Mayhem

Make a little Mayhem in your world

Team

Group Manager: Eli White
Design: Jabili Kaza & Jen Smith
Documentation: Meaghan Kjelland
Testing: Kyle Zemek

Problem and Solution Overview

Computers aren't just used for writing papers or browsing social networks -- they can interact with hardware peripherals and other software applications. While the average computer user views a computer as a product of the applications that exist on it, developers see a computer as a tool they can use to make anything happen. Mayhem is our approach to bring that mindset to those who love technology and want to use it to help simplify and automate aspects of their life.

Using Mayhem, you simply select an event and a reaction for that event to create a connection. An event can be anything from when motion is detected with your webcam, the weather changes, you get an email, etc. A reaction can similarly be anything from turning on your lights to going to the next song. Any event can be paired with any reaction allowing the user to effectively tell his computer what he wants it to do, without needing to know how to program!

Tasks

Task 1 (easy): You are watching a movie on your laptop and don't want to miss a minute of it, even if you have to get up to answer the phone. You know that Mayhem can talk to your webcam to sense when you are not in front of the screen. Set up your computer to automatically pause your movie when you are not there.

Task 2 (medium): You are constantly missing the bus because you forget to leave your house in time to make it to the bus stop. Set up your computer to notify you when your bus (207) is 10 minutes away from your stop.

Task 3 (hard): You are giving a PowerPoint presentation and would like to control the slides from your smartphone. Use Mayhem to allow you to use your phone to change to the next and previous slides in the presentation.

Interface Revision

After doing our user testing we found that there were a few things that we needed to revise on our interface.

One problem we saw is that people were surprised after they selected the event and reaction and it was magically put into the list and enabled. They didn't expect the connection to actually work. In order to fix this, we want to add a "success" tooltip that confirms the connection has been added. To avoid being overly annoying, the tooltip will also contain a checkbox to allow users to "never show this again" (figure 2).

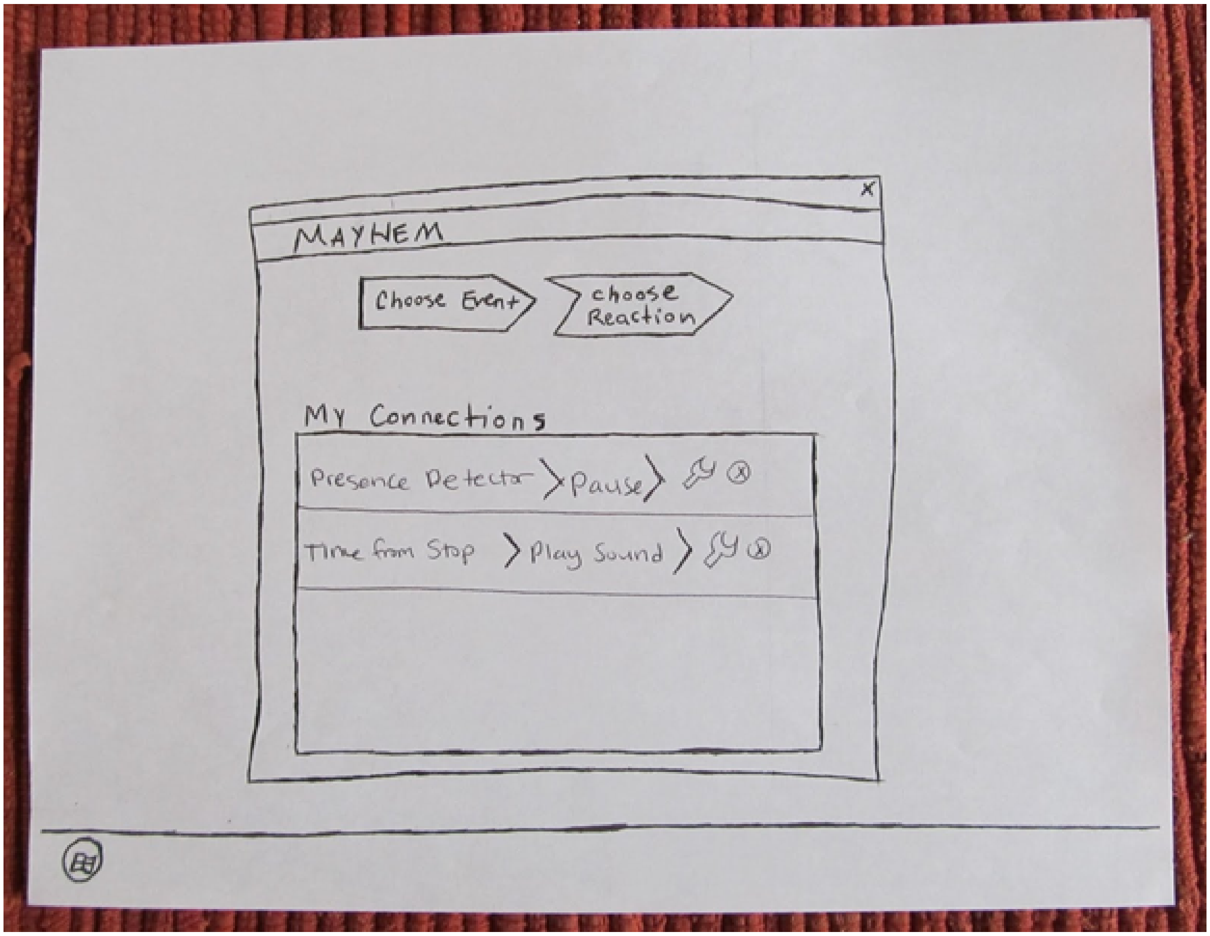
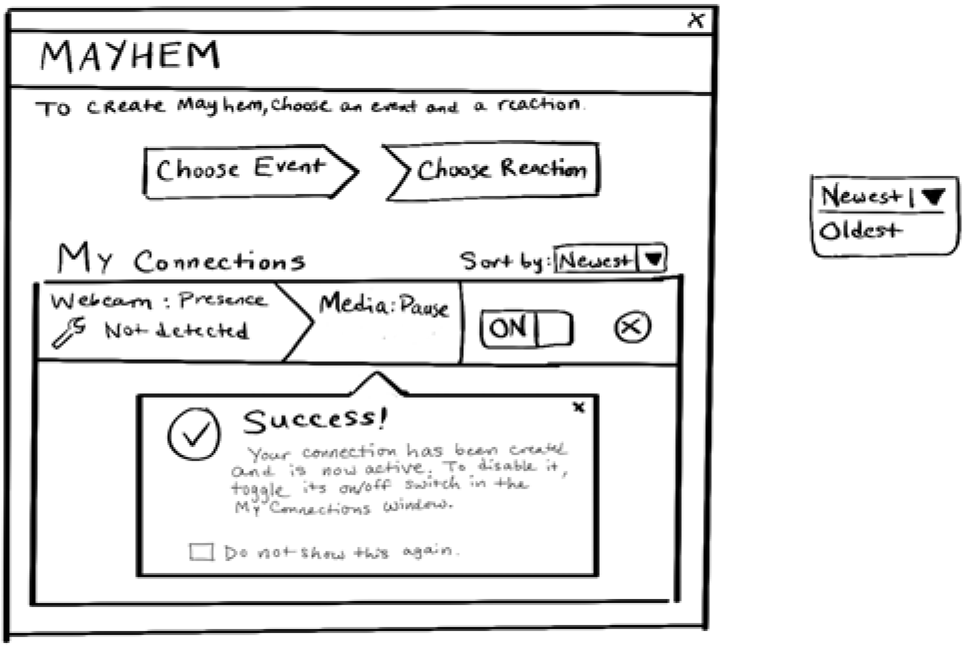


Figure 1. Original home screen from the low-fidelity prototype.



To create Mayhem, choose an event and a reaction.

Your connection has been createl and is now active. To disable it, toggle its on/off switch in the My Connections window.

□ Do not show this again.

Figure 2. Revised home screen with notification after a new connection has been created.

In the previous version of the interface, we had configuration menus for each connection that were accessible by clicking the wrench icon at the right side of the connection (figure 1). We intended that users would be able to enable or disable the connection, as well as perform any other configuration tasks, from this screen. However, in our tests, none of our participants clicked on this icon to open the menu. We decided that since there was only the on/off toggle in the configuration screen, we would just put it on the main screen where the wrench icon was (figure 2). This provided the added benefit of clearly identifying a connection's status without the user having to dig through menus to find it.

We also added helpful messaging to streamline the learning process and help orient the user. For example, when we asked people to complete a task, they didn't know where to begin. We are adding a simple line of explanatory text on the main screen (figure 2), as well as short descriptions beneath each item in the event and reaction lists that indicate what each module does. We will also display configuration information in the connection lists.

We are reorganizing the event list by taking all the available modules out of their parent packages. The names of the packages were distracting and it was confusing what the terms "package" and "module" meant, while not being a useful thing for users to know. This will also improve usability, since users will now be able to find events and reactions without needing to know what package it came from.

We also drastically modified the experience of pairing phones and adding buttons. In our tests, users consistently overlooked the fact that buttons wouldn't be created on their phones until they scanned the QR code. We realized that we needed to provide a better call to action to prevent users from being able to set up phone button events without ever actually pairing their phones. In our revised interface, if you are opening a phone event configuration dialogue for the first time, then the save button is disabled. Instead of immediately showing the QR code, there is a button that says “Pair with Phone”. When you click on that, another window pops up with some text explaining what you need to do on the phone, and presents the QR code (figure 4). Clicking “Done” on that window will take the user back to the previous configuration screen, where the save button is now enabled.

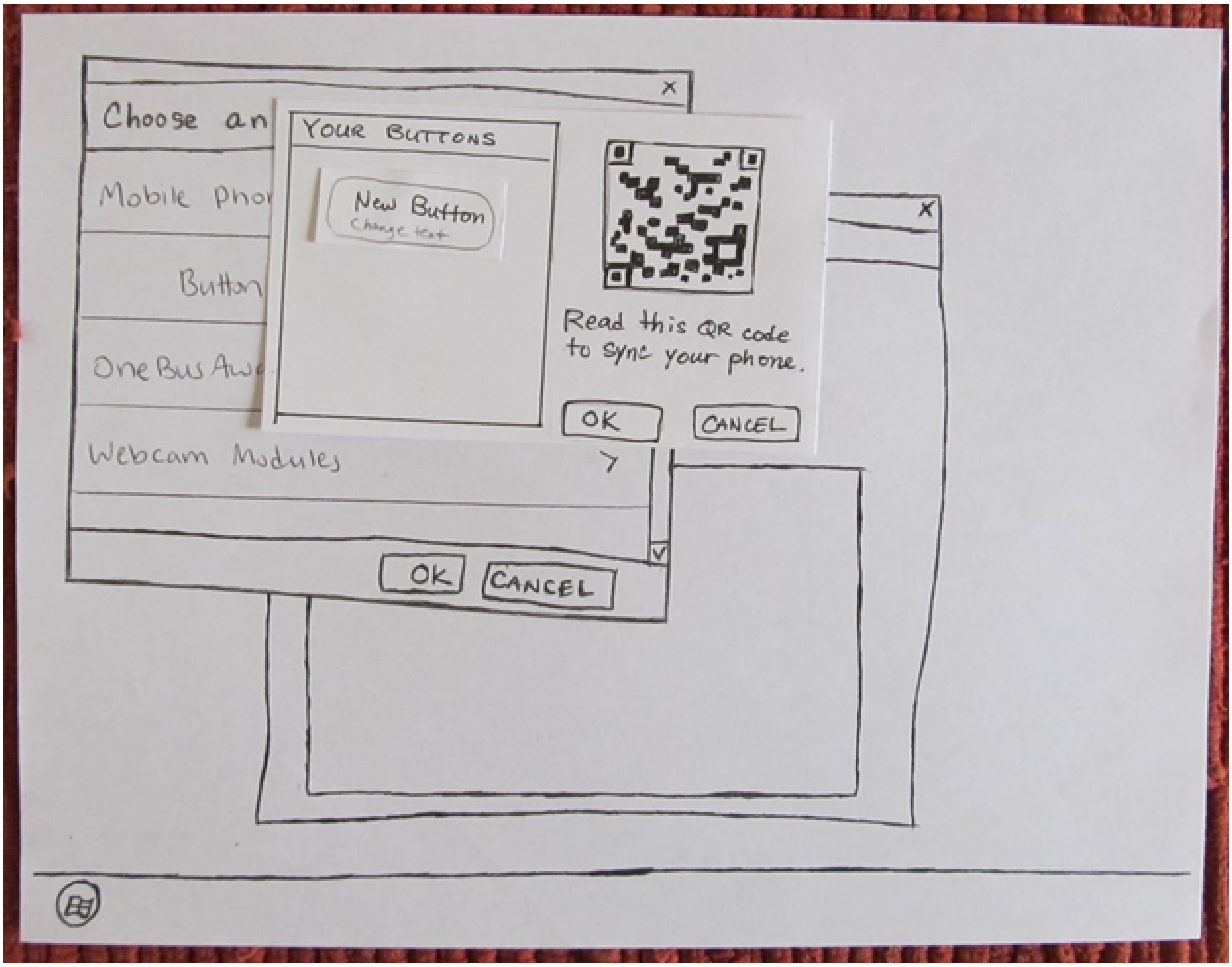


Figure 3. Original interface for creating a phone button. Users did not understand that they needed to scan the QR code in order to sync the button to their phone.

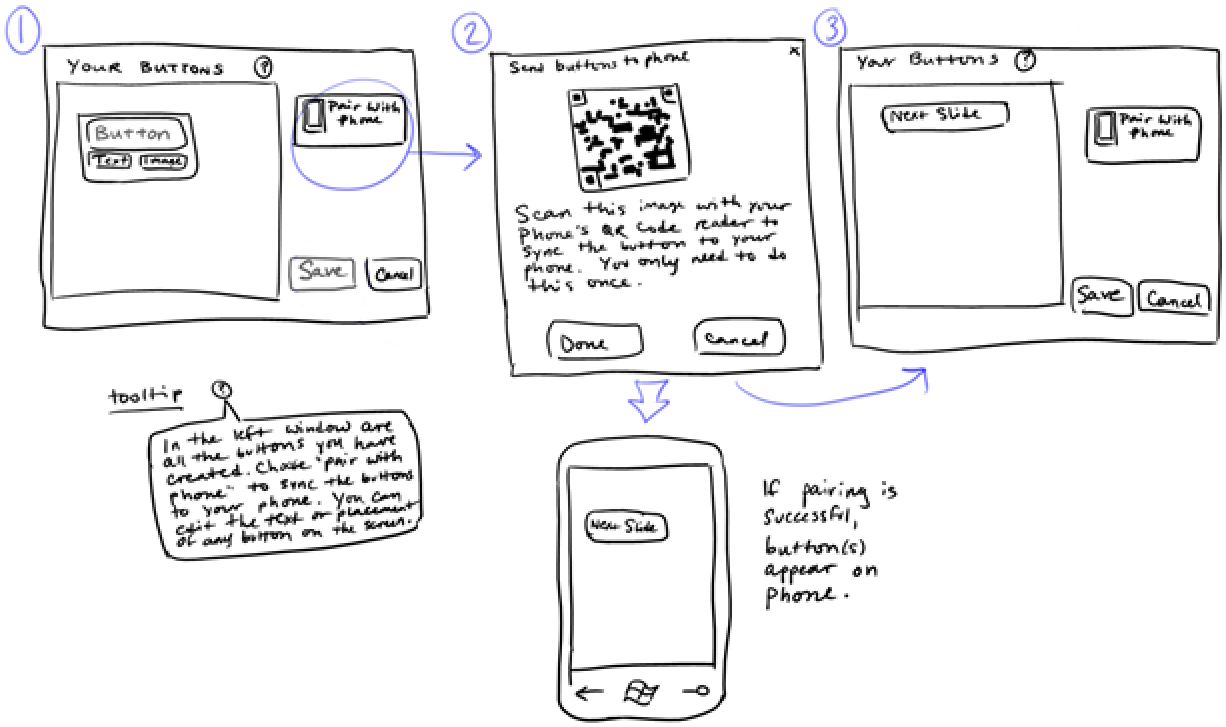

1
YOUR BUTTONS ?
Button
Text
Image
Pair With Phone
Save
Cancel
2
Send buttons to phone
Scan this image with your phone's QR code reader to sync the button to your phone. You only need to do this once.
Done
Cancel
3
Your Buttons ?
Next Slide
Pair With Phone
Save
Cancel
tooltip ?
In the left window are all the buttons you have created. Chose 'pair with phone' to sync the buttons to your phone. You can edit the text or placement of any button on the screen.
New Slide
If pairing is successful, button(s) appear on phone.

Figure 4. Revised flow of creating a mobile phone button and syncing it to a phone.

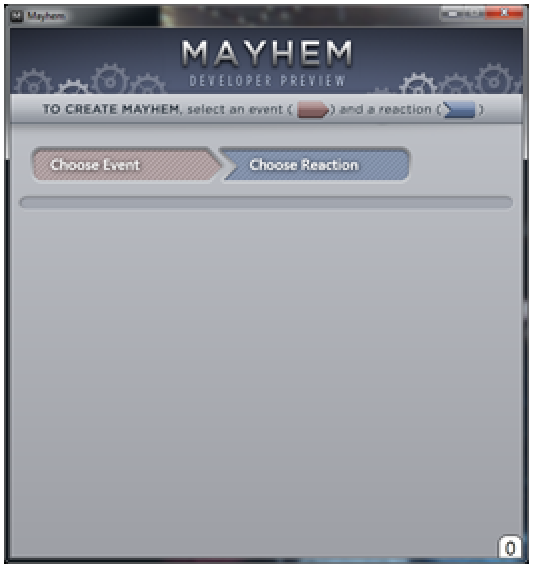
Prototype Overview

Overview of implementation

Since much of the core functionality of this project has previously been implemented, we choose to make a fully functional version of our interface so that we could give an informative demonstration. Thus, our prototype is implemented as an actual executable application. It is written using the .NET framework, and can be run on Windows computers. During our low-fidelity prototype user testing, we found that many people got caught up in not understanding what Mayhem was actually doing. In the future, we would like to do more user testing with our high-fidelity prototype. Having the functionality actually work during our next round of user testing will help us to see how quickly users will understand Mayhem when they can see that what they are doing actually translates into something in the real world.

Scenarios for 3 tasks

Task 1:

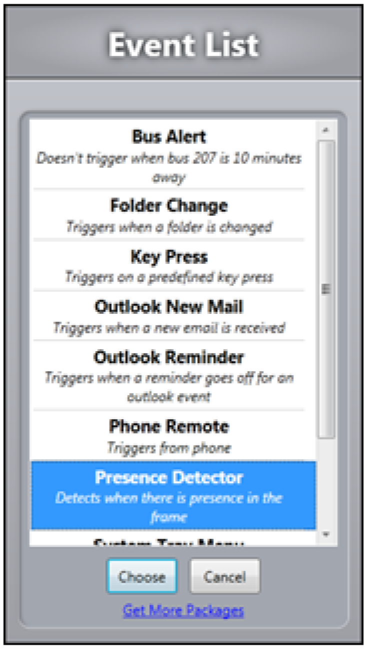


TO CREATE MAYHEM, select an event ( ) and a reaction ( )

Figure 5. The start screen of Mayhem

1) Select the red-grey “Choose Event” button near the top of the window in figure 5.

2) Navigate down and select “Presence Detector” from the scrolling list in the popover “Event List” window.



Bus Alert Doesn't trigger when bus 207 is 10 minutes away

Folder Change Triggers when a folder is changed

Key Press Triggers on a predefined key press

Outlook New Mail Triggers when a new email is received

Outlook Reminder Triggers when a reminder goes off for an outlook event

Phone Remote Triggers from phone

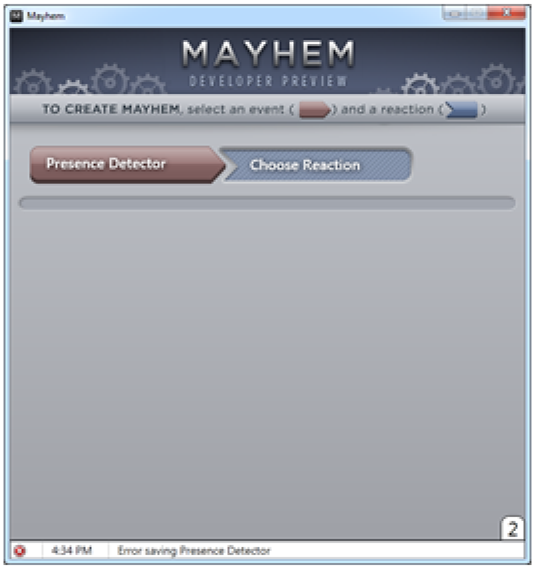
Presence Detector Detects when there is presence in the frame

Figure 6. The list of events

3) Click the “Choose” button near the bottom of the popover “Event List” window in figure 6.

4) In the “Presence Detector” popover select any region in the box near the top, scroll to any sensitivity on the scroll bar, and select any attached camera (or nothing if there are no cameras attached) in the dropdown menu.

5) Click the “Save” button near the bottom of the “Presence Detector” popover in figure 6.



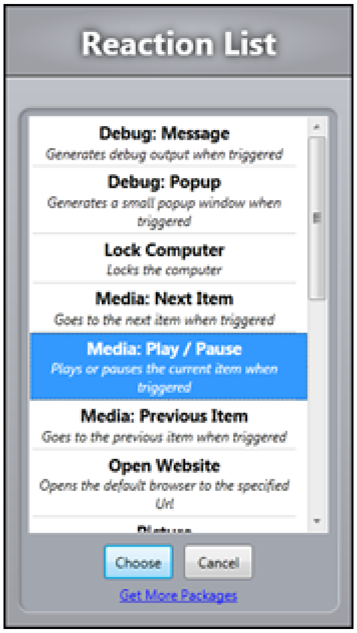
TO CREATE MAYHEM, select an event ( ) and a reaction ( )

4:34 PM Error saving Presence Detector

Figure 7. The start screen with presence detector event chosen

6) Select the blue-grey “Choose Reaction” button near the top of the window in figure 7.

7) Navigate down and select “Media: Play/Pause” from the scrolling list in the popover “Reaction List” window in figure 8.



Debug: Message
Generates debug output when triggered

Debug: Popup
Generates a small popup window when triggered

Lock Computer
Locks the computer

Media: Next Item
Goes to the next item when triggered

Media: Play / Pause
Plays or pauses the current item when triggered

Media: Previous Item
Goes to the previous item when triggered

Open Website
Opens the default browser to the specified Url

Picture

Figure 8. The list of reactions

8) Click the “Choose” button near the bottom of the popover “Reaction List” window in figure 8.

9) Select the toggle option currently labeled “OFF” located next to reaction pair like the ones shown in figure 10 in order to turn the connection on.

10) The connection should now be illuminated in red and blue as well as currently active.

Task 2:

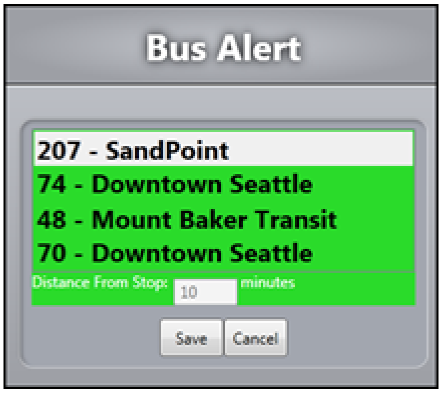
1) Select the red-grey “Choose Event” button near the top of the window in figure 5.

2) Navigate down and select “Bus Alert” from the scrolling list in the popover “Event List” window in figure 6.

3) Click the “Choose” button near the bottom of the popover “Event List” window in figure 6.

4) In the “Bus Alert” popover (figure 9) select the option labeled “207 – SandPoint” from the provided list.

5) Type 10 into the minutes box.



Bus Alert

207 - SandPoint
74 - Downtown Seattle
48 - Mount Baker Transit
70 - Downtown Seattle

Distance From Stop: 10 minutes

Figure 9. The bus alerts configuration screen

6) Click the “Save” button near the bottom of the “Bus Alert” popover in figure 6.

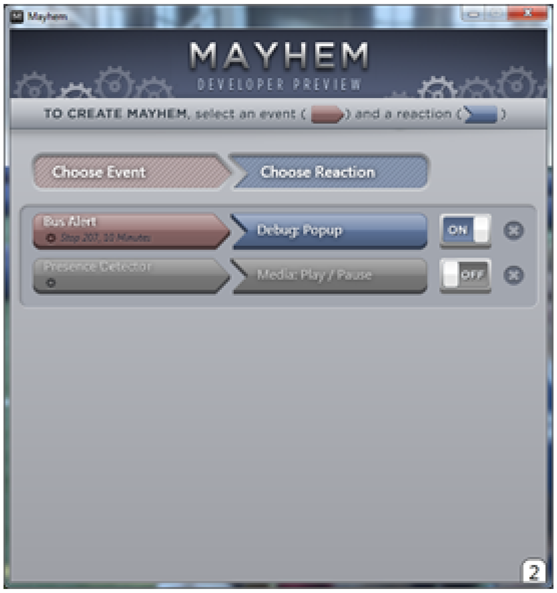
7) Select the blue-grey “Choose Reaction” button near the top of the window as shown in figure 7.

8) Navigate down and select “Debug: Popup” from the scrolling list in the popover “Reaction List” window in figure 8.

9) Click the “Choose” button near the bottom of the popover “Reaction List” window in figure 8.

10) Select the toggle option currently labeled “OFF” located next to reaction pair in order to turn the connection on in figure 10.

11) The connection should now be illuminated in red and blue as well as currently active, as seen in figure 10.



TO CREATE MAYHEM, select an event ( ) and a reaction ( )

Bus Alert — Stop 207, 10 Minutes → Debug: Popup — ON

Presence Detector → Media: Play / Pause — OFF

Figure 10. The start screen of Mayhem with the bus alerts task completed

Task 3:

1) Select the red-grey “Choose Event” button near the top of the window in figure 5.

2) Navigate down and select “Phone Remote” from the scrolling list in the popover “Event List” window in figure 6.

3) Click the “Choose” button near the bottom of the popover “Event List” window.

4) In the “Phone Remote” hover over the button labeled button in the center of the black screen and surrounded by a pulsing white border in figure 11.

5) Click on the button that reads “Text” that appears next to the button in figure 11.

6) Highlight the text reading “Button” and type “Previous Slide”.

7) Drag the Button now labeled “Previous Slide” to the top left of the black box.

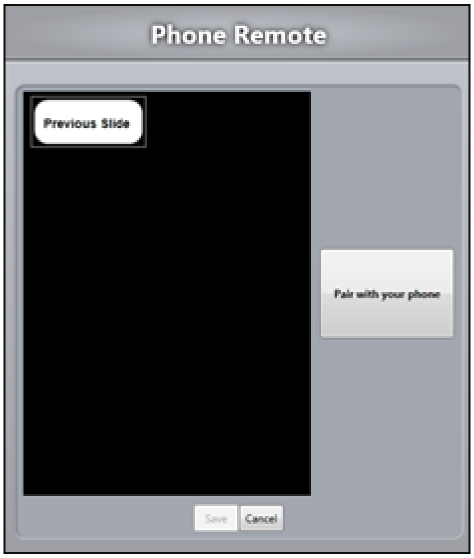
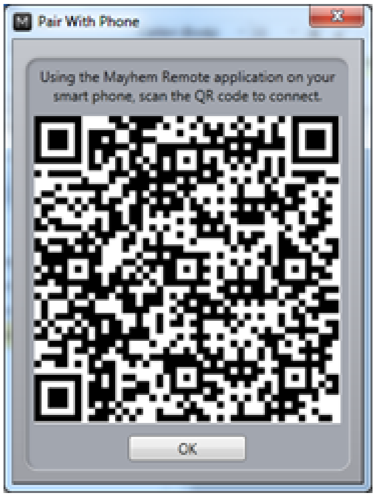


Figure 11. The phone remote configuration screen

8) Click on the “pair with your phone” button on the right side of the window in figure 11.

9) If you have a phone with the Mayhem phone remote app installed pair the device by scanning the QR code in figure 12 now otherwise ignore this step.



Using the Mayhem Remote application on your smart phone, scan the QR code to connect.

Figure 12. The phone pairing screen

10) Click the “OK” button near the bottom of the “Pair With Phone” popover window in figure 12.

11) Click the “Save” button near the bottom of the “Phone Remote” popover in figure 11.

12) Select the blue-grey “Choose Reaction” button near the top of the window in figure 10.

13) Navigate down and select “PowerPoint: Last Slide” from the scrolling list in the popover “Reaction List” window in figure 8.

14) Click the “Choose” button near the bottom of the popover “Reaction List” window in figure 8.

15) Open your powerpoint presentation.

16) Select the toggle option in figure 13 currently labeled “OFF” located next to reaction pair in order to turn the connection on.

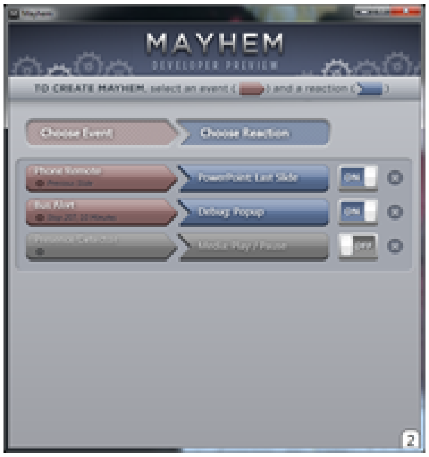


Figure 13. The start screen of Mayhem with the PowerPoint last slide connection completed

17) The connection should now be illuminated in red and blue as well as currently active, as shown in figure 13.

18) Select the red-grey “Choose Event” button near the top of the window in figure 13.

19) Navigate down and select “Phone Remote” from the scrolling list in the popover “Event List” window in figure 6.

20) Click the “Choose” button near the bottom of the popover “Event List” window in figure 6.

21) In the “Phone Remote” hover over the button labeled button in the center of the black screen and surrounded by a pulsing white border in figure 14.

22) Click on the button that reads “Text” that appears next to the button in figure 14.

23) Highlight the text reading “Button” and type “Next Slide”.

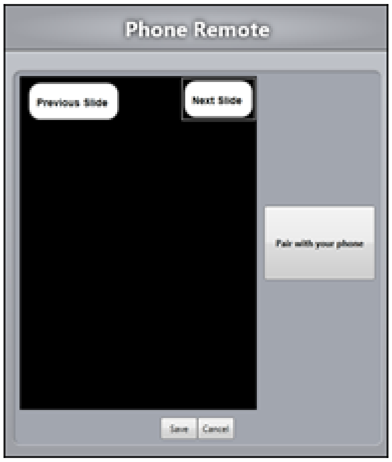
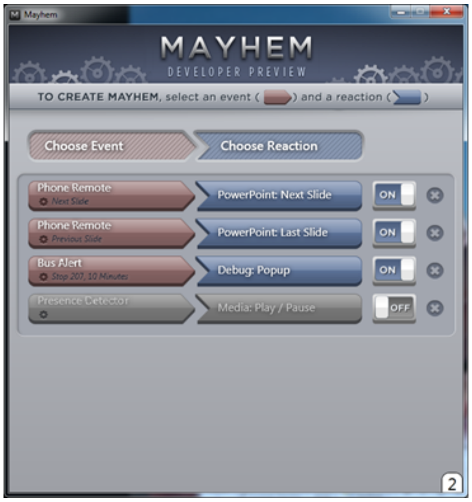


Figure 14. The phone configuration screen

24) Drag the Button now labeled “Next Slide” to the top right of the black box in figure 14.
25) Click the “Save” button near the bottom of the “Phone Remote” popover in figure 14.
26) Select the blue-grey “Choose Reaction” button near the top of the window in figure 13.
27) Navigate down and select “PowerPoint: Next Slide” from the scrolling list in the popover “Reaction List” window in figure 8.
28) Click the “Choose” button near the bottom of the popover “Reaction List” window in figure 8.
29) Open your PowerPoint presentation
30) Select the toggle option currently labeled “OFF” located next to reaction pair in figure 15 in order to turn the connection on.
31) The connection should now be illuminated in red and blue as well as currently active, as shown in figure 15.



TO CREATE MAYHEM, select an event ( ) and a reaction ( )

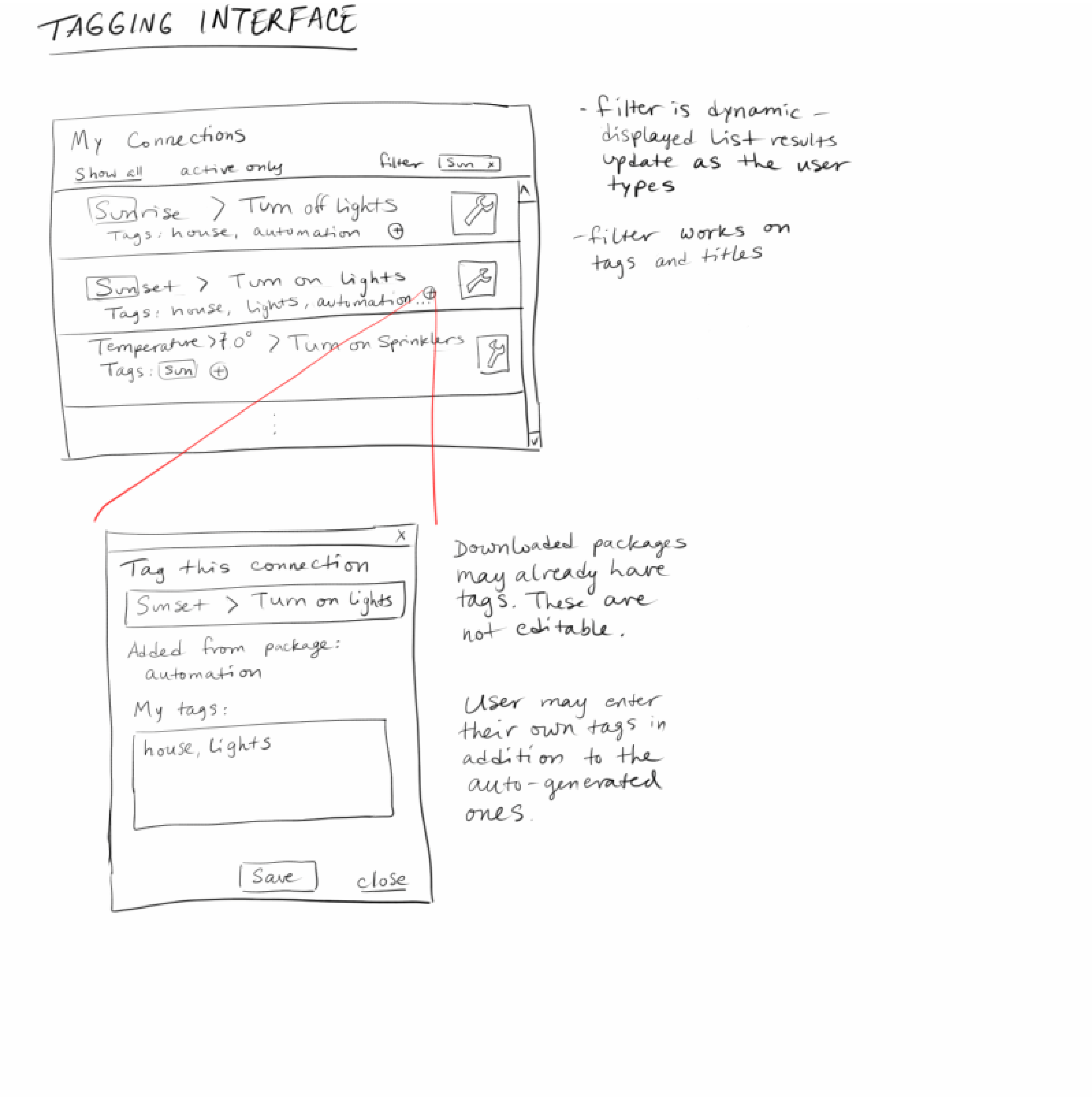
Phone Remote Next Slide	PowerPoint: Next Slide	ON
Phone Remote Previous Slide	PowerPoint: Last Slide	ON
Bus Alert Stop 207, 10 Minutes	Debug: Popup	ON
Presence Detector	Media: Play / Pause	OFF

Figure 15. The Mayhem home screen with the PowerPoint connections completed

Tools Used

The Mayhem application is written in C# and WPF, using the .NET framework. This was necessary because the core application code had already been written in C#. Developing with an actual programming language is incredibly powerful because it enabled us to create a prototype that actually works. However, working with an already-existing code base introduced some problems. For example, the presentation language isn't as nimble or flexible as, say, HTML and javascript, so we didn't have complete freedom to implement some of the functionality we originally planned. We also had to be much more restrained in the design, simply because we had to take into account any affects that the existing back-end code would have on the proposed front-end changes.

Unimplemented Interface Functionality



- filter is dynamic - displayed list results update as the user types

- filter works on tags and titles

Downloaded packages may already have tags. These are not editable.

User may enter their own tags in addition to the auto-generated ones.

Figure 16. Sketch of dynamically filtering a connection list, with tags.

We chose not to implement many features which we had originally wanted to include. One of the main problems we foresee for our users is that it will be difficult for them to search through a large number of events, reactions, and connections. This is a problem that we had originally wanted to solve during this course, however, we made the decision before our user testing to simply implement the basic features of Mayhem in order to ensure that the core functionality was working optimally. We had many ideas, but would have implemented custom folders (figure 17) and dynamic filtering (figure 16) had we had the time. We also wanted to allow users to custom sort their lists of connections by most/least new, as seen in the dropdown sketch in figure 2. This was left out because we decided that given the relatively low number of modules currently available, most early adopters of Mayhem won't be creating enough connections to need to sort them. We decided that this functionality would be better suited for a final release version rather than the prototype.

We were also forced to leave out the tooltip that is supposed to display when a connection is completed (figure 2), mainly because the framework we were using didn't allow us to implement it the way we wanted. Instead we decided to leave it out for now, and work on a better messaging solution for a future design iteration.

In addition, not all of the functionality in the prototype is working to its full extent, and users of our interactive prototype should note that they must have the corresponding hardware and software installed on their computer to use each event or reaction. An example is the presence detector event: the event's function is not currently implemented correctly, so it is actually doing motion detection. Furthermore, you must have a motion detector on your computer that is open for the event to be selected.

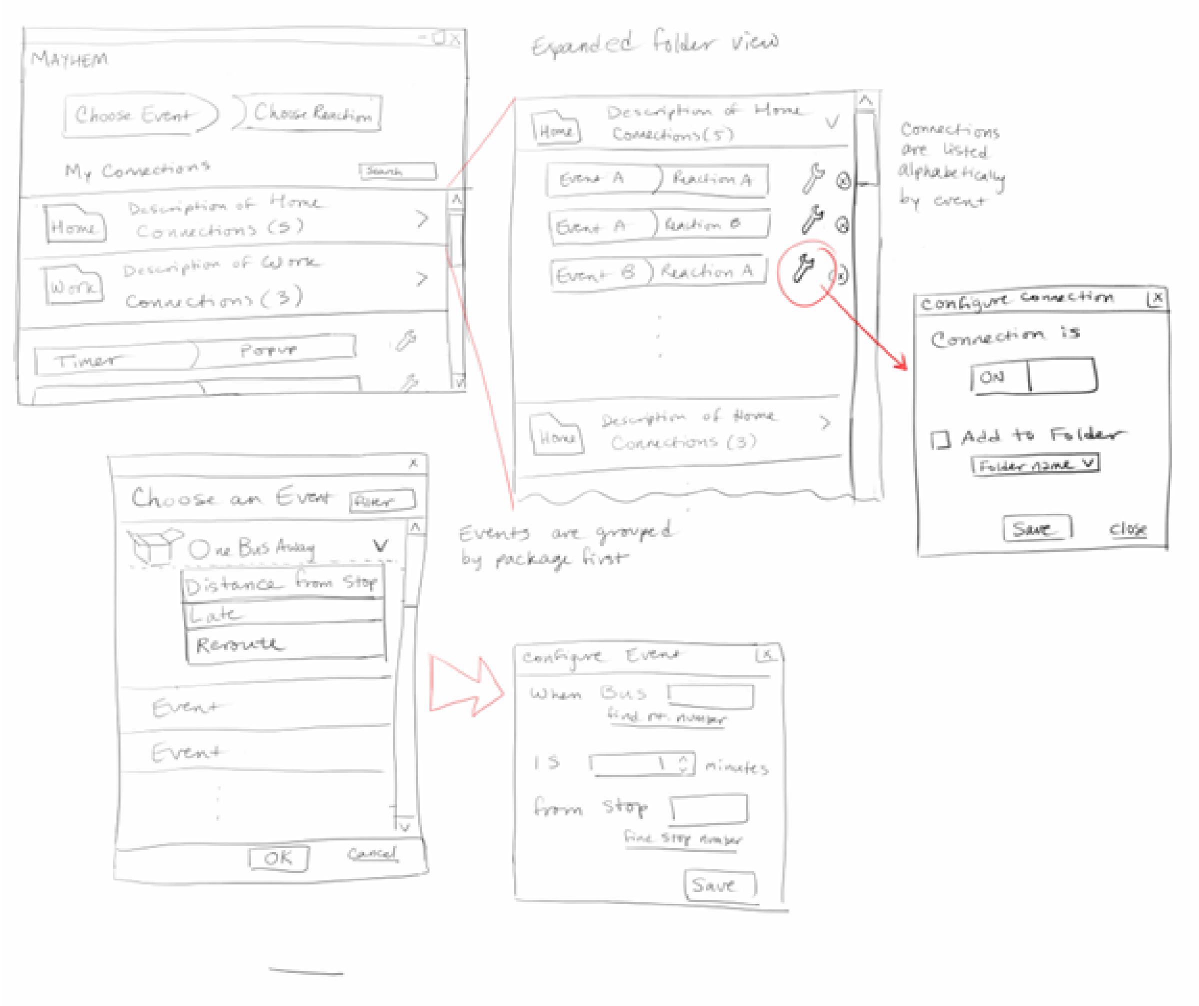


Figure 17. Grouping events by folders