IS3 – IMPLEMENTATION AND EVALUATION REPORT

Mamas Nicolaou, Hector Grebbell
Henri Pelkonen, Dmitrijs Jonins
Dan Tomosoiu, Nicole Kearns

FUNCTIONALITY

The main features we implemented in our calendar are:

Add appointments – To add a new event to the calendar, there is simply an 'Add event' button. When the user clicks this, a new window is displayed with fields for all details stored within an event, i.e. Time, date, location, etc. Once the user has entered the details, simply click save and the event will appear in the calendar. Should they decide to cancel, exiting the window returns without saving.

Delete appointments – To delete an event the user may either click the remove button within the event details window, or click on the event from the day or upcoming event view to select it and push the delete key. A yes/no confirmation window will then appear, to avoid deleting events by accident.

Edit an appointment – To edit an appointment, the user will double click on the event from either the day or upcoming event view. This opens the event details panel. All fields within this video can be edited. Click 'Save' to apply the changes. For example, to move it to another day, the user will simply change the date. To close without saving changes, exit the window.

Set a recurring appointment – When adding or editing an event the user has the option to select recurrence. An event can be repeated Daily, Weekly, Two Weekly, Four Weekly or Yearly. The event will be shown on each recurrence in every vie except Upcoming events. Here only the first shall be listed.

Change views – Along the bottom left of the calendar, there are 6 buttons, one for each of the views (Day, Week, Month, Year, Upcoming Events and Busyness). When the appropriate button is clicked, the system will navigate to the referenced panel. For example, if the 'Month' button is clicked, the calendar will display the month view, with all events for that month.

Find the busiest and quietest days in the month
Within month view, the number of events for each day is displayed. To empathize this days are color coded. The user can set 4 levels of “busyness” within the settings panel and select a color for each.
This allows the user to see how many events they have on each day, and clearly see how busy/quiet their day is.
An overview of “busyness” throughout the year can be seen from the busyness view. This shows a line graph of how busy the user is.

Set different categories
When adding or editing an event the user has the option to select a category for an event or set this to “None”. This can later help the user filter by category and see all events that have been categorized under the selected label. Available categories are Work, University, Social and Birthdays.

Filter by Category
There is a button for each category, located at the top of our interface. Clicking on a category button will cause the system to show only appointments of the selected category. The default filtering option is “All Categories”, for showing all events without any filtering.

SIMILARITIES OF PAPER PROTOTYPE & JAVA PROTOTYPE

All main features of the Paper Prototype have been implemented in the Java version. There are a few minor structural changes within the Java prototype, such as having current date visible in all views.

Another small change is there are no colors associated with each category. As a result when ‘All categories’ is active, the user is unable to distinguish which events are in each without using the filtering function. This was done since color-coding was implemented to represent “busyness” instead.

In month view, there is a number to indicate how many events a user has planned for the day. The entire day box is colored according to the number of events instead of having a number of small boxes representing each event.

The “To-Do” list proposed on the paper prototype was implemented as “Upcoming Events” to avoid confusion.

A busyness visualization has been added to the Java prototype. This allows the user to see a bar chart of how busy he or she is in any given period according to the amount of time used up in activities in the calendar.

SPECIFIC PROBLEMS

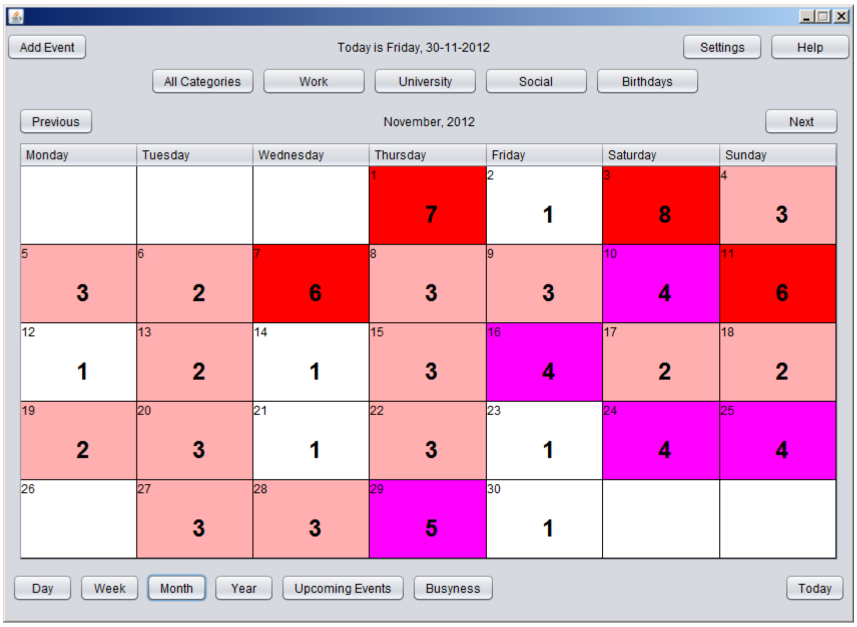
When implementing the calendar, we encountered the following problems:

The user is unable to add new categories. Initially, we implemented 4 categories to show the functionality. However, the user cannot delete the current categories or add new categories.
All events will last 1 hour. When adding a new event, the user has no option to select duration of an event or when it will end.

In week view, the calendar is unable to show more than 1 event at the same time. For events starting at the same time, it will select one of the events and display only this event for the given time. For events starting within the same hour, it will display the event which occurs first for that hour.

Recurrences are infinite. When adding a recurrence to an event, you can select how often the event has to occur, however there is no way to indicate when this will stop. When deleting a recurring event, all instances of this event are removed. There is no way to remove a singular date.

When a user adds a new event with a nonexistent date selected, the system will add the month to the maximum day number of the selected month plus any remaining days added to the following month. For example, if a user adds an event with the selected date being 31/11/2012 the event will be created for the 1/12/2012 automatically.

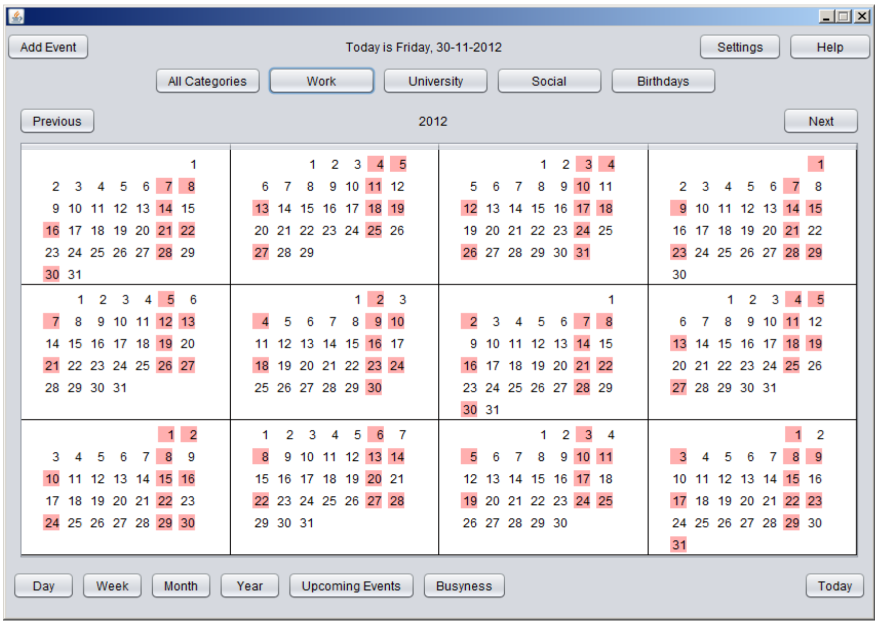


Today is Friday, 30-11-2012

November, 2012

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
			1 7	2 1	3 8	4 3
5 3	6 2	7 6	8 3	9 3	10 4	11 6
12 1	13 2	14 1	15 3	16 4	17 2	18 2
19 2	20 3	21 1	22 3	23 1	24 4	25 4
26	27 3	28 3	29 5	30 1		

Figure 1: The month view showing all the events in November 2012



Today is Friday, 30-11-2012

2012

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Figure 2: The year view showing only work events

Figure 2 shows how we actually implemented the level of busyness in the year view showing only work events occurring in the current year. It is possibly to toggle between different categories in all views.

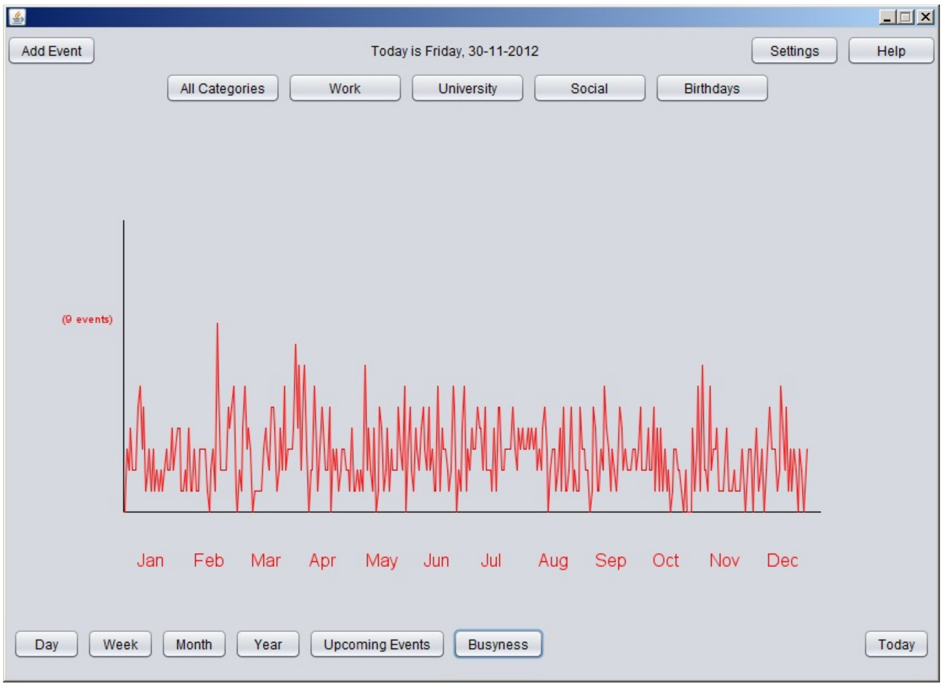


Figure 3: Level of busyness in the year

Figure 3 demonstrates the biggest change from the paper prototype. This is a completely new feature to the calendar. When Busyness button is pressed this graph pops out on the screen. It takes too much time to do a different views to allow a user to choose busyness in other categories.

USABILITY EXPERIMENT

Experimental Evaluation is good for determining a causal relationship between interface characteristic and user performance behaviour. We decided to use the Within Subjects design since it requires half the users between-users design requires (McBurney et.al, 2009:313)and it also reduces any error variance associated with individual differences which can become a confounding variable(Gravetter & Forzano, 2011:274).
We have defined 5 hypotheses which will prove whether the improvements we have implemented on our own calendar have indeed increased the efficiency of the calendar interface and thus, decreased the amount of time it takes each user to complete each task.
7 will take part in our experiment as volunteer testers. We made sure that the volunteers are all novice users with our system and the iCalendar (iPhone Calendar) interface.

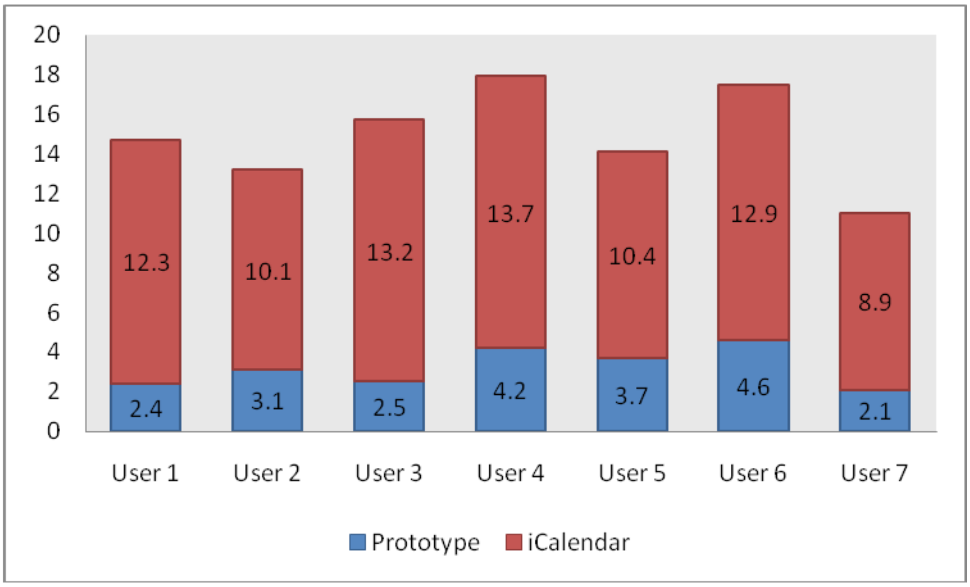
Hypotheses:

1. Having the “Today” button located in the bottom-right corner of the window will help the users locate it easier and thus, make the frequent task of navigating to today’s page faster

   Independent Variable: Location of “Today Button”
   Dependent Variable: Time it takes for a user to go to today.

   Condition 1: Users going to today on an iCalendar (iPhone, button located in the bottom,
   Condition 2: Users going to today on the prototype Calendar

   Results in seconds:
   Average time for Prototype users: 18.4 seconds
   Average time for iCalendar users: 23.3 seconds.

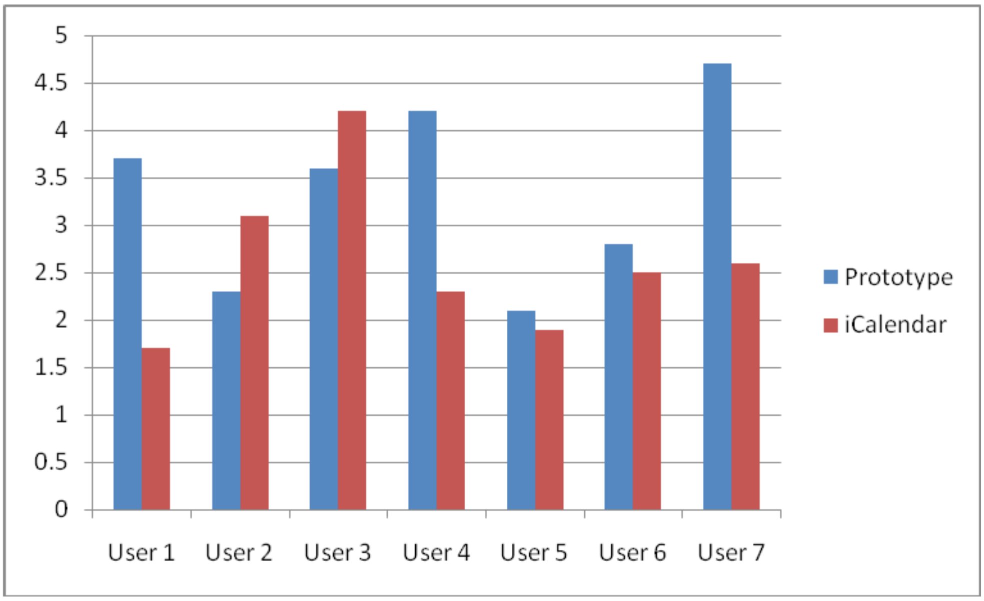


2. Having a clearly marked “Add Event” button will help the user recognize it faster even if he/she is not familiar with the application.

   Independent Variable: “Add Event” label name
   Dependent Variable: Time it takes for a user to recognize it.

Condition 1: Use a “+” button on iCalendar.
Condition 2: Use an “Add Event” button on our Prototype.

Results in seconds:
Average time for Prototype users: 3.3 seconds
Average time for iCalendar users: 2.6 seconds.



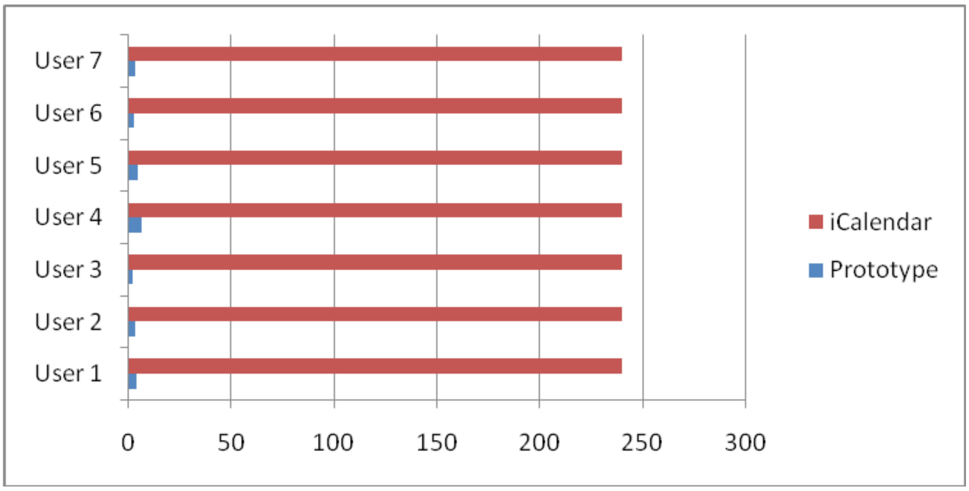
3. Having the week view button clearly marked next to “Day”, “Month” and “Year” will help the user switch to Week view faster.

Independent Variable: The way of navigating to weekly view.
Dependent Variable: Time it takes a user to switch to weekly view.

Condition 1: Rotate Screen on iPhone to reveal Week view..
Condition 2: Have weekly view button displayed in the bottom left group of buttons.

Results in seconds:
Note: All of the iCalendar testers failed to find out how to navigate to the weekly view or thought that they are is no such screen on the phone.

Average time for a user to find out how to switch to week view:
Prototype users: 3.8 seconds
iCalendar users: 240+ seconds.



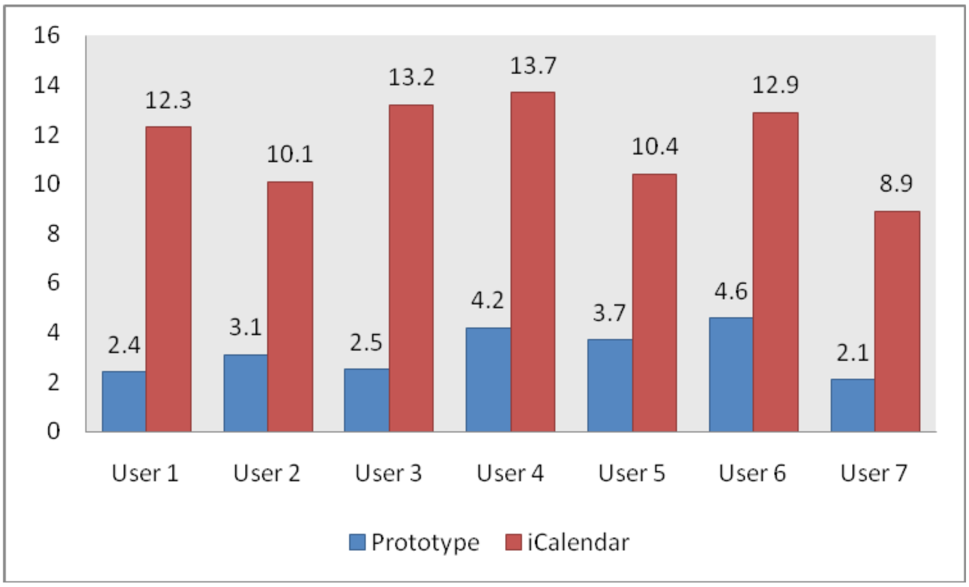
4. Having the Categories visible on the top of the screen will help the user locate it and filter by category faster.

Independent Variable: The way of navigating to weekly view.
Dependent Variable: Time it takes a user to switch to weekly view.

Condition 1: Have a drop down list called "Calendars" (iCalendar)
Condition 2: Have all filters visible as buttons on the top of the screen (Prototype)

Results in seconds:
Average time for a user to find out how to only display social events (filter by "social" category):

Prototype users: 2.7 seconds
iCalendar users: 10.6 seconds.

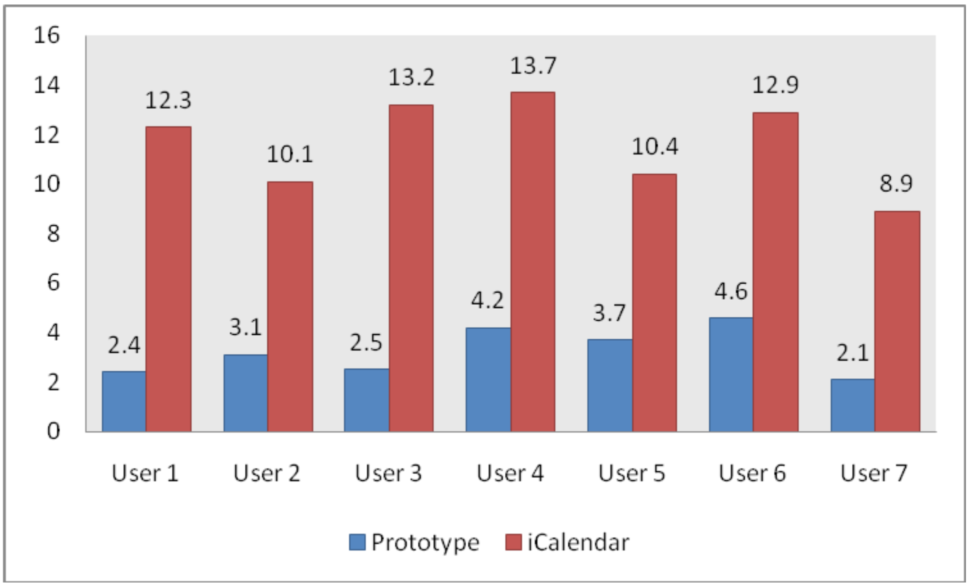


5. In month view showing the number of events instead of showing one dot per event will help the user understand which days are the busiest faster.

Independent Variable: The way of navigating to weekly view.
Dependent Variable: Time it takes a user to switch to weekly view.

Condition 1: Showing one dot per event (iCalendar)
Condition 2: Showing the total number of events as an integer below the day number.

Results in seconds:
Average time it took for a user to figure out how many events there are on 2 days with 6 events each.
Prototype users: 3.22 seconds
iCalendar users: 11.64 seconds.



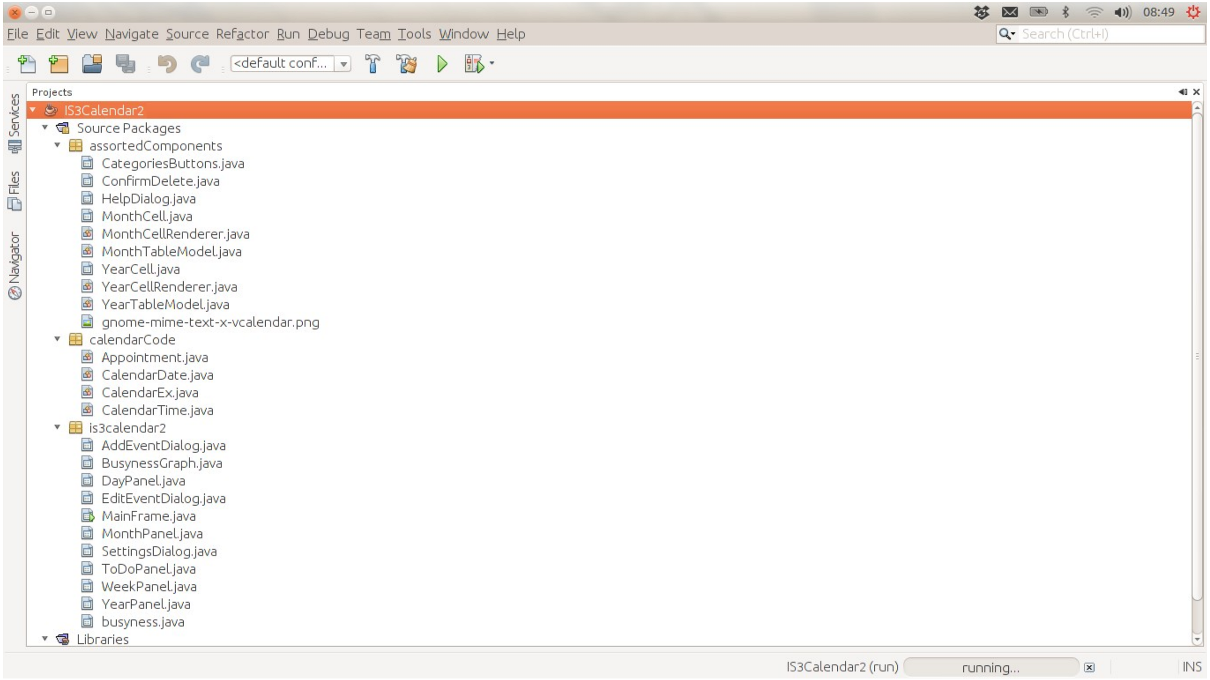
6.

CITATIONS:

1. Frederick J. Gravetter &Lori-Ann B. Forzano, Research Methods for the Behavioral Sciences, Cengage Learning, Belmont, USA, 2011
2. Donald H. McBurney, Theresa L. White, Donald H. McBurney, Research Methods, Wadsworth Cengage Learning, Belmont, USA, 2009

Appendix

File tree



```
IS3Calendar2
  Source Packages
    assortedComponents
      CategoriesButtons.java
      ConfirmDelete.java
      HelpDialog.java
      MonthCell.java
      MonthCellRenderer.java
      MonthTableModel.java
      YearCell.java
      YearCellRenderer.java
      YearTableModel.java
      gnome-mime-text-x-vcalendar.png
    calendarCode
      Appointment.java
      CalendarDate.java
      CalendarEx.java
      CalendarTime.java
    is3calendar2
      AddEventDialog.java
      BusynessGraph.java
      DayPanel.java
      EditEventDialog.java
      MainFrame.java
      MonthPanel.java
      SettingsDialog.java
      ToDoPanel.java
      WeekPanel.java
      YearPanel.java
      busyness.java
  Libraries
```