SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

Date: December 2, 2016

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Jason Chen
Matthew Koken

ENTITLED

SmartMirror: A Glance into the Future

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

_____
Thesis Advisor

_____
Department Chair

_____
Department Chair

SmartMirror: A Glance into the Future

by

Jason Chen
Matthew Koken

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
December 2, 2016

SmartMirror: A Glance into the Future

Jason Chen
Matthew Koken

Department of Computer Engineering
Santa Clara University
December 2, 2016

ABSTRACT

In todays society, information is available to us at a glance through our phones, laptops, desktops, etc. But that still requires an extra level of interaction in order to access the information. As technology grows, people slowly grow further and further away from the traditional style of interaction with their goods. In, the past, information was relayed through paper, then through computers, and in todays day and age, through our phones and multiple other mediums. We hope to push the envelope further, into the future. We propose a new simple way of connecting with your morning newspaper. We present our idea, the SmartMirror, information at a glance. Our product hopes to deliver your information quickly and comfortably, with a new modern aesthetic. While modern appliances require input through modules such as keyboards or touch screen, we hope to accomplish a model that functions purely on voice and gesture. We seek to deliver your information during your morning routine and throughout the day, when taking out your phone isnt always possible. This will pander to a larger audience base, as the average consumer nowadays hope to accomplish tasks with the wave of their hands. This idea has many future applications, such as integration with new virtual/augmented reality devices, or simplifying all consumer personal media sources.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 The Problem

In our rapidly developing world, information is always right at your fingertips - on your phone, on your computer, maybe even on your watch. Staying connected with new information is both important for entertainment and daily life. With such a variety of options, there is difficulty in following all of your data streams. Often, during your day, you may end up in a position where it is inconvenient, or even impossible, to take out your phone or computer and check the newest update. You cannot commit to a slower interaction. You need a display to glance at, with the information you need ready to go. However, aesthetics are just as important as displaying information. Keeping an extra computer in your bathroom or hall would be inconvenient, and would not fit well with the look of a modern room. A sleek, simple display, easy for an average consumer, is a necessity in todays world.

1.2 Other Solutions

There are several products in the market that attempt to be your attractive hub of daily information. The Amazon Echo and the upcoming Google Home present themselves as a small speaker that can relay information through sound. You can request news or music, fulfilling your need to obtain media content in a hands-free manner. However, not all data is suitable for conveyance by voice. Both designs lack the key ability to convey information visually. Asking for the morning traffic can give you a time estimate, but it barely comes close to a detailed map with your route information. Having the news read to you is convenient, but many prefer reading the news at their own pace. A smart display would be a product that would be able to answer all of these concerns, while staying smoothly modern. The Nest thermostat has a small display for information. However, it is not intended for interaction. The interface can be clunky, and not something an average consumer would interact with on a day-to-day basis. The recently Kickstarted Perseus hides a screen and computer behind a two-way mirror. This allows users to interact with the mirrors applications via touch

screen, voice, and camera controls. Perseus, however, is a finished product, and does not allow user hardware customization. It claims to have an available API and third party applications, but currently, there is little to no information or documentation on this matter. With months to go before its delivery, the success of this product remains uncertain. A few Do-It-Yourself (DIY) alternatives are also currently available. Both the MirrorMirror and Smart-Mirror projects provide an application to display information on a monitor behind a mirror. However, these require legwork on the user end, as not all users are willing to manually construct the project from scratch. Manual configuration and tinkering with modules can be a tedious and difficult process.

1.3 Our Solution

Our solution is an open platform for discrete display development. We offer an aesthetically pleasing mirror, with a hidden smart display underneath. With a generic display, the mirror can be built to any size so the information can be both in your face while showing you your face. Our product differs from the competition with an easy-to-use interface that is both simple for the average user and open for the advanced developer. A sleek display gives all levels of users a modern hub of technology for their personal daily interaction, one which both displays visually all the information you could need or want, and operates with a simple interaction that you could fit into your daily routine. By creating a platform open to modification, developers will also be add new functionality at their own pace. This will allow our display to be a tailorable and adaptable platform. A web application provides the interface that the user sees and interacts with. An online configurator will relieve the frustration and difficulty of personalizing your information, as well as allow streamlined development of new modules. Powered by a small computer, the smart mirror will have great potential for expansion by developers. As an open platform, consumers and developers will be able to easily build, adapt, and hack their smart mirror to fit their own needs. Our product will be a step in the future of IoT, connecting your daily mirror to your tech-savvy world.

Chapter 2

Functional Requirements

The following requirements define the goals of the project outlined in the introduction. The functional requirements define features that must be done for the project to be considered a success, while the non-functional requirements define how the functional requirements are achieved. Requirements are categorized into critical, recommended, and suggested. Critical requirements are absolutely necessary, recommended are highly desirable, and suggested requirements are not necessary but would be very nice to add.

Design constraints are criteria that the solution must adhere to. The constraints are set by the client and are non-negotiable.

2.1 Functional Requirements

Critical:

1. Must be able to display information on screen
2. Must be controlled by something without requiring direct input
3. Must be connected to the web to receive incoming data
4. Must be module-based and contain sample default modules
5. System defaults in low power sleep mode
6. Must be able to scale to multiple screen sizes

Recommended:

1. Controlled by alternative input methods
2. Live RSS feed displays
3. Integrate more advanced web modules, perhaps a browser
4. Sleeps when certain time has passed

Suggested:

1. Allow users to integrate their own web modules

2.2 Non-Functional Requirements

Critical:

1. A simpler user interface than a computer
2. System has good performance for users
3. System maintains good reliability for users
4. Display disappears and becomes a mirror

Recommended:

1. A friendly user interface that works by selecting modules
2. System remembers user name and can reply to user by name

Suggested:

1. Ability to augment a reflection

2.3 Design Constraints

Constraint

1. Solution must be standards compliant for popularly supported interfaces
2. Solution must be open-source
3. System must be accessible for hobbyists
4. System must be scale-able, extensible, and extendable

2.4 Requirements Justification

Due to the limited time and man-effort available for this project, we may not be able to implement all requirements. Each requirement will be fulfilled based on level of priority in order to create a minimum deliverable system. The minimum system for delivery will serve as a baseline for further discrete display development. As an open-source system, SmartMirror will be a platform for developers, hackers, and makers. Here, developers will be the main target as they will be able to add additional features to the extendable and extensible system.

Adding recommended features will improve usability and functionality for hackers and makers. A more user-friendly system will allow greater access to the platform for the public. Advanced features such as alternative input methods are extremely important for long-term usability, but are much more difficult to implement and will require additional research.

Chapter 3

Use Cases

Below are a few use cases for our platform. Use cases are examples of how our system is to be used. The main actor of our system is the student using the planner. Figure 1 displays our use case diagram for our system.

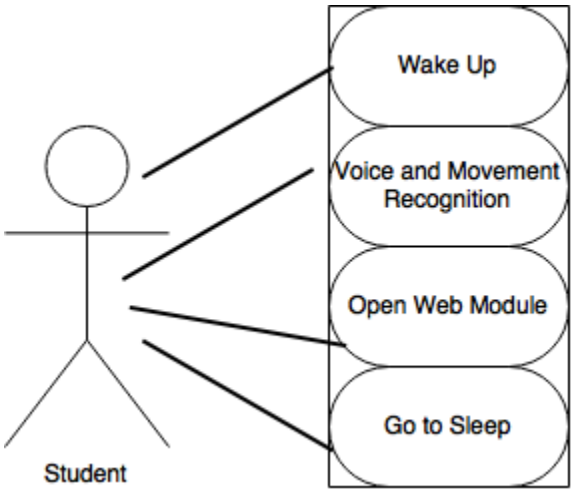

Figure 3.1: Use Case for Project

3.1 Use Case 1:

Name: Wake up

Goal: User is able to start up the system

Actors: User

Pre-conditions:

- System is low power mic on mode
- System is attached to power

Post-conditions:

- System starts up and navigates to user's GUI

Steps:

1. User says startup phrase (e.g. "Wake up SmartMirror", "Hello SmartMirror")

2. System powers up
3. System navigates to user GUI
4. System displays user GUI to user

Exceptions:

- System is off
  - User must physically turn system on with power switch
  - System boots up

3.2 Use Case 2:

Name: User Opens Module

Goal: User is able to access provided interface modules (i.e. app, widget)

Actors: User

Pre-conditions:

- System is displaying GUI
- User is present in front of the screen

Post-conditions:

- Selected application is opened

Steps:

1. User selects module
2. Selected Application is opened
3. Selected Application is displayed on the screen for user interaction

Exceptions: N/A

3.3 Use Case 3:

Name: Voice and Movement Recognition

Goal: User is able to input data via voice or motion command

Actors: User

Pre-conditions:

- System is displaying GUI
- User is present in front of screen

Post-Conditions:

- System successfully recognizes user motions/voice as input

Steps:

1. User moves hands/speaks keywords
2. System correctly interprets user activity as command
3. System correctly executes user's command

Exceptions: N/A

3.4 Use Case 4:

Name: Go to sleep

Goal: User is able to turn off the system

Actors: User

Pre-Conditions:

- System is on

Post-Conditions:

- System is in low power sleep mode

Steps:

1. User says key phrase, such as ”SmartMirror Off”
2. System interprets phrase as off command
3. System closes all current applications
4. System goes to low power sleep mode

Exceptions: N/A

Chapter 4

Activity Diagram

The general work-flow of the planner can be graphically represented in an activity diagram. Figure 2 (Pg. 12) shows how an average user will use the SmartMirror, and the step by step process they will go through as they progress through the site. The diagram shows the work flow for all average users. Users will start up the system from low power mode by saying the phrase ”Wake Up”. The system will then load the user GUI. The user is then able to interact with selected modules, or open new modules. The system is always checking for the phrase ”Go to Sleep”, once it it said, the system will close all applications, then return to low power mode.

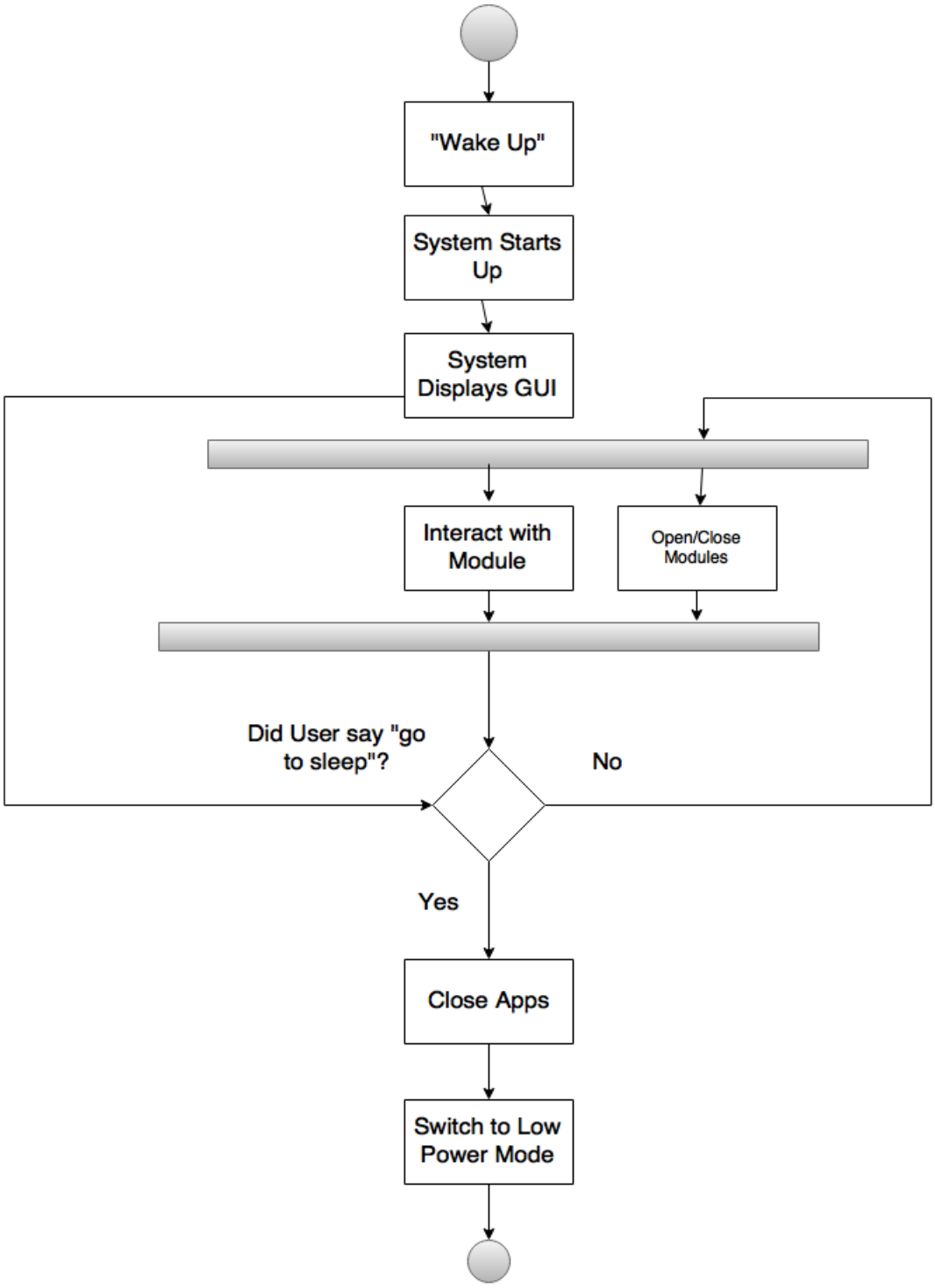

Figure 4.1: Activity Diagram for Project

Chapter 5

Conceptual Model

We propose a mirror covering a screen screen, both of which approximately measure 23" diagonally. We plan to implement a web camera above, perhaps integrated, as well as a strong microphone system on the sides. This will be encapsulated all within a wooden frame box.

For the interface, we wish to aim for a simple design, not unlike Apple's iPhone GUI design. The center should be kept clear, as we wish for users to be able to interact through the camera, and modules overlapping with their model would get complicated. Therefore, the web modules will be to one side, while the user stands in the center of the screen.

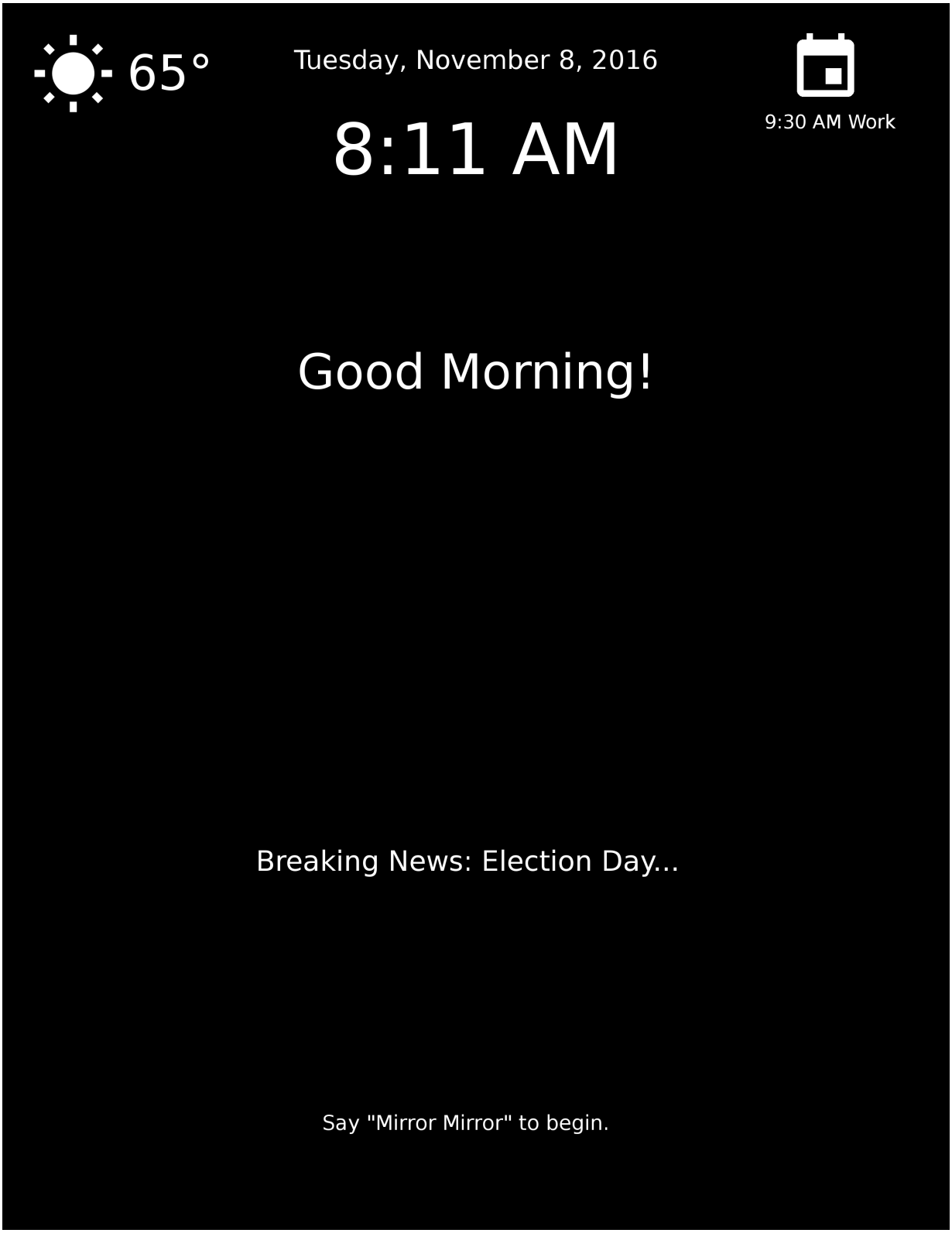

Figure 5.1: Conceptual Model

Chapter 6

Architectural Diagram

The design of the system will be a combination of a layered architecture and a client-server architecture. The user will be interacting primarily with the GUI that is built upon the OS we use for the development for our system. When they attempt to make requests to edit their settings directly, they will really be making a call as an application to the underlying OS, which will change the system accordingly. The system itself will communicate with remote clients via the Internet. When receiving information for interface modules, the system will act as a client to the web services. In interacting with other smart devices, the system will act as a server for information. Figure 3 highlights a possible example of a layered architecture:

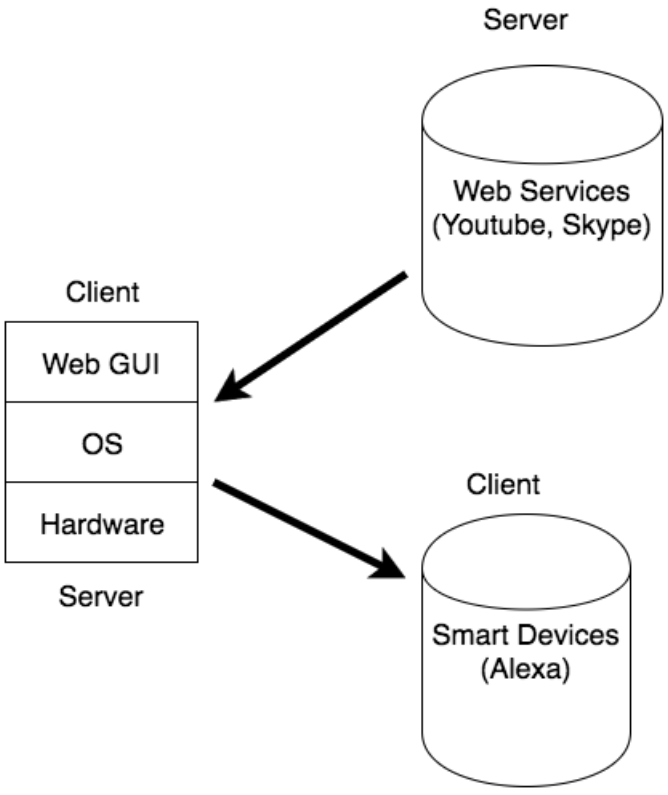

Figure 6.1: Architectural Diagram

Chapter 7

Technologies Used

Below are a list of technologies we plan to use for our project. Being a web-based application, we used a variety of web tools including HTML and JavaScript.

- XML
- CSS
- JavaScript/JQuery/NodeJS
- PHP
- Python
- Raspbian

We are choosing to use XML and JavaScript mainly due to the fact that they are the most fundamental ways of creating a simple web page, while also being able to be creative and allow interactivity. To make our website more visually appealing, we will also use CSS. Our main focus will be to create an interface that is simple for the user to interact with, but sleek for the user to read.

PHP will be important for writing scripts that execute via user commands. NodeJS allows for a much easier handling of I/O data. Because our model uses Raspberry Pi, Python may be used to be able to modify the underlying OS beneath the user GUI.

Raspbian is the standard operating system for our development board, the Raspberry Pi. Other operating systems and embedded devices may need to be supported in the future, but Raspbian will be our baseline for development.

Chapter 8

Costs

We propose that this project will be self-funded. Therefore, we will not be requesting funding for this project, as the costs that it requires are not large enough to request specific university funding. However, the specific materials that are used will be listed below:

Raspberry Pi 3 Model B Starter Kit	$49.99
One-way Mirror	$40.00
Generic Monitor	$80.00
Wood for Frame	$60.00

Table 8.1: Costs Chart.

Chapter 9

Design Rationale

We decided to make our model open source, due to the fact that it allows a plethora of users to enjoy the product on different levels: the average user will not concern themselves with a difficult world of development, and the talented programmer will be able to modify the product to suit their needs. We feel that this supports and builds a healthy community, as all other marketed renditions on the market today do not allow for customization. We feel that allowing our product to be extensible not only allows for a much easier integration, should a group wish to continue developing this product, but also allow for users to be able to create some truly unique modifications.

We decided to go with our languages, due to the fact that our design is a web module. Users will interact through the web module, and be able to access their data through said web module. This allows for a much easier integration into the web, and displaying constantly updated data will be much easier to retrieve through web modules.

We decided that an always-on-mic style of activation would create a very unique touch: Applications like Apple's "Siri" or Amazon's "Alexa" already have this type of feature integrated in, and we felt that it would be a vital part in developing new IoT hub technology.

Chapter 10

Testing Plan

We plan to create a working model by the end of Christmas Break. We plan to further develop more web modules as more time progresses, but we hope by that time to specifically get the fundamental aspects of the project down.

We will begin by testing that the web modules are able to be interacted with, and they are being displayed in a manner that is pleasing to the user. Then, afterwards, we will develop into alternative sources of interactivity, such as our aforementioned voice and movement recognition. Then, we will test our system again, but only by using those alternative sources of input. If the system successfully is able to be interacted by voice/movement recognition, and a user is successfully able to interact with the provided modules, we will move on to alpha testing.

Alpha testing will consist of Matt and I both using the product for a day, and seeing if it successfully accomplishes all the tasks that we require of it. We must make sure that it successfully turns on when asked, our interactions with the product are not too clunky and function smoothly, and it successfully turns off when asked.

Once our group completes this, we will then conduct beta testing. This will include inviting volunteers to work with our system, and taking their feedback for improvements to the design and user interface. This will be an important step in order not only to determine design, aesthetics, and functionality, but also to root out bugs with different requests from different people. We will then accommodate user preference and client requirements, and then perhaps move on to adding extra small functionalities.

Chapter 11

Risk Analysis

For our risk analysis table, as seen in Figure 4, it is broken down into six different categories.

- Name of Risk
- Consequences
- Probability
- Severity
- Impact
- Mitigation Strategies

Each risk has consequences for what happens if the risk actualizes. It also carries the probability of occurring, along with the severity of the risk. Probability and severity factor into the impact the risk has on the project. Finally, the table outlines mitigation strategies our team has prepared to help lessen either the probability or severity of the risk. This helps lessen the impact of the risk should it still happen.

The largest risk to our project is our team members becoming ill, impacting our schedule for our project. This relates to our other major risk that we will not be able to finish on time. It is still possible that the final deliverable will not be ready by the due date, as the system might still have bugs that critically affect performance.

One large potential risk is not correctly being able to recognize voice commands, thus rendering turning on and going to sleep as impossible to fulfill. This would be a huge problem, as it would require a complete overhaul of design, due to the fact that such an important function would not function properly, or for more research and development into the specific end of voice recognition. Either way, it would end up eating up a large chunk of time.

Risk	Consequences	Probability	Severity	Impact	Mitigation Strategy
Bugs	Project does not run; Project not finished	.8	10	8.0	-Rigorously test project -Develop code in steps -Leave comments for team members to troubleshoot
Failure to satisfy client	Project is unusable in desired setting	.7	10	7.0	-Regularly check-in with client -Seek outside feedback
Sickness	Inability to work, push deadlines back on project	.8	7	5.6	-Eat Healthy -Regular sleep schedule -Follow good hygiene
Failure to meet deadline	Project is not finished on time	.5	10	5.0	-Follow development time-line -Do not put off tasks
Lack of Language Knowledge	Progress is slowed, project not finished on time	.5	7	3.5	-Become familiar with languages -Plan which languages will be needed

Figure 11.1: Risk Analysis for Project

Chapter 12

Development Timeline

As shown in Figure 6, our group has prepared a schedule for meeting our project goal and deadlines. Each team member is assigned tasks to help ensure project success. The legend above identifies which team member has which task, along with group tasks, and deadlines. While most of the group is working on tasks together, certain group members have been appointed leads on different areas, such as customer liaison, implementation lead, and testing lead. The timeline has been broken into four different sections: Requirements, Design, Implementation, and Testing.

For Requirements, this is divided into two parts. The first part is the initial requirements gathering from the client, during weeks 8 and 9. The requirements supplied by the customer help formulate our design idea. The second part, from weeks 8 through 7, is a period of customer interview and feedback on our product, to help mitigate the risk of failing to meet customer expectations.

For Design, this phase is conducted during weeks 8 and 9. Using the information from the Requirements phase, our group will develop a design for the online interface that users will experience.

For Implementation, this phase is conducted during weeks 8 through 6. Building off on our design, our group will build the code to program the web modules, then do the research to develop voice recognition and motion recognition.

For Testing, this phase is conducted during weeks 1 through 9. The group will start with basic cases and functionality testing, before moving on to our alpha and beta testing. The feedback from the beta testing will be used to fix any design issues that may not have been prevalent during design and implementation.

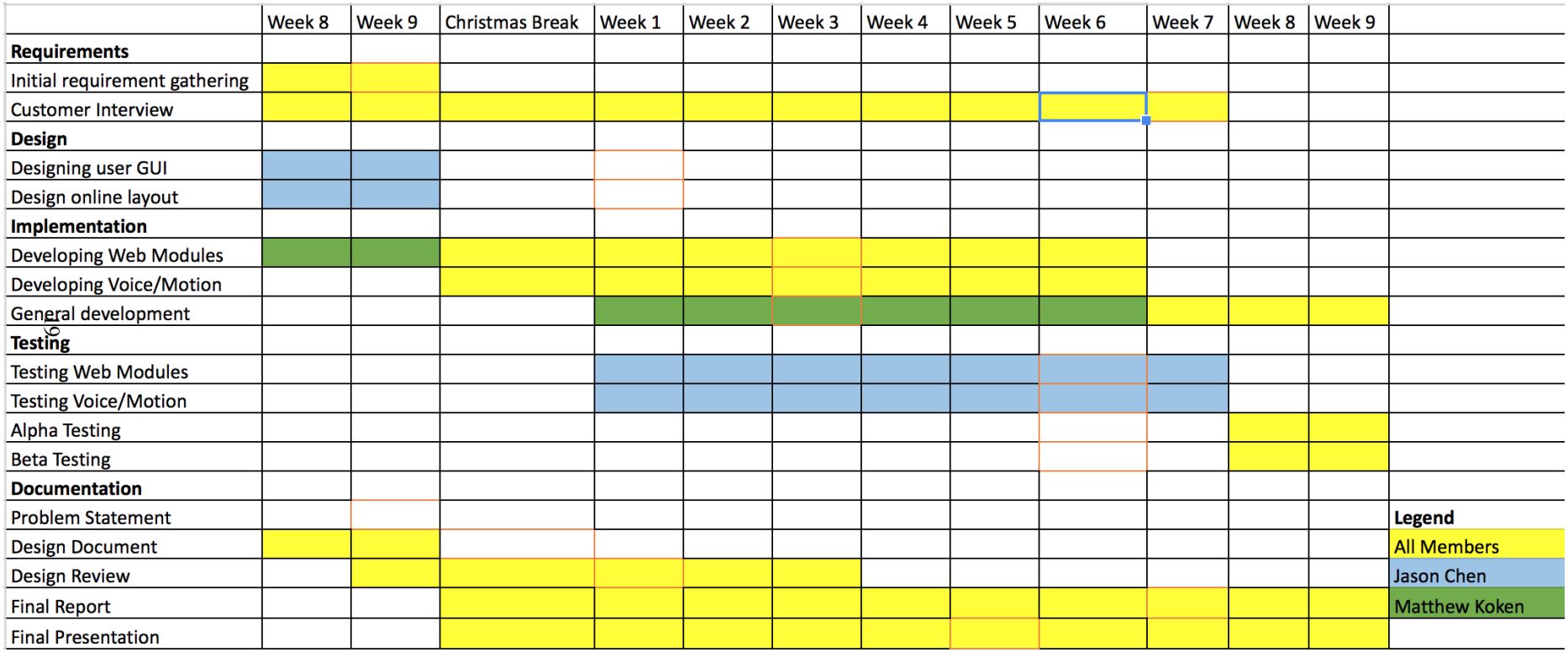

Figure 12.1: Timeline for Project

Chapter 13

Annotated Bibliography and References

- Mcheick, Hamid. "Ubiquitous Computing and Context-aware Applications: Survey and Contributions." 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE) (2016): n. pag. IEEEXplore. Web. 23 Oct. 2016.

  This paper discusses the current research into the field of ubiquitous computing, a field which tackles similar material to our current senior project. The paper discusses some of the most important properties when developing ubiquitous computing/IoT.

- Yousfi, Alaaeddine, Adrian De Freitas, Anind K. Dey, and Rajaa Saidi. "The Use of Ubiquitous Computing for Business Process Improvement." IEEE Transactions on Services Computing 9.4 (2016): 621-32. Web. 23 Oct. 2016.

  This paper discusses the current state of our computing era, one in which business as a whole is being heavily affected by the current state of rising IoT technology. IoT technology is affecting all business, due to the rise of need for businesses to further optimize their performance in competition. This paper provides an interesting knowledge towards trying to focus our product towards a business consumer market.

- Xu, Guangquan, Yan Ren, Gaoxu Zhang, Bin Liu, Xiaohong Li, and Zhiyong Feng. "HyCPK: Securing Identity Authentication in Ubiquitous Computing." 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom) (2015): n. pag. IEEEXplore. Web. 23 Oct. 2016.

  This paper discusses HyCPK, a new identity authentication algorithm created specifically for ubiquitous computing products. It provides an interesting direction in terms of cybersecurity, and also provides a good look into security and authentication for our senior project.

- Lee, Byoungcheon. "Hybrid-Style Personal Key Management in Ubiquitous Computing." Proceedings of the 11th International Conference on Security and Cryptography (2014): n. pag. IEEE. Web. 23 Oct. 2016.

  This paper discusses ID-based cryptography for multiple computing devices. It provides an interesting look into cryptographic methods for a user communicating with multiple devices, which would be useful for a project such as ours which introduces another IoT device to the world.

- Yagmur, Serap. "A Literature Review: Usability Aspects of Ubiquitous Computing." 2016 International Conference on Platform Technology and Service (PlatCon) (2016): n. pag. IEEE. Web. 23 Oct. 2016.

  This paper discusses usability in ubiquitous computing. It mentions the most important aspects to consider when designing a new ubiquitous computing item, as well as discussing the research trends towards new development in the field of ubiquitous computing. This article is useful to both the development aspect, as well as reading up on trends of IoT development.

- O'donovan, Paul. "The Tv's Vanishing Act: Radical New Display and Content-delivery Technologies Will Doom the Television Set." IEEE Spectrum 53.5 (2016): 48-53. IEEEXplore. Web. 23 Oct. 2016.

  This paper describes the history behind the rise and fall of our home television, and the trends of consumers as the idea of the home television is soon coming to an end. This paper is not a technical concept, rather more of a study into consumer market trends and history. It will allow us to understand what exactly our product is aiming to replace, and what functions and needs it should seek to replace for the modern consumer.

- Montfort, Nick, and Ian Bogost. "Random and Raster: Display Technologies and the Development of Videogames." IEEE Annals of the History of Computing 31.3 (2009): 34-43. IEEEXplore. Web. 23 Oct. 2016.

This paper discusses the changes we see in the development of graphics in the video game world. It summarizes the development of random-scan, raster-scan, and now flat-scan technologies that are used nowadays for displaying games. While not a direct requirement for our project, it could prove useful, especially for a side module that could become one of the ideas we end up advertising about our project.

- Nakajima, Tatsuo. "How to Reuse Exisiting Interactive Applications in Ubiquitous Computing Environments?" Proceedings of the 2006 ACM Symposium on Applied Computing - SAC '06 (2006): n. pag. ACMDL. Web. 23 Oct. 2016.

  This article discusses how to reuse GUI toolkits for new ubiquitous computing development. It discusses how separating the GUI from the output device are important for creating a model that is extensible. This will be specifically useful if we try to connect different types of interactions to our specific GUI.

- Heun, Valentin, Shunichi Kasahara, and Pattie Maes. "Smarter Objects." CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13 (2013): n. pag. ACMDL. Web. 23 Oct. 2016.

  This article discusses an idea on combining a TUI with a GUI for a simpler user interface. This type of information could be useful in regards to our project, because our idea also markets to a similar user, and information in regards to this idea could be implemented in our own development.

- Hounshell, Jonathan, and Todd Emma. "The Invisible Interface: Traditional Design Principles in Modern Electronic Design." 2016 20th International Conference Information Visualisation (IV) (2016): n. pag. IEEEXplore. Web. 23 Oct. 2016.

  This article describes the steps in order to create an interface that allow a users experience to be as smooth as possible. It discusses spacing, continuity, color usage, and game design, with the term invisible interface used to describe the relation the user ends up having with the display. This is useful due to the fact that we are designing our own display, and to have an optimized design would be extremely beneficial for our product.