British Columbia Institute of Technology

Computer System Technology

Human Computer Interface

COMP 8521: Selected Topics in Advanced Interface Design

Final Project

Student: Leo Alekseev

2012

**Introduction**

The interface to analyze, create, and test is a prototype of a web

time-management application.

The idea for the first stage – Content, Architecture, and Analysis Study Method – is to:

- select a design brief (from the common ones)

- choose an activity that relates to the selected brief

- select at least three individuals and observe them

performing the chosen activity

- take notes during the observation (those are going to be

findings/possible design opportunities)

- brainstorm a list of specific user needs based on the

findings

- find inspirations

- create a point of view that relates to the selected design

brief and chosen activity

- come up with two design ideas that address/engage the

created point of view; illustrate each of these ideas with a

storyboard

- make two prototypes in Balsamiq Mockups application that

implement the stated two ideas

- pretest these two prototypes according to the Heuristic

Evaluation (HE) list

- select one of the two previously stated ideas; the selected

one is going to be implemented

- compose a list of changes for the selected idea based on the

HE results

- develop content inventory

- create site map.

The list of user needs, storyboards, and the list of changes after pretesting are the source of the requirements for the prototype to be developed.

**Analysis Study Method**

**Design Brief**

There are three (3) common/main design briefs; please, see the following:

1) Change

This design brief might be described as follows:

“Change is hard. Sometimes we lack information. Other times, our routines and habits are really persistent, even if we wish they weren't. Can technology help people and communities change their behavior to meet their goals? New electronic devices (computers, phones, tablets, etc.) can help by providing information, by reminding us, and by connecting us with others. Change might mean exercising more, eating healthier, helping make a more sustainable planet, or participating in local government. Also, it might be becoming a better chess player, carving out time to read, or remembering to see the world from a new perspective”.

The idea is to create an application or service that facilitates personal or social behavior change.

2) Glance

This design brief might be described as follows:

“We are surrounded by information. Some might even call it overload. How might technology show us the essential pieces at a glance, so we can quickly navigate through the noise to get to what we really want? We compulsively check email, Twitter, Facebook, and the news — just in case there's something there. Right now we are doing the filtering and finding ourselves, why not let our devices do it for us? How can a screen summarize information and present just the most relevant parts (especially if it is tiny)? How can these devices use social and physical context to more effectively have the key information ready at a glance? Today the home screen of many devices is a grid of icons, or a static picture. That's not very creative. We can do way better!”

The idea is to find people and design a personal dashboard tailored to their needs.

3) Time

This design brief might be described as follows:

“The way people represent time changes how they think about it. Wall calendars remind us of years, seasons, and the dentist appointment 6 months in the future. They codify weeks by wrapping every seven days, and it's easy to find the weekends -- they are on the edges. Clocks help us coordinate with others. Historically, many countries' citizens adopted pocket watches and clocks along with the railroad. Before the railroad, there was no need for precise time. Daily schedules help us plan. They can encourage us to "fill" our days, or talk about being "free". When we punch the clock, or bill hours, we turn time into money. These representations are human inventions. Most digital time representations — clocks, daily and monthly calendars — simply translate paper and gears into pixels and beeps. With the computation and sensing capabilities of mobile devices, can we find a more personal and joyful way to interact with time?”

The idea is to redesign the way we experience or interact with time.

**Activity and Observation**

My observation falls under the "Time" design brief.

The activity I observed is the process of scheduling people to work according to their availability and day off requests.

For a period of time I was managing a coffee shop. One of my duties was to schedule people to work particular shifts. There were approximately 13 employees. They send me their day off requests and availability one week before the current schedule ends. Then, I make a new 3-week schedule based on employees' requests. The whole process was related to the time manipulation procedures such as:

- creating shifts (time intervals) according to the coffee shop needs

- assigning people to particular shifts based on their availability and

proficiency

- keep track of each employee working hours to avoid/maintain overtime

work

- creating day-to-day breaks list for each employee

- managing coffee/food/equipment delivery time schedule based on

employees' work load.

- designing time line for the schedule

- trying to "visualize time" in order to combine and concern everyone's

day off requests and availability.

I observed three (3) managers creating a schedule for their employees according to their day off requests and availability.

All three managers - Wayne, Giles, and Asha - take similar steps:

- use Excel to create schedules

- ask employees for their day off requests and availability approximately

one week (depending on the length of the new schedule) before the new

schedule

- write down employees' requests and try to create a sequential line of

shifts (basically all three (3) of them think about the time as line of

Excel cells)

- in case of requests conflict, notify employees and try to work that out

between them. If nothing happens, they ask all employees. If this

doesn't help, they take this shift (those shifts)

- use historical data and employees capability (stamina) to define the

amount of shifts for each particular time interval.

- when the first draft is ready, the schedule is published for employees'

revision. If there are any mistakes, the schedule is "rescheduled", and,

then, published again for the revision. When all shifts and day off

requests are worked out, the schedule is officially published.

Wayne and Giles don’t leave empty shifts. They like to have a solid plan for the week. If they cannot find a person to work, they schedule themselves. In contrast, Asha lives empty shifts in case someone shows up later and takes the shift. If nobody can work, Asha works herself.

**Giles** (see figure 1 for Giles’ observation notes) schedules all employees evenly. Every employee gets a piece of the "bad" shift. Also, he assigns core shifts to people who are good at work and who likes working those shifts. Giles has a list of tasks. Each shift is responsible for certain combination of tasks (i.e. team/shift work). Giles likes 1-week schedule.

**Asha** (see figure 2 for Asha’s observation notes) has a list of employees. As closer an employee to the top of the list as more priorities in the schedule creation procedure that employee has. People who work longer go to the top of this list, while new people begin at the bottom. Next, Asha tries to balance everyone’s day offs and hard shifts according to their previous requests and schedule. She has a list of tasks. Each employee has a task assigned to him/her once a week. Asha likes a 3-week schedule.

**Wayne** (see figure 3 for Wayne’s observation notes) has a list of core (main) employees. They always work the defined shifts and the rest of the schedule is based on that. Sometimes employees trade their shifts by themselves; for example, if someone gets sick, that person looks for other employee who can take the shift, so Wayne doesn’t need o worry bout shift reassignment. Then, there is a list of tasks and each particular shift is responsible for them (the same way as Giles does). Wayne is fine to move shifts by several hours (early/later) if someone cannot start on time. Wayne prefers a 3-week schedule. He uses Facebook to gather employees day off requests and availability. Also, he likes SMS as a communication option with his employees.

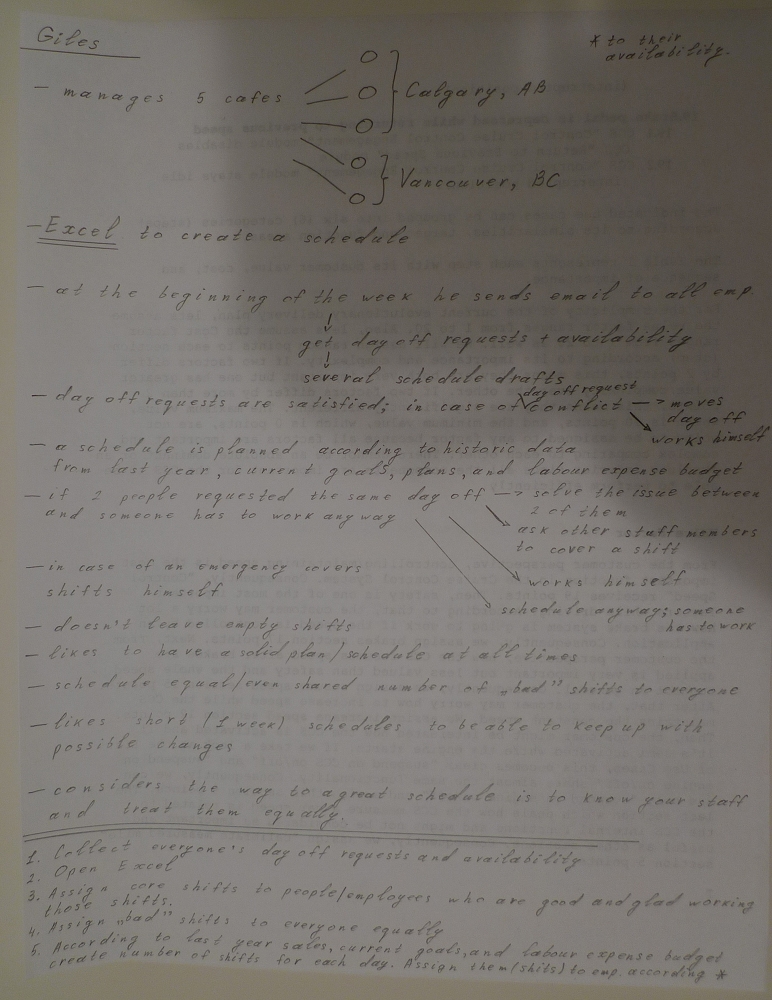


Figure 1. Giles’ observation notes

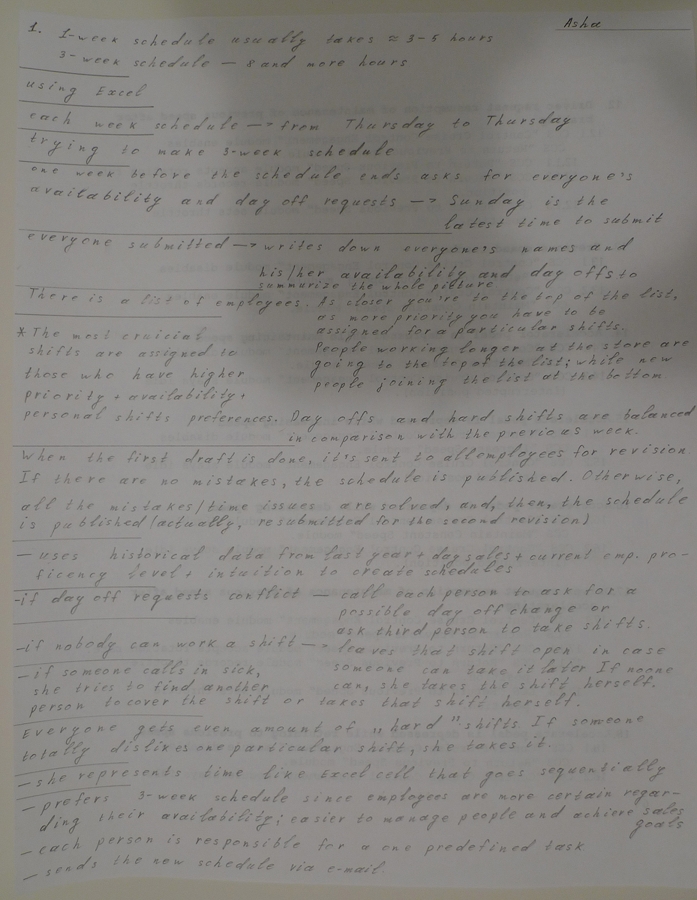


Figure 2. Asha’s observation notes

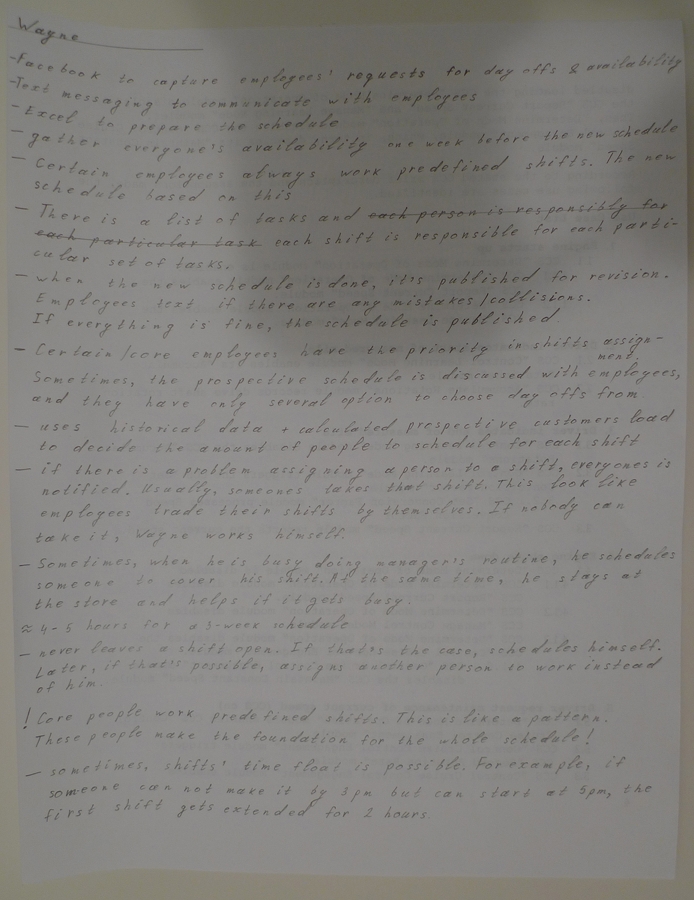


Figure 3. Wayne’s observation notes

**List of user needs**

The following is the list of user needs based on the Giles, Asha, and Wayne’s observations creating a schedule for their employees.

1. Represent time line effectively so each person can get a grasp of the

schedule in 15 seconds.

2. Distribute the amount of hard/bad shifts evenly among all employees.

3. Establish a solid back up plan to cover someone's shift in case that

person gets sick.

4. Let employees communicate with each other seamlessly regarding day off

requests and availability prospective conflicts.

5. Let employees to be able to trade shifts without third party

interaction but under the manager's supervision.

6. Let employees to submit their day off requests and availability to the

centralized application and allow them to edit those requests in real

time (like Google Docs).

7. Represent a shift task assignment effectively.

8. Create a schedule patterns applicable for different times of the year.

9. Automatically notify employees about all schedule changes and coming

shifts; for example, send SMS telling something like "you shift starts

in 1 hour".

10. Automatically notify employees about all work place

events/updates/news (probably integrate a Twitter line).

11. Create a schedule wizard that automatically creates a prototype

schedule according to the entered employee's day off requests and

availability.

12. Assemble an employees' day off requests tracking system.

13. Create a shift tasks tracking system that rotates tasks assigned to

each person automatically based on tasks difficulty and time required.

14. To make the whole system to be able to interconnect/communicate with

the social media applications (Facebook, VK, Twitter).

15. Notify employees about day off requests and availability submission

dead line.

**Inspirations**

The following are inspirations I chose for the prototype I developed.

1. Dry Erase Clock

(<http://gizmodo.com/5243230/dry-erase-wall-clock-helps->

keep-tabs-on-your-appointments)

(http://deliacreates.blogspot.ca/2011/11/dry-erase-clock.html)

The clock is really simple, uncluttered, and easy to understand.

This clock provides great, intuitive visual interaction with tasks you

scheduled at a particular time. Also, you can see all your plans at

once.

2. Cycle Life Watch (<http://www.coroflot.com/K4Y/Watch-Cycle-of-life>)

This is a black humor concept designed for an average IT working

person. What I found interesting is the representation of a day phase

along  with digital time. Also, if someone wants to

change/improve/customize the idea, that person can add more phases

representing his/her day. This is great visual representation of a

particular part of a day.

3. Muji Chronotebook

(<http://blog.jackcheng.com/post/2521777514/muji-chronotebook>)

The pages are crystal clear (well, except the clock in the center).

This doesn't distract you from what you originally were planning to

plan or simply schedule. The possibilities are actually endless and the

notebook is open for everything. I guess this is simply clever!

4. Shift Planning (http://www.shiftplanning.com/schedule/)

This looks like really comprehensive application that has auto schedule

creation wizard, employees' collaborative schedule approach, and

multiple presentation views. The idea that employees can assign

preferred shifts, trade and pick up shifts is really smart!

5. UNO 24 watch by Botta-design

(http://www.bottadesign.de/en/einzeigeruhr-uno-24.html#)

The watch hand position represents/corresponds to the sun position in

the sky. This really helps to visualize your day.

**Point of View**

My point of view:

"Concerning employees' day off requests and availability shouldn't take you 3+ hours to do a 1-week schedule draft. Applying concise and visually self-descriptive time representation technique/method/application will reduce doing that draft to about half an hour".

Later, the draft might be used to make a schedule in any other application a user feels comfortable with; for example, in Excel.

**Storyboards**

The following storyboard shows how an application can effectively consolidate and process employees’ day offs and availability.

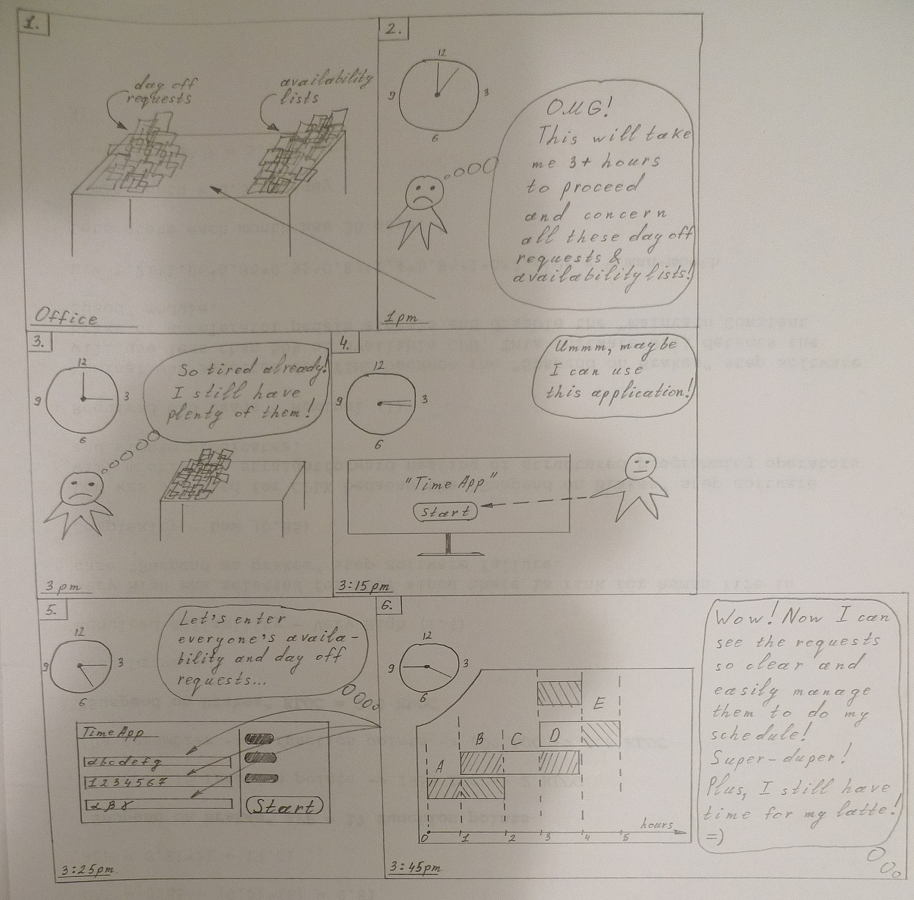


Figure 4. Storyboard 1

The following storyboard shows an alternative idea of an application that can effectively process employees’ day offs, availability, and can be used as a communication portal between employees.

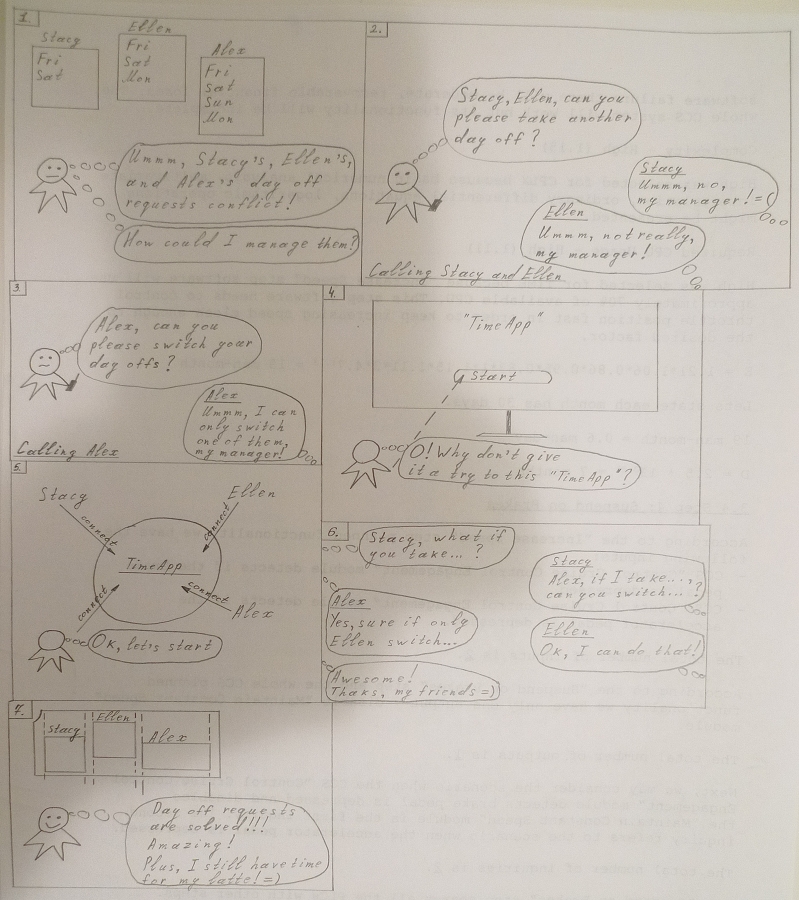


Figure 5. Storyboard 2

**Prototypes**

I’ve created two prototypes in Balsamiq Mockups application.

First prototype (*timeApp\_1.pdf*) represents an application with a tabbed layout. Each schedule creation step has its own tab; for example, “Day Off Requests” allows a user to select/enter day offs for a specific employee.

All the prototype files are located in the folder “*timeApp\_1\_files*”.

Second prototype (*timeApp\_2.pdf*) represents an application with the command line functionality/UI. Users type in the specific commands. Commands are executed by the application and the result is displayed in the command line.

All the prototype files are located in the folder “*timeApp\_2\_files*”.

Both prototypes represent an application doesn't actually create the final schedules. Instead, that application prepares schedule drafts that will be used later as a skeleton for the final schedule, which might be done, for example in Excel.

**Heuristic Evaluation**

Jacob Nielsen defines the heuristic evaluation as the following:

“Heuristic evaluation is a discount usability engineering method for quick, cheap, and easy evaluation of a user interface design.

Heuristic evaluation is done as a systematic inspection of a user interface design for usability. The goal of heuristic evaluation is to find the usability problems in the design so that they can be attended to as part of an iterative design process. Heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics")”.

Also, he outlines the ten general usability heuristics:

Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention

Even better than good error messages is a careful design, which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design

Dialogues should not contain information, which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

I asked 3 of my friends (after explaining/teaching them the HE) to test/evaluate the two prototypes I’ve created in Balsamiq according to the heuristic evaluation list. The evaluation ratings are based on the Jacob Nielsen’s ratings for usability problems:

0 = I don't agree that this is a usability problem at all   
1 = Cosmetic problem only: need not be fixed unless extra time is available on project   
2 = Minor usability problem: fixing this should be given low priority   
3 = Major usability problem: important to fix, so should be given high priority   
4 = Usability catastrophe: imperative to fix this before product can be released

The following is the heuristic evaluation feedback I received, plus my own suggestions.

Heuristics broken in prototype 1 (*timeApp\_1.pdf*)

* *Visibility of system status*: currently selected employee is not displayed on other screens, and there is no information whether the days off and availability changes where saved, and for which employee. Severity 3
* *User control and freedom*: If user realizes that the changes are being made to a wrong employee, there is no way to un-do or restore original selections. Severity 3
* *Consistency and standards*: user may wonder what is the difference between  "Schedule Draft" and "Start Draft" - both open the same screen Severity 2
* *Consistency and standards*: user may wonder what "OR" means on the Availability screen - does selection on the top part of the screen overrides the bottom part, or vice versa? Or both are true? Severity 3
* *Error prevention*: are there any limits for user selections on each screen (min/max), are there any warnings/error messages when user exceeds them? Not displayed. Severity 3
* *Recognition rather than recall*: how the user is supposed to remember which employee was selected 2 screens ago? Severity 3

Heuristics broken in prototype 2 (*timeApp\_2.pdf*)

* *Match between system and the real world*: in real word command-line interfaces are no longer being developed. Severity 4
* *User control and freedom* - no un-do functions. Severity 3
* *Recognition rather than recall*: the user has to remember a lot of information, all commands and all parameters, or visually search through extended multi-lines "help" text. Severity 4

Self:

Issue: slightly difficult to understand the system

Severity: 1

Heuristic Violated: Match between System and World

Description: at some point the app looks complicated and that becomes slightly difficult to navigate.

Other:

That doesn't clear how the app creates schedule drafts. Also, that's slightly difficult to understand what functionality each button has. Maybe adding several pop-up help messages or tooltips will help to clarify the app interface.

**Prototype to Implement**

According to the heuristic evaluation feedback I received and self-analysis of the two prototypes (*timeApp\_1* and *timeApp\_2*), I’ve decided to build the first prototype (*timeApp\_1*). Its concept is more realistic and covers more users needs than the second prototype (*timeApp\_2*).

The following is the list of changes I’m going to make to my prototype – timeApp\_1 – based on the heuristic evaluation feedback.

1. Add "Employee: selectedEmployeeName" text field on "Employee", "Day Off

Requests", and "Availability" tabs to notify/remind a user which

employee he/she is currently working with. This text field is going to

be placed right under the current step description text, i.e. under the

"1/3. Please select one of your employees:", "2/3. Please, select day

offs:" and "3/3. Please, select availability for the week of: ".

2. Change the steps description text from "1/3. ...", "2/3. ...",

"3/3. ..." to "Step 1/3. ...", "Step 2/3. ...", "Step 3/3. ...". This

will clarify to a user the sequence/description of steps, although,

that was not stated as heuristic violation.

3. Add a little pop-up window/tooltip saying "selectedEmployeeName was

selected" when a user clicks on (presses on) the "Submit" button on the

"Employee" tab. When the button is released (pressed off), the tooltip

disappears and the application functionality goes on.

4. Add a little pop-up window/tooltip saying "Day offs for

selectedEmployeeName were saved" after a user selects day offs on the

"Day Off Requests" sceen and clicks on the "Availability" button.

5. Add a little pop-up window/tooltip saying "Availability of

selectedEmployeeName was saved." after a user enters the selected

employee availability on the "Availability" tab and clicks "Start

Draft" button.

6. Add "Undo" button to the "Day Off Requests" tab to let a user

cancel/undo the previously selected day off. Also, the browser "Back"

button provides the same functionality. The "Undo" button will be

placed under the calendar on the left side to destruct a user less and

let him/her concentrate on the day off selection more.

7. Change the "Start Draft" button text to "Make Draft". This should

clarify the functionality of this button to a user.

8. Remove the top part of the "Availability" tab and leave only the bottom

part to let a user specifically select the selectedUserName

availability. On one hand, both parts provide more freedom to a user to

enter availability. The top part is simple and fast, while the bottom

one is more specific. On the other hand, the existence of both parts

might be confusing for a user since this is not clear which

selection(part) is going to be saved/prioritized/submitted or if one of

them overrides the other. Plus, the implementation/development part of

this might be difficult. Consequently, I guess, this is better to leave

only one option to select availability, and the bottom one is better

because it's more specific and versatile.

9. Add the "Anytime" checkbox and "Days/week" dropdown list to the

"Availability" tab to let a user quickly provide the

selectedEmployeeName availability in case that employee can work

any day, anytime. The checkbox and the dropdown list will be placed on

the top of the "Availability" page right below the text fields (Step

description and selectedEmployeeName). When a user checks the box or

clicks the list, a little pop-up window/tooltip appears saying "This

disables the more specific availability selection", and the bottom part

becomes disabled. When a user unchecks the box and returns the dropdown

list to the "n/a" position (this going to be one of the coded option),

the bottom availability selection becomes enabled.

10. Add a little pop-up window/tooltip to each button: "Schedule Draft",

"Employee", "Day Off Requests", "Availability", "Make Draft". When a

user hoovers to one of those buttons, the tooltip will appear

describing the button functionality.

11. Add a little pop-up window/tooltip to the "Save As" dropdown list on

the "Schedule Draft" tab saying "Select the type of file you want to

download the displayed draft to your computer as. After pressing

"Enter", the download will start."

12. Add the "Discard all" button to the "Availability" tab to let a user

discard all selections/submissions made on the previous screens.

13. Add a little pop-up window/tooltip to the "Discard All" button on the

"Availability" tab saying "Discards all selections/submissions you

made on the "Employee", "Day Off Requests", and "Availability" tabs.".

This will describe/clarify the button functionality to the user.

Since the selection of an employee on the "Employee" page is designed as a group of radio buttons where each button corresponds to only one employee, only one employee can be selected at a time. If a user selected an employee, clicked on the "Submit" button, and later on realized another employee should be selected, the user can go back to the "Employee" page and select another employee. Unfortunately, all the submitted data such as day off requests and availability will be lost. This is one of the disadvantages of the application.

When a user presses the "Make Draft" (previously "Start Draft") button, the new schedule draft is created and displayed at the "Schedule Draft" page. Plus, all the selections made previously on the "Employee", "Day Off Requests", and "Availability" pages are saved to let the user to change them and make another schedule draft to see the result of changes.

The improved prototype is the *timeApp\_1\_Improved.pdf*.

**Content Inventory**

The prototype to implement is *timeApp\_1\_Improved*.

The prototype consists of the head section, main navigation menu that represents four (4) logical steps of the schedule creation process, four (4) corresponding tabs/pages that uncovers those steps, addition navigation menu, and duplicate link/text navigation menu.

The head consists of the following elements/items:

- Company logo

- Application name text field

- “Home” link that represent the log-in page

- “Help” link that represent a help section

- “FAQ” link that represent a questions/answers section

- “Log-Out” link that logs a user out and redirect the application flow

to the log-in screen

Beneath the head there is a 24-hour watch widget that is supposed to help users (i.e. coffee shop managers) to indicate the time easily.

The main navigation menu consists of five (5) buttons (<button></button>):

- Schedule draft

- Employee

- Day Off Requests

- Availability

- Make Draft

Also, there is a duplicate navigation menu on the bottom which is text/link based. That menu consists of the four (4) text links:

- Schedule Draft

- Employee

- Day Off Requests

- Availability.

Next, there is an addition navigation menu that consists of three (3) text links to the external resources: Google, Twitter, Facebook. According to the “activity observation” phase, coffee shop managers tend to use those resources (Google, Twitter, and Facebook) quite often to communicate with employees. The addition navigation is located beneath the main navigation menu but above the duplicate navigation menu.

There are four (4) tabs/pages. Each tab represents the corresponding main navigation menu button.

First, the “Schedule Draft” tab consists of the following elements/items:

- Date picker/calendar tool to select the required week

- A text representation of the schedule

- Drop down list that offers a selection of file types a schedule might

be saved as.

- Three buttons – “Prev”, “Next”, and “Week of …” – to navigate between

different schedules according to a date

- Several text lines/boxes for different indication purposes.

Second, the “Employee” tab consists of the following elements/items:

- Several text lines/boxes for different indication purposes

- Several radio buttons belonging to one radio group. Each button

represent each employee. Consequently, a user can select only one

person at a time

- “Submit” button to save the selected employee.

Third, the “Day Off Requests” tab consists of the following elements/items:

- Several text lines/boxes for different indication purposes

- Two month calendar selection tool with individual clickable days. When

a user clicks a particular day, that day is saved as a selected

employee day off

- “Undo” button to discard the previously selected day.

Fourth, the “Availability” tab consists of the following elements/items:

- Several text lines/boxes for different indication purposes

- Date picker/calendar tool to select the required week

- Drop down list to select how many days per week an employee can work

- Several check box button to select shifts for the selected employee

- “Discard All” button that undo every selection made on the previous

steps: “Employee”, “Day Off Requests”, and “Availability”.

**Site Map**

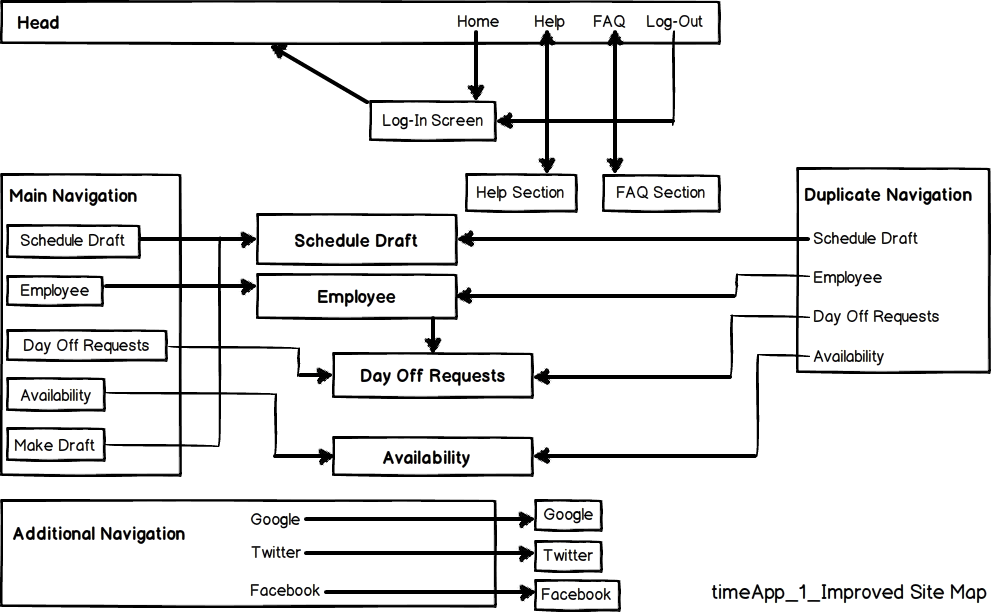


Figure 6. timeApp\_1\_Improved Site Map (see *timeApp\_1\_Improved\_SiteMap.pdf*)

**Prototype Development and Design Study Method**

The design study method for the second phase of the final project is prototyping. This method was chosen because of its interactive and presentation nature, for example:

* Prospective users can play with a live approximation of the actual application
* Prospective users can easily visualize where the development is moving and how the final project will look in reality
* Site content, element organization / layout, and navigation might be visually tested against first level (iteration) requirements
* Potential implementation technology problems might be discovered at early stages leading to a “safe” implementation plan revising
* Scope of the interface might be tested and deeper re-identified.

**Study Plan / Methodology**

There was created a low/medium fidelity prototype at the first stage covering the following interface areas:

- Head section with links to the “Home”, “Help”, and “FAQ” sections.

“Home”, “Help”, and “FAQ” sections were not implemented due to its less

importance comparing to the main tabs/ screens

- Main navigation section with its five (5) buttons: “Schedule Draft”,

“Employee”, “Day Off Requests”, “Availability”, and “Make Draft”. Each

button opens the corresponding tab

- “Schedule Draft” tab

- “Employee” tab

- “Day Off Requests” tab

- “Availability” tab

- Duplicate navigation that opens the corresponding tabs: “Schedule

Draft”, “Employee”, “Day Off Requests”, and “Availability”

- Additional navigation section that redirects users to external web

sites / media: Google, Twitter, Facebook.

The functional version of the prototype was done in *Axure RP* application. All the files are located in the *timeApp\_1\_Improved\_Prototype* folder.

To launch the prototype, copy and paste the following URL (<http://dl.dropbox.com/u/116974462/start.html>) to the address bar of the Firefox or Chrome, or navigate to the *timeApp\_1\_Improved\_Prototype* folder and open the “start.html” file with Mozilla Firefox (preferred) or Google Chrome browsers.

Also, there was done a slightly redesigned version of the prototype to let users compare two different design ideas of the prospective application during testing phase.

The redesign part is the left side of the original prototype (timeApp\_1\_Improved) and the navigation structure.

The redesigned functional prototype was done in *Axure RP* application.

All the files are located in the *timeApp\_1\_Improved\_Redesign* folder.

To launch the redesigned prototype, copy and paste the following URL (<http://dl.dropbox.com/u/116974462/timeApplicator_redesign/start.html>) to the address bar of the Firefox or Chrome, or navigate to the *timeApp\_1\_Improved\_Redesign* folder and open the “start.html” file with Mozilla Firefox (preferred) or Google Chrome browsers.

Please note, this functional prototype consists of only the initial “Schedule Draft” page; links to other pages are disabled.

After both prototypes – the initial one and the redesigned one - are reviewed and a user testing is done, there will be designed a plan for an evolutionary prototyping covering medium and high fidelity levels. According to this plan, a structured, robust, and responsive prototype and its different variations will be built and constantly refined leading to the final application creation.

**Prototype Rational Statement**

Assumptions / Justifications:

- A user opens the initial prototype and its slightly redesigned

variation in Mozilla Firefox or Google Chrome. The prototypes were not

tested and refined to work with Safari and IE.

- “Home”, “Help”, “FAQ”, and “Log-Out” text links in the head section do

not redirect a user to the corresponding pages / sections. This

functionality will be implemented in the next prototype iterations if

applicable (i.e. if the user testing indicates these links are

important and suitable)

- “Schedule Draft” section week selection tool doesn’t always work. The

reason is unknown. If it doesn’t display the calendar, a user needs to

reload the page. This bug will be corrected in the next prototype

iterations if applicable. Also, there is only one schedule displayed.

- “Schedule Draft” section week selection tool doesn’t always close

itself while clicking on it second time. This bug will be corrected in

the next iterations.

- “Schedule Draft” section “Prev” button doesn’t show the previous week

schedule. When a user clicks on this button the corresponding

message is displayed explaining the desired functionality. The original

functionality is scheduled to be implemented in the next iteration if

applicable.

- “Schedule Draft” section “Next” button doesn’t show the next week

schedule. When a user clicks on this button the corresponding

message is displayed explaining the desired functionality. The original

functionality is scheduled to be implemented in the next iteration if

applicable.

- While on the “Employee” section tab, a user is supposed to select an

employee and press the “Submit” button in order to enable the

“Day Off Requests” and “Availability” buttons.

- “Day Off Requests” tab is not interactive. This tab functionality is

scheduled to be rebuilt in the next prototype iteration. The desired

functionality is that a user clicks a day, and that day is assigned as

a day off for the selected employee (the employee name is displayed on

the top). By pressing the “Undo” button, the previously selected day

will be discarded.

- “Availability” tab is not interactive. This tab functionality is

scheduled to be rebuilt in the next prototype iteration. The desired

functionality is that a user selects a week for which the selected

employee availability will be provided. Then, the user might select

“Anytime” check box indicating the selected employee can work anytime

and the amount of working days / week. The other option is to select

availability for each day particularly. All changes are saved

automatically. By pressing the “Discard All” button, all the selections

will be discarded.

**User Study and Revision**

The following is the test evaluation plan I’ve created for user testing.

The name of the prototype was chosen to be “timeApplicator”.

**What to test:** - web application “timeApplicator” that creates schedule

drafts based on the provided employees’ day offs and

availability;

- the application interactions with a user;

- sequence of steps incorporated in “timeApplicator”;

- visual effectiveness of the “timeApplicator”

elements layout.

**The main test goal** is to find out if the “timeApplicator” helps users to reduce the process of preparing a schedule draft from several hours to less than half an hour.

*Justification: this is the main goal of the prototype, so it’s going to be the main goal of testing.*

**Additional test goals** are to find out:

- how visually effective the steps of preparing a schedule draft

introduced in the application are;

- how logical the steps of preparing a schedule draft introduced

in the application are;

- if an application can be effective enough to process a schedule draft

since this is more a creative process than a straight forward task;

- if the process of creating a schedule draft first and a final schedule

based on that draft second can be effective, productive, and less time

consuming;

- how effective the time representation technique introduced in the

application is;

- if the application screens/tabs are too cluttered or complicated;

- if the components are laid out effectively;

- if the application navigation feels comfortable, intuitive, and simple;

- how fast a user can understand how the application works;

- if users would like to use this kind of application;

- what kind of tool users would like to see/use as a schedule preparation

assistant.

*Justification: the application is supposed to help users to prepare a schedule draft easily, but not to confuse and make them even busier. Consequently we need to test how effective the procedure, layout, and time representation are, how logical the steps are, how cluttered/uncluttered the application tabs are.*

**Participants**

I’m going to invite 5 people to test the application.

Since the main user group of “timeApplicator” consists of coffee shop managers, I’ll ask three (3) of them to participate in the testing. The rest two (2) people are going to be my colleagues who are interested in HCI area. This way, I guess, I’ll be able to test the UX and professional/practical side of the application.

I’ll ping/text three managers I know and politely invite/ask them to participate in the testing. Also, I’ll ask my colleagues who wants to volunteer for the testing and select two of them if there will be more people willing to do that. I’ll provide a consent form for everyone.

**Time:** the testing will be done from Nov 11 to Nov 17 for coffee shop

managers depending on their availability and on Nov 16, Friday, for

my colleagues approximately at lunch time (12:30pm).

**Place:** I’ll bring my laptop to the coffee shops where my friends-managers

work and do the testing there (also, they can use their desktops if

des4ired); while I’ll use desktops at my work place for my

colleagues testing.

**Process:** first, I’ll show testers the application and ask them to describe

how they understand what this app does and what its main purpose

is by looking only at the main page. This, I guess, will show how

simple, intuitive, and self-describing the application appears to

be. Second, I’ll ask testers to perform a schedule creation

procedure (see the explanation below). Third, I’ll interview the

testers and write down their answers. Fourth, I’ll show the

redesign component of the application and ask testers to play

with that for several minutes. After, I’ll ask several questions

regarding their experience.

**Timing:** each testing procedure is planned to take approximately 23

minutes: 5 minutes for each of the three question phrases, 3

minutes for tasks, and 5 minutes for the redesign part testing:

talking about the application first, performing a task second,

answering questions third, and testing the redesign part fourth.

**Tasks**

First, I’ll show a tester the first page/tab of the application and ask him/her to describe what the application does, what its main purpose is, if the application looks attractive and engaging, if the first page/tab looks simple or complicated and cluttered. This will help to test the effectiveness and clarity of the application when a user looks at it for the first time.

Second, I’ll explain the tester how the application works and ask him/her to create a schedule draft. Unfortunately, I could not finish all the logic and interactions of the application, so when the tester hits the spot which is not yet implemented, I’ll provide explanation how it’s supposed to work and what kind of interactions are going to be there. Consequently, I’ll fake the parts of the application, which are not functional yet. The process of creating a schedule consists of selecting an employee, selecting that employee day off requests and availability, and pressing the button to make a schedule draft. Then, I’ll ask the tester to download the created draft to a local machine (my laptop or a desktop at my work place depending where the testing is going to be). Also, I’ll ask the tester to proceed with the same task again, but this time to undo several selected day offs for the selected employee. Then, the tester will be asked to discard all selections that were made. Finally, I’ll ask the tester to browse through previously created schedule drafts. This sequence of steps will test how effective, intuitive, cluttered, and simple the application navigation, steps, and components’ layout are.

Third, I’ll ask the tester several questions regarding the interactions with the interface and how it feels to work with the application.

Fourth, I’ll show the tester the redesign part of the application prototype and let him/her to play with that for several minutes. Then, I’ll ask questions regarding the interactions with the redesign part and how it feels to work with that in comparison with the original part.

*Justification: this will test, according to a certain degree, how effective, logical, and self-descriptive the interface is.*

**Measures**

I’ll write down the answers for the first, third, and fourth parts of the testing and analyze them later. Also, I’ll prepare the scale from 1 to 5 for several questions and ask a tester to chose the degree that relates to a particular question. For example, 1 will stay for the “so-so” and 5 for the “attractive” for the question related to how attractive the application first screen/tab looks.

I’ll record how fast a tester completes each task for the second part of the testing.

*Justification: since the main purpose of the application is to help users reduce the time preparing a schedule draft from several hours to less than half an hour, time required to finish each task is really important and informative measure.*

Also, I’ll record all the errors that will/might occur during the testing.

The testing might be considered successful if the tester could finish all the tasks and navigate easily in the application.

Also, each tester will be given an informed consent form (see informedConsentForm.docx) to sign. All the signed consent forms are located in the “ConsentForms” folder.

The following is the experimental protocol.

timeApplicator **experimental protocol**

A. Set-up

1. Arrive 15 minutes before the testing session.

2. Turn on a computer / laptop and make sure the Internet connection is on and running.

    2.1 If there is no Internet connection, plug in to the computer /

laptop the provided USB drive.

    2.2 Launch Google Chrome or Mozilla Firefox.

    2.3 Click on the "File / Open file" at the top navigation menu of the

browser. The pop-up window apperas asking you to indicate the

required file location.

    2.4 Navigate to the USB drive, choose the folder "timeApplicator".

    2.5 Select the file "start.html" and press "OK".

    2.6 Make sure the application prototype starts loading.

    2.7 Open second tab of the browser (that will be used for the testing

of the redesign prototype).

    2.8 Repeat steps 2.3 and 2.4

    2.9 Select folder "timeApplicator\_redesign".

    2.10 Select the file "start.html" and press "OK"

    2.11 Make sure the application prototype starts loading.

    2.12 Switch to the first tab of the browser.

    2.13 Go to the step 15.

3. Lauch Google Chrome or Mozilla Firefox.

4. Since the application prototype is hosted via Dropbox, go to the

Dropbox web site and sign-in with the following cridentials:

    Email:       xyz@abc.com

    Password: qwerty1234

5. Navigate to the "Public" folder, find the "start.html" file, right

click mouse on it, and choose "Copy the public link".

6. Confirm the copy.

7. Open a new tab in Chrome or Firefox and paste the copied link to the

address bar.

8. When the application prototype is loaded, scroll up / down, click all 5

navigation menu buttons on the left consequently, click the date

picker  icon on the top to make sure the prototype works correctly. If

you experience any problems, refresh the page. This will help to avoid

confusions during testing that might be caused by Internet connection.

9. Go to the Dropbox Public folder again.

10. Navigate to the "timeApplicator\_redesign" folder, find the

"start.html" file there, right click mouse on it, and choose "Copy the

public link" (this is the redesign version of the application

prototype).

11. Confirm the copy.

12. Open another tab in the Chrome or Firefox and paste the copied link to

the address bar.

13. When the application prototype is loaded, scroll up / down, click the

calendar on the left, select different months, and click the "Prev" /

"Next" buttons of the slider to make sure the prototype works

correctly. If you experience any problems, refresh the page. This will

help to avoid confusions during testing that might be caused by

Internet connection.

14. Return to the second tab of the browser.

15. Turn off the computer / laptop monitor so the tester won't see the

application at arriving.

16. Get forms ready for tester arriving. The required forms are:

*Informed consent form (see the informedConsentForm.docx)*  
*Questionnaire form (see the timeApplicator\_questionnaire.docx)*

17. Get a blank list of paper ready for you to take notes.

B. Informed Consent:

1. Read the following:

"Hi! Thank you for agreeing to participate in this application testing. Today, I'm asking you to serve as an evaluator of this application and to answer / complete a set of questions and scenarios. My goal is to see how easy or difficult you find the application to use.

I’m here to record / note your reactions and comments of the application you’ll view.

First, I'll show you the application initial screen / tab and give you 3 minutes to simply look at that without clicking on any element. You can use a mouse to scroll up or down, but please, don't click on anything yet. When time is over, I'll ask you if you would like more time to look or think about the application. If you do, I'll give you another 3 minutes. When you are ready I'll ask you several question regarding what you think about the application.

Second, I'll explain you what the application is about, what's its main purpose and what it does. Then, I'll ask you to perform several simple tasks and record the time you need to complete them. If you will have any problems, or get stuck, don't worry, that’s the UI at fault not you. If you decide not to perform any given task at anytime for any reason, don't worry as well. We'll simply move to the next one. Unf0rtunately, not all the elements of the application are fully functional, so when you hit such an element, I'll explain you how that works and provide any additional information. If a not fully functional element prevents you from a task completion, we'll move to the next task. Also, time to time I'll take photos if you don't mind. This will help me with the testing results evaluation.

Third, when you finish with the tasks, I'll ask you several question regarding your experience and record your answers. Some questions will contain the scale of 1 to 5, where 1 means not at all, while 5 stands for very something.

Fourth, I will show you the redesign version of only one element of the prototype and give you 5 minutes to check / play with that. When time is over, I'll ask you several questions regarding your thoughts or how you like / dislike the redesign element comparing to the original version."

"Ok, can I ask you to take a look at the consent form. It basically outlines that there’s no benefit for you taking part in this testing, but equally there’s no risk. It also asks you to give permission for me to take photos. Any photos I do take might be used for the BCIT

“COMP 8521: Selected Topics in Advanced Interface Design” course final project, but your face and any identifiable details will be removed or blurred out first. To help me take notes, I might ask you questions time to time. So could you take a look at that and sign it if you're comfortable."

2. Have the tester read and sign the consent form. You keep the consent

form.

3. Turn on the computer/ laptop monitor and show the tester the original

application (second tab on your browser).

4. Start counting 3 minutes.

5. When time is over, ask the tester if more time is required. If yes,

give 3 more minutes. If not, proceed to the next step.

6. Inform the tester you are going to ask him/her several question

regarding the application's first impression.

7. Read the following:

"Remember, that it’s the application under test here, not you. So don't worry about making mistakes. If you have any problems, or get stuck, that’s the UI at fault not you, and those are exactly the things that I want to find in the test. Also, there is no right or wrong answer. I really just want to know if the application is designed well for you."

8. Ask the "part\_1" questions from the questionnaire (see the

*timeApplicator\_questionnaire.docx*) and record the tester's answers.

9. Inform the tester you are going to ask him/her to perform several

tasks.

10. Start with the task 2.1 from the questionnaire (see the) and finish

with the task 2.5. Record time the tester needs to complete each task.

If the tester is stuck, ask the reason and move to the next task. If

the tester hits a not fully functional application element, explain

Its functionality and let the tester continue based on the explanation

You provided.

11. When all tasks are done, inform the tester you are going to ask

several question regarding his/her experience playing with the

interface.

12. Ask the "part\_3" questions from the questionnaire (see the

*timeApplicator\_questionnaire.docx*) and record the tester's answers.

13. Inform the tester you are going to show the redesign element of the

prototype.

14. Open the third tab of the browser, show the redesign version, and

start counting 5 minutes.

15. When time is over, ask the tester "part\_4" questions from the

questionnaire (see the *timeApplicator\_questionnaire.docx*) and record

the tester's answers.

16. Write down any additional notes or suggestions the tester might have.

17. When you are done with the questions, thank the tester and indicate

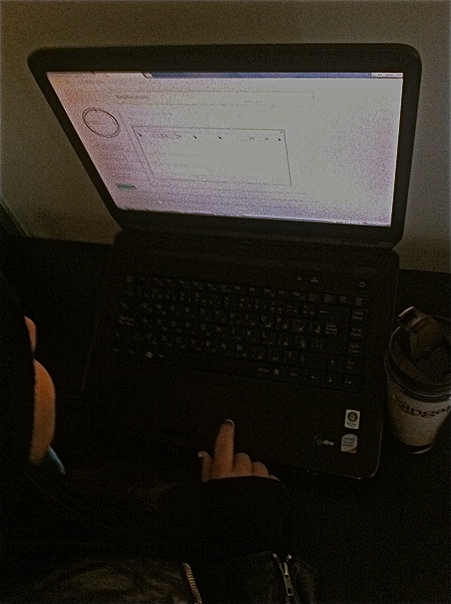
that was really great to work with him/her.

Test questions are stored in the *timeApplicator\_questionnaire.docx* file.

The testers answers to those questions are located in the *TestersQuestionnaireAnswers* folder.

The following are three photos of user tests in action.

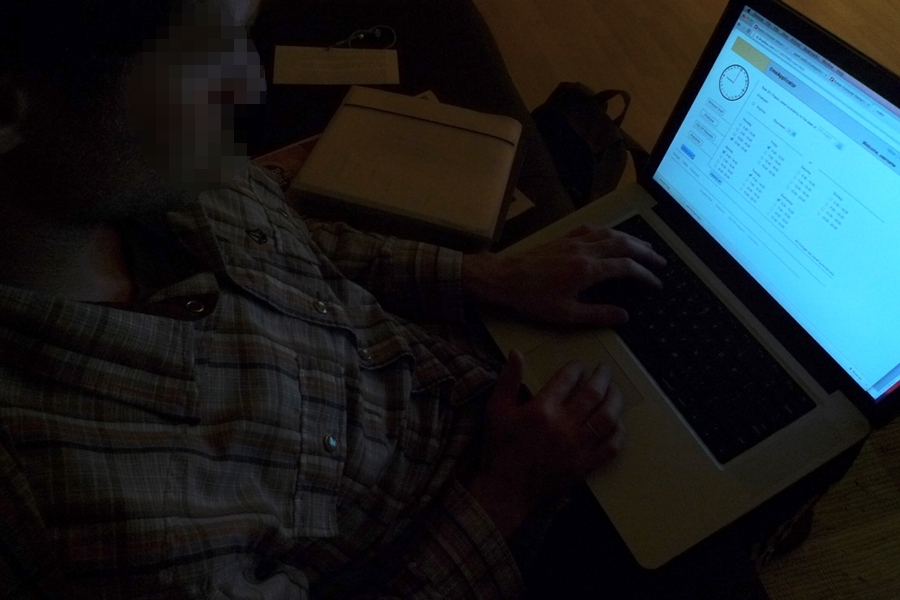
This photo indicates the tester couldn’t understand for quite a while what to do, where to go, and what to click on the “Day Off Requests” page:



The photo below shows the tester is confused and disturb looking into the too colorful schedule draft table. Also, the tester couldn’t understand how to save/download the draft to the local machine:



The photo below shows the tester is confused about the functionality of the “Make Draft” button since it behaves as the “Schedule Draft” button:



**Summary of findings:**

- the main purpose of the application was understood correctly by all

testers: schedule employees’ time, manage employees’ shifts.

This indicates that the application elements layout and its overall

organization is self-descriptive enough to be understood easily during

first 25-30 seconds.

- all testers positively admitted the duplicate textual navigation on the

bottom. All testers said this is very helpful and supports intuitive,

simple navigation. - all testers could understand and liked what the

“Google”, “Twitter”, and “Facebook” links are for. Those social

platforms are really helpful while managing people’s time.

- all 5 testers finished the main practical task of the testing – creating

a schedule draft – in less than a minute. An average time for completion

is 39.33sec. In real working environment this time will be greater and

approximately equal to 2 minutes at least per each employee. Considering

the fact that an average coffee shop team consists of 13 employees, the

overall time for schedule draft completion will be 13 x 2 = 26 minutes

at least. Consequently, the main testing goal is achieved

– to find out if the “timeApplicator” helps users to reduce the process of

preparing a schedule draft from several hours to less than half an hour.

- the rest 4 testing practical tasks were executed almost without

incidents and in less than 20sec. However, the “Day Off Requests” page

was really confusing for 2 testers. They didn’t understand how to select

day offs and how to save/submit them. Also, one tester didn’t understand

how to save/download schedule draft to the local machine. The tester

indicated there was no call to action button.

- all the testers could easily understand how many pages/tabs the

application prototype consists of. They indicated there are 5-7

pages/tabs while in fact there are 4 main tabs and 3 not yet implemented

tabs. Consequently, the whole application prototype structure was easily

understood by all testers.

- the application navigation is simple, clear, well organized and provides

the good contrast among all other elements. However, the navigation

transition between pages/tabs might be confusing because 2 out of 5

testers were slightly lost while browsing the application prototype.

Also, all 5 testers stated the relation between buttons and

corresponding pages/tabs is not clear (awful) and buttons’ text is not

self-describing enough to be understood in the first 5-10 seconds.

- the application home page doesn’t look attractive enough to fully

encourage / welcome users to work with it. An average score for the home

page attractiveness is 3.2 out of 5, where 1 means not attractive at all

and 5 stands for very attractive.

- addition links such as “Home”, “Help”, “FAQ” acquired much testers’

attention instead of being simply noticed. This indicates the home page

is kind of cluttered because the testers paid more attention to minor

functionality instead of noticing the main navigation buttons and

functionality first.

- the purpose of the calendar tool on the home page was not understood

correctly. Most testers thought this is simple day selection tool for

any kind of reference. The desired purpose of the calendar is to select

schedule drafts based on the calendar days. None among the testers

noticed the text explaining the purpose of the calendar – “The schedule

draft for the week of:”.

- the “Home” link on the top of the initial screen / tab confused the

testers. They didn’t understand if the link is there so what the page

they are currently at then. This indicates the confusion between the

application initial screen and the desired but not implemented home

page.

- the “Employee” button was confusing since 4 out of 5 testers thought

this will redirect them to the list of employees. Consequently, the

button text is not sufficiently self-describing. The revision has to be

considered.

- the “Day Off Requests” button was slightly confusing because 2 out of 5

testers thought this will redirect them to the list of the employees’

day offs. Those 2 testers stated the button indicates like employees

submit their day offs which are stored under the “Day Offs Requests”

button later. The desired functionality is to manually assign day offs

for employees based on their requests which are out of scope of this

application.

- the “Availability” button was confusing since 3 out of 5 testers thought

this will redirect them to the list of employees’ availability which

they submitted before. However, the desired functionality is to manually

enter employees’ day offs based on their requests which are out of scope

of this application.

- 3 out of 5 testers didn’t understand what the “Save as:” text line and

the drop down list are for. After the provided explanation, the testers

pointed to the absence of the call to action elements there. They

mentioned that would be better to have a button instead of a textual

line.

- the color schema of the schedule draft (the actual table) is too

aggressive and disturbing. The testers indicated this should be changed

since this causes too much distortion. Consequently, the actual schedule

draft table has to be redesigned.

- the 24 analog watch on the top left corner of the application might be

slightly confusing since the time shown in the schedule draft table is

digital. There is no “recognize rather than recall” mechanism here.

- the logic of the application prototype data flow is low and unclear for

the first sight. All testers experienced difficulties understanding what

to do next, where to click next.

- the testers indicated the application prototype is simple, neat, old-

school, clear, almost uncluttered, and efficient. However, the data

flow, logic, buttons’ text, and pages/tabs component layout strongly

needs improvements.

**Summary of findings for the redesign component:**

- 4 out of the 5 testers indicated the redesign navigation layout and,

consequently, the home/initial page is approximately 70% less

uncluttered than the initial version

- the redesign version is more logical and clearly shows the sequence of

steps comparing to the original version. However, the navigation still

looks slightly complicated for the first sight

- the redesign version is easy to grasp and understand

- the redesign version is clean, neat, simple, and well-structured - this

is clearer how to browse between different schedule drafts and how to

save a draft to your local machine

- the calendar on the left side can be reduced to one month instead of two

- buttons' text plus steps sequence numbers really helps to understand the

data flow, however more work in this direction should be considered.

**Recommendations**

Changes I'd make to my application prototype based on the testing results and findings:

- Rethink the color palette of the schedule draft table. This will help

users to understand the schedule better, clearer, and easily as well as

will stop disturbing by possible aggressive and too bright colors.

Efficient color coding of the schedule draft table will also allow users

faster understand employees’ shifts and different scheduling options.

- Completely redesign the “Day Off Requests” section/tab. According to the

testers feedback this was the most hard to understand area of the

application. The redesign might include a pop-up window saying, “Please,

select day offs”, when a user clicks on the “Day Off Requests” button

and redirects to this section/tab. Only one month should be displayed to

minimize the confusion. Plus, the user has to be able to select day from

the next month as well, but not from the previous. This will

minimize possible errors by preventing the user from assigning past day

offs. Also, if the selection calendar is placed on the left side of the

section, a text field area might be placed on the right side. When the

user selects a day, that day is displayed in the right side text field.

This will show the user which days are already have been selected by

him/her.

- Remove the “Make Draft” button and transfer its functionality to the

“Schedule Draft” button. When a user finishes entering employees’ day

offs and availability, he/she can simply press on the “Schedule Draft”

button, and the new draft will be generated. “Make Draft” and

“Schedule Draft” buttons are somewhat redundant.

- Improve the hover effects of the links and buttons plus provide

additional navigation hints to allow users easily understand where they

are while browsing the application prototype. For example, the redesign

version contains text lines “1/3”, “2/3”, and “3/3” right before each

button indicating the logical flow of steps plus the position in the

interface. By making the opacity of this textual links equals to 0.25 by

default and equals 1 during hovering will definitely highlight the

current user position.

- Change the text of the buttons to make them more self-describing. This

will help users to understand the application data flow better plus will

help predict which buttons relates to each step of the schedule making

process.

- Minimize the home/initial page calendar (look at the redesign version)

to 1.5 months. Testers indicated 2 months calendar is too big / long,

while 1 month might be short due to monthly planning specifics and end

of the month reports while managers have to glance to the next month as

well. Consequently, 1.5 month calendar on the home page should work more

efficiently.

- Rethink the home/initial page watch. While the 24 hour analog watch is

closely related to the sun rotation and day light flow, this still might

be slightly confusing for several users. Plus, the time on the schedule

draft table indicated as digital, i.e. numbers. Consequently, having

digital watch on the page might benefit UX.

- Remove the “Home”, “Help”, “FAQ” links on the top right corner to make

the home/initial page simpler and guide the user attention to the main

functions/buttons – Employees, Day Off Requests, Availability. The

initial screen will play the role of the home screen. “Help” and “FAQ”

section might be incorporated into one, which can be introduced as a

question mark pictogram located on the right bottom corner next to the

text navigation links. This will still allow users to access the help

section while minimize the initial page element load.

- Remove the calendar tool that is located above the schedule draft (look

at the original version) and replace it with a slider (look at the

redesign version). This will streamline the process of browsing between

old/new schedule drafts. However, the option for “jumping” over several

drafts has to be designed/ created.

- Replace the “Save as:” button and drop down list with only one button –

for example, “Save draft” or simply “Save”. When a user clicks on that

button a pop-up dialog box will introduce the user several file formats

the draft should be saved as. This will minimize the confusion in saving

procedure.

- Think about the “Employee”, “Day Off Requests”, and “Availability”

button combination. All three of them might be incorporated into one

section since all are basically related to one task – schedule an

employee based on his/her availability. Consequently, “Employee” and

“Availability” sections might be combined into one area eliminating “Day

Off Requests” section since day off might be considered as a part of the

availability. As a result, the design might consist of the one/two tabs.

The first tab is for schedule draft table. The second part is for

employee-availability area. An employee might be selected from a

dropdown list on the left, while his/her availability might be chosen

from a list on the right. This approach/layout will simplify the overall

application structure, might improve the logic of data flow, and

decrease the navigation confusions the testers had.