Name TBD

Deliverable 3 Report

Features Discussed

Room/Resource Bookings

Description

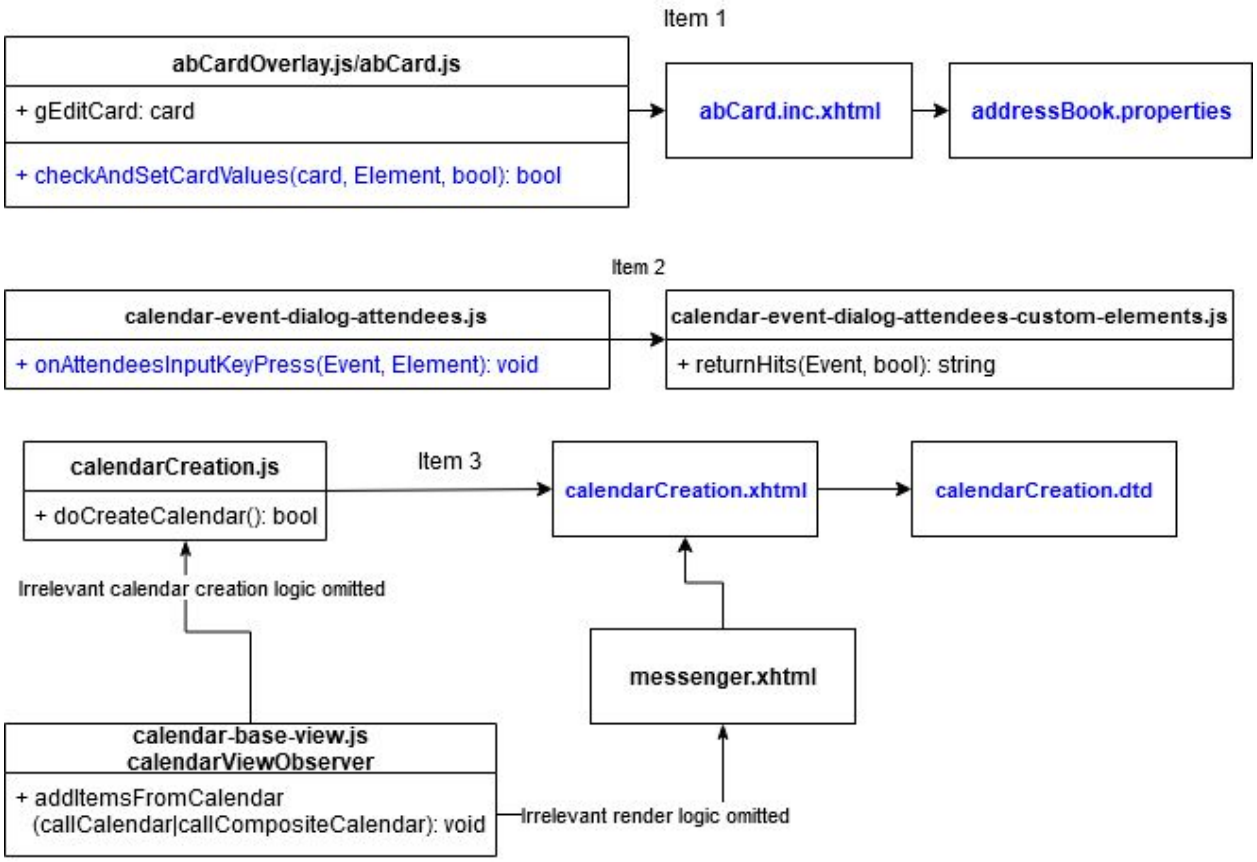
Original description on Bugzilla: https://bugzilla.mozilla.org/show_bug.cgi?id=481021

In other calendar providers, admins of a building have the ability to set up rooms (for example) as resources. Then, users are able to book those rooms using the calendar service. The result is that when the user books the room, an “invitation” is sent out to the room to see if the room is available in that time and if it is, then the room “accepts” the “invitation” and the booking is successful. Furthermore, the user should be able to view the calendar of each resource. The ticket asks to implement this feature into Lightning Calendar.

Changes/Relevant Files

There are a couple things to add.

1. Feature for user to add a new resource.
    a. A tab in the address book for resources and rooms.
        - addressBook.properties
        - abCard.inc.xhtml
    b. Code for adding resources. It should be the same for general resources and rooms, but Lightning makes a distinction. The specification for it is [here](https://icalendar.org/CalDAV-Access-RFC-4791/5-3-2-creating-calendar-object-resources.html).
        - abCardOverlay.js and abCard.js
2. In the attendees list for an event, the user is able to specify what kind of attendee is being invited, whether it be a person, a group, a resource, or a room. If resource or room is selected, then pull from the list of resources/rooms.
    a. calendar-event-dialog-attendees.js
3. In the new calendar dialogue, add UI to let the user add a calendar based on a resource or room.
    a. calendarCreation.xhtml
    b. calendarCreation.dtd



Blue text indicates modification.

Free/Busy Status Indicator Update

Description

Original description on Bugzilla: https://bugzilla.mozilla.org/show_bug.cgi?id=685542

User A subscribes to User B's free/busy calendar so that they can know when User B is available. This causes User A's day view to be filled with the blocks of time in which User B has marked themselves as busy. As well, if multiple free/busy calendars are subscribed to, these busy blocks do not differentiate themselves based on the user. This is not very helpful for three reasons. The first reason is that it would be more useful for User A to know when User B is free rather than busy. The second reason is that it is confusing if the busy blocks are not distinguishable by which user is busy. The second reason is that it is overwhelming for User A to have their day view filled with these blocks of time.

The requested change is to make it so that there is either a more general way to indicate that another user is free or busy than the time blocks. For example dialog saying, "this user is busy

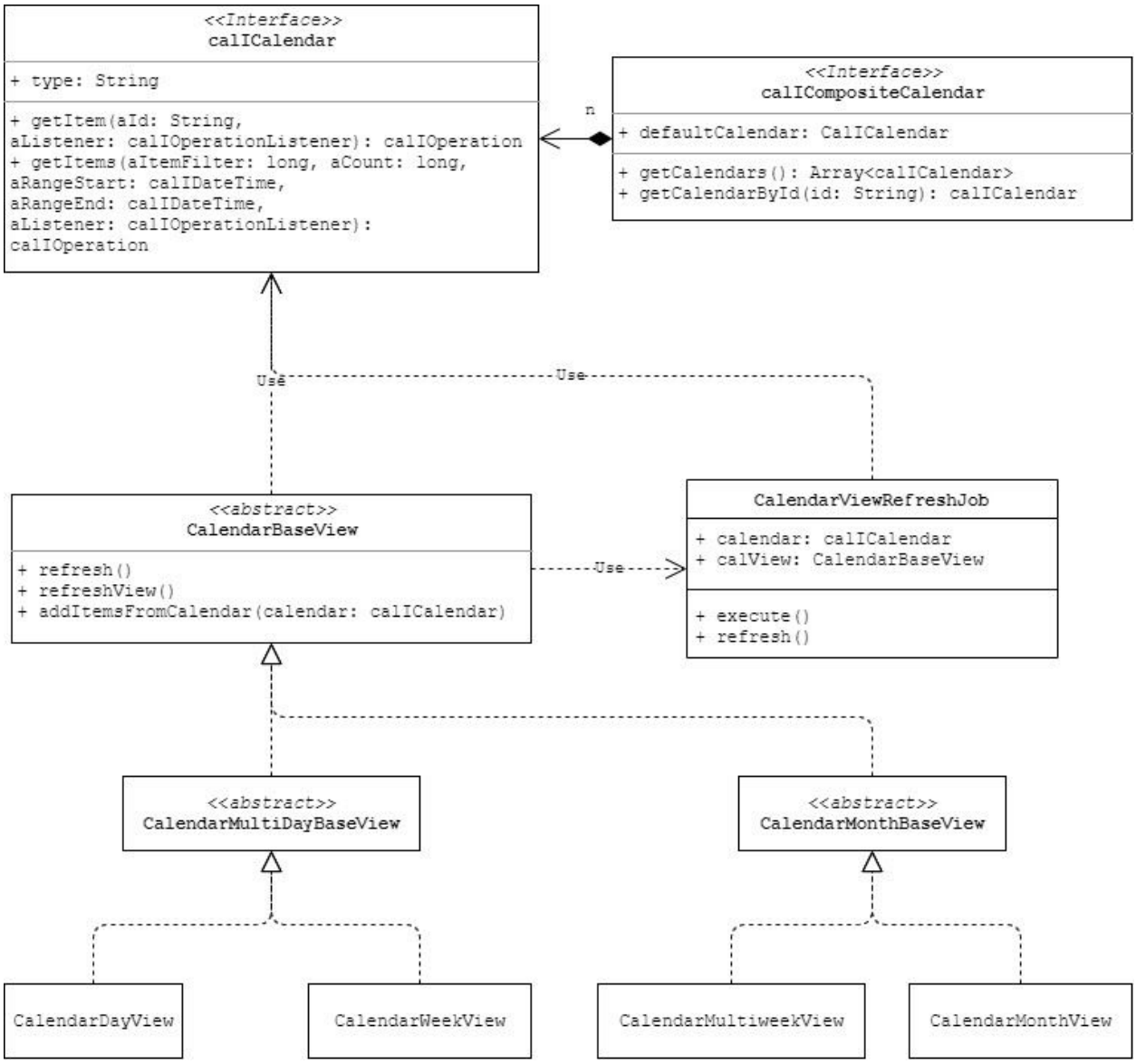
right now”. As well, a way to distinguish which user is busy in a given time block is also requested. Another alternative suggestion is to indicate other user’s free/busy status in a different manner such as highlighting time ranges rather than making entire blocks that appear the same as events.

Relevant Files

- calendar\base\content\calendar-base-view.js
- calendar\base\content\calendar-views.js
- calendar\base\public\calICalendar.idl
- calendar\providers\composite\CalCompositeCalendar.jsm

Code Diagram

Note: on this diagram many functions and variables that are not relevant to this issue are omitted to avoid information overload and to keep the diagram simpler and more readable.



Changes Required

In the file “calendar-base-view.js” there exists a class called “CalendarViewRefreshJob” whose purpose is to update a calendar view. It primarily does this through the function called “refresh” which ends up calling the function “execute”. In this function it checks for the type of calendar that is being updated. This is necessary when updating a composite calendar - a collection of many calendars. Here it also checks the types of the calendar items being displayed. All of this is called in the class “CalendarBaseView” in the function

Free/busy blocks could be filtered out by filtering away either calendars of the free/busy type or by filtering away events of the free/busy type. Once filtered out, we could treat them in a variety of ways depending on which suggestion we wish to follow. We could render them differently on the day view as highlighted ranges, but this would require a new class. Alternatively we could modify their current rendering based on a property such as the source email address to be able to differentiate them from one another if multiple are imported.

Selected Issue - Room/Resource Bookings

Selection Reason

We chose to implement the ability to reserve rooms or resources in events as it's a natural extension to the ability to add attendees to an event, which we've worked on in the previous deliverable. In addition, similar systems already exist in many other calendar add-ons for other email clients and the CalDAV standard which Lightning supports, so it will be simpler to research about how they allow calendar events to utilize resources or rooms, and likewise create an intuitive implementation for Lightning using the already-available standard.

Implementation Plan

Our Implementation Plan will be as follows:
1. Research about how room/resource bookings are created in similar applications, as well as the supported standards in Lightning (CalDAV/Libical). As Libical appears to support it, it will likely be the standard of choice that we work with.
2. Create a special type of addressee, creatable in the Address Book, called a Resource, that can be added to Events, similar to an Attendee. Like other addresses in the Address Book, it will be available for Autocomplete functionality for any applicable Events, Sendees, and so on.
3. Similar to an Attendee, it will also be able to show its free/busy status among the other Attendees. This will be structured such that a Resource can only be assigned to (at maximum) any one Event during any one time slot. In addition, it will have its own Calendar view available, and it will be able to accept or decline Event invitations automatically due to being on a server.
4. Once created, we ensure all tests pass, then submit a pull request for other group members to verify that our system works.

Tests Designed

Test 1. Ensure adding/removing Resources to/from the set of Calendars is successful.
Steps:
1. Navigate to the Calendar tab.
2. Right Click on the list of Calendars.
3. Press “New Calendar”.
4. Choose the “On the Network” option and press “Next”.

5. Choose the appropriate calendar format and provide the link to the location of the Resource's remote calendar.
6. Press "Next", add any additional identifying info, and press "Next".
7. Press "Finish" and ensure the Resource appears as a Calendar.
8. Right-click on the Calendar.
9. Choose "Unsubscribe Calendar".
10. Press "Unsubscribe".
11. Ensure the Calendar no longer appears.

Test 2. Ensure adding a Resource to an Event works.
Steps:
1. Right click on a Calendar.
2. Click "New Event...".
3. Click "Invite Attendees"
4. Type in the name of the Resource.
5. Press Enter.
6. Ensure the Resource has been added to the Attendees list.
7. Press "OK".
8. Press "Attendees".
9. Ensure the Resource appears in the Attendees list.
10. Ensure "Notify Attendees" is checked.
10. Press "Save and Close".
11. Ensure you receive an email from the Resource confirming attendance, and that the selected time slot appears busy on the Resource's calendar.

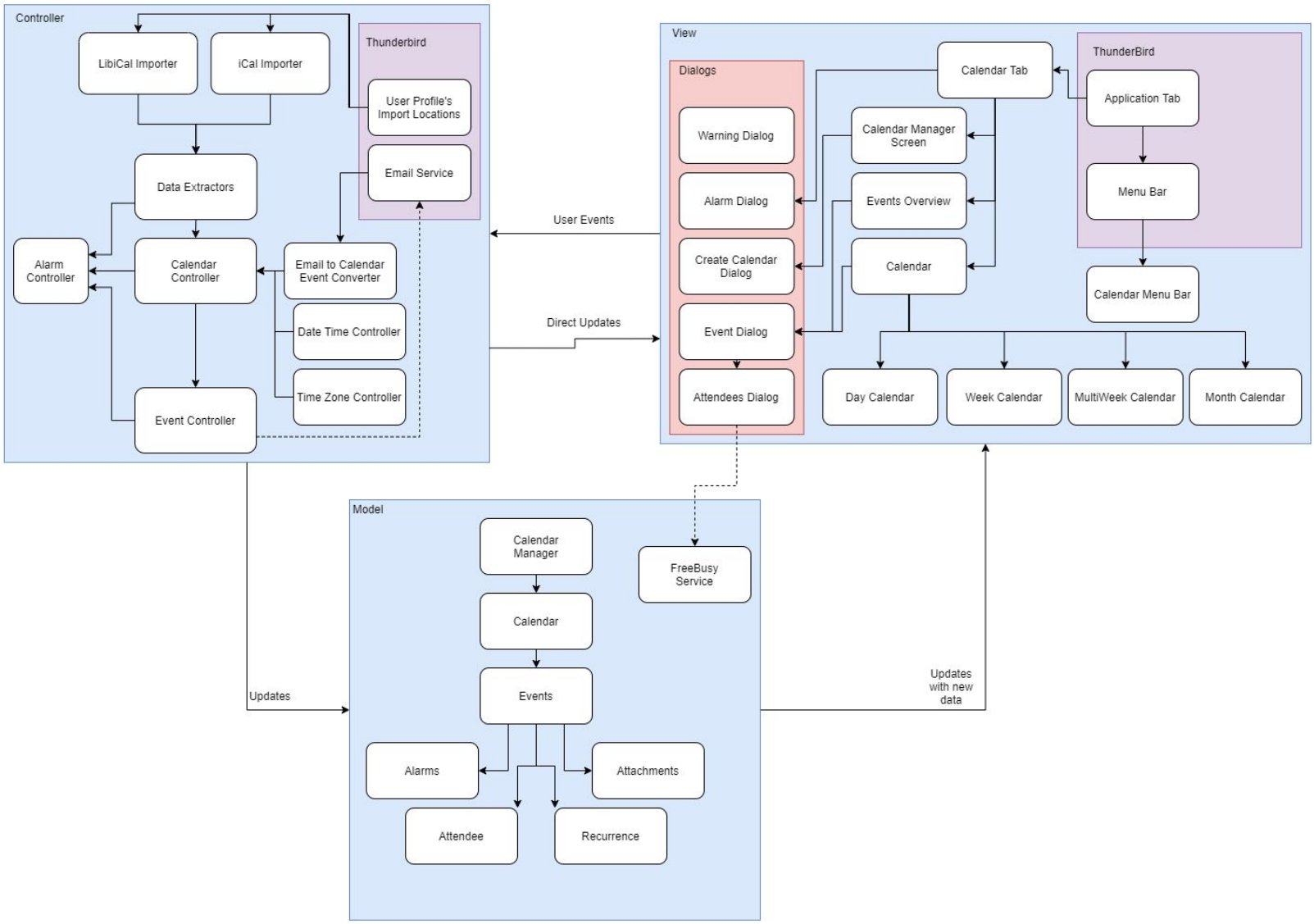
Test 3. Ensure removing a Resource from an Event works.
Steps:
1. Right click on a Calendar.
2. Click "New Event...".
3. Click "Invite Attendees"
4. Type in the name of the Resource.
5. Press Enter.
7. Press "OK".
8. Ensure "Notify Attendees" is checked.
9. Press "Save and Close".
10. Ensure you receive an email from the Resource confirming attendance.
11. Double-click on the added Event.
12. Press "Invite Attendees".
13. Remove the Resource from the Attendees list by deleting its name.
14. Press "OK".
15. Press "Save and Close".
16. Press "Attendees".
17. Ensure the Resource no longer appears in the Attendees list.

18. Press “Save and Close”.
19. Ensure you receive an email from the Resource confirming the cancellation, and that the selected time slot does not appear busy on the Resource’s calendar.

Test 4. Ensure adding a Resource with a busy time slot to an Event in that time slot fails, and modifying the time-slot to an empty one succeeds.

Steps:
1. Right click on a Calendar.
2. Click "New Event...".
3. Click "Invite Attendees".
4. Type in the name of the Resource.
5. Press Enter.
6. Select a time slot for the Event.
7. Press "OK".
8. Ensure “Notify Attendees” is checked.
9. Press "Save and Close".
10. Ensure you receive an email from the Resource confirming attendance.
11. Right click on the same Calendar as in Step 1.
12. Click "New Event...".
13. Click "Invite Attendees".
14. Type in the name of the same Resource as in Step 4.
15. Press Enter.
16. Select the same time slot for the Event as the one in Steps 2-9.
17. Press "OK".
18. Ensure “Notify Attendees” is checked.
19. Press “Save and Close”.
20. Ensure you receive an email from the Resource declining attendance.
21. Double-click on the Event created in Steps 12-19.
22. Modify the time slot on the Event such that it does not conflict with the time slot on the Event from Steps 2-9.
23. Press “Save and Close”.
24. Ensure you receive an email from the Resource confirming acceptance.

Updated Overall Project Design Diagram



Now that the team has more experience with the thunderbird application we can see clearly that the application closely resembles that of a web app with an MVC model. This was likely chosen as the architecture due to being built on the Mozilla core. There are multiple other links between each section that are not fully documented for diagram clarity reasons.