The Virtual Office Door | Team Conquistadoors

REQUIREMENTS DOCUMENT

James Hauser, Mitchell Hewitt, Nicolas Melillo, David Snow, Tyler Tollefson

Mentor: Dr. Eck Doerry

Clients: Dr. Eck Doerry and Dr. Michael Leverington

Date: 12/7/16

Version: 2.0

Accepted baseline requirements for the project:

Client: _____

Team: _____

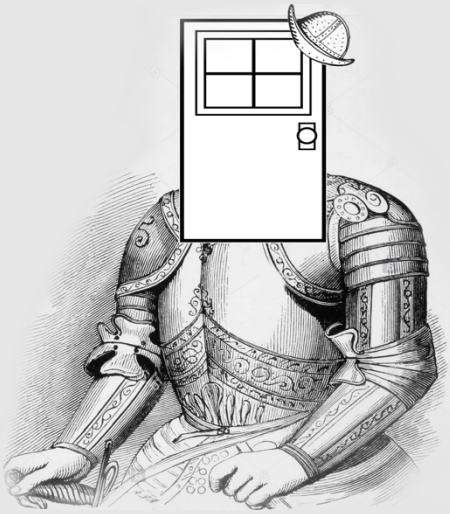


Table of Contents

Introduction

In many large organizations employees interact and reside in large physical office spaces that are often populated by cubicles and large areas where employees of the organization can meet and discuss ideas or issues. However, in most organizations cubicles and even office doors serve as a valuable tool for communication between workers and can help convey urgent information to someone in the physical office door space. This physical office door space is not only seen in the professional world though; it is heavily used Academia as well. Office doors used by teachers at universities are often used to communicate with students about a variety of topics ranging from office hours, class announcements and even the occasional cartoon or comic strip.

This is where our client, Dr. Michael Leverington comes into play. A past teacher at the University of Nevada, Reno and a new professor at NAU. Dr. Leverington's main business is teaching classes, within that though resides a far more important business, communication with students. Through this business of communication Dr. Leverington and other teachers convey updates about students' classes, grades being posted, and other class related information. However, Academia is not like other big businesses, teachers are paid through a university, but the real value lies in the knowledge that teachers impart onto students. In this day and age with more and more people attending college Academia is booming and the size of the business is growing, and in order to keep up with the growth teachers need a more efficient way to communicate with students.

Specifically, at NAU and at Dr. Leverington's previous school, the major limiting factor in communicating with students just so happens to be the physical space, or the professor's office door. A few major issues arise from this:

- The door owner has to physically be at the office door to update it.
- There is only so much space on an office door.
- A person must physically be in front of the office door to view information.
- A person's success may depend on clear communication of the door's information.

Figure 1: A cluttered office door

Problem Statement

In order to visualize and understand the workflow of our client's business we created the flowchart below, in coordination with feedback from our client and mentor, to identify where problems can arise.

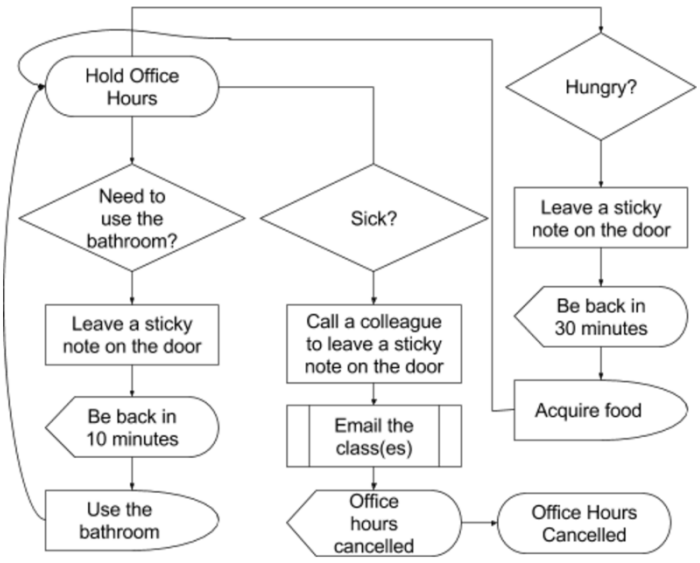


Figure 2: Office hours workflow diagram

In every case dealing with holding office hours the one major limiting factor is that the teacher needs to be physically at the door to update whether or not there are going to be office hours. When a professor has to leave for a class or an emergency there is not always time to return to their office to post a message on their door or send out an email to students. Even though a teacher could ask a colleague to leave a message on their door, there is still the issue of emailing the entire class or classes about the cancellation of office hours.

Often times though these messages, even if they are posted on office doors and sent out to students, are forgotten or not even looked at. This can then negatively affect students by not knowing where the professor is or in some cases when multiple messages are posted on doors, students become confused and often return home with unanswered questions. However, communications to students could be about issues other than office hours, such as:

- Status of the class
- Class cancellations
- Information about homework assignments

Which ties us back into our major issues that we outlined earlier, and in order to remedy these issues our client came to us with an idea. The idea is for a virtual office space, something that has never been done before, where teachers can post messages online and students can access this space to view communications as soon as they occur.

Solution Vision

To solve the client's problem, we intend to build a *Web 2.0* application, a website whose pages have dynamic content, that represents an office door but online. This website will be account based in order for users to securely modify their own personal door. The user's office door will be accessible online, allowing any visitor to view any office door on any platform with any web browser. Key features of this website include:

- Virtual Office Door space that allows door owners to post schedules, sticky notes, etc.
- Doors that are visible by visitors, such as students, to the website.
- An unlimited space for posting different communications.
- Ease of use and ability to send messages to subscribers of a door
- The door owner can update information via a web interface.

For the purpose of conveying what the door might look like to our clients, we drafted up a simple design for what the interface could look like.

For the main door interface we decided to go with a clean look that allows for ease of use on the door owner's side. The menu shown above will actually be collapsible in the final interpretation of the UI and it will contain more option such as: Load Door, Save Door, etc. All the different "widgets", or mediums to communicate schedules, quick messages, and display pictures, will be moveable as well as resizable and can have content that is set by the office door owner. All of which can be edited from a web interface that the user can access after signing in to their account from any computer and web browser.

Figure 3: Door UI mockup

In addition to that our software enables working environments to communicate information that would conventionally be transmitted physically to communicate that information through the internet. This allows any person to receive the information put out by another person regardless of where the two are located on the planet. We have also designed this application so that it complements other channels of communication rather than competes with them.

For example, our application provides basic messaging, social interactions as well as professional announcements in a way that focuses on content directed to specific known recipients and making our application's content visible to visitors of the office door whether or not they are known or unknown.

Project Requirements

After outlining our client's problem and our proposed solution, we needed to flesh out a multitude of requirements to serve as specifications for this project. Throughout our iterative process of defining requirements, we managed to formulate several "User Domain" level requirements.

Each of the User Domain requirements then correspond to a multitude of functional requirements, and from those stem performance requirements. For this project there are also several environmental requirements that our sponsors have specified that need to be mentioned. Each domain requirement is accompanied by a short summary of what our client specifically wanted as well as a summary of what the requirement entails. Following that the functional requirements also are accompanied by descriptions that cover more detail about the specific requirement.

User Domain Level Requirements

UD1. Website login must be secure and reliable.

For this requirement our client had specified that there needs to be a way to be able to log into the website for teachers. This could extend to other users being able to log in to the site; however, our client did not specify a specific way to implement the login system so this is covered in detail in the functional requirements.

UD2. Allows for notifications from the owner of an "office door" to users who opt in.

For this specific component of the project the client requested that there is a way for a one-way communication, the medium to be determined by us, between the office door owner and the viewers of the door. The method in which the user opts in for said communications is to be determined by our team as well. Our proposed method of communication is detailed in the functional requirements.

UD3. User needs to be able to create a customizable "office door".

One of the main domain requirements, our client had requested is that users are able to create a virtual office door. The client had specified that the office door should be configurable with widgets or apps, which are covered in another domain requirement. Our client had also specified that the office door needs to be viewable by an outside user, or someone who is not registered through the site.

UD4. Support multiple "apps" that are editable by the user.

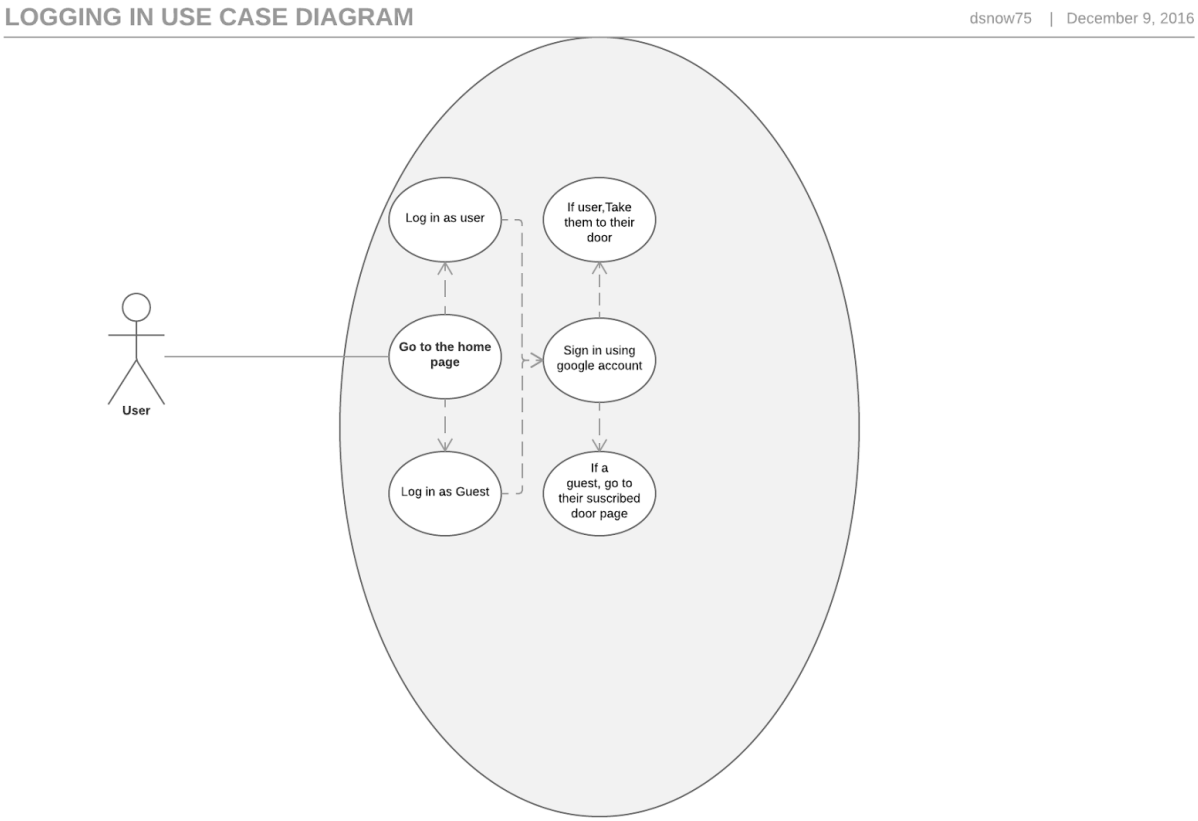
This requirement specifically relates to the widgets that the client had requested. For this the user specified they would like several different widgets that can range from a calendar widget to a sticky note widget.

UD5. A touchscreen door display that can be placed on an office door that shows the specific user's virtual door.
Our client requested that along with the virtual office door, that there is still a physical display that can project the user's office door on their actual office door. Our client did not specify a specific way to solve this particular domain requirement, but they did give several suggestions that we are taking into account with our research.

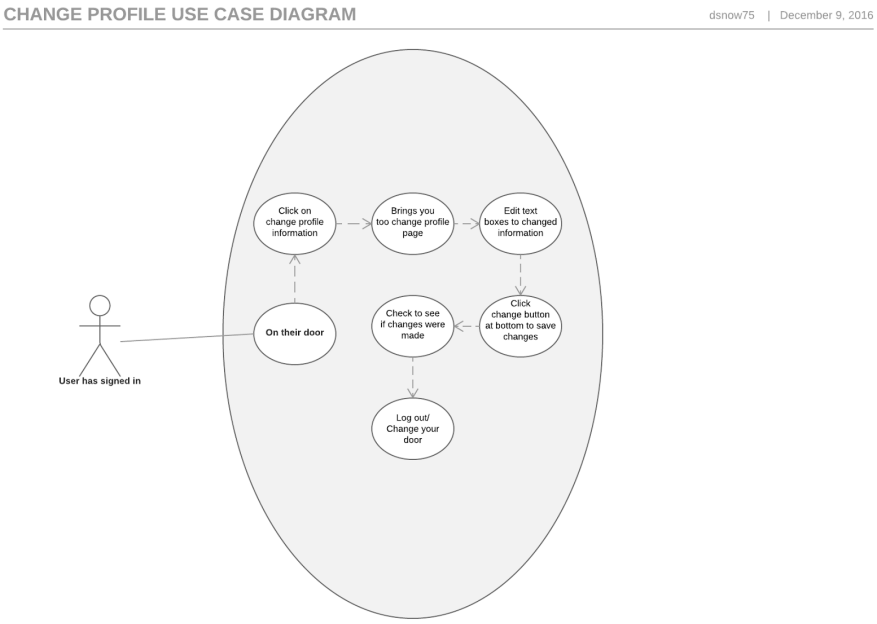
UD6. Utilize a cloud based server to store user and office door data.
This requirement covers the client's request to have the web app and the backend to be hosted on a cloud based server, using a web service such as Amazon Web Services. This cloud based server will enable portability of this application in case it is going to be used across multiple departments at Northern Arizona University or expanded for use in a professional context.

Use Cases

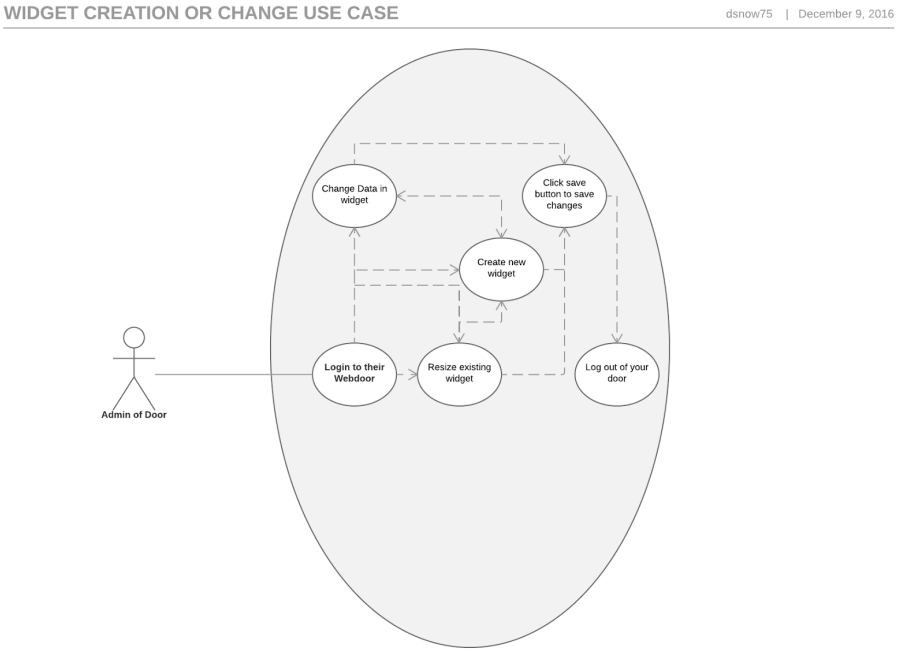
After detailing all of our domain level requirements, we went into further detail and started to formulate Use Cases. We split the use cases into two possible scenarios, sunny day use cases and rainy day use cases. Sunny day use cases relate to the most likely use cases and are the one the client and their business will most likely encounter, whereas rainy day use cases relate to the edge cases that would not occur as frequently.

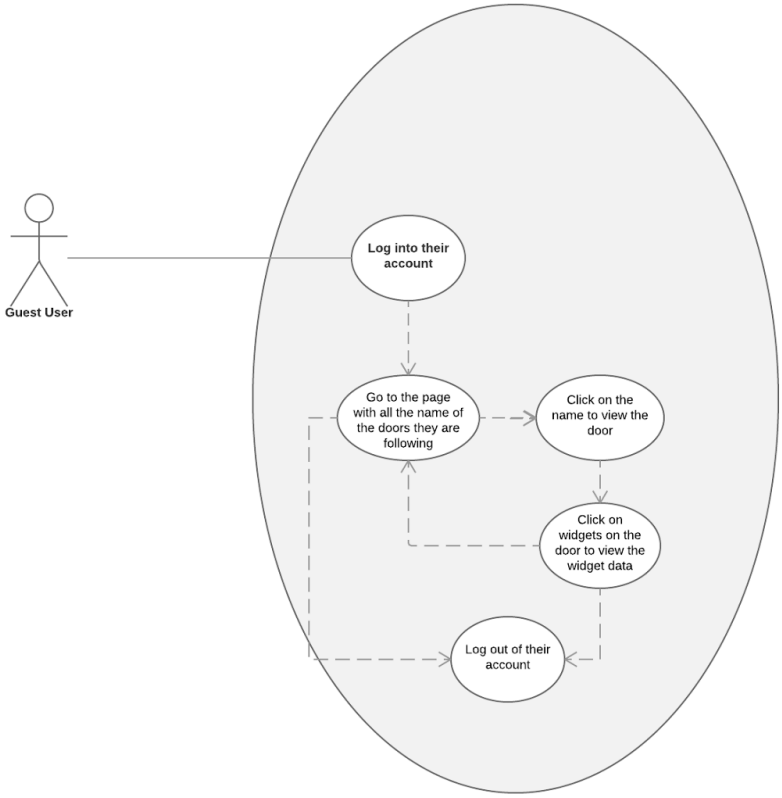


For this use case we decided to figure out the basic flow of what would happen after a user signs in to our website. Specifically after the user has logged in we needed to determine the flow of what would happen after the login was a success.



For this use case we go through the basic program flow for when a user has successfully logged in and wants to change their profile information. Then the use case below has to deal with when an admin of a group of doors wants to create a new widget.




Guest User
Log into their account
Go to the page with all the name of the doors they are following
Click on the name to view the door
Click on widgets on the door to view the widget data
Log out of their account

Functional Requirements

Each of the following functional requirements are attributed to a specific User Domain requirement, where “UD#” stands for User Domain Requirement # (where # can be 1-6) followed by “FR#” which stands for Functional Requirement. Within some of the functional requirements there are sub-requirements which are denoted with “SR#”.

UD1.FR1. User must be able to securely sign up for an account using a google login system

- Using the Google login API, user’s email will be registered with the backend and stored in the database.
- Using the backend application, create a web space for the registered user.

UD1.FR2. The user should be directed to their “home page” after successful registry/log in

- User should be able to view an image or icon representing their current office door on their homepage.
- User should be able to edit their personal information, such as name, etc. in a “profile edit” web page.

UD2.FR1. A guest visiting a user’s office door should be able to opt in for notifications

- User should be able to set whether or not they allow notifications to be sent to guests who sign up.
- User notifications will be sent via email or sms to the guest that is taken from a mini-form on the user’s office door.

UD2.FR2. A guest who previously signed up for notifications should be able to opt out via the user’s office door, or the medium of communication

- Guests should be able to click an “unsubscribe” link in the email sent from the system that removes their email from the database.

UD3.FR1. The user must be able to navigate to their personal “office door” page

- Users should be able to click a button or link that takes them to an editable view of their office door.
- The backend must be able to load all user office door data, including positon of widgets, door background, and widget content, as soon as the navigation is complete.

UD3.FR2. If the user has not created an office door yet, they must be able to create a new office door

- User must be able to click a “create office door” button if no other previous office doors are available for the user.
- The creation of the office door must be noted in the database and attributed to the specific user.

UD3.FR3. The user must be able to change the background of their new office door

- User must be able to select a different office door background from a drop down menu.
- The different choices for office door backgrounds must be large enough to cover the entire door webpage.

UD4.FR1. Widgets need to be created so the user can put content on their door

- User will have access to different types of widgets that can be selected from a modal.
- Once selected the widget will be placed in a default position.

UD4.FR1.SR1. User must be able to add widgets

- When a new widget is added it will have no previous content in it.
- If multiple widgets are added, subsequent widgets will be placed adjacent to the original.

UD4.FR1.SR2. User must be able to remove widgets

- User must be able to click a button on a widget to remove it from their door.
- The remove button must not be visible to guests or the office door owner.

UD4.FR1.SR3. User must be able to change the position of the widget on their office door

- User must be able to drag and move a widget to a different position on an office door.

UD4.FR1.SR4. User must be able to resize a widget

- User should be able to drag a corner of a widget to either enlarge it or shrink it.
- If a widget is shrunk it should show a miniature version of the original.

UD4.FR2. User will have access to a calendar widget

- Calendar widget will allow users to add events to a specific day
- Calendar widget will show user's events on a specific day

UD4.FR3. User will have access to a sticky note widget

- User can add short messages to the widget
- User can change the current message on the widget

UD4.FR4. User will have access to a notifications widget

- Widget will take in guest contact information and store it in a database
- Widget will allow guests to unsubscribe from notifications

UD4.FR4. User will have access to a picture widget

- Office door owner will be able to upload a picture to be displayed on their door
- Uploaded picture will be limited to a specific size, ex: 800px by 800px

UD4.FR5. Widgets should emphasize ease of use in regards to placing widgets and configuring an office doorr.

UD5.FR1. The tablet display will be accessible to the user via ssh on a secure network

- Using the same network or VPN, the user will be able to remotely access the tablet connected device.

UD5.FR2. User office door will be displayed on a tablet

- A user's office door will not be able to be changed via the tablet display

UD6.FR1. Frontend will communicate with a database hosted on an Amazon cloud server

- Use an architecture that allows quick and easy communication between the front end UI and the backend application.
- The back end application will need to handle get, pull, push, and delete requests.
- Cloud hosting should be free or very cheap.

UD6.FR2. User profile data will be directly saved into database tables

- Using the data from the Google Login, store user profile data in a database table.
- Data in the tables must be accessible based on a given key when a login is performed.
- User door data will be stored in structured JSON files

Nonfunctional Requirements

The following requirements make up the nonfunctional requirements, which encompass performance and environmental requirements, both of which do not go into any technical depth like the functional requirements did. Performance requirements were gathered by calculating the load and wait times of multiple websites that will function similarly to our project, such as Facebook. From those load and wait times we modified our performance requirements to acceptable times that are easily achievable with a moderate internet connection and an average computer (newer than 2005). The performance requirements will also be tested during the development cycle by means of starting a timer in the code that returns the total amount of time an operation took to complete. Those metrics will then be compared against our requirements document and if need be, changed to fit the performance requirements stated in this document. Following the performance requirements are the environmental requirements that are specified by our client's as parts of the project that are set in stone and cannot be negotiated.

Performance Requirements

UD1.PR1. Main page content population, includes user's office doors and menu options.
Loading data: 3 seconds maximum

UD1.PR2. Creation of a new user in the database during registry.
Sending data: 2 seconds for database to receive data
Loading data: 2 seconds to load empty user home page

UD1.PR3. User login through Google Login.
Sending data: 1 second to send request to server for authentication

UD1.PR4. Loading the user's door page.
Loading data: 5 seconds to load widget configurations, i.e. position, content, size

UD2.PR1. Notifications sent to a guest registered for notifications.
Sending data: send an email/SMS to a registered guest within 10 minutes of an update

UD2.PR2. Guest unsubscribing from office door notifications.
Sending data: 1 second to send delete request to guest user database

UD3.PR1. User redirection to their personal office door.
Requesting data: 1 second to receive data from the database
Loading data: 3 seconds to load the user's office door configuration

UD3.PR2. User edits made to an office door.

Sending data: 1 second to send user edits, position, content, size of widgets, and office door background to the database

UD4.PR1. Creation and setup of an office door.

Office Door Creation: Should take no more than 10 seconds to create an office door without widgets

Office Door Setup: Should take no more than 30 seconds to fully set up

UD5.PR1. Tablet door display refresh rate.

Requesting data: Should take no more than 3 seconds to populate the webpage on the door display

Refresh data: Office door display should refresh every 30 minutes

UD6.PR1. Backend database operations.

Get, Push, Pull, Delete Requests: Should take no more than 5 seconds to fulfill any backend database operation

Environmental Requirements

1. The web pages must be coded in HTML, CSS, and Javascript to be considered a web 2.0 application.
2. The display tablet that is going to be placed on an office door has to be affordable.
3. NAU Web Hosting cannot be used for this project if it is to be expanded outside of academic use, a cloud-based web service needs to be utilized.

Potential Risks

Risks and potential issues with technologies is the most overlooked area of a project. In order to address this, we took our notes from talking with our client as well as issues we ran into in our own research, and created a table of issues. The following issues are the highest priority ones that we identified that will need to be addressed towards the end of our project.

Risk	Description	Risk Level (1 – low risk, 5 – high risk)	Likelihood (1 – low, 5 – high)
Lack of usage after project completion	Once this project solution is complete and delivered, a possible lower level risk is that this project sees no use in the short time after it has been introduced to the client. To combat this risk, we need to perform user tests and surveys to see what the creators of office doors like and dislike to cater to their wants and needs.	2	3
Likeability and usability for students.	If the students, or any viewer of an office door, does not like the layout, the navigation, or anything about the application when it comes to viewing office doors, they may not use this project solution. If that becomes true, half the purpose of this project solution, easily conveying information to others, becomes useless and with it the project becomes a failure. To lower the likelihood of an issue such as this, further user tests will need to be performed to make sure all aspects of the project from an office door viewer is made as easy as possible.	3	3
Application usage outside of academia.	A very low level, but still possible, risk is the issue if this application sees use outside of academia. If this happens, we need to be sure that the technologies and ideas that we use and implement do not infringe on any copyrights of intellectual properties. This will require further research into the technologies we ultimately decide to use.	1	2
Amazon web service trial ending.	The service we have decided to host our website on is going to be done on a trial service. Once this trial service ends, and if our product is actively being used, someone needs to pay for the web hosting service or move the website to a self hosting service or another 3rd party hosting service. To combat this problem, we need to talk with either our client and see if they want to take care of it, or possibly with our Mentor and see if the Computer Science department will take care of it.	5	5

Project Plan

After we had laid out the project requirements and analyzed potential risks involved with the project we had to come up with a development plan. This was originally a word document that had our list of milestones on it and a general idea of what dates we need to complete them by. After meeting with our mentor we decided that aside from simply laying out milestones we needed to go more in depth, so we created a Gantt chart that serves as the conclusive project plan for this project.

Specifically, we created several key milestones that need to be achieved before we can officially call our project completed. These are:

MS1. Create the user login via the Google API and have a test user successfully register and sign in.

MS2. Have a user create an office door and then add widgets to it.

MS3. Create a backend application that can fulfill pull, push, get, and delete requests.

MS4. Show that user login information and other user data can successfully populate their home page.

MS5. Successfully pull data from the database to populate a user's office door page with their previous configuration.

MS6. Connect all the webpages so that they follow a designated navigation flow.

MS7. Successfully set up a physical tablet display on an office door.

MS8. Successfully display a user's office door on a tablet display.

MS9. Release a version 1.0 of the project in the beginning of March 2017.

MS10. Conduct user testing and refine the application based on feedback given.

MS11. Release the Virtual Office Door web application in May 2017.

In regards to the Gantt chart below, milestones 1 – 8 are slated to be completed in the development block, and 9-10 will be completed during the user testing block, with milestone 11 being completed at the very end of the Refinement and Release Block. The dotted line in the Gantt chart represents where we currently are in the project life cycle and the large gap that is approaching represents winter break. In our project we also have two big overlapping blocks which are development and testing. These two blocks overlap because we will be performing bug testing and fixing on our software during the development process. This allows us to output code that can be put through a testing suite, so that when it comes time to release we have functioning and bug free code. Addressing the user testing block now, this is where/when we plan on sitting students and teachers down to actually use our software and then gain feedback from them. This block is expected to shrink though if we are behind on development come next semester. The refinement and release block is time that we set aside to do any last minute bug

fixing, feedback gathering, testing, etc. which is why this coincides with release. Release will more than likely occur at the very end of that block, but before we actually release the final product we plan on having multiple rough versions before we can definitively say we have finished.

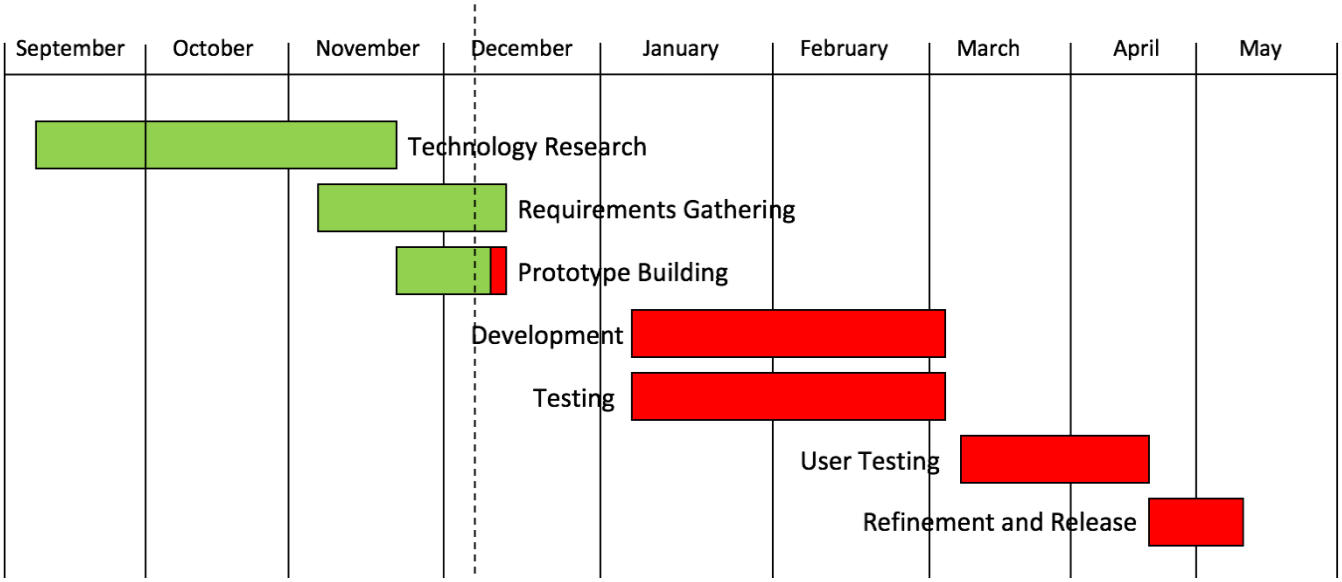


Figure 4: Virtual Office Door Gantt Chart

Conclusion

The main problem that our clients are facing is that, in academia or the professional world, time is literally money, and when someone posts urgent or critical messages on an office door these usually get overlooked, or are not seen at all. This leads to a lack of communication and organization which then leads to time being lost trying to find a fellow professional even if they are just away for 5 minutes. Our solution to this problem is to create a web application that serves as a virtual office door, where a user can post quick notes, their calendar, or whatever else they deem necessary.

In this specific document we laid out the largest component of our project, the project requirements. These requirements covered user domain level requirements, functional requirements that were created from the user domain, and then nonfunctional requirements that included performance and environmental requirements. In the beginning of the document we did lay out the problem statement as well as the solution overview which have evolved since the creation of the technical feasibility document. Then at the end of the document we covered our project plan which detailed several milestones as well as what the development plan looks like in 2017. Lastly, we covered potential risks that we will have to face going forward with this project, their level of severity and how we plan on dealing with the risks if they do pose a threat to the project.

As a whole our team is very optimistic about the completion of this project and we are making significant headway into our prototype that will be due at the end of this semester. We believe that this web application and office door display will revolutionize the way teachers communicate with students, and provide a new and more accessible medium for students to receive communications from their teachers.