**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 all the details necessary for an event, ie. Time, date, location, etc. Once the user has entered the details, simply click save and the event will appear in the calendar.  
  
  
**Delete appointments** – To delete an event, double click on the event in the calendar and it will bring up the details. There will be an option within this window to delete this event. When deleting, the user will be asked to confirm if they want to delete the event, to avoid deleting events by accident.  
The user will also be able to delete the event directly from the calendar by clicking the event which they want to delete and simply pressing the delete button on their keyboard. Again, the user will have to confirm that they want to delete this event.  
  
  
**Edit an appointment to move it to a free slot in another day –** To edit an appointment, the user will double click on the event in the calendar which will bring up the event details. Within this window, there is an option to edit. Once the user has clicked 'Edit', they will be able to change any details for the event and click 'Save' to apply the changes. For example, to move it to another day, the user will simply change the date.  
  
  
**Set a recurring appointment to occur once a week and finish in three months time -**  
  
  
**Change views from month, to week, to day –** Along the bottom of the calendar, there are 5 buttons, one for each of the views (day, week, month, year, to-do). When one of the buttons is clicked, the calendar will then display the appropriate view. For example, if the 'Month' button is clicked, the calendar will then display the month view, with all events for that month.  
  
  
**Find the busiest and quietest days in the month –** In month view, the number of categories for the day is displayed. This allows the user to see how many events they have on each day, and clearly see how busy/quiet their day is.  
  
  
**Set different categories for university, social and job appointments -**  
  
  
**View just one category of appointments –** We created buttons for each category created, which appear at the top of our interface. When a specific category button is selected, only events in that category will be displayed on the calendar. We have a 'All Categories' button, which allows us to go back to a view showing all events.

**SIMILARITIES OF PAPER PROTOTYPE & JAVA PROTOTYPE**  
  
All main features of the paper prototype have been implemented in the java prototype. There are a few changes within the java prototype, such as adding the current day/date to the calendar so that it is visible in all views. There are no colours associated with the categories, so in the ‘All categories’ view, the user is unable to see which events are for which category.   
In month view, instead of coloured boxes to show the events that day, there is just a number to indicate how many events they have planned that day.  
  
  
  
**SPECIFIC PROBLEMS**

* Cannot add new Categories - We set up a few starter categories to show how they work. However, there is no way for a user to add a new category.
* No way to create an event with a duration less than or over an hour - When adding/editing an event, there is no duration for the event, so all events will be set to 1 hour long.
* Cannot add Recurrences //TEST
* Cannot set reminders for any events -
* Calendar code doesn’t work for leap year
* Can’t have more than one event at a time
* the help button does nothing!
* the settings button is not implemented

We have defined NUMBER\_OF\_HYPOTHESES  
  
**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 to test:**

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.

1. 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.

1. 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.

1. 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.

1. In month view showing the number of events instead of showing one dot per event will help the user understand

**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.

**CITATIONS:**

1. Research Methods for the Behavioral Sciences, Frederick J. Gravetter &Lori-Ann B. Forzano, 2011, Cengage Learning BLA BLA need to add info
2. Research Methods, Donald H. McBurney, Theresa L. White, Donald H. McBurney 2009