



**Socialendar**  
**Project Requirement**

Course/CRN	Software Engineering / 13329
Date	28/10/2017
Team Members	Yunus Güngör Sinan Kartal Mustafa Sağlam M.Enes Kayılioğlu Recep Can Babaoğlu

## 1.Introduction

With the rise of social events, people are getting more sociable. They meet each other at social events and organize a plan together. The Socialendar application can bring together people who will join same organizations efficiently, easily and precisely. This report presents how to user use this application, how to run this application, logic of project and relations. The project consists three main section: Usage Scenario, Early System Models, User Stories.

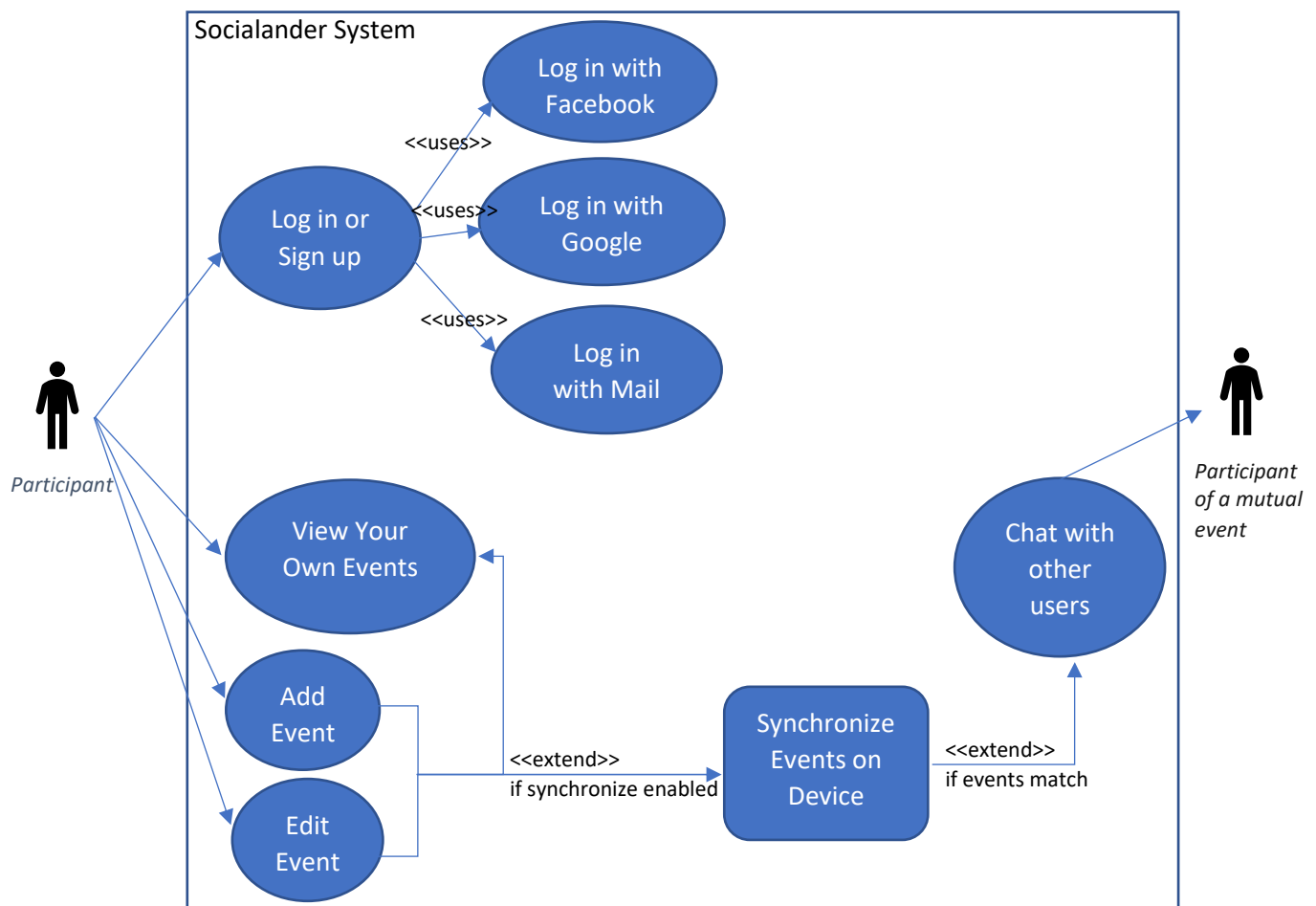
## 2.Usage Scenario

### 2.1 User Types

User Type	Examples	Expected Features
Participant of a recurrent or one-time event	Students	Share information about class in a chatroom
	Academic Staff	Share ideas in an isolated and professional chatroom in a conference
		Keep in touch with academic personal even after a conference
	Bored people at an event like party or concert	Find people with same interest as yours in an event
	Businessmen	Talk about meeting requirements before meeting with attendees.

- This project has only one user type, participant. Every participant also an organizer because participants add events to the system.

## 2.2 Use Case Diagram



- There can be infinite number of participants. But there is a technical limit on how much participant can be matched for an event and give access to chatroom. Each participant also an organizer and a user.

## 2.3 Use cases

### 2.3.1 Use case: Log in or sign up

**Primary actor:** Participant

**Scope:** Starting to use the app

**Level:** User request

**Brief:** The user who want to sign up or already have signed, enters the program. First stage to use program.

**Stakeholders:** User

**Postconditions:**

**Minimal Guarantees:** Anyone who accepts terms and conditions able to enters the system.

**Success Guarantees:** Login and sign up system have already tested before the release. Developer guarantees the login and sign up.

**Preconditions:** Anyone who have Google account, Facebook account or an e-mail can log in. User must have at least one of them.

**Triggers:** The user invokes a request.

**Flow:** The system provides a basic interface which has log in or sign up options. If the already signed user tries to enter, the system will verify the account via Facebook, Google or mail account then makes required calls to program. If new user tries to sign up, system will call to another interface that creates the new account for user. The program will automatically connect the user's Facebook or Google accounts for new account when user accepts.

### 2.3.2 Use case: Chat with another user

**Primary actor:** User who has a common event with other users

**Scope:** Chatting with each other

**Level:** System, User

**Brief:** User who has a common event with other users can enter the same chatroom for chatting.

**Stakeholders:** Logged in users who have same event on calendar

**Postconditions:**

**Minimal Guarantees:** Anyone who accepts the enter chatroom able to chat.

**Success Guarantees:** Systems creates chatrooms when notice the new event. Also, system guarantees to add user if calendar gets new event.

**Preconditions:** Event chatroom must be created. Also, user's calendar must have the event.

**Triggers:** The system invokes a request if room has not created yet. Additionally, if room already exist another request will be invoked.

**Flow:** The system checks the all users calendar if new event noticed, new chatroom will be created by the system. When user adds an event that is not existing, system checks the event information if event appropriates to terms and provides conditions new chatroom will be created by the system. Furthermore, the user can participate an already existing event. Naturally, system checks the user profile then, if profile eligible for chatroom user will be added the room.

### 2.3.3 Use case: Add event

**Primary actor:** User who has permission

**Scope:** Adding event to user's calendar

**Level:** User

**Brief:** The user able to add event to calendar.

**Stakeholders:** User who want to add event

**Postconditions:**

**Minimal Guarantees:** Any user who provides essential information about event able to add event.

**Success Guarantees:** Mobile phones calendar application provides adding event. System also guarantees synchronizing event to the cloud and matching with events that has a high similarity points.

**Preconditions:** Predetermined and obligatory features of event must be full-filled including Title, category and time of the event.

**Triggers:** The user invokes a request to add event, system checks.

**Flow:** User must open the calendar application. Then, he can easily add new event to calendar same as the old calendar. Additionally, some features of events are obligatory like date, time and title. System gets the request then, checks the features if it is suitable, adds to the database and device calendar. After that checks the existing events if they match user will be added chatroom as already mentioned. Moreover, if event is new system holds the event record.

#### 2.3.4 Use case: Edit event

**Primary actor:** User who has permission

**Scope:** Editing an event on user's calendar

**Level:** User

**Brief:** The user able to edit event on their calendar.

**Stakeholders:** User who want to edit an event

**Postconditions:**

**Minimal Guarantees:** Any user who already has an event on their calendar, can edit their events.

**Success Guarantees:** Mobile phones calendar application provides editing event. System also guarantees synchronizing event to the cloud and matching with events that has a high similarity points after editing.

**Preconditions:** User must have been added an event. Predetermined and obligatory features of event must be full-filled on unchanged including Title, category and time of the event.

**Triggers:** The user invokes a request to edit event, system checks.

**Flow:** User must open the calendar application. Then, he can easily invoke a request to edit an event while viewing that event. Additionally, some features of events are obligatory like date, time and title. System gets the request then, checks the features if it is suitable, adds to the database and device calendar. After that checks the existing events if they match; user will be added chatroom as already mentioned and will be removed from old chatroom which matched with previous event.

### 2.3.5 Use case: View event

**Primary actor:** User who has permission

**Scope:** Viewing an event on user's calendar

**Level:** User

**Brief:** The user able to view event on their calendar.

**Stakeholders:** User who want to get information about an event

**Postconditions:**

**Minimal Guarantees:** Any user who already has an event on their calendar, can view their events.

**Success Guarantees:** Mobile phones calendar application provides viewing event. System also adds additional info about weather.

**Preconditions:** User must have been added an event. Predetermined and obligatory features of event must be full-filled on unchanged including Title, category and time of the event.

**Triggers:** The user invokes a request to view event, system returns event information.

**Flow:** User must open the calendar application. Then, main page fetches information about user's event. User can invoke a request to view an event simply by touching that information. After request, system returns the event information. After returning event information, system checks weather for the event time and loads this information. After loading information displayed to the user.

### 2.3.6 Use case: View user's next available time

**Primary actor:** User who has permission

**Scope:** Viewing next available time on calendar

**Level:** User

**Brief:** The user able to view next available time on calendar.

**Stakeholders:** User who want to get information about their available time

**Postconditions:**

**Minimal Guarantees:** Any user who already has an event which did not occur yet on their calendar, can view their next available time.

**Success Guarantees:** System will calculate available times on user's calendars and will show the closest one to the user.

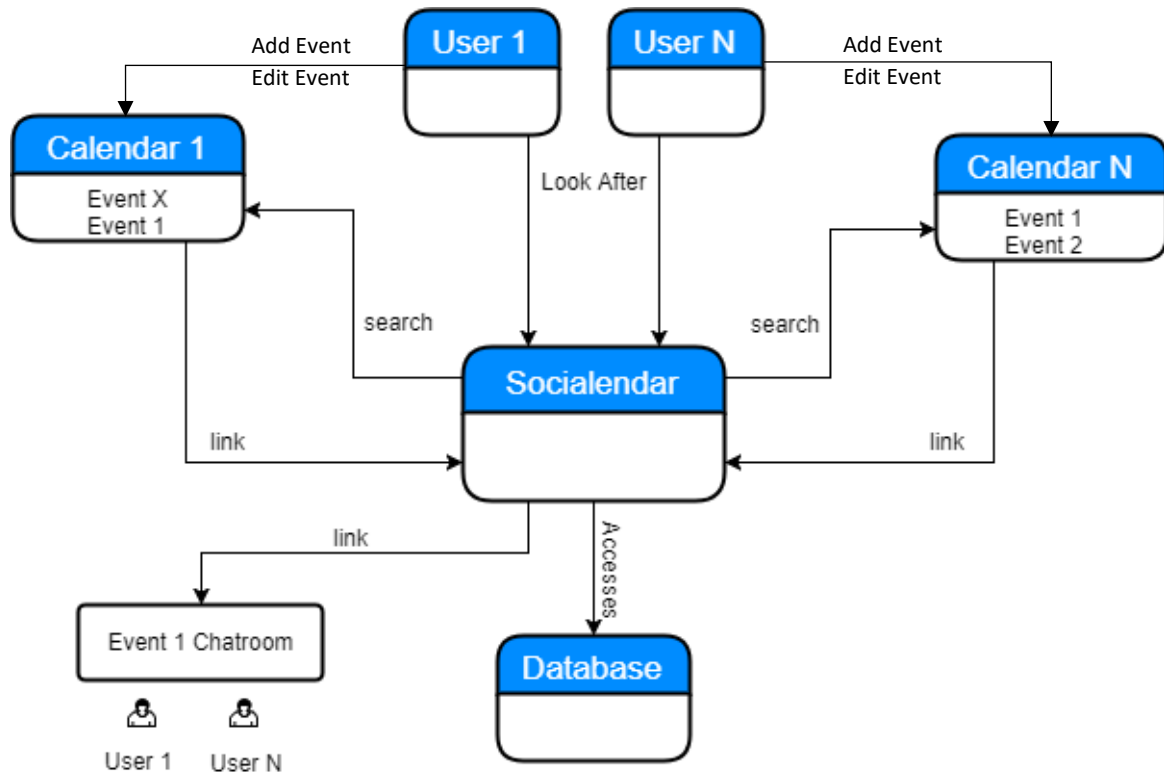
**Preconditions:** User must have been added an event which did not occur yet.

**Triggers:** The user invokes a request to view next available time, system returns available times.

**Flow:** User must open the calendar application. Then, main page fetches information about user's available times. Then shows next available time to the user.

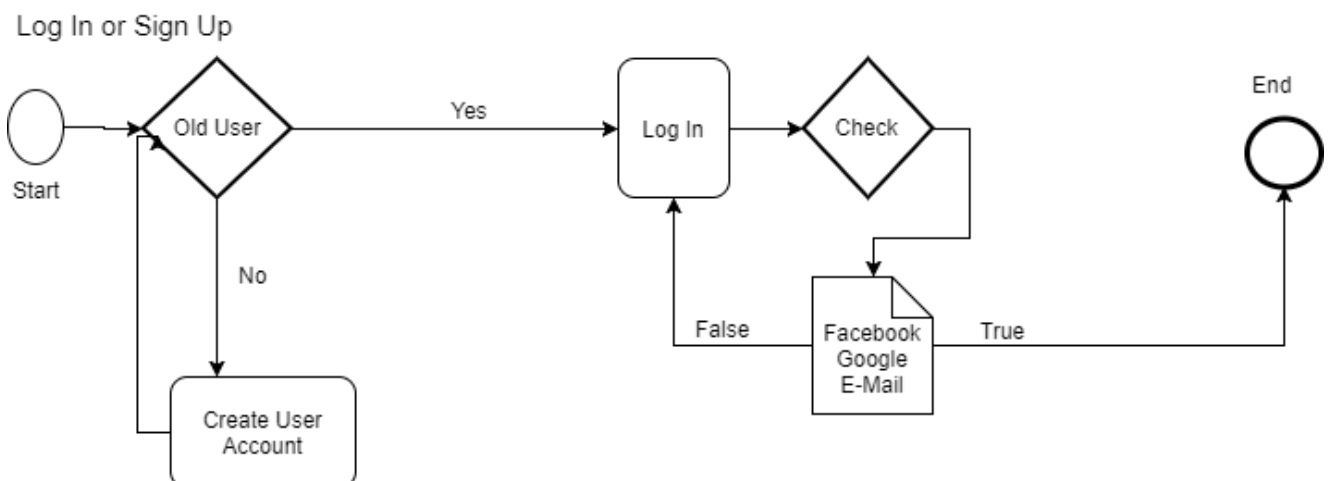
### 3. EARLY SYSTEM MODELS

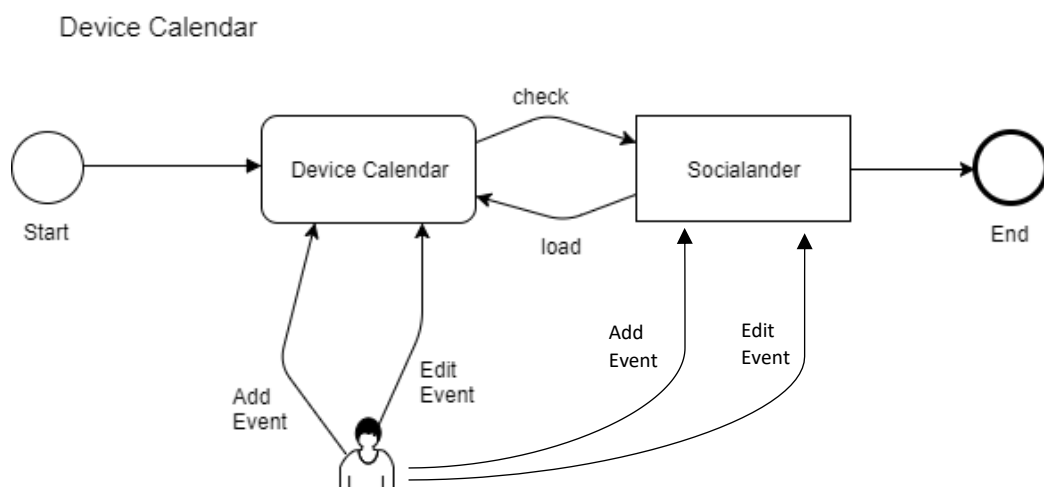
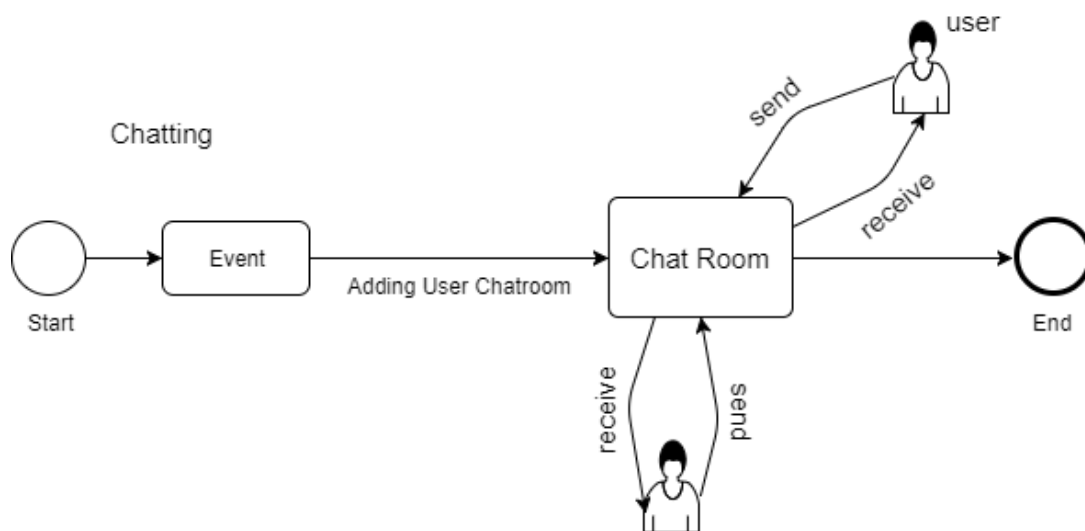
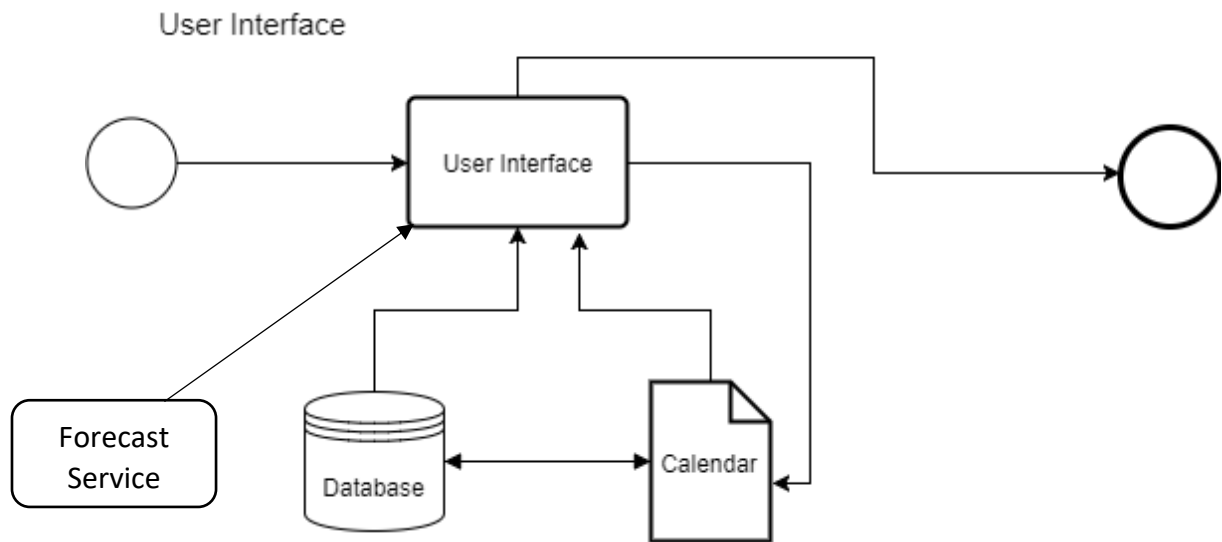
#### 3.1 Conceptual model



- There can be infinite number of users (N).
- There can be infinite number of events which belongs to a user (X).
- Each user has a specific calendar and each calendar has events in it.

#### 3.2 Flow Diagrams







## 4. User Stories

### 4.1 Log In Page

Feature: Log in

Users should not have access to any data until they log in with an existing account

A failed login must display an error message

Scenario: Login with a Facebook account

Given user has a Facebook account

And user has an account in system

When user presses Log in with Facebook button

Then authenticate using Facebook Authentication API

Scenario: Login with a Google account

Given user has a Google account

And user has an account in system

When user presses Sign in with Google button

Then authenticate using Google Authentication API

Scenario: Login with any mail account

Given user has a mail account

And user has an account in system

When user enters credentials for email and password

And presses Log in/Sign up button

Then authenticate using provided credentials

Scenario: Login with any mail account

Given user has a mail account

And user has an account in system

When user enters credentials for email and password

And presses Log in/Sign up button

Then authenticate using provided credentials

Scenario: Login automatically

Given user has an account in system

And user has already logged on to the system on their device

When user opens app

Then authenticate using existing credentials from previous login

Scenario: Login failed

Given user enter credentials wrong for a user

When user presses Log in/Sign up button

Then display an error message

And refresh log in page

Scenario: Login with Google failed

Given user pressed Sign in with Google button

When Google Authentication API returns an error

Then display an error message

And refresh log in page

Scenario: Login with Facebook failed

Given user pressed Log in with Facebook button

When Facebook Authentication API returns an error

Then display an error message

And refresh log in page

Feature: Sign up

Users should not have access to any data until they log in with an existing account

Users should be able to create a new account if they don't have one

Users must be logged in automatically after signing up

A failed sign up must display an error message

Scenario: Sign up with a Facebook account

Given user has a Facebook account

And user does not have an account in system

When user presses Log in with Facebook button

Then authenticate using Facebook Authentication API

And add user to the system

And log in user to the system

Scenario: Sign up with a Google account

Given user has a Google account

And user does not have an account in system

When user presses Sign in with Google button

Then authenticate using Google Authentication API

And add user to the system

And log in user to the system

Scenario: Sign up with any mail account

Given user has a mail account

And user does not have an account in system

When user enter credentials for email and password

And user presses Log in/Sign up button

Then add user to the system with given credentials

And log in user to the system with given credentials

Scenario: Sign up failed

Given tried to sign up

When sign up failed

Then display an error message

And refresh log in page

## 4.2 Home Page

Feature: View events

Users should be able to view their events on homepage

Scenario: View events

Given user has been logged in

When user gets to the homepage

Then fetch user's events from database

And show on page

Feature: View next available time

Users should be able to view when they are available (when they don't have an event)

Scenario: View next available time

Given user has been logged in

When user gets to the homepage

Then fetch user's events from database and calculate their available times

And show available times on UI starting from the closest one.

Feature: View recent chats

Users should be able to view their chat messages in short texts on homepage

Scenario: View recent chats

Given user has been logged in

When user gets to the homepage

Then fetch user's recent chats

And show on the UI starting from the closest one

Feature: View add event page

Users should be able to access add event page from home screen

Scenario: View add event page

Given user has been logged in

When user presses add event button

Then display add event page

Feature: View event page

Users should be able to access event page from home screen

Scenario: View event page

Given user has been logged in

When user presses on an event

Then display event page with pressed event's information

### 4.3 Add Event Page

#### Feature: Add event

Users should be able to add an event to their calendars

Added event must be synchronized with database

Added events must be matched with other users' events and linked with a chatroom

An error message must be displayed if adding fails

#### Scenario: Add event with required information

Given user has been logged in

And user has been entered required information

When user presses on add event button

Then add event to the device calendar

And add event to the database

And match event with other user's events to find appropriate chatroom

#### Scenario: Add event with missing information

Given user has been logged in

And user has not been entered required information

When user presses on add event button

Then display an error message

### 4.4 Event Page

#### Feature: Display Event Information

Users should be able display information about a event

Users should be able to view forecast on the event time

#### Scenario: Display event page

Given user has been logged in

When event page displayed

Then fetch event information from database

And fetch forecast information from a forecast service

And show acquired information on event page

Feature: Access chatroom

Users should be able view event's chatroom from event page

Scenario: Display event's chatroom

Given user has been logged in

And user displaying event page

When user presses chat button

Then display chat page

Feature: Edit Event

Users should be able to edit given information before

Edited event must be synchronized with database

Previous link to chatroom must be dismissed

Edited events must be matched with other users' events and linked with a chatroom

An error message must be displayed if editing fails

Scenario: Edit an event with required information

Given user has been logged in

And user pressed edit button

And user entered required information

When user presses done

Then change information on device calendar

And unlink event from chatroom

And change information on database

And match event with other user's events to find appropriate chatroom

And refresh view event page

Scenario: Edit event with missing information

Given user has been logged in

And user pressed edit button

And user has not been entered required information

When user presses on done button

Then display an error message

#### 4.5 Chatroom

##### Feature: Chatting

Users should be able to send and receive messages

##### Scenario: Send messages to chatroom online

Given user has been logged in

And displaying chatroom page

And user entered a message

When user presses send

Then publish message to chat service

##### Scenario: Send messages to chatroom offline

Given user has been logged in

And displaying chatroom page

And user entered a message

And user does not have internet access

When user presses send

Then publish message to chat service

And display an icon that means message will be send when user got online

##### Scenario: Receive messages

Given user has been logged in

And other users in chatroom published messages

When user displays chatroom page

Then fetch messages

And display on page