CS 422 - AS3

A/B Testing Analysis Report

By Rohan Tadphale (UIN 657662650)

Problem

Calendar Event Interface - The Interface should let user create various events such as meetings, birthdays etc throughout the year. The user should be able to create recurring events along with the once happening events. User should be able to set specific timing of the said event. User should also be able to set end repeat for the repeated events. The interface should give user the flexibility to create customizable repeat events.

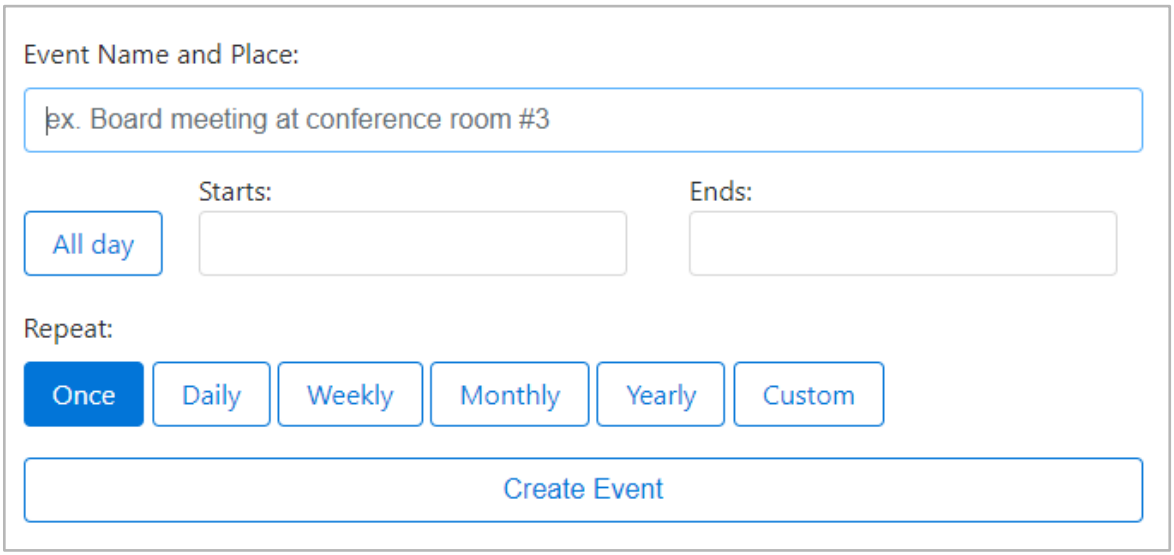
The two interfaces will collect data of the users such as user ID, target information, time. From this we can find out the approximate efficiency of the models using aspects such as tasks completed per unit time.

Collaborators

No collaborators

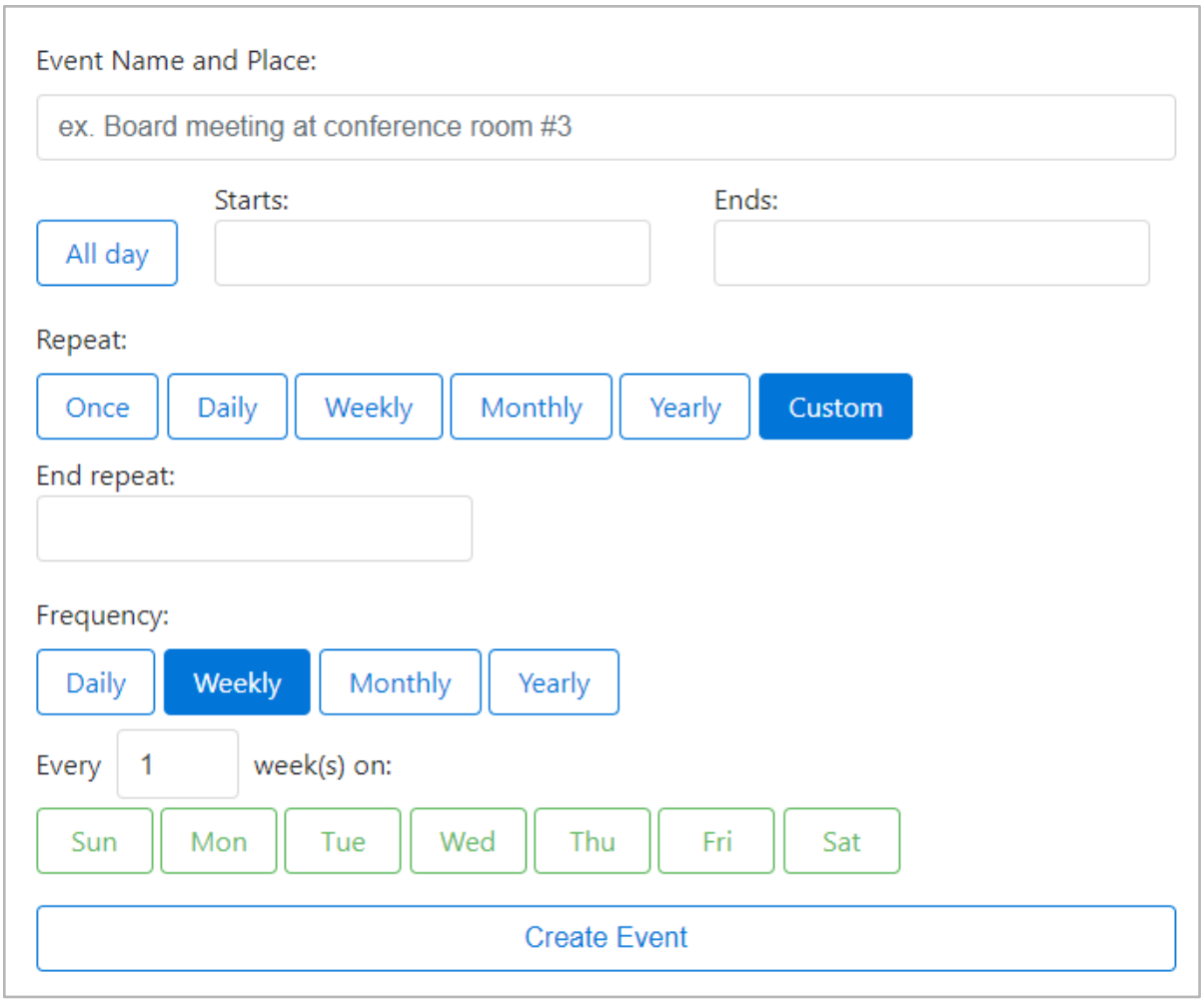
Illustration

[Calendar interface B]



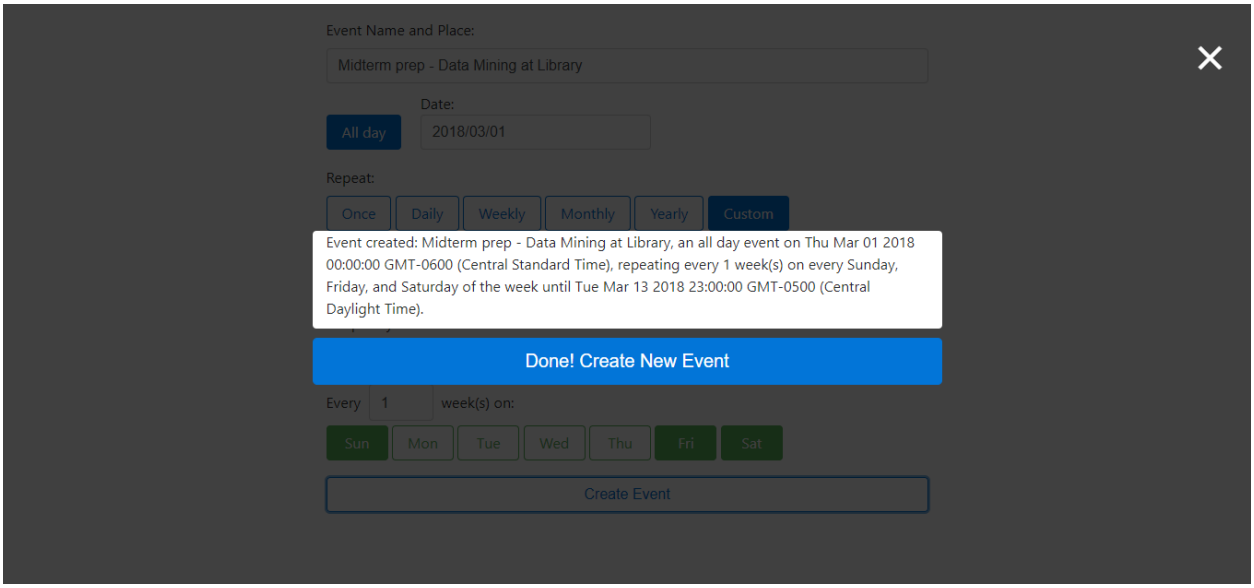
Initial page

Aesthetic and Minimalist design - I have used the form design as the user is already familiar with the elements used in the design. The aesthetic and minimalist design matches real world. (The monthly option in the customized section is as per the design of a physical calendar.)



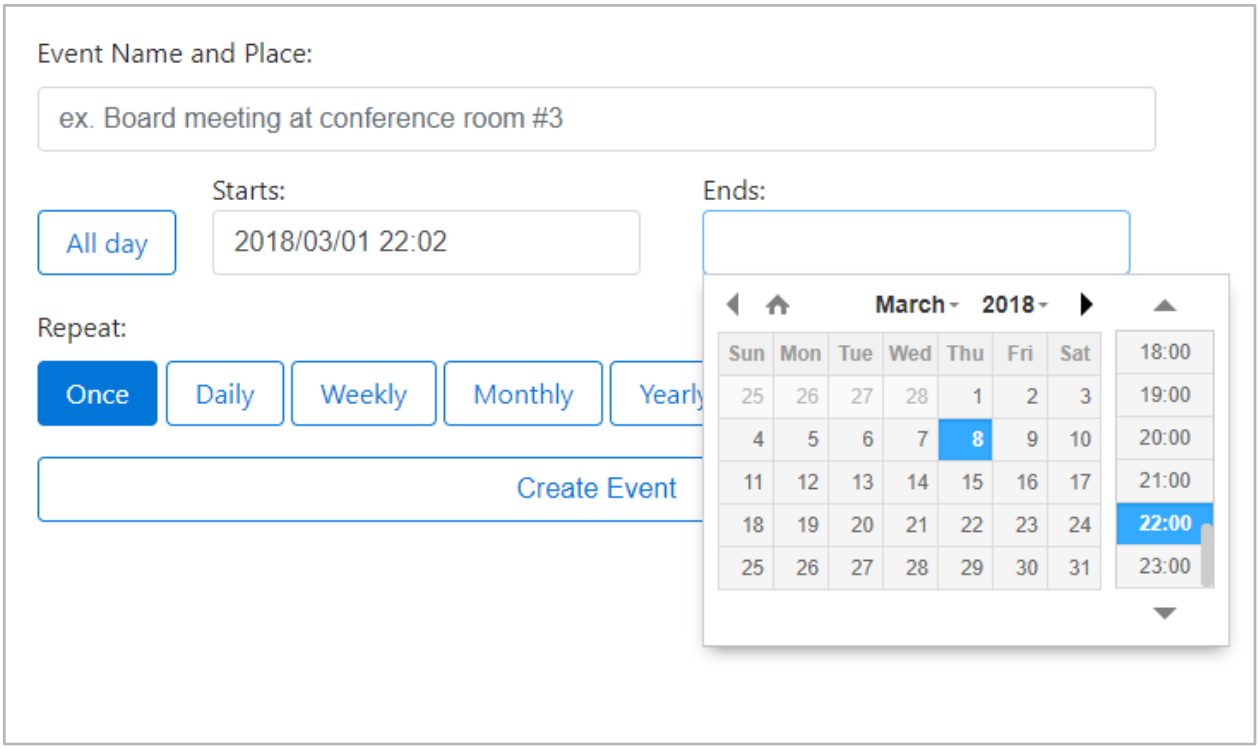
Click on the 'Custom' button in 'Repeat' section and then 'weekly' in 'Frequency' section to see this.

Visibility and efficiency: Provided the large digital real estate, just the right amount of information is hidden from the user. For example, instead of using a drop down menu, a toggle button group is used. The buttons used for checkboxes are large, the placement is appropriately close. This makes it efficient for user to complete the task. (Fitts's law)



When clicked on the 'create event'

Visibility of system status - Important content is isolated from the user to underline the importance of it.

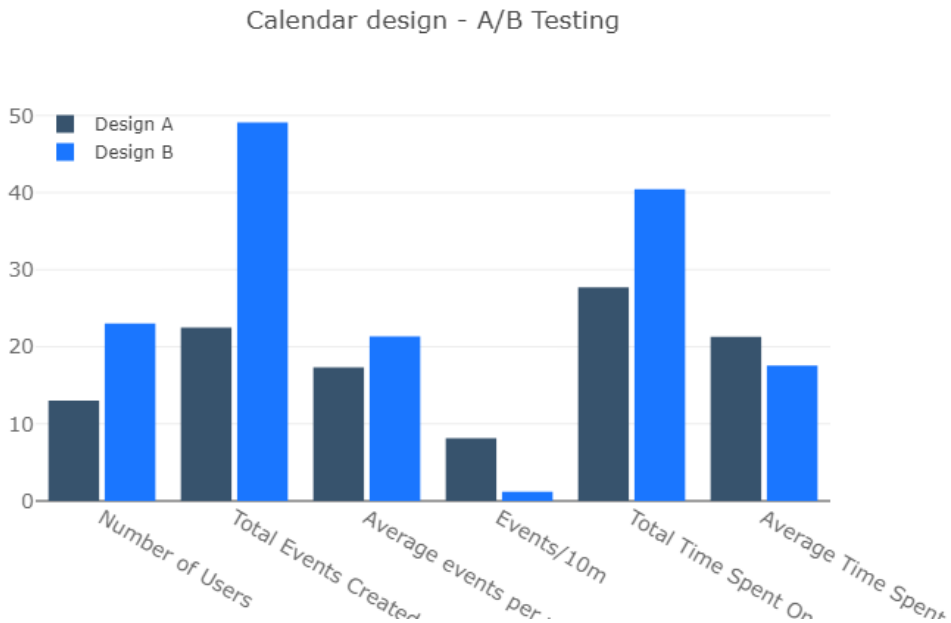


For **error prevention (safety),** The dates have been disabled. Example of the content to be written in the target area has been used as ghost text (placeholder).

Experimental hypothesis

I believe, version B will be more efficient than version A. The buttons used in the design have larger target size. The typeface and the colors used will help improving the visibility as the interface is much clearer and clutter free.
The design employs number of elements to avoid slips and other mode errors. (e.g. confirmation box, inability to select previous dates in the calendar etc.) This efficiency can be measured by average time taken to create an event.

Datasets Overview



Dataset Overview Graph

	Number of users	Total events created (tasks completed)	Total time spent (min)	Success rate (%)
Dataset A	13	225	404	-
Dataset B	23	491	226	98

The two datasets were collected as .csv files. I created a python script to analyze both datasets and plot graphs using plot.ly module (open source).

Results

The two datasets logged were dataset A and dataset B, for design interface A and design interface B respectively. Some of these users did not create any events. Some users kept the interface link tabs open for some days, that has affected the analysis quite a bit.

As per the success criterion added in the B design of the calendar interface (no errors detected when clicked on the event creation), 98% users were able to create events without any errors.

Events created per 10 minutes for design A were close to that of design B. But with correction for the days people left the interface tab open, the difference is about 3 events. With design B, users were able to create 3 more tasks per 10 mins than users using design A.

Users also created more events on an average with design B than users using design A.

Even though users collectively created more events and spent more time on interface B, on an average more time was spent on design A per user.

From this graph we can build an educated guess that calendar interface design B is more efficient than design A, although more controlled user tests would yield much clearer results that can state so strongly.