CSE 222 HOMEWORK #1

Orhan Aksoy

09104302

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Requirements

1.1. Overall Description

The application will be a graphical user interface (GUI) that can keep a calendar. The sample program's appearance will be as follows:

Figure 1 Calendar GUI

When the program first runs, it shows the current date. If the user clicks on one of the days, then a new dialog box pops up and displays a textbox that can be used to enter information for that day. The information for a specific day is not lost and kept on disk to be able to restore later.

1.2. Requirement Definitions

1.2.1. The application shall be written with java/swing.

1.2.2. The application's main window shall have no borders and no title bar.

1.2.3. There will be three main parts of the GUI from top to buttom: The month/year selection bar, the day titles and the day grid.

1.2.4. The month/year selection bar shall be surrounded with a dashed line. It shall show the current month and year in Bold characters.

1.2.5. The current date shall be shown on the month/year selection bar after initialization.

1.2.6. There shall be two arrows on the right side of the month/year selection bar: When the user clicks on the left arrow, the current month/year shall be decremented.

When the user clicks on the right arrow, the current month/year shall be incremented.

1.2.7. The arrows on the month/year selection bar shall be highlighted when clicked on.

1.2.8. The day grid shall be updated according to the selected month/year pair after the month update on the month/year selection bar.

1.2.9. The day titles shall be shown below the month/year selection bar. The titles shall be written from Sunday until Saturday from left to right in 7 columns with appropriate abbreviations.

1.2.10. The day grid shall have 7 columns and 6 rows. The 1st of the selected month shall be on the appropriate column of the first raw. The following days shall be shown starting from the next cell until the last day of the selected month.

1.2.11. The days of the current month will be highlighted. The days before the 1st and after the last of the selected month shall also be shown but will not be highlighted.

1.2.12. When the user clicks on one of the days on the day grid, note entry dialog box shall pop up for that day.

1.2.13. The title of the note entry dialog box shall have the current date, shall contain a list of notes entered for that day and shall also have a field for the user to add a new note for that day.

1.2.14. If there's at least one note for a day on the day grid, that day shall be shown in Bold.

1.2.15. The application shall store the notes in a binary file on the disk. The file contents shall be immediately updated upon new note entry.

1.2.16. The application shall restore the stored notes from the binary file during application initialization.

1.2.17. The days with stored notes shall be shown with bold font after initialization following the requirement 1.2.14

2. Use Case Tables

2.1. Application Initialization

Table 1 Initialization Use Case

STEP	User's Action	System's Response
1	User issues a command to the operating system to load and run the calendar application.	
2		The calendar application is started. The current date is read from the system and the current month/year on the month/year selection bar is set to the current date. The initial note list is restored from the note file into the calendar note list. The days with note entries are highlighted with bold fonts on the active month grid.

2.2. Changing Current Month

Table 2 Changing Month Use Case

STEP	User's Action	System's Response
1	User clicks on one of the arrows on the month/year selection bar.	
2		If the user clicks on the right arrow, the right arrow is highlighted, and the current month is incremented on the month/year selection bar. If the current month changes from December into January of the next year, the year is also incremented. The day grid contents are updated according to the current month: - The days are updated on the day grid - The days with notes are highlighted with bold font. If the user clicks on the left arrow, the current month is decremented and rest of the operations are same.

2.3. New Note Entry / Note List Display

Table 3 Note Entry/List Use Case

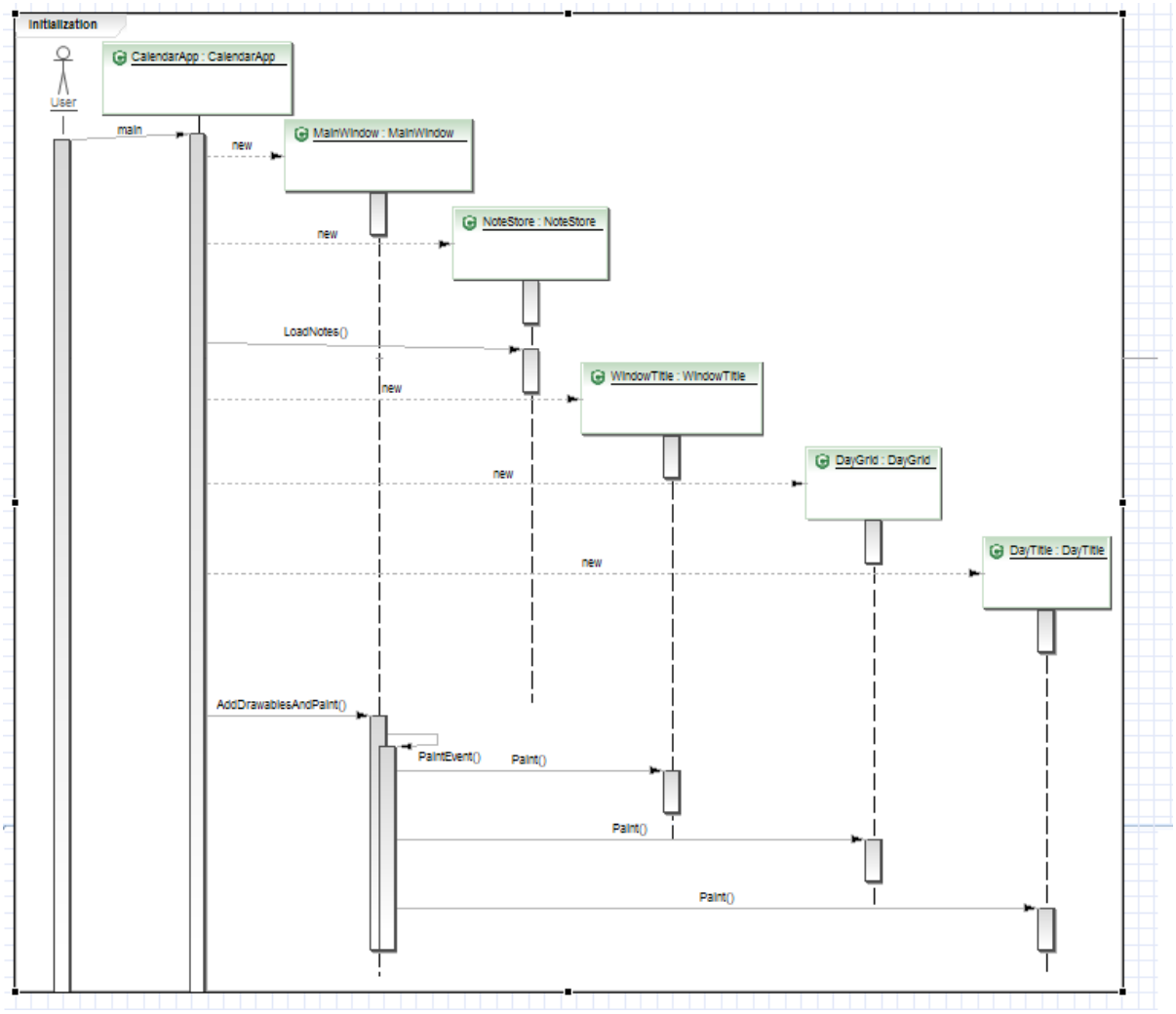
STEP	User's Action	System's Response
1	User double-clicks one of the days on the day grid.	
2		A dialog box pops up with the title "Notes of the day"followed with the selected date. The dialog box lists the current list of notes, and also has an edit field to enter a new note for that day.

3	The user enters a new note	After entering the note, if the user closes the window rather than pressing OK, the entered note is discarded and the window is closed.
4		The entered note is stored in the selected date's note list and the dialog box is closed.

3. UML Diagrams (Class and Sequence Diagrams)

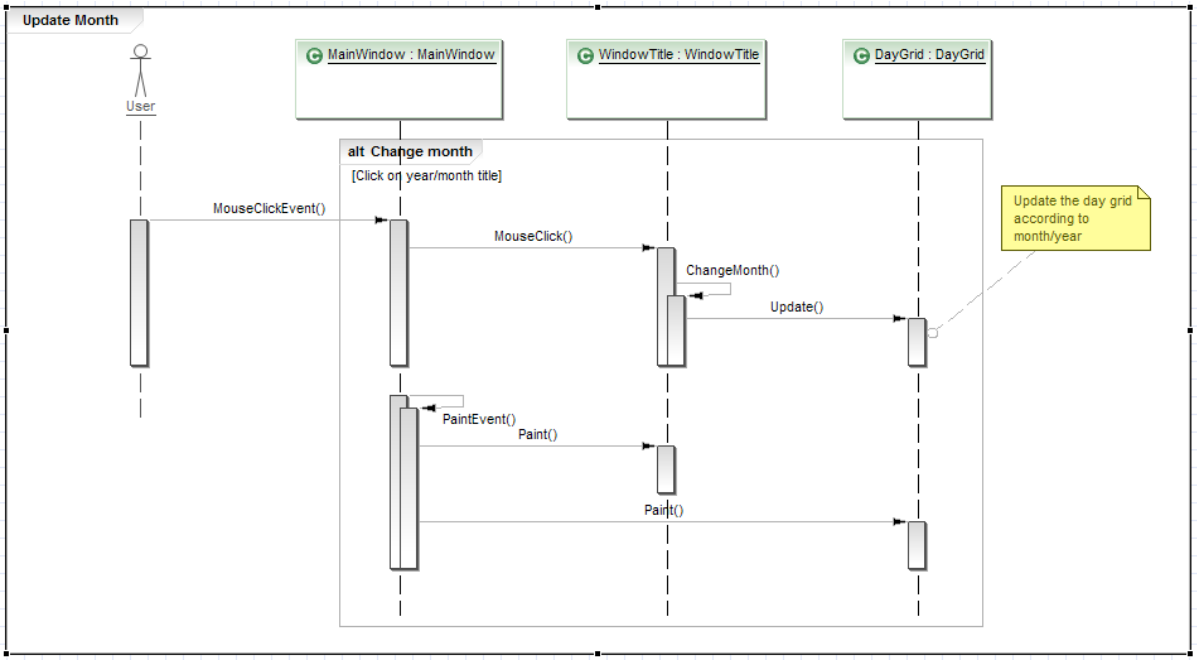
3.1. Initialization Sequence Diagram

Figure 2 Initialization Sequence Diagram



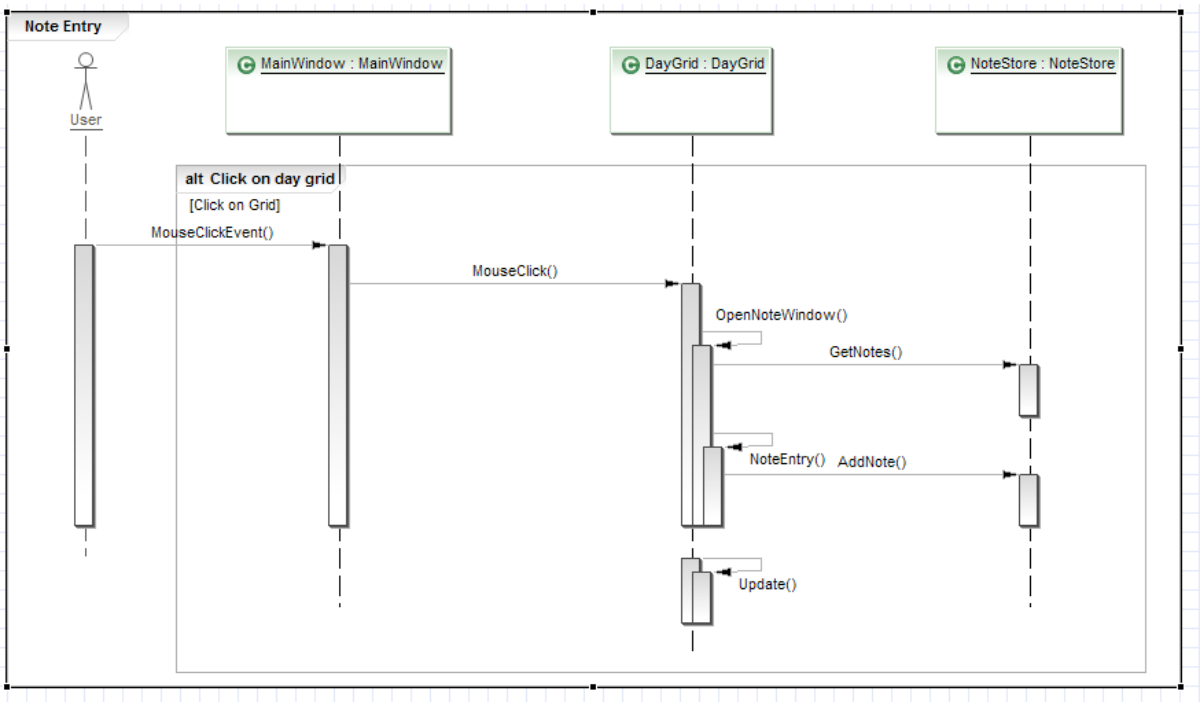
3.2. Month Update Sequence Diagram

Figure 3 Month Update Sequence Diagram



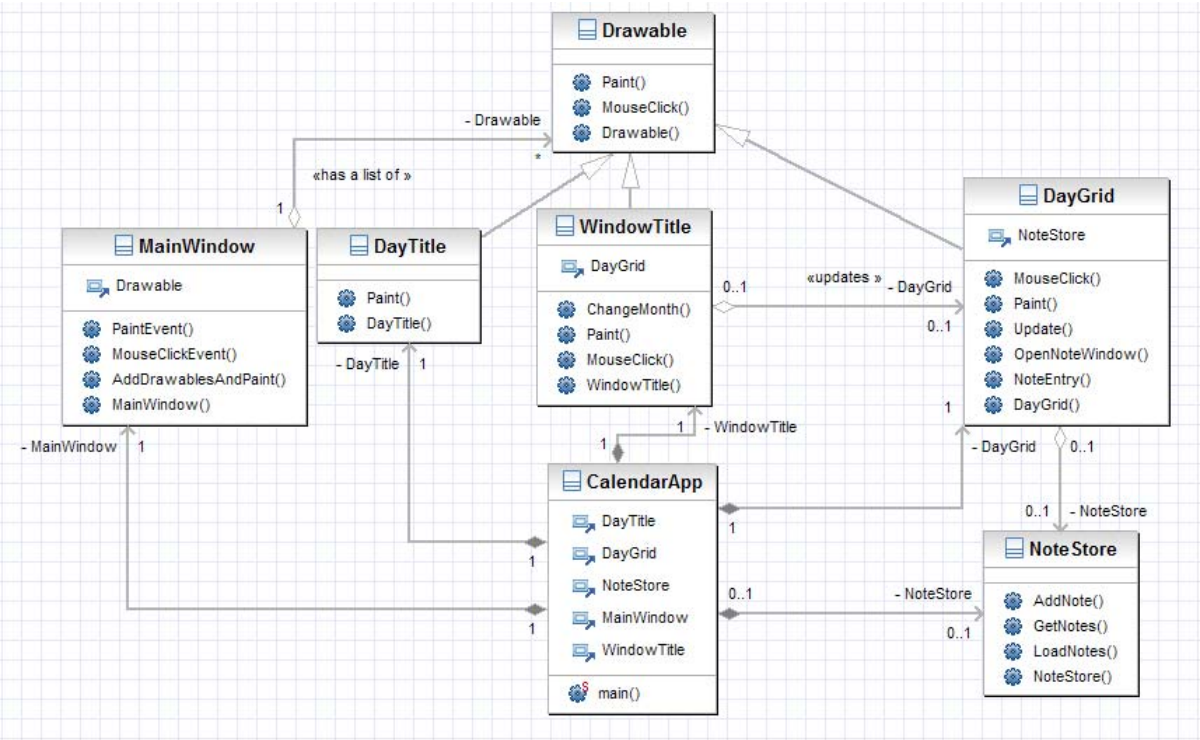
3.3. Note Entry/List Sequence Diagram

Figure 4 Note Entry/List Sequence Diagram



3.4. Calendar Application Class Diagram

Figure 5 Calendar Application Class Diagram



4. Methods and Behavior Tables

4.1. CalendarApp class Behavior Table

Table 4 CalendarApp class Behavior Table

public static void main(String [] args)	Creates the DayGrid, WindowTitle, DayTitle, NoteStore and MainWindow objects, informs the MainWindow about the Drawable objects and shows the MainWindow.

4.2. NoteStore Behavior Table

Table 5 NoteStore class Behavior Table

public void AddNote(Date date, String note)	Adds a note to the stored list of notes for a date, and appends it to the note file.
public String[] GetNotes(Date date)	Returns an array of notes stored for a date.
public void LoadNotes(String filename)	Loads the date-note pairs from the file 'filename'.
Public NoteStore()	Initializes NoteStore class members.

4.3. MainWindow Behavior Table

Table 6 MainWindow class Behavior Table

public void PaintEvent()	Repaints itself, and calls the paint methods of all of the stored 'Drawable' objects.
public void MouseClickEvent(int x, int y)	Checks the coordinates of its Drawable object list and calls the mouseClick method of the appropriate Drawable object.
public void AddDrawablesAndPaint(Drawable[] dList)	Stores the Drawable list and calls its PaintEvent method.
public MainWindow	Initializes MainWindow class members.

4.4. Drawable Behavior Table

Table 7 Drawable class Behavior Table

public void Paint ()	Paints the background.
public void MouseClick ()	-
public Drawable	Initializes Drawable class members.

4.5. DayTitle Behavior Table

Table 8 DayTitle class Behavior Table

public void Paint ()	Paints the day abbreviations from Sunday until Saturday.
public DayTitle	Initializes DayTitle class members.

4.6. WindowTitle Behavior Table

Table 9 WindowTitle class Behavior Table

public void Paint ()	Writes the current month/year and draws the left and right arrows.
public void ChangeMonth(bool increment)	If increment is true, sets the next month the current month and repaints itself. Otherwise, sets the previous month the current month and repaints itself.
public void MouseClick(int x, int y)	Checks if the right arrow or the left arrow are clicked. If the left arrow is clicked, highlights it and calls ChangeMonth with increment value 'false'. If the right arrow is clicked, highlights it and calls ChangeMonth with increment value 'true'.
public WindowTitle	Initializes WindowTitle class members.

4.7. DayGrid Behavior Table

Table 10 DayGrid class Behavior Table

public void Paint ()	Paints a 7 columns by 6 rows grid. Finds out which day the first of the selected month is and then writes the days of the month starting from the appropriate column (Column 1 is Sunday, 2 is Monday, etc). It also writes the dates before the first and after the last of the selected month on the day grid. The days of the selected month are highlighted. During painting, it also checks if a date contains stored notes. If so, writes that date in Bold font.
public void MouseClick(int x, int y)	Using the coordinates, determines the date and calls OpenNoteWindow using this date.
public void Update(Date date)	Using the month and year values of the date, sets the current month/date and repaints itself.
public void OpenNoteWindow(Date date)	Obtains a list of notes for the selected day and then calls NoteEntry to popup the note entry dialog box.
public void NoteEntry(String[] notes)	Calls JOptionPane.showMessageDialog using the appropriate inputs (A list of current notes, and an edit field to enter a new note) to show the note list/entry dialog box. If a note is entered on the dialog, stores it using the NoteStore.addNote method.
public DayGrid(NoteStore ns)	Initializes the DayGrid class members, and stores the NoteStore reference.

5. Algorithms for Data Calculation Methods

5.1. CalendarApp class Methods

5.1.1. Algorithm for method main

This method is called by the Java runtime after the user runs the application. Creates the instances of drawable objects and passes them to the MainWindow that it also creates. It also creates the NoteStore to be used by the DayGrid.

1. Create MainWindow
2. Create NoteStore and load notes
3. Create the WindowTitle, DayGrid and DayTitle objects
4. Provide these 'Drawable' objects to the MainWIndow, and paint the MainWindow..

5.2. NoteStore class Methods

5.2.1. Algorithm for method AddNote

This method appends a note to the list of notes for a specific date.

1. Obtain a reference to the list of notes for the date
2. Append the note to the list
3. Store it on the disk

5.2.2. Algorithm for method GetNotes

This method returns a list of notes for a specific date.

1. Obtain a reference to the list of notes for the date
2. Return it in a String array

5.2.3. Algorithm for method LoadNotes

This method returns a list of notes for a specific date.

1. Loads the note lists from a file.

5.3. MainWindow class Methods

5.3.1. Algorithm for method PaintEvent

This method repaints itself, and then calls the Paint methods of all of the stored Drawable objects.

1. Repaint itself
2. for every Drawable reference in its list
3. Call the Paint method of the Drawable.

5.3.2. Algorithm for method MouseClickEvent

This method figures out on which Drawable object the mouse was clicked. Then calls the MouseCLick method of the appropriate Drawable object.

1. for every Drawable reference in its list
2. Check if the Drawable object is hit by this click. If so, call it's MouseClick member

5.3.3. Algorithm for method AddDrawablesAndPaint

This method stores the list of Drawable objects and repaints itself during initialization.

1. Store the Drawable list.
2. Call 'PaintEvent'.

5.4. Drawable class Methods

5.4.1. Algorithm for method Paint

This method paints the background according to the highlight status.

1. Check if the component is highlighted using data members
2. Select the background color and paint the background.

5.5. DayTitle class Methods

5.5.1. Algorithm for method Paint

This method paints the day abbreviations from Sunday until Saturday.

1. Above the corresponding column positions of the DayGrid, write the abbreviations of the days: {"Su", "Mo", "Tu", "We", "Th", "Fr", "Sa"};

5.6. WindowTitle class Methods

5.6.1. Algorithm for method Paint

This method writes the current month/year and draws the left and right arrows.

1. Get the stored month and year values from the data members and paint them on the bar
2. Paint the left and right arrows on the bar.

5.6.2. Algorithm for method ChangeMonth

This method updates the current month and repaints the title bar.

1. Using the Calendar class's 'add' method, increment or decrement the current month according to the provided 'increment' Boolean value.
2. Call Paint to repaint itself.

5.6.3. Algorithm for method MouseClick

This method highlights the arrows if clicked on. If either of the arrows are clicked, ChangeMonth method is called with the appropriate direction.

1. Using the x and y coordinates of the mouse click, check if the left arrow or the right arrow is clicked on.
2. Highlight the arrow during mouse pressed period
3. If the left arrow is clicked, call 'ChangeMonth' with parameter increment value set to false.
4. If the right arrow is clicked, call 'ChangeMonth' with parameter increment value set to true.

5.7. DayGrid class Methods

5.7.1. Algorithm for method Paint

This method paints the 7x6 day grid.

1. Find out which day of the week is the 1st of the selected month
2. Go back day by day until the current day is 'Sunday'. Set this date the current date of the day grid.

3. For every 6 row in the grid
4. For every 7 columns in the grid
5. If the current date is inside the current month highlight it.
6. If the current date has at least one note stored, set the font to bold, else regular.
7. Paint the current date
8. Increment the current date.

5.7.2. Algorithm for method MouseClick

Using the coordinates, this method determines the date and calls OpenNoteWindow using this date.

1. Find out which grid cell is clicked on
2. Find out which date that cell corresponds to.
3. Call OpenNoteWIndow for this date

5.7.3. Algorithm for method Update

Using the month and year values of the provided date, this method sets the current month/date and repaints itself.

1. Store the date
2. Call “Paint”

5.7.4. Algorithm for method OpenNoteWindow

This method obtains a list of notes for the selected day and then calls NoteEntry to popup the note entry dialog box..

1. Retrieve the list of notes for the current date from the NoteStore class’s ‘GetNote’ method.
2. Call ‘NoteEntry’ using the current date and this note list.

5.7.5. Algorithm for method NoteEntry

This method pops up the note entry dialog and returns the entered note.

1. Create a listbox filled with the notes
2. Create a TextBox for entering new notes.
3. Create a JComponent list using these list and text boxes, together with their labels
4. Provide this list to JOptionPane.showMessageDialog method to create the note entry dialog.
5. Upon return, check if the returned text is empty. If not, append it to the notes of the day using the NoteStore method ‘addNote’.