

---

# **CalendarHub**

## **Use Case Diagram**

Submitted to:

Prof. Ma. Rowena C. Solamo  
Faculty Member  
Department of Computer Science  
College of Engineering  
University of the Philippines, Diliman

Submitted by:  
Jairus John Garcia  
Jan Daniel Laborada  
Diane Abegail Recuerdo

In partial fulfillment of Academic Requirements  
for the course  
CS 191 Software Engineering I  
of the  
1<sup>st</sup> Semester, AY 2016-2017

---

### **Unique Reference:**

The documents are stored in the <https://github.com/dbrecuerdo/CS191-Project>

This file may be accessed through this link: <https://github.com/dbrecuerdo/CS191-Project/raw/master/02-Requirements-Engineering/CalendarHub%20-%20Use%20Case%20Model.pdf>

### **Document Purpose:**

This document is to provide an overview of the functionalities of the project, brief discussion of these, and show the relationship among these through a visual material, the use-case diagram.

### **Target Audience:**

Any interested party in reviewing or knowing more about the project.

### **Revision Control**

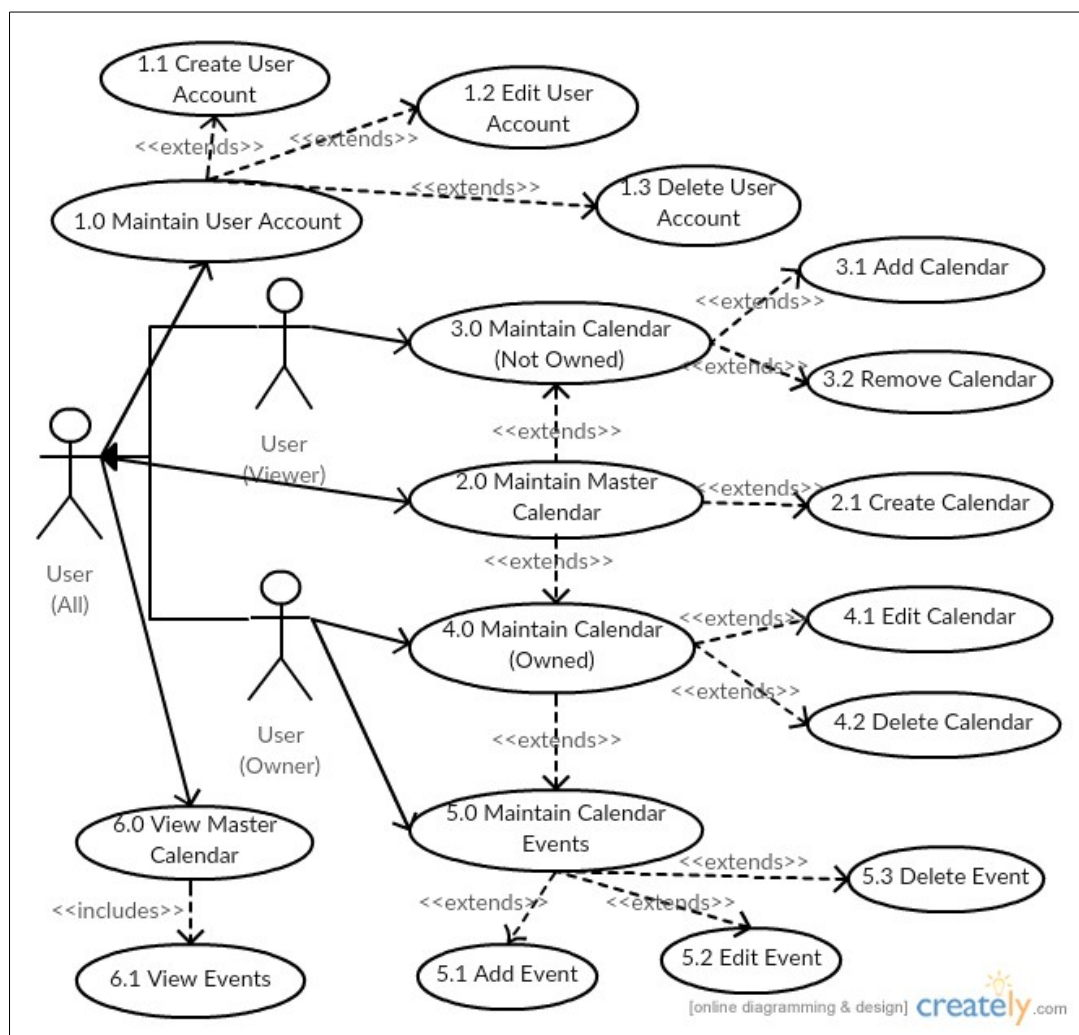
#### *History Revision:*

<b>Revision Date</b>	<b>Person Responsible</b>	<b>Version Number</b>	<b>Modification</b>
09/30/16	Diane Abegail Recuerdo	1.0	Initial Document; Added use case diagram; Added actors and use-case descriptions

**System Name:** Calendar Sharing Web Application System

**Description:** CalendarHub is an online calendar and planner geared specially towards academic use. It enables its users to create and share calendars to specific people for easy scheduling and announcements of events such as exams and deadlines. It also helps students keep track of school requirements and events to aid in efficient and effective time management.

**Use-Case Diagram:**



---

*List of Actors:*

<b>Actors</b>	<b>Description</b>
User (All)	This actor group is the generalization of users. It embodies the actions every user can perform on his/her account and the functionalities he/she can access without regards to calendar ownership. Users include teachers and students.
User (Owner)	This actor group embodies users who have created and own a calendar. They have a different level of management on the information contained in the calendar they own. Calendar owners are, ideally, teachers who share class calendars with students.
User (Viewer)	This actor group represents the users who are accessing calendars that the users do not own. They cannot edit any information on these calendars but can view them. Calendar viewers are, expectedly, students who would view a teacher-controlled class calendar.

*List of Use-cases:*

<b>Use-Case</b>	<b>Description</b>
Use-Case 1.0 Maintain User Account	This super-use case enables the user to manage his/her own user account. The grouping is based on the changes that are to be made to the user's account details. It extends to three other use-cases: create, edit, and delete user account.
Use-Case 1.1 Create User Account	This use-case creates a user account for the user. The process takes in user information as input and outputs an account for the user to use. This account is needed to use the functionalities of the application.
Use-Case 1.2 Edit User Account	This use-case is for when the user wishes to alter the information previously provided either through the user account creation process or a previous edit. It takes in new user information as input. The output of the process is the updated user information.
Use-Case 1.3 Delete User Account	This use-case lets the user delete all of his/her information on the application. When this is invoked by the user, all user data including created calendars are deleted and removed from the viewers' master calendars.
Use-Case 2.0 Maintain Master Calendar	The master calendar is the collection of all the user's both owned and view-only calendars. This super-use case maintains the calendar data of the user. It extends to create calendar, maintain calendar(not owned), and maintain calendar(owned).
Use-Case 2.1 Create Calendar	This use-case enables the user to create a calendar under the ownership of the user. The output calendar can be populated with events related to the subject of the calendar. The required input for this use-case is the name of the calendar. Upon creation, a calendar code will also be produced which can then be shared to other users to let them view the created calendar as well as its events.
Use-Case 3.0 Maintain Calendar (Not Owned)	This super-use case manages calendars that are not owned by the user. It is a subset of the master calendar. The functionalities of this case show the control of a viewer-only over a calendar.

---

---

<b><i>Use-Case</i></b>	<b><i>Description</i></b>
Use-Case 3.1 Add Calendar	This use-case enables a user to include a calendar he/she does not own to his/her master calendar and view it. This process requires the calendar code of the calendar to be added.
Use-Case 3.2 Remove Calendar	This use-case enables a user to remove a previously added view-only calendar. It removes the selected calendar from the users master calendar, disabling the user from viewing the events of the removed calendar.
Use-Case 4.0 Maintain Calendar (Owned)	This super-use case encapsulates the functionalities a user has access to over his/her calendar.
Use-Case 4.1 Edit Calendar	This allows the owner to edit the information of a previously created calendar. The user is asked to input new information for the calendar. The calendar information is the updated in accordance to the input.
Use-Case 4.2 Delete Calendar	This use-case lets the owner delete a calendar. All events in the said calendar will be lost. The viewers can no longer view the calendar and it's events as well.
Use-Case 5.0 Maintain Calendar Events	This super-use case generalizes the actions that a calendar owner can perform on the events of his/her calendar.
Use-Case 5.1 Add Event	The owner can add an event on a calendar through this function. The user is required to input necessary details of the said event. The event will then be included in the specified calendar and can be viewed by both the owner and the viewers of the calendar.
Use-Case 5.2 Edit Event	This use-case lets the calendar owner edit a previously added event. The user must input the updated details of the event. The output will be an edited event as input by the user.
Use-Case 5.3 Delete Event	This use-case allows the user (owner) to delete a previously added event. The event will be removed from the calendar and can no longer be viewed by the user and the calendar's viewers.
Use-Case 6.0 View Master Calendar	The user can view all the calendars the user has access to through this. It includes the use-case view events.
Use-Case 6.1 View Events	This use-case happens every time a view master calendar case is performed. This allows the user to view all the events in all of his/her calendars.