Technical Feasibility

SPONSORS:
Michael Leverington, Eck Doerry

MENTOR:
Eck Doerry

TEAM MEMBERS:
James Hauser, Mitchell Hewitt, Nicolas Melillo, David Snow, Tyler Tollefson

Table of Contents

Introduction

This project, adequately titled, The Virtual Office Door is the combination of multiple disciplines as well as skill levels. Our team, comprised of James Hauser, Mitchell Hewitt, Nicolas Melillo, David Snow, and Tyler Tollefson, comes from a multitude of diverse backgrounds, each of which is crucial to the completion of this project. The project itself is to create a "virtual office door" that replaces the outdated method of posting teacher to student messages on a physical office door.

Our sponsors for this project are Eck Doerry, a computer science professor at NAU, and Michael Leverington, previously a computer science professor at University of Nevada, Reno and an incoming computer science professor for NAU.

Our sponsors' problem is that even though posting messages to students on office doors is a reliable method, there is no way to guarantee that the posted messages will arrive to the student in time. The other problem is that our sponsors constantly run into the issue of posting too many messages on their office doors, and to reduce clutter as well as promote organization, our sponsors need a streamlined solution that solves the issues of communicating with the students as well as maintaining a clean and organized workspace.

The solution that we and our sponsors envision is a web 2.0 application that is cross-platform and cross-disciplinary that enables the user to post messages on their virtual office door. The application itself will be able to support a multitude of users as well as their respective office doors. Each "door" will be able to support multiple widgets that perform different functions such as a calendar widget and a sticky note widget.

This document serves as a technical feasibility document, or proof that we have considered the different approaches to our solution and have weighed the pros and cons of each approach. In this document the different challenges and tasks that we have identified will be explained, as well as different approaches that we will take to solve the specified problems. After that, the issues that we have identified, but have not resolved yet, will be addressed as well as our approaches to solving the specified problems.

Technological Challenges

For this project the technological challenges cover a wide range of necessary technologies as well as necessary skills. This section of the document is used to list out the different technological challenges we will face and need to address. After listing the different challenges, we will go into an in-depth analysis of each problem and the possible technologies that could be used to overcome each challenge. The following table contains all the technological challenges that we thought of as a group and will be accompanied by their specific area of the project (i.e. Frontend, Backend, etc.):

Challenge	Project Area
We will need to create a user login system that either lets the user sign up for an account or sign up/sign in through a Google account.	Frontend
We will need to create a page where a user can view their door and modify its contents, widgets.	Frontend
We will need to acquire/maintain/setup a server that will contain our database.	Backend
We will need to create/maintain a database with user information related to their profile and office door.	Backend
We will need a way to either store user door information (setup, selected widgets, content) or a way to give each user their own site within the website's domain.	Backend
We will need to create a clean interface and easy to navigate UI across all the different webpages within the Virtual Office Door domain.	Frontend

Technology Analysis

In this section, we will cover the challenges we outlined above in more depth. Along with that we will present the different possible solutions to each challenge that we have researched, weigh their pros and cons and then choose a specific one based on certain criteria. After that we will outline our plan for continuing with our specific solutions and how we plan on testing each one to make sure that they are actually feasible for this project. For the last subsection, we will cover how each of the different technologies can be integrated into the final solution. Below is the first subsection, the analysis and potential solutions to each challenge:

1. We will need to create a user login system that either lets the user sign up for an account or sign up/sign in through a Google account.

   Analysis: This challenge is the more generic and one that will be found across multiple projects. We essentially need to either create or use an API to create a login system on the front page of the website that allows the user to sign up for an account through our website or using a Google account. If we were to create our own login API we would need to also be concerned with how secure the login system would be.

   Alternatives: For the login challenge we arrived at multiple different solutions. The first of which is using PHP and MySQL. For this solution, the pros are: we know PHP and MySQL, easy to implement, and free. However, for this solution the cons hold more weight than the pros: we would need to create an entire login system, we would need to build in security measures, and PHP is not secure or safe. The next solution we found was using the Google API for user login. The pros for this approach are: just one line of code to enable Google login, well documented, secure, and most everyone has a Google account (especially at NAU). The con for this approach though is that we would not be able to have a different way for users to sign up/sign in, so they would have to use a google account. The final solution that we researched involved using the Facebook login API with PHP. This approach was immediately rejected by everyone on the team just because we do not want to make people use Facebook to login, especially if this web app is going to be used in professional settings.

   Chosen Approach: For this challenge, we decided to use the Google login API. The reason for this is because of the ease of use and how secure it is. Given our current time constraints it would be difficult to create an entire login system, this way we can finish the login system with Google and then use the freed man power to finish other parts of the project.

2. We will need to create a page where a user can view their door and modify its contents, widgets.

   Analysis: This challenge is simpler than most of the others but does require experience in UI design and implementation. For this we need to create a page, or a series of pages, where the user can interact with their office door, move widgets, remove widgets, and edit widget content.

   Alternatives: For this challenge, we arrived at multiple different solutions again. The first solution that we found is jQueryUI. For this solution, there were no cons, except that we would have to write everything ourselves in Javascript; however, in respect to pros this solution has rich documentation, is free to use, and if we do more research we might be able to find libraries that we could use. The other solution that we found was webix, and for this solution the cons were that there is moderate documentation and we would have to make our project open source, and there were no pros that set it apart from the other solutions. The last solution is EasyUI, which had, again, more cons than pros. The cons being: commercial use costs money and the documentation is more of a tutorial than an explanation. The pros were that it is easy to use, living up to its name, and it had an extremely helpful user base that would answer most questions you had.

   Chosen Approach: For this challenge, we decided to use jQueryUI. The amount of flexibility and customizability that we get with this solution was too much to overlook, on top of the rich code base and the free to use nature of it. Even though it will take longer to create the UI this means that we can create exactly what we want in a decent amount of time.

3. We will need to acquire/maintain/setup a server that will contain our database.

   Analysis: This challenge requires a large amount of research and testing, mostly because we as a group haven't ever needed to deal with a physical or cloud based server in our studies. This required us to consider cloud services as well as NAU based hosting.

   Alternatives: For this challenge, we researched three different solutions that could be used for our server. The first solution was Amazon Cloud Services, provided through the same Amazon website that is used for shopping, Amazon Cloud Services pros ended up outweighing the cons. The pros being: there was a free tier for cloud services, we could use MySQL (database we are all familiar with), and there is good customer support. The cons were: we would have to learn everything about Amazon web services (AWS) and it takes a large amount of time to simply figure out how to set up some simple code for a demo. The next solution we found was Microsoft Azure. This option, unfortunately, only had tiers of service that you had to pay for, there was no free tier, this outweighs most pros simply because for this project we do not want to spend money on a service when there are others that are free. The last solution we researched was the servers at NAU. The pros for this solution were that it is local, relatively easy to use and we could

possibly integrate with PeopleSoft. However, the major con was that by using the NAU servers we wouldn't easily be able to distribute this app elsewhere.

Chosen Approach: For this challenge, we decided to use the AWS cloud server solution. This solution might be harder to approach but it is more beneficial to the longer-term status of this project. We were also able to acquire a free 12-month trial which contributed a lot to the decision.

4. We will need to create/maintain a database with user information related to their profile and office door.

   Analysis: This challenge utilizes our groups experiences more than most of the other challenges and we thankfully have experience when it comes to creating and maintaining databases. Also, this challenge does require us to consider the performance of our solutions, unlike most of the other challenges.

   Alternatives: For this challenge, we researched the different database languages and DBMs out there that we could use. The first of which was simple MySQL, this solution easily integrates with our AWS cloud server and is a database language/manager that we all have had experience using. The cons are that MySQL is not as mature as other relational DBMs and it is not community driven. The next solution we had was NoSQL. In regards to NoSQL the pros are favorable: it supports huge processes and it has elastic scaling. However, the cons are that it is still in its infancy, there is a lack of support, and it requires a large amount of expertise. The last solution we have is Amazon DynamoDB, but this solution implements a NoSQL database which is already not favorable; however, because this solution is already integrated with Amazon it would work favorably for us.

   Chosen Approach: For this challenge, we decided to use MySQL. The main reason for this is because it is heavily documented, it is in the realm of technologies that we all as a group know, and it is completely free. If it is determined that going with a NoSQL database would be more beneficial then we would reconsider, but for the scope of this project now we are going to implement a MySQL database.

5. We will need a way to either store user door information (setup, selected widgets, content) or a way to give each user their own site within the website's domain.

   Analysis: This is one of the challenges that we haven't looked that deep into yet but one that still needs to be addressed because of how crucial it is. Either approach to this challenge, storing user information or giving a user a specific website address, will be challenging and needs to be considered before the end of this semester.

   Alternatives: Because of the lack of research into this solution we only know a few options of what we could possibly do to overcome this challenge. The first of which is by storing user information in our database, it would be simple to get the information and

then populate a “my door” page with all that information. However, if we go the route of giving the user a specific domain we can simplify the user table in the database and split it into a user table and a door table, where the door table simply contains widget information, positions, etc. In that approach, we would just need to query the database based on the given URL for the user.

Chosen Approach: For this challenge, we have not yet decided on an approach.

6. We will need to create a clean interface and easy to navigate UI across all the different webpages within the Virtual Office Door domain.

   Analysis: This challenge will be the most time consuming and taxing out of each of them. It will require that we create webpage mockups and drawings to get the website design to conform to how the clients envision it. This is also where most the coding will happen, besides from coding the communication between the frontend and backend.

   Alternatives: In our research, we came across a few solutions, the first of which was AngularJS. The pros to AngularJS are: development is fast and simple once familiar with the language, easily testable, and it extends HTML. The cons are: very steep learning curve, weak documentation, and most of the features are complex. The next option we considered was just using plain Javascript, CSS, and HTML. The pros to this solution are: we can build basically whatever we want how we want it, the documentation is extensive, and we all have experience coding in Javascript, CSS and HTML. The cons are: we would need to code everything from the ground up, lack of security and we rely on the user having scripts enabled in their browser.

   Chosen Approach: For this challenge, we decided to use Javascript, HTML and CSS. We made this decision based on the main idea that we all know how to code in those three languages. So, if someone is falling behind in a certain area others can easily jump in and assist them. Also with the amount of documentation there is out there we can research different approaches and possibly find libraries that will make creating the functionality of the website simple.

Proving Feasibility

In this subsection, we will cover our plans on how we decide to prove that the solutions we chose will work for our project. Also, we will outline our current demos and what the future demos will look like, specifically the ones that will be related to the demo we are to do later in this semester.

In regards to testing our Google login API, for the demo, after we have a mock database created, we will basically show that it is possible to log into the website via the Google login. When it comes to demoing the frontend technology we chose, JQueryUI, we already have a demo showing some very basic functionality. For the main demo, we are going to create a

calendar widget that can be moved around on the page, collapsed, expanded, and have content added into it. To demo the use of the AWS cloud server we already have two simple demos, however, for the main demo we are going to show that we can host our database on the cloud server and possibly do some benchmark testing. This ties into testing MySQL, for this we will create a mock database table which will be populated either by random user generated information or contain information related to user logins. When it comes to the challenge of storing user information that demo will be done concurrently with the database demo. The final challenge, creating the entire web space's UI and navigation can be demoed by having a few pages drafted out and then we navigate to them and show the different pages we made.

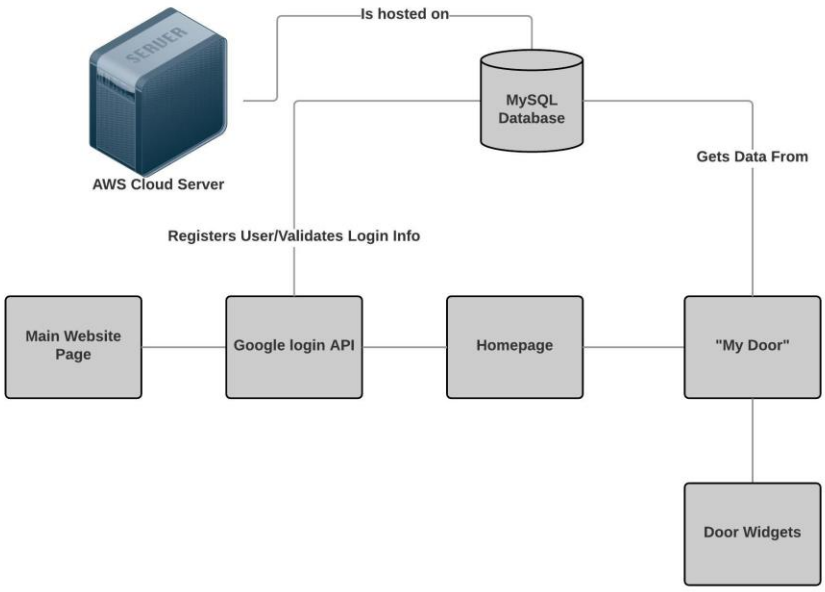
Technology Integration

We will cover the overall integration approach for our project and all the mini-solutions in this section. The section will start by listing out the different major challenges that we have, then it will be followed by a diagram, displaying how each challenge relates to each other. Then in the final part of this subsection we will discuss the different integration issues that might arise.

List of Challenges:

1. Creating a login system
2. Creating widgets to be used on the virtual door
3. Acquiring a cloud server to host our database
4. Creating and maintaining a database
5. Storing user door information
6. Creating a clean and easily navigable frontend

Integration Diagram



Some possible integration issues do exist in this model though. The first main integration issue is that if the Google API does not integrate with AWS. This could easily be solved just by creating demoes and testing to see if they can work together. The other integration issue that we could encounter is with the MySQL database not being compatible with AWS. This issue could easily be resolved by either finding a workaround or switching to a NoSQL database or another type of database that would be compatible with AWS. The last integration issue that we identified would be that of how we plan on getting door data from the database. Depending on the tool that we eventually use there could be issues loading the data.

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 regards to this document the main purpose is to discuss the different technological challenges that we as a team face, analyze them, and then discuss our solutions and prove that we have a plan that is feasible. This document also serves as a milestone and the precursor to our requirements document that will be crucial before we start the bulk of the project work. To summarize the document the below table contains the different challenges we will face, our proposed solutions and our confidence level in each (on a scale from 1-5, 5 being most confident 1 being least confident). After the table, we will summarize and then address any open questions.

Challenge	Solution	Confidence Level
We will need to create a user login system that either lets the user sign up for an account or sign up/sign in through a Google account.	Google login API	5
We will need to create a page where a user can view their door and modify its contents, widgets.	jQueryUI	4
We will need to acquire/maintain/setup a server that will contain our database.	Amazon Web Services cloud server	4
We will need to create/maintain a database with user information related to their profile and office door.	MySQL	4
We will need a way to either store user door information (setup, selected widgets, content) or a way to give each user their own site within the website's domain.	Unknown	1

We will need to create a clean interface and easy to navigate UI across all the different webpages within the Virtual Office Door domain.	Javascript, HTML, CSS	5

In summary, this document serves as a stepping stone to the completion of our project, and hence will be evolving with the project as well. We have solutions for all but one technical challenge, of which we just need to do more research to find a feasible solution. In regards to the challenges that we do have solutions for we are confident that the proposed solutions will work and integrate with each other. Although if a more promising solution would arise we would not hesitate to explore that option.

As of this moment we have most of our questions answered; however, we are constantly running into parts of the project that we did not realize had existed before, so questions always arise. The major concern that we have as a group right now concerns the descoping of the project. Given time constraints we will be able to finish the main parts of the project, but we need to communicate with the client more to narrow down their requirements and make it so that we can successfully create a product that the client can use to its fullest extent. We know that our analysis so far of the problem and our approach to a solution has been thorough and addresses what we understand to be the correct way to go about the solution. This feasibility document proves that we have given the problem, process and our solution thought but are constantly asking questions to make sure we deliver a solution to the client that is as close to what they envision as possible. In the future, this project and solution could not only be used in academia but it could be expanded for use in the professional world where it could save businesses thousands of dollars.