# Personal Calendar

## Team 6: Iteration 2

| | |
|---|---|
| William Hilliard | (1000885126) |
| Caesar Llaga | (1000939172) |
| Cyril Varughese | (1000609828) |
| Peggy Soh | (1000873075) |

# Requirements

# Requirements

**1.** The calendar shall allow the user to switch between monthly, weekly and daily views.

- **1.1.** The calendar shall allow the user to view past events.
  - **1.1.1.** The current date shall be highlighted.
- **1.2.** The monthly view shall have a change month option.
- **1.3.** The weekly view shall have a change week option.
- **1.4.** The daily view shall have a change day option.
- **1.5.** The views shall allow user to quickly go to today.

**2.** The calendar shall have a summary view.

- **2.1.** The summary view shall have an option to show the next ten events.
- **2.2.** The summary view shall have an option to show all previous events.

**3.** The calendar shall allow the user to add an event.

- **3.1.** The calendar shall allow an event to be an all-day event.
- **3.2**. The calendar shall allow an event to have a start and end time.

# Requirements Continue

**3.3.** The calendar shall allow an event to have an optional description.

**3.4.** The calendar shall allow an event to have an optional location.

**3.5.** The calendar shall allow the user to schedule repeating events.

**3.6.** The calendar shall allow the user to enter emails of participants for an event.

**4.** The calendar shall allow the user to remove an event.

**5.** The calendar shall allow the user to edit an event.

**6.** The calendar shall allow the user to send email reminders.

**6.1.** The calendar shall send an email to the selected event participants.

**6.2.** The calendar shall send an email to the user.

**7.** The calendar shall allow users to customize their calendars with different settings.

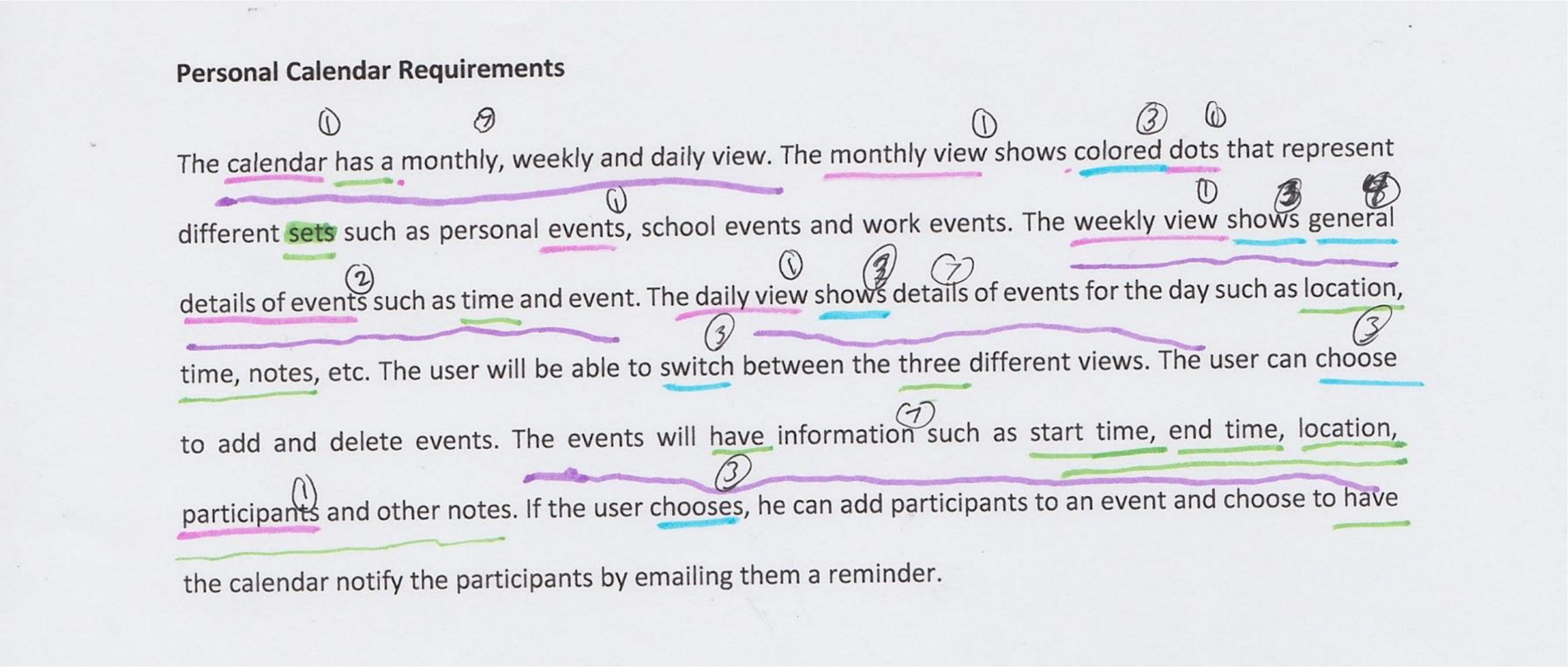
**7.1.** The settings shall allow users to change and personalize the calendar.

# Domain Modeling

# Brainstorming

1. ● Noun/ Noun Phrases
2. ● X of Y
3. ● Transitive Verbs
4. ● Adjectives
5. ● Numerics/ Enumerations
6. ● Possession Expressions
7. ● Constituents
8. ● Containment
9. ● X is a Y

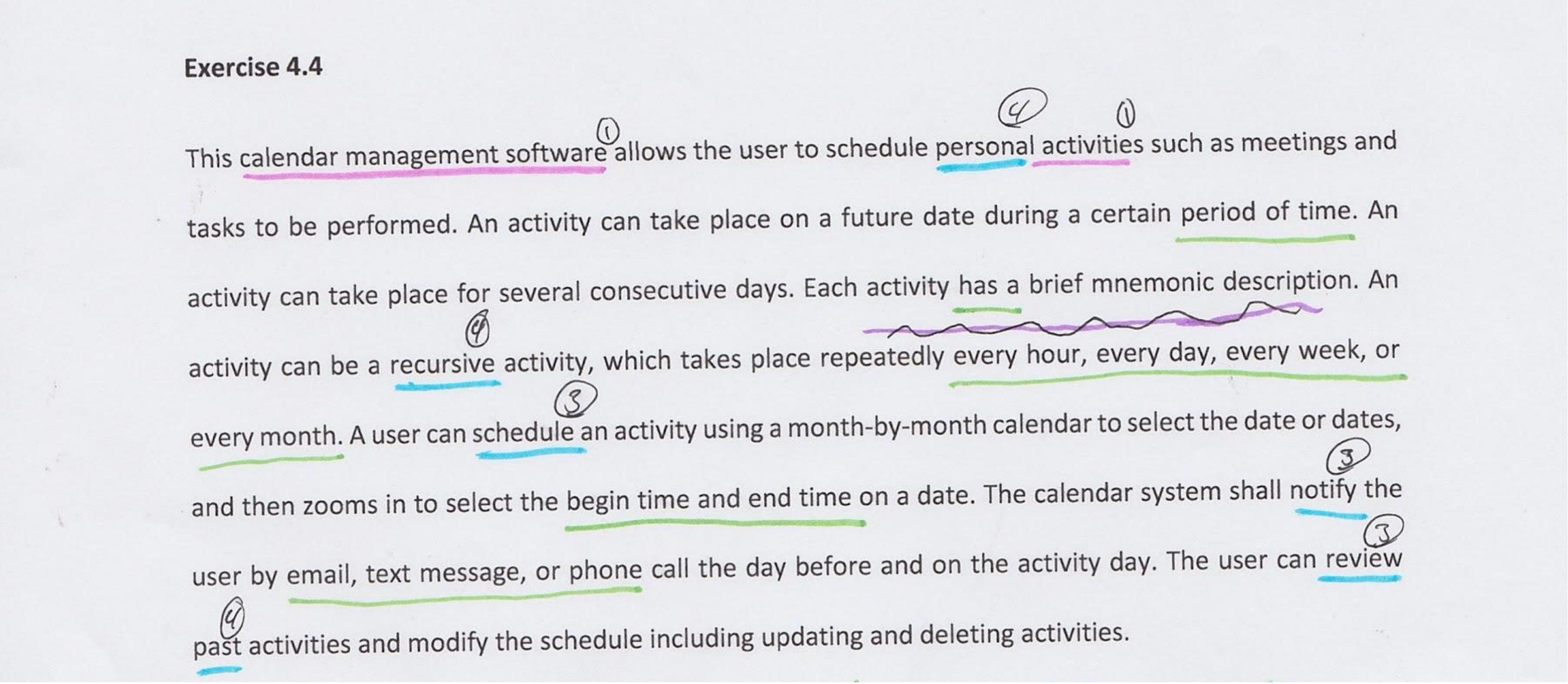
# Brainstorming Continue



**Personal Calendar Requirements**

The calendar has a monthly, weekly and daily view. The monthly view shows colored dots that represent different sets such as personal events, school events and work events. The weekly view shows general details of events such as time and event. The daily view shows details of events for the day such as location, time, notes, etc. The user will be able to switch between the three different views. The user can choose to add and delete events. The events will have information such as start time, end time, location, participants and other notes. If the user chooses, he can add participants to an event and choose to have the calendar notify the participants by emailing them a reminder.

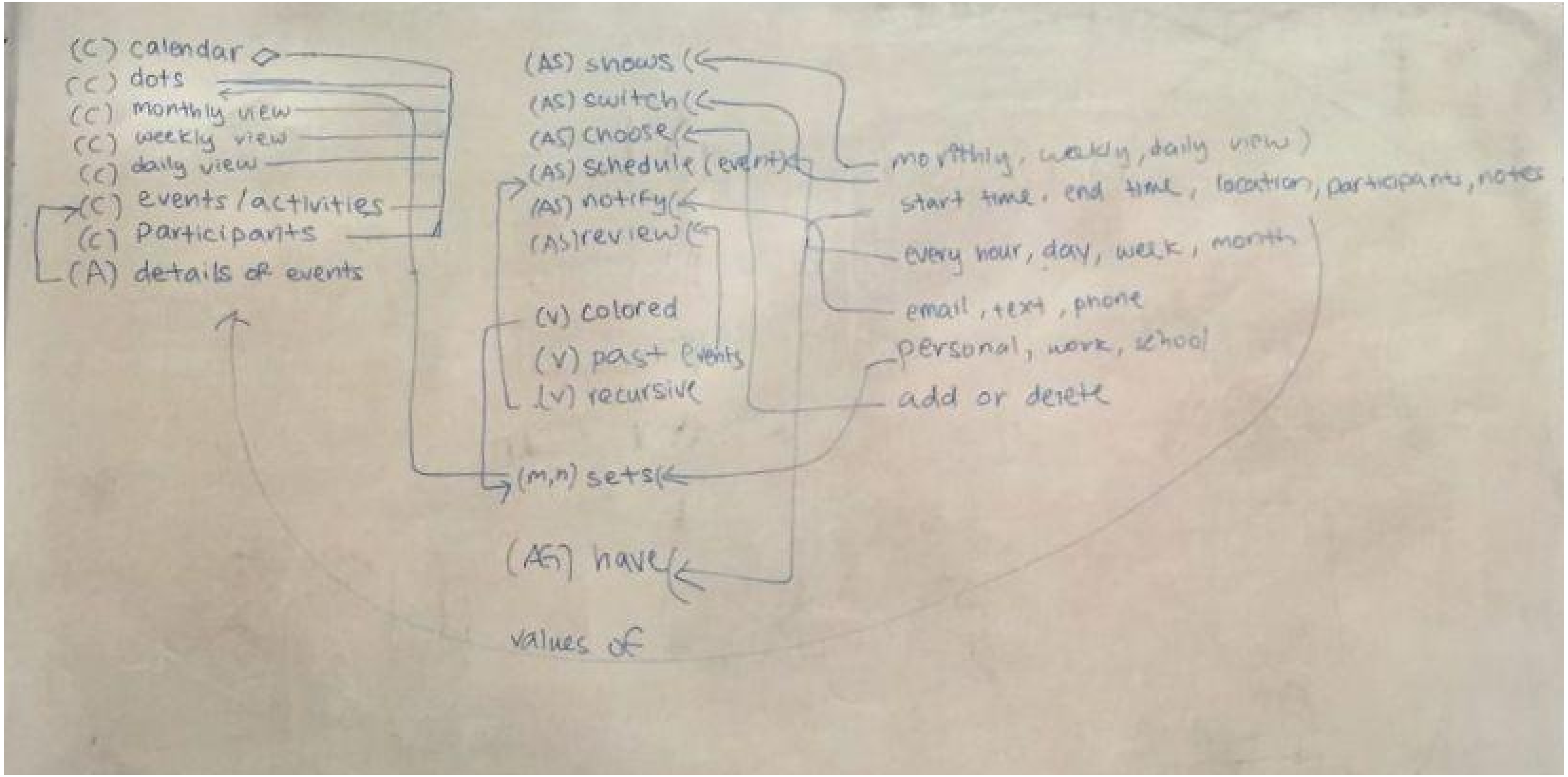
# Brainstorming Continue



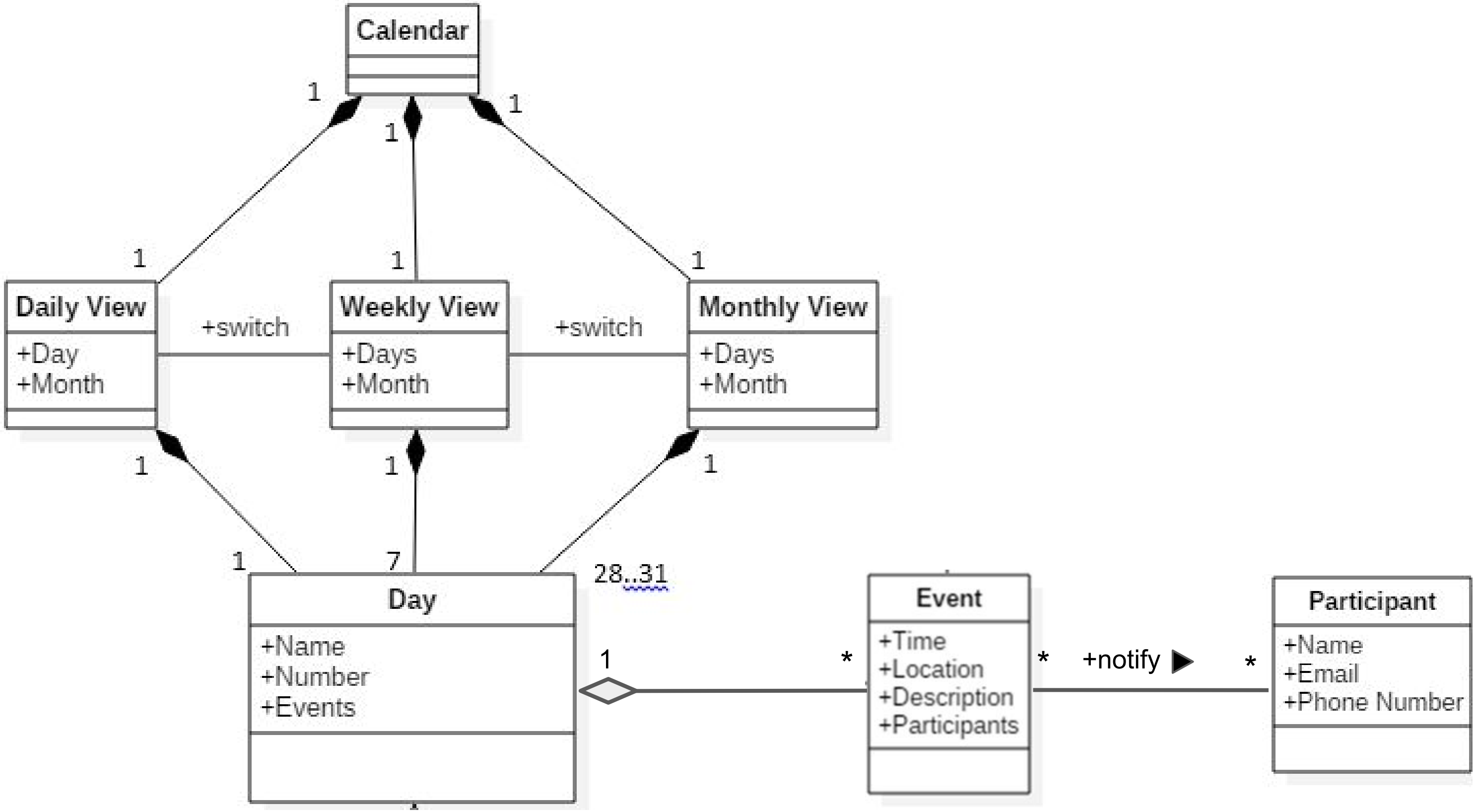
**Exercise 4.4**

This calendar management software ① allows the user to schedule personal ④ activities ① such as meetings and tasks to be performed. An activity can take place on a future date during a certain period of time. An activity can take place for several consecutive days. Each activity has a brief mnemonic description. An activity can be a recursive ④ activity, which takes place repeatedly every hour, every day, every week, or every month. A user can schedule ③ an activity using a month-by-month calendar to select the date or dates, and then zooms in to select the begin time and end time on a date. The calendar system shall notify ③ the user by email, text message, or phone call the day before and on the activity day. The user can review ③ past ④ activities and modify the schedule including updating and deleting activities.

# Classification



# Domain Model Class Diagram



# Use Case Modeling

# Abstract Use Cases

**UC1**. Add event (Actor: User, System: Personal Calendar)

**UC2.** Share event (Actor: User, System: Personal Calendar)

**UC3**. Edit event (Actor: User, System: Personal Calendar)

**UC4**. Delete event (Actor: User, System: Personal Calendar)

**UC5.** Add contact (Actor: User, System: Personal Calendar)

**UC6.** Delete contact (Actor: User, System: Personal Calendar)

**UC7.** Edit contact (Actor: User, System: Personal Calendar)

**UC8.** Customize calendar (Actor: User, System: Personal Calendar)

# High-Level Use Cases

**UC1.** Add Event

TUCBW the user clicks on “Add Event” button on the calendar.
TUCEW the user sees the newly added event on the calendar.

**UC2.** Share Event

TUCBW the user clicks on “Share” button on the event detail window.
TUCEW the user sees the email application open filled with event details.

**UC3.** Edit Event

TUCBW the user clicks the event to edit on the event detail window.
TUCEW the user sees the changes of the event details have been updated on the calendar.

**UC4.** Delete Event

TUCBW the user clicks “Delete” button on the event detail window..
TUCEW the user sees the event has been removed from the calendar.

# High-Level Use Cases Continued

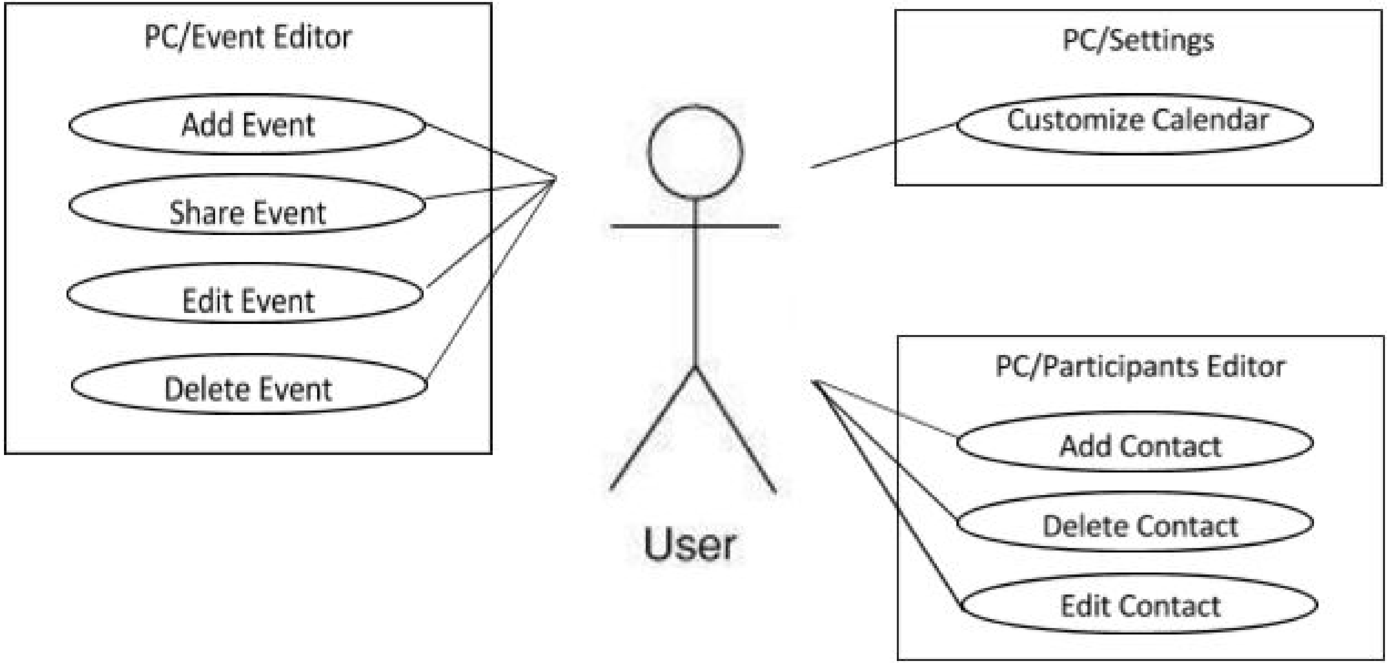
**UC5.** Add Contact
TUCBW the user clicks “Add Contact”.
TUCEW the user sees the contact added to the list.

**UC6.** Delete Contact
TUCBW the user clicks “Delete Contact”.
TUCEW the user sees the contact is removed from the list.

**UC7.** Edit Contact
TUCBW the user clicks “Edit Contact”.
TUCEW the user sees the contact information has been updated.

**UC8.** Customize Calendar
TUCBW the user clicks “Settings”.
TUCEW the user sees the changes made updated on the calendar.

# Use Case Diagram



# Expanded Use Case (UC1: Add Event)

| | |
|---|---|
| Precondition: None | |
| Actor: User | System: Personal Calendar |
| | 0. System displays a add event button on the lower right corner of the calendar view. |
| 1. TUCBW user clicks on the add event button. | 2. System displays a create event form for the event name, start time, end time, description, location, and participants. |
| 3. User enters event name, start time, end time, description, location, and participants and clicks the “Save’” button. | 4. System saves the entered fields into the storage and displays a message confirming that event has been saved, or else the system displays a time conflict message. |
| 5. TUCEW user sees the confirmation message and the newly added event on the calendar. | |
| Post condition: None | |

# Expanded Use Case (UC2: Share Event)

| | |
|---|---|
| Precondition: User must be in event detail view and there must be participant(s) in the event. | |
| Actor: User | System: Personal Calendar |
| | 0. System displays a share button on the event details page. |
| 1. TUCBW user clicks on the share button. | 2. System displays a list of participants for event. |
| 3. User selects the participants to send reminder email to. | 4. System displays a list of applications for user to select to send the email. |
| 5. User selects an email application. | 6. System fills email with event name, time, location and description to selected participants. |
| 7. TUCEW user sees the the email with event details and selected participants filled out in the selected email application. | |
| Post condition: None | |

# Expanded Use Case (UC3: Edit Event)

| | |
|---|---|
| Precondition: There must be an existing event. | |
| Actor: User | System: Personal Calendar |
| | 0. System displays the calendar view. |
| 1. TUCBW the user selects an event in the calendar view. | 2. System displays the event details (event name, time, location) in the event details view. |
| 3. User modifies the appropriate fields with new information and clicks “Save” button. | 4. System updates event and displays a confirmation message. |
| 5. TUCEW the user sees the confirmation message and the event details updated. | |
| Post condition: None | |

# Expanded Use Case (UC4: Delete Event)

| Precondition: There must be an existing event. | |
|---|---|
| Actor: User | System: Personal Calendar |
| | 0. System displays the calendar view. |
| 1. TUCBW the user selects an event in the calendar view. | 2. System displays the event details (event name, time, location) and “Delete” button in the event details view. |
| 3. User clicks the delete button. | 4. System displays a message asking to delete event. |
| 5. User clicks the “Confirm” button on the message. | 6. System removes the event from the calendar view and displays a confirmation message. |
| 7. TUCEW the user sees the confirmation message and the event is removed from the calendar view. | |
| Post condition: None | |

# Requirements - Use Cases

| Requirement | Priority | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 | UC8 |
|---|---|---|---|---|---|---|---|---|---|
| R1 | 6 | | | | | | | | X |
| R2 | 5 | | | | | | | | X |
| R3 | 1 | X | | | | X | X | X | |
| R4 | 3 | | | | X | | | | |
| R5 | 2 | | | X | | | | | |
| R6 | 4 | | X | | | | | | |
| R7 | 7 | | | | | | | | X |
| UC Priority | | 1 | 4 | 2 | 3 | 1 | 1 | 1 | 5 |

# Object Interaction Modeling

# Scenarios (Add Event)

UC1: Add Event

**1.** *User enters event name, date, start-time, end-time, location, participants in view window.

**2.** Event creator searches for available time slot in calendar storage.

**2.1.** If storage has availability during given time frame

**2.1.1.** Event creator saves newly entered data in calendar storage.

**2.1.2.** View window displays a confirmation message.

**2.2.** Else

**2.2.1.** View window displays a time conflict message.

# Scenarios (Share Event)

UC2: Share Event

**1.** *User selects an event and clicks the share button.

**2.** Share event controller fetches event name, date, start-time, end-time, location and participants.

**3.** View window display a list of participants for user to select.

**4.** Share event controller gathers emails of selected participants.

**5.** Share event controller opens an email application and sends event name, date, start-time, end-time and location to selected participants.

# Scenarios (Edit Event)

UC3: Edit Event

**1.** *User selects an event for editing.

**2.** Edit event controller fetches event name, date, start-time, end-time, location, participants.

**3.** View window displays event name, date, start-time, end-time, location and participants for user to edit.

**4.** If storage has availability during given new time frame

- **4.1.** Edit event controller updates event in new time slot in calendar storage.
- **4.2.** View window displays a confirmation message

**5.** Else

- **5.1.** View window displays a time conflict message.

# Scenarios (Delete Event)

UC4: Delete Event

**1.** *User selects an event and clicks the delete button.

**2.** Delete event controller searches for event in calendar storage.

**3.** Delete event controller deletes event data in calendar storage.

**4.** View window displays a confirmation message.

# Scenario Table (UC1: Add Event)

| # | Subject | Subject Action | Other Data/Objects | Object Acted Upon |
|---|---|---|---|---|
| 1. | User | enters | event name, date, start-time, end-time, location, participants | view window |
| 2. | Event creator | searches | time slot | calendar storage |
| 2.1. | If storage has availability during given time frame | | | |
| 2.1.1. | Event creator | saves | event name, date, start-time, end-time, location, participants | calendar storage |
| 2.1.2. | View window | displays | “event added” | message |
| 2.2. | Else | | | |
| 2.2.1. | View window | displays | “time conflict” | message |

# Scenario Table (UC2: Share Event)

| # | Subject | Subject Action | Other Data/Objects | Object Acted Upon |
|---|---|---|---|---|
| 1. | User | selects | share button | event |
| 2. | Share event controller | fetches |  | event name, date, start-time, end-time, location, participants |
| 3. | View window | display | user | participants |
| 4. | Share event controller | gathers | participants | emails |
| 5. | Share event controller | opens | event name, date, start-time, end-time, location, participants | email application |

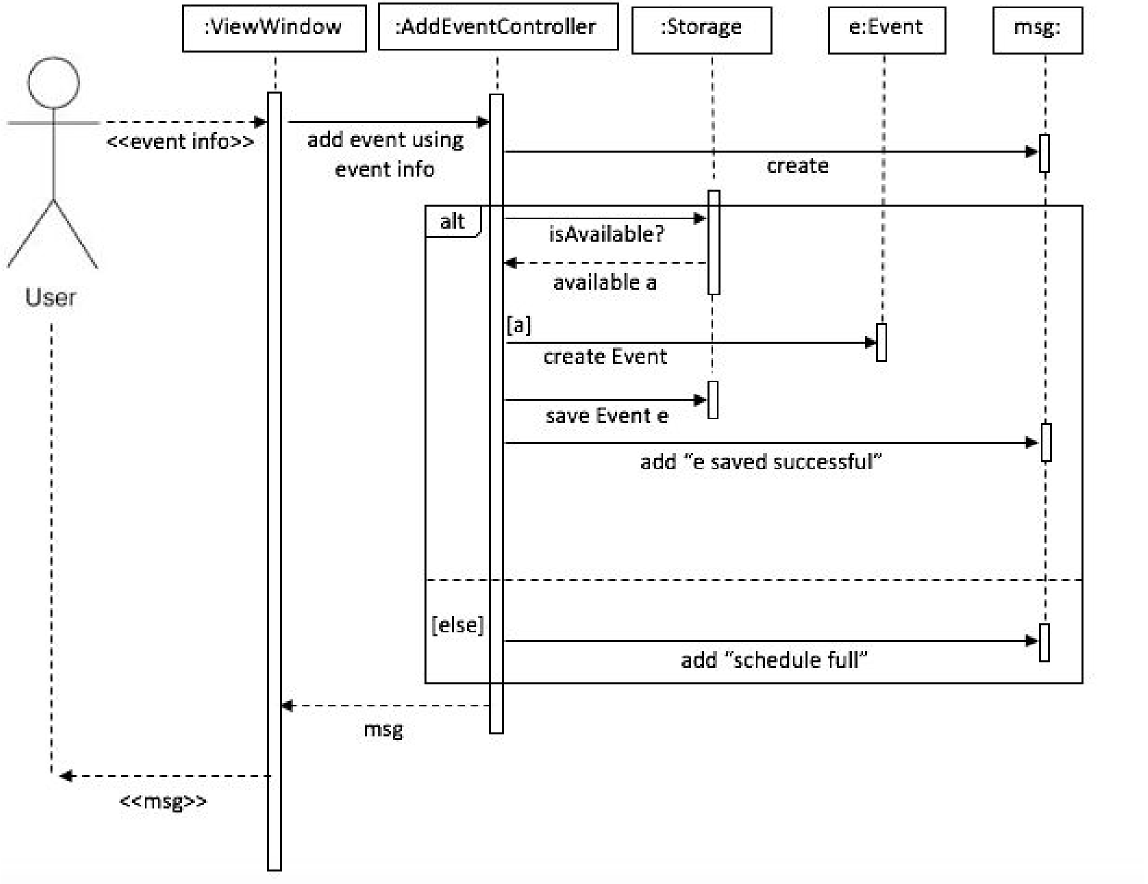
# Scenario Table (UC3: Edit Event)

| # | Subject | Subject Action | Other Data/Objects | Object Acted Upon |
|---|---|---|---|---|
| 1. | User | selects | | event |
| 2. | Edit event controller | fetches | | event name, date, start-time, end-time, location, participants |
| 3. | View window | displays | user | event name, date, start-time, end-time, location, participants |
| 4. | If storage has availability during given new time frame | | | |
| 4.1. | Edit event controller | saves | calendar storage | event |
| 4.2. | View window | displays | “event updated” | message |
| 5. | Else | | | |
| 5.1. | View window | displays | “time conflict” | message |

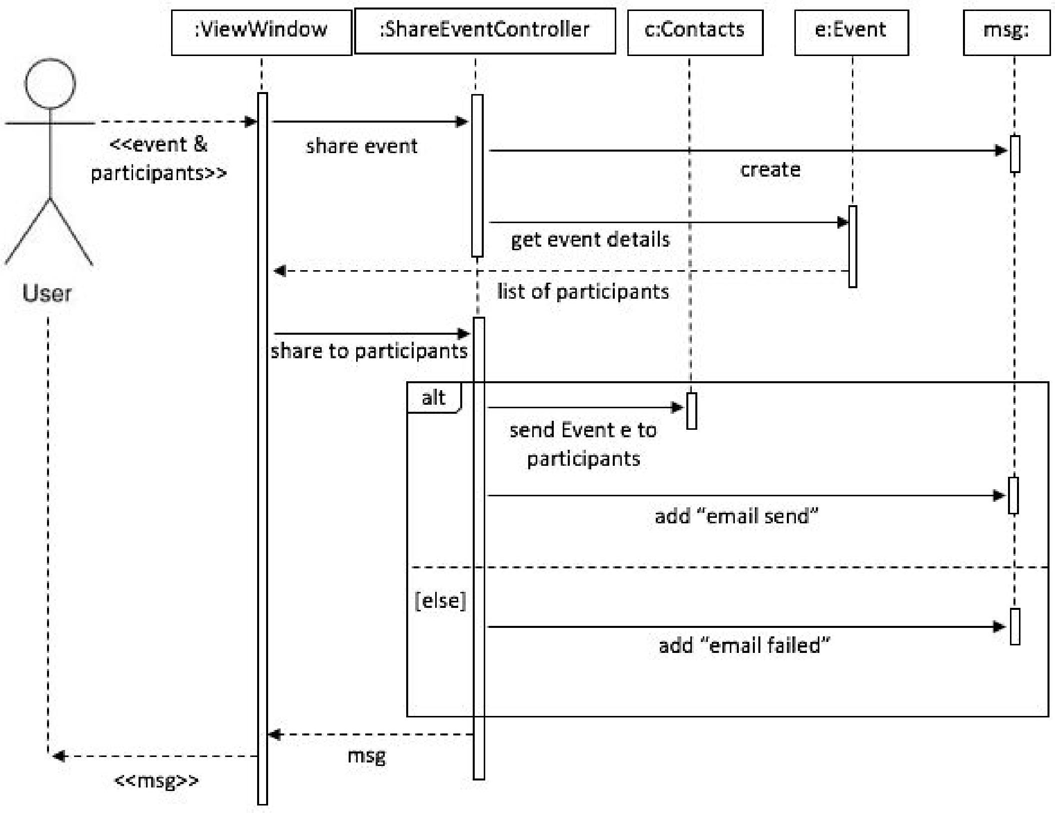
# Scenario Table (UC4: Delete Event)

| # | Subject | Subject Action | Other Data/Objects | Object Acted Upon |
|---|---|---|---|---|
| 1. | User | selects | delete button | event |
| 2. | Delete event controller | searches | event | calendar storage |
| 3. | Delete event controller | deletes | event data | calendar storage |
| 4. | View Window | displays | “event deleted” | message |

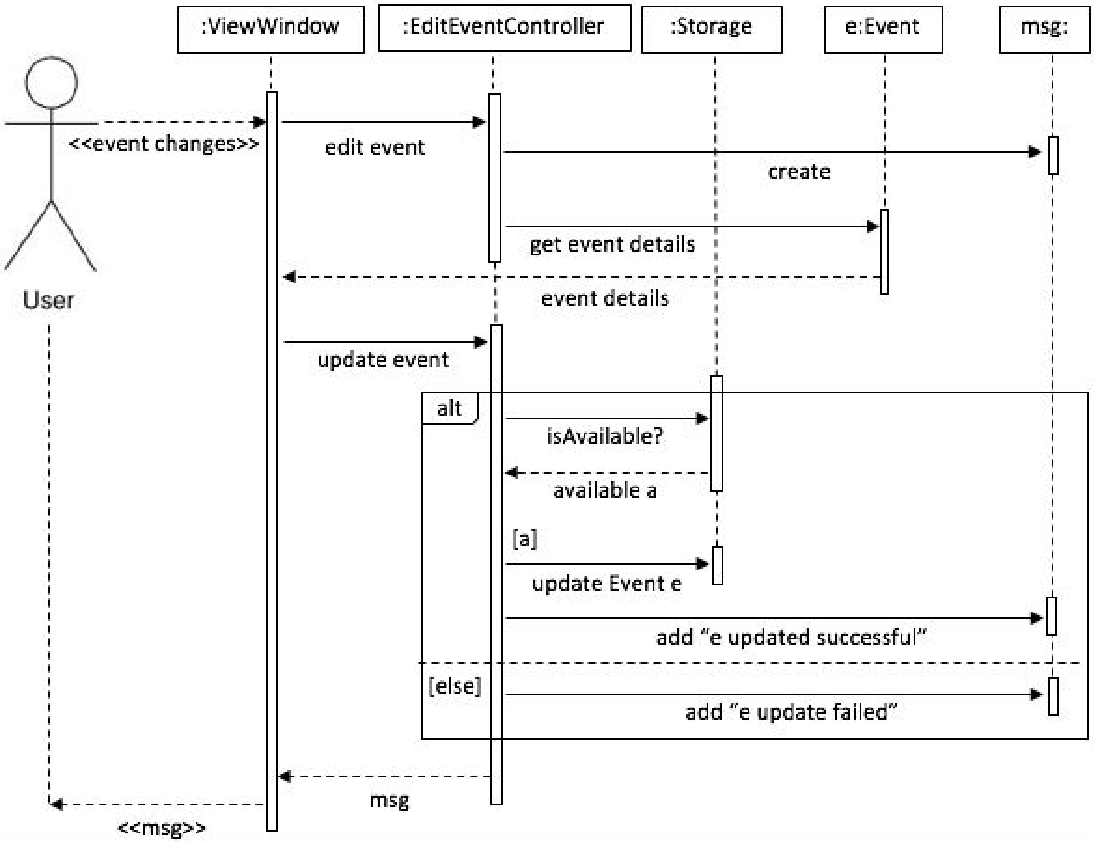
# Informal Sequence Diagram (UC1: Add Event)



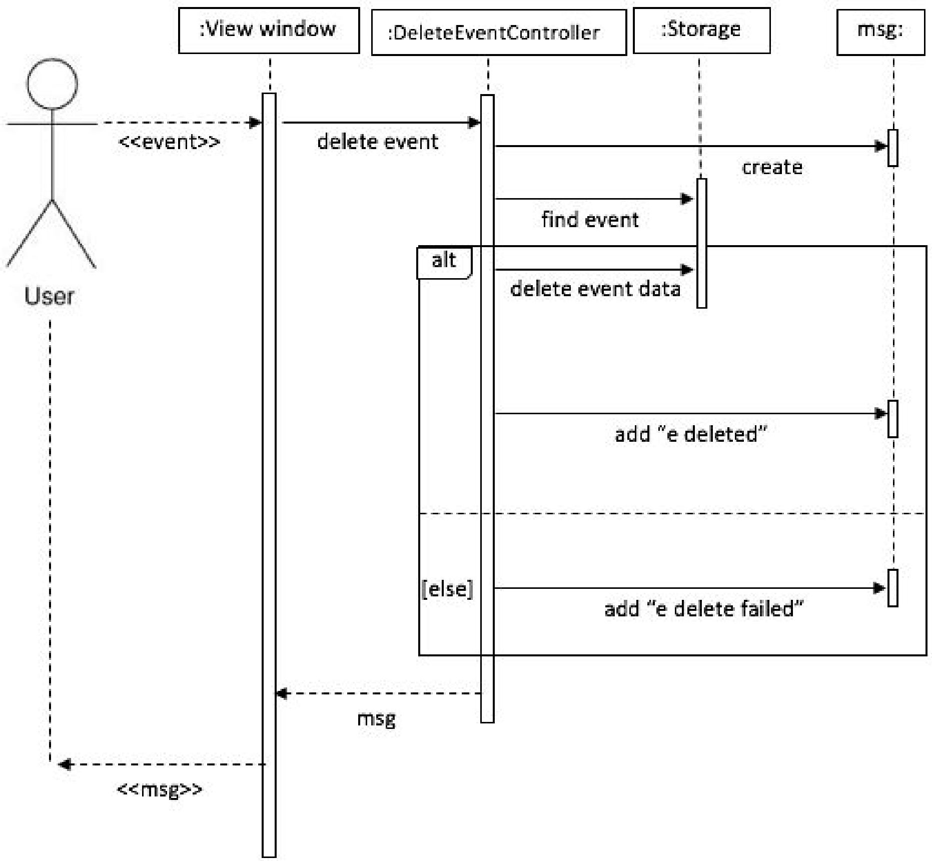
# Informal Sequence Diagram (UC2: Share Event)



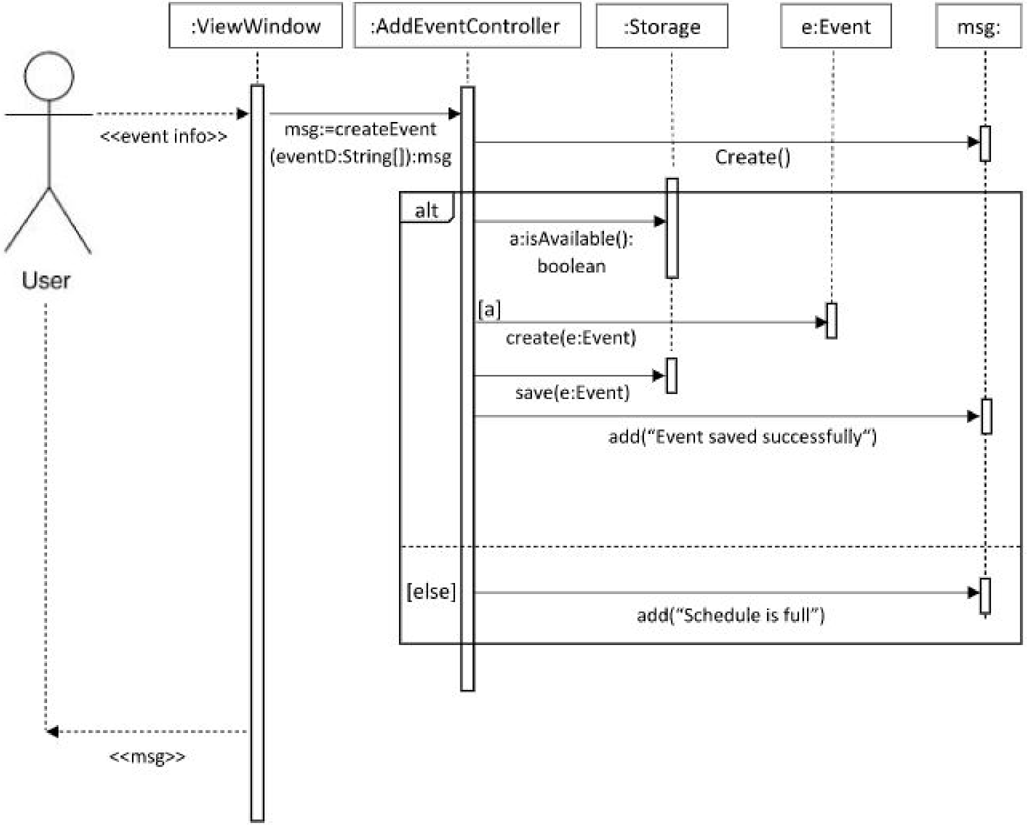
# Informal Sequence Diagram (UC3: Edit Event)



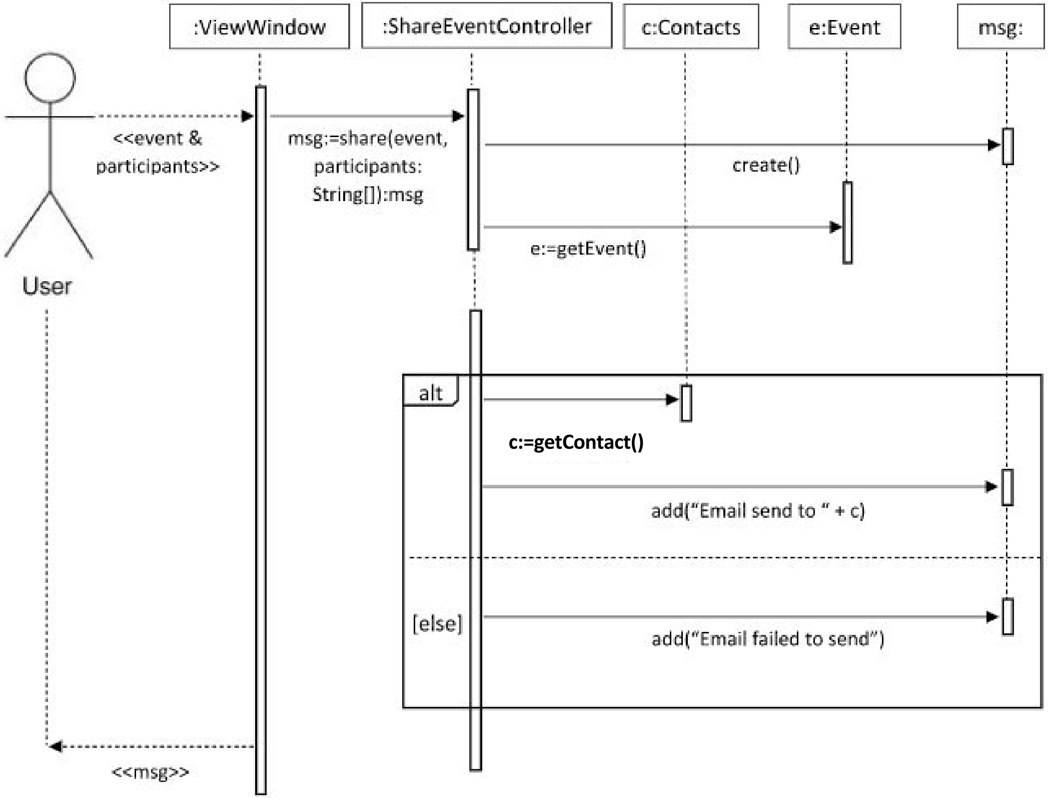
# Informal Sequence Diagram (UC4: Delete Event)



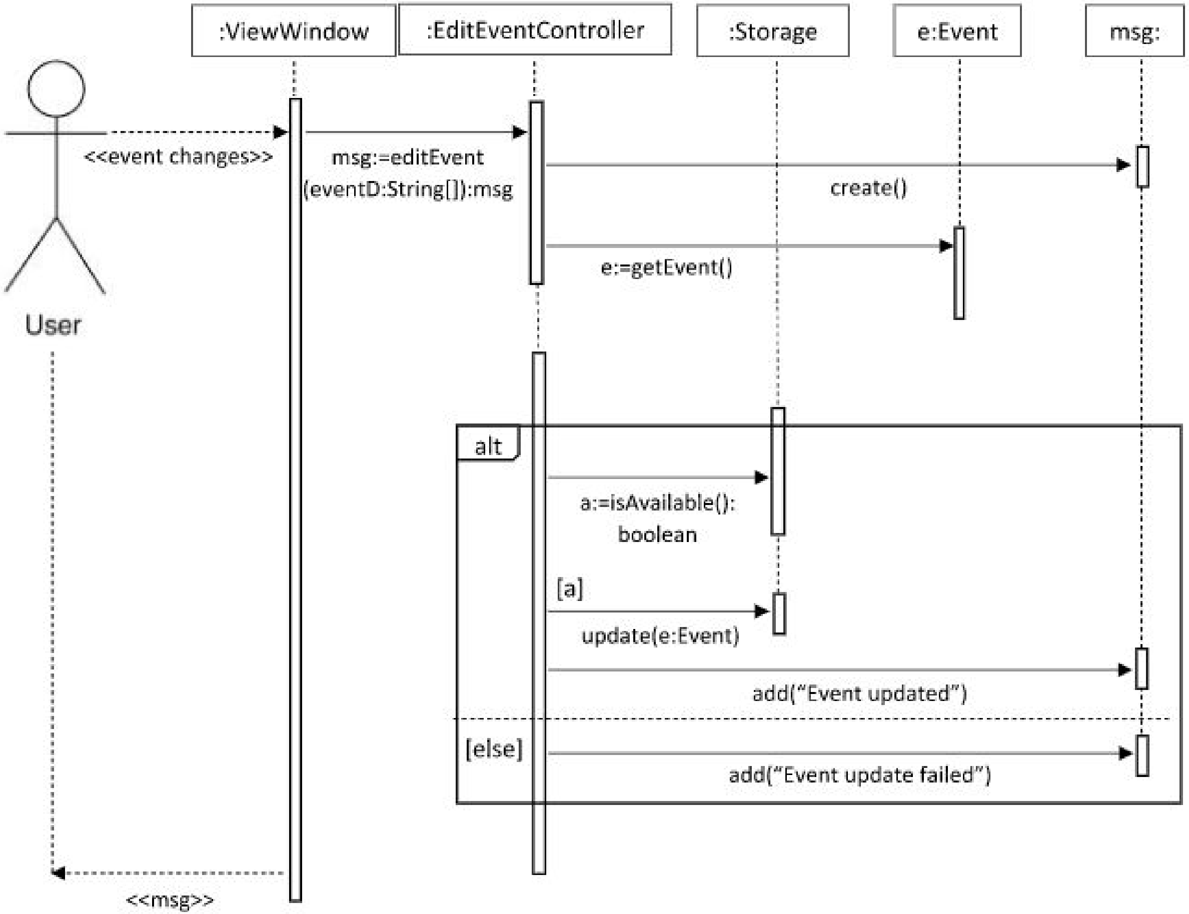
# Formal Sequence Diagram (UC1: Add Event)



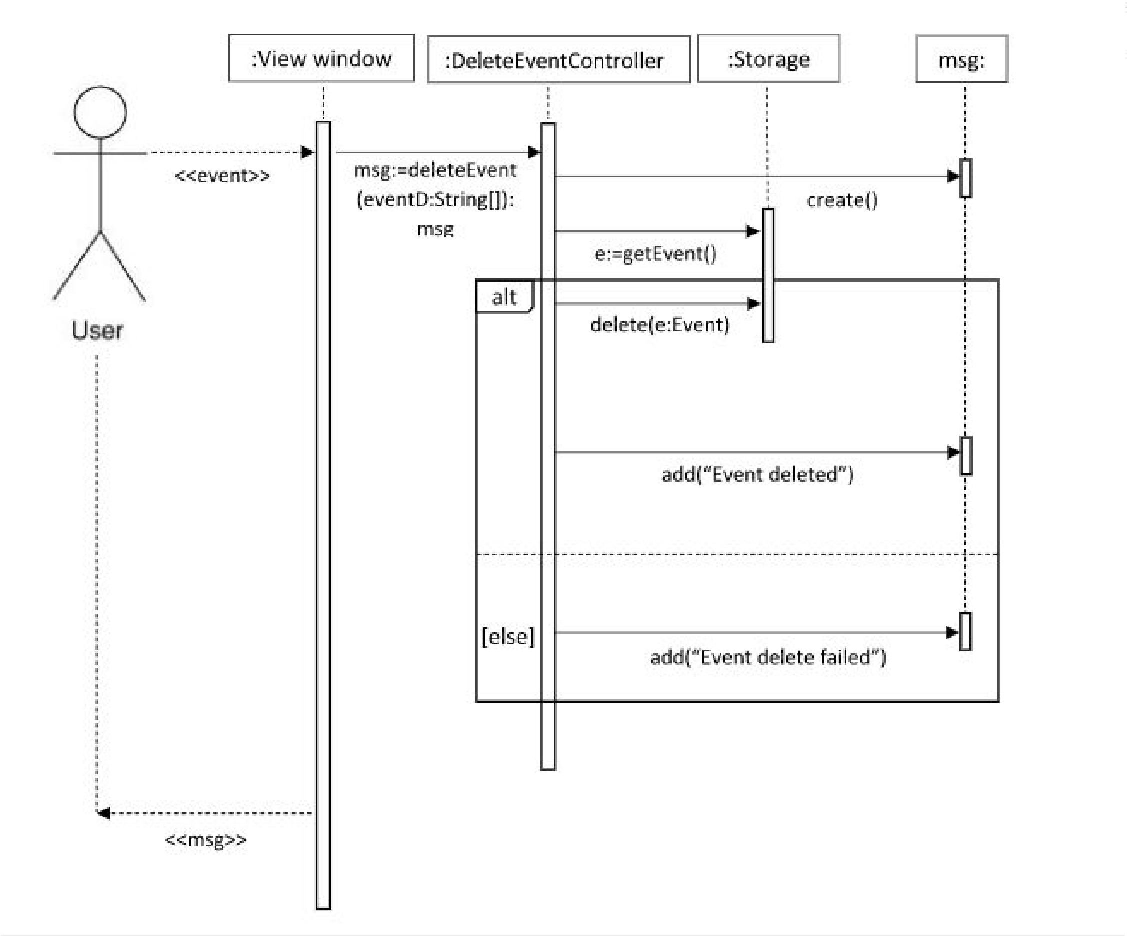
# Formal Sequence Diagram (UC2: Share Event)



# Formal Sequence Diagram (UC3: Edit Event)

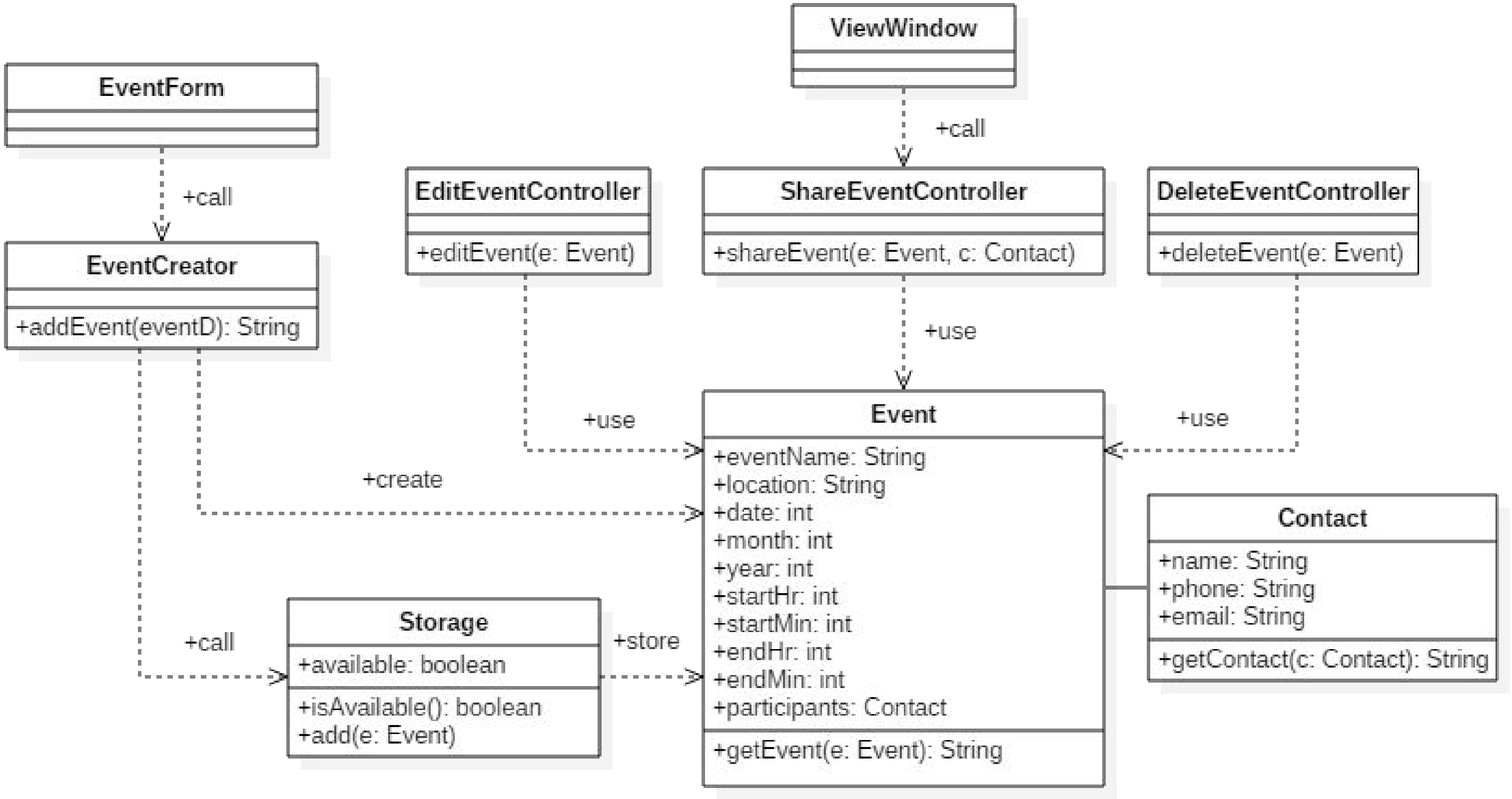


# Formal Sequence Diagram (UC4: Delete Event)



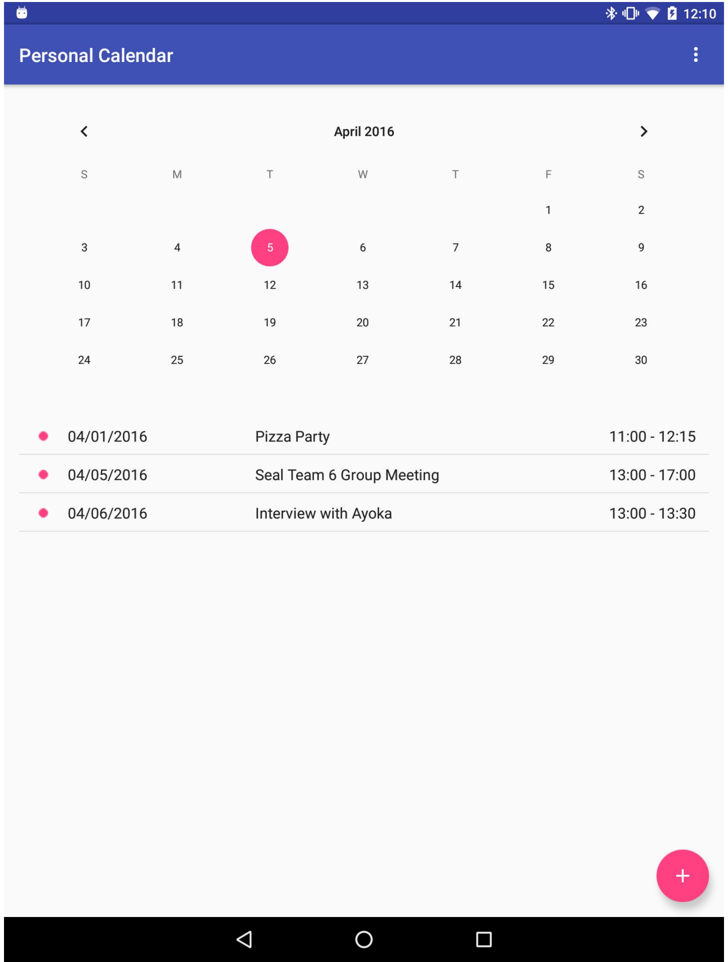
# Design Class Diagram

# Design Class Diagram



# Screen Shuts

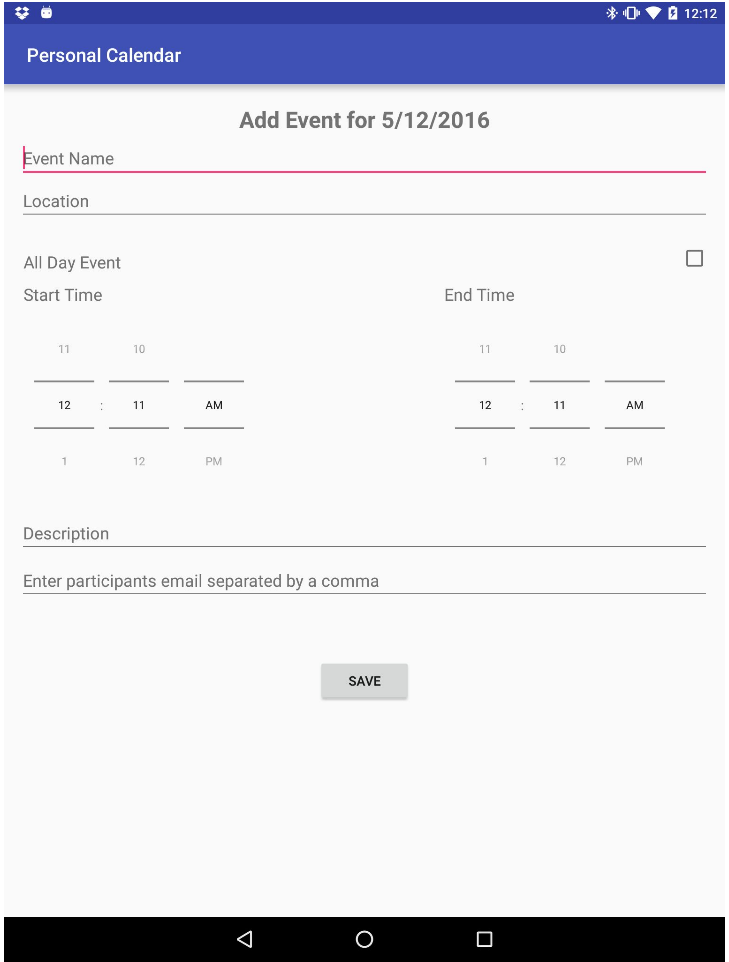
# Calendar View



| S | M | T | W | T | F | S |
|---|---|---|---|---|---|---|
| | | | | | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |

| | | | |
|---|---|---|---|
| ● | 04/01/2016 | Pizza Party | 11:00 - 12:15 |
| ● | 04/05/2016 | Seal Team 6 Group Meeting | 13:00 - 17:00 |
| ● | 04/06/2016 | Interview with Ayoka | 13:00 - 13:30 |

# Add Event



Add Event for 5/12/2016

Event Name

Location

All Day Event □

Start Time

| | | |
|---|---|---|
| 11 | 10 | |
| 12 : | 11 | AM |
| 1 | 12 | PM |

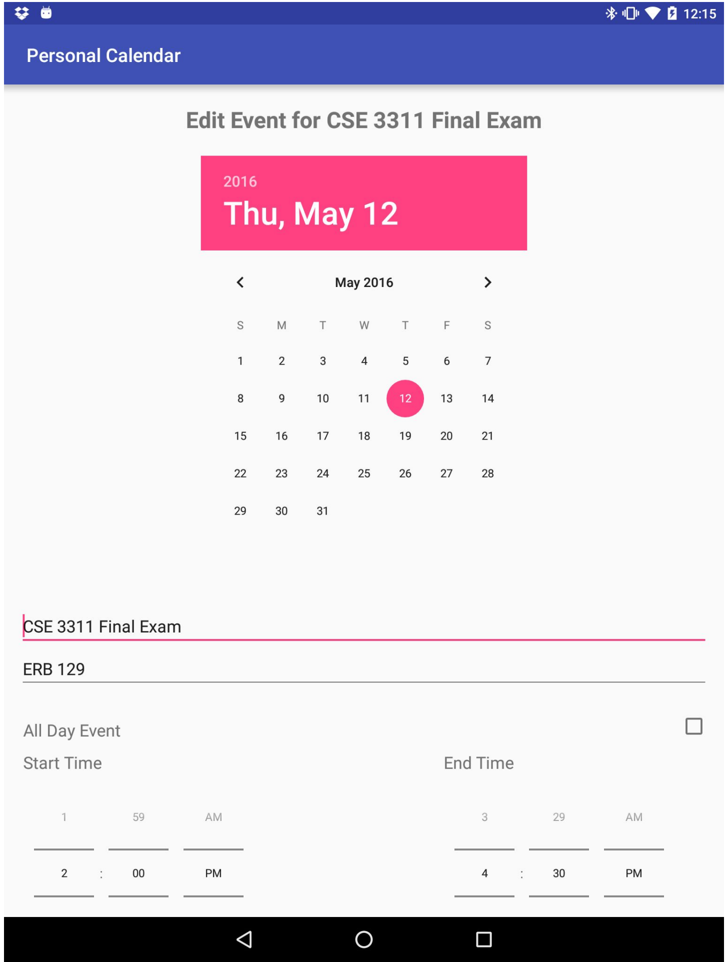
End Time

| | | |
|---|---|---|
| 11 | 10 | |
| 12 : | 11 | AM |
| 1 | 12 | PM |

Description

Enter participants email separated by a comma

# Event Details Page



Edit Event for CSE 3311 Final Exam

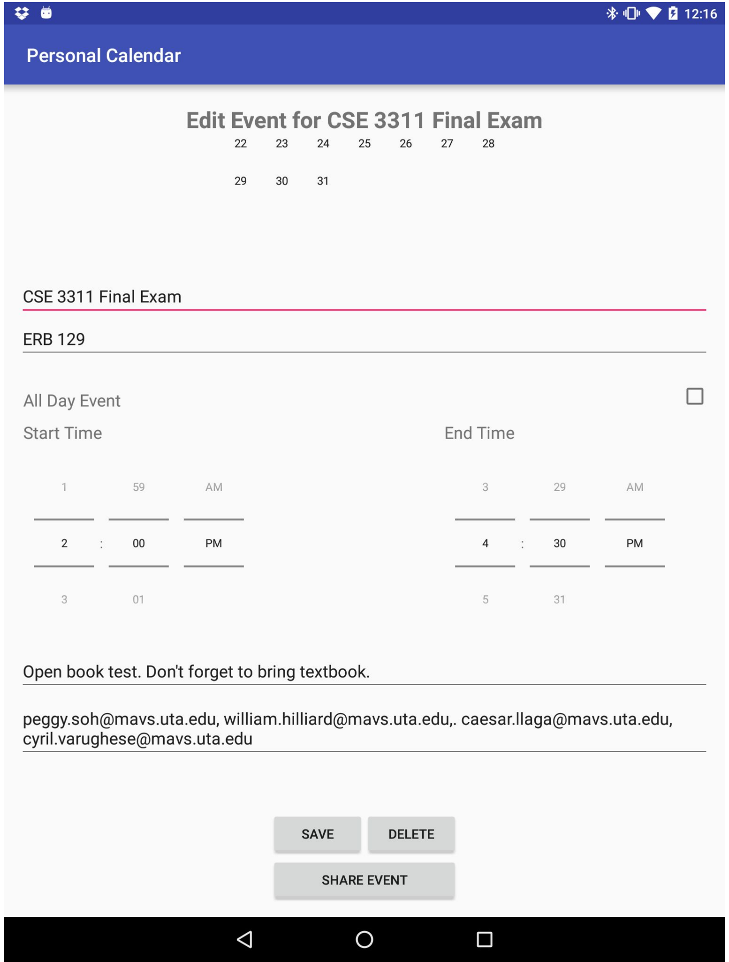
CSE 3311 Final Exam

ERB 129

All Day Event □

Start Time 2 : 00 PM

End Time 4 : 30 PM



Edit Event for CSE 3311 Final Exam

CSE 3311 Final Exam

ERB 129

All Day Event □

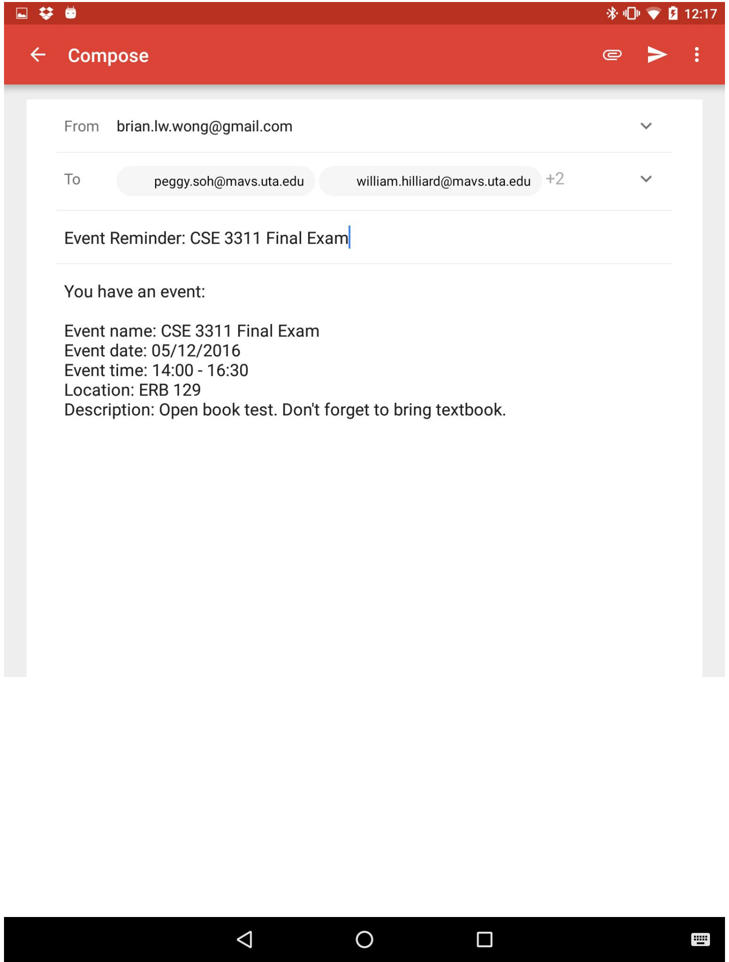
Start Time 2 : 00 PM

End Time 4 : 30 PM

Open book test. Don't forget to bring textbook.

peggy.soh@mavs.uta.edu, william.hilliard@mavs.uta.edu,. caesar.llaga@mavs.uta.edu, cyril.varughese@mavs.uta.edu

# Share Event



From brian.lw.wong@gmail.com

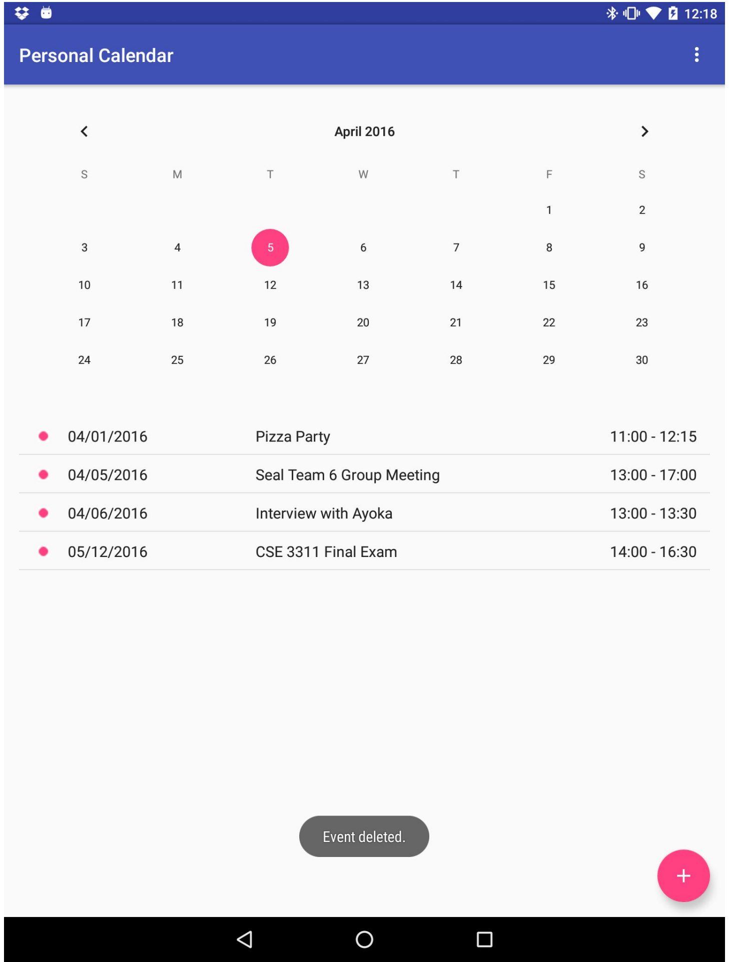
To peggy.soh@mavs.uta.edu william.hilliard@mavs.uta.edu +2

Event Reminder: CSE 3311 Final Exam

You have an event:

Event name: CSE 3311 Final Exam
Event date: 05/12/2016
Event time: 14:00 - 16:30
Location: ERB 129
Description: Open book test. Don't forget to bring textbook.

# Message



April 2016

| S | M | T | W | T | F | S |
|---|---|---|---|---|---|---|
|  |  |  |  |  | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |

| | | | |
|---|---|---|---|
| ● | 04/01/2016 | Pizza Party | 11:00 - 12:15 |
| ● | 04/05/2016 | Seal Team 6 Group Meeting | 13:00 - 17:00 |
| ● | 04/06/2016 | Interview with Ayoka | 13:00 - 13:30 |
| ● | 05/12/2016 | CSE 3311 Final Exam | 14:00 - 16:30 |

Event deleted.