Software Requirement Specification

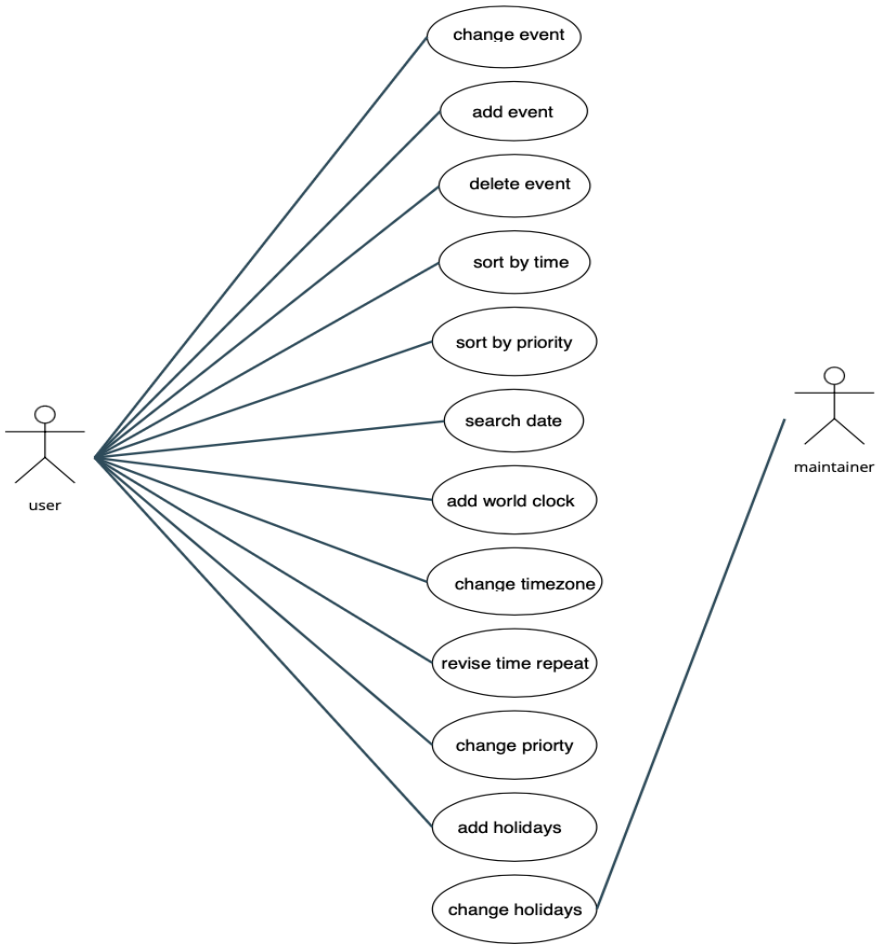
CP317 Software Engineering

Group 10

Instructor: Dr. Zia Ud Din

Table of content:

Use case diagram:



1. Introduction

Calendar317 is designed as a straightforward and user-friendly multi-platform software developed on Mac and Windows. The software combines the advantage of multiple productivity software and the purpose of it is to help people organize, plan, and record their daily critical tasks and increase their efficiency.

1.1 Purpose

The purpose of this document is to describe the service provided by Calendar317 including its features and functions. This document is aimed at the user eager for the basic logic of the system and helping the developer to deepen their understanding of the software. All current and future developers carrying out future development should follow the guidelines of this document.

1.2 Scope

Calendar317 has six modules in total and three of them are called Eventer as a union. The event added to the Eventer will synchronize with the rest of the module in the Eventer. Any events added to the Eventer can be marked by color or added priority. The other three modules are basic function, plug-in adder, and account. It is worth mentioning that the plug-in adder can add recommend events, clock, event time analysis, and export events. The details of those will be discussed in the later document.

1.3 Definitions

Module: The main type of each single integrated function can be used independently.

Eventer: The event management module in the software, including month view table, to do list, timeline.

Month View Table: One of the modules in the software.

1.4 Reference

IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998

https://bohr.wlu.ca/cp317/notes/IEEE_830.pdf

1.5 Overview

After introduction, this document goes into details on the description and function of main settings and plugs. In the end, the constraint of these and process of each function will be mentioned later.

2. Overall description

2.1 Product perspective

Calendar317 is based on Java GUI with front end sections and back-end sections.

Front End:

The front end is the calendar interface that the user can interact with, and it is based on Java GUI. It is a template for holding elements of a calendar, including date, time, textbox, events, buttons.

Back End:

The back end is written in java. It is responsible for storing user notes, displaying the dates, reminder functions and calculations. The program is independent, changing the operating system does not affect the program usability if a java virtual machine is installed in the system.

2.2 Product function

Notice each event has the property of customizing time, label, and priority. Only public information such as holidays in plug-in flow can be accessed by both user and maintainer, the other flow is fully controlled by the user.

Built-in event property flow:

The built-in flow provides a basic function allowing users to search and jump to specific dates.

Month view table flow:

One of the core functions of Calendar317 and it gives the user a general view over a month table. It automatically displays the current month highlight current date. Each date takes one cell, and each cell is used for adding numbers of events.

To-do list flow:

It is used to create a list of events or tasks that the user needs to complete daily.

Timeline flow:

It is a list of major events that the user arranged in the order they are going to happen.

Plug-in flow:

There are two modules in plug-in flow and there will be more functions added in the future. One of the modules is called World Clock and it displays the time for various cities around the world and it is used for changing the timezone. Another function is called Holiday and it allows the user to choose the festival in any country they want.

2.3 User Characteristics

The targeted user group of Calendar317 are people who have busy schedules or those who want to improve their time management skills. It is not necessary for the user to have deep technical knowledge and the user can use it with only a mouse click and text input for naming the event.

2.4 Constraints

Due to the limitation of client expectations, only Mac and Windows versions were released for this software.

Due to the limitation of funding, we cannot afford to rent a server, a software-based application cannot synchronize recorded events to another platform with the same software.

Due to the limitation of the Java GUI technique, the user using the software should have a Java virtual machine or Java Runtime Environment installed on their computer.

2.5 Assumptions and dependencies

This document is based on the Calendar317 prototype, the actual application may appear different from the requirement that has been listed in this document. More functions will be added to future updates.

This is the second iteration of the application and the assumption for Calendar317 is software developed based on Eventer and omit the other function which allows developers to quickly release a minimal viable product.

3. Requirements Specification

3.1 External interfaces

Calendar317 is pure software and there is no external interface for this system.

3.2 Functions Requirement

Use Case ID	UC-01
Use Case Name	Search Date
Actors	User
Use Case Overview	Allows any user to search for a date.
Trigger	The user wants to search for a date using calendar317 The user clicks “Down arrow” button
Precondition	1. A valid date is entered
Normal Flow	1. The use case begins when the user presses the “Down Arrow” button on the top left corner of Calender317. 2. Select desired year by pressing the “left arrow” button or the “right arrow” button. 3. Select the desired month. 4. Correct date will be shown in the calendar cell.
Use Case Associations	UC-04
Priority	Medium

Use Case ID	UC-02
Use Case Name	Add Event
Actors	Users
Use Case Overview	It allows the user to add Event to Calendar Cell.
Trigger	The user wants to add an event to Calendar317.
Precondition	None
Normal Flow	**Case 1**: Add Event in To-Do list and Timeline 1. The use case begins when the user presses the “Add Event” button.

	2. An empty event is created. 3. User enters an event name. 4. (Change Priority) UC-08 is performed (optional). 5. User selects an event starting time. 6. User selects an event due date 7. User enters a detailed description of events, e.g., location of event and participants of event. Users may choose to skip this step. 8. User presses "save" button to save event. 9. Event name, starting time, due date, and optional description such as priority, location and participant are recorded and displayed on the To-Do list. 10. The Use case ends successfully. **Case 2**: Add Event in Calendar 1. The use case begins when user right clicks a Calendar Cell. 2. The calendar cell displays a list of alternatives that are available for the user to choose. The user selects "Add Event" 3. An empty event is created and shown in the calendar cell that the user selects. 4. User enters an event name. 5. (Change Priority) UC-08 is performed (optional). 6. User selects an event starting time. 7. User selects an event due date 8. User enters a detailed description of events, e.g., location of event and participants of event. Users may choose to skip this step. 9. User presses "save" button to save event. 10. Event name, starting time, due date, and optional description such as priority, location and participant are recorded and displayed on the To-Do list. 11. The Use case ends successfully.
Alternative Flow	**A 1: Missing due date** In both case 1 and case 2, the user is required to select a due date. If due date is not given by user, then 1. The use case ends with a failure condition, user will have to redo all the step.

Use Case Associations	UC-05
Priority	High

Use Case ID	UC-03
Use Case Name	Delete Event
Actors	Users
Use Case Overview	It allows user to delete Event from Calendar Cell.
Trigger	The user wants to delete an event from Calendar317.
Precondition	A valid date is selected At least one Event are added before using delete function
Normal Flow	Case 1 1. The use case begins when the user selects To-Do list. 2. The user right clicks an unneeded event on To-Do list. 3. The selected event displays a list of alternatives that are available for the user to choose from. The user selects “Delete Event”. 4. Event will be removed from To-Do list, Timeline and Calendar. 5. The use case ends successfully Case 2 1. The use case begins when user selects Timeline. 2. The user right clicks an unneeded event on Timeline. 3. The selected event displays a list of alternatives that are available for the user to choose from. The user selects “Delete Event”. 4. Event will be removed from To-Do list, Timeline and Calendar. 5. The use case ends successfully Case 3 1. The use case begins when user selects Calendar. 2. The user right clicks an unneeded event on Calendar Cell. 3. The calendar cell displays a list of alternatives that are available for the user to choose. The user selects “Delete Event”.

	4. Event will be removed from To-Do list, Timeline and Calendar. 5. The use case ends successfully.
Use Case Associations	None
Priority	High

Use Case ID	UC-04
Use Case Name	Change event
Actors	User
Use Case Overview	It allows the user to modify existing events, e.g., change event name, change event time, change priority, revise location and participant.
Trigger	User wants to change information of an event in Calendar 317.
Precondition	At least one event is created for change.
Normal Flow	1. The user left clicks an event. 2. The event displays a list of alternatives that are available for the user to choose. The user selects "Change Event". 3. A page with detailed information of the event is displayed. 4. The user can change events in this step. 5. The user clicks "Save" button. 6. New records are added and saved. 7. The use case ends successfully
Alternative Flow	**A 1: Missing due date** Users are required to select a due date. If due date is not given by user, then 1. The use case ends with a failure condition, user will have to redo all the step.
Use Case Associations	UC-01
Priority	Medium

Use Case ID	UC-05
Use Case Name	Repeat Event

Actors	User
Use Case Overview	It allows the user to edit the time of event repeat. The repeat event is used only for the event repeat at least once a week.
Trigger	User wants to repeat a single eve weekly.
Precondition	1. At least one day is added in a week
Normal Flow	1. The use case begins with user click “add event” in timeline and month view table, the repeat event will automatically appear as an option for the user choosing mon. to sun. in a week for a single event. 2. Case 1: user text in times of repeat for an event, and the event will repeat n times. Case2: user chooses the end date from the date board to stop the repeating event. 3. The repeated event is rendered on the Eventer. 4. The user clicks region other than the add event 5. The use case ends successfully
Alternative Flow	1. Users choose days but click another region except for two cases listed. The data is abundant since it is an infinite loop.
Exception Flow	
Associations	UC-02
Priority	Medium

Use Case ID	UC-06
Use Case Name	Sort by priority
Actors	Users
Use Case Overview	It allows the user to organize their task by priority.
Trigger	The user wants to sort their events by priority.
Precondition	1. At least two events are added before using sort function
Normal Flow	1. The use case begins when the user selects To-do List. 2. The user clicks “Sort” button. 3. The To-do list displays two options available. The user selects “priority”. 4. Events are sorted by priority. 5. The use case ends successfully.

Use Case Associations	UC-07
Priority	High

Use Case ID	UC-07
Use Case Name	Sort by time
Actors	Users
Use Case Overview	It allows the user to organize their event by time. Events that are becoming due have higher priority and will be listed first.
Trigger	The user clicks “Sort” button
Precondition	1. At least two tasks are added before using sort function
Normal Flow	1. The use case begins when the user selects To-do List. 2. The user clicks “Sort” button. 3. The To-do list displays two options available. The user selects “Time” 4. Events are sorted by time. 5. The use case ends successfully.
Use Case Associations	UC-06
Priority	High

Use Case ID	UC-08
Use Case Name	Change timezone
Actors	User
Use Case Overview	It allows any user to alter Calendar317’s built-in timezone so that all events and tasks will switch their time to the newly selected timezone.
Trigger	The user wants to Change timezone.
Precondition	Change timezone plug-in is installed.
Normal flow	1. The use case begins when the user press “Add plug-in” button. 2. The user presses the “clock” button on the left list of Calender317. 3. The user changes the timezone and city. 4. The timezone and city are changed.

	5. The use case ends successfully.
Use Case Associations	UC-09
Priority	Medium

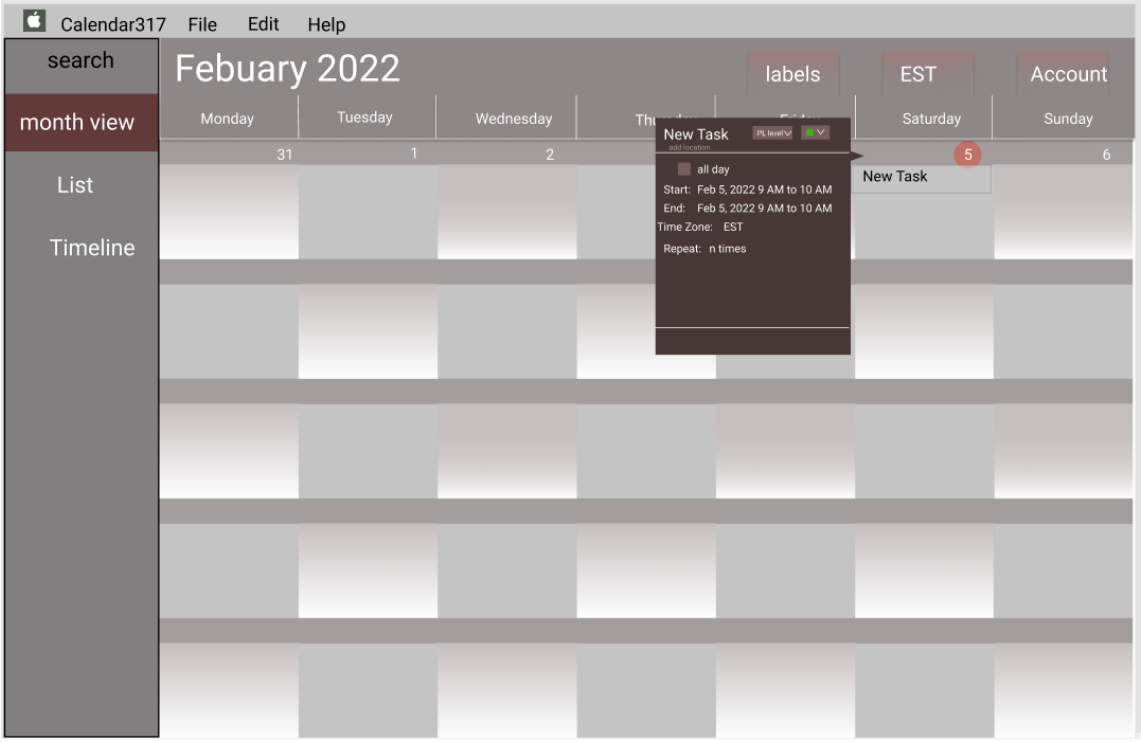
Use Case ID	UC-09
Use Case Name	Add world clock
Actors	User
Use Case Overview	It is a plug-in function that includes world time and date for cities around the world.
Trigger	User wants to check time for another city.
Precondition	Add world clock plug-in is installed.
Normal flow	1. The use case begins when the user press “Add plug-in” button. 2. The user presses the “clock” button on the left list of Calender317. 3. The user adds world clock 4. The world clock has changed. 5. The use case ends successfully.
Use Case Associations	UC-08
Priority	Low

Use Case ID	UC-10
Use Case Name	Add holiday
Actors	User, Maintainer
Use Case Overview	It allows maintainer to change holiday. Users can browse different holidays.
Trigger	Users want to browse different holidays.
Precondition	Add holiday plug-in is installed.
Normal flow	1. The use case begins when the user press “Add plug-in” button. 2. The user presses the “clock” button on the left list of Calender317. 3. The user adds holiday. 4. The holiday has changed.

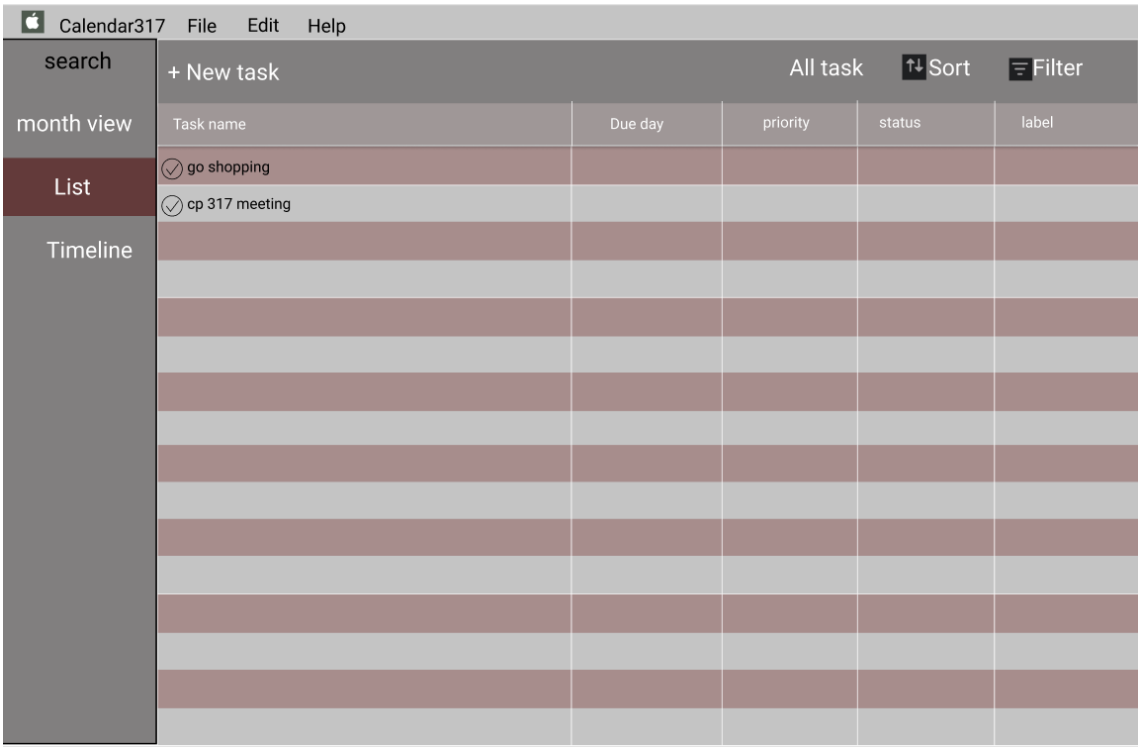
	5. The use case ends successfully.
Use Case Associations	None
Priority	Low

Appendix A: prototype

Month view page



To-do-list page:



Plug-in Clock page: