

Iskolendar Analysis Model

Submitted to:

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

Submitted by:

Espino, Christel Anne A.
Sanchez, Patrick Joseph
Tanquintic, Allure Migy Daidelle R.

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2018-2019



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Unique Reference:

The documents are stored in iskolendar.wordpress.com.

Purpose:

The purpose of this document is to derive the system's analysis model from the requirements model and to define the behavior, responsibilities, and attributes of the analysis classes.

Audience:

- UP Diliman College of Engineering organizations who need publicity for their events and announcements
- UP Diliman students who wish to be informed of the organizations' events

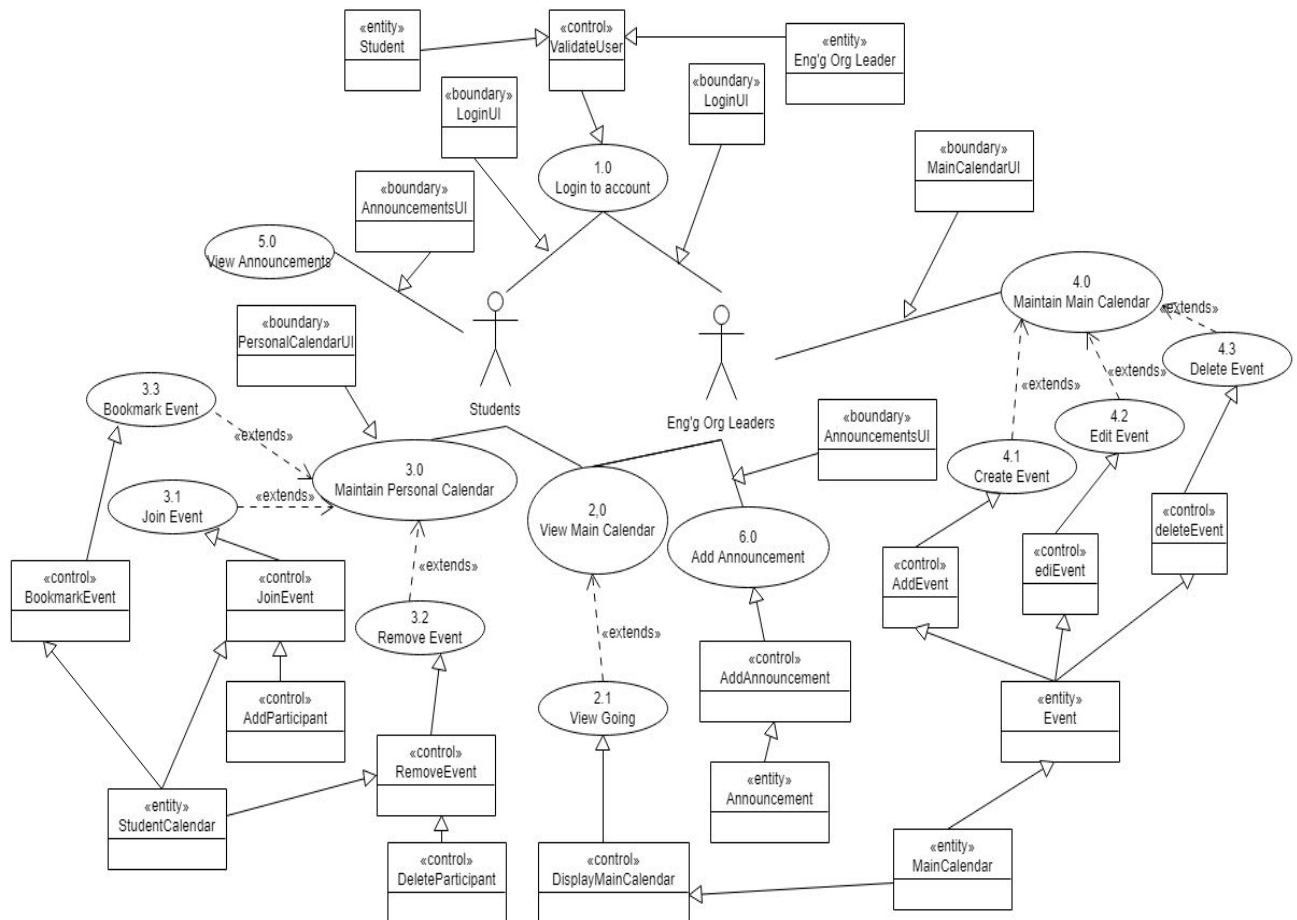
Revision Control:

<i>Revision Date</i>	<i>Person Responsible</i>	<i>Version Number</i>	<i>Modification</i>
09/21/18	Allure Tanquintic	1.0	Prepared Initial Document
10/03/18	Christel Espino	1.1	Created Behavioral Models for Use Case 3.0.
10/03/18	Patrick Joseph Sanchez	1.2	Created Behavioral Models for Use Case 4.0.
10/04/18	Patrick Joseph Sanchez	1.3	Modified the Behavioral Models for Use Case 4.0.
10/05/18	Allure Tanquintic	1.4	Created Behavioral Models for Use Cases 1.0 to 2.1
10/05/18	Christel Espino	1.5	Added Boundary Classes, Control Classes, and Entity Classes.
10/05/18	Patrick Joseph Sanchez	1.6	Added Boundary Classes, Control Classes, and Entity Classes.
10/05/18	Christel Espino	1.7	Partly did the Analysis Model
10/05/18	Patrick Joseph Sanchez	2.0	Finalized the Analysis Model

System Name: Iskolendar

Description: Iskolendar is a simple, centralized, calendar-like application where student organizations from the UP College of Engineering can post and promote their events such as seminars, workshops, review sessions, and the like so that students can view everything in one place. Student users may also view a personal calendar where only the events they're interested in joining will be displayed so that they will be reminded of important dates. On the other hand, the system will be beneficial to the organizations when it comes to event publicity and information dissemination.

Analysis Model:



Boundary Classes:

Class Name	Description
LoginUI	<p>This is the interface of the user to the system whenever he or she needs to login and access the system.</p> <p><u>Responsibilities:</u></p> <pre>public void login(String studentNumber, String username, String password) public void showErrorMessage(Boolean loginSuccessful) public void displayCalendar(Boolean loginSuccessful)</pre>
PersonalCalendarUI	<p>This is the interface of the student user to the system whenever he or she needs to maintain his or her personal calendar.</p> <p><u>Responsibilities:</u></p> <pre>public Event getEvents(String studentNumber) public void removeEventButton(Boolean removeEventIsClicked)</pre>
MainCalendarUI	<p>This is the interface of the eng'g org leader to the system whenever he or she needs to maintain the main calendar.</p> <p><u>Responsibilities:</u></p> <pre>public void enterEventData(String eventname, String date, String time, String eventDetails, String status)* public Event findEvent() public boolean viewGoingButton(Boolean viewGoingIsClicked) public Event viewEvents() *for Eng'g Org Leaders only</pre>
AnnouncementsUI	<p>This is the interface of the users to the system whenever he or she needs to view the announcements.</p> <p><u>Responsibilities:</u></p> <pre>public Announcement viewAnnouncements()</pre>

Control Classes:

Class Name	Description
ValidateUserController	This is the control that verifies the login details entered by the user. <u>Responsibilities:</u> public void validateUser(String username, String password)
DisplayMainCalendarController	This is the control that displays the Main Calendar. <u>Responsibilities:</u> public MainCalendar viewMainCalendar() public Event viewDateEvents() public Event viewGoing() public Event getGoingData()
AddEventController	This is the control that adds an Event to the system. <u>Responsibilities:</u> addEvent(Event a, MainCalendar a)
EditEventController	This is the control that updates the Event chosen from the system. <u>Responsibilities:</u> editEvent(Event a, MainCalendar a)
DeleteEventController	This is the control that deletes an Event from the system. <u>Responsibilities:</u> removeEvent(Event a)
AddAnnouncementController	This is the control that posts announcements about the changes in the events. <u>Responsibilities:</u> AddAnnouncement(Event a)
JoinEventController	This is the control that adds an Event to the student's personal calendar. <u>Responsibilities:</u> public void AddEvent(Event a, StudentCalendar a, String studentNumber)
AddParticipantController	This is the control that adds a Student to the event's list of participants. <u>Responsibilities:</u> public void AddParticipant(Event a, Student a)
BookmarkEventController	This is the control that adds an Event to the student's bookmarks. <u>Responsibilities:</u> public void AddBookmark(Event a, StudentCalendar a, String studentNumber)
RemoveEventController	This is the control that removes an Event from the student's personal calendar. <u>Responsibilities:</u> public void RemoveEvent (Event a, StudentCalendar a)

DeleteParticipantController	<p>This is the control that removes a Student from the event's list of participants.</p> <p><u>Responsibilities:</u></p> <p>public void DeleteParticipant (Event a, Student a)</p>
-----------------------------	--

Entity Classes:

Class Name	Description
Event	<p>This is the entity class event, which contains the data about the event.</p> <p><u>Attributes:</u></p> <p>public String Date</p> <p>public String Title</p> <p>public String Description</p> <p>public String location</p> <p>public String Org</p>
Student (UserAccount)	<p>This is the entity student, which contains the data about registered students.</p> <p><u>Attributes:</u></p> <p>public String studentNumber</p> <p>public String username</p> <p>private String password</p> <p>public String name</p> <p>public String degProgram</p> <p>public String college</p> <p>public String emailAd</p>
EnggOrgLeader (UserAccount)	<p>This is the entity eng'g org leader, which contains the data about the registered eng'g org leaders.</p> <p><u>Attributes:</u></p> <p>public String studentNumber</p> <p>public String username</p> <p>private String password</p> <p>public String name</p> <p>public String position</p> <p>public String organization</p> <p>public String degProgram</p> <p>public String college</p> <p>public String emailAd</p>
MainCalendar	<p>This is the entity MainCalendar, which contains the data about the Main Calendar.</p> <p><u>Attributes:</u></p> <p>public Event[] events</p>
StudentCalendar	<p>This is the entity student calendar, which contains the data about a student's personal</p>

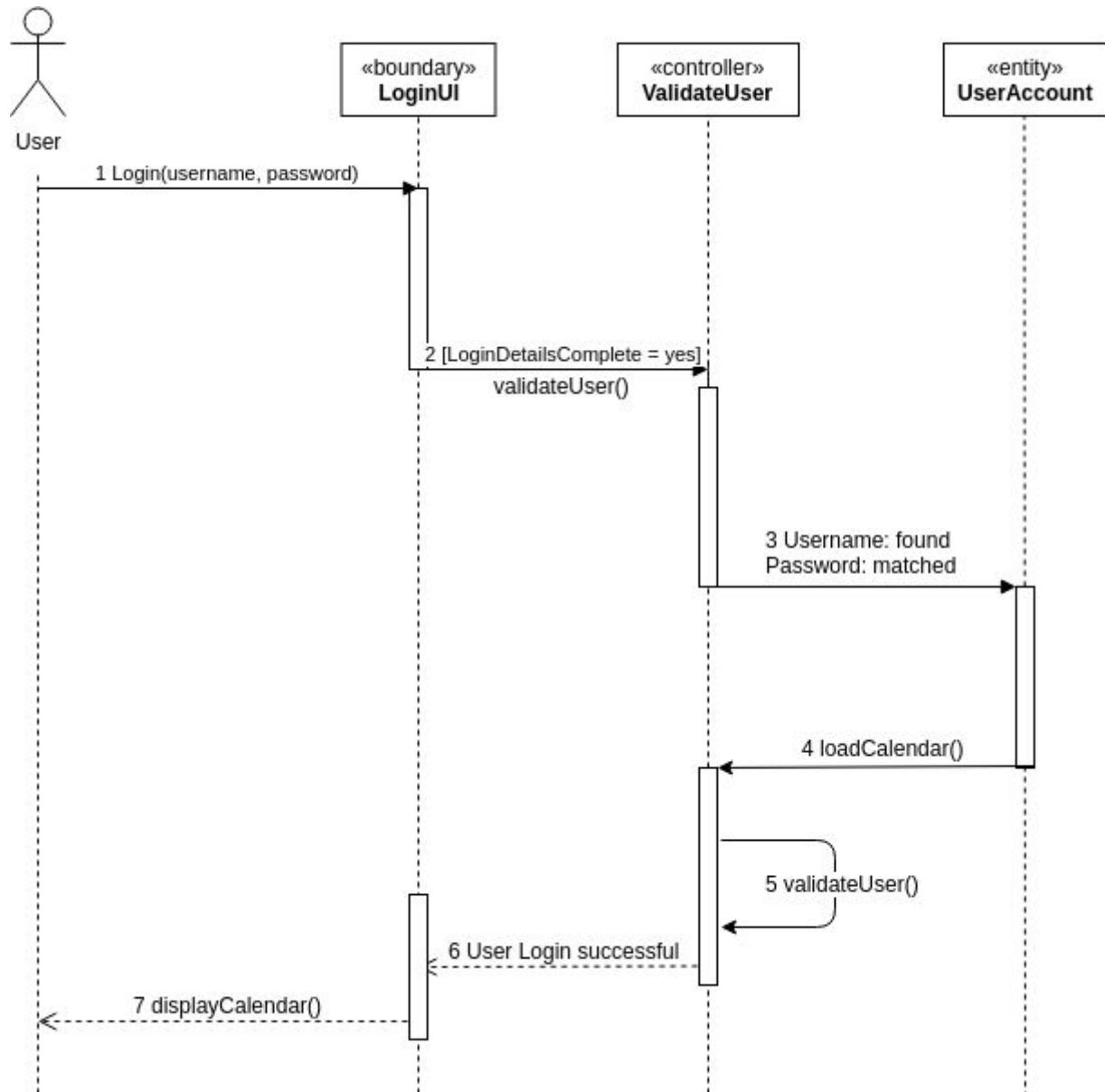
	<p>calendar.</p> <p><u>Attributes:</u></p> <p>public String studentNumber (owner of calendar)</p> <p>public Event[] eventsJoined</p> <p>public Event[] eventsBookmarked</p>
Announcement	<p>This is the entity class announcement, which contains the data about the changes in the events.</p> <p><u>Attributes:</u></p> <p>public String announcement</p> <p>public String event</p> <p>public String org</p>

Behavioral Model:

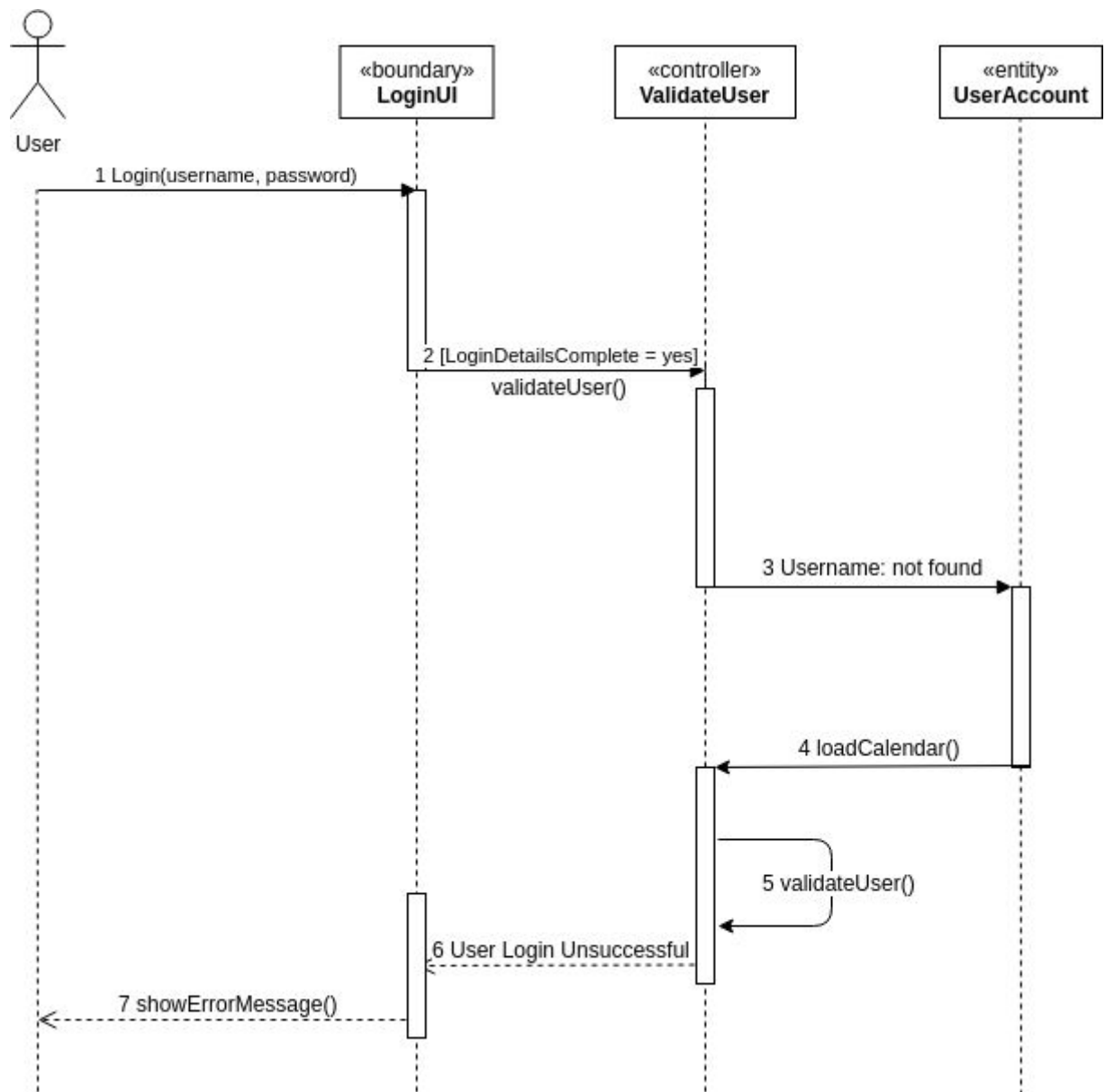
Use-Case Name: 1.0 Login

Description: The system should be able to allow the user to have log-in session.

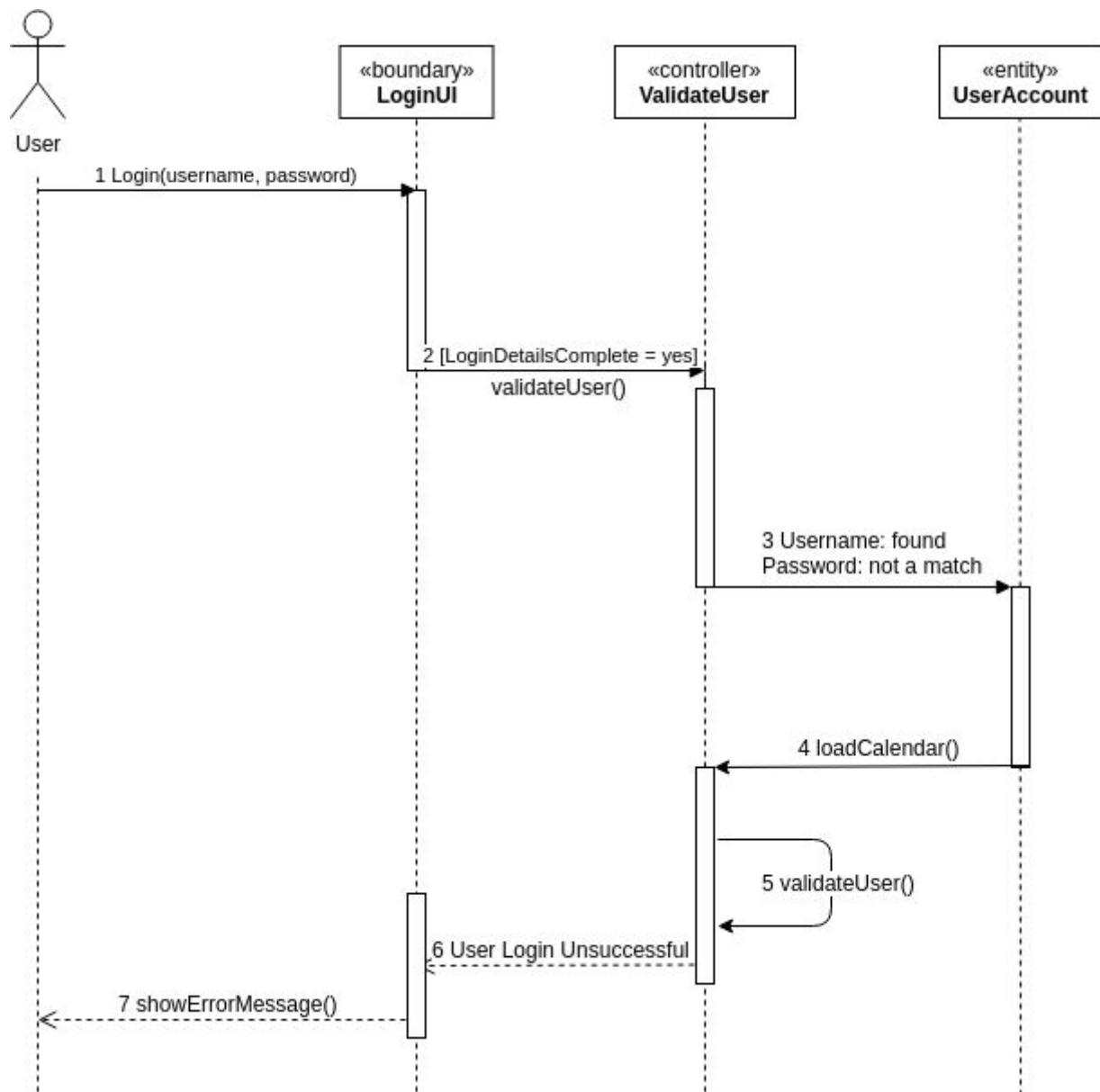
Scenario 1: Successful Log-in (Basic Flow)



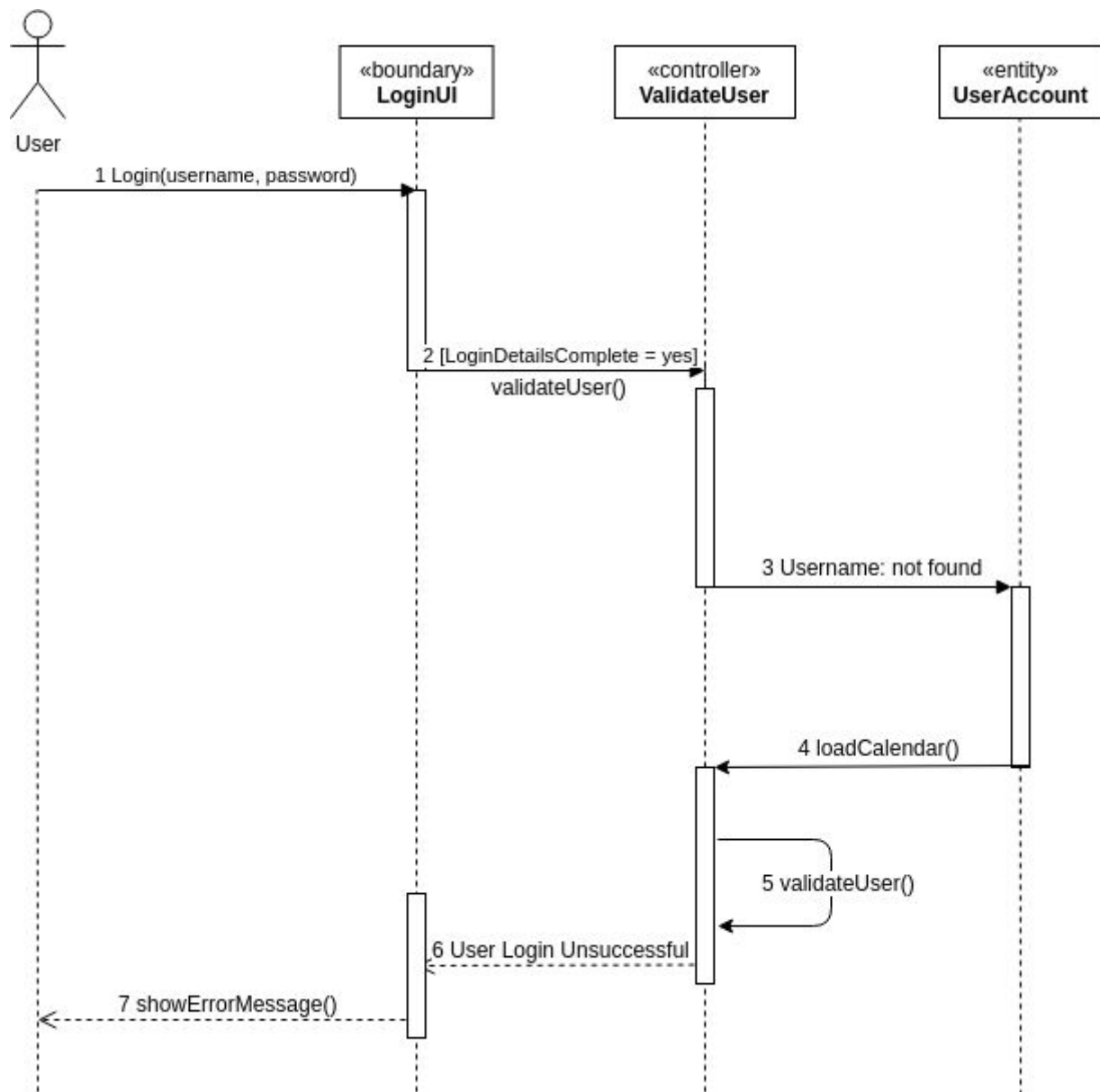
Scenario 2: Invalid Username



Scenario 3: Invalid Password



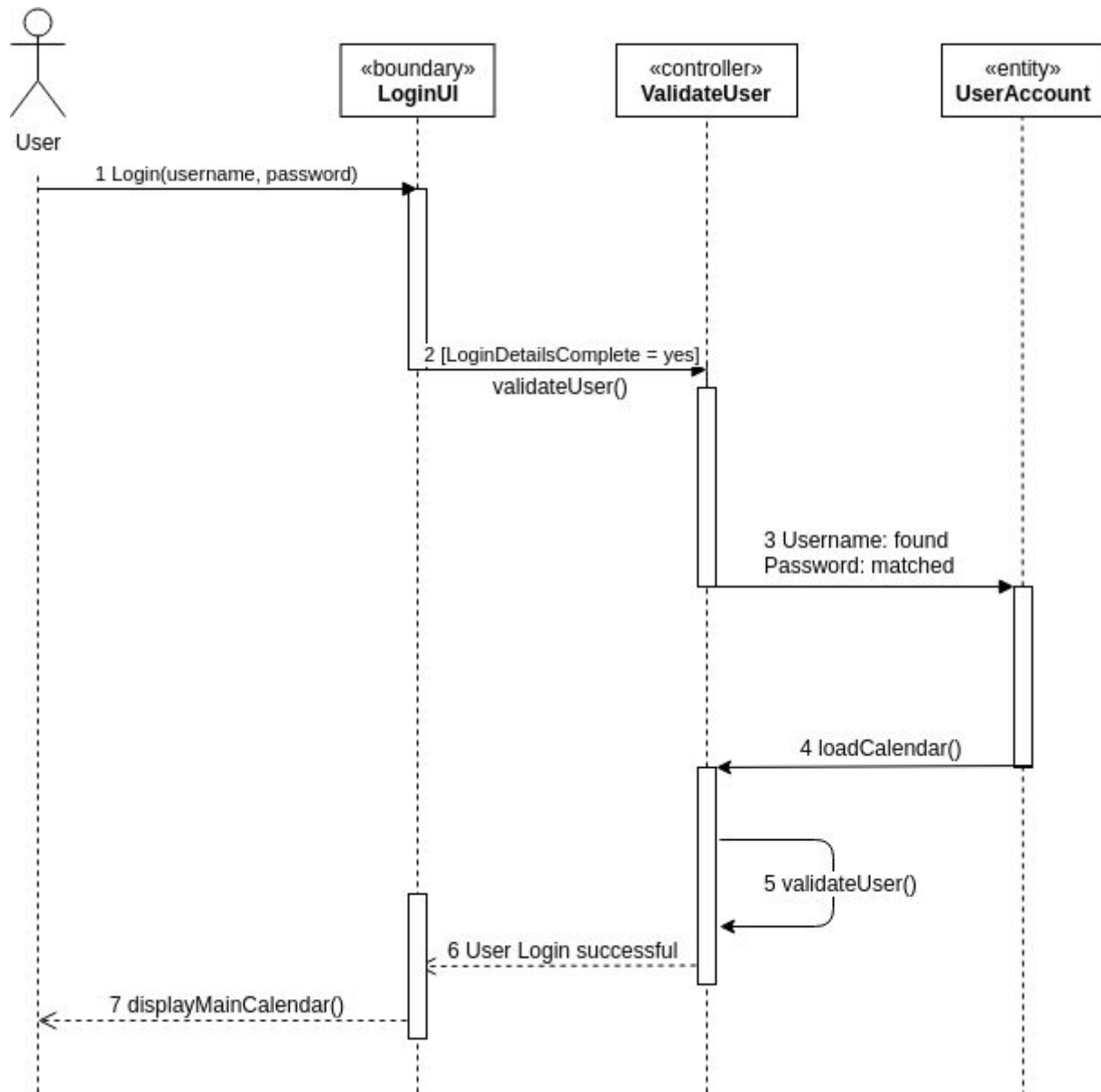
Scenario 4: Non-registered user



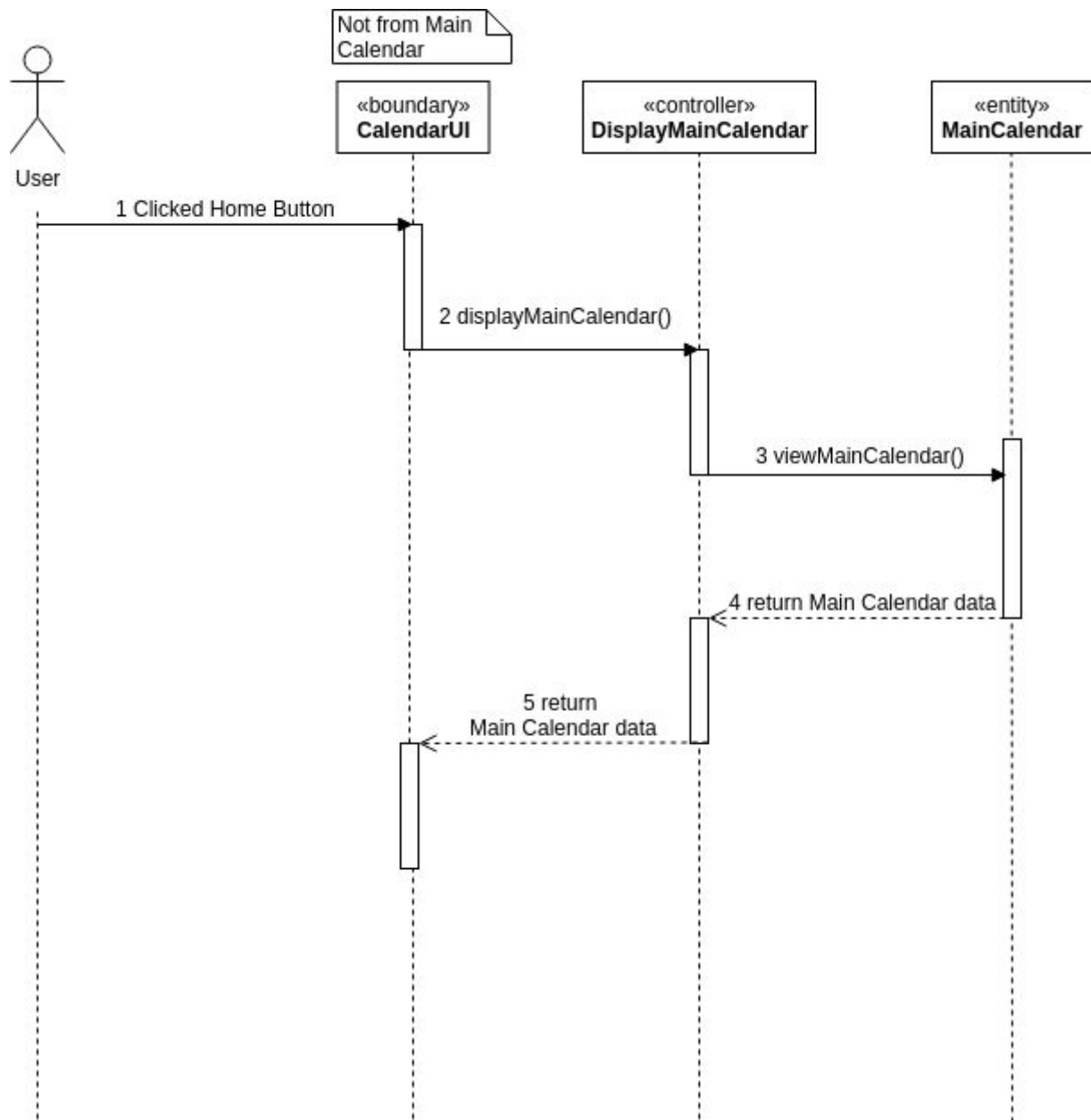
Use-Case Name: 2.0 View Main Calendar

Description: This use case describes how a user can view the main calendar of the system. This includes viewing the event details.

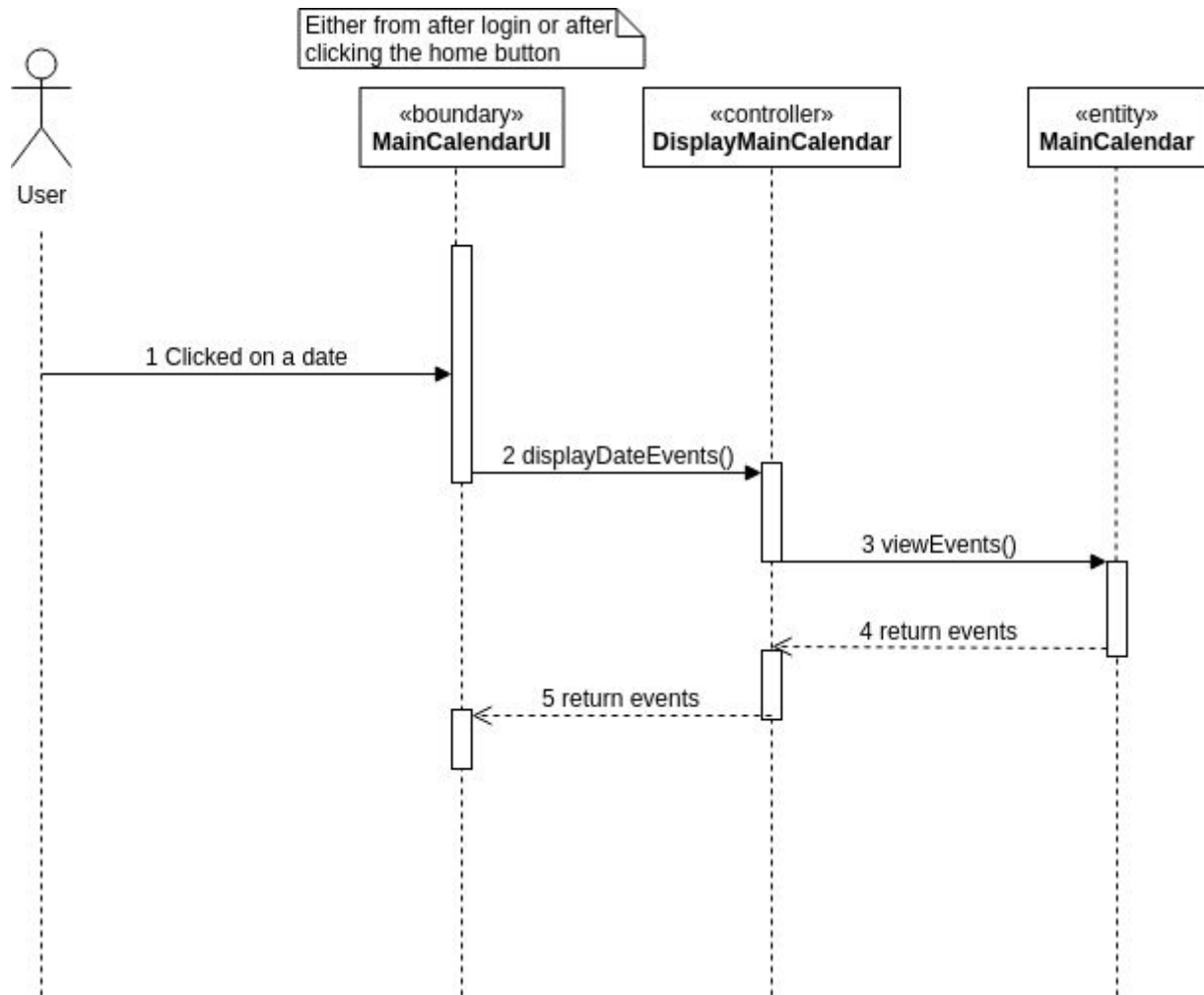
Scenario 1: Successful viewing of the main calendar after login (Basic Flow)



Scenario 2: Successful viewing of the main calendar using home button



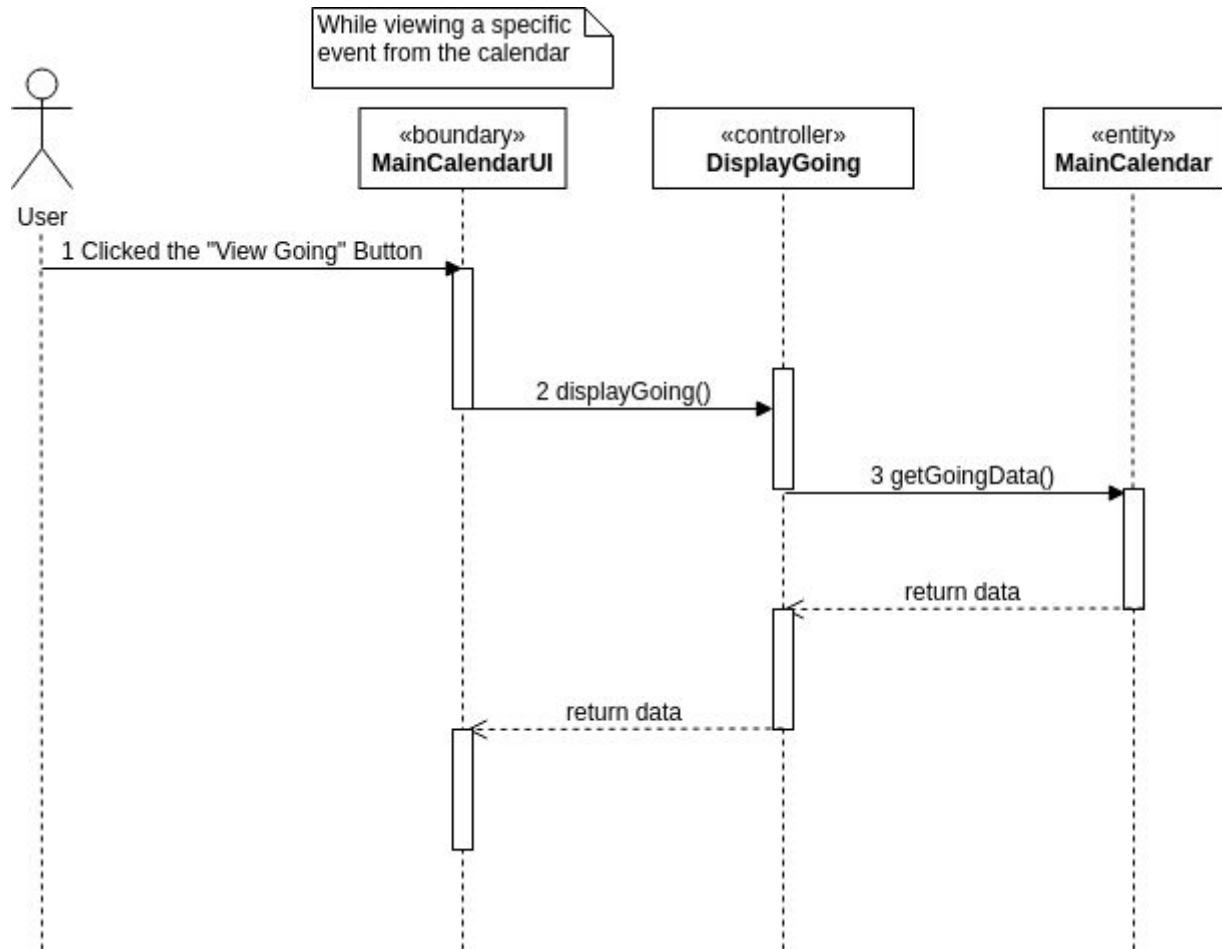
Scenario 3: Successful viewing of events (specific date)



Use-Case Name: 2.1 View "going"

Description: Enables the user to view the people who are planning to go to a specific event.

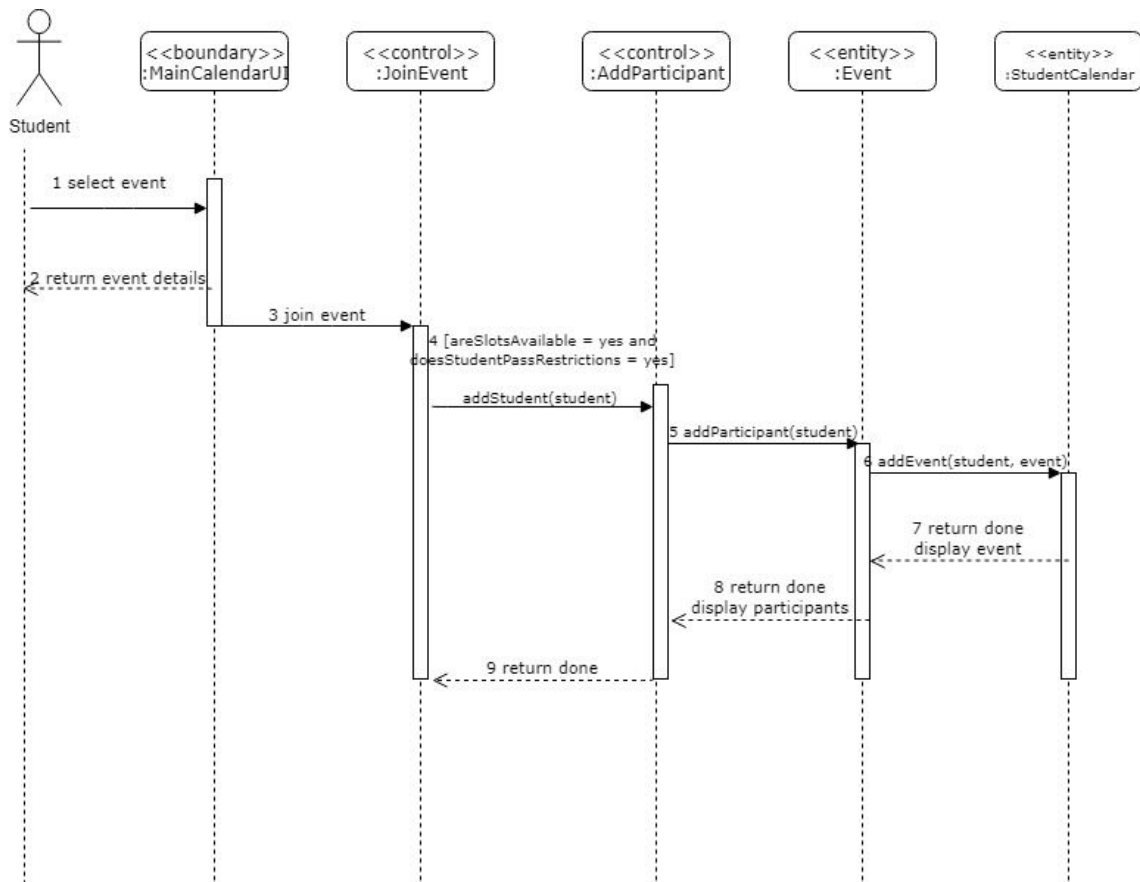
Scenario 1: Successful viewing of participants (Basic Flow)



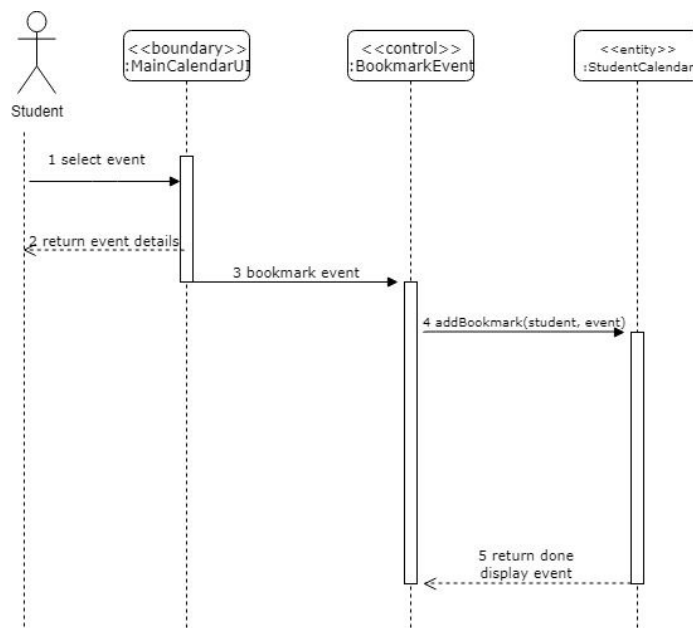
Use-Case Name: 3.0 Maintain Personal Calendar

Description: Each student user has his own personal calendar which is not viewable by and not available to the public. This calendar shows the events where the student marked himself as “going”. The calendar also displays events that the student bookmarked. This is to keep track of important dates.

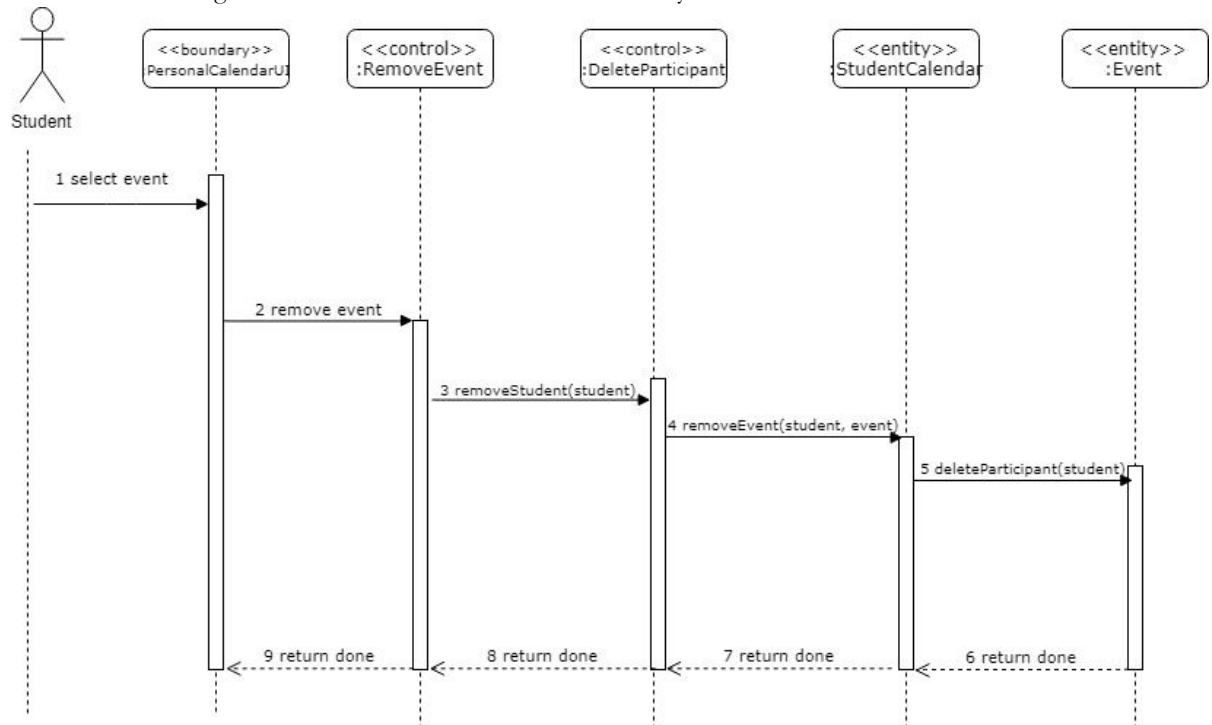
Scenario 1: Basic Flow. Student confirms participation / attendance to an event.



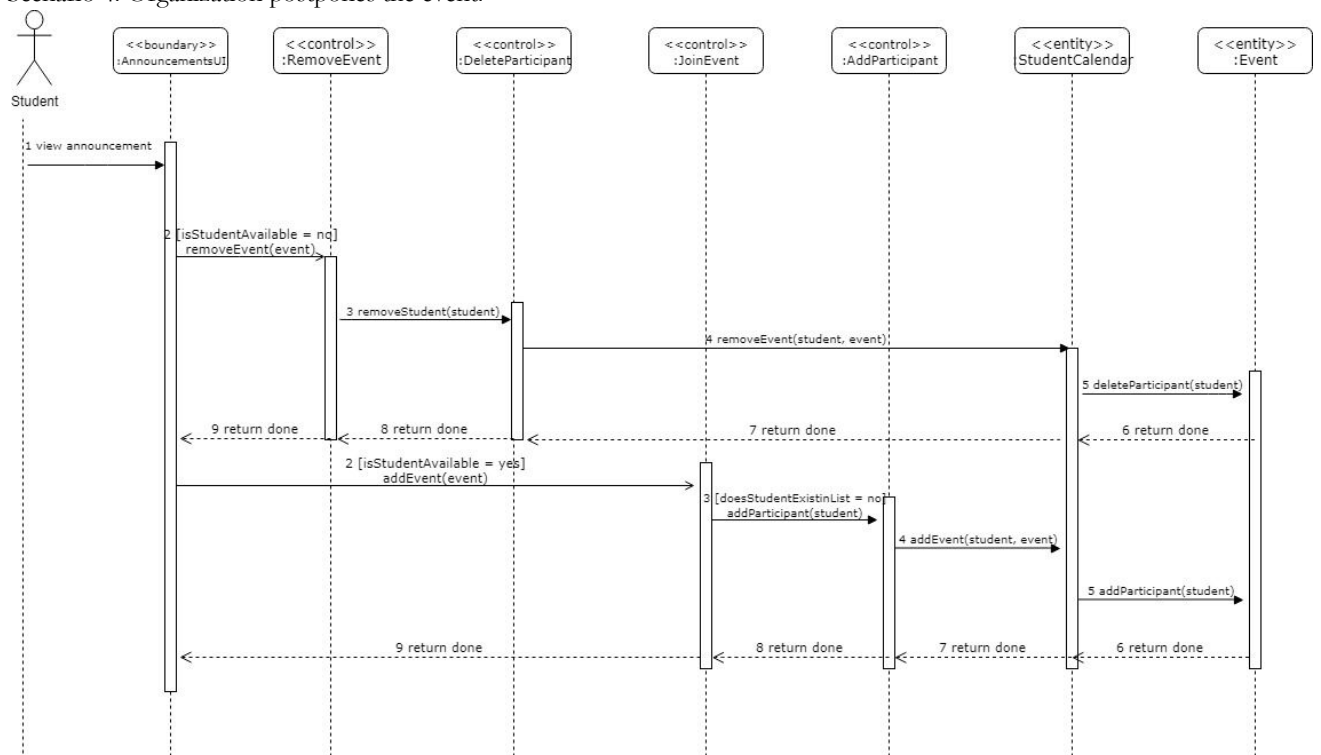
Scenario 2: Student is interested in an event but is not confirmed to go yet.



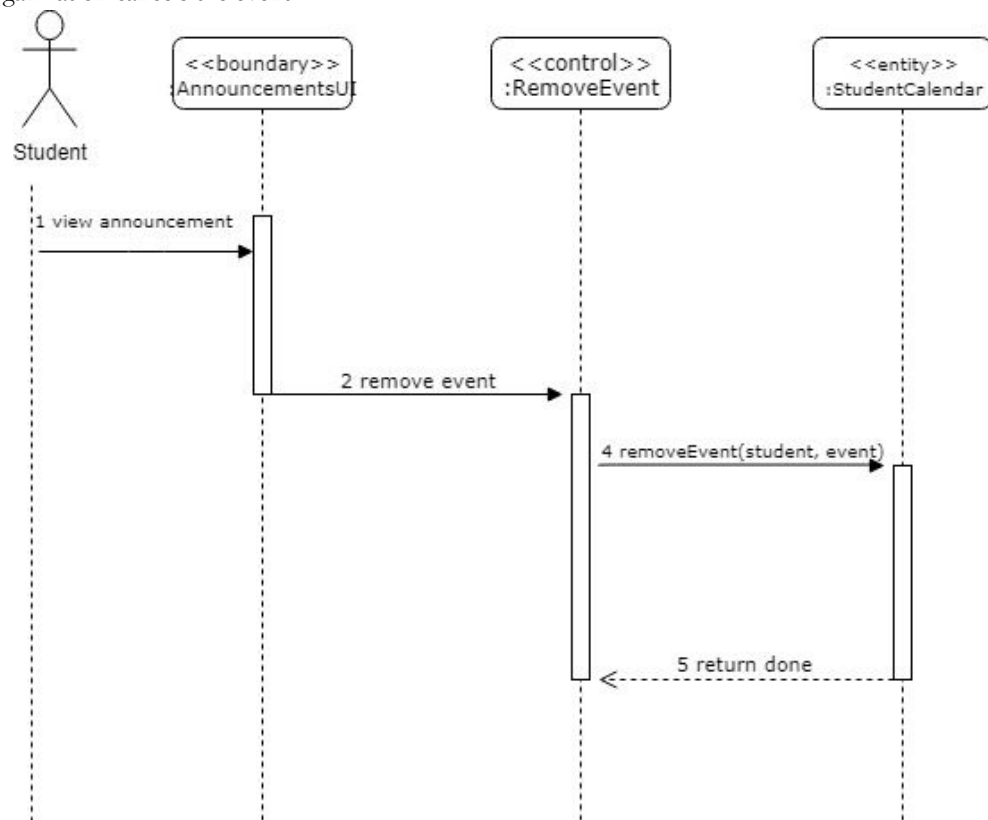
Scenario 3: Student changes his mind and will not attend the event anymore.



Scenario 4: Organization postpones the event.



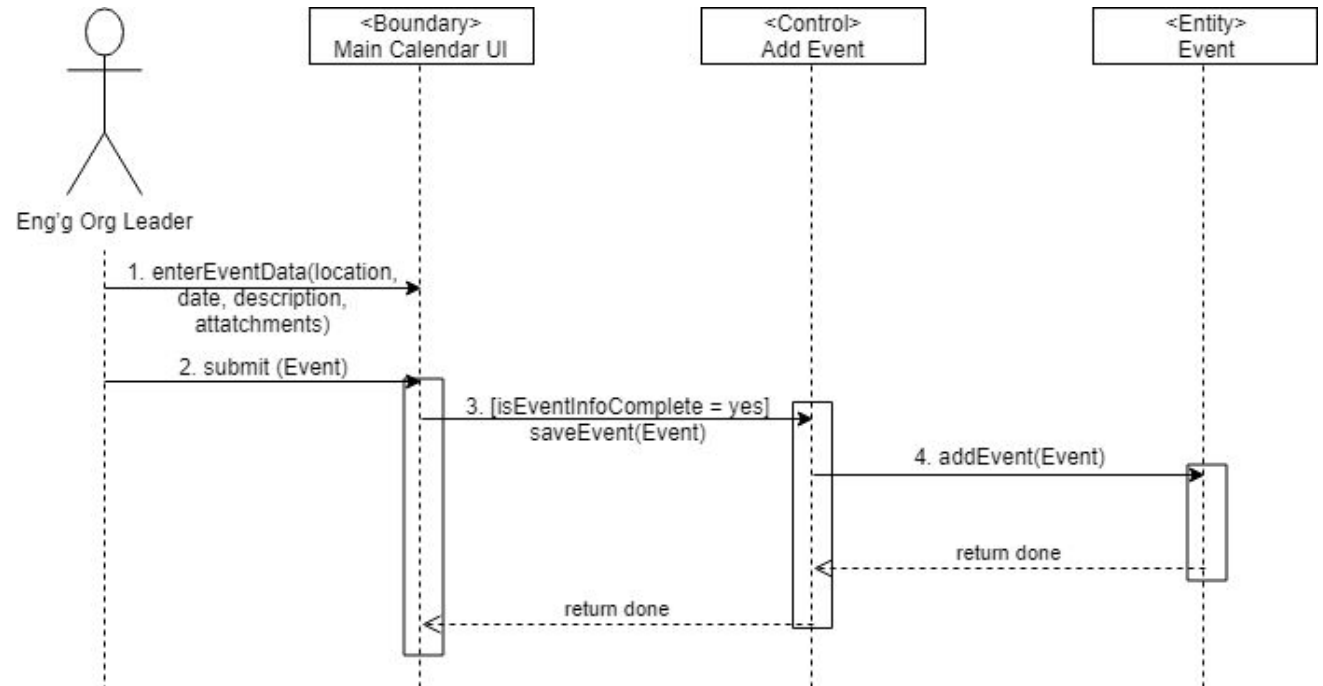
Scenario 5: Organization cancels the event.



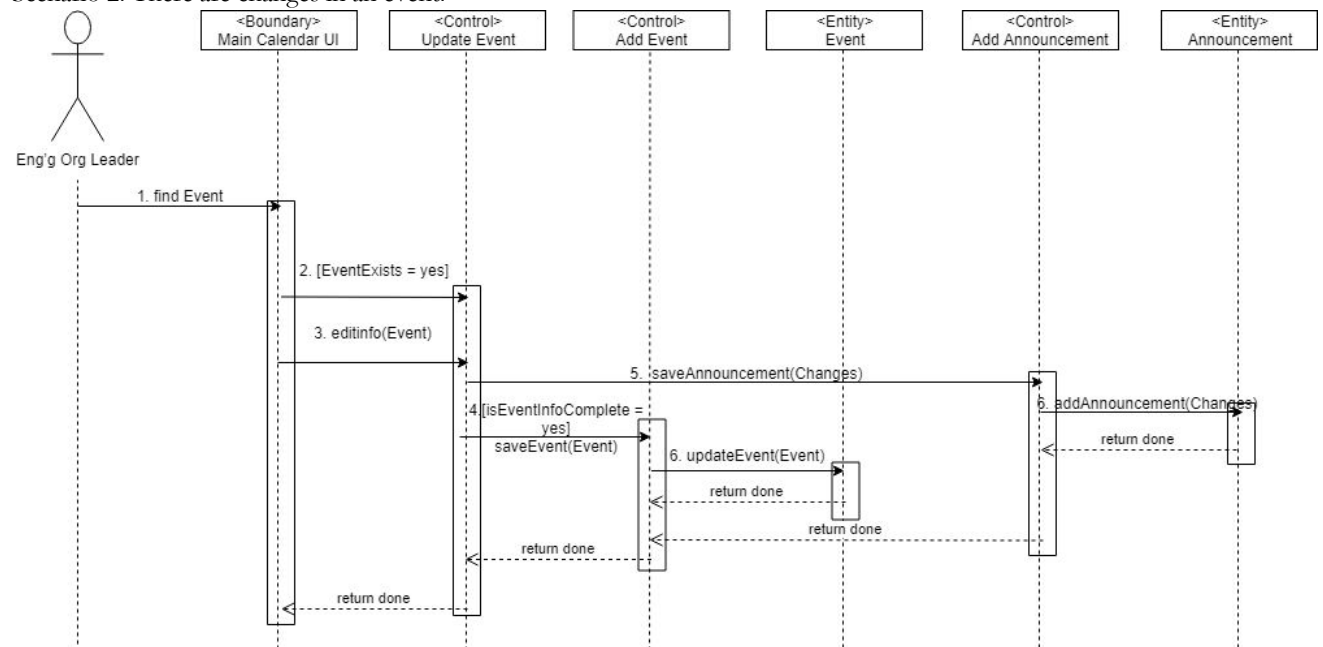
Use-Case Name: 4.0 Maintain Main Calendar

Description: This is exclusive for the Eng'g Org Leaders only. They have the power to create, edit, and delete the events that will be posted under their account. All of these will reflect on the Main Calendar.

Scenario 1: Basic Flow. EOL creates an event.



Scenario 2: There are changes in an event.



Scenario 3: An event is cancelled.

