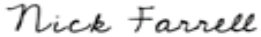
Howdy Neighbor

by

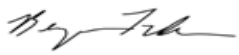
Nicholas Farrell, Benjamin Tenhover, Sergio Berrios and Evan Batsch

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

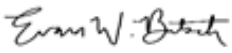


Nicholas Farrell Date 4/12/2021

Sergio Berrios Date 4/12/2021



Benjamin Tenhover Date 4/12/2021



Evan Batsch Date 4/12/2021

Bander Alyami, Faculty Advisor Date:4/12/2021

University of Cincinnati
College of
Education, Criminal Justice, and Human Service
April 2021

Table of Contents

List of Illustrations

Abstract

The Howdy Neighbor web application is an online resource designed for anyone planning a move. While moving, which the average American will do over 11 times during their lifetime (Bureau, 2020), there are many factors that may be overlooked, which could positively or negatively affect the outcome of the move. Some of these factors include area age demographics, nearby crime statistics, and local school district information. As well as obtaining all of this important information, one must also remember to set up necessary utilities for the new home prior to moving in. By using Howdy Neighbor, movers will have access to a virtual checklist to use during the home search process, as well as a pre-move-in task reminder checklist to use prior to the move. Once these checklists are complete, the user will be able to move to their new home with the confidence and peace of mind they deserve.

Introduction

Introduction

While deciding to move to a new home can be exciting, the stresses and difficulties that can arise are inevitable. This can cause many to overlook important factors that can seriously impact the overall moving experience. During the home search process, as well as prior to moving into a new home, there are many steps that need to be completed to ensure a successful move. Our goal is to help those relocating by providing virtual checklists that users can work through to make sure that nothing goes forgotten, while both increasing confidence and decreasing stress during this extremely difficult process.

Problem

When moving into a new home, there are many factors that can affect one's new living experience. Some of these factors include the area's age demographics, access to parks, stores, and banks, local crime statistics, and the school district the home is located within. Then after deciding on a new home to move into, the mover will also need to complete many important pre-move-in tasks, such as contacting the electric company, obtaining renter's or homeowner's insurance, or scheduling moving services, to ensure they have a successful move.

Our research showed that "the average American moves 11.7 times in their lifetime while nearly 31 million people move each year" (US Census Bureau, 2020). Due to the stressful nature of moving, many of these factors or important tasks may often be overlooked. Although there are resources currently available which provide some of this vital information to potential movers, such as Zillow and Trulia, these websites focus more on the sale of these homes.

Obtaining information from these websites can sometimes be difficult, as well, considering some of these websites may have neighborhood data that other similar websites do not have. Having access to an easy-to-use, complete, virtual moving guide that focuses solely on the mover's experience would make the entire moving process much easier, while ensuring confidence in the decision to move to a specific area.

Solution

Howdy Neighbor is a responsive web application in which users who may be interested in moving to a new area can work their way through a virtual checklist that covers different factors that may affect their move. Some of these factors include area age demographics, local crime statistics, local school district information, and essential locations that users may need to locate after moving (banks, supermarkets, pharmacies, hospitals, gas stations, etc.). Then once the user has selected a home to move into, Howdy Neighbor will also provide a customizable checklist that will consist of reminders to complete certain pre-move-in tasks, such as calling to set up electric in your new home, scheduling moving services, or obtaining renter's or homeowner's insurance. Once a user has completed the Howdy Neighbor checklists, the user will be able to move with confidence, while having the peace of mind knowing that nothing will be forgotten or overlooked during this stressful process.

Project Goals

The goal of our project is to provide anyone moving to a new home the ability to create and organize virtual checklists containing important factors that may impact their move, as well as important move-in reminders. Users will be able to obtain information about the area they

are interested in moving to, which could affect their everyday lives, such as the area age demographics, crime statistics, and nearby locations to visit. They will also be able to rate homes based on these factors, providing the ability to easily compare searched homes. By providing this all-in-one, easy-to-use web application, we hope to relieve some of the stress of moving, while providing confidence in the decision to move to their newly selected home.

Overview

Throughout this report we will cover the processes used to create and design the framework for our Howdy Neighbor Web Application. The report includes our design objectives, methodology, budget, timeline, problems encountered, and future upgrades.

Discussion

Project Concept

In our initial research for the requirements analysis, we compared our project concept with existing websites that respond to similar problems, such as Zillow, Trulia, and Realtor.com. While these services allow users to search for properties within certain neighborhoods, as well as offering some rudimentary information pertaining to these neighborhoods, oftentimes, one of these websites will have certain information that the others do not. We have discovered that you have to navigate to multiple websites in order to obtain all the information you are interested in while searching for a new home. After some research, we have determined that the main focus of these resources is on the sale of the homes. Howdy Neighbor will focus solely

on the mover's experience by providing big-picture overviews of areas of interest, while also providing reminders leading up to the actual move-in.

Design Objectives

Our design objective has changed from our initial idea, but not substantially. We approached the problem by trying to centralize everything a user would need to ensure a successful move. However, we quickly found out that other resources like Zillow and Realtor had something like our design. We researched solutions and different ways that we could separate the project from being like those other resources.

We decided to develop a web application from scratch using Visual Studio and writing the code ourselves. We reviewed cloud hosting options such as Azure but determined for our initial implementation use a dedicated physical server to host the web server and database. We investigated using Docker to have a uniform environment for testing and developing the web application, but we realized that using Docker would add a significant amount of complexity to our workflow without providing much benefit since the application's environment was easy to replicate within Visual Studio. We decided to use the GitHub repository to organize our project files, images, and data and for our team to be able to collaborate and contribute to the project all in the same space. Using these technologies will continue to help our project in the future.

Methodology and Technical Approach

Responsive Web Applications

The goal for the website was to have all the information the user would need in a centralized location. After some research and deliberation, we decided to use Visual Studio for

development. Visual Studio is an IDE from Microsoft that is used for developing software applications, specifically with languages and frameworks such as C# and ASP.NET The Web application will include some of the following features:

- Users will be able to obtain information relating to the area or neighborhood that they plan to move into.
- The web application will be optimized for mobile usage as well as desktop usage.
- Users will be able to rate specific factors of searched addresses, which they can then save and return to at a later point.
- Users will be able to customize their own pre-move-in reminder checklist to include any tasks they need to complete.

Dedicated Server

Our team has decided to use a dedicated physical server for hosting of the web server and SQL server. This allows us to keep hosts low for our initial deployment with a dedicated system that has the required hardware to process the requests with no issue. Any changes we made could be done instantly and server could be online at all times with no cost.

Coding Languages

The web application is programmed in C# using the ASP.NET Core framework. Specifically, it uses ASP.NET Core's Razor Pages framework. Within the application, the code for individual pages is written in CSHTML along with regular C# code. Web pages are rendered in HTML, CSS, and JavaScript within the browser. HTML provides content for the page, CSS provides the styling, and JavaScript provides interactive functionality.

User Profiles

The user profiles below describe the three types of people who are most likely to use our application – someone who is planning to move to a new area (Figure 1), someone who is looking to buy a home (Figure 2), and someone who will act as the web developer (Figure 3). Following these user profiles has helped guide the development of our application in determining which features are relevant, as well as proving why we need to maintain a simplistic design throughout the application.

User Profile Form
Potential User: Someone looking for a new home.
Application Used: Howdy Neighbor Web Application
Software and Interface Experience: Has experience with basic internet usage, as well as using other moving website interfaces, such as Zillow.
Related Application Experience: Has experience in other web applications, such as Zillow and Realtor.com. Also has experience with popular websites, such as Facebook and Google. Has lots of experience using common internet browsers, such as Google Chrome, Internet Explorer, and Firefox.
Key Task Experience: Basic website navigation as well as using computer accessories and peripherals, such as mice and keyboards. Has no experience moving to a new home though and does not know how to begin the process.
Key Goals: Would like to obtain information, such as age demographics, area school districts, crime statistics, etc. about a new home in an unfamiliar area.
Frequency of Use: When looking for a new home, the user will use this application as much as needed to feel comfortable with their decision to move to their desired location. The user will use this

application to find a desirable location until the user is satisfied and has made the decision to make the move.
Key Interface Design Requirements Required Based on Profile: Since the user has never moved to a new home and is unaware of how to begin the process, the website should be very user-friendly, straight-forward, clear, and easy-to-navigate.

Figure 1: User profile 1

User Profile Form 2
Potential User: Someone moving into a new home.
Application Used: Howdy Neighbor Web Application
Software and Interface Experience: Has experience with basic internet usage, as well as using other moving website interfaces, such as Zillow.
Related Application Experience: Has experience in other web applications, such as Zillow and Realtor.com. Also has experience with popular websites, such as Facebook and Google. Has lots of experience using common internet browsers, such as Google Chrome, Internet Explorer, and Firefox.
Key Task Experience: Basic website navigation as well as using computer accessories and peripherals, such as mice and keyboards. Has no experience moving to a new home though and does not know how to begin the process.
Key Goals: Considering the difficulty and stress of moving, user would like to make sure nothing is overlooked or forgotten during, before, and after the move.
Frequency of Use: The user will use the application after completing any of the reminder tasks to check off the boxes and ensure everything is taken care of for a successful move. After moving, the user will no longer use the application.
Key Interface Design Requirements Required Based on Profile: Since the user has never moved to a new home and is unaware of how to begin the process, the website should be very user-friendly, straight-forward, clear, and easy-to-navigate.

Figure 2: User profile 2

User Profile Form 3
Potential User: Web App Developer/System Administrator
Application(s) Used: Visual Studio, GitHub, Microsoft IIS, Notepad++, Microsoft SQL Database
Software and Interface Experience: User has experience in C#, HTML, CSS, JavaScript, as well as database knowledge. Helped develop the web application and is, therefore, familiar with the project. Also has experience using GitHub for source control and Visual Studio for an IDE and can navigate and modify data in the SQL Database.
Related Application Experience: User has experience in other coding languages, such as Java. Also has experience using SQL server for database needs.
Key Task Experience: Helped develop the Howdy Neighbor web application using the mentioned coding languages and software. Also helps maintain the website and monitor any malicious activity.
Key Goals: Maintain and continue improving usability throughout the Howdy Neighbor web application.
Frequency of Use: The user will check the web application daily to make sure everything is functioning properly and there are no bugs/issues occurring. The user will also continue to improve the website as needed, based on user feedback and new ideas.
Key Interface Design Requirements Required Based on Profile: This project will need to continue being written using the same coding languages and software, in which the user is familiar with. This will ensure a well maintained, secure web application.

Figure 3: User profile 3

Use Case Diagram

The Use Case Diagram below (Figure 4) describes how the application will be used by our target audience as outlined in the User Profile section, as well as by system administrators who are responsible for managing the application.

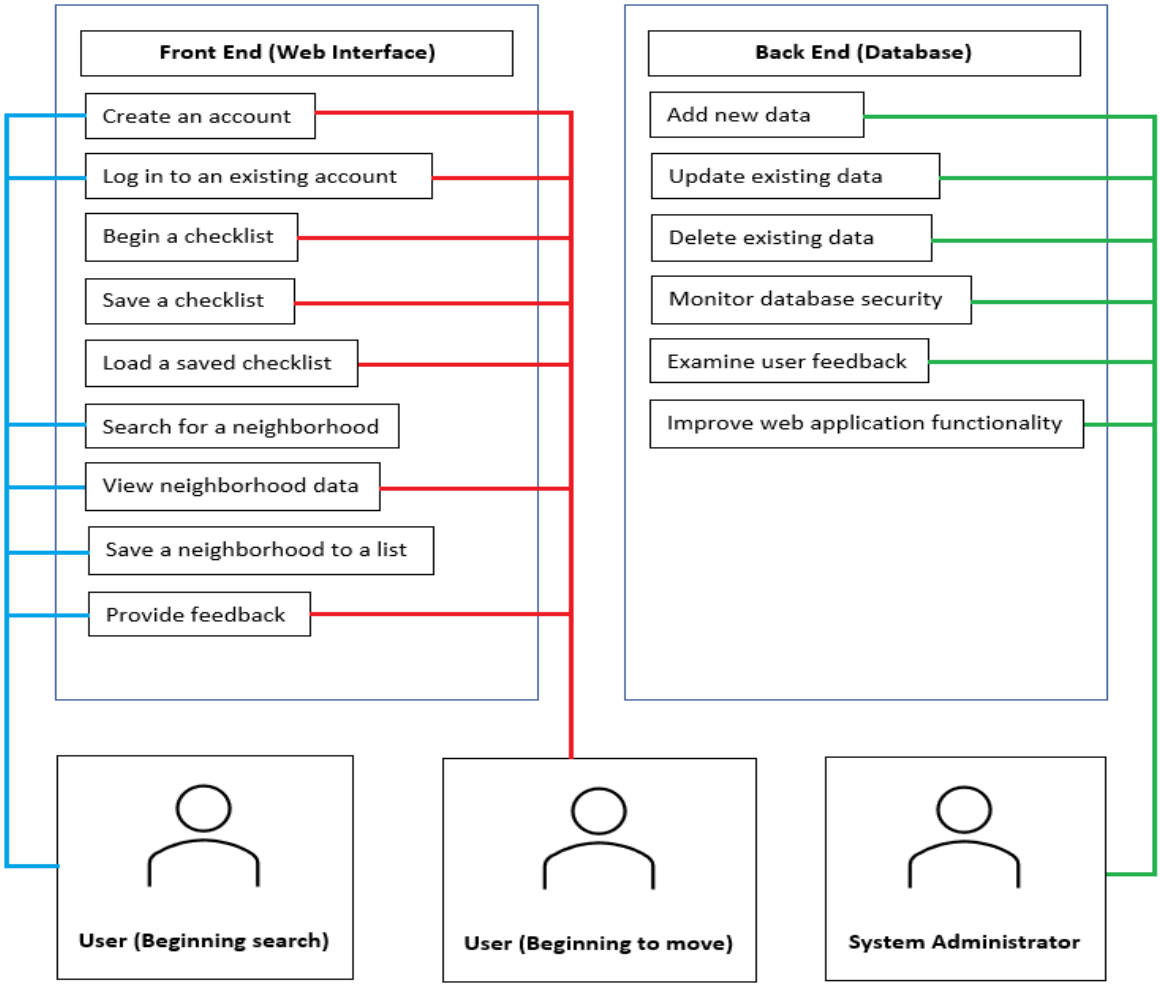


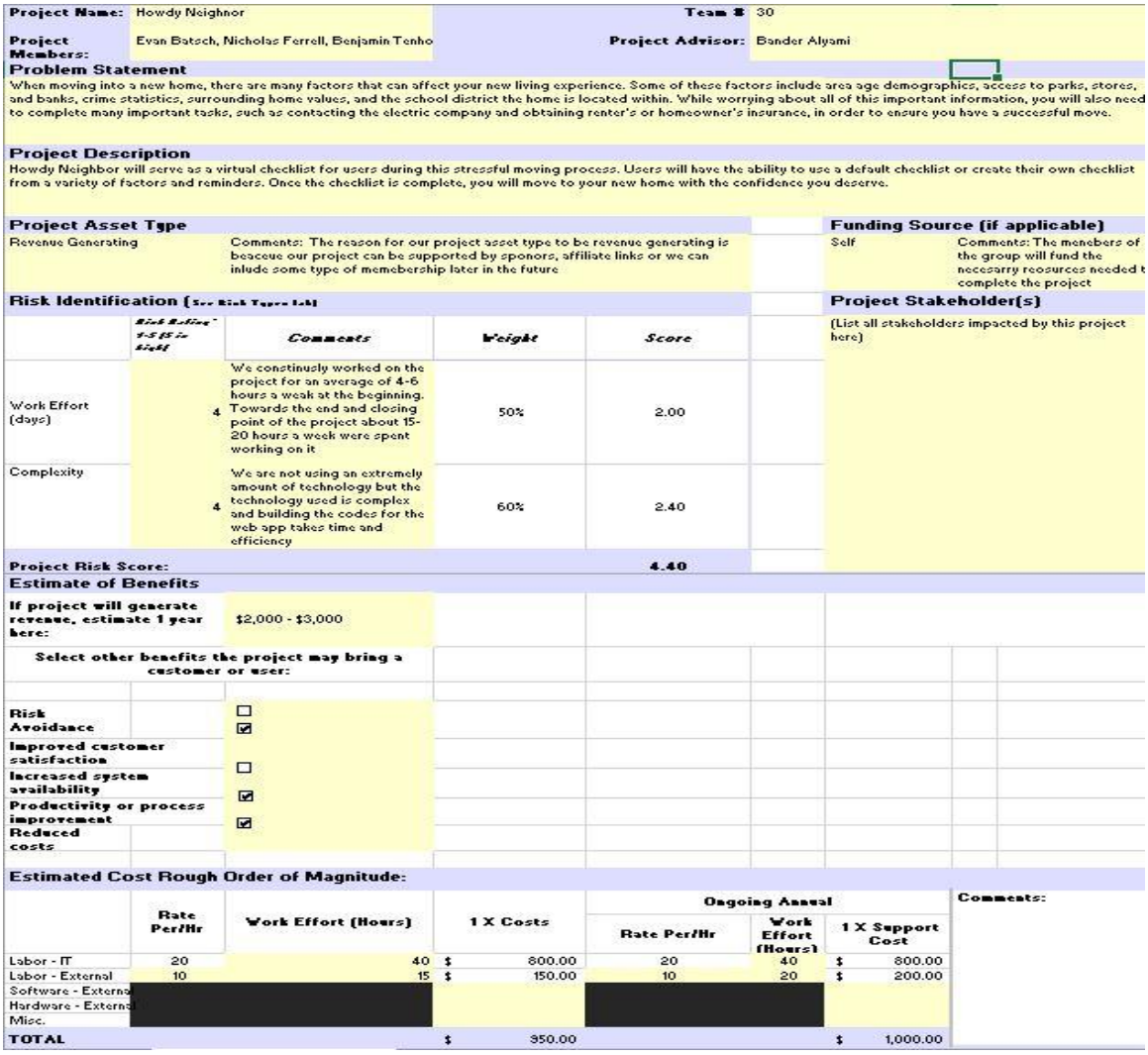
Figure 4: Use case diagram

Budget

The below figure (Figure 5) shows the budget for our project, as well as Risk Assessments and a 5-ROI analysis.

The budget we originally planned stayed consistent throughout the entire project. As a team, we discussed using free APIs and services in order to provide the information the user would need in order to have a successful move. As we progressed with our project, while taking into consideration some of the feedback we had received regarding how we would support our

web application if it continues to be a free service; we have been considering the idea of adding a one-time payment for users that would like to sign up with an account and save their checklists. We also considered using affiliate links and including ads from moving companies to support our application, but if we decide to go either of these routes, this will be applied in the future.



Project Name:	Howdy Neighnor		**Team #**	30
Project Members:	Evan Batsch, Nicholas Ferrell, Benjamin Tenho		**Project Advisor:**	Bander Alyami

Problem Statement

When moving into a new home, there are many factors that can affect your new living experience. Some of these factors include area age demographics, access to parks, stores, and banks, crime statistics, surrounding home values, and the school district the home is located within. While worrying about all of this important information, you will also need to complete many important tasks, such as contacting the electric company and obtaining renter's or homeowner's insurance, in order to ensure you have a successful move.

Project Description

Howdy Neighbor will serve as a virtual checklist for users during this stressful moving process. Users will have the ability to use a default checklist or create their own checklist from a variety of factors and reminders. Once the checklist is complete, you will move to your new home with the confidence you deserve.

Project Asset Type

Revenue Generating

Comments: The reason for our project asset type to be revenue generating is because our project can be supported by sponors, affiliate links or we can inlude some type of memebership later in the future

Funding Source (if applicable)

Self

Comments: The menebers of the group will fund the necessarry reosurces needed t complete the project

Risk Identification (See Risk Types List)

	Risk Rating 1-5 (5 is high)	Comments	Weight	Score
Work Effort (days)	4	We constinusly worked on the project for an average of 4-6 hours a weak at the beginning. Towards the end and closing point of the project about 15-20 hours a week were spent working on it	50%	2.00
Complexity	4	We are not using an extremely amount of technology but the technology used is complex and building the codes for the web app takes time and efficiency	60%	2.40
Project Risk Score:				4.40

Project Stakeholder(s)

(List all stakeholders impacted by this project here)

Estimate of Benefits

If project will generate revenue, estimate 1 year here: $2,000 - $3,000

Select other benefits the project may bring a customer or user:

Risk Avoidance	☐
Improved customer satisfaction	☑
Increased system availability	☐
Productivity or process improvement	☑
Reduced costs	☑

Estimated Cost Rough Order of Magnitude:

	Rate Per/Hr	Work Effort (Hours)	1 X Costs	Ongoing Annual Rate Per/Hr	Ongoing Annual Work Effort (Hours)	Ongoing Annual 1 X Support Cost
Labor - IT	20	40	$ 800.00	20	40	$ 800.00
Labor - External	10	15	$ 150.00	10	20	$ 200.00
Software - External						
Hardware - External						
Misc.						
TOTAL			$ 950.00			$ 1,000.00

Comments:

Figure 5: Budget

Technical Architecture

Howdy Neighbor is a web application developed in C# and ASP.NET using the Razor Pages framework. Its functionality is separated into frontend and backend layers which interact with a database, along with external APIs. The diagram below (Figure 6) provides an overview of the application's technical architecture, showing how the frontend and backend layers interact with the database and APIs.

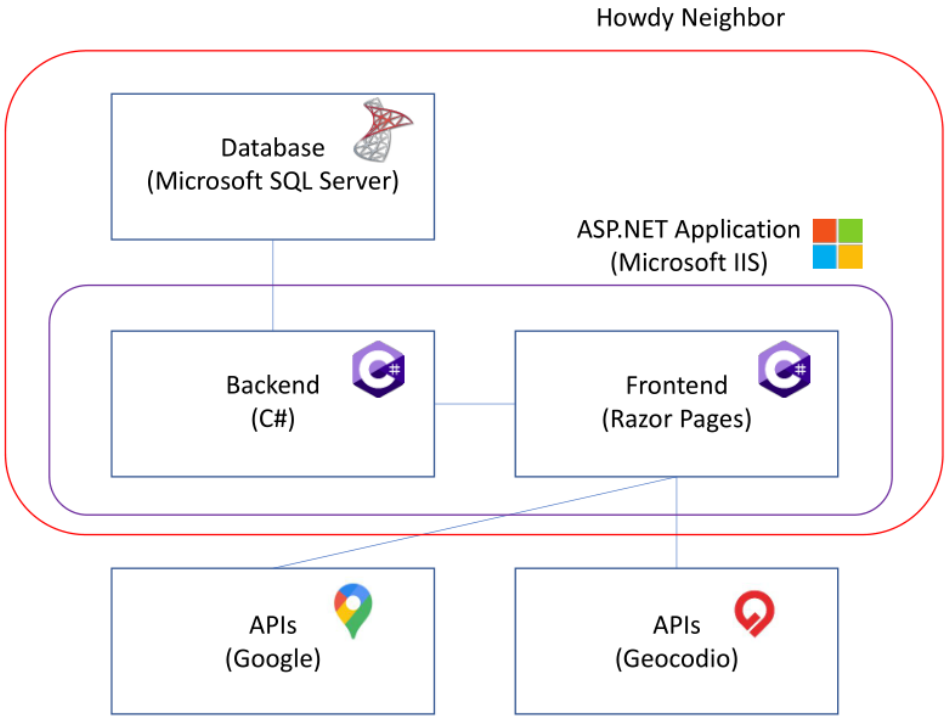


Figure 6: Technical architecture diagram

The frontend consists of web pages which are rendered from Razor Pages templates. These pages contain logic to interact with the application backend and external APIs, along with markup and styling to display the data from these sources.

Various API services provided by Google and Geocodio are called by the frontend to retrieve data for the user. API requests are structured using data that the user submits through forms on the frontend, and once this data is retrieved, it is displayed on the frontend in the form of tables or charts an example of the demographic can be seen below (Figure 7).

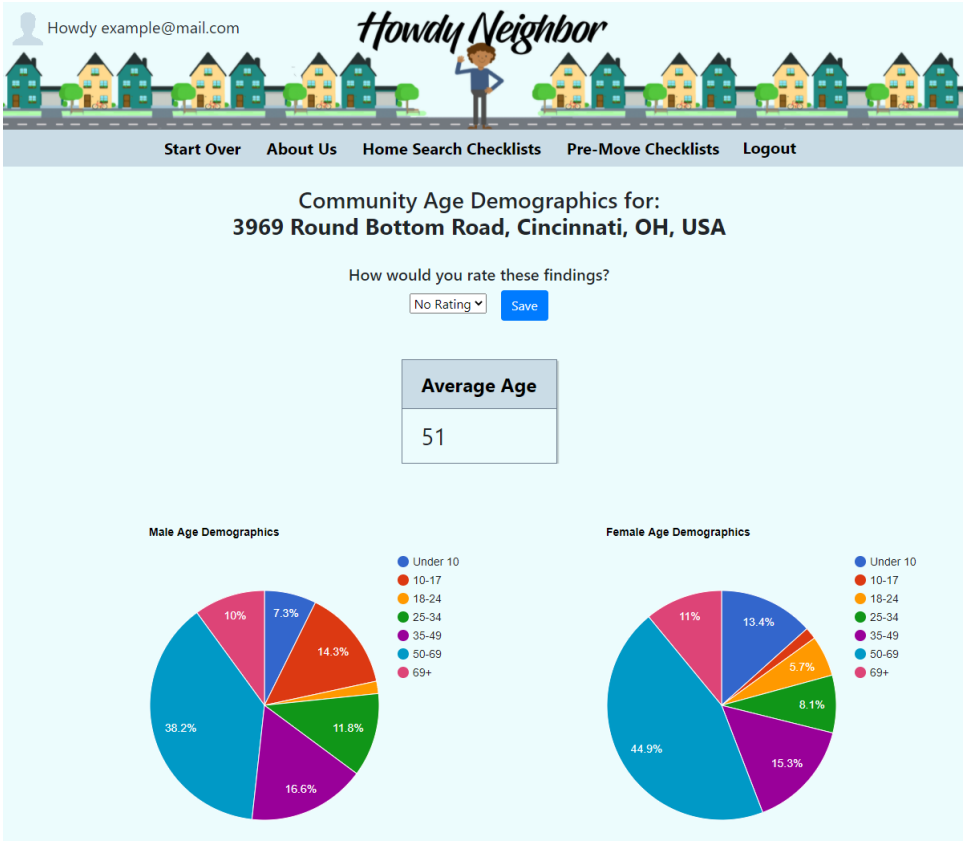


Figure 7: Front end screenshot

The backend database is an instance of Microsoft SQL Server which is used to store data that the user has entered or searched for, such as addresses and checklists. The application itself is served through Microsoft IIS. Backend application code is used to map operations between the frontend user interface and the tables stored in the database, as well as to provide

authentication so that users are only able to interact with data that has been saved under their own credentials. All this data is stored in an SQL table and an example of a user record can be seen below (Figure 8).

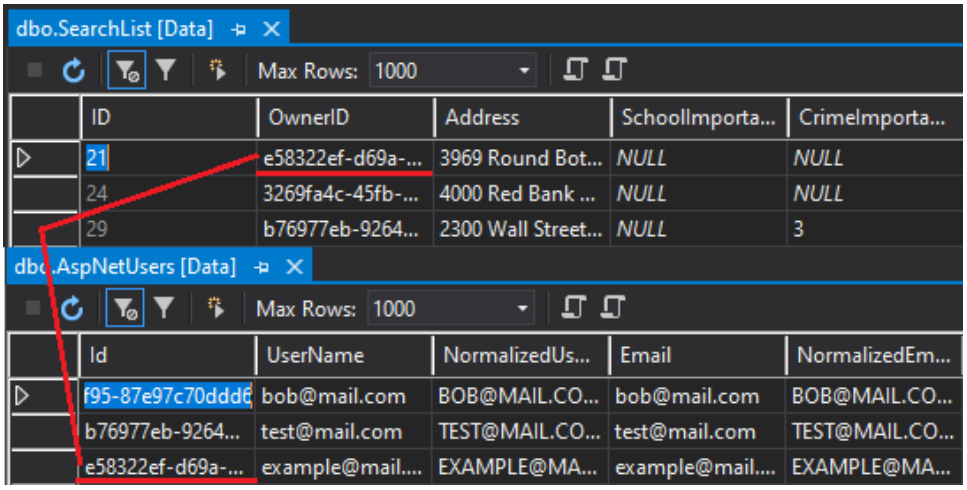


Figure 8: Database screenshot

Testing

Testing Overview

Howdy Neighbor is an online tool that allows users to search addresses that they are interested in moving to, in order to gain knowledge about the area, as well as being able to create a pre-move task reminder checklist prior to their move. This will provide users with peace of mind while going through the incredibly stressful moving process, by ensuring no factors are forgotten.

Since our application is aimed at simplifying the moving process, our focus was to provide an easy-to-use, simplistic website for our users to interact with. Because of this, a lot of

our testing focused on the usability factors associated with our application. Some of these factors included color scheme choices, website layout, and overall user satisfaction.

On the outside, we aimed to create an application that was simple and easy-to-use, while also being pleasant to use. However, we also had many important functionalities that also needed to be tested, in order for us to confirm everything was working correctly under the many different scenarios that different users may be in, while moving.

Testing Methodology

Our usability testing approach was to have close friends and/or relatives of our team use and interact with the Howdy Neighbor web application, while members of our team observed and recorded these interactions. We gave these users minimal guidance, which helped us gauge the ease-of-use of our application. By observing the users during this interaction, we were able to make note of any frustrations or difficulties they had encountered.

As well as the observations we made by watching users navigate our application, we also provided them with a couple usability forms to fill out directly after their interaction, which helped us to gain a better understanding of which changes were the most critical to address.

The first form we gave to our users was a System Usability Scale, also known as an SUS (Affairs, 2021). The questions in this form are shown below:

SUS Form for Howdy Neighbor

(Circle or Check if True)

1. I think that I would like to use this website frequently if I am planning to move.
2. I found the website unnecessarily complex.
3. I thought the website was easy to use.
4. I would need technical support to be able to use this website.
5. I found functions in this website integrated.
6. Too much inconsistency in this website.

7. Most people would learn to use this website very quickly.
8. I found the website very cumbersome to use.
9. I felt very confident using the website.
10. I needed to learn a lot of things before I could get going.

The next form we provided to our users to fill out was the AttrakDiff form. This was another valuable usability tool that we used in order to help understand how users were feeling while using our application. This form is shown below (Figure 9):

AttrakDiff Form for Howdy Neighbor

human	○	○	○	○	○	○	○	technical
isolating	○	○	○	○	○	○	○	connective
pleasant	○	○	○	○	○	○	○	unpleasant
inventive	○	○	○	○	○	○	○	conventional
simple	○	○	○	○	○	○	○	complicated
professional	○	○	○	○	○	○	○	unprofessional
ugly	○	○	○	○	○	○	○	attractive
practical	○	○	○	○	○	○	○	impractical
likeable	○	○	○	○	○	○	○	disagreeable
cumbersome	○	○	○	○	○	○	○	straightforward

Figure 9: AttrakDiff form

Aside from having users navigate through the app and fill out these usability forms, we also implemented unit testing within our code. This allowed us to test the more technical aspects of the program to make sure that everything was working as expected, and to eliminate bugs which may had otherwise gone unnoticed. We were able to add unit testing classes and methods to the project in a straightforward way, and Visual Studio made it easy to run these tests at compile-time.

Scope of Testing

The web application test of Howdy Neighbor included the testing of the different functionalities available. This included searching for new addresses in order to obtain area information, as well as viewing the data provided for each “factor” that was available given a searched address. These factors included Local School District Information, Local Crime Data, Area Age Demographics, and Local Traffic Data. The user was then able to rate these factors based on personal preference while working through and completing the checklist containing these factors. After this, the user was able to work through a pre-move task reminder checklist, while customizing the tasks within the checklist to fit their personal moving needs.

Aside from testing the functionalities, the web application test also focused on the ease-of-use of our application. We had aimed to create a simple design and layout that is still pleasant to look at. We gauged what our users thought in this regard and made any necessary adjustments.

The unit testing of our application was implemented mainly for actions where the user was submitting data to the application, or where the application was returning data to the user. These tests were as simple as checking whether the data in question was successfully able to be sent or received. Testing these parts of the application were important, because if they were regularly successful, it showed that the frontend and backend code was working as intended. On the other hand, if these tests failed, it was a clear indicator that there were bugs in the frontend or backend which could be tracked down and fixed.

Testing Objectives

Our strategies in testing included identifying any remaining bugs in our code, so that they could be resolved, as well as seeing how users interact with and navigate our application so that necessary changes could be made to the user interface.

Objectives included:

- All use cases should reflect the two primary user roles that we identified in the planning process: users who are searching for a neighborhood to move into, and users who have already found a neighborhood and are beginning the actual moving process.
- Resolve any remaining bugs, particularly in feature-heavy areas of the application. The Search Checklist and the Moving Checklist will be the focus of attention, as these are intended for the two primary user roles for our application.
- Identify any UX changes that need to be made based on our observation of users interacting with our application, as well as the usability forms filled out by the users.
- Make use of Unit Testing and CI/CD to identify bugs at compile-time and when commits are made to branches on GitHub.
- Identify any errors that may occur, such as an invalid address being searched by the user. While identifying these errors, we can make sure we return user-friendly, readable text to our user to inform them of the error.

The end goal of our testing was to ensure that every aspect of our application is as easy-to-use as possible, while still providing accurate data for any given user. While being easy-to-use, our application must also be visibly appealing to our users. By following through on this

testing plan and completing our testing objectives, we have been able to provide the best version of the Howdy Neighbor web application possible.

Test Logs and Procedures

Regarding the System Usability Scale, each of us had two family members or friends work through these questions after interacting with our website, for a total of eight users. The following graph displays the results for this testing (Figure 10). Since the questions are not displayed within the graph itself, the questions are provided below, followed by the graph representing the results:

1. I think that I would like to use this website frequently if I am planning to move.
2. I found the website unnecessarily complex.
3. I thought the website was easy to use.
4. I would need technical support to be able to use this website.
5. I found functions in this website integrated.
6. Too much inconsistency in this website.
7. Most people would learn to use this website very quickly.
8. I found the website very cumbersome to use.
9. I felt very confident using the website.
10. I needed to learn a lot of things before I could get going.

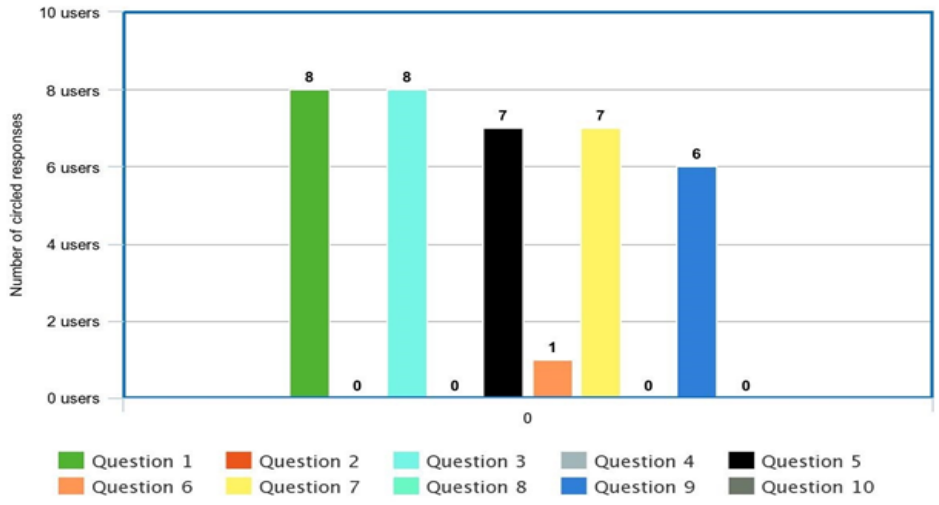


Figure 10: Testing log results

Regarding the AttrakDiff testing form, each of us also had two family members or friends work through these questions after interacting with our website, for a total of eight users. The following shows the form we provided our users, followed by the results we received (Figure 11):

human	○	○	○	○	○	○	○	technical
isolating	○	○	○	○	○	○	○	connective
pleasant	○	○	○	○	○	○	○	unpleasant
inventive	○	○	○	○	○	○	○	conventional
simple	○	○	○	○	○	○	○	complicated
professional	○	○	○	○	○	○	○	unprofessional
ugly	○	○	○	○	○	○	○	attractive
practical	○	○	○	○	○	○	○	impractical
likeable	○	○	○	○	○	○	○	disagreeable
cumbersome	○	○	○	○	○	○	○	straightforward

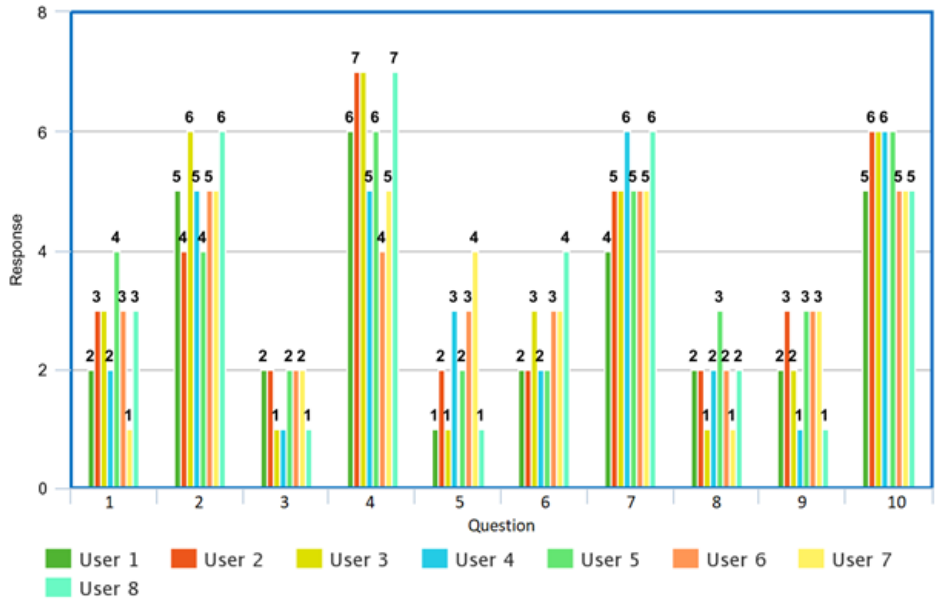


Figure 11: Testing log form and results

Aside from the forms we had users complete after interacting with our web application, we also observed the users interacting with our website. The results from this observation were overall very good, as everyone was able to navigate the website easily and figure out how to view data and create custom move-in reminder checklists.

Unit Testing

In addition to UI testing, we implemented unit testing functionality within the codebase. For this, we decided to use the xUnit unit testing library. xUnit is a free and open-source project that is used by developers to add unit tests to .NET applications. These unit tests were kept largely separate from the rest of the codebase, which was useful for avoiding cluttered code and managing inconsistencies across files.

To implement the unit tests, we added assert statements to the various Get and Post methods that were called in the application so that the data used by these operations could be parsed and tested. In this way, if a bug in the application meant that improper data was being sent to these methods, the assert statement would activate and deliver a message so that the bug could be easily tracked down.

Unit tests can be configured to run automatically when the application compiles, or manually at the developer's specification. Being able to switch between these two options was useful, as disabling automatic unit tests allowed us to make changes rapidly and test the program without having its execution pause due to the assert statements, whereas it could be enabled again when we wanted to go back through the code to clean up any bugs.

Testing Review

After evaluating our UI testing results, while also observing the users interacting with our web application, we were pleased with the results. Overall, the website met the expectations our team had set forth when beginning the testing process. The main focus was to remain simple and easy-to-use, while also providing as much important information as possible for the users, which was displayed through the results of our tests conducted. We did make some minor changes based on some feedback, such as increasing the font-size and starting with a blank pre-move task reminder checklist. Originally, we had a pre-made checklist that the user could start from, but since everyone will have different tasks within their own personal checklist, it was decided that we should give the user a blank checklist to begin. This way, the user has the ability to create a complete, customizable checklist based on their own personal needs.

During our user testing process, we also received some feedback regarding the addition of some other important moving factors, such as the cost of living and nearby points of interest. We decided to add these factors, along with a couple others, based on the feedback we had received.

Project Timeline

The table below (Table 1) shows our estimated project timeline from inception to completion, complete with keystones for specific dates.

Task #	Task Name	Duration	Start Date	End Date
1.0	**Project Management**	233 days	August 24th, 2020	April 13th, 2021

1.1	Project approval	7 days	August 31st, 2020	September 7th, 2020
1.2	Create name, slogan, logo	7 days	August 31st, 2020	September 7th, 2020
1.3	Create repository for project	14 days	September 7th, 2020	September 21st, 2020
1.4	Prepare presentation	14 days	March 30th, 2021	April 13th, 2021
2.0	**Project Research**		August 31st, 2020	October 5th, 2020
2.1	Research and compare web hosting providers	7 days	August 31st, 2020	September 7th, 2020
2.2	Research and compare domain registrars	7 days	August 31st, 2020	September 7th, 2020
2.3	Research and compare project idea with similar, pre-existing services	14 days	August 31st, 2020	September 14th, 2020
3.0	**Project Design**	36 days	August 31st, 2020	November 9th, 2020
3.1	Plan web app infrastructure	14 days	August 31st, 2020	September 14th, 2020
3.2	Develop user interface prototypes	21 days	September 14th, 2020	October 5th, 2020
4.0	**Project Development**	113 days	October 12th, 2020	February 1st, 2020
4.1	Set up web hosting for project	14 days	October 12th, 2020	October 26th, 2020
4.2	Register domain and link with web server	14 days	October 12th, 2020	October 26th, 2020
4.3	Set up SSL on web server	14 days	October 12th, 2020	October 26th, 2020
4.4	Set up SQL Database for web app	14 days	October 12th, 2020	October 26th, 2020
4.5	Develop initial user interface for web app	28 days	November 2nd, 2020	November 30th, 2020

4.6	Add Authentication to web app	7 days	December 7th, 2020	December 14th, 2020
4.7	Add graphic design elements to web app	14 days	December 14th, 2020	December 28th, 2020
5.0	**Project Testing**	72 days	February 1st, 2020	April 13th, 2020
5.1	Test web app on mobile and desktop platforms	21 days	February 1st, 2020	February 22nd, 2020
5.2	Test database security	21 days	February 22nd, 2020	March 15th, 2020

Table 1: Estimated project timeline

The table below (Table 2) shows our actual project timeline from inception to completion, complete with keystones for specific dates.

Task #	Task Name	Duration	Start Date	End Date
1.0	**Project Management**	233 days	August 24th, 2020	April 13th, 2021
1.1	Project approval	7 days	August 31st, 2020	September 7th, 2020
1.2	Create name, slogan, logo	7 days	August 31st, 2020	September 7th, 2020
1.3	Create repository for project	14 days	September 7th, 2020	September 21st, 2020
1.4	Prepare presentation	14 days	March 30th, 2021	April 13th, 2021
2.0	**Project Research**		August 31st, 2020	October 5th, 2020
2.1	Research and compare web hosting providers	7 days	August 31st, 2020	September 7th, 2020
2.2	Research and compare domain registrars	7 days	August 31st, 2020	September 7th, 2020
2.3	Research and compare project	14 days	August 31st, 2020	September 14th, 2020

	idea with similar, pre-existing services			
3.0	**Project Design**	36 days	August 31st, 2020	November 9th, 2020
3.1	Plan web app infrastructure	14 days	August 31st, 2020	September 14th, 2020
3.2	Develop user interface prototypes	21 days	September 14th, 2020	October 5th, 2020
4.0	**Project Development**	113 days	October 12th, 2020	February 1st, 2020
4.1	Set up web hosting for project	14 days	October 12th, 2020	October 26th, 2020
4.2	Register domain and link with web server	14 days	October 12th, 2020	October 26th, 2020
4.3	Set up SSL on web server	14 days	October 12th, 2020	October 26th, 2020
4.4	Set up SQL Database for web app	14 days	October 12th, 2020	October 26th, 2020
4.5	Develop initial user interface for web app	28 days	November 2nd, 2020	November 30th, 2020
4.6	Add Authentication to web app	21 days	December 7th, 2020	December 28th, 2020
4.7	Add graphic design elements to web app	7 days	November 14th, 2020	November 21st, 2020
5.0	**Project Testing**	72 days	March 1st, 2021	April 5th, 2021
5.1	Implement unit testing in the codebase	14 days	March1st, 2021	March 15th, 2021
5.2	Gather data from user tests	21 days	March 1st, 2021	March 22nd, 2021
5.3	Oral Presentation	1 day	March 29th, 2021	March 29th, 2021

5.4	IT EXPO	1 day	April 13th, 2021	April 13th, 2021
5.5	Final Report and UC Library	14 days	April 12th, 2021	April 26th, 2021

Table 2: Actual project timeline

Problems Encountered and Analysis of Problem Solved

One minor problem we faced was that not all the members of the group had used ASP.NET Core's Razor Pages framework in the past, so we had to individually put in the time to learn basic knowledge on how Razor Pages work and how the code is used to create the functionality we desire. Razor Pages and MVC (Model-View-Controller) are two different frameworks which are used for the same purpose in ASP.NET. However, we have occasionally had to translate code and concepts that we found in MVC documentation into code that works with Razor Pages. Razor Pages is newer than MVC, so it has comparatively less documentation and resources available for us to refer to.

We are using Visual Studio to write our application and after creating our project we realized that the Bootstrap framework was already implemented in the application. However, we ran into an issue when we also realized that certain Bootstrap classes were not available for us to use. To solve this problem, we replaced the Bootstrap CDN link within the Layout file of our project with the most current one available.

Additionally, when using Visual Studio to create our application, we did not realize the default version of ASP.NET that the software lists was not the most current version. When implementing authentication services in the backend code, it required us to review the

compatibility of these features with the framework version. We realized that we would have to stick with using the older version 2 of ASP.Net unless we made significant changes to upgrade to a newer version, thus allowing us to use the framework's newer features. Once this was discovered, we had to determine the steps needed to update our code to this version without minimal broken features.

Hosting the website on a webserver instead of using a virtualized system proved to be quite easy to accomplish through tasks such as setting domain records, importing an SSH certificate and setting up the SQL Database. Though initial setup proved to be a straightforward process, getting the IIS service to interact with the SQL database was not an easy task. There appeared to be a permission issue that we were finally able to resolve.

Conclusion

Fall Conclusion:

Throughout the brainstorming process in the Fall, we have had many challenges to overcome. Deciding how we would differentiate our product to be unique compared to other moving websites and applications (such as Zillow or Trulia) proved to be a difficult task. We wanted to do this while having a positive impact on those moving, by providing all the necessary information to ensure a confident, successful move. These are both topics that we have spent much time on, working to gain a visualization of our thought process, while also being designing and developing an easy-to-use website. It took us several discussions, as well as many changes, to get to where we felt confident that our product had merit and would improve the overall moving experience for anyone preparing to move to a new home. After our initial discussions and months of early development we had an idea and a plan to accomplish our

goals. While we only had the initial footprint such as searching, demographic information and creating a checklist this was all just the starting for Howdy Neighbor.

Spring Conclusion:

After the Fall semester and our initial design phase was complete, we worked towards the implementation of the different APIs we needed and authentication onto the platform. The usage of the APIs we have implemented, which include Google Maps JavaScript API, Google Geocoding API, Google Places API, and Geocodio API, all allow us to search by location. However, many times we would have to use the Google Geocoding API in order to get a specific piece of the address provided by the user to get the results we wanted to display. Since we did not include any paid APIs throughout the project, if data was not available to us using free APIs, we created URLs that the user is provided to direct them to more information about specific factors. For instance, the Geocodio API provided school district information, but not as much as we wanted to provide for users. To add to the data, we were providing from the API, we built a URL that directed the user to another website that provides more information about the schools within the school district that is being displayed.

Abilities and Skills Developed Throughout Project:

As a team, we have learned a couple different technologies throughout the project. Using Azure's webapp, we were able to easily update from our GitHub repository and then run the web server to view the changes live. This has allowed us as a group to push updates to a branch then quickly view the changes in our development environment. We have begun implementing multiple APIs into our project thus far as well, such as embedded Google Maps so

a user can search and navigate within the Google Maps view. Additionally, we have decided to make use of ASP.NET Identity so that we can have account creation capabilities by working with these APIs, we have also gained more experience using JavaScript since this has been the main method for fetching data.

Another skill we have all improved on is the use of GitHub for source control. Although we all have been exposed to using GitHub in the past, using it for our project collaboration has been very helpful in familiarizing us all with it even more than we had been in the past. We have also improved on our general team-working skills. By meeting weekly and working on different parts of the project, we have all gained valuable experience working as a team to accomplish the goal of completing a project.

References

References

10 Things to Consider Before Moving to a New State. (2018, February 08). Retrieved from https://www.mrcooper.com/blog/2018/02/08/10-things-consider-moving-another-state/

Affairs, A. (2013, September 06). System usability scale (sus). Retrieved April 13, 2021, from https://www.usability.gov/how-to-and-tools/resources/templates/system-usability-scale-sus.html

United States Census Bureau. (2020, July 19). Calculating Migration Expectancy Using ACS Data. Retrieved from https://www.census.gov/topics/population/migration/guidance/calculating-migration-expectancy.html

Wood, T. (2020, November 04). 29+ Moving Industry Statistics [Updated 2020]. Retrieved from https://www.movebuddha.com/blog/moving-industry-statistics/

Zillow, I. (n.d.). Zillow API. Retrieved from https://www.zillow.com/howto/api/faq.htm

Appendix

Appendix A. Additional Information

A.1 Howdy Neighbor Poster (Figure 12)

Figure 12: Poster