JetLAG SOS Tool
Use-Case-Realization Specification

Version 1.0

Revision History

Date	Version	Description	Author
22/10/2022	1.0	First draft	JetLAG team

Table of Contents

Use-Case-Realization Specification

1. Introduction

1.1 Purpose

This document contains an overview of each use-case, using sequence diagrams to show how the use case is realized.

1.2 Scope

This document further describes the use cases that were previously discussed in the Use-Case Specifications Document. This document will only provide sequence diagrams, as discussed in class.

1.3 References

Use-Case Specification Document

1.4 Overview

Following this section (1. Introduction), each following major section will describe a use case previously discussed in the Use-Case Specifications Document. For each use case, its flow of events will be described briefly, a use case diagram from the Use-Case Specifications Document will be provided, and a sequence diagram will be provided.

Account Actions

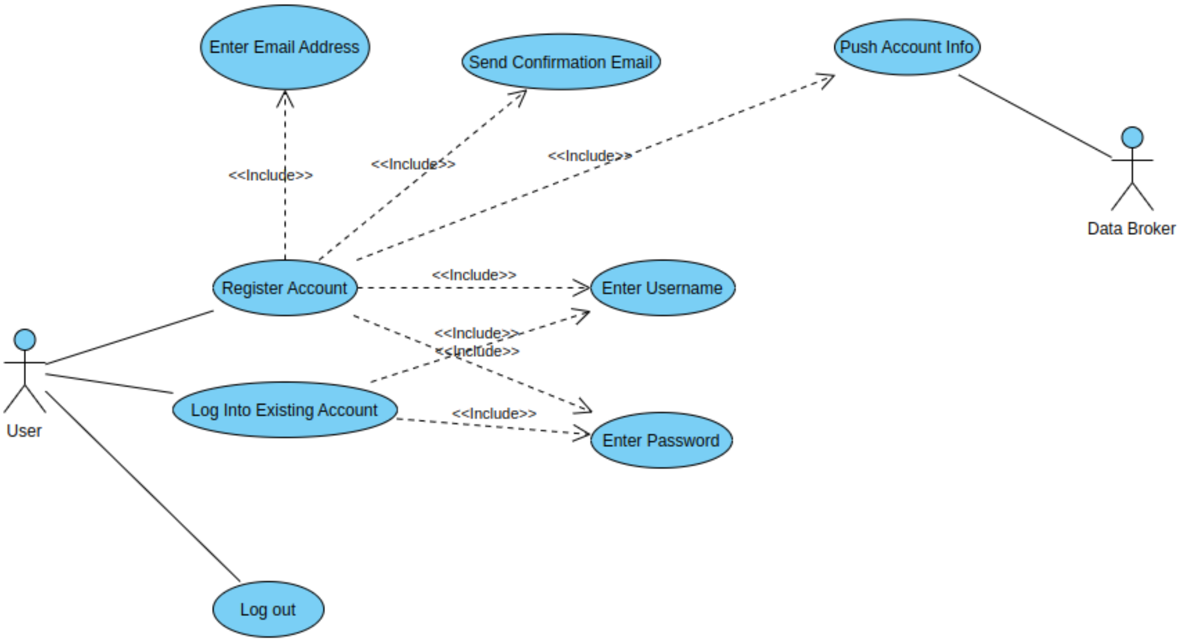


Fig. 1.1: Account Actions Use-Case diagram (from Use-Case Specifications Document)

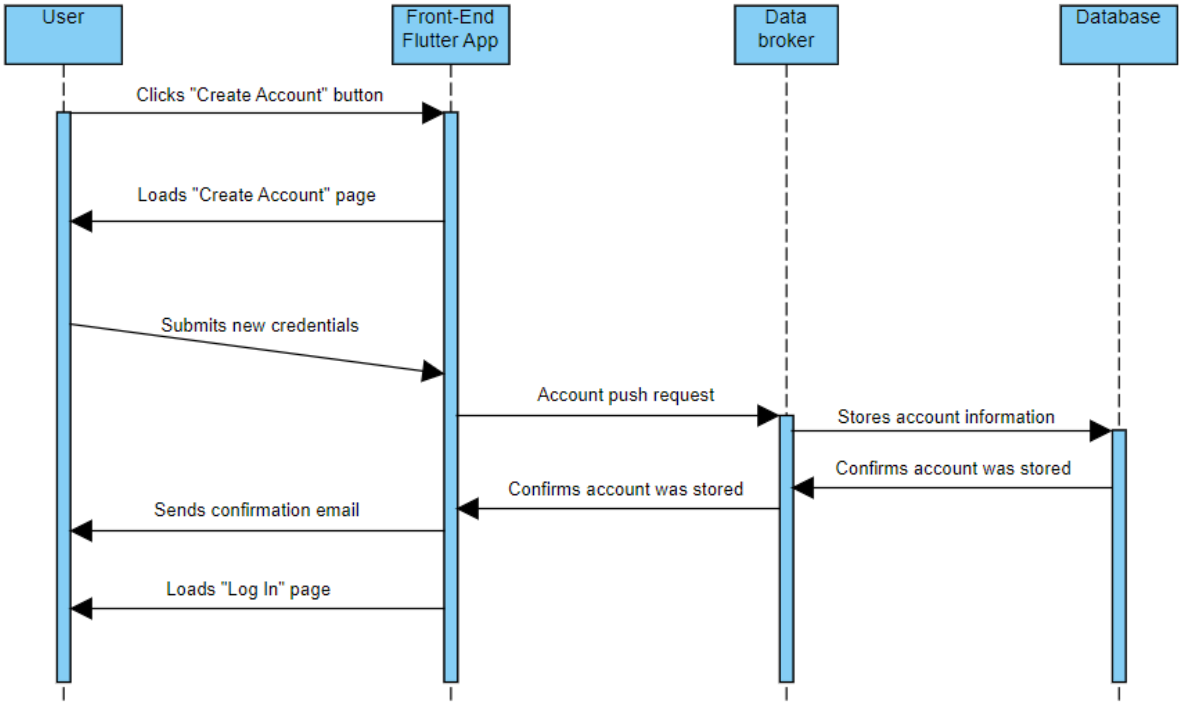
1. Create an Account

1.5 Flow of Events - Design

Upon the user clicking the “Create Account” button, the front-end will load the create account page. Then the user enters their new credentials and clicks submit. The data broker pushes the new account information to the database and then confirms that it was stored. Then the front-end sends a confirmation email to the user, and loads the log in page.

1.6 Interaction Diagrams

1.6.1 Sequence Diagrams



User
Front-End Flutter App
Data broker
Database
Clicks "Create Account" button
Loads "Create Account" page
Submits new credentials
Account push request
Stores account information
Confirms account was stored
Confirms account was stored
Sends confirmation email
Loads "Log In" page

Fig. 1.2: Create Account Sequence Diagram

i. Participating objects

Front-End The application that handles UI and basic front-end functionality

Data Broker The application that is stored on a virtual machine in Azure. This object facilitates communication between front end and database, and handles things like authentication

Database The database stores data related to accounts, courses, notes, and events

1.7 Class Diagrams

1.8 Derived Requirements

2. Log Into Account

2.1 Flow of Events - Design

Upon opening the app, the user will be greeted by a log in page. After entering their credentials, the front end will send a verification request to the data broker which will request the credentials from the database and authenticate the user. Then the data broker will send a confirmation to the front end, which will load the user's homepage.

2.2 Interaction Diagrams

ii. Sequence Diagrams

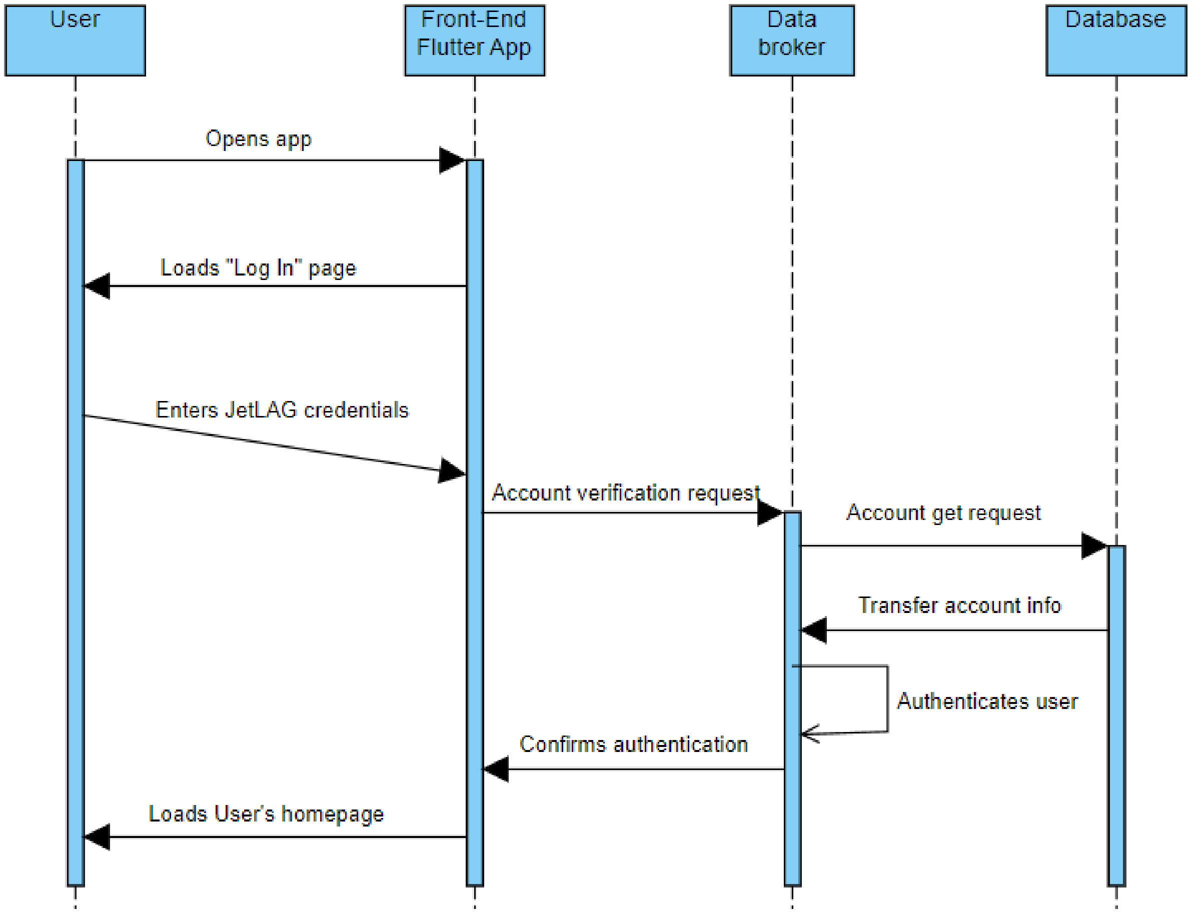


Fig. 1.3: Log in Sequence Diagram

2.2.1 Collaboration Diagrams

2.2.2 Participating objects

Front-End The application that handles UI and basic front-end functionality

Data Broker The application that is stored on a virtual machine in Azure. This object facilitates communication between front end and database, and handles things like authentication

Database The database stores data related to accounts, courses, notes, and events

2.3 Class Diagrams

2.4 Derived Requirements

3. Log Out of Account

3.1 Flow of Events - Design

The user will click the log out button. The front end sends an end session request to the data broker which confirms that the session has ended, and the front end will load the log in page.

3.2 Interaction Diagrams

3.2.1 Sequence Diagrams

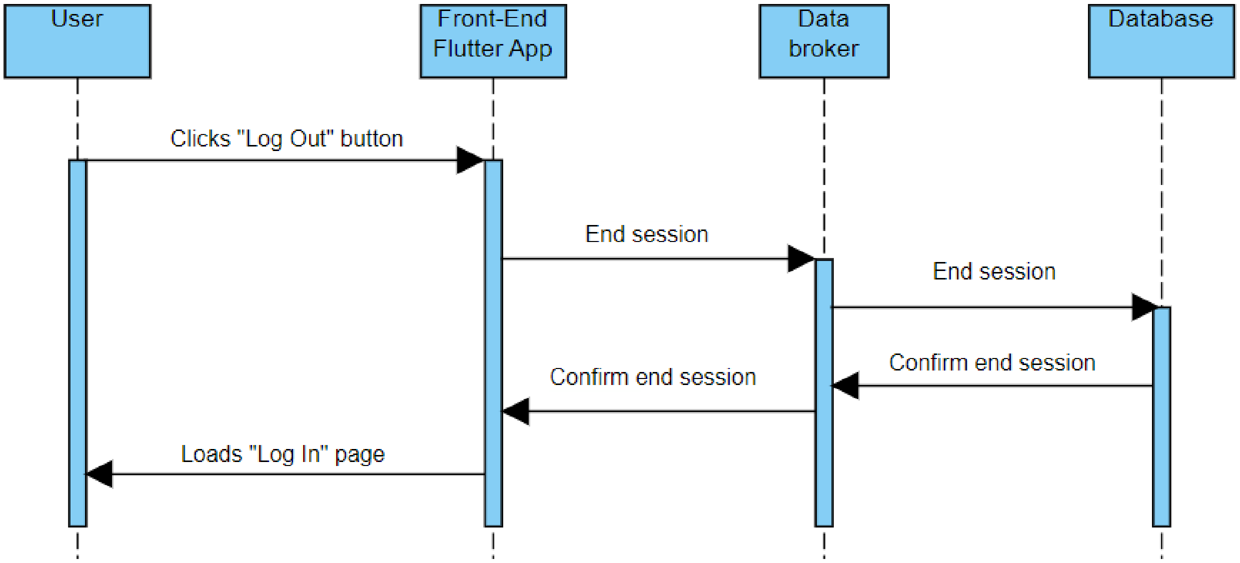


Fig. 1.4: Log Out Sequence Diagram

3.2.2 Participating objects

Front-End The application that handles UI and basic front-end functionality

Data Broker The application that is stored on a virtual machine in Azure. This object facilitates communication between front end and database, and handles things like authentication

Database The database stores data related to accounts, courses, notes, and events

3.3 Class Diagrams

3.4 Derived Requirements

Course Management

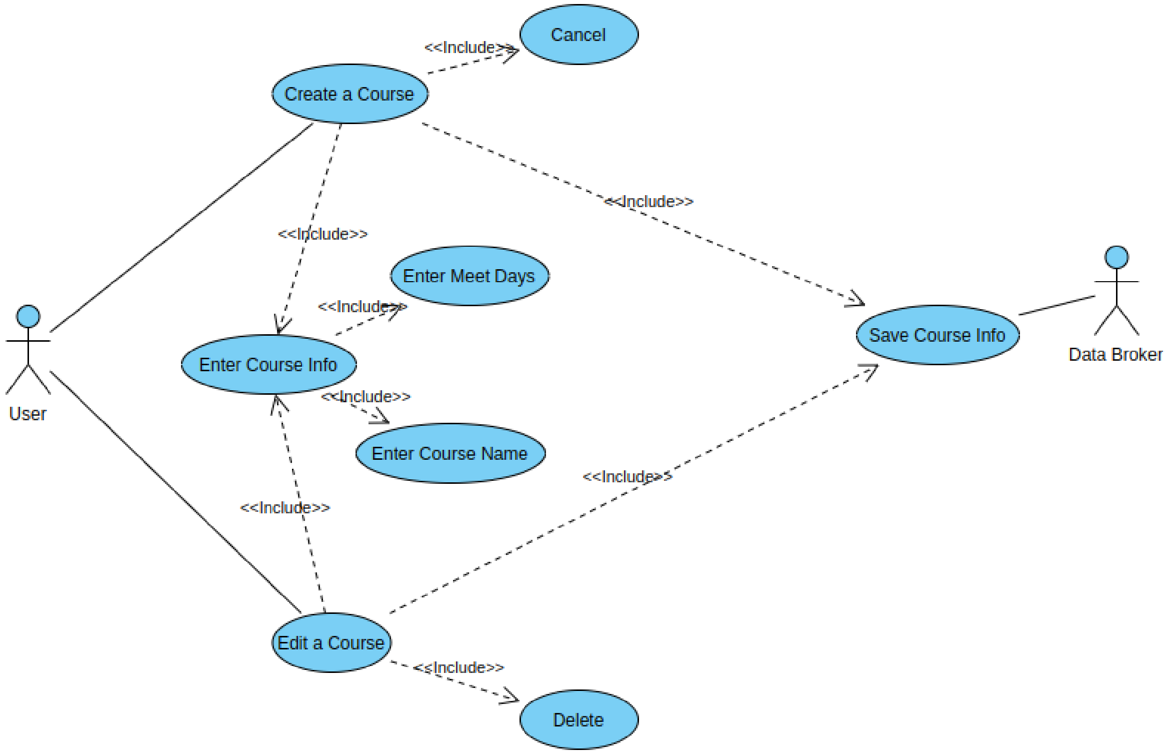


Fig. 2.1: Course Management Use-Case diagram (from Use-Case Specifications Document)

4. Create Course

4.1 Flow of Events - Design

The user will click the create course button. The front end will load the edit course page where the user will enter new course information (course name, dates, description, etc.). After submitting the course information, the front end will push it to the data broker which will push it to the database. A confirmation will be sent back to the front end which will then load the courses page.

4.2 Interaction Diagrams

4.2.1 Sequence Diagrams

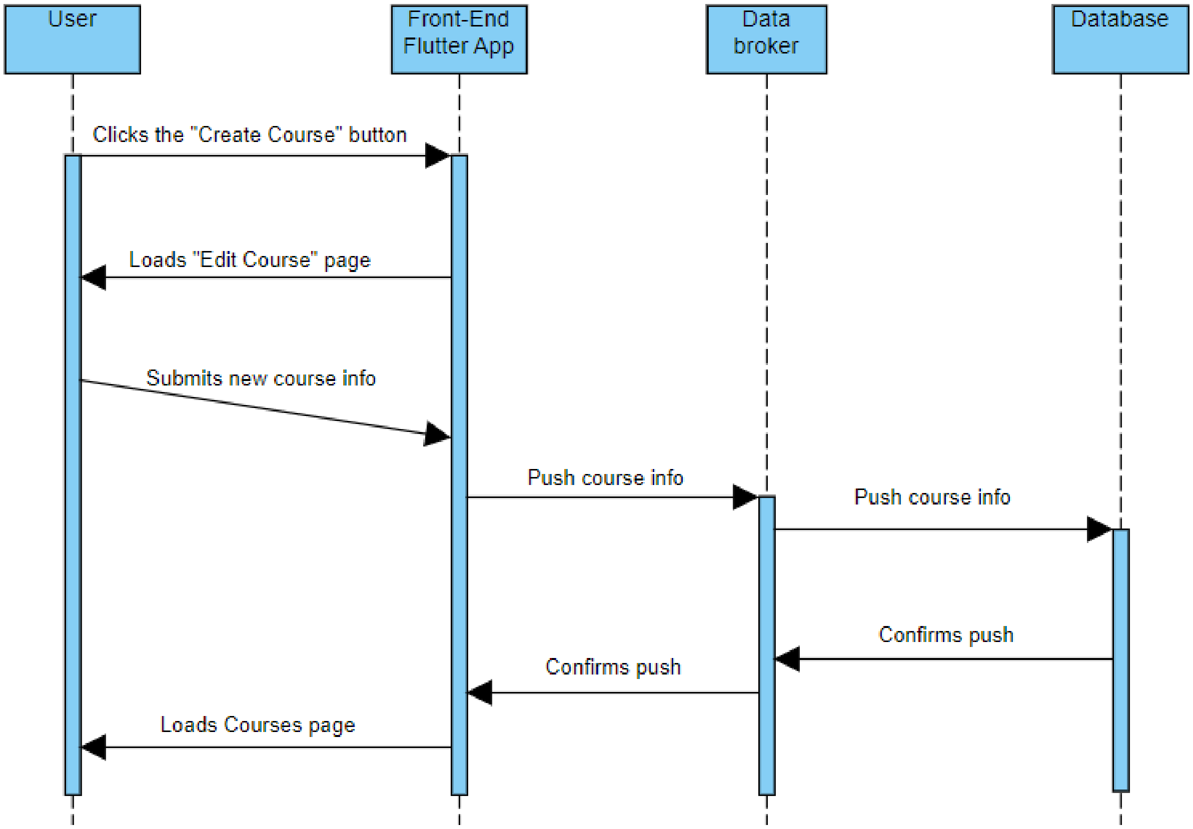


Fig. 2.2: Create Course Sequence Diagram

4.2.2 Participating objects

Front-End The application that handles UI and basic front-end functionality

Data Broker The application that is stored on a virtual machine in Azure. This object facilitates communication between front end and database, and handles things like authentication

Database The database stores data related to accounts, courses, notes, and events

4.3 Class Diagrams

4.4 Derived Requirements

5. Edit Course

5.1 Flow of Events - Design

Much like the create course use-case, upon clicking the edit course button, the front end will load the edit course page, where the user will submit new course information. This information will be pushed to the data broker and to the database and a confirmation will be pushed back to the front end, which will then load the courses page.

5.2 Interaction Diagrams

5.2.1 Sequence Diagrams

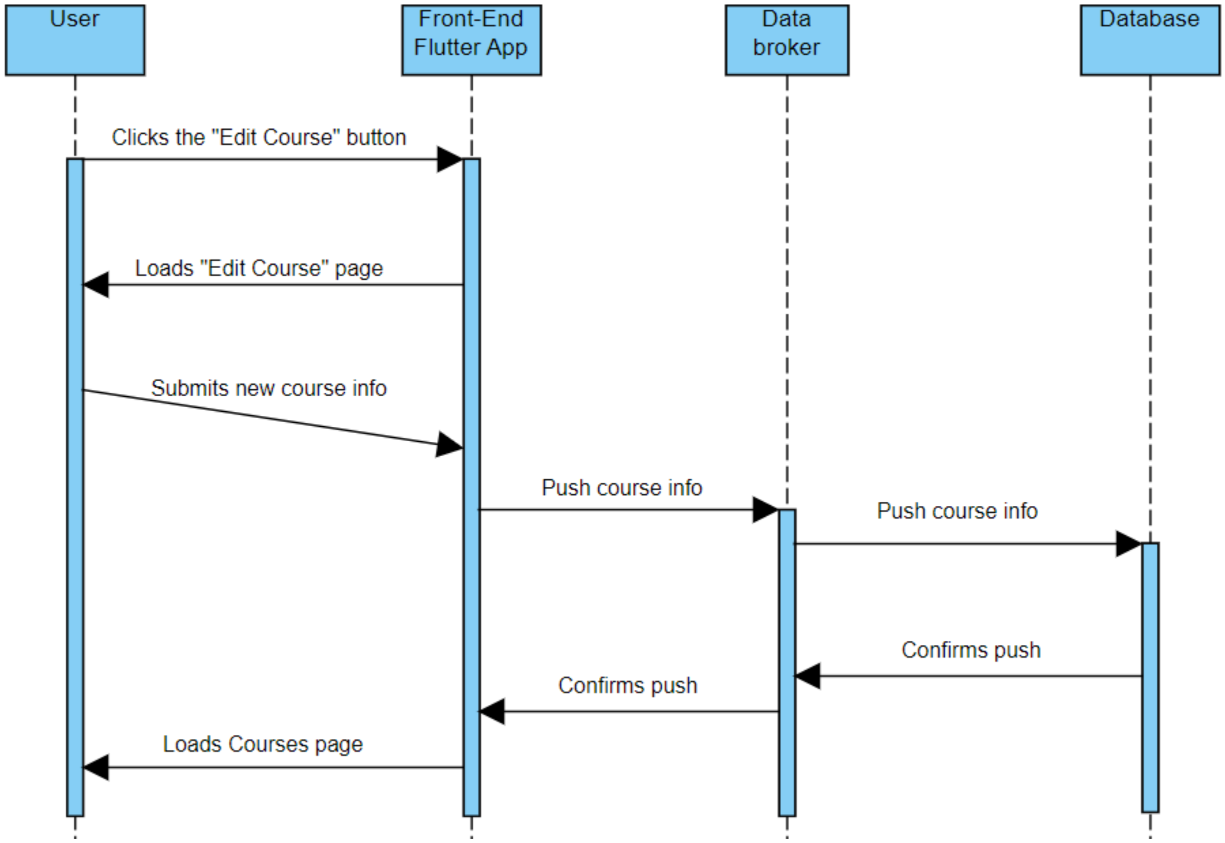


Fig. 2.3: Edit Course Sequence Diagram

5.2.2 Participating objects

Front-End The application that handles UI and basic front-end functionality

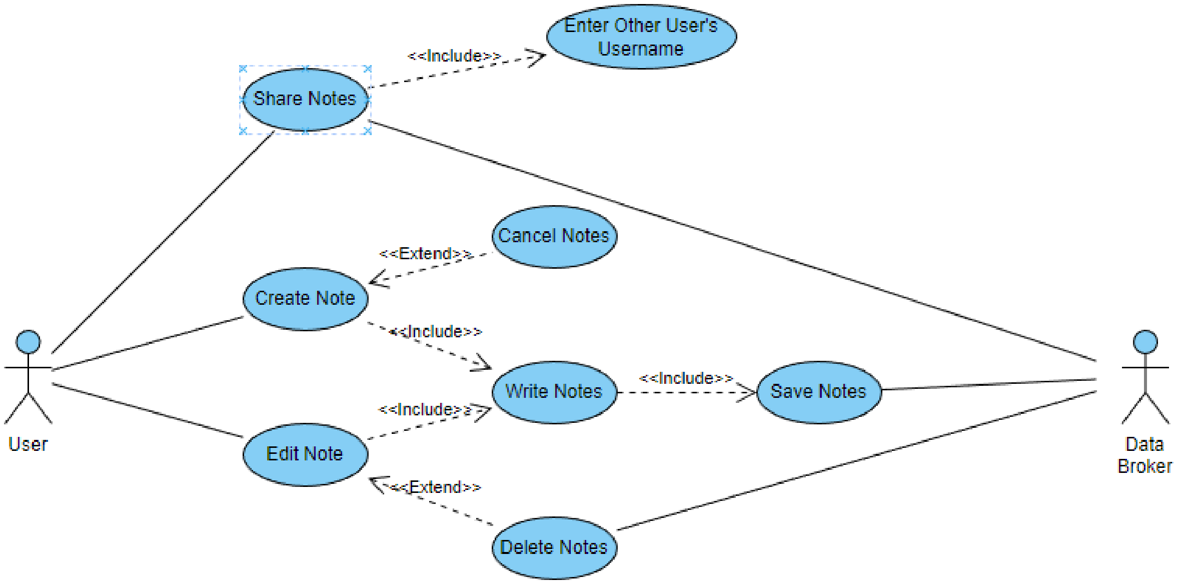
Data Broker The application that is stored on a virtual machine in Azure. This object facilitates communication between front end and database, and handles things like authentication

Database The database stores data related to accounts, courses, notes, and events

5.3 Class Diagrams

5.4 Derived Requirements

Note-Taking



F

ig. 3.1: Note-Taking Use-Case diagram (from Use-Case Specifications Document)

6. Create Notes

6.1 Flow of Events - Design

First the user will click the create notes button, then the front end will load the edit notes page. After submitting new notes information, the notes info will be pushed to the database through the data broker, and a confirmation will make its way back to the front end which will display the notes page.

6.2 Interaction Diagrams

6.2.1 Sequence Diagrams

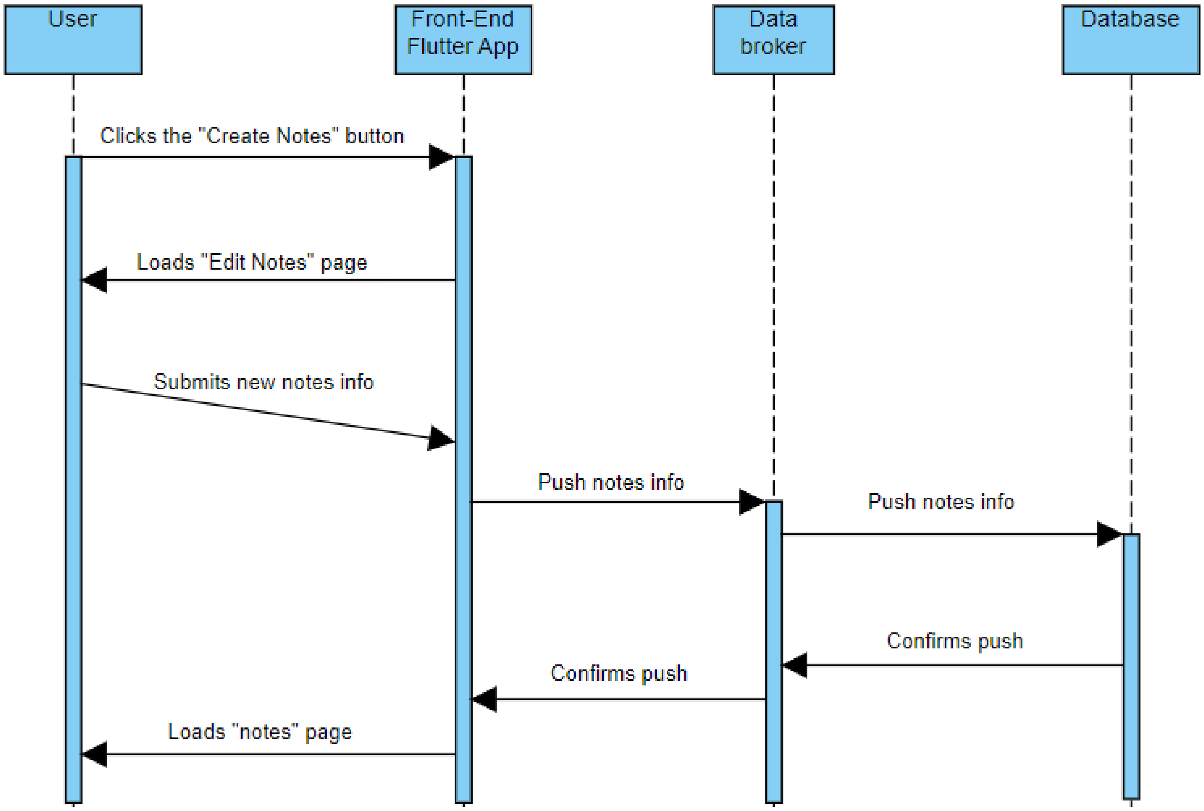


Fig. 3.2: Create Notes Sequence Diagram

6.2.2 Participating objects

Front-End The application that handles UI and basic front-end functionality

Data Broker The application that is stored on a virtual machine in Azure. This object facilitates communication between front end and database, and handles things like authentication

Database The database stores data related to accounts, courses, notes, and events

6.3 Class Diagrams

6.4 Derived Requirements

7. Edit Notes

7.1 Flow of Events - Design

The user will click the edit notes button and similar to the create notes use-case, the front end will load the edit notes page. The user will enter new information for the notes being edited and it will be pushed to the database through the data broker. A confirmation will be sent back to the front end which will display the notes page.

7.2 Interaction Diagrams

7.2.1 Sequence Diagrams

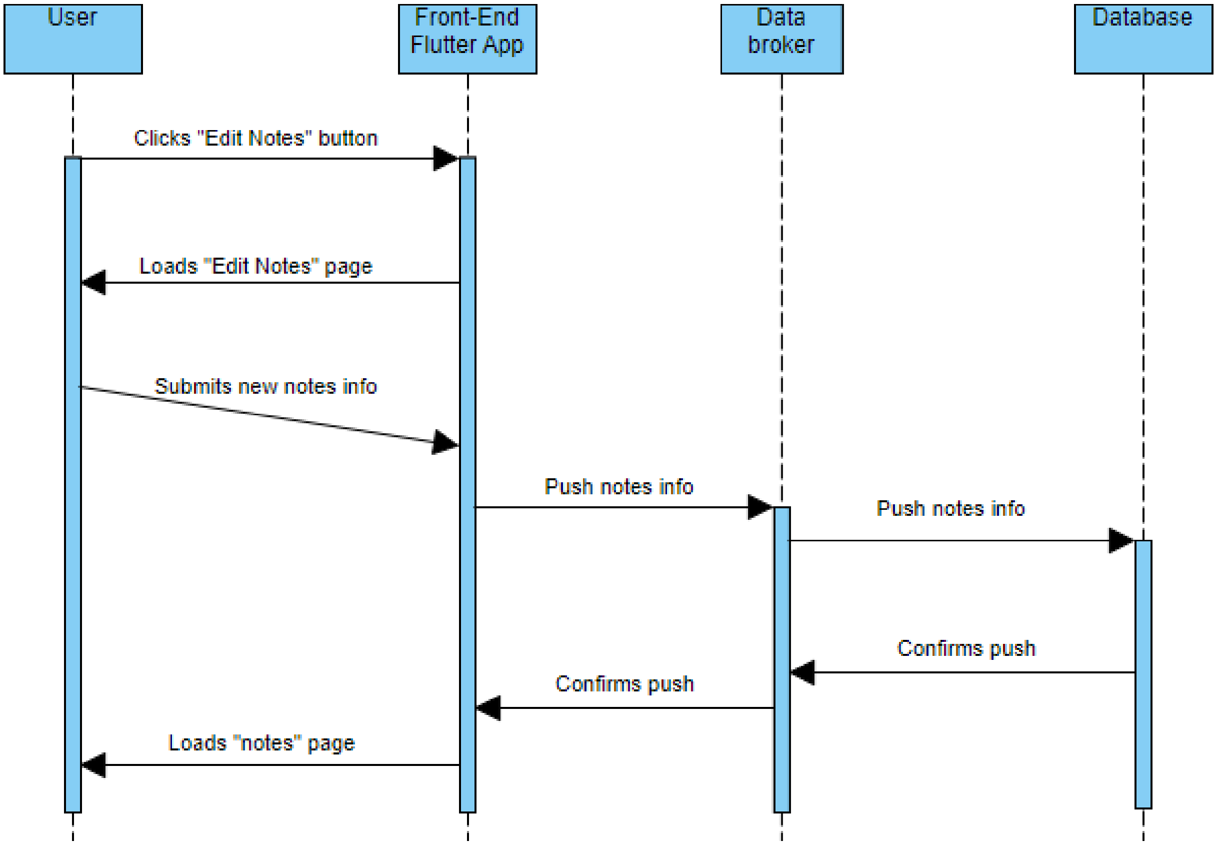


Fig. 3.3: Edit Notes Sequence Diagram

7.2.2 Participating objects

Front-End The application that handles UI and basic front-end functionality

Data Broker The application that is stored on a virtual machine in Azure. This object facilitates communication between front end and database, and handles things like authentication

Database The database stores data related to accounts, courses, notes, and events

7.3 Class Diagrams

7.4 Derived Requirements

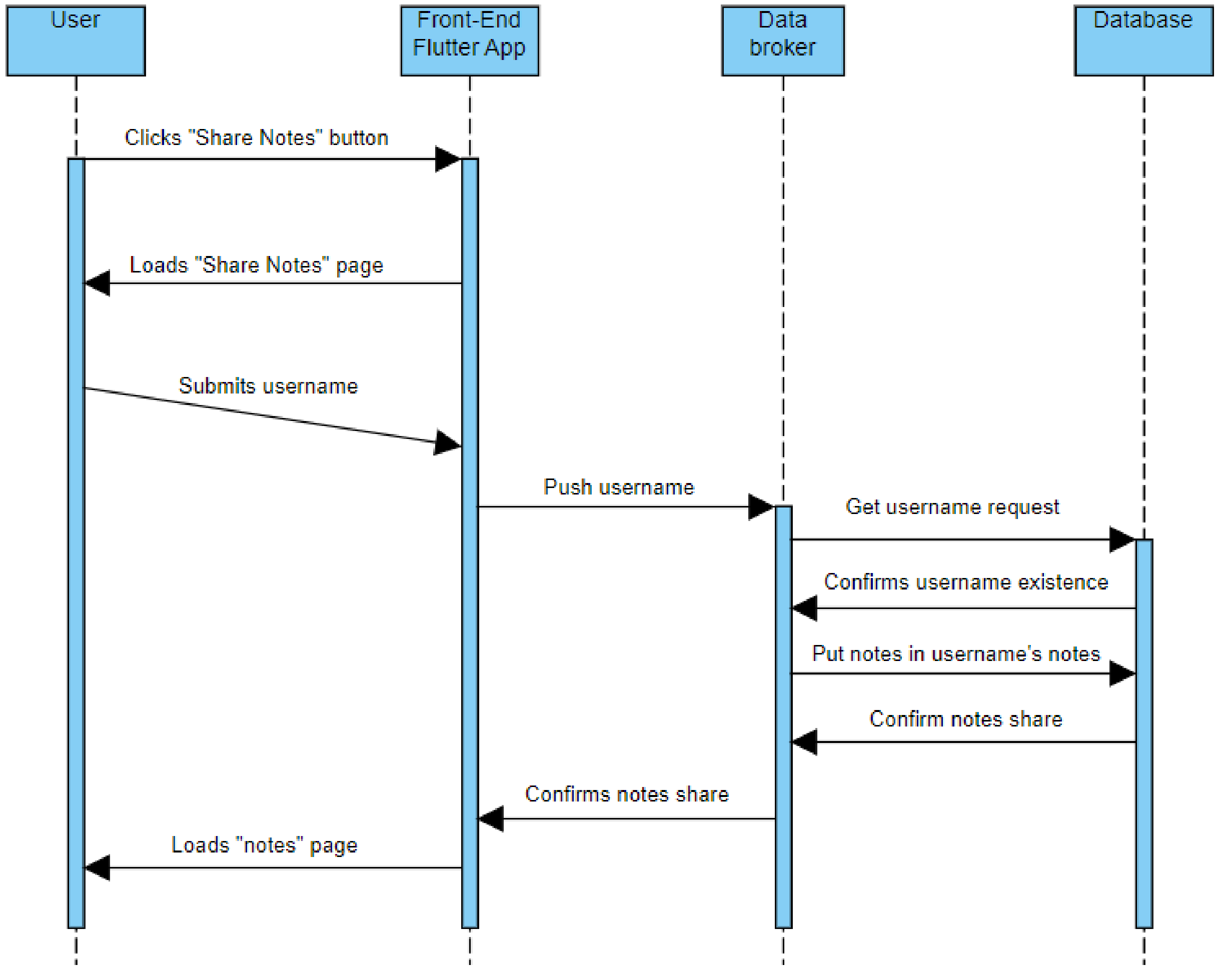
8. Share Notes

8.1 Flow of Events - Design

The user will click the share notes button and the front end will load the share notes page. The user will enter in the username of the person that they want to share notes with. The front end will push the username to the data broker which will attempt to get the username from the database to confirm that it exists. Then the data broker will put the notes in the entered username's notes. Then a confirmation will make its way back to the front end, which will load the notes page.

8.2 Interaction Diagrams

8.2.1 Sequence Diagrams



F

ig. 3.4: Share Notes Sequence Diagram

8.2.2 Participating objects

Front-End The application that handles UI and basic front-end functionality

Data Broker The application that is stored on a virtual machine in Azure. This object facilitates communication between front end and database, and handles things like authentication

Database The database stores data related to accounts, courses, notes, and events

8.3 Class Diagrams

8.4 Derived Requirements

Event Management

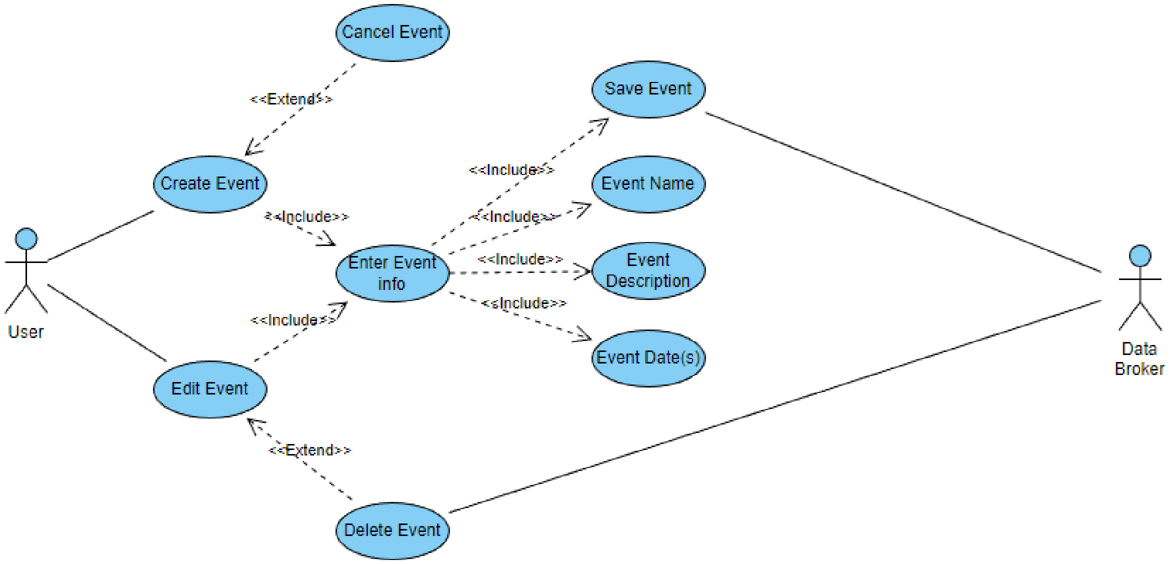


Fig. 4.1: Event Management Use-Case diagram (from Use-Case Specifications Document)

9. Create Event

9.1 Flow of Events - Design

Upon the user clicking the create event button, the front end will load the edit event page. The user will submit the event information and the front end will push the event information to the database through the data broker. A confirmation will make its way back to the front end which will load the calendar page.

9.2 Interaction Diagrams

9.2.1 Sequence Diagrams

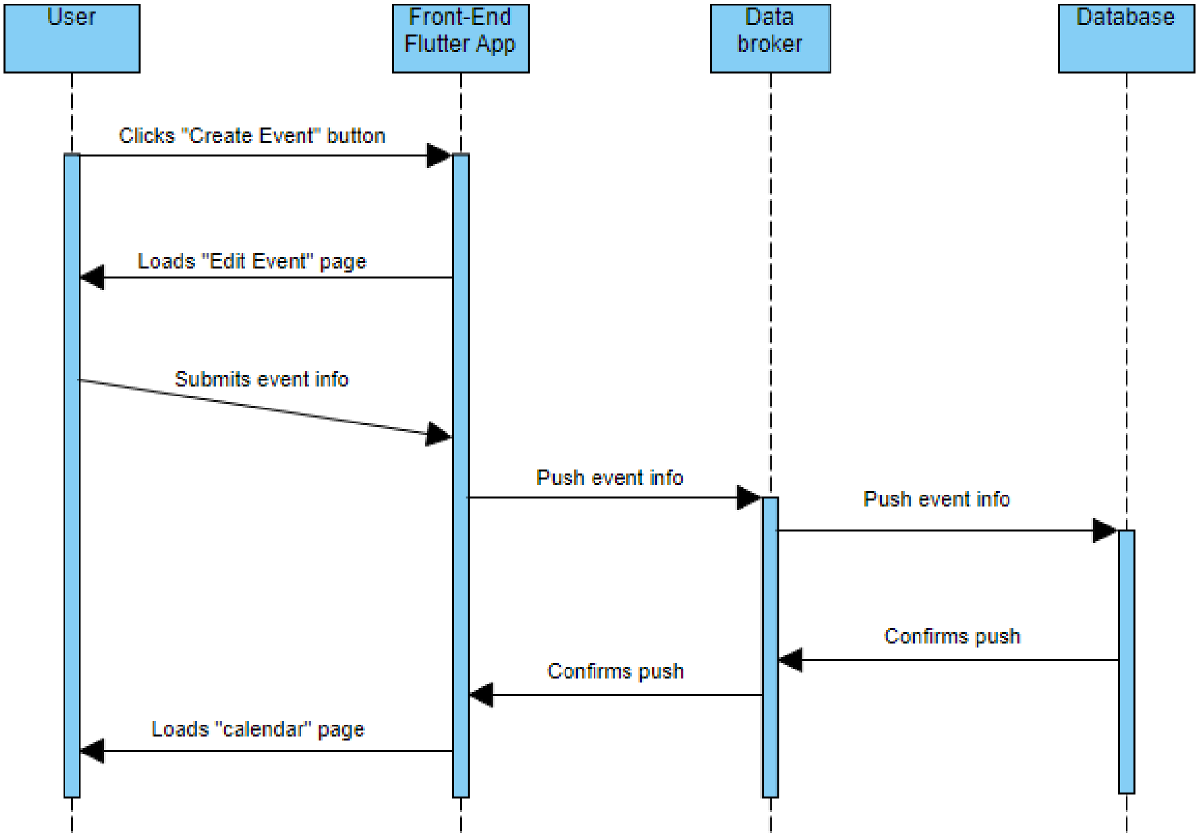


Fig. 4.2: Create Event Sequence Diagram

9.2.2 Participating objects

Front-End The application that handles UI and basic front-end functionality

Data Broker The application that is stored on a virtual machine in Azure. This object facilitates communication between front end and database, and handles things like authentication

Database The database stores data related to accounts, courses, notes, and events

9.3 Class Diagrams

9.4 Derived Requirements

10. Edit Event

10.1 Flow of Events - Design

Upon clicking the edit event button, similarly to the create event use-case, the front end will load the edit event page. The user will submit the new event information which will be pushed to the database through the data broker. A confirmation will make its way back to the front end which will load the calendar page.

10.2 Interaction Diagrams

10.2.1 Sequence Diagrams

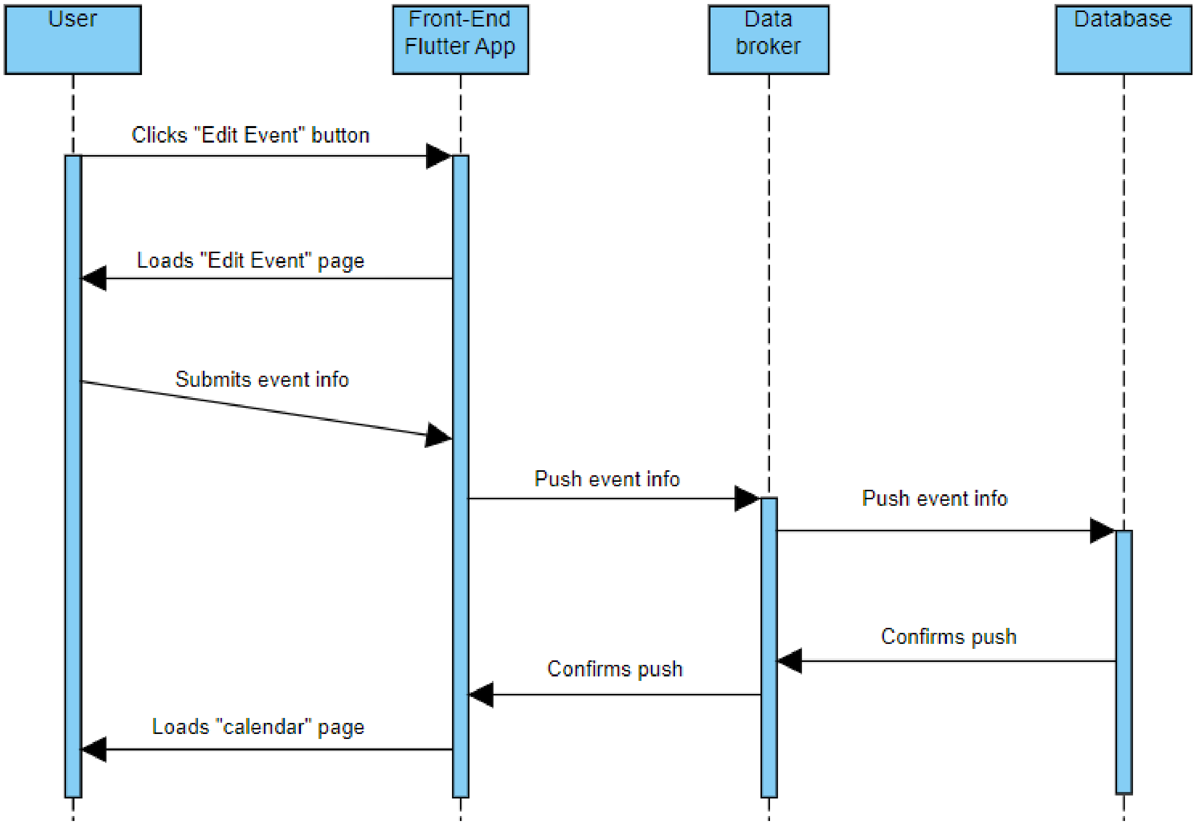


Fig. 4.3: Edit Event Sequence Diagram

10.2.2 Participating objects

Front-End The application that handles UI and basic front-end functionality

Data Broker The application that is stored on a virtual machine in Azure. This object facilitates communication between front end and database, and handles things like authentication

Database The database stores data related to accounts, courses, notes, and events

10.3 Class Diagrams

10.4 Derived Requirements