JetLAG

Student Organization Software
Use-Case Specifications

Version 2.0

Revision History

Date	Version	Description	Author
29-09-2022	1.0	First Draft	JetLAG Team
08-10-2022	1.1	Finished adding diagrams and figure numbers	JetLAG Team

Table of Contents

Use-Case Specifications

1. Use-Case Model

1.1 Introduction

Our app is primarily focused on facilitating notetaking and productivity for students. Thus, the primary user of SOS is the everyday high school or college student that needs a seamless solution for taking effective notes quickly. The student can access the SOS application on any device. The student can create an account, log into an existing account, or log out of their account if already logged in. The student can create any number of courses and create notes associated with those courses. The notes are also associated with the date they were created. The student can edit their notes and create and view tasks that are associated with a due date. A data broker hosted on Azure will communicate with the Flutter app and the Cosmos DB database to allow notes, account information, and course information to be stored in the cloud.

1.2 General Actors Descriptions

1.2.1 Student

The student is the primary user of SOS. The student can own an account affiliated with the software, create courses, notes, and tasks quickly and efficiently.

1.2.2 Microsoft Azure

Microsoft Azure (a.k.a., Azure) is a cloud computing service that provides cloud computing resources to facilitate cloud-based applications like SOS.

1.2.3 Data Broker

The data broker is an application run on a Virtual Machine in Azure. The data broker communicates between the front-end Flutter application and the Cosmos DB database. The data broker directs information between the database and the flutter app.

1.2.4 Cosmos DB

Cosmos DB is a NoSQL database running on Azure that stores information like notes, course information, and login credentials to ensure that all the student's data is accessible on any device.

1.2.5 Front-End Application (Flutter)

Flutter is a front-end framework designed by google to have support for both webapp development and smart device app development without requiring major code changes between them. This user interface will be what the student primary interacts with.

1.3 Use-Case Model Hierarchy

1.3.1 Establish connection

- **Description**
  In the system when a connection is established, both sides will be able to send and receive data so long as the connection is both authenticated and remains open.

- **Use Cases**
  Save
  Sync
  Login
  Logout
  Database connection

- **Actors**
  User – Responsible for interacting with the UI
  Developer – responsible for automation connection for all programmatic connections
  Flutter app- made by developer, communicates to data broker
  Data broker – handles logic for saving to and retrieving data from database
  Database- communicates with data broker to hold user data.
- **Relationships**
  Connection with flutter app – flutter app connects to data broker over https connections
  Connection to database - data broker connects to the database over TCI connections
- **Packages Owned**
  Data broker
  Flutter app
  Database

1.3.1.1 Packages Diagram

1.3.1.2 <Sub Package One>

- **Description**
- **Use Cases**
- **Actors**
- **Relationships**

1.4 Diagrams of the Use-Case Model

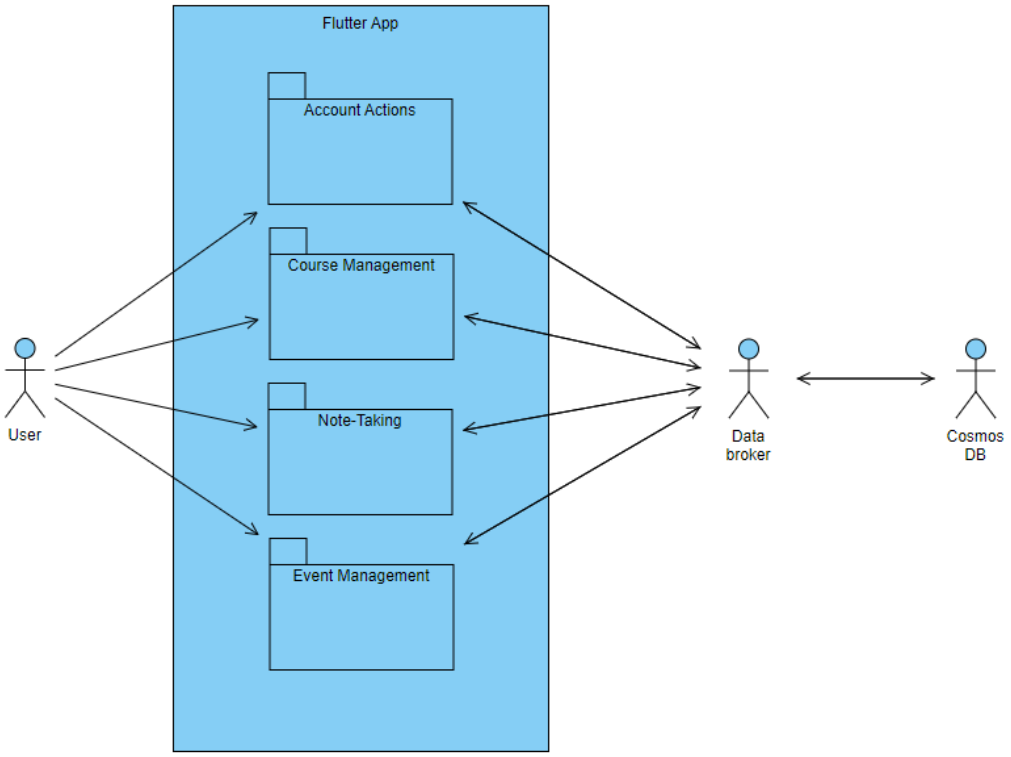

Figure 1: Overview of Use-Cases; each package will be broken down in the following sections

Account Actions

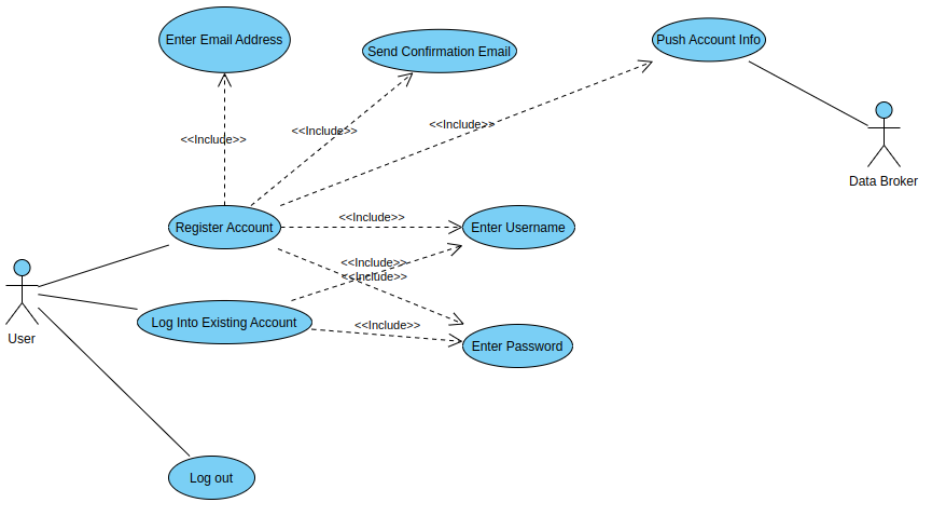

Figure 2: Account Actions

2. Create an Account

2.1 Brief Description

The user will be able to create a username and password, to use along with an email to create an account.

2.2 Flow of Events

2.2.1 Basic Flow

The user enters an email, username, and password into a simple registration UI. When the user clicks on the "Create Account" button, the front-end application sends an email to the address provided, requests to push the username, email, and password to the data broker, and sends a request to the data broker to create a space on the database to store that user's data later. The data broker will then send a response message to the front-end to verify that the request was made successfully.

2.2.2 Alternative Flows

2.2.2.1 Invalid Input

The user entered an email that does not exist, or the user entered invalid characters, or the user left any of the fields blank. In this case, the front-end app will show an error message and ask for input again.

2.3 Special Requirements

None.

2.4 Preconditions

2.4.1 Internet Connection

The user must have an internet connection for the front-end to send requests to the data broker

2.4.2 Downloaded Front-End (optional)

If the user is not connected to the web app through a browser, they will have to have the device-specific application downloaded.

2.5 Postconditions

2.5.1 Successful Registration

The front-end will create a new blank homepage for the new user.

2.5.2 Unsuccessful Registration

The front-end will reload the registration screen and ask for email, username, and password again.

2.6 Extension Points

None.

2.7 Use-Case Diagrams

See Figure 2 above.

2.8 Other Diagrams

None.

3. Log Into Account

3.1 Brief Description

The user will use their previously created credentials in order to access the application and all their data.

3.2 Flow of Events

3.2.1 Basic Flow

The user will enter their username and password into a simple login UI and click the "Log In" button. Then, the front-end will validate the user by sending a request to the data broker. Next, the user's home page will load, allowing the user access to their data.

3.2.2 Alternative Flows

3.2.2.1 Invalid Input

The user entered invalid characters or left any input fields blank. An error message will appear, and the front-end will ask for credentials again

3.2.2.2 Invalid Username or Password

The user entered a username and password pair that did not correspond to an existing account. An error message will appear, and the front-end will ask for credentials again

3.3 Special Requirements

None.

3.4 Preconditions

3.4.1 Internet Connection

The user must have an internet connection for the front-end to send requests to the data broker

3.4.2 Downloaded Front-End (optional)

If the user is not connected to the web app through a browser, they will have to have the device-specific application downloaded.

3.5 Postconditions

3.5.1 Successful Log-In Attempt

The front-end will take the user to a home screen where the user can perform other actions.

3.5.2 Unsuccessful Log-In Attempt

The front-end will reload the registration screen and ask for email, username, and password again.

3.6 Extension Points

None.

3.7 Use-Case Diagrams

See Figure 2 above.

3.8 Other Diagrams

None.

4. Log Out of Account

4.1 Brief Description

The user will log out of their account so that their information is hidden to others and so that another user could use the app on the same device.

4.2 Flow of Events

4.2.1 Basic Flow

The user will navigate the home screen to the “Log Out” button. Then the front-end will go to the login screen to allow another user to login or register.

4.2.2 Alternative Flows

None.

4.3 Special Requirements

None.

4.4 Preconditions

4.4.1 Downloaded Front-End (optional)

If the user is not connected to the web app through a browser, they will have to have the device-specific application downloaded.

4.4.2 Logged Into Application

The user must have previously logged into or registered an account.

4.5 Postconditions

4.5.1 Back to Login Screen

The front-end will go back to the login screen.

4.6 Extension Points

None.

4.7 Use-Case Diagrams

See Figure 2 above.

4.8 Other Diagrams

None.

Course Management

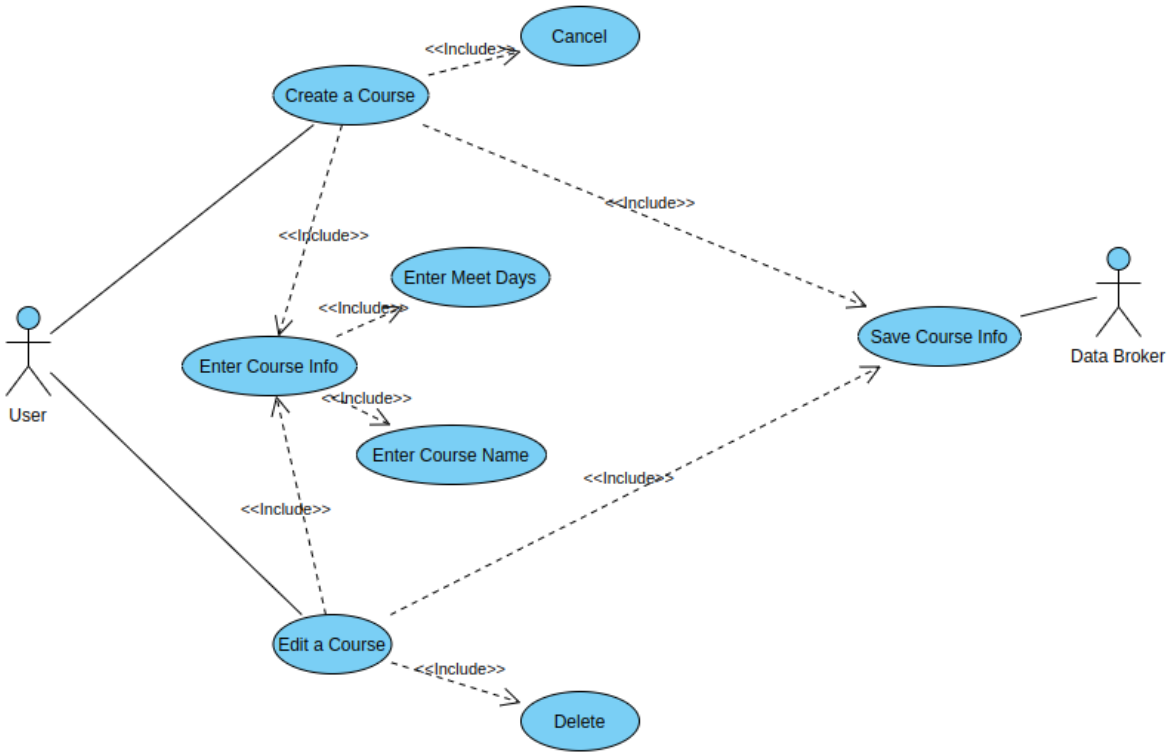

Figure 3: Course Management

5. Create Course

5.1 Brief Description

The user will be able to create a course through which they can take notes and organize their tasks.

5.2 Flow of Events

5.2.1 Basic Flow

The user will click the “Create Course” button. The front-end will take the user to the create course screen where the user will enter a name for the course and select the days that the class meets. The user will then click the “save” button. The front-end will push the course information to the data broker.

5.2.2 Alternative Flows

5.2.2.1 Invalid Input

The user inputs invalid characters for the course name. In this case, the course cannot be created, an error will appear, and the user will be asked to input the course name again.

5.2.2.2 Cancelling Course

The user clicks the “Cancel” button. The front-end takes the user back to the home screen.

5.3 Special Requirements

None.

5.4 Preconditions

5.4.1 Internet Connection

The user must have an internet connection for the front-end to send requests to the data broker.

5.4.2 Downloaded Front-End (optional)

If the user is not connected to the web app through a browser, they will have to have the device-specific application downloaded.

5.4.3 Logged Into Application

The user must have previously logged into or registered an account.

5.5 Postconditions

5.5.1 Back to Home screen

The front-end will display the home screen as it was before, except now with the new course displayed in the course section.

5.6 Extension Points

None.

5.7 Use-Case Diagrams

See Figure 3 above.

5.8 Other Diagrams

None.

6. Edit Course

6.1 Brief Description

The user will be able to edit course details (name and meeting days) or the user can delete the course.

6.2 Flow of Events

6.2.1 Basic Flow

The user will navigate to the courses tab and click the "Edit" button next to the course they wish to edit. The user will change course name or meeting days and click "Save". Then the front-end will send a request to update the course information to the data broker.

6.2.2 Alternative Flows

6.2.2.1 Invalid Input

The user inputs invalid characters for the course name. In this case, the course cannot be created, an error will appear, and the user will be asked to input the course name again.

6.2.2.2 Deleting Course

The user clicks the "Delete" button. The front-end sends a request to delete the course from the database to the data broker. Any data associated with the course is also deleted.

6.3 Special Requirements

None.

6.4 Preconditions

6.4.1 Internet Connection

The user must have an internet connection for the front-end to send requests to the data broker.

6.4.2 Downloaded Front-End (optional)

If the user is not connected to the web app through a browser, they will have to have the device-specific application downloaded.

6.4.3 Logged Into Application

The user must have previously logged into or registered an account.

6.4.4 Created a course

The user must have created a course previously.

6.5 Postconditions

6.5.1 Back to Home screen

The front-end will display the home screen as it was before, except now with the new course displayed in the course section.

6.6 Extension Points

None.

6.7 Use-Case Diagrams

See Figure 3 above.

6.8 Other Diagrams

None.

Note-Taking

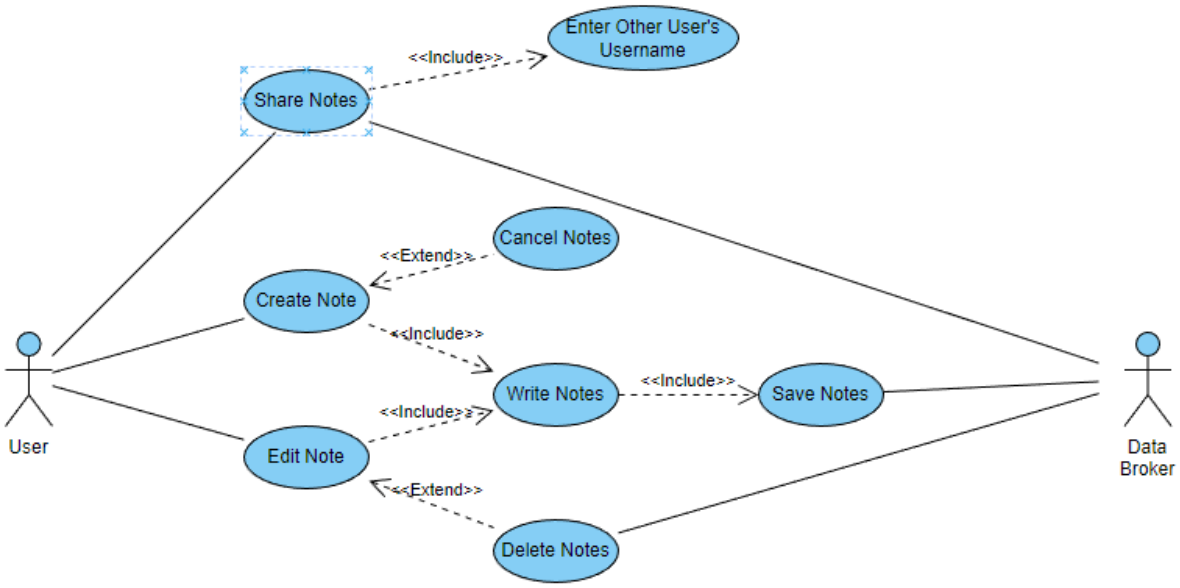

Figure 4: Note-Taking

7. Create Note

7.1 Brief Description

The user will be able to create notes in simple .txt format and save them to the cloud.

7.2 Flow of Events

7.2.1 Basic Flow

The user will select a previously created course and then click “Add Note”. The front-end will switch to a text editor and the user will type their notes. When the user is done, they will click “Save Notes”. The front-end will then push the notes file to the data broker.

7.2.2 Alternative Flows

7.2.2.1 Cancel Note

The user can click the “Cancel” button to cancel the creation of the notes. The front-end will go back to the home screen with an indicator on the course indicating that it has a note associated with it.

7.3 Special Requirements

None.

7.4 Preconditions

7.4.1 Internet Connection

The user must have an internet connection for the front-end to send requests to the data broker.

7.4.2 Downloaded Front-End (optional)

If the user is not connected to the web app through a browser, they will have to have the device-specific application downloaded.

7.4.3 Logged Into Application

The user must have previously logged into or registered an account.

7.4.4 Created a course

The user must have created a course previously.

7.5 Postconditions

7.5.1 Back to Home screen

The front-end will display the home screen as it was before, except now with an indicator showing that there is a note associated with a course.

7.6 Extension Points

None.

7.7 Use-Case Diagrams

See Figure 4 above.

7.8 Other Diagrams

None.

8. Edit Notes

8.1 Brief Description

The user will be able to edit existing notes or delete them.

8.2 Flow of Events

8.2.1 Basic Flow

The user will navigate to a course that has notes associated with it and click on the “Edit notes” button. The user will change the notes and then click the “Save” button. The front-end will push the new version of the notes to the data broker.

8.2.2 Alternative Flows

8.2.2.1 Deleting Notes

The user clicks the “Delete” button. The front-end sends a request to delete the notes from the database to the data broker.

8.3 Special Requirements

None.

8.4 Preconditions

8.4.1 Internet Connection

The user must have an internet connection for the front-end to send requests to the data broker.

8.4.2 Downloaded Front-End (optional)

If the user is not connected to the web app through a browser, they will have to have the device-specific application downloaded.

8.4.3 Logged Into Application

The user must have previously logged into or registered an account.

8.4.4 Created a course

The user must have created a course previously.

8.4.5 Created a note

The user must have created a note for that course previously.

8.5 Postconditions

8.5.1 Back to Home screen

The front-end will display the home screen as it was before. If the user deleted the note, then there will be no indicator for that note.

8.6 Extension Points

None.

8.7 Use-Case Diagrams

See Figure 4 above.

8.8 Other Diagrams

None.

9. Share Notes

9.1 Brief Description

The user will be able to share notes with another user of SOS.

9.2 Flow of Events

9.2.1 Basic Flow

The user will click the “share” button next to one of their notes. Then they will be prompted to enter the username of another user. After hitting the “Submit” button, the front-end will make a request to push the notes to the other user’s database.

9.2.2 Alternative Flows

9.2.2.1 Cancelling

The user clicks the “Cancel” button and the front-end takes the user back to the home screen.

9.2.2.2 Invalid User

The username typed into the input field is not the username of an existing user.

9.3 Special Requirements

None.

9.4 Preconditions

9.4.1 Internet Connection

The user must have an internet connection for the front-end to send requests to the data broker.

9.4.2 Downloaded Front-End (optional)

If the user is not connected to the web app through a browser, they will have to have the device-specific application downloaded.

9.4.3 Logged Into Application

The user must have previously logged into or registered an account.

9.4.4 Created a course

The user must have created a course previously.

9.4.5 Created a note

The user must have created a note for that course previously.

9.5 Postconditions

9.5.1 Back to Home screen

The front-end will display the home screen as it was before.

9.6 Extension Points

None.

9.7 Use-Case Diagrams

See Figure 4 above.

9.8 Other Diagrams

None.

Event Management

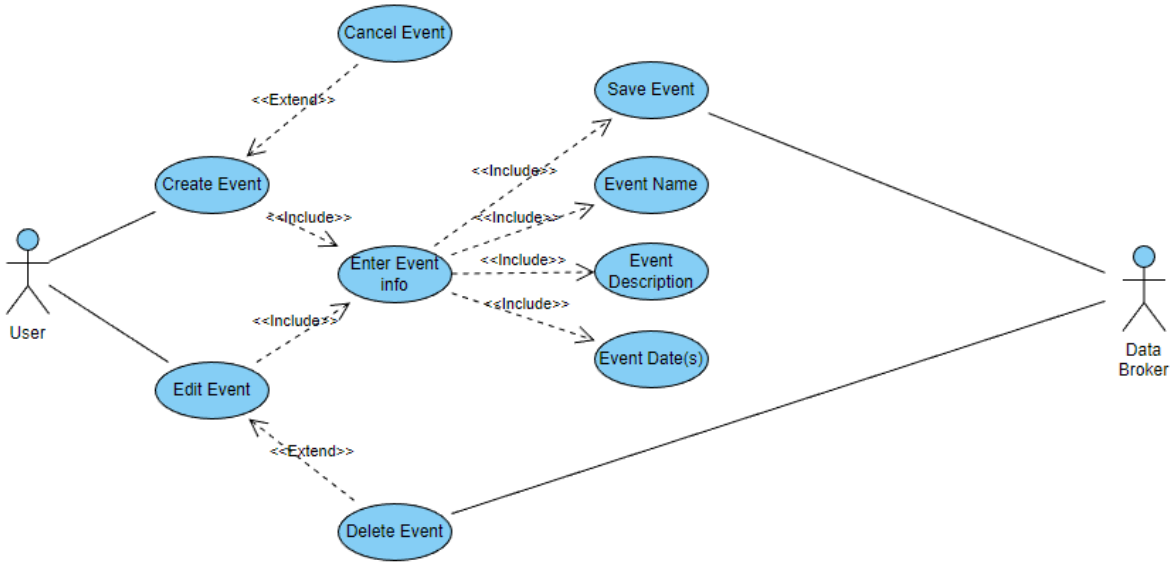

Figure 5: Event Management

10. Create Event

10.1 Brief Description

The user will be able to create events for specific classes, set for specific days.

10.2 Flow of Events

10.2.1 Basic Flow

The user will click the “Create Event” button next to a course on the home screen. Then they will name the event, give a description for it, and set a date for it. The user will click the “Save” button and the front-end app will push the event to the data broker.

10.2.2 Alternative Flows

10.2.2.1 Cancel

The user clicks the “Cancel” button. The front-end displays the home screen. There is no indicator of any events existing for that class.

10.3 Special Requirements

None.

10.4 Preconditions

10.4.1 Internet Connection

The user must have an internet connection for the front-end to send requests to the data broker.

10.4.2 Downloaded Front-End (optional)

If the user is not connected to the web app through a browser, they will have to have the device-specific application downloaded.

10.4.3 Logged Into Application

The user must have previously logged into or registered an account.

10.4.4 Created a course

The user must have created a course previously.

10.5 Postconditions

10.5.1 Back to Home screen

The front-end will display the home screen as it was before. This time with an indicator showing that there is an event for the class.

10.6 Extension Points

None.

10.7 Use-Case Diagrams

See Figure 5 above.

10.8 Other Diagrams

None.

11. Edit Event

11.1 Brief Description

The user will be able to edit events and their details.

11.2 Flow of Events

11.2.1 Basic Flow

The user will click the edit button next to an event and change the name, description, and/or date and then click “Save”. The front-end app will then push the new event to the data broker.

11.2.2 Alternative Flows

11.2.2.1 Deleting an Event

The user clicks the “delete” button. The front-end sends a request to the data broker to delete the event.

11.3 Special Requirements

None.

11.4 Preconditions

11.4.1 Internet Connection

The user must have an internet connection for the front-end to send requests to the data broker.

11.4.2 Downloaded Front-End (optional)

If the user is not connected to the web app through a browser, they will have to have the device-specific application downloaded.

11.4.3 Logged Into Application

The user must have previously logged into or registered an account.

11.4.4 Created a course

The user must have created a course previously.

11.4.5 Created an Event

The user must have created an event.

11.5 Postconditions

11.5.1 Back to Home screen

The front-end will display the home screen as it was before. If the user deleted the event, then there will be no indicator for that event.

11.6 Extension Points

None.

11.7 Use-Case Diagrams

See Figure 5 above.

11.8 Other Diagrams

None.