

Date Tracking Application

<https://github.com/RichieZrq/OOP-Final-Project.git>

OOP_Project_14
Richie Zhang

Date of Submission: 12/13/2023

Table of Contents

NOTE: RUN MainGUI.java to start the application

- **System Analysis**
 - Project Description
 - General Description
 - Goals and Benefits
 - System inputs and outputs
 - Design Details
 - Uses Cases Diagram and use cases description
- **System Design**
 - Sequence Diagrams
 - Class diagram
- **Conclusion**
 - Difficulties
 - Next Steps
- **Appendix**

System Analysis

Project Description

General Description

The date tracking application is designed to help its users manage and track important dates efficiently. Its primary function is to display important dates to users in ascending order based on their proximity, offering a clear and intuitive display of upcoming events. The application caters to both personal and professional needs, allowing users to categorize dates into various types such as work, holidays, anniversary, or any custom type the users create. Users can also adjust the display using the “sticky to top” option and selecting to only display a certain type of date to better organize their dates

Goals and Benefits

- To provide a user-friendly platform for tracking important dates
- Enable users to flexibly categorize and prioritize dates based on type and urgency
- Offer a date calculator feature for computing days between two dates

System Inputs and Outputs

Inputs

- Dates and date information (name, stickied status, customized date types)
- Two dates from the date calculator feature
- Type of date to display

Outputs

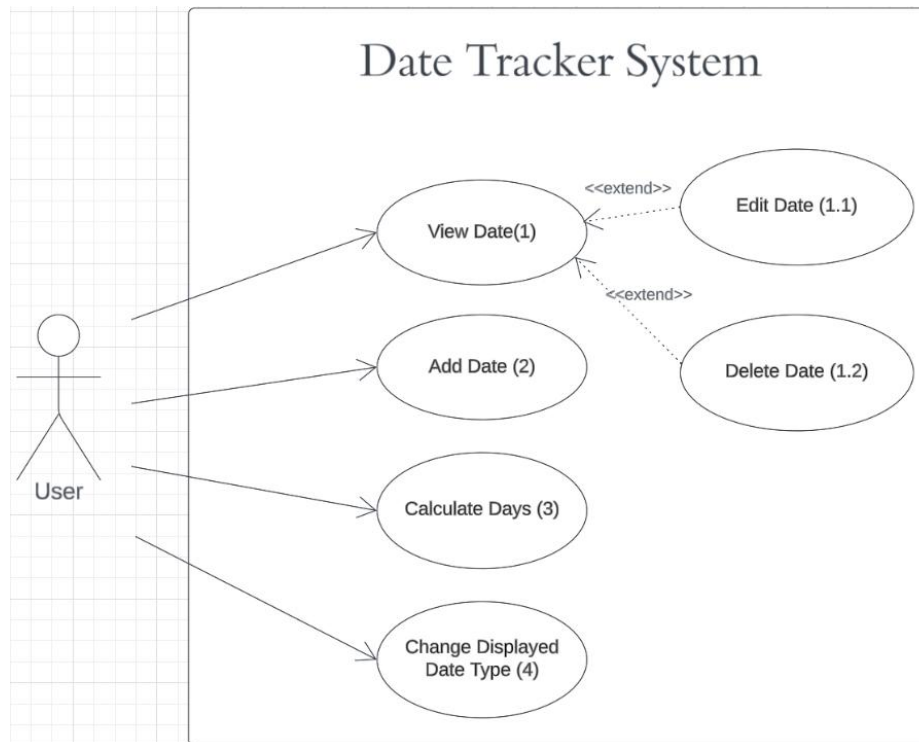
- Display of upcoming dates in an organized manner
- Calculation results from the date calculator feature

Design Details

- The application's backend will be coded in JAVA
- GUI development will be based on Java swing
- Data is stored in .dat files and is read every time the application is opened

Use case Diagram and Use Case Descriptions

Use Case Diagram



Descriptions

UC number: 1	
Overview	Allows the user to view a selected date
Related Use Case	1.1
Actors	User

UC number: 1.1	
Overview	Allows the user to edit a selected date's previous information
Related Use Case	1
Actors	User

UC number: 1.2	
Overview	Allows the user to delete a selected date
Related Use Case	1
Actors	User

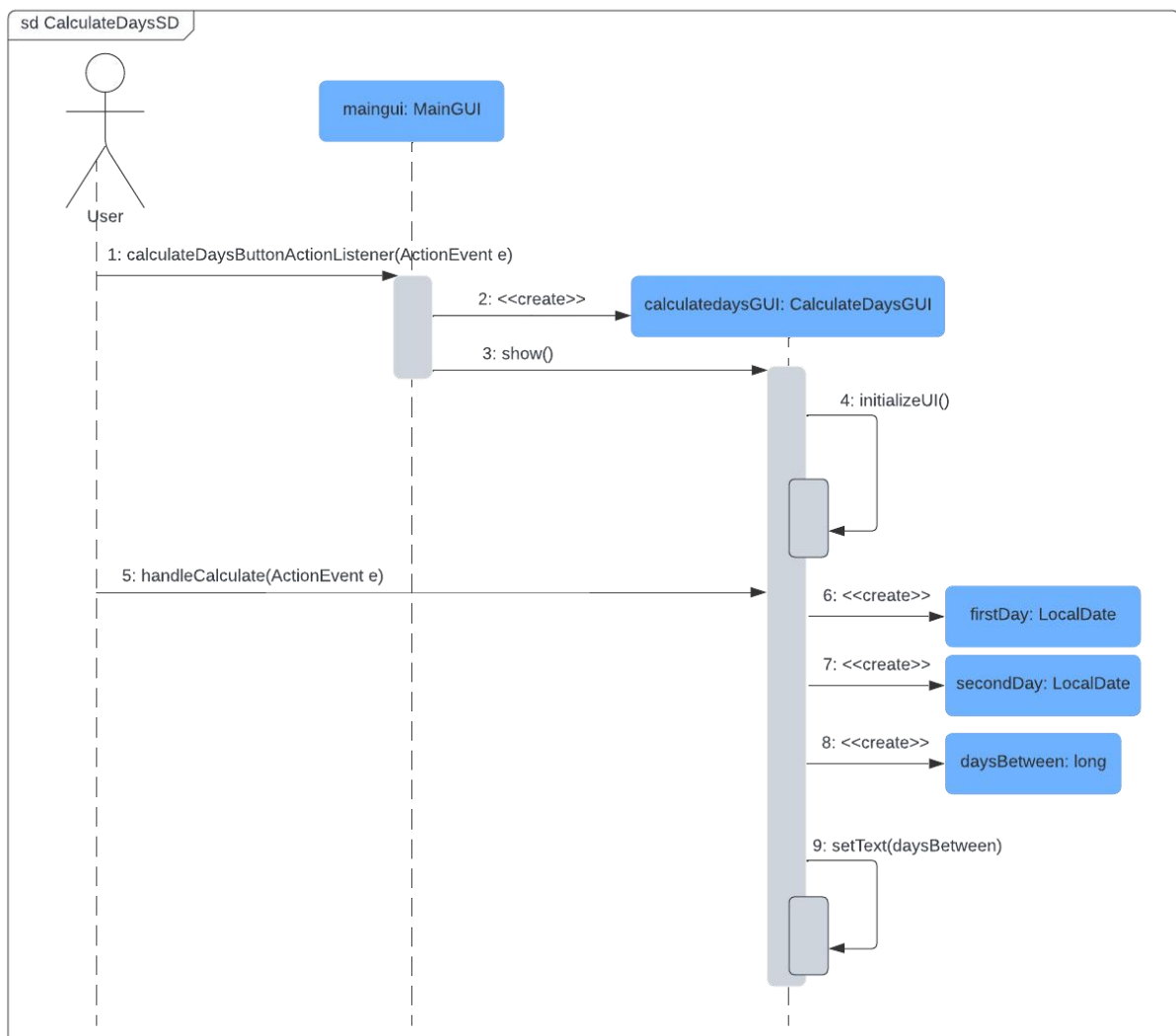
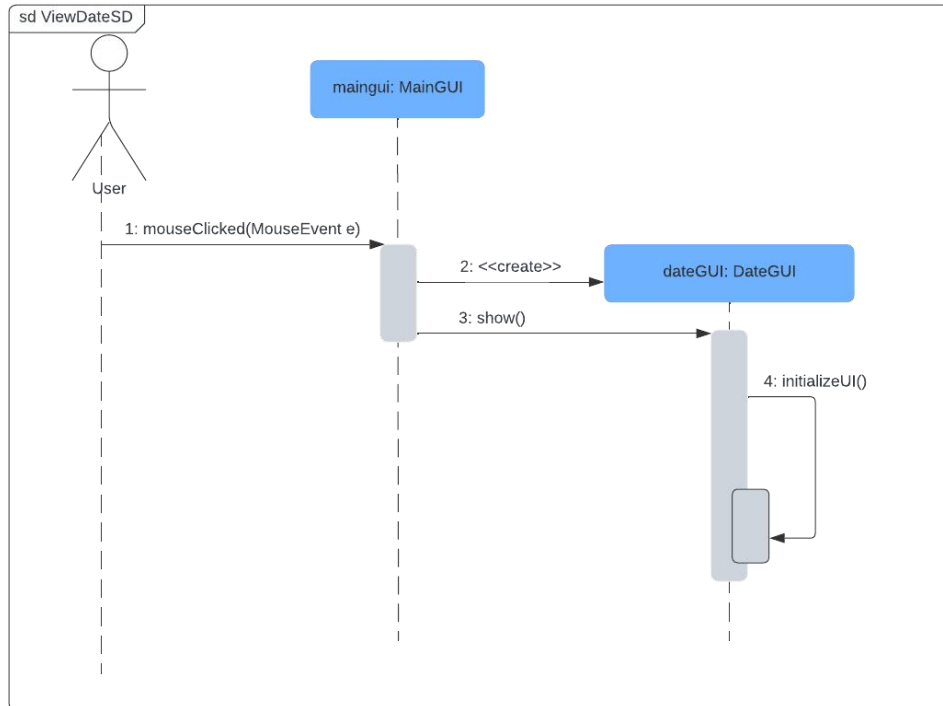
UC number: 2	
Overview	Allows the user to add a new date
Related Use Case	None
Actors	User

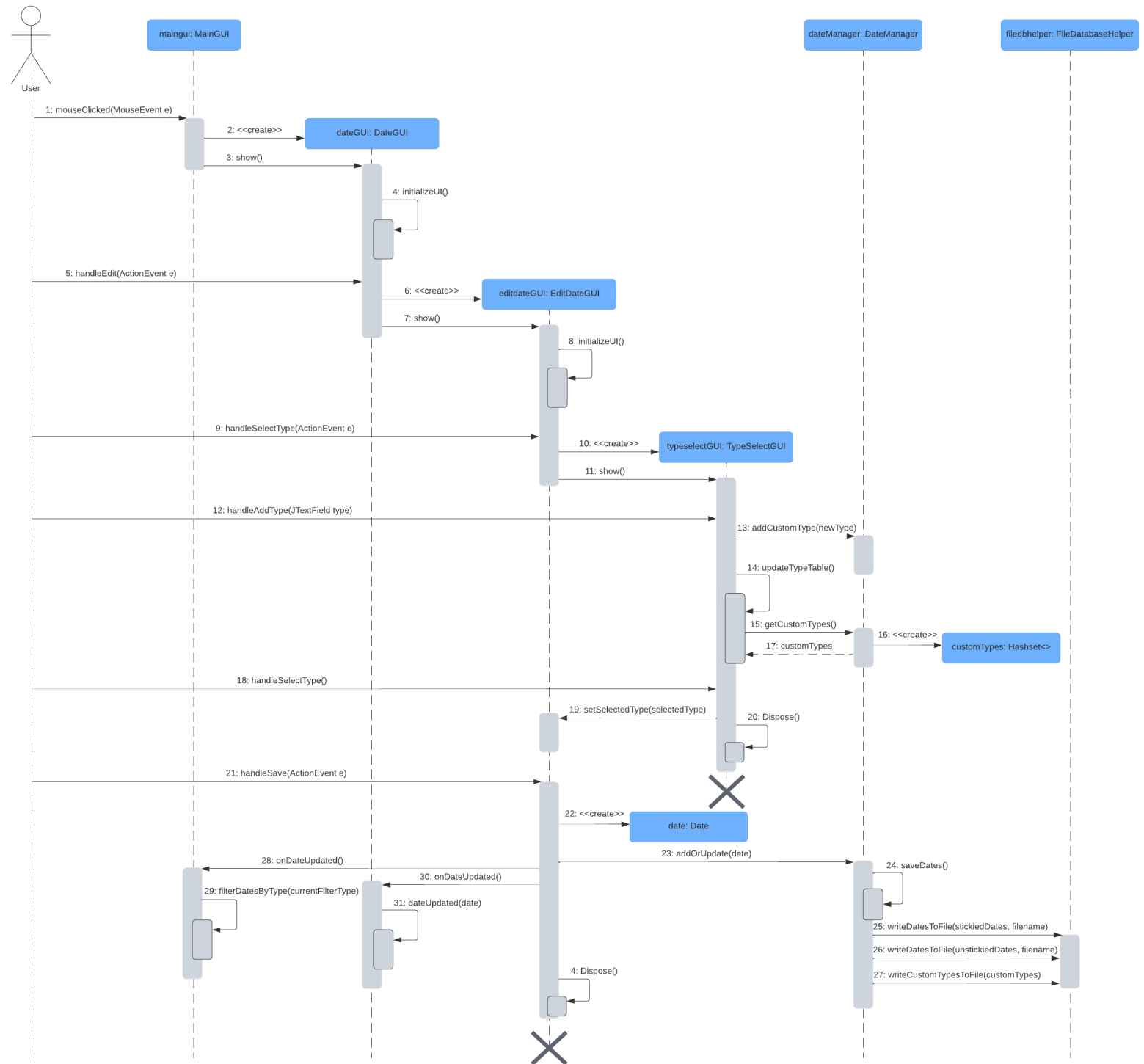
UC number: 3	
Overview	Allows the user to calculate the days difference between two dates
Related Use Case	None
Actors	User

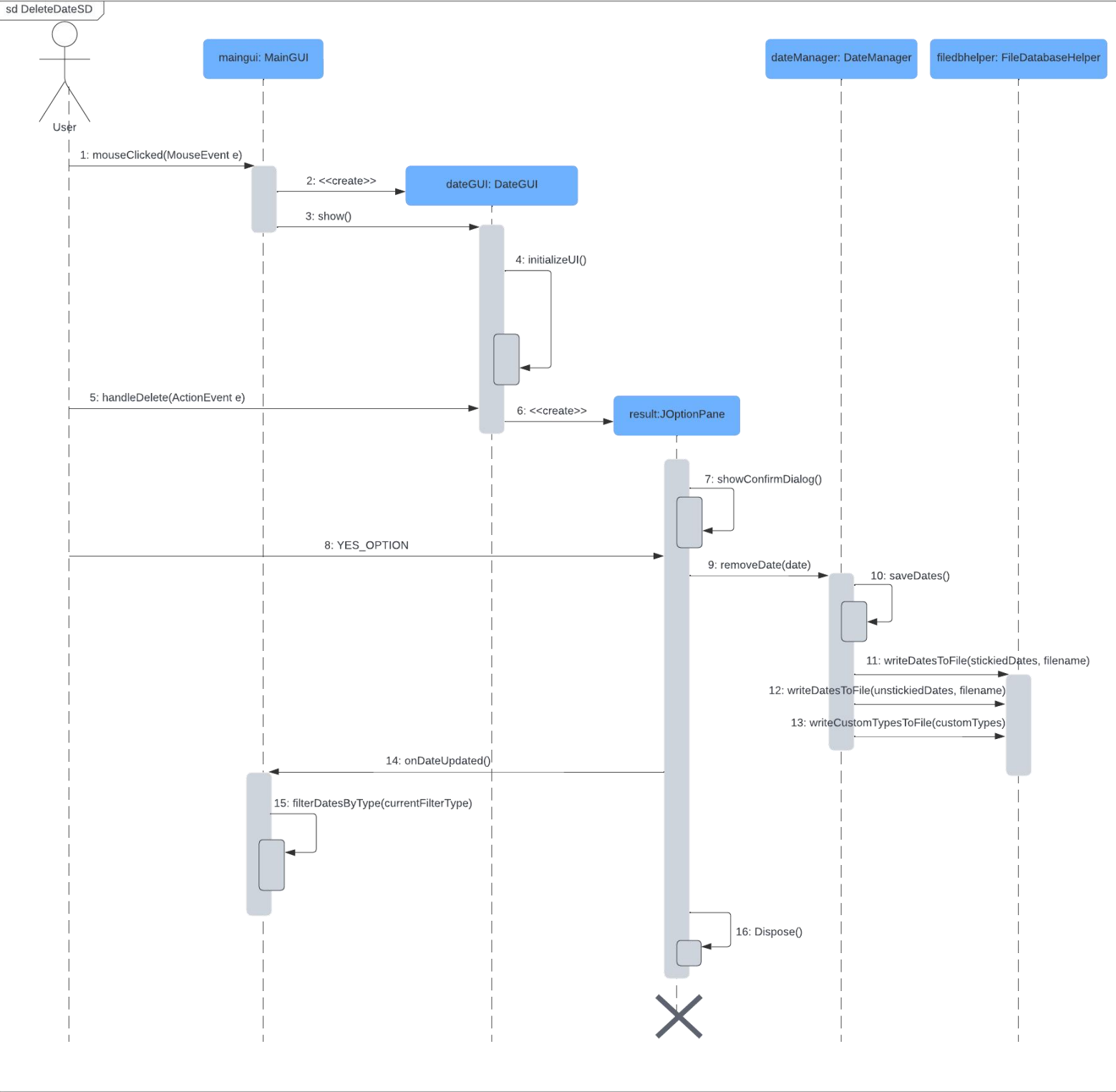
UC number: 4	
Overview	Allows the user to change the type of dates displayed
Related Use Case	None
Actors	User

System Design

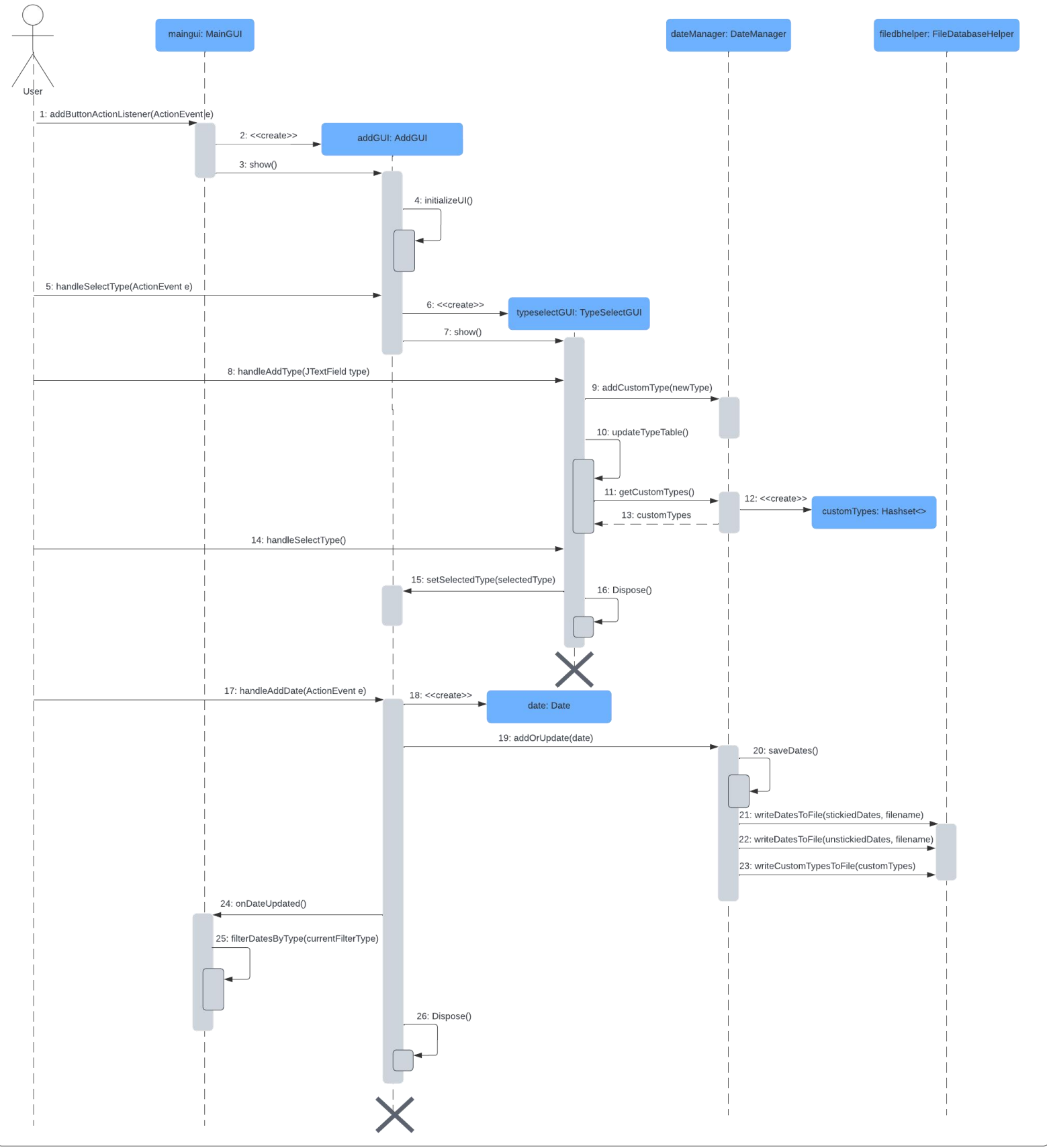
Sequence Diagrams



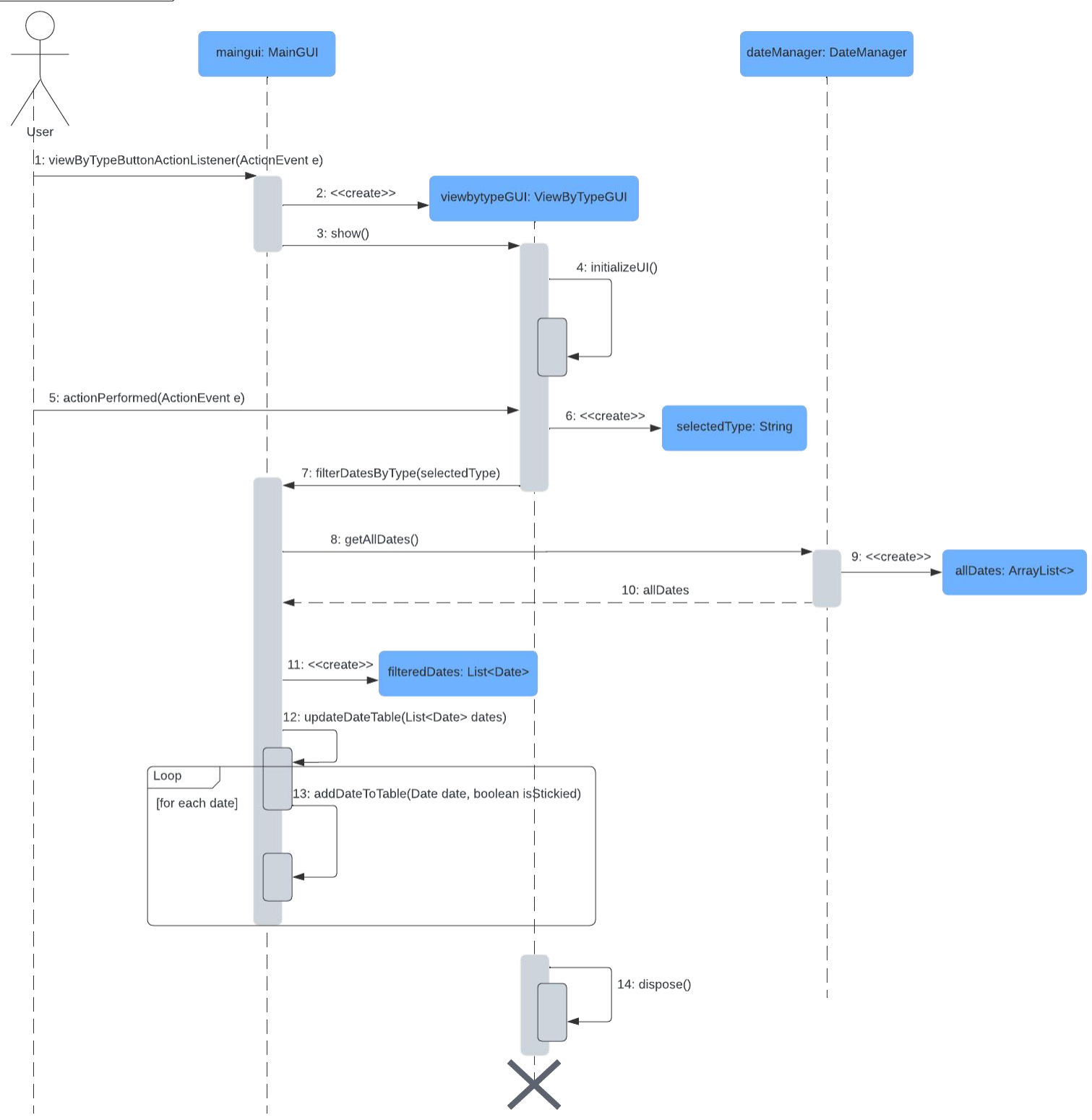




sd AddDateSD



sd ChangeDisplayedDayTypeSD



Class Diagram



Conclusion

Difficulties

I had trouble implementing the MainGUI table as I did not have prior experience. The table structure I envisioned was complicated, and I had to use some inefficient tricks to make it function properly. Specifically, it was difficult to separate the table into stickied and unstickied statuses and highlight only the stickied dates.

Formatting the GUIs by purely changing the code was tedious as I had to move them step by step to where I wanted them to be. I learned a ton of details around java's swing package as I utilized many features that were not covered during class. Some examples may include BorderLayout, HorizontalGlue, TableCellRenderer, Jdialog, and many more.

Next Steps:

Throughout the project, there were many design details with room of improvement. Specifically, updating tables and lists were not optimized as I simply used sort() instead of potentially implementing a more suitable and efficient data structure such as a PriorityQueue.

Secondly, the UI's design could be improved drastically. I only built a simple basis of the GUIs due to how much time I committed on learning and implementing JFrame. Moreover, if I were to expand on this project, I would implement a feature where the application is always stickied on the user's desktop for convenience.

A final improvement could be to change the user data's storage method to using an actual database like MySQL to better support the application. It would also be a huge improvement to implement a user login system so each user can have their own dates stored in a common database instead of having to store their data in a local file.

Appendix

<https://www.geeksforgeeks.org/java-time-localdate-class-in-java/>

<https://www.geeksforgeeks.org/java-jframe/>

<https://www.geeksforgeeks.org/java-swing-jdialog-examples/>

<https://docs.oracle.com/javase/7/docs/api/javax/swing/Box.html>

<https://docs.oracle.com/javase/8/docs/api/java/time/temporal/ChronoUnit.html>

<https://www.geeksforgeeks.org/java-swing-jtable/>

<https://docs.oracle.com/javase/7/tutorial/swing/EventListener.html>

<https://www.geeksforgeeks.org/comparator-interface-java/>

<https://docs.oracle.com/javase/8/docs/api/javax/swing/table/DefaultTableCellRenderer.html>

https://www.tutorialspoint.com/swing/swing_windowadapter.htm