

# Wat2Search-SRP

rechstee, hudspero, lindo, milettal, sunamotl

February 10, 2018

Project name: Wat2Search, Team:Team Chronic

## **Description:**

Our project, Wat2Search, is designed to be an interactive and engaging flowchart that helps educate and inform users of what exactly is plaguing their machine. Through a multi-directional, easily traversable, web-implemented, graphic UI, we aim to help direct our users to the correct terms, phrases, and ideas to get a better grasp on a variety of technical issues. From hardware to software, peripherals, and almost everything in between, Wat2Search will cover it all. Our tool will even include helpful links and phone numbers to call for additional help.

Our set of targeted users aims to be beginner and intermediate users; those who don't know what things are, and those who wish to know what specific things are when they know the general information. In the same way that every human being needs to learn how to crawl before they can walk, every user is new at something. And if our tool can even shorten a single call by about 3-4 minutes, that will add up considerably in the long run.

Looking at community sites such as Reddit's Tales From Tech Support [3], along with Dell and other computer support forums [1], it's clear that there are users who don't know where to begin when they encounter one or more issues with their computer. Most of these users will attempt to resolve their issues over the phone, meaning there's not many physical records available to the public.

The only other sources of consolidated, well-defined information for our customer base come from web forums, developer support articles, various haphazardly-made guides, or by calling your IT support line and having the technician explain and fix the problem for the user. Even then, when choosing

any of those options, the user is expected to know exactly what their problem is and where to find the fix. If the user doesn't know what a folder is, what operating system they're running, or how to restart their machine, how can they be expected to troubleshoot anything?

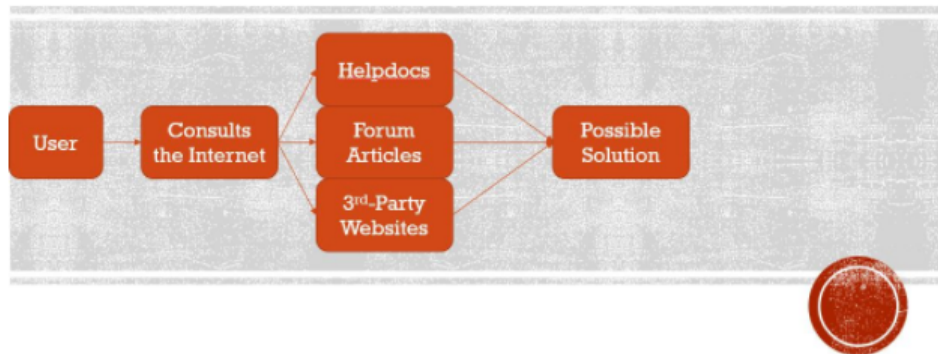
This tool will serve to link all relevant support articles, phone numbers, and even basic, relevant, informational tips into a single centralized source for all Oregon State students and faculty that are supported by the ISCS Service Desk (approval pending for ROOTS, ENGR, COSINE, Forestry, UDHS, OSU Foundation IT, and all other support teams). Because this will be a simple web-based application without the need for external database management or complex authentication/encryption, we can make this in Javascript and embed it in HTML.

While many hardware manufacturers and software developers do have knowledge bases and help documents [2], including OSU's helpdocs [4], as well as phone numbers that can be contacted for further assistance or support, none of these tools actively work with the user in real time in the event that there's no phone connections, nor make use of directly interacting with the user through a visual medium by guiding them through an easy, simplified, click-minimal, and scroll-minimal system with no changes in the URL.

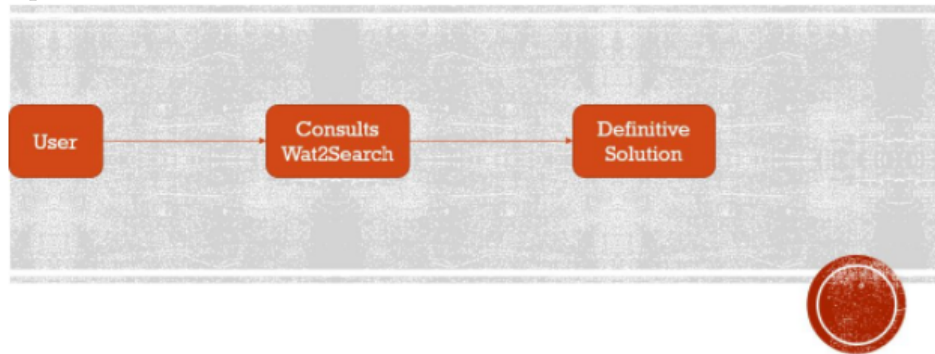
The ultimate features for our tool are readability (Where am I?), traversability (Where am I going?), good content (What am I looking at?), and some sort of solution to a problem (What is the ending?). Because our tool is an interactive support document, there's not much of a need for external documentation in the first place. Two stretch goals/future features we have are to have this exported and used in mobile devices, and for our tool to be modified and implemented in other OSU IT services.

## Diagrams:

- Option 1:



• Option 2:



**Use Cases:**

1. **Use Case Name:** Setup Wifi

**Goal:** Allows a non-technical user to set up or reconnect to their wifi through network preferences

**Actor:** Non-technical user

**Preconditions:**

- User owns a device that is able to connect to wifi (example: Macintosh or Windows XP laptop)
- Application is designed to assist with the specific device they are using to connect to wifi
- A working wifi (private/public) is currently available for use
- User knows their wifi password and wifi name

**Postconditions:**

- The wifi is successfully connected, therefore providing internet access to the non-technical user
- User verifies with application that they no longer need assistance with this specific problem

#### **Flow of Events:**

- User is provided instructions/visuals to locate network preferences  
Example(Windows XP): Open the Start menu in the bottom left-hand corner of your screen, and then select the Connect To option from the right-hand column.
- User is prompted to select/connect to desired wireless network connection
- Application aids user in navigating network preferences menu by providing visuals highlight specific elements (example: the drop-down menu containing wifi names to select from) they might be aided with identifying a wifi available for public use if this is requested by the user
- User is guided through process of providing wifi password
- Application validates with the user that they have successfully connected to wifi and will no longer need assistance

#### **Quality Requirements:**

- visuals are high-quality and are efficient in aiding users ability to navigate their computers user interface (Example: picture of network preferences menu)
- User is able to go back to previous steps when issues arise such as the user chose the wrong wifi network
- Instruction wording/diction is simplified in a concise and clear way to ensure the users comprehension

#### **2. Use Case Name:** Reset Password

**Goal:** Allow for an OSU student or staff to reset their ONID password

**Actor:** Non-technical user who is in need of resetting their password

**Preconditions:**

- User is an OSU student or staff with a valid ONID username and account
- Application is designed to walk them through the resetting process
- A working wifi (private/public) is currently available for use
- User knows their ONID username as well as how to use a basic web browser

**Postconditions:**

- The specific user is able to print from their personal printer or on-campus OSU printer
- User verifies with application that they no longer need assistance with this specific printer problem

**Flow of Events:**

- User is provided a link that takes them to the reset password page on OSUs website Example: <http://onid.oregonstate.edu/chpw.shtml>
- User is prompted to select Login to ONID if they know their current password
- If the user does not know their password, they are prompted to select the click here option below:  
Two links may also be provided such as:
  - Current password is known: [https://secure.onid.oregonstate.edu/cgi-bin/my?type=want\\_auth](https://secure.onid.oregonstate.edu/cgi-bin/my?type=want_auth)
  - Forgot password: [https://secure.onid.oregonstate.edu/cgi-bin/chpw?type=want\\_auth](https://secure.onid.oregonstate.edu/cgi-bin/chpw?type=want_auth)
- The user will be aided with links and screenshots accordingly
- User is guided through the reset process and given specific detail on what to enter at every step
- Examples are given as well
- Application checks with the user if they were successful. If not, a number for tech support and tech support hours will be provided.

**Quality Requirements:**

- Visuals are high-resolution and are labeled clearly with important information
- User is able to go back to previous steps if an error occurs with resetting their password
- Links are kept up to date in order to not provide incorrect or confusing information
- Screenshots reflect the most current method to resetting ones password

### 3. **Use Case Name:** Printing Issues

**Goal:** Solve the printing issue for the user and allowing them to print successfully

**Actor:** Non-technical user who is struggling to print either on campus or off

**Preconditions:**

- User is non-technical and are unable to print
- User is either trying to print on campus or on a personal printer
- A working wifi (private/public) is currently available for use
- User knows the basics of operating a web browser

**Postconditions:**

- The user was successfully able to print their document
- User verifies with application that they no longer need assistance with this specific problem

**Flow of Events:**

- User is prompted to select whether they are trying to print on their own printer or on an OSU printer
- If the user selects their own printer, the flowchart will diverge into a separate path
- Questions will prompt the user to check if their printer is on, plugged in to power, or plugged in to the computer

- Links will be provided to find the appropriate driver for the printer if the user is unsure if the driver has been installed or not  
Example: Select brand of printer  
Link: <http://support.brother.com/g/b/productsearch.aspx?c=us&lang=en&content=dl>
- The user will be aided with links and screenshots accordingly
- User is guided through the install process of the drivers
- If that all fails, the user will be prompted to use Windows troubleshooter
- Application checks with the user if they were successful. If not, a number for tech support and tech support hours will be provided.

### **Quality Requirements:**

- Visuals are high-resolution and are labeled clearly with important information
- User is able to go back to previous steps if an error occurs with fixing their printer or installing drivers
- Links are kept up to date in order to not provide incorrect or confusing information

The three use cases above cover the most important scenarios for our app Wat2Search. While there are a countless number of scenarios possible, the three we selected are likely the most common to occur. Three of the main issues people have that requires the IT helpdesk is internet connection issues, resetting their ONID password, and issues with printing. The solutions for other problems will be based on these same flow of events. The user will continue to be prompted for questions until a solution is found. In the case of an error, such as the user is unable to understand or follow the directions, we will provide the Helpdesk phone number and hours of operation. This will always be an error case as the Helpdesk is capable of handling more complicated issues that the app may not be able to solve.

## **Planning**

### Milestones

1. Documentation

- Goal: Cover all the potential support info our users may need and work out basic navigation concept.
- Timeline: Week 3 - Week 4.
- Reasoning: Having all of our goals and use cases set first will allow us to have clear goals when we get to implementation. This will allow us to concentrate on how to incorporate the info we will present to the client during implementation without worrying about what it is. We'll figure that out in this part of the timeline.
- Reasoning: Having all of our goals and use cases set first will allow us to have clear goals when we get to implementation. This will allow us to concentrate on how to incorporate the info we will present to the client during implementation without worrying about what it is. We'll figure that out in this part of the timeline.

## 2. Paper Prototype

- Goal: Craft a paper prototype for testers to run through with hypothetical situations clients run into.
- Timeline: Week 4 - Week 5.
- Testing with the paper prototype: Week 6 - Week 7
- Reasoning: Before actually plugging away at the program itself, we want to make that our systems actually make sense to users on a conceptual level. We'll throw in multiple hypothetical problems to ensure our dialogue tree has proper categorization of problems. This is being made for Client Services so we can use staff and student workers for troubleshooting, the extra week for this step allows us to get maximum feedback from our user base.
- Tracking: We can track this by listing the positives and negatives that users have. We'll know we're done when we have enough consistent data. We should aim to test at least a handful of participants each.

## 3. Program Prototype

- Goal: Have a functional prototype with the support information integrated for at least basic test cases.
- Timeline: Week 5 - Week 6



- Reasoning: With the paper prototype hopefully completed, we can move on to actually seeing how users interact with the widget on their computers. Coding the tree should be easier. We program the most important as well as common problems, determined by the paper prototype, to make sure that the experience of using our app to troubleshoot is solid before implementing the rest. On functionality, well at least add the ability to go between different troubleshooting subjects.
- Tracking: Well know were done with this when we have taken added the most relevant of the data compiled in our documentation. We also solidified what we need to change from the paper prototype for our user data in the development of the program.

#### 4. Programming Features

- Goal: Add the rest fo the support doc data into the program. Also add refined navigation through the app
- Timeline: Week 7 - Week 8
- With the bones of the program in place we can build off of the prototype. Well add the entirety of the documented test cases that we deem fit then move on to more advanced functionality like actual integration in multiple browsers as well as different actions for navigation within the network of troubleshooting issues we create for the widget.
- Tracking: We can make sure were on track for this part by referring to our description of the project in this assignment and making sure our requirements are met. We should also look back on the goals we initially set and look at the program as it is throughout this part of the project.

#### 5. UI+Graphics

- Goal: Make the program look user friendly and aesthetically pleasing. Finalize the UI with simplicity in mind.
- Timeline: Week 9 - End of term.
- Reasoning: After the program has all of the data needed and is completely functional, we can add visuals to make the program

pleasing to the eye. Were saving this step for last so that we ensure every technical aspect is solid before polishing it to look nice. Were going with the philosophy that a functional product is more important than one thats nice to look at. Graphic design and a nice ui will make the client feel more comfortable using it and draw them in, but the actual program will make it reliably used.

- Tracking: We need to put this program in front of people and see what they do first to make sure our UI is understandable. For visuals we can debate amongst ourselves and make mini prototypes with powerpoint to test how people respond to different looks. Well consider it done once we settle and implement a visual style.

#### Main Risks:

1. Lack of Communication: If we arent getting input from every member on when and what were doing for the development to come we cant plan ahead. Were using Discord as our primary means of discussion, an email chain for important matters and were going to try to meet up either in person or online for status updates.
2. Laziness: In a group project with multiple parts that requires effort from every member this could be one of the deadliest risks. We have to make sure our own initiatives are personally in check so that we can maintain a healthy working relationship with each other. This is just on all of us to make sure were sharing work as get this project done at every step of the way.
3. Losing Scope: This project is meant to be simple in nature. If we give into the feature creep discussed in class wed run the risk of making our troubleshooting app so complex that it needs troubleshooting itself. The best thing we can do is focus and worry about adding anything extra later in the term when we have already met the goals we initially made.

#### Meeting Report:

Our progress this week involved setting up our desired communication method and distributing tasks needed to complete this assignment. Initially, we used email to decide our method of communication because we were unable to

meet in person from conflicting schedules. We decided to communicate with each other using the mobile application, Discord. Furthermore, we finished this Software Requirement and Plan(SRP) which includes project information such as our target users and system requirements. This assignment allowed us to test the use of Discord along with Google Doc. Next week, we are going to follow our plan and work on it, and we need to have more communication with each other so we can exchange minds more quickly and improve efficiency. Through discord and emails we were able to distribute specific tasks to individual members. However, we were unfortunately unable to communicate with our customers to meet with them in person. Though, we were able to make good use of their vision statement and presentation. For example, the vision statement helped in determining use cases that are relevant and should be considered and evaluated for the project.

## References

- [1] Dell forums, 2004.
- [2] Free independent troubleshooting helpdoc from gcf learn free, 2015.
- [3] Reddit: Tales from tech support, 2017.
- [4] Oregon state help docs, 2018.