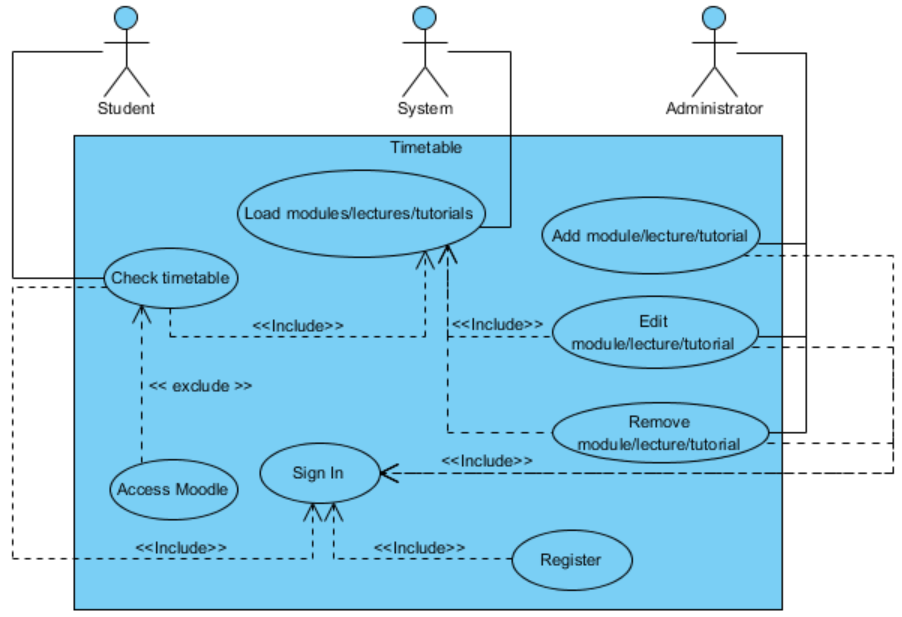
**7. Post-conditions**

**7.1 Successful completion**

The user has deleted a module/lecture/tutorial successfully.

**7.2 Failure condition**

The system displays failure reasons accordingly.



**Use case 41 (Transport - Check transport status)**

**1. Brief description**

This use case describes how the user will use Student Portal to check transport status. This includes Live and Weekend updates of Tube lines and stations and Cycle Hire updates.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “Transport”

3. The system will display a list or “wall” of panels. The first four panels will be for the user’s choice, as they will be labelled “Tube - Now”, “Tube - This Weekend”, “Tube Map” and “Cycle Hire – Availability Updates”. The other panels represent each line within the London Tube network and its current status at the point of when the user will access it.

4. The user will select an option.

5. The system will load the user’s choice.

**5. Alternative flows**

**5.1 System fails to retrieve data from TFL (Transport for London)**

The system shall display the message an error message.

The use case resumes at step 2.

**6. Key Scenarios**

**6.1 System fails to retrieve data from TFL (Transport for London)**

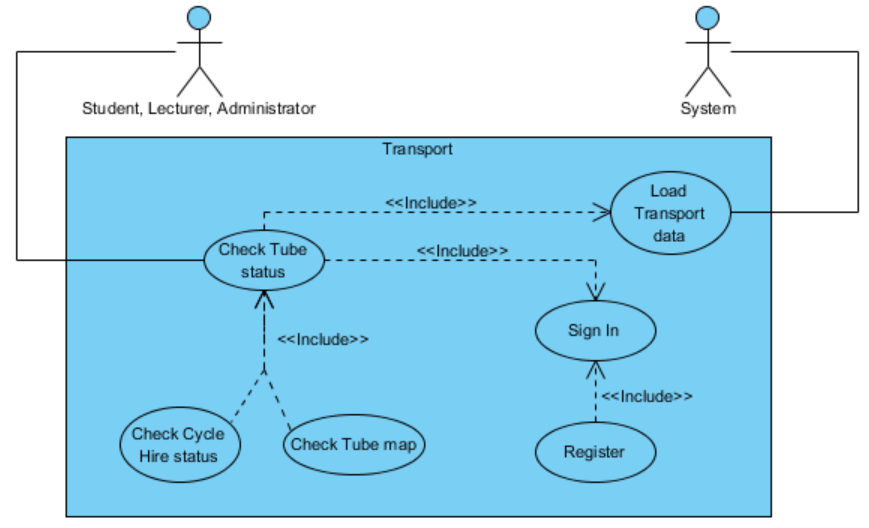
**7. Post-conditions**

**7.1 Successful completion**

The user has checked the transport status successfully.

**7.2 Failure condition**

The system displays failure reasons accordingly.



**Use case 42 (University Map - Check the university map)**

**1. Brief description**

This use case describes how the user will use Student Portal to book and pay for an event.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

The user has signed into the system.

**4. Basic flow of events**

1. The system will load the “Overview” page

2. The user will click/tap on “University Map”

3. The system will load up a map containing map markers for a number of university buildings and libraries

4. The user will be able to click on the map markers to get more information on the lecture theatres, computer rooms and student services that are within the university buildings

**5. Alternative flows**

**5.1 Map load failure**

The system shall display the message an error message.

The use case resumes at step 2.

**6. Key Scenarios**

None

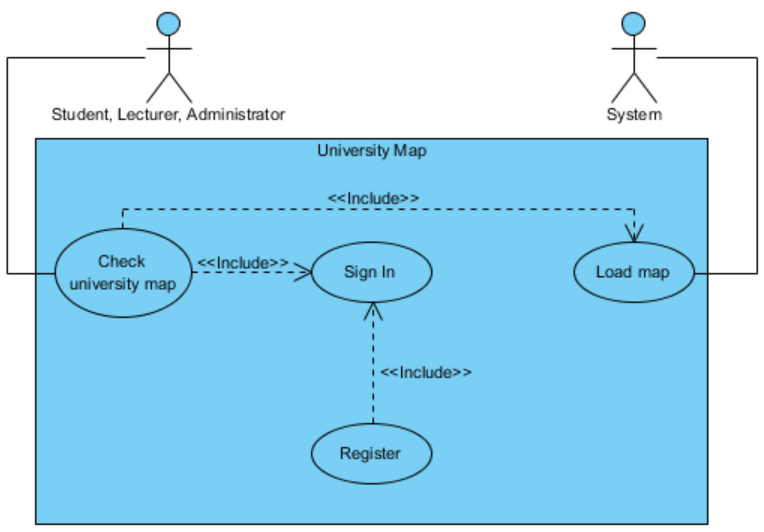
**7. Post-conditions**

**7.1 Successful completion**

The user has successfully checked the university map.

**7.2 Failure condition**

None



**Use case 43 (Home page - Sign In)**

**1. Brief description**

This use case describes how the user will use Student Portal to sign in.

**2. Actors**

2.1 User (any type)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

The user has an account.

**4. Basic flow of events**

1. The user will launch an internet browser

2. The user will type in the URL of Student Portal in the address bar

3. The system will display a “Sign In” form

4. The user will fill in the form

5. The user will click/tap the Sign In button

6. The system will authenticate the user

7. The system will redirect and display the Overview page

**5. Alternative flows**

**5.1 Invalid e-mail address**

The system shall display an error message

The use case resumes at step 3.

**5.1 Invalid password**

The system shall display an error message

The use case resumes at step 3.

**6. Key Scenarios**

**6.1 Invalid e-mail address**

**6.1 Invalid password**

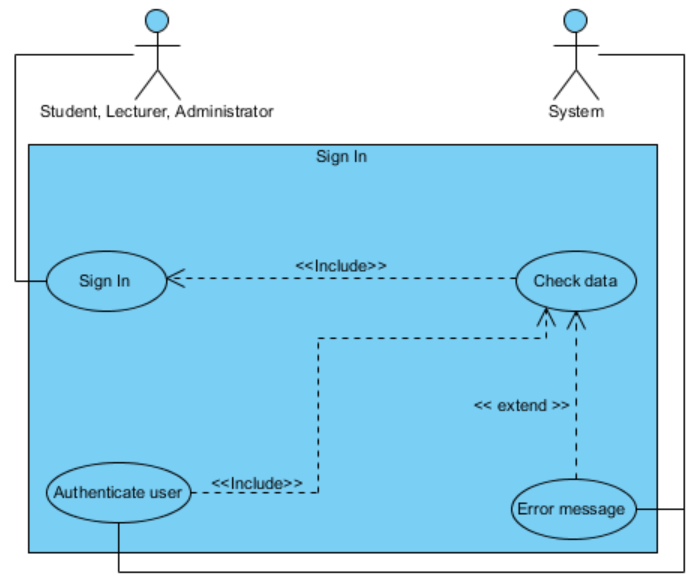
**7. Post-conditions**

**7.1 Successful completion**

The user has signed in successfully.

**7.2 Failure condition**

The system displays failure reasons accordingly.



**Use case 44 (Register page - Register)**

**1. Brief description**

This use case describes how the user will use Student Portal to register.

**2. Actors**

**2.1 User** (account type: student only)

**2.2 System**

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

**4. Basic flow of events**

1. The user will launch an internet browser

2. The user will type in the URL of Student Portal in the address bar

3. The system will display a “Register” form

4. The user will fill in the form

5. The user will click/tap the Register button

6. The system will display a confirmation message.

**5. Alternative flows**

**5.1 Invalid e-mail address**

The system shall display an error message.

The use case resumes at step 3.

**5.2 E-mail address already exists**

The system shall display the message “A user with this email address already exists.”

The use case resumes at step 3.

**5.3 Password doesn’t meet the character length requirement**

The system shall display an error message.

The use case resumes at step 3.

**5.4 Password doesn’t match the “minimum one capital, one lowercase, one number” requirement**

The system shall display an error message.

The use case resumes at step 3.

**5.5 Password and Password confirmation don’t match**

The system shall display an error message.

The use case resumes at step 3.

**6. Key Scenarios**

**6.1 E-mail address already exists**

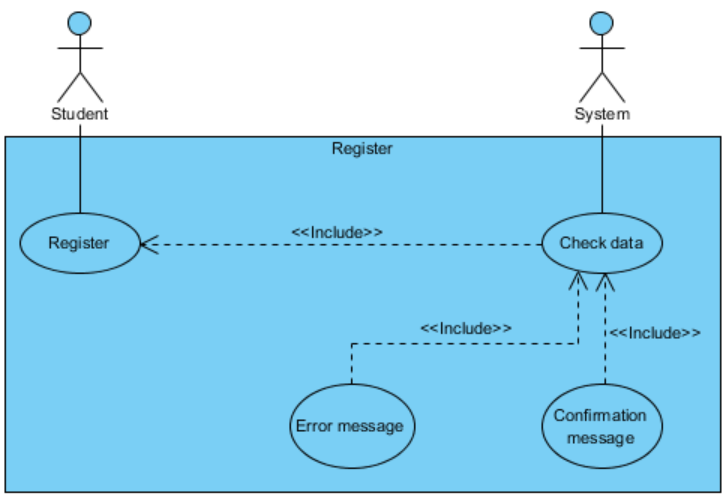
**7. Post-conditions**

**7.1 Successful completion**

The user is registered successfully.

**7.2 Failure conditions**

The system displays failure reasons accordingly.



**Use case 45 (Forgotten Password - Reset password - part 1)**

**1. Brief description**

This use case describes how the user will use Student Portal to reset their password that they have forgotten.

**2. Actors**

2.1 User (account type: any)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has an e-mail address.

The user has an internet browser installed.

**4. Basic flow of events**

1. The user will launch an internet browser

2. The user will type in the URL of Student Portal in the address bar

3. The system will display a “Sign In” form

4. The user will click/tap on the link labelled “Forgotten your password?”

5. The system will load the “Forgotten your password” page

6. The system will display a “Forgotten your password” form

7. The user will type in their e-mail address in the single textbox

8. The user will click/tap the Continue button.

9. The system will send an e-mail to the e-mail address provided containing a URL that directs the user to the “Password Reset” page.

10. The system will display a confirmation message.

**5. Alternative flows**

**5.1 E-mail address doesn’t exist**

The system shall display an error message.

The use case resumes at step 3.

**5.2 Invalid e-mail address**

The system shall display an error message.

The use case resumes at step 3.

**6. Key Scenarios**

**6.1 Invalid e-mail address or password**

**7. Post-conditions**

**7.1 Successful completion**

The user has passed the e-mail verification and has received an e-mail with password reset instructions successfully.

**7.2 Failure condition**

The system displays failure reasons accordingly.

**Use case 46 (Password Reset - Reset password - part 2)**

**1. Brief description**

This use case describes how the user will use Student Portal to reset their password that they have forgotten.

**2. Actors**

2.1 User (any type)

2.2 System

**3. Preconditions**

The user has internet connection.

The user has e-mail address.

The user has an internet browser installed.

**4. Basic flow of events**

1. The user will check their e-mail

2. The user will open the e-mail sent by Student Portal

3. The user will click on the link contained within the e-mail

4. The user’s installed internet browser will launch and load the “Password Reset” page

5. The system will load up a “Password Reset” form

6. The user will fill in the form

7. The user will click/tap the Continue button.

8. The system will refresh and display a confirmation message

9. The system will send a confirmation e-mail to the user.

**5. Alternative flows**

**5.1 E-mail address doesn’t exist**

The system shall display an error message.

The use case resumes at step 5.

**5.2 Invalid e-mail address**

The system shall display an error message.

The use case resumes at step 5.

**6. Key Scenarios**

**6.1 Invalid e-mail address or password**

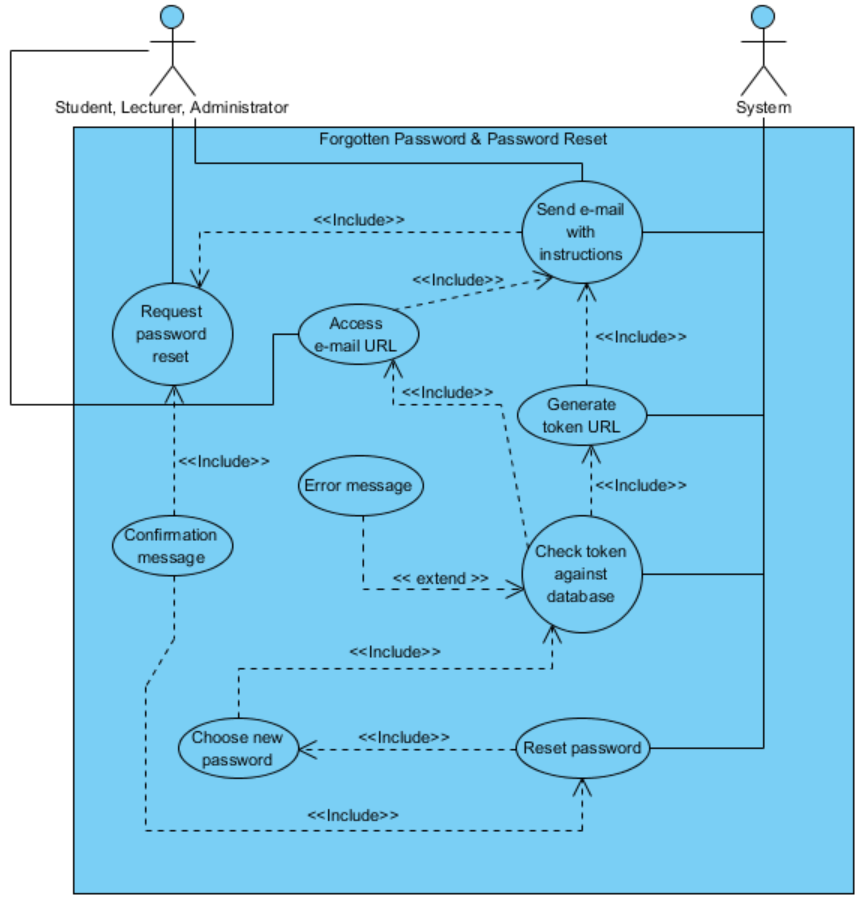
**7. Post-conditions**

**7.1 Successful completion**

The user has reset their password to a new one, and is able to login using the new password successfully.

**7.2 Failure condition**

The system displays failure reasons accordingly.



**Database scripts**

**DROP TABLE** user\_feedback\_lookup;  
**DROP TABLE** user\_feedback;  
**DROP TABLE** system\_lectures;  
**DROP TABLE** system\_tutorials;  
**DROP TABLE** system\_exams;  
**DROP TABLE** user\_timetable;  
**DROP TABLE** system\_modules;  
**DROP TABLE** reserved\_books;  
**DROP TABLE** system\_books;  
**DROP TABLE** user\_tasks;  
**DROP TABLE** booked\_events;  
**DROP TABLE** system\_events;  
**DROP TABLE** system\_map\_markers;  
**DROP TABLE** user\_messages\_lookup;  
**DROP TABLE** user\_messages;  
**DROP TABLE** paypal\_log;  
**DROP TABLE** user\_fees;   
**DROP TABLE** user\_token;  
**DROP TABLE** user\_details;  
**DROP TABLE** user\_signin;

**DELETE Statements**

CREATE TABLE `student\_portal`.`user\_signin` (  
 `userid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE PRIMARY KEY,  
 `account\_type` VARCHAR(8) NOT NULL,  
 `email` VARCHAR(300) NOT NULL UNIQUE,  
 `password` CHAR(70) NOT NULL UNIQUE,  
 `isSignedIn` TINYINT(1) NOT NULL,  
 `created\_on` DATETIME NOT NULL,  
 `updated\_on` DATETIME  
) ENGINE = InnoDB;

**user\_signin**

CREATE TABLE `student\_portal`.`user\_details` (  
 `userid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE,  
 `firstname` VARCHAR(70) NOT NULL,  
 `surname` VARCHAR(70) NOT NULL,  
 `gender` VARCHAR(6) NOT NULL,  
 `nationality` VARCHAR(70),  
 `studentno` INT(9) NOT NULL UNIQUE,  
 `degree` VARCHAR(70),  
 `dateofbirth` DATE,  
 `phonenumber` VARCHAR(70),  
 `address1` VARCHAR(70),  
 `address2` VARCHAR(70),  
 `town` VARCHAR(70),  
 `city` VARCHAR(70),  
 `country` VARCHAR(70),  
 `postcode` VARCHAR(70),  
 `created\_on` DATETIME NOT NULL,  
 `updated\_on` DATETIME,  
FOREIGN KEY (userid)  
REFERENCES user\_signin(userid)  
ON UPDATE CASCADE  
ON DELETE CASCADE  
) ENGINE = InnoDB;

**user\_details**

CREATE TABLE `student\_portal`.`user\_token` (

`userid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE,

`token` CHAR(70) UNIQUE,

`created\_on` DATETIME,

FOREIGN KEY (userid)

REFERENCES user\_signin(userid)

ON UPDATE CASCADE

ON DELETE CASCADE

) ENGINE = InnoDB;

**user\_token**

CREATE TABLE `student\_portal`.`user\_fees` (

`userid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE,

`fee\_amount` NUMERIC(15,2) NOT NULL,

`created\_on` DATETIME NOT NULL,

`updated\_on` DATETIME,

FOREIGN KEY (userid)

REFERENCES user\_signin(userid)

ON UPDATE CASCADE

ON DELETE CASCADE

) ENGINE = InnoDB;

**user\_fees**

CREATE TABLE `student\_portal`.`paypal\_log` (  
 `userid` INT(11) NOT NULL,  
 `payment\_id` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE,  
 `invoice\_id` BIGINT(10) NOT NULL UNIQUE,  
 `transaction\_id` VARCHAR(17) NOT NULL UNIQUE,  
 `product\_id` INT(1) NOT NULL UNIQUE,  
 `product\_name` VARCHAR(70) NOT NULL,  
 `product\_quantity` INT(11) NOT NULL,  
 `product\_amount` NUMERIC(15,2) NOT NULL,  
 `payer\_firstname` VARCHAR(70) NOT NULL,  
 `payer\_surname` VARCHAR(70) NOT NULL,  
 `payer\_email` VARCHAR(300) NOT NULL,  
 `payer\_phonenumber` VARCHAR(70),  
 `payer\_address1` VARCHAR(70) NOT NULL,  
 `payer\_address2` VARCHAR(70),  
 `payer\_town` VARCHAR(70),  
 `payer\_city` VARCHAR(70) NOT NULL,  
 `payer\_country` VARCHAR(70) NOT NULL,  
 `payer\_postcode` VARCHAR(70) NOT NULL,  
 `payment\_type` VARCHAR(5) NOT NULL,  
 `payment\_status` VARCHAR(9) NOT NULL,  
 `created\_on` DATETIME NOT NULL,  
 `updated\_on` DATETIME,  
 `completed\_on` DATETIME,  
 `cancelled\_on` DATETIME,  
FOREIGN KEY (userid)  
REFERENCES user\_signin(userid)  
ON UPDATE CASCADE  
ON DELETE CASCADE  
) ENGINE = InnoDB;

**paypal\_log**

CREATE TABLE `student\_portal`.`user\_messages` (

`messageid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE PRIMARY KEY,

`message\_subject` VARCHAR(300) NOT NULL,

`message\_body` VARCHAR(5000),

`created\_on` DATETIME NOT NULL

) ENGINE = InnoDB;

**user\_messages**

CREATE TABLE `student\_portal`.`user\_messages\_lookup` (

`messageid` INT(11) NOT NULL AUTO\_INCREMENT,

`message\_from` INT(11) NOT NULL,

`message\_to` INT(11) NOT NULL,

`isRead` TINYINT(1) NOT NULL,

FOREIGN KEY (messageid)

REFERENCES user\_messages(messageid),

FOREIGN KEY (message\_from)

REFERENCES user\_signin(userid),

FOREIGN KEY (message\_to)

REFERENCES user\_signin(userid)

ON UPDATE CASCADE

ON DELETE CASCADE

) ENGINE = InnoDB;

**user\_messages\_lookup**

CREATE TABLE `student\_portal`.`system\_events` (

`eventid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE PRIMARY KEY,

`event\_name` VARCHAR(300) NOT NULL,

`event\_notes` VARCHAR(5000),

`event\_url` VARCHAR(70),

`event\_class` VARCHAR(15) NOT NULL,

`event\_from` DATETIME NOT NULL,

`event\_to` DATETIME NOT NULL,

`event\_amount` NUMERIC(15,2) NOT NULL,

`event\_ticket\_no` INT(11) NOT NULL,

`event\_category` VARCHAR(10) NOT NULL,

`event\_status` VARCHAR(10) NOT NULL,

`created\_on` DATETIME NOT NULL,

`updated\_on` DATETIME

) ENGINE = InnoDB;

**system\_events**

CREATE TABLE `student\_portal`.`booked\_events` (

`eventid` INT(11) NOT NULL,

`userid` INT(11) NOT NULL,

`event\_name` VARCHAR(300) NOT NULL,

`event\_amount` NUMERIC(15,2) NOT NULL,

`tickets\_quantity` VARCHAR(11) NOT NULL,

`booked\_on` DATETIME NOT NULL,

FOREIGN KEY (userid)

REFERENCES user\_signin(userid),

FOREIGN KEY (eventid)

REFERENCES system\_events(eventid)

ON UPDATE CASCADE

ON DELETE CASCADE

) ENGINE = InnoDB;

**booked\_events**

CREATE TABLE `student\_portal`.`system\_books` (

`bookid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE PRIMARY KEY,

`book\_name` VARCHAR(300) NOT NULL,

`book\_author` VARCHAR(300) NOT NULL,

`book\_notes` VARCHAR(5000),

`book\_copy\_no` INT(11) NOT NULL,

`book\_status` VARCHAR(9) NOT NULL,

`created\_on` DATETIME NOT NULL,

`updated\_on` DATETIME

) ENGINE = InnoDB;

**system\_books**

CREATE TABLE `student\_portal`.`reserved\_books` (

`bookid` INT(11) NOT NULL,

`userid` INT(11) NOT NULL,

`book\_class` VARCHAR(15) NOT NULL,

`reserved\_on` DATE NOT NULL,

`toreturn\_on` DATE NOT NULL,

`returned\_on` DATE NOT NULL,

`isReturned` TINYINT(1) NOT NULL,

FOREIGN KEY (bookid) REFERENCES system\_books(bookid),

FOREIGN KEY (userid) REFERENCES user\_signin(userid)

ON UPDATE CASCADE

ON DELETE CASCADE

) ENGINE = InnoDB;

**reserved\_books**

CREATE TABLE `student\_portal`.`user\_tasks` (

`userid` INT(11) NOT NULL,

`taskid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE PRIMARY KEY,

`task\_name` VARCHAR(70) NOT NULL,

`task\_notes` VARCHAR(5000),

`task\_url` VARCHAR(300),

`task\_class` VARCHAR(15) NOT NULL,

`task\_startdate` DATETIME NOT NULL,

`task\_duedate` DATETIME NOT NULL,

`task\_category` VARCHAR(10) NOT NULL,

`task\_status` VARCHAR(10) NOT NULL,

`created\_on` DATETIME NOT NULL,

`updated\_on` DATETIME,

`completed\_on` DATETIME,

FOREIGN KEY (userid)

REFERENCES user\_signin(userid)

ON UPDATE CASCADE

ON DELETE CASCADE

) ENGINE = InnoDB;

**user\_tasks**

CREATE TABLE `student\_portal`.`system\_modules` (

`moduleid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE PRIMARY KEY,

`module\_name` VARCHAR(300) NOT NULL,

`module\_notes` VARCHAR(5000),

`module\_url` VARCHAR(70),

`module\_status` VARCHAR(10) NOT NULL,

`created\_on` DATETIME NOT NULL,

`updated\_on` DATETIME

) ENGINE = InnoDB;

**system\_modules**

CREATE TABLE `student\_portal`.`user\_timetable` (

`userid` INT(11) NOT NULL,

`moduleid` INT(11) NOT NULL,

FOREIGN KEY (userid)

REFERENCES user\_signin(userid),

FOREIGN KEY (moduleid)

REFERENCES system\_modules(moduleid)

ON UPDATE CASCADE

ON DELETE CASCADE

) ENGINE = InnoDB;

**user\_timetable**

CREATE TABLE `student\_portal`.`system\_lectures` (

`moduleid` INT(11) NOT NULL UNIQUE,

`lectureid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE,

`lecture\_name` VARCHAR(300) NOT NULL,

`lecture\_lecturer` INT(11) NOT NULL,

`lecture\_notes` VARCHAR(5000),

`lecture\_day` VARCHAR(9) NOT NULL,

`lecture\_from\_time` TIME NOT NULL,

`lecture\_to\_time` TIME NOT NULL,

`lecture\_from\_date` DATE NOT NULL,

`lecture\_to\_date` DATE NOT NULL,

`lecture\_location` VARCHAR(300) NOT NULL,

`lecture\_capacity` INT(11) NOT NULL,

`lecture\_status` VARCHAR(9) NOT NULL,

`created\_on` DATETIME NOT NULL,

`updated\_on` DATETIME,

FOREIGN KEY (moduleid)

REFERENCES system\_modules(moduleid)

ON UPDATE CASCADE

ON DELETE CASCADE

) ENGINE = InnoDB;

**system\_lectures**

CREATE TABLE `student\_portal`.`system\_tutorials` (

`moduleid` INT(11) NOT NULL UNIQUE,

`tutorialid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE,

`tutorial\_name` VARCHAR(300) NOT NULL,

`tutorial\_assistant` INT(11) NOT NULL,

`tutorial\_notes` VARCHAR(5000),

`tutorial\_day` VARCHAR(9) NOT NULL,

`tutorial\_from\_time` TIME NOT NULL,

`tutorial\_to\_time` TIME NOT NULL,

`tutorial\_from\_date` DATE NOT NULL,

`tutorial\_to\_date` DATE NOT NULL,

`tutorial\_location` VARCHAR(300) NOT NULL,

`tutorial\_capacity` VARCHAR(11) NOT NULL,

`tutorial\_status` VARCHAR(9) NOT NULL,

`created\_on` DATETIME NOT NULL,

`updated\_on` DATETIME,

FOREIGN KEY (moduleid)

REFERENCES system\_modules(moduleid)

ON UPDATE CASCADE

ON DELETE CASCADE

) ENGINE = InnoDB;

**system\_tutorials**

CREATE TABLE `student\_portal`.`system\_exams` (

`moduleid` INT(11) NOT NULL UNIQUE,

`examid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE,

`exam\_name` VARCHAR(300) NOT NULL,

`exam\_notes` VARCHAR(5000),

`exam\_date` DATE NOT NULL,

`exam\_time` TIME NOT NULL,

`exam\_location` VARCHAR(300) NOT NULL,

`exam\_capacity` INT(11) NOT NULL,

`exam\_status` VARCHAR(9) NOT NULL,

`created\_on` DATETIME NOT NULL,

`updated\_on` DATETIME,

FOREIGN KEY (moduleid)

REFERENCES system\_modules(moduleid)

ON UPDATE CASCADE

ON DELETE CASCADE

) ENGINE = InnoDB;

**system\_exams**

CREATE TABLE `student\_portal`.`user\_feedback` (

`feedbackid` INT(11) NOT NULL AUTO\_INCREMENT UNIQUE PRIMARY KEY,

`feedback\_subject` VARCHAR(300) NOT NULL,

`feedback\_body` VARCHAR(5000) NOT NULL,

`created\_on` DATETIME NOT NULL

) ENGINE = InnoDB;

**user\_feedback**

CREATE TABLE `student\_portal`.`user\_feedback\_lookup` (

`feedbackid` INT(11) NOT NULL,

`feedback\_from` INT(11) NOT NULL,

`moduleid` INT(11) NOT NULL,

`module\_staff` INT(11) NOT NULL,

`isApproved` TINYINT(1) NOT NULL,

`isRead` TINYINT(1) NOT NULL,

FOREIGN KEY (feedbackid)

REFERENCES user\_feedback(feedbackid),

FOREIGN KEY (feedback\_from)

REFERENCES user\_signin(userid),

FOREIGN KEY (moduleid)

REFERENCES system\_modules(moduleid),

FOREIGN KEY (module\_staff)

REFERENCES user\_signin(userid)

ON UPDATE CASCADE

ON DELETE CASCADE

) ENGINE = InnoDB;

**user\_feedback\_lookup**

CREATE TABLE `system\_map\_markers` (

`markerid` INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`marker\_title` VARCHAR (70) NOT NULL,

`marker\_description` VARCHAR (5000) NOT NULL,

`marker\_lat` FLOAT(10,6) NOT NULL,

`marker\_long` FLOAT(10,6) NOT NULL,

`marker\_category` VARCHAR (70) NOT NULL

) ENGINE = InnoDB;

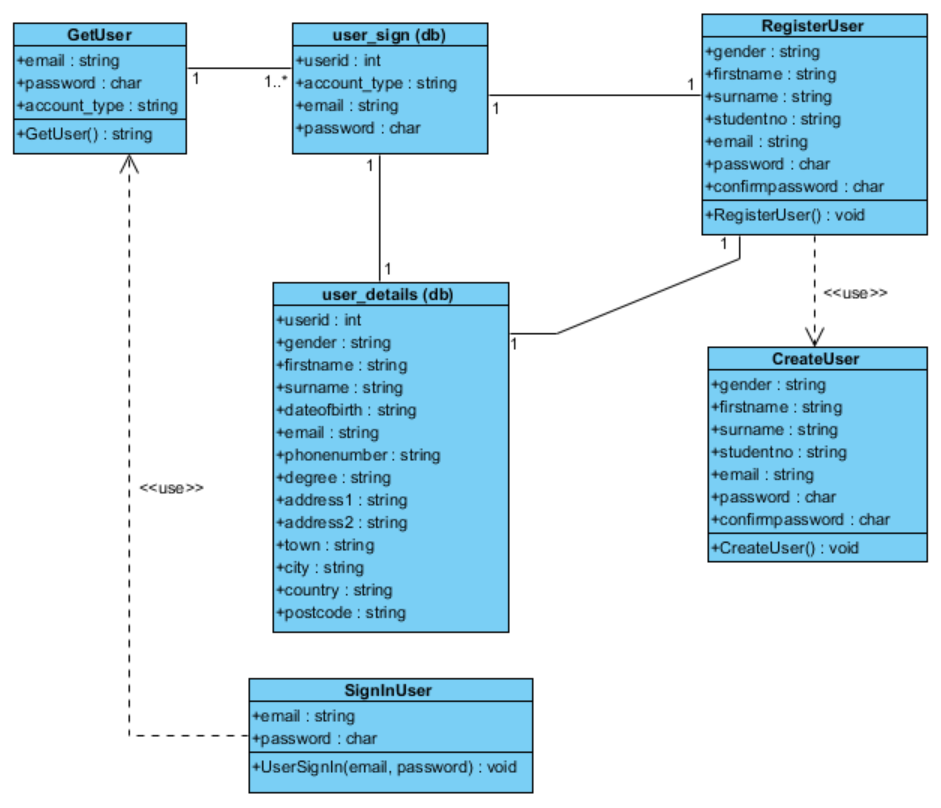
**system\_map\_markers**

**ER Diagram**

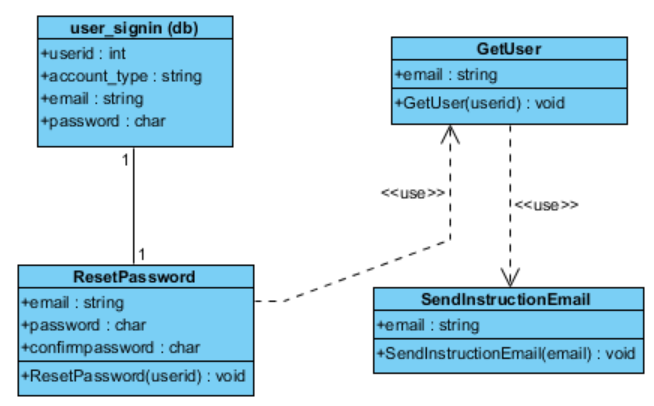


**Class Diagrams**

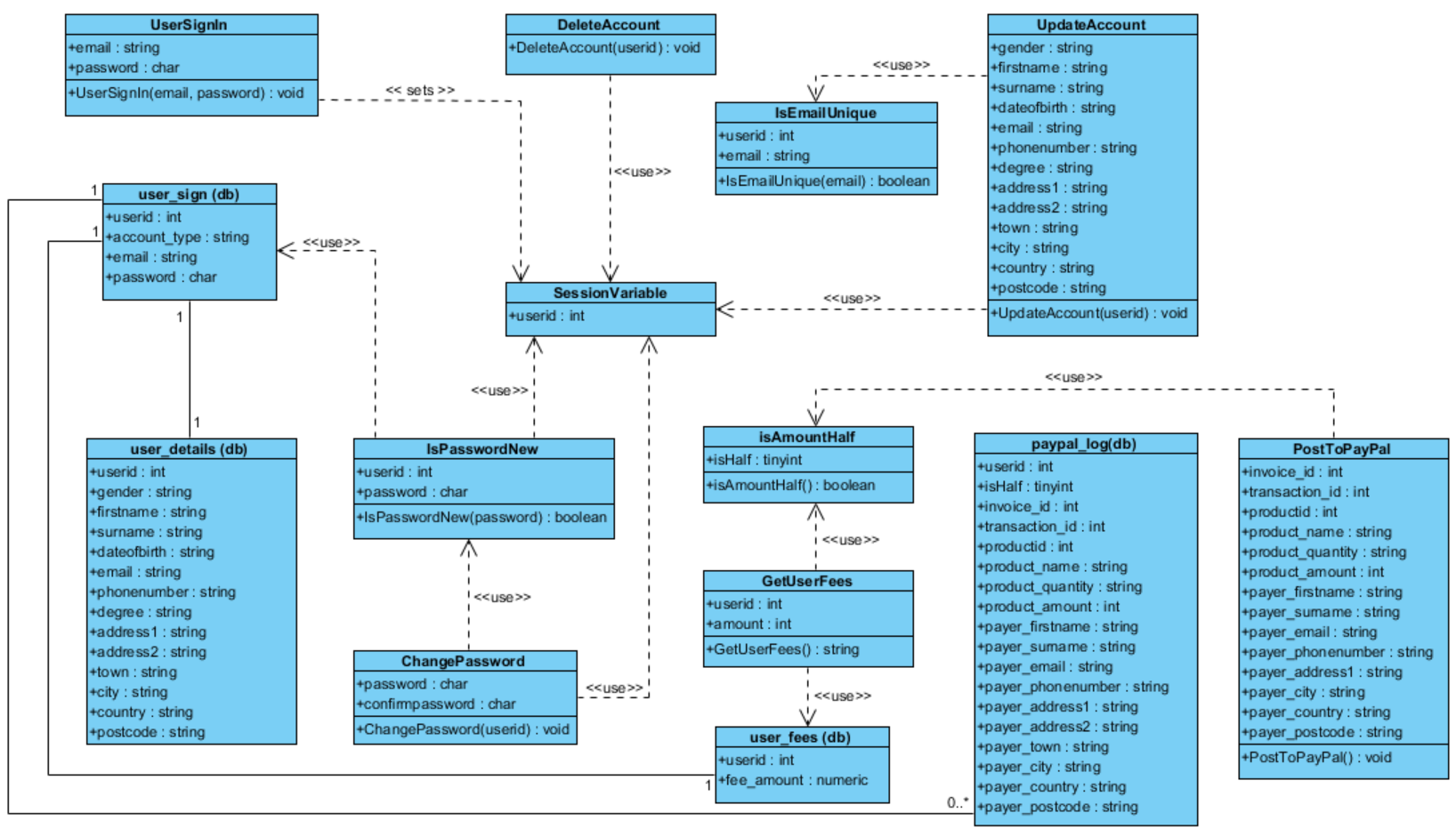
**Sign In / Register**



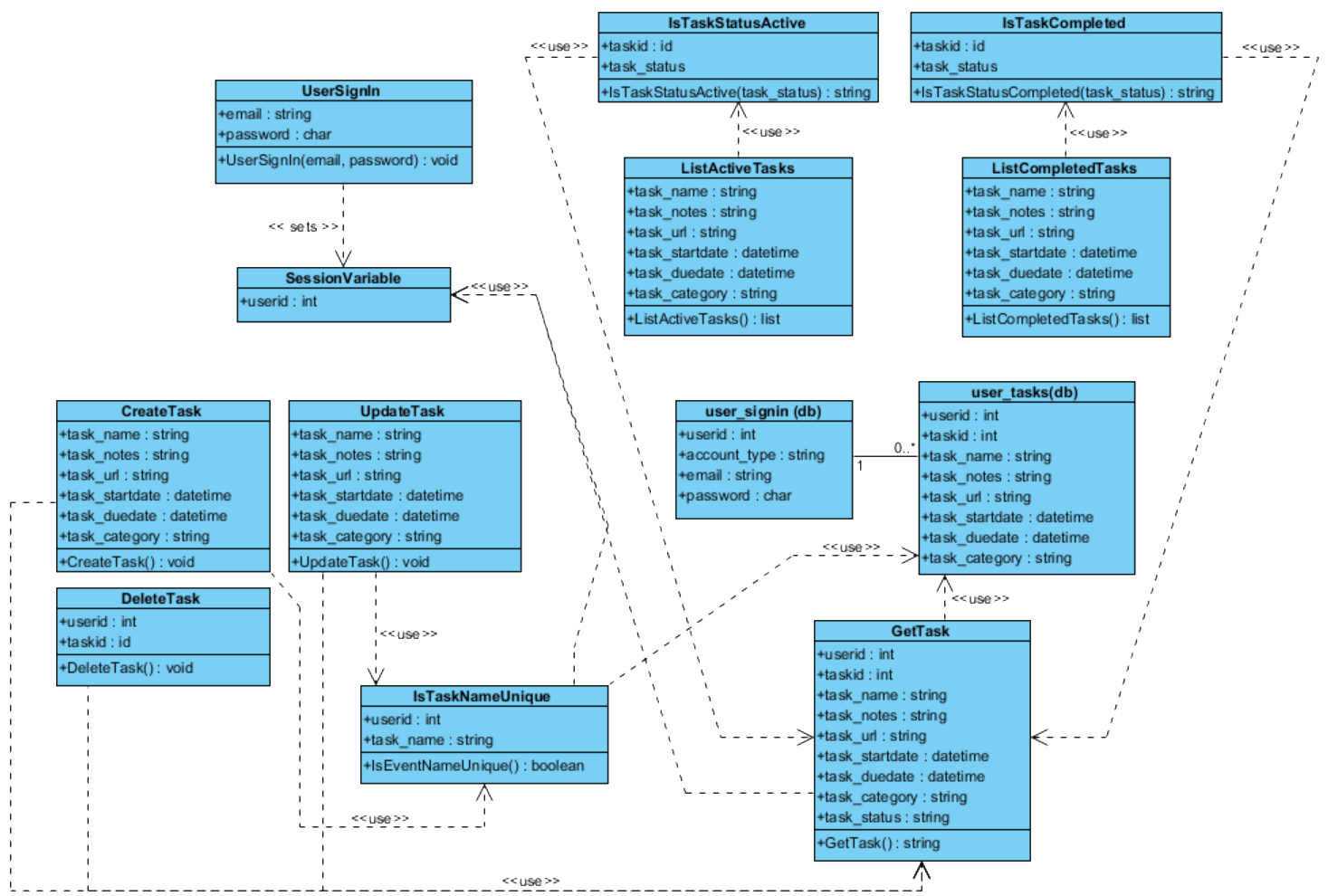
**Forgotten Password / Password Reset**



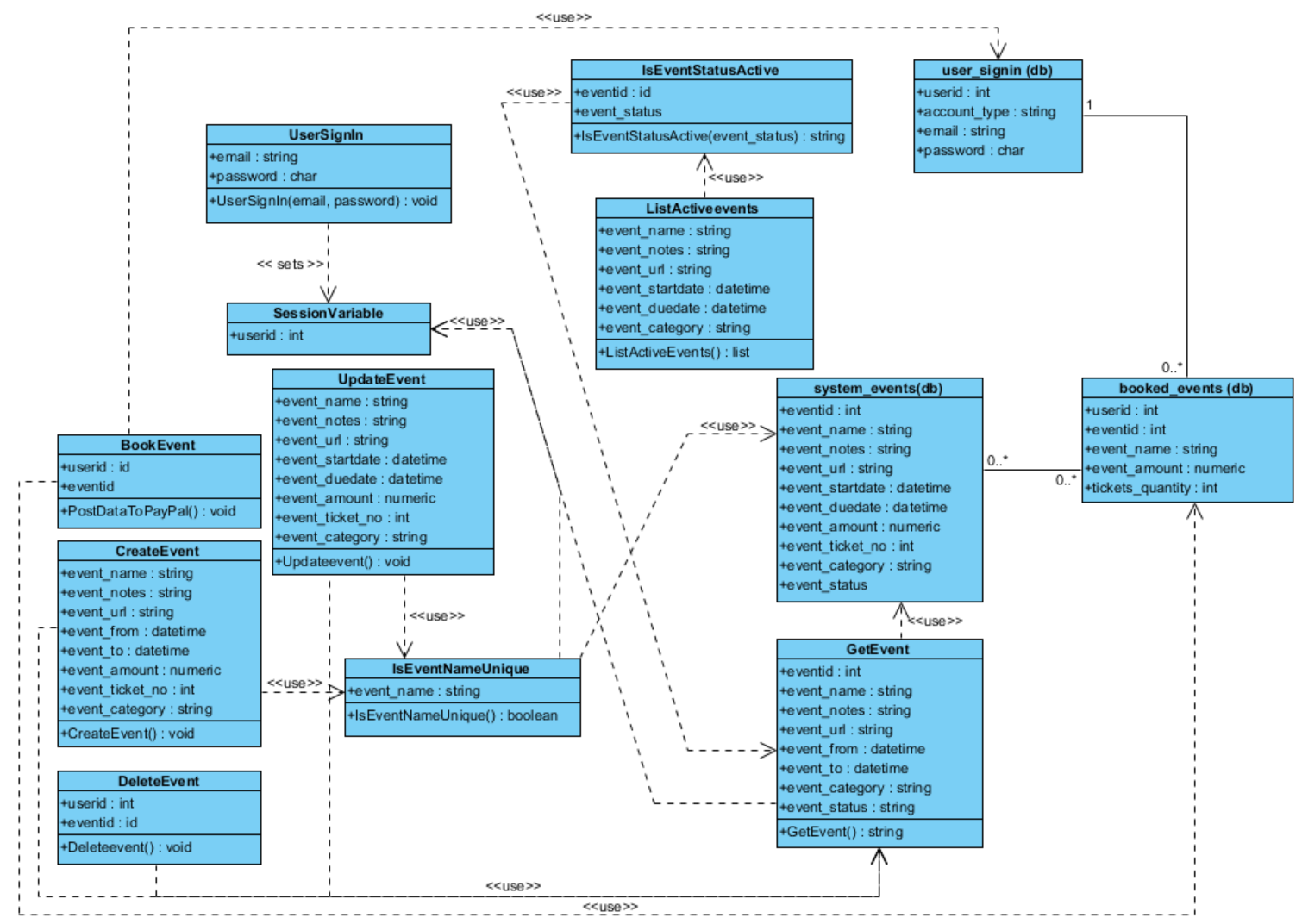
**Account**

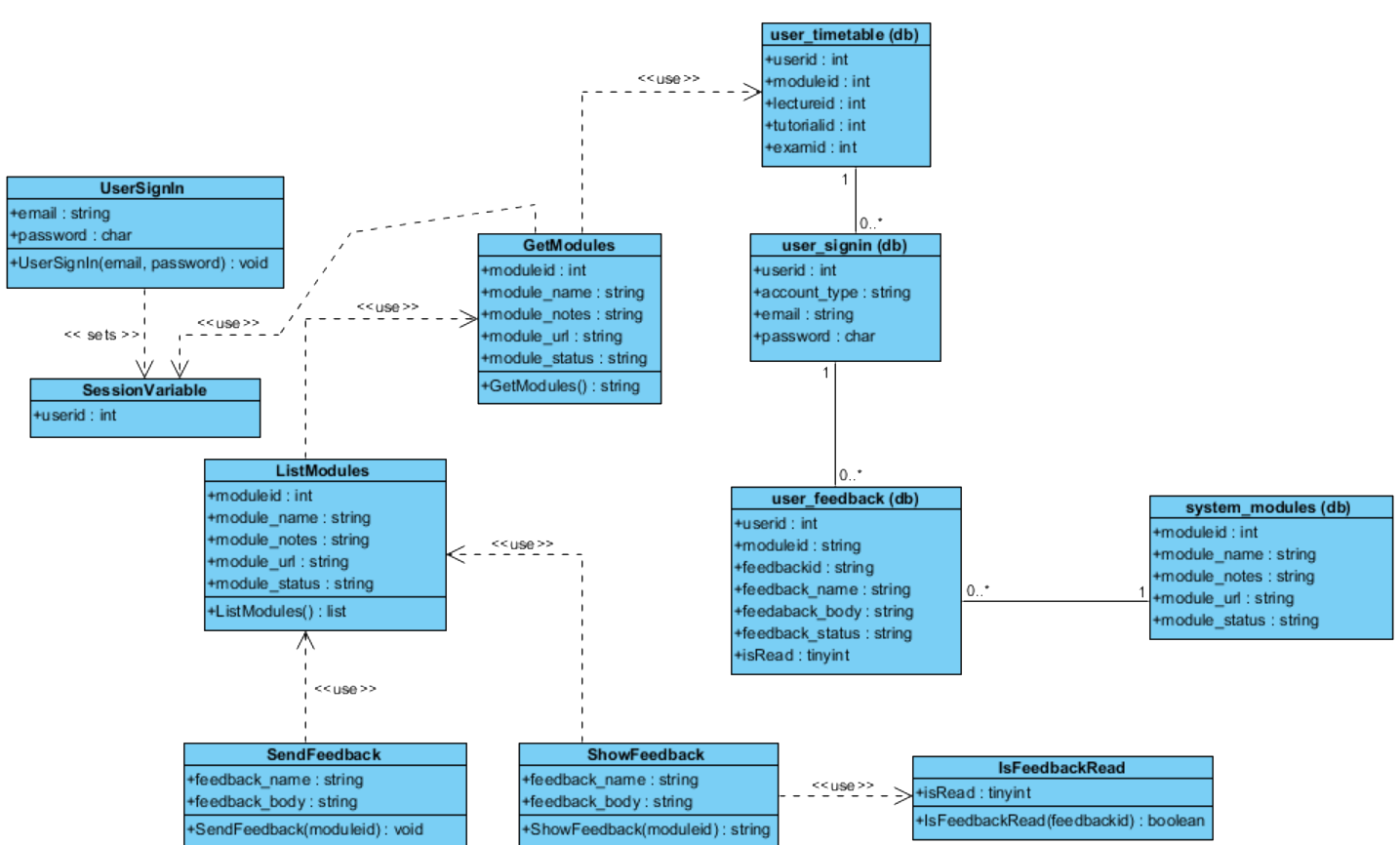


**Calendar**

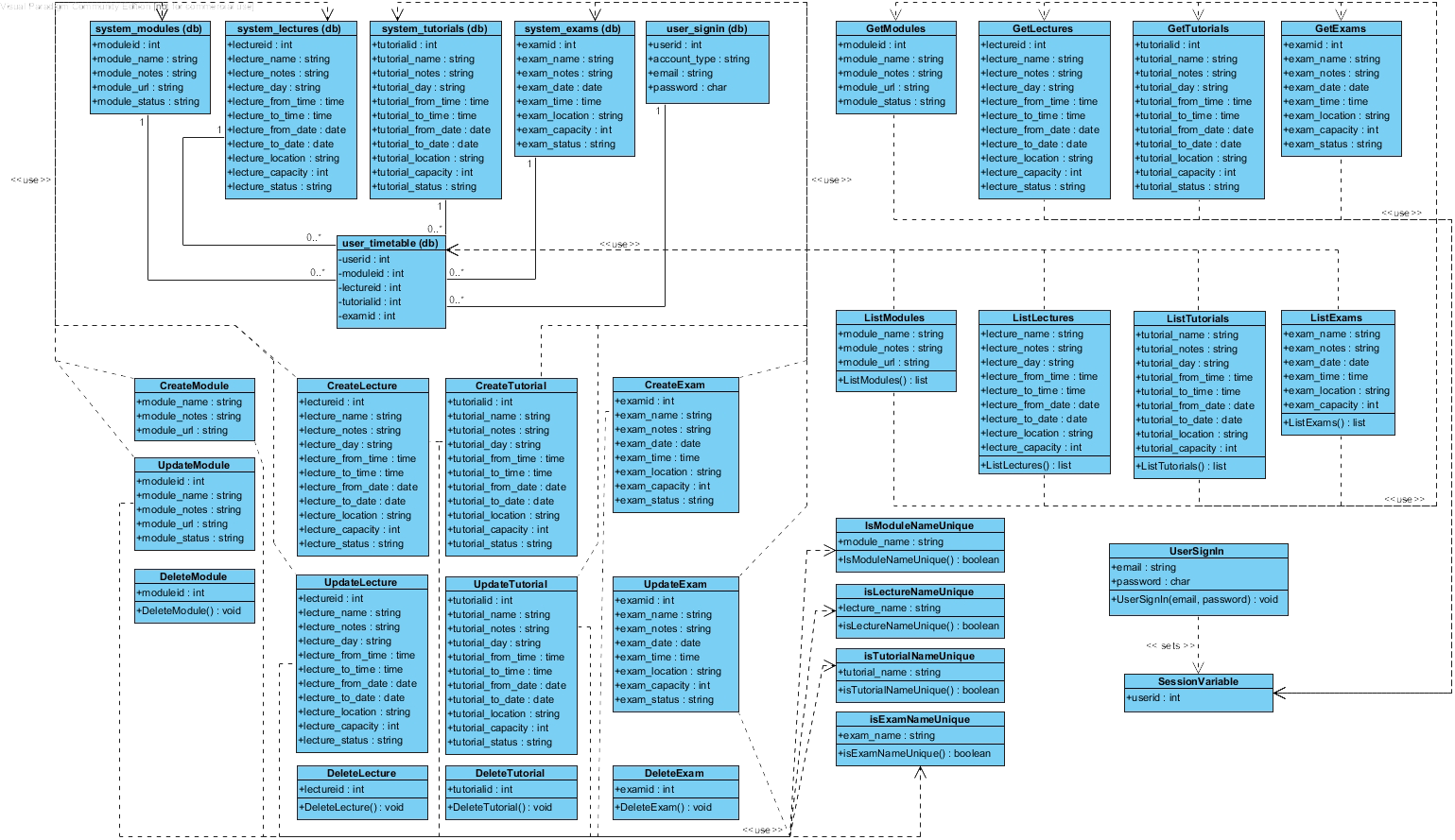


**Events**

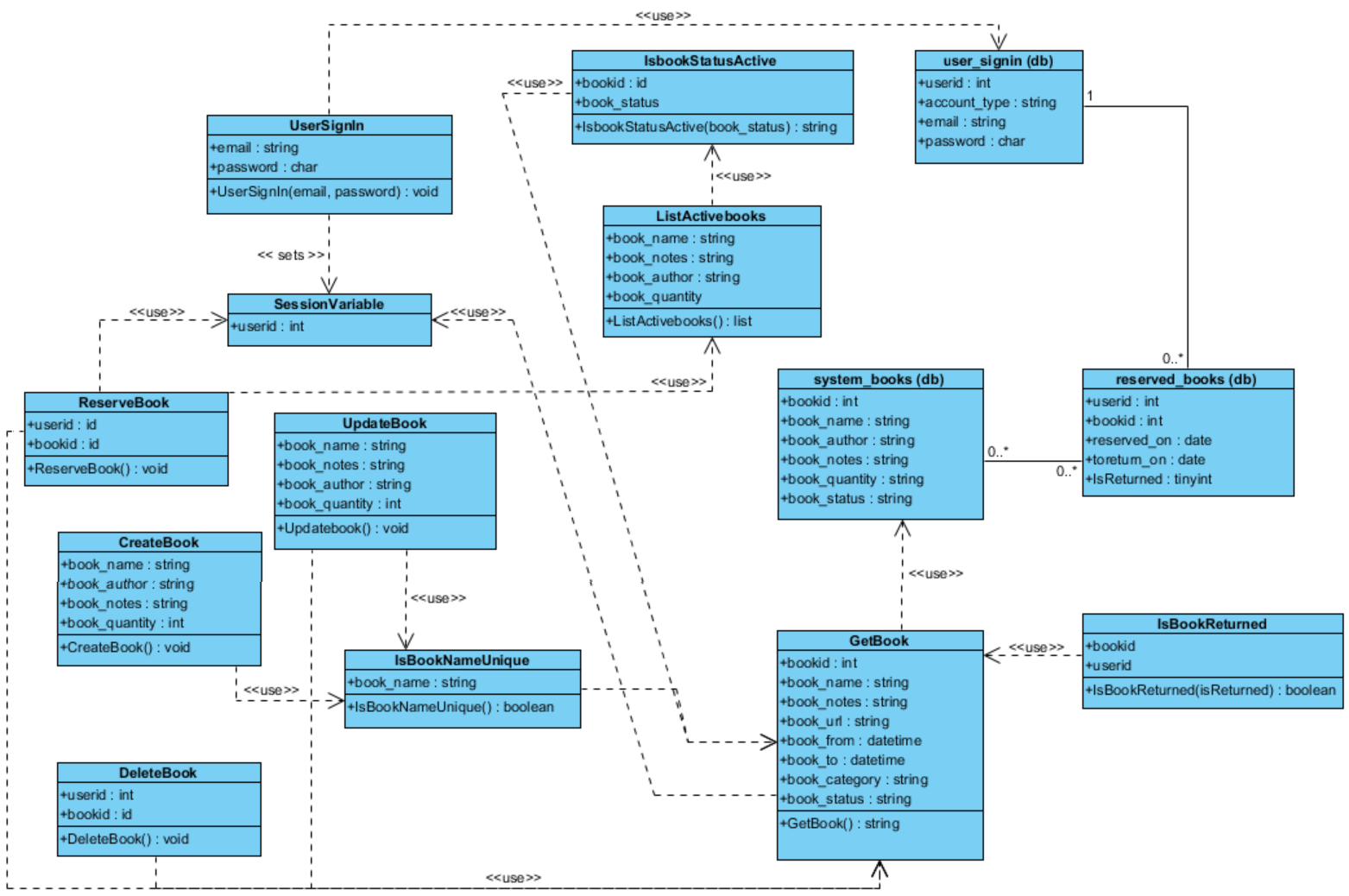


**Feedback**

**Exams**



**Library**



**Messenger**

