

Mayhem

Make a little Mayhem in your world.

Team

Group Manager - Eli White

Documentation - Meaghan Kjelland

Design - Jabili Kaza & Jen Smith

Testing - Kyle Zemek

Problem and Solution Overview

Most people who aren't developers don't think of their computers as a tool that they truly control. Instead, they are reliant on downloading apps, built by unknown developers, in the hope that it can solve the problem at hand. With Mayhem, we want to empower users with minimal backgrounds in technology to be able to fully control their computers and perform exactly the tasks they desire. We aim to solve this problem by providing a simple interface that connects events, in the form of both tangible and networked services, to reactions, which can range from home automation to tasks that take on a purely digital form. The application itself is designed to solve the problem of technological barriers. We want everyday users to be able to use computers to accomplish the tasks they would otherwise have to manually complete every day. If the user can define criteria and specify a desired outcome we want to make it easy for them to automate the process, thus making creativity the sole barrier. Our interface will allow users to quickly select an event, pair it with a reaction, and save it as an application in the practical sense of the term.

Task Analysis

Who is going to use the system?

We want to create a system that is accessible to everyone. It will be especially targeted towards those who do not have experience with programming or computer science. This system should be readily operable by the average consumer: someone who owns a computer but is not concerned with its inner workings. The ideal implementation should appeal to anyone who wants to utilize their computer to simplify or automate their daily tasks.

What tasks do they now perform?

Currently many people are not automating things because they do the technical know-how to do so. They apply a repetitious manual approach rather than automating those things that have clear triggers and desired outcomes. Anything that can be boiled down to mindless behavior falls within the realm of targeted tasks.

What tasks are desired?

Automating tasks would mean that users would not have to manually perform them anymore, but would have to use our application to set up each automation once. Consequently, the new tasks would be the user understanding something can be automated and setting up a connection between an event and a reaction. It is on the user to recognize their repetitious behavior, but it is on the application to automate the task.

How are the tasks learned?

Any user should be able to quickly set up connections which automate tasks; they would have

to learn to use the system, but should be able to do so very quickly and with no prior knowledge.

Where are the tasks performed?

The tasks (that is, the actual pairing of events and reactions) will be performed on the user's computer. However, once set up, the physical execution of the automation can occur on any device that can be connected to another. Some examples are on the user's computer desktop or smartphone, inside the user's home, or at the user's workplace.

What's the relationship between customer and data?

The customer needs to be able to find their events and reactions from a list of thousands depending on how extensively the user engages with the system. The user must also be able to find the connections that they have set up if they want to change or delete them. However, the customer will be shielded from the implementation details of the connections, to keep it very easy to understand. The user should also be able to easily acquire new event/reaction modules if they are unable to find an appropriate one in their existing collection.

What other tools do the customers have?

Customers will need to be able to think up the connections that they want; they could get these ideas from friends. They will be able to search for and download new event and reaction modules from a centralized marketplace. In some cases, they will also need to have the physical hardware required to make the connections: for example, sensors for home automation.

How do the customers communicate with each other?

Customers can tell each other what connections they have set up, and inspire other customers to create new connections themselves. Highly tech-savvy customers will be able to develop their own event/reaction modules and share them via the marketplace, but this will be kept separate from the desktop application design.

How often are the tasks performed?

The task must be performed only once for each connection that the user wants to set up. How often the user sets up tasks is entirely up to them: they could automate thousands of tasks, or they could automate only one. Each event/reaction pair will then automatically take place for as long as the user keeps the connection active.

What are the time constraints on the tasks?

Users must be able to perform tasks very quickly, if it takes too long to set up a connection, they will get frustrated and might give up and stop using the system. Once a connection is set up, the triggers between events and reactions are instantaneous.

What happens when things go wrong?

If a user cannot quickly and easily understand the system, they will probably stop using it.

Simply finding what they're looking for could be a huge challenge for a user with a lot of data to search through. This is why usability and simplicity are so important in our design.

However, the system will still need to account for several scenarios which could confuse or complicate the user's experience. There will need to be some way to test or simulate a connection that is based on an event outside of the user's control. For example, a user who sets the outdoor lights to come on at sunrise needs to be sure that the system will actually perform when the sun rises. This could be accomplished by showing the user a "simulate sunrise"

button and turning the house lights on when the button is clicked. Also, if something goes wrong with one of the physical pieces of hardware that enables a connection, the user will need to be notified somehow. One way this could be accomplished is by displaying a status message inside the application.

Tasks

Pause your movie when you leave your computer

When a user is watching a movie on his laptop, he can use his webcam as a motion sensor to recognize if he gets up. By connecting the event of leaving the laptop with the reaction of pausing the media player, the user will no longer have to worry about missing vital scenes in a movie should he forget to manually pause the media player.

Display a notification on the Windows desktop when a specific bus is 10 minutes away

A user could connect the event of One Bus Away saying her bus is 10 minutes away to the reaction of an audible notification popping up on her computer screen. This reminder would not take any extra effort on the user's part, but would ensure that she never misses that particular bus.

Use a smartphone to operate a PowerPoint presentation

PowerPoint presentations are a convenient public speaking aid, but the presenter may not always have access to an effective means of controlling the presentation, and thus be tied to the computer keyboard by having to manually press a key whenever he or she wants to advance to the next slide. Mayhem will allow a user to configure a connection between their phone and PowerPoint presentation that can always be active, regardless of location. This will enable the presenter to walk around at will without being tethered to the computer, enabling them to give more engaging presentations.

Storyboards

Interface 1: Tabs and Tagging

This design uses a tabbed interface to allow the user to easily search for and switch between events, reactions, and connections. It is highly reminiscent of a web application, utilizing web 2.0 functionality such as tagging, drag and drop, and dynamic list filtering.

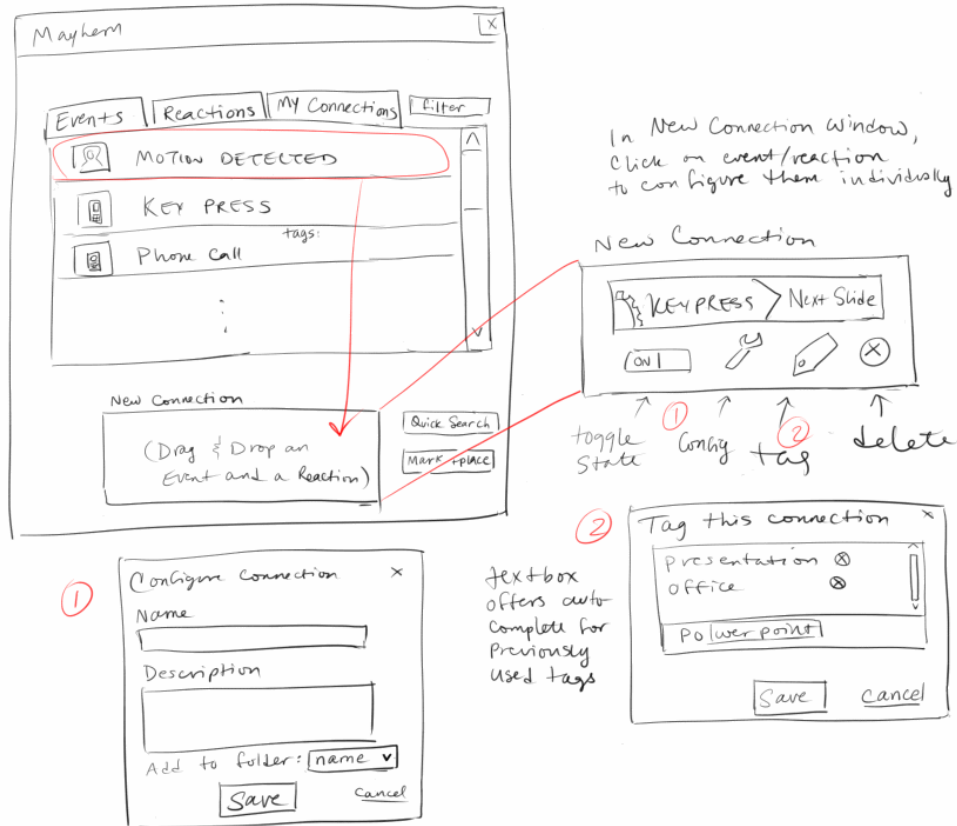


Figure 1A.

Events, reactions, and connections can all be given custom tags, enabling the user to tailor list organization in a way that makes the most sense to them. From the home screen, a “Quick Search” button launches a tag cloud showing a visual representation of all of the user’s tags. Each tag in the cloud is clickable and launches a list of events and reactions associated with the given tag.

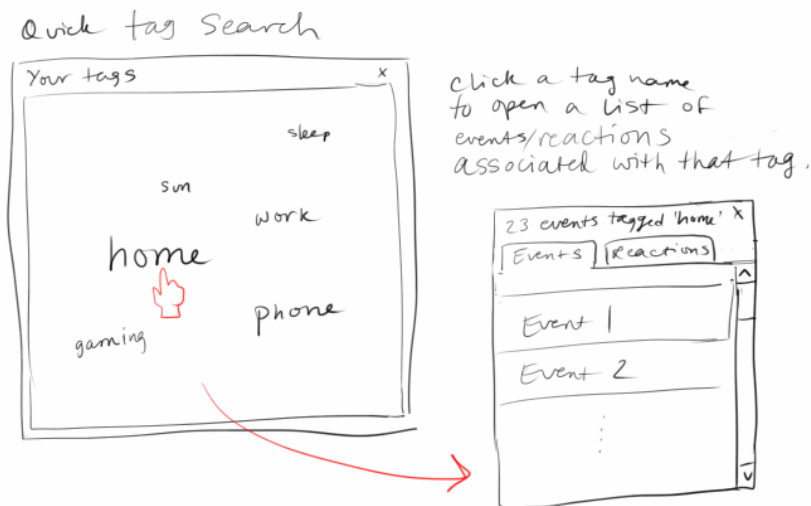
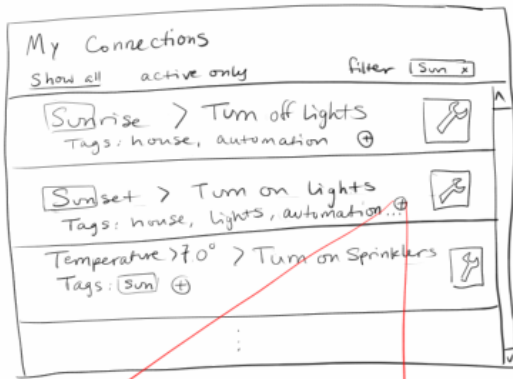


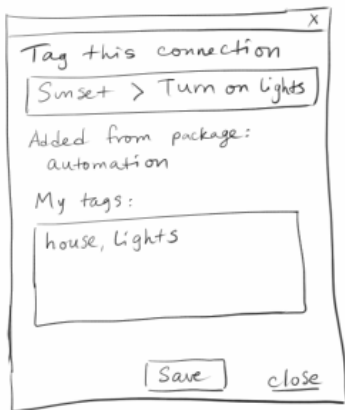
Figure 1B.

TAGGING INTERFACE



- 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 1C.

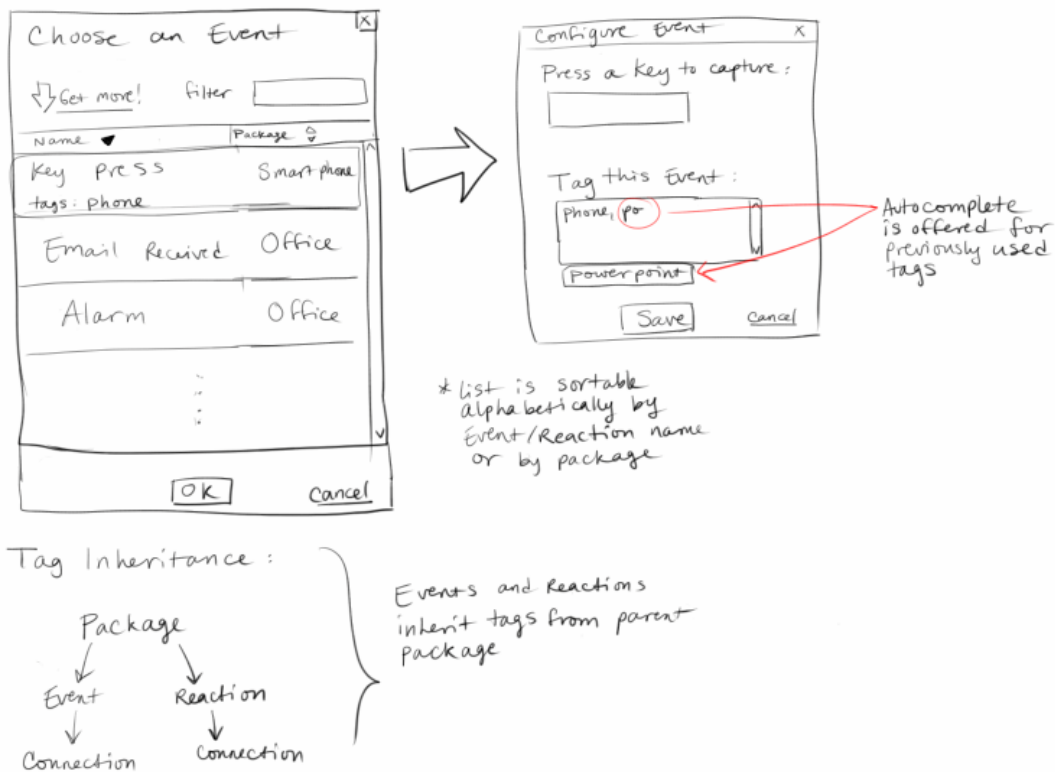


Figure 1D.

Interface 2: Folder View

Event and reaction modules can be downloaded as packages of multiple events and reactions related to a specific theme or application. This interface extends that idea by using a folder-based organization system similar to the file explorer on the user's desktop computer. This design aims to reduce the user's learning curve by using a paradigm the user is already familiar with. Users can create folders to sort their connections into, thus keeping things as loose or as organized as they want.

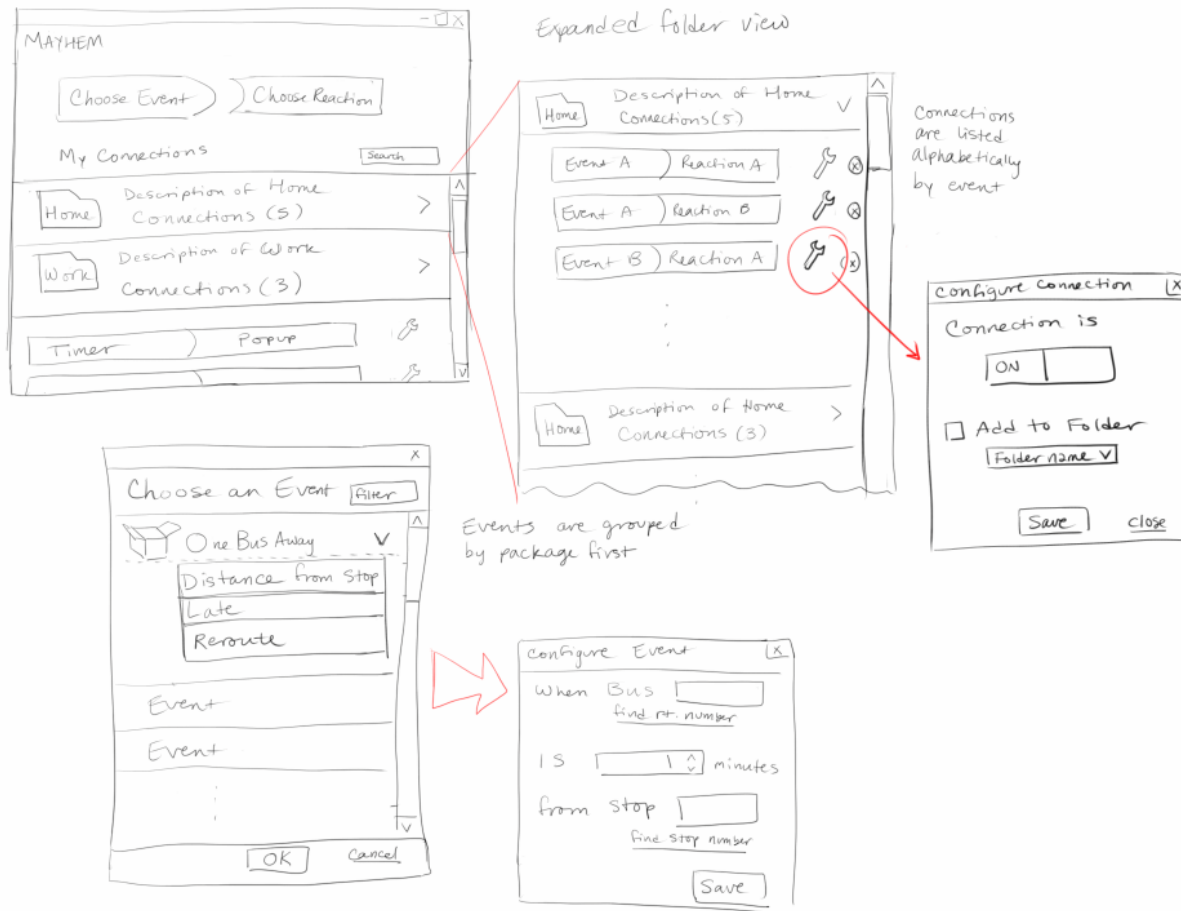


Figure 2.

Interface 3: Visual and touch-friendly

This design places high importance on visual elements to help users quickly find what they're looking for. Important interface elements are large, to call attention to them in the design. This design would be well suited to run on a touchscreen device.

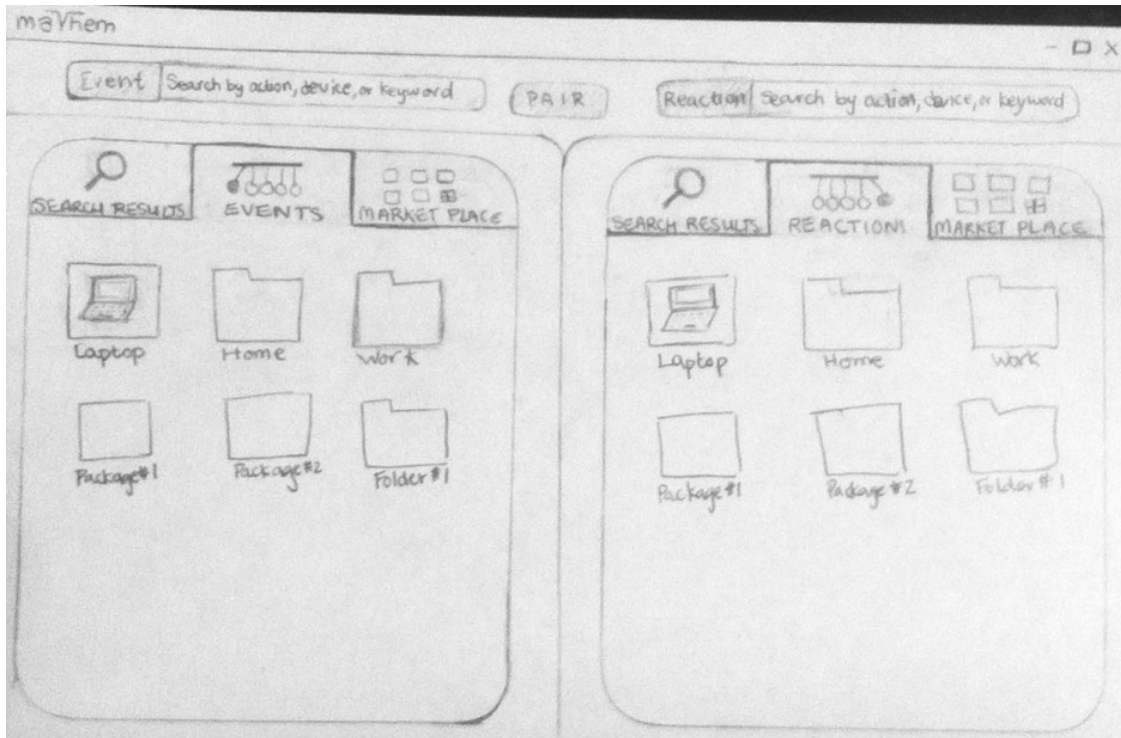


Figure 3A.

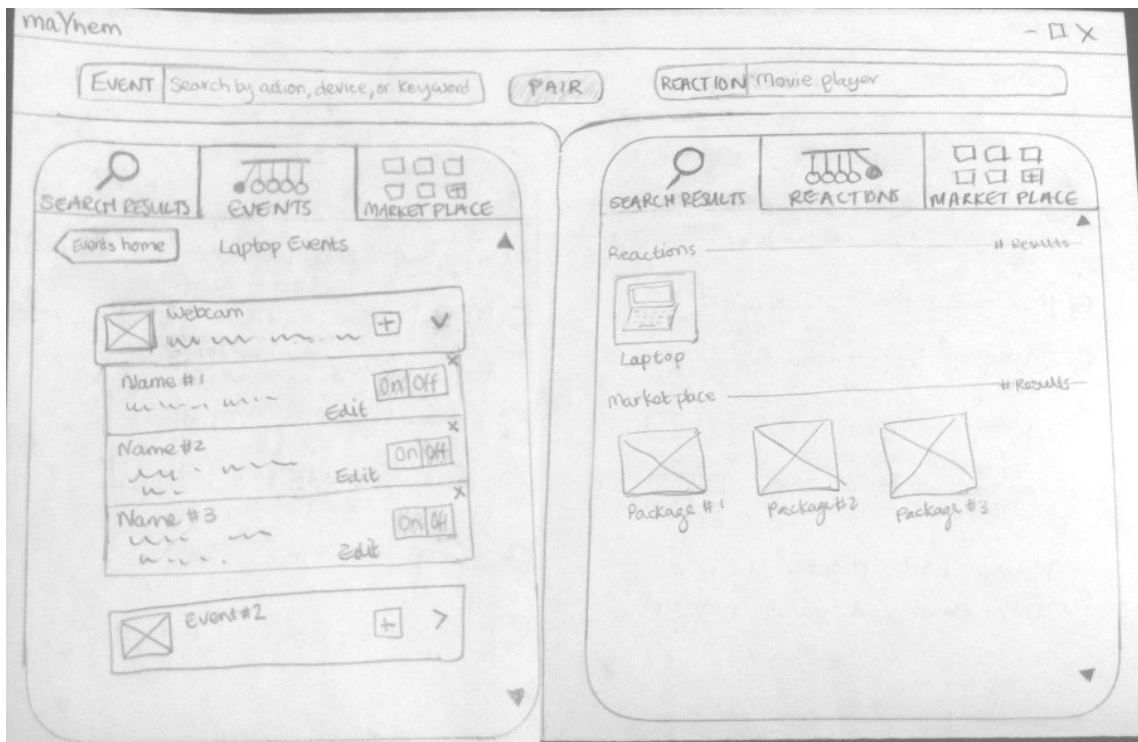


Figure 3B.

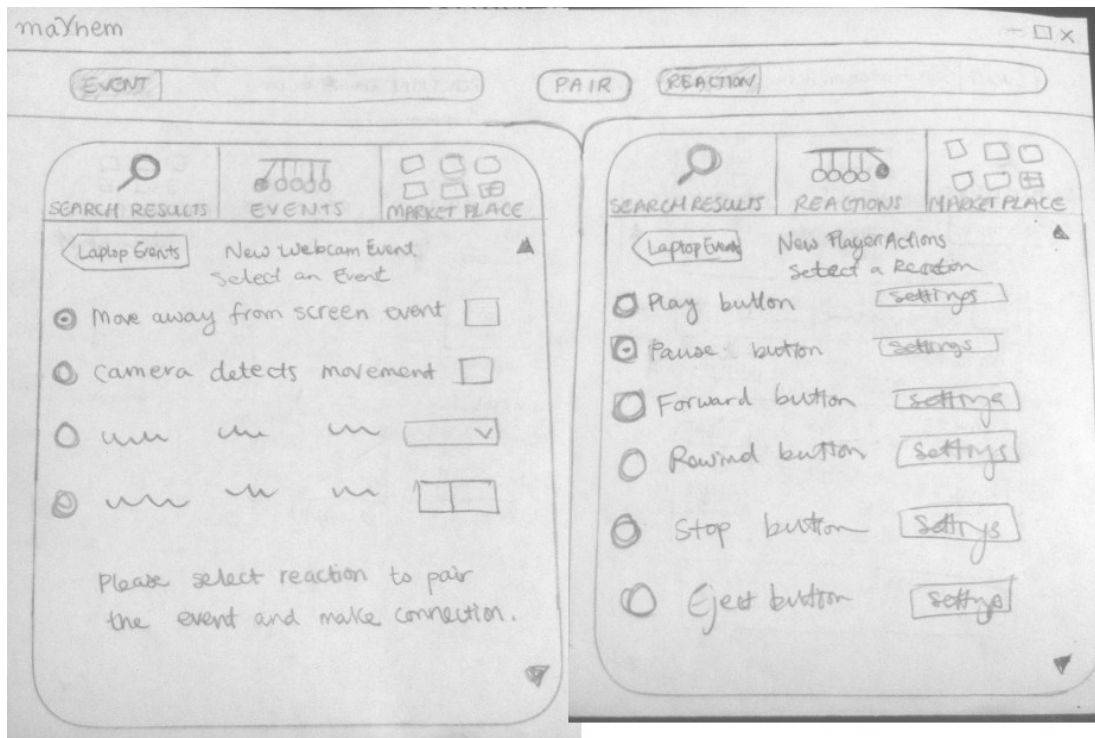


Figure 3C.

Selected Interface Design

We have selected a design that combines features from two of our sample designs. We like the idea of having a simple interface, as shown in the design in figure A in the Appendix, but want the user to be able to find their connections and the events and reactions that they have already downloaded with more ease. Making the interface minimalistic and familiar is important because we want the learning curve to be as small as possible, however, users who use the interface a lot must be able to still find the information that they require.

The first page that the user will see is the top left screen from figure 2. To create a connection, the user simply clicks the “Choose Events” button in the figure. They are given a new window which lists out all of their events they have currently, sorted by the package they came in, as shown in the lower left screen of figure 2. Once the user has found and selected their event, they may be asked to configure the event if the event is configurable. Once configuration is complete, the user is taken back to the home screen, where their chosen event has been updated with any relevant configuration information they provided. The user then clicks “Choose Reaction,” and goes through the same process as they went through when choosing and configuring their event.

Once a connection has been set up, the user has the option to deactivate it, delete it, or configure it. This can be seen in the expanded folder view in the top middle screen of figure 2. Here, the user can click the wrench icon to configure the connection, which would open up a connection configuration window. Here they can toggle the connection on or off, or add it

to a folder. To delete a connection, the user can click on the x next to the connection on the Expanded Folder View list.

A user could potentially have a very large list of events, reactions, or connections, depending on how extensively s/he uses Mayhem. When the user gets new events and reactions from the marketplace, they will actually be downloading packages of multiple events and/or reactions, all related to a specific application or theme. This very closely resembles the folder paradigm, which is an intuitive and familiar way to sort the events and reactions. An example of this can be seen in bottom left screen in figure 2. This user has downloaded a OneBusAway package, which contains events for a bus's distance from a stop, if the bus is late or if the bus has been rerouted. The user may have many different packages, and can connect events and reactions from any of the packages together. Unlike a traditional folder navigation structure where one drills down into sublevels, hiding the parent folder's content, expanding a package's contents simply displays its contents inline in the element list. This makes it easy for the user to see at a glance what the package contains, without the potential for getting lost in the folder hierarchy.

To sort connections in a more user-friendly way, we have added a tagging function, which is an important component of helping the user quickly locate an item s/he is searching for. By using tags, we greatly enhance the filter function, which is shown in figure 1C. Entering a search term into the text field at the top of a list will filter the list for titles and tags which match the current text. The list updates dynamically as the user types. Users can tag their connections with as many custom terms as they like, giving them a highly efficient way to customize sorting according to their own preference.

Connection folders are another design feature that we have added to the interface. In addition to allowing the user to create groups of related connections, they will also be able to toggle these groups active or inactive with the click of a button. An illustration of how this would look is in the top left screen of figure 2. There are two connections folders here, "Home" and "Work." You can click the arrow next to the folder to expand the folder and allow the user to see the connections in the folder.

All of these features together will allow users to search for and organize events, reactions and connections in a way that is intuitive and easy to manage. The design is still largely minimalist, and would be made to be aesthetically pleasing. It gives the user control over the organization of their information, and should allow the user to understand how to use it quickly even if they have never used the application before.

Scenarios

Pause your movie when you leave your computer

The user will begin on the main screen, seen in the upper left screen of figure 2. When he clicks Choose Event, a window pops up displaying a list of events, as shown in the lower left of figure 2. The user selects the Motion Detecting package from the list of events, expands it by clicking the > next to the package, and selects the "Camera Detects Sudden Movement" event

within the package. Since the event doesn't require new information, it is not configurable, so he will be taken back to the main screen, where the screen now displays his chosen event instead of the Choose Event button. The user then clicks the Choose Reaction button to open a Choose Reaction window, which is identical to the Choose Event screen, except it contains reactions instead of events. The user will find the Media Player package, expand it, and select the Pause reaction. This reaction will also not be configurable, so the user will be taken back to the main screen, and his new connection will appear in the My Connections section.

Display a notification on the Windows desktop when a specific bus is 10 minutes away

The user will begin on the main screen, and click Choose Event to be taken to the Choose Event screen. She will find the OneBusAway package in her list of events, expand it, and click the "Distance From Stop" event. Since this event is configurable, she is then taken to the Configure Event screen shown on the bottom right of figure 2. She will fill in the answers to the questions using her mouse for combo boxes, or her keyboard for text boxes. In this case, she will enter the bus number, the amount of time until the bus gets to the stop (10 minutes), and the stop number. Once this information has been entered, she clicks Save and is taken back to the main screen. Now the "Distance From Stop" event is displayed instead of the Choose Event button. She will now click Choose Reaction, and be taken to the Choose a Reaction screen. She searches for the Computer Notifications package, expands it, and selects the audible notification reaction. This event is not configurable, so she will be taken back to the main screen and her connection will now display in the My Connections section.

Use a smartphone to operate a PowerPoint presentation

The user will begin on the main screen. First he clicks Choose Event to be taken to the Choose Event screen. He then locates his Devices package and expands it. Inside the package, he will see a "Phone Button" event, so he clicks on it and is brought to the Configuration screen for that event. He will be prompted to plug his phone into his computer so that the application can recognize it, and be asked to verify the device is correct. Then he is prompted to click the button he wants to use as his event. Once he has done this, he clicks Save and is taken back to the main screen, where the screen will display the Phone Button event in place of the Choose Event button. Then the user will click the Choose Reaction button, and be taken to the Choose a Reaction screen. The user will find the PowerPoint package, expand it, and choose the Next Slide reaction from inside the package. This reaction does not need to be configured, so he is taken back to the main screen, where the new connection will be displayed under his My Connections section. Since he knows that he will have multiple events that are related, he can add them to a folder to better organize them. He does this by clicking the wrench icon to the right of his new connection, which opens up a dialog box allowing him to add the connection to a folder. He can create the folder on the fly by typing its name, "PowerPoint", into the dialog, and clicking "Save". This will take him back to the home screen, where he repeats the entire process to pair a different phone button event with the Back One Slide reaction. After this connection has been created, he can add it to the PowerPoint folder which he created in the previous dialog.

Appendix: Additional Sketches

This is a simplified view of the interface's behavior from the home screen. There are two general buttons on the home screen of the interface, one for selecting the event and the other for selecting the reaction. When you have selected both an event and reaction, the selections make a new connection and are added to the bottom of the connection list.

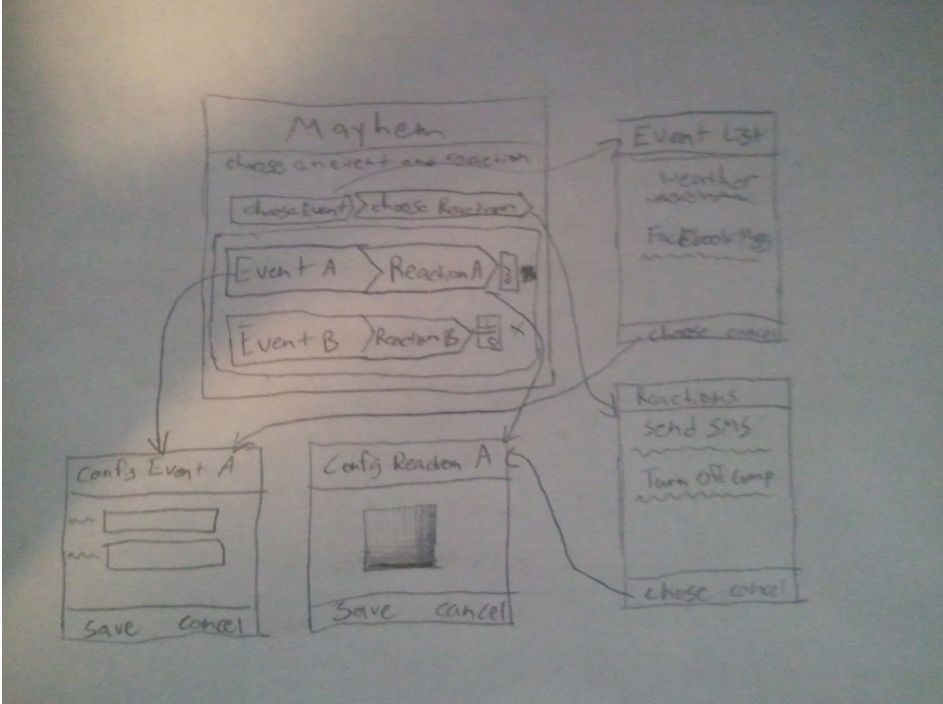


Figure A.