GARMIN®

Heuristic Evaluation

The **Introduction**

Purpose of the Document

This document serves to find the LiveTrack's baseline usability issues within the Garmin Connect application. This document will aid in the the involved parties' understanding of the Purdue UX team's redesign. The results of this evaluation will be used to ensure that, at the bare minimum, the problems found will be addressed in the solution. Please understand that both the beneficial and problematic aspects of LiveTrack will be noted in this document. Additionally, there will be an analysis of these results with a plan to move forward.

What is a Heuristic Evaluation?

A **Heuristic Evaluation** is an evaluation that is done using a set of specific heuristics. For example, in this document we will use Nielsen and Molich's ten usability heuristics. The evaluation is a, "usability engineering method for finding the usability problems in a user interface design so that they can be attended to as part of an iterative design process" (Nielsen 1994). It is conducted using a small group of evaluators and is a quick, cheap, and efficient way for a product to be tested. Once the report for each heuristic is described, it will be assigned a number on the severity scale from zero to four. This ensures that the major problems are addressed.

The **Heuristics**

GARMIN

The usability heuristics used in this document originate from Nielsen and Molich's ten usability heuristics proposed in 1990. The ten heuristics are as follows:

1. **Visibility of system status** – The system should always keep users informed about what is going on in the system, through appropriate feedback within reasonable time.

2. **Match between system and real world** – The system should use the language of the user and use proper vocabulary and diction that is easily understood. Additionally, information should be presented in a natural and logical order and manner.

3. **User control and freedom** – The user should always have the opportunity to escape an unwanted state at all times without extended dialogue. Additionally, the software should support undo and redo functionality.

4. **Consistency and standards** – Users should not be presented with various words or actions that mean or do the same thing. Additionally, the UI should follow platform conventions and style guides for familiarity.

5. **Error prevention** – Eliminate conditions that are likely to cause user error and always present a user with a confirmation before completing an important action.

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 system to another.

7. **Flexibility and efficiency of use** - This is for more advanced users of a system. Provide accelerators or shortcuts for the expert user that can make interaction faster. Allow users to tailor frequent actions.

8. **Aesthetic and minimalist design** – Dialogue 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 the user's relative visibility. Additionally, the information should be presented in a visually appealing manner. Using visual hierarchy cues, the information should be laid out in a way where the use can focus on what's important and be engaged.

9. **Help users recognize diagnose, and recover from errors** – Error messages should be explained in plain language (not codes), precisely indicate the problem, and constructively suggest a solution.

10. **Help and documentation** – Though it is better for a system to be used without documentation, it is 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.

****The evaluation will be done in this order**

References: https://www.nngroup.com/articles/ten-usability-heuristics/

Severity **Scale**

The severity scale which will be used is as follows:

0 = Ok or Excellent: This is not a usability problem at all.

1 = Cosmetic Problem Only: Need not be fixed unless additional time is available on project.

2 = Minor Usability Problem: Fixing this should be given low priority.

3 = Major Usability Problem: Important to fix and should be given high priority.

4 = Usability Catastrophe: Imperative to fix before this product can be released.

Visibility of System Status

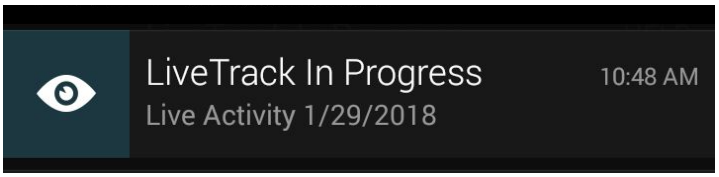
Severity

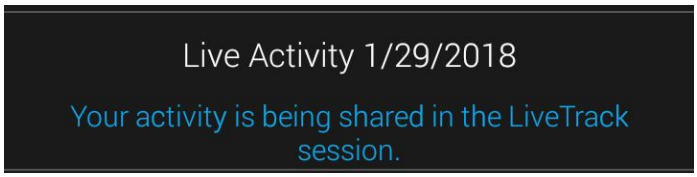
OK or Excellent	Cosmetic Problem	Minor	Major	Catastrophe
0	**1**	2	3	4

Description

The user is aware of the status of LiveTrack; whether it is started or stopped. The small blue text is the message that communicates this to the user. However, the text is very small and does not promote visual hierarchy. It does not appear important and could easily be missed if the user does not pay attention.

Examples





Possible Improvements

- Fade in/fade out popup
- More prominent styled text
- Sound/Voice
- Haptic Feedback/Vibration
- Confirmation (Are you sure you want to start?)
- Started page (Map, green check)

GARMIN

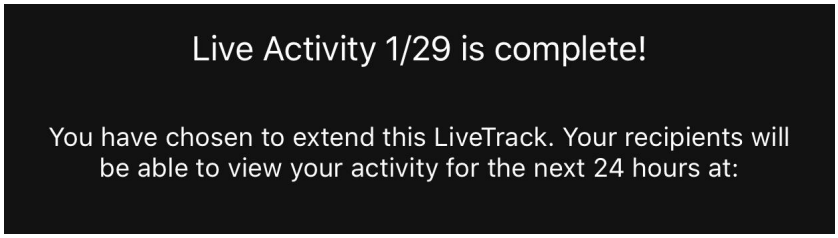
Match Between System and Real World

Severity

OK or Excellent	Cosmetic Problem	Minor	Major	Catastrophe
0	1	**2**	3	4

Description

The language used when interacting in this application is very natural and easy to understand. It uses very casual and conversational language that really aids in user understanding. However, if the user is using Auto-Start, the date for the name does not update on each activity.



Possible Improvements

- Ensure the default activity name contains the correct date.
  - Perhaps the type of activity could be in the name as well. (1/29 Run)
- Improve visual hierarchy cues
  - The example utilizes larger font and secondary, smaller font. However, the color is the same.

User Control and Freedom

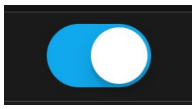
Severity

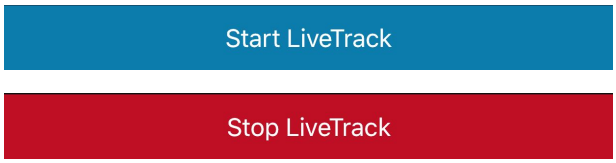
OK or Excellent	Cosmetic Problem	Minor	Major	Catastrophe
0	1	2	3	4

Description

It is very easy for users to recover from unwanted states. Most controls utilize a common UI toggle or a stop/start button. They are very understandable and allow the user to fix or undo their mistakes.

Examples





Possible Improvements

- N/A

Consistency and Standards

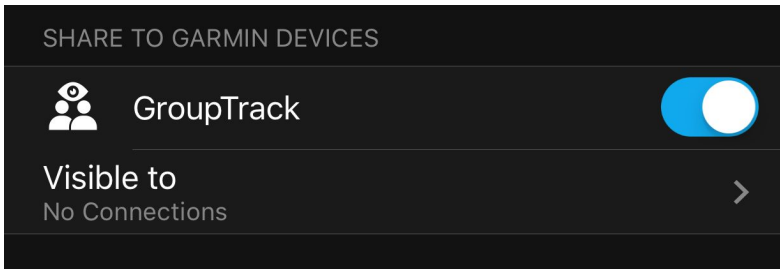
Severity

OK or Excellent	Cosmetic Problem	Minor	Major	Catastrophe
0	1	2	**3**	4

Description

Problem 1: The distinction between GroupTrack and LiveTrack is not entirely clear and this leads the user to wonder if this is the same feature as LiveTrack, and will it share the activity the same way.

Example



Problem 2: Setting up LiveTrack is very different than the setting up the app and the Garmin device for the first time. There is not a consistent set up process for different aspects of the Garmin Connect application.

Possible Improvements

- Include an easier onboarding process
    - Provide a description of LiveTrack and GroupTrack with visuals

Error Prevention

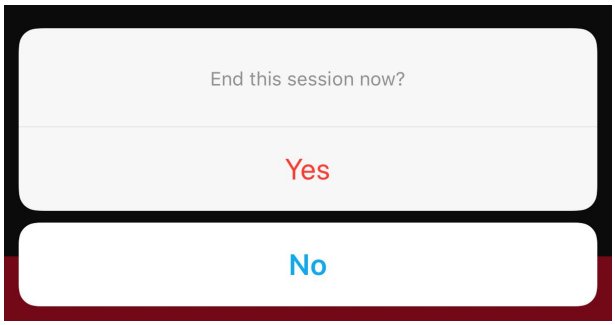
Severity

OK or Excellent	Cosmetic Problem	Minor	Major	Catastrophe
0	1	**2**	3	4

Description

There is no popup to ensure the user wants to turn on LiveTrack. However, there is a popup when attempting to end livetrack.

Examples



Possible Improvements

- Add a verification popup when “starting” live track.

Recognition Rather than Recall

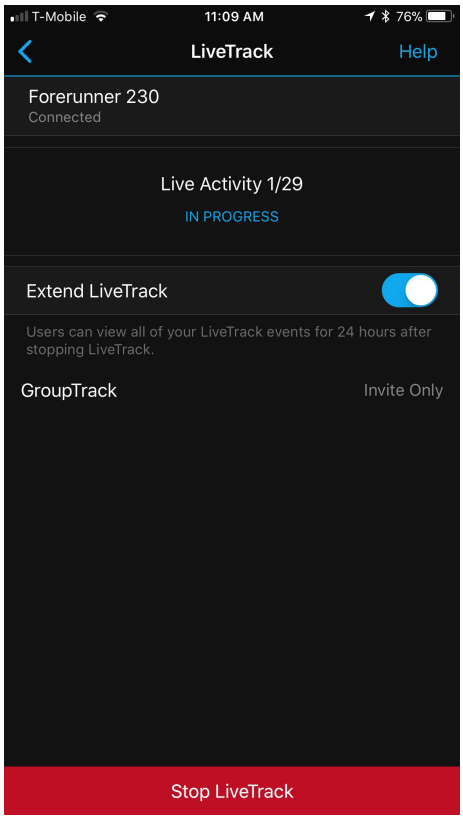
Severity

OK or Excellent	Cosmetic Problem	Minor	Major	Catastrophe
0	1	**2**	3	4

Description

The following page does not show who you are specifically sharing your activity with, or the outlets you are sharing it to.

Example



Possible Improvements

- Add a verification popup when “starting” live track.

Flexibility and Efficiency of Use

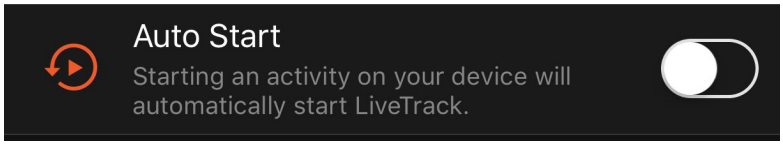
Severity

OK or Excellent	Cosmetic Problem	Minor	Major	Catastrophe
0	1	2	3	4

Description

The Auto Start feature allows users to shortcut having to stop and start LiveTrack, by having it start automatically.

Example



Possible Improvements

- N/A

Aesthetic and Minimalist Design

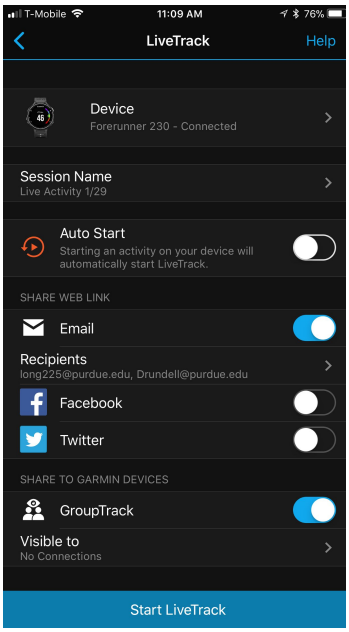
Severity

OK or Excellent	Cosmetic Problem	Minor	Major	Catastrophe
0	1	2	**3**	4

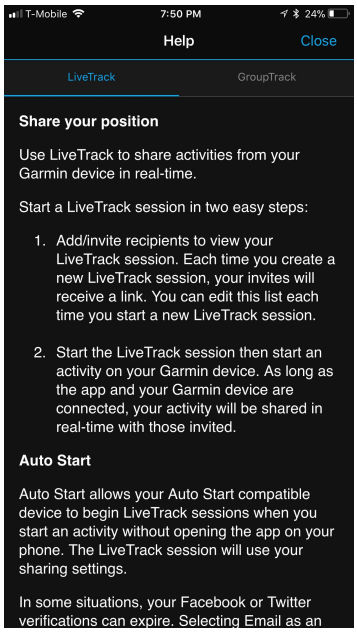
Description

The app is very minimal, and shows relevant information. However, it often times does not display enough information or shows too much textual based information.

Example



Not enough information



Too much textual information

Possible Improvements

- Onboarding process
- More visual help screen

Help Users Recognize, Diagnose, and Recover from Errors

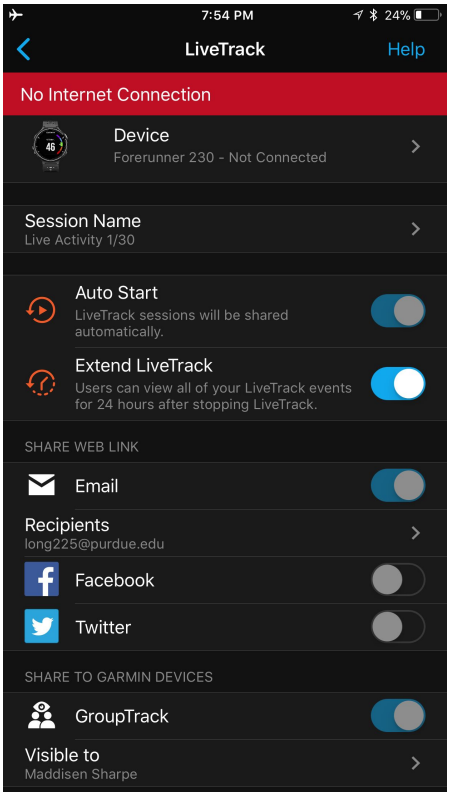
Severity

OK or Excellent	Cosmetic Problem	Minor	Major	Catastrophe
0	1	2	3	4

Description

The user is easily made aware of the issue through the use of natural language.

Example



Possible Improvements

- N/A

Help and Documentation

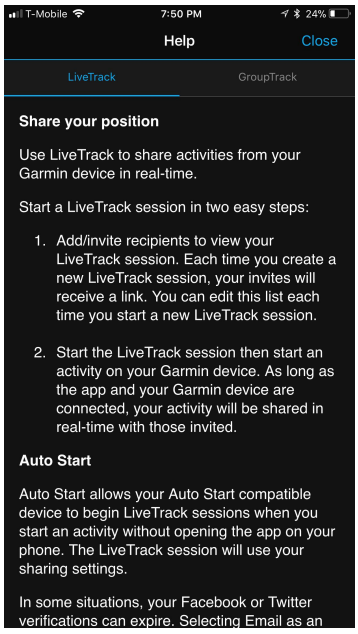
Severity

OK or Excellent	Cosmetic Problem	Minor	Major	Catastrophe
0	1	2	**3**	4

Description

The help page contains information that first time users need. This page states what the feature does, and how to use it. Users should not have to access a further help page; this should be a last resort. Without seeing this page or being exposed to external instruction, the user is not made aware of this information in any other manner. Additionally, when the user does utilize this page, it's very text heavy, which deters users from reading the information and enhances confusion.

Example



Possible Improvements

- Onboarding process
- Simplified instructions
- More visuals

There were two main areas we found that, specifically, require attention.

1. Lack of setup and onboarding process
2. Unclear information architecture

Lack of setup and onboarding process

The following heuristics pointed to the need for a setup and onboarding process:

- Consistency and Standards
- Aesthetic and Minimalist Design
- Help and Documentation

We believe these usability issues could be improved upon utilizing a setup and onboarding process because these violations all pertain to user understanding. The highest ranked violation was **help and documentation**. The help page is a major area of concern for us. All of the information that helps the user understand the feature and how to use it, is crammed onto one page in an all text format. We believe that this information can be simplified and visualized through an efficient, approachable onboarding process.

Unclear information architecture

The following heuristics pointed to the need for clearer information architecture:

- Match Between System and Real World
- Aesthetic and Minimalist Design
- Help and Documentation

Surprisingly, the areas that could benefit from a clearer information architecture, are very similar to the areas that could benefit from a simplified onboarding and setup process. On the lighter side, **match between system and real world** could be fixed through cosmetic updates and restyled text. However, in other areas, the information needs to be dissected and parsed for relevance. Additionally, it should be placed in order of encounter and importance. These design decisions will be supported by our research.


GARMIN®