The Good, The Bad, and The Interesting UI + Heuristic Evaluation (15%)

Welcome to the first assignment! This assignment aims at getting you started with having an eye of a user interface designer and be able to critically evaluate an interface.

There are two parts of this individual assignment. In the first part (5%) you are going to collect a total of three user interface examples. In the second part (10%) you are going to evaluate one user interface.

Part 1: The Good, The Bad, and The Interesting UI (5%)

Collect one example from **your house** for each category of good UI, bad UI, and interesting UI. For each example, include the following:

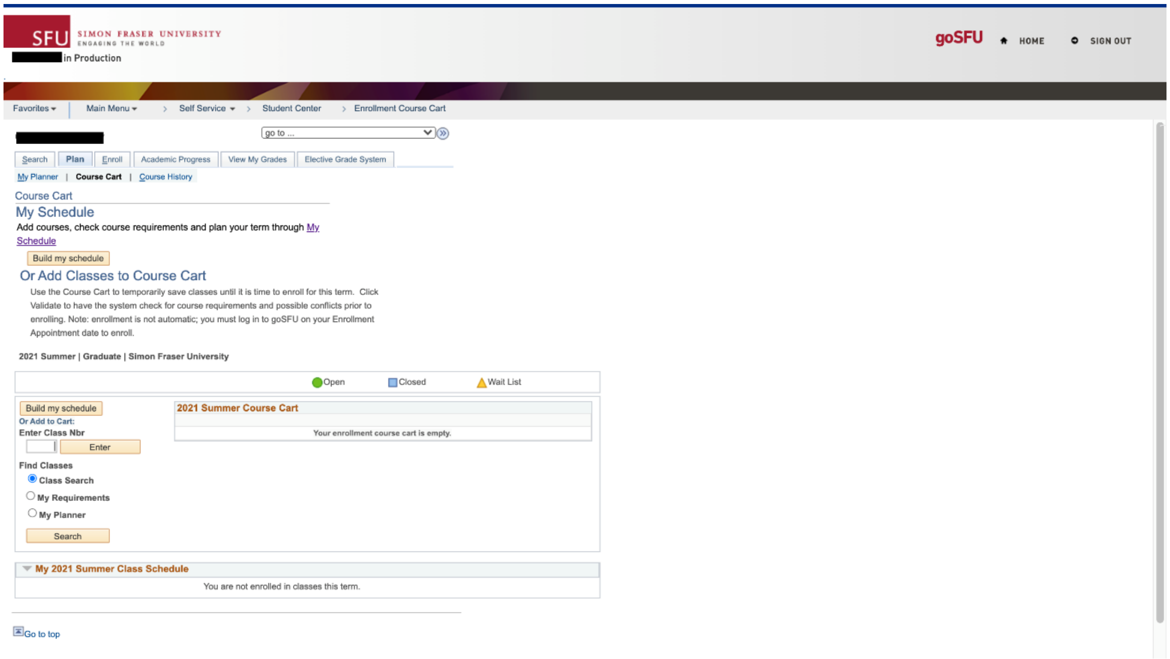
- At least one image clearly showing the interface.
- 1-2 sentences of why you think it is good/bad/interesting.
- Provide information on what it is, for example, brand, model, what does it do, …etc.

Here are several overall requirements:

- You must not use the examples shown in the lecture slides, or any of the suggested readings.
- The examples must be real, i.e., they must have appeared in existing products and cannot be something from design concepts or exercises.

Part 2: Heuristic Evaluation (10%)

Heuristic evaluation is a popular UX evaluation technique to access the overall usability of a software application without requiring actual end-users. In this assignment, you will conduct a thorough heuristic evaluation (using Nielsen's heuristics: https://www.nngroup.com/articles/ten-usability-heuristics/) for the goSFU web interface. For this evaluation, you can assume a PC browser is used to access this app.



[redacted] in Production

[redacted]

Course Cart

My Schedule

Add courses, check course requirements and plan your term through My Schedule

Or Add Classes to Course Cart

Use the Course Cart to temporarily save classes until it is time to enroll for this term. Click Validate to have the system check for course requirements and possible conflicts prior to enrolling. Note: enrollment is not automatic; you must log in to goSFU on your Enrollment Appointment date to enroll.

2021 Summer | Graduate | Simon Fraser University

Or Add to Cart:

Enter Class Nbr

Find Classes

● Class Search

○ My Requirements

○ My Planner

2021 Summer Course Cart

Your enrollment course cart is empty.

My 2021 Summer Class Schedule

You are not enrolled in classes this term.

A screenshot of the goSFU web interface (https://go.sfu.ca, yours might be slightly different).

Instructions

1. Familiarize yourself with the application as appropriate, completing tasks related to exploring and using the chosen system. Consider the heuristics discussed in class as you use the system and make note of problems and good aspects you encounter. Focus on student users and what their typical tasks may be.
2. Use the provided usability inspection report template (in the same folder as this description document on Canvas) for your findings.
   - **Identify two usability problems** (where heuristics are being violated), including severity level 2 (minor), 3 (major) or 4 (critical), grouped by heuristic. Use screenshots within the form to illustrate your findings (take the screenshots while you are identifying the problems, as later on the interface might be updated and you might not be able to find it again).
   - In addition, **include two examples of good usability** (i.e. where a heuristic is met instead of being violated) within your report.
   - **Summarize your overall process and the main findings** of your heuristic evaluation (max 1 page + individual evaluation forms).

Submission (Both Part1 & 2)

Submit both parts in a single PDF file to the corresponding folder on Canvas **by 11:59p, Jun 4**. Begin your document with a cover page stating that it is Assignment 1, followed by your name, SFU email, and student ID. Name the file in this format: **firstname_lastname_9-digit-studentID_Assignment1.pdf**

Assignment late penalty: 10% per calendar day (each 0 to 24 hour period past due), max 2 days late.

Academic Honesty

It is expected that within this course, the highest standards of academic integrity will be maintained, in keeping with SFU's Policy S10.01, "Code of Academic Integrity and Good Conduct." In this class, collaboration is encouraged for in-class exercises and the team components of the assignments, as well as task preparation for group discussions. However, individual work should be completed by the person who submits it. Any work that is independent work of the submitter should be clearly cited to make its source clear. All referenced work in reports and presentations must be appropriately cited, to include websites, as well as figures and graphs in presentations. If there are any questions whatsoever, feel free to contact the course instructor about any possible grey areas.

Some examples of unacceptable behavior:

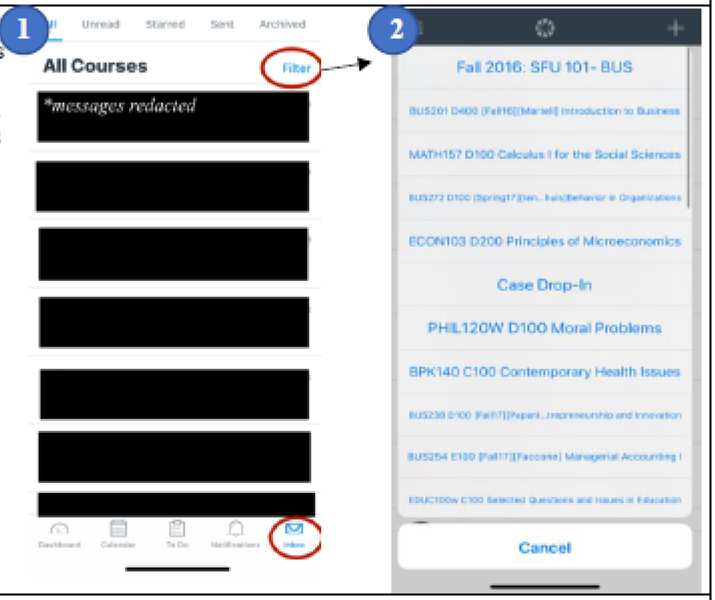
- Handing in assignments/exercises that are not 100% your own work (in design, implementation, wording, etc.), without a clear/visible citation of the source.
- Using another student's work as a template or reference for completing your own work.
- Using any unpermitted resources during an exam.
- Looking at, or attempting to look at, another student's answer during an exam.
- Submitting work that has been submitted before, for any course at any institution.

All instances of academic dishonesty will be dealt with severely and according to SFU policy. This means that Student Services will be notified, and they will record the dishonesty in the student's file. Students are strongly encouraged to review SFU's Code of Academic Integrity and Good Conduct (S10.01) available online at: http://www.sfu.ca/policies/gazette/student/s10-01.html.

Examples from Previous Terms (evaluating SFU's Canvas app)

They do not necessary represent the perfect answers but are here to show a general idea.

<table>
<tr><td>#: 2</td><td><b>Problem/Good:</b> Usability Problem</td></tr>
<tr><td colspan="2"><b>Name:</b>
Ineffective display of filters on inbox messages</td></tr>
<tr><td colspan="2"><b>Relevant heuristic:</b>
Flexibility &amp; Efficiency of Use, Consistency &amp; Standards</td></tr>
<tr><td colspan="2"><b>Evidence of issue:</b>
The filter issue is found under the inbox tab as seen in image 1. Selecting the filter option opens up a list of potential filters which are in order of oldest to newest class enrollments, as seen image 2.

Messages in image 1 have been redacted.

1
All
Unread
Starred
Sent
Archived
All Courses
Filter
Dashboard
Calendar
To Do
Notifications
Inbox
2
Fall 2016: SFU 101- BUS
BUS201 D400 [Fall16][Martell] Introduction to Business
MATH157 D100 Calculus I for the Social Sciences
BUS272 D100 [Spring17][Den...hals]Behavior in Organizations
ECON103 D200 Principles of Microeconomics
Case Drop-In
PHIL120W D100 Moral Problems
BPK140 C100 Contemporary Health Issues
BUS238 D100 [Fall17][Papani...trepreneurship and Innovation
BUS254 E100 [Fall17][Faccone] Managerial Accounting I
EDUC100w C100 Selected Questions and Issues in Education
Cancel
*messages redacted
[redacted]</td></tr>
<tr><td colspan="2"><b>Detailed explanation:</b>
The canvas app allows users to filter messages found in their inbox. However, you are limited to only filter by class and the order of the filters displayed are from oldest to newest. Thus, if a user is looking for a specific message by keyword, they are unable to. Moreover, filters for old classes are displayed first. Whereas, it is more likely that a user would be filtering messages for a current class rather than, for example a class from 2016. This violates the heuristic of "Flexibility and Efficiency of Use", as the use and display of filters is ineffective, increases time and mental effort.

The heuristic of "Consistency &amp; Standards" is also violated as the filters are not displayed in a uniform manner with some being extremely detailed in small font and others being vague in large font.</td></tr>
<tr><td colspan="2"><b>Severity or Benefit (low, medium, high):</b>
2 (Minor)

<b>Justification:</b>
This issue is minor as the filter option is an accessory feature moreover, the impact is not too significant as all filter are still present, merely users must scroll to the bottom to find most current/applicable class filters.</td></tr>
<tr><td colspan="2"><b>Possible solution and/or Trade-offs:</b>
A possible solution is to introduce the option to filter by keyword and change the order of class filters displayed; with the most recent classes showing up first in the list.</td></tr>
</table>

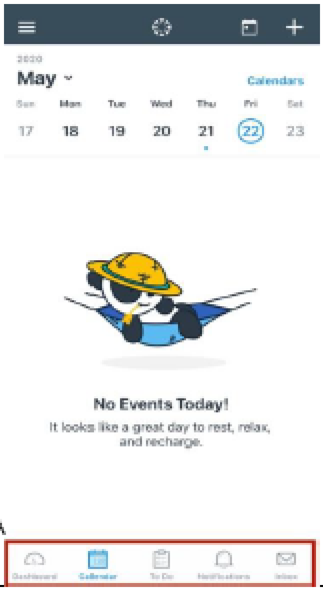
Comment: Good use of annotation and discussion of issue in the explanation. A potential trade-off of the solution could be searches for older classes might be slower when the order is most recent classes showing up first. Evaluation of severity is also reasonable as while it is an infilter, it is still functional and does not cause issues to the running of the app.

#: 4	**Problem/Good:** Good Usability

Name:
Good visibility of the system through the use of tabs

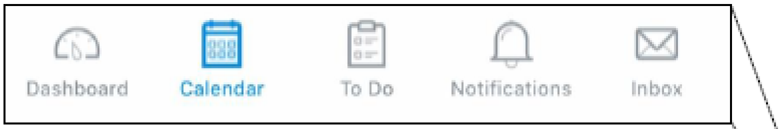
Relevant heuristic:
Visibility of System Status

Evidence of issue:
Various functionalities of the app can be accessed through the tabs at the bottom of the interface.



No Events Today!

It looks like a great day to rest, relax, and recharge.



Each tab view represents a functionality of the app.

Detailed explanation:
The heuristic of "Visibility of System Status" focuses on keeping users informed about what is going on in the system. The tab views of the Canvas Student app meet this heuristic as they serve as visual feedback that shows user location. Each tab represents a functionality, but also a location within the app. Thus, the tab view answers the user's question of "Where am I ?". The tabs make users aware of where they are in the app which promotes good navigation experience, furthermore, adds to the user's sense of control and trust.

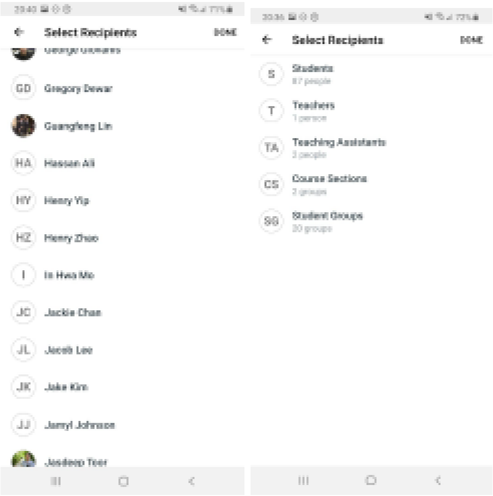
Severity or Benefit (low, medium, high):
4 (Critical)

Justification:
This app is complex and offers multiple functionalities. It is critical that the user is easily able to navigate to the functionality they want but, also are aware of where they are in the app. The tab view is the backbone of the interface, and plays a critical role in the operation of the app.

Possible solution and/or Trade-offs:
Realistically, there are no tradeoffs to the tab view. Without the tabs, the app's functionalities would be futile.

Comment: Great use of "blow-up" view to illustrate the functionality and good explanation on why the tab views are helpful. There is however a trade-off when including labels: added space requirement for the labels and they have to be short while explanatory.

2.

Problem/Good: Good
Name: List of relevant people is provided when composing a message.
Relevant heuristic: #6 Recognition Rather Than Recall
Evidence of issue:  Select Recipients DONE GD Gregory Dewar Guangfeng Lin HA Hassan Ali HY Henry Yip HZ Henry Zhao I In Hwa Mo JC Jackie Chan JL Jacob Lee JK Jake Kim JJ Jarryl Johnson Jasdeep Toor Select Recipients DONE S Students 87 people T Teachers 1 person TA Teaching Assistants 2 people CS Course Sections 2 groups SG Student Groups 20 groups
Detailed explanation: When sending a message, upon clicking the + button, the user is presented with a list containing groups of people, and then a list of people in the selected group. This meets the heuristic, because it saves the user from having to remember exact spelling of names, or their user identifiers in order to send messages.
Severity or Benefit: critical Justification: This is an essential feature to have in any messaging application. The convenience of not having to remember identifiers to send messages to known contacts is significant. Without this feature, most users would consider the messaging feature unusable.
Possible solution and/or Trade-offs: There is the possible minor trade-off of additional complexity in the application. A mere identifier field that the user can type into with the keyboard that they are already familiar with is clearly simpler than having to use multiple different UI inputs to enter an identifier.

Comment: Good illustration of the heuristic and linking to the use of the application. Possible solution ("a mere identifier field that the user can type into the keyboard) could be more clear – what kind of identifier?