

Test Procedure

PREREQUISITES

Examiners

In order to run this test, you will need at least two examiners. One is responsible for presenting the prototype to the subject as well as asking and answering questions, while the other is responsible for documenting the subjects' responses (either through text, audio or video, but ideally a combination), and for having the subject fill out the questionnaires. If possible, perform each test with three examiners, so subject responses can be documented both in written form as well as through audio/video.

When conducting the tests, please adhere to the principles outlined in the '[Observer Guidelines](#)' document.

For information about how to use the system, please see the '[User Manual](#)'.

Location

You will need to conduct your tests in the Brown Building on the McGill Campus. Be conscious of the building's opening hours when scheduling your tests.

Hardware

As we mention in the [User Manual](#), please use an iPhone if one is available. The system does not work optimally on Android devices at this time, and the quality of test results may be affected.

Test subjects

When recruiting subjects for this test, please try to select a diverse set of people. When we created our personas we considered three scales: (1.) social comfort, (2.) familiarity with the space, and (3.) time consciousness. Through this we created three user segments that would use our system; "independent wanderers", "social travellers", and "physically impaired visitors". To read a more thorough description of our user segments and personas, please read our user personas section in our [Proposal](#).

To correspond with our "independent wanderers" and "social travellers", you should select a range of people with different social comfort. In our context, the range starts from users who always seek out people to ask for directions to those who avoid asking questions at all costs. These individuals prefer to look for information on signs or maps.

In regards to our “physically impaired visitors”, you should ideally test with at least one person with a physical disability (specifically, one which can hinder their movement throughout a building), but this is not an absolute requirement. Please do not inappropriately ask physically impaired strangers to participate in our usability tests.

In general, two other qualities to look for when choosing test subjects is technical competency and prior knowledge of the Brown Building. We want to test the usability of our system, so it is important to test people who have different levels of comfort using smartphones and computer software in general. Also, this system is intended for people who are not familiar with a specific space and are looking for directions. Since the testing will be done in the Brown Building, test subjects should not be very familiar with the building (and certainly not people working there).

FORMAL PROCEDURE

Introduction

Before the subject arrives, or immediately upon arriving, have them read the ‘[User introduction](#)’ document. You can optionally choose to read it aloud to them to ensure they understand some of the fundamental test points.

Then, have them fill out the ‘[Pretest questionnaire](#)’. It is up to you if you would like to use paper copies then manually enter data in a spreadsheet, or use Google forms to collect responses.

Test execution and tasks

When the subject is ready, have the examiner execute the following test tasks with them. Always start with Task 1, but try to vary the order of execution of the other test tasks between the different test subjects.

NOTE: During test execution, you will have to configure the app’s starting location for each test, as well as bring the subjects to different locations in the building. For instructions on how to do this, see the [User manual](#).

Task 1: Describe the functionality of the system

Show the subject the prototype system, and let them look at it freely for 20 seconds. Then, ask them what functionality of the system they recognize. (e.g. “What are all the things this system can do?”)

Task 2: Navigate to a Student Service office

Configure the app starting location to "Campus Life & Engagement", and take the subject to this location (see the [User manual](#) for specific instructions and maps). Give the user the prototype, then ask them to navigate to the International Student Services office.

Task 3: Navigate to a conference room

Configure the app starting location to "Main lobby", and take the subject to this location. Give the user the prototype, then ask them to navigate to Conference Room 3001.

Task 4: Navigate using a wheelchair friendly route

Configure the app starting location to "SSMU Basement entrance", and take the subject to this location. Give the user the prototype, and ask them to navigate to the Counselling Office, but tell them to pretend that they are in a wheelchair, and are therefore unable to use any stairs to get around the building (unless they are actually in a wheelchair, in which case you don't have to pretend).

Task 5: Navigate to the nearest bathroom

Configure the app starting location to "Dr. Penfield Entrance", and take the subject to this location. Give the user the prototype, and ask them to navigate to the nearest bathroom

During task execution for all tasks: record the test results using the [Data collection sheet](#). Please use the parts of the sheet which correspond to the specific tests.

Post-test wrap-up

After finishing all the test tasks, have the user fill out the [Post-test questionnaire](#). Again, it is up to you if you would like to use paper copies then manually enter data in a spreadsheet, or use Google forms to collect responses.

HEURISTIC EVALUATION

In addition to the usability test described above in the 'Formal test procedure' section, we would like you to do a heuristic evaluation of the prototype. Instructions and Heuristics for this have been extracted and summarized from Nielsen's "How to Conduct a Heuristic Evaluation" and Nielsen's 10 Usability Heuristics for User Interface Design.

The Heuristic Evaluation should be done by each team member individually in a separate space. Team members should only discuss and combine their findings after all the evaluations are finished. In a section that follows, you will find selected heuristics from Nielsen's list and a justification on why they are relevant to our system. Using this list, go through the interface several times while noting the different elements that make up the interface design, and evaluate them in comparison to the list of heuristics.

As an evaluator, you have freedom in deciding on how to evaluate the heuristics that are most important to our system. However, it is recommended that you go through the interface at least twice (after reading the [User manual](#))

to guarantee a general understanding of the interface. The **deliverable** for this evaluation is a list of usability issues referencing the specific heuristic(s) that are being violated. This can be submitted in any form the evaluator prefers.

Note: *Since the prototype you are evaluating is not a complete product, it only has a select number of navigation routes and features implemented. Please refer to the "[Prototype Limitations](#)" section in the User manual.*

HEURISTICS:

1. Visibility of system status:

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

Justification: As a navigation application, users should not feel more lost than they already may be, which is why it is vital that users are aware of what the application is doing and where they are in the navigation process.

2. 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."

Justification: The main feature of this system is that it gives directions. If the directions are not clear because of lack of real-world conventions, the system has failed in finding a solution for improving indoor navigation.

3. 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."

Justification: If a user makes a mistake, there should be a clear exit option otherwise it would be very frustrating for them. For example, they would have to swipe through directions that are not relevant to them in hope that it will return them to the home screen.

4. Consistency and standards:

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

Justification: In the application, there are different ways to find destinations (search option, different menu tabs). It is essential that different options are cohesive and clear to user for them to find their destination.

5. 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.”

Justification: The system holds many different directions and options (i.e. handicap option). It is important that the system does anything it can to make sure the right direction and options are chosen as to not waste the user's time.

6. 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.”

Justification: This heuristic centers on our interface being familiar and easy to use which is essential considering the context our system will be used in. It is likely that users will be using the application under time-constraint and confusion trying to get to their destination.

7. 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.”

Justification: Users will use this application solely to get directions from their location to their destination. For this reason, the interface design should be minimalistic as possible to ensure that users can simply find their destination without inefficiently spending their time.

8. 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.”

Justification: The system should not make users feel more lost as that would defeat the purpose of the system. If they make an error, a solution should be shown so that they can continue their navigation.

CONCLUSION

Thank you again for participating in our usability tests.
