Home Finder

Design Document

SDCC 23-11
Dr. Timothy Bigelow
Christian Boughton
Ella Knott
Daniel Chrisman
Lith Almadani
Michael Wieland
Team Email
https://sddec23-11.sd.ece.iastate.edu/#

Revised: 18 TUS 2023/V1

Executive Summary

Development Standards & Practices Used

IEEE 802.11 ac, g, b, a,n are required for universal wifi access to our application. Because our application will process information server side to limit user resource use, this 802.11 suite of standards will be required for access

IEEE 802.3 defines the Ethernet standard. wired or ethernet-based connectivity will be required for our application to be used by the maximum number of individuals.

IEEE 4003. This standard defines the ability for data from constellations of satellites to be used in navigation. This is crucial to our application as the global positioning system, governed under the Global Navigation Satellite System (IEEE 4003), identifies locations, aids in calculating commute times, and provides the data necessary to create the ultimate goal of this project: the weighted heat map of ideal areas to live.

Summary of Requirements

- The application will be fully accessible on web-connected mobile phones, desktops, and laptops.
- The server side will process inputted information within 5 seconds of form submission (constraint)
- Creating a heat map based on a list of data input by users like locations, frequency, and time.
- Creating a distance calculator that takes an input of locations and calculates the distance.
- The ability to create accounts and log in to existing ones. The ability to create, save, access, and delete records.
- Ability to access a web-based browser for viewing the application
- A hosting platform will be required to display our web page and allow interaction.
- The application must be visually accessible to users, including text size, font, color contrast, etc.
- The application must be responsive to ensure operability ubiquitously across browsers and platforms.

Applicable Courses from Iowa State University Curriculum

- COM S 319
- CPR E 231
- SE 329
- COM S 309

New Skills/Knowledge acquired that was not taught in courses

1. Developing front-end applications
2. Utilizing API calls for data processing
3. Creating web applications with HTML, CSS, JS

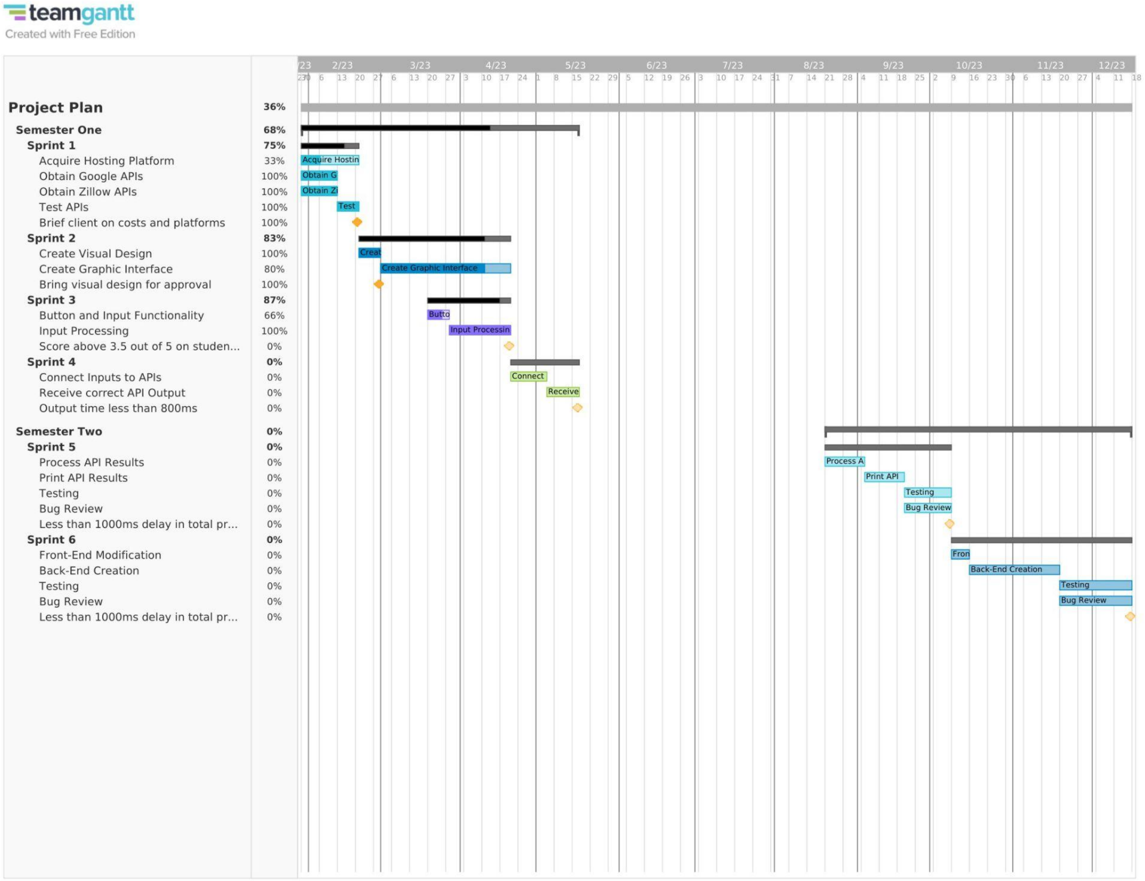
Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

Overview of sprints using a parallel model (Backlog, In Progress, Completed, Future)

Sprint 1: 01/30 - 02/19 All Base Requierments	Sprint 2: 02/19 - 04/19 Team 1 Creative	Sprint 3: 03/19 - 04/19 Team 2 Functionality	Sprint 4: 04/19 - TBD All Integration	Sprint 5: TBD - TBD All Final Front-End	Sprint 6: TBD - EOS All Final Product
Get Hosting Platform	Visual Design	Buttons	Input to APIs	Process Results	Modify Front-End
Obtain APIs	Graphic Interface	Save Input	Output from APIs	Display Results	Create Back-End
Test APIs		Delete Input		Bug Review/Testing	Create database
		Process Input			Bug Review/Testing

- Figure 3.2 Sprint Decomposition



Project Plan	**36%**
Semester One	**68%**
Sprint 1	**75%**
Acquire Hosting Platform	33%
Obtain Google APIs	100%
Obtain Zillow APIs	100%
Test APIs	100%
Brief client on costs and platforms	100%
Sprint 2	**83%**
Create Visual Design	100%
Create Graphic Interface	80%
Bring visual design for approval	100%
Sprint 3	**87%**
Button and Input Functionality	66%
Input Processing	100%
Score above 3.5 out of 5 on studen...	0%
Sprint 4	**0%**
Connect Inputs to APIs	0%
Receive correct API Output	0%
Output time less than 800ms	0%
Semester Two	**0%**
Sprint 5	**0%**
Process API Results	0%
Print API Results	0%
Testing	0%
Bug Review	0%
Less than 1000ms delay in total pr...	0%
Sprint 6	**0%**
Front-End Modification	0%
Back-End Creation	0%
Testing	0%
Bug Review	0%
Less than 1000ms delay in total pr...	0%

Figure 3.4 Gant Chart

Task	**Sprint**	**Time** (hours)
1. Acquire Hosting Platform	1	.5
2. Obtain Google APIs	1	.5
3. Obtain Zillow APIs	1	.5
4. Create Visual Design	2	5
5. Create Graphical Interface	2	10
6. Button and Input functionality	3	5
7. Input processing	3	10
8. Connect Inputs to APIs	4	5
9. Test APIs for correct output	4	3
10. Process API results	5	2
11. Print API Results	5	2
12. Testing & Bug review	5	7
13. Front End Modification	6	10
14. Backend creation	6	10
15. Backend input processing	6	10
16. Backend output generation	6	15
17. Database	6	10
18. Testing & Bug review	6	10

Figure 3.6 Person Hours

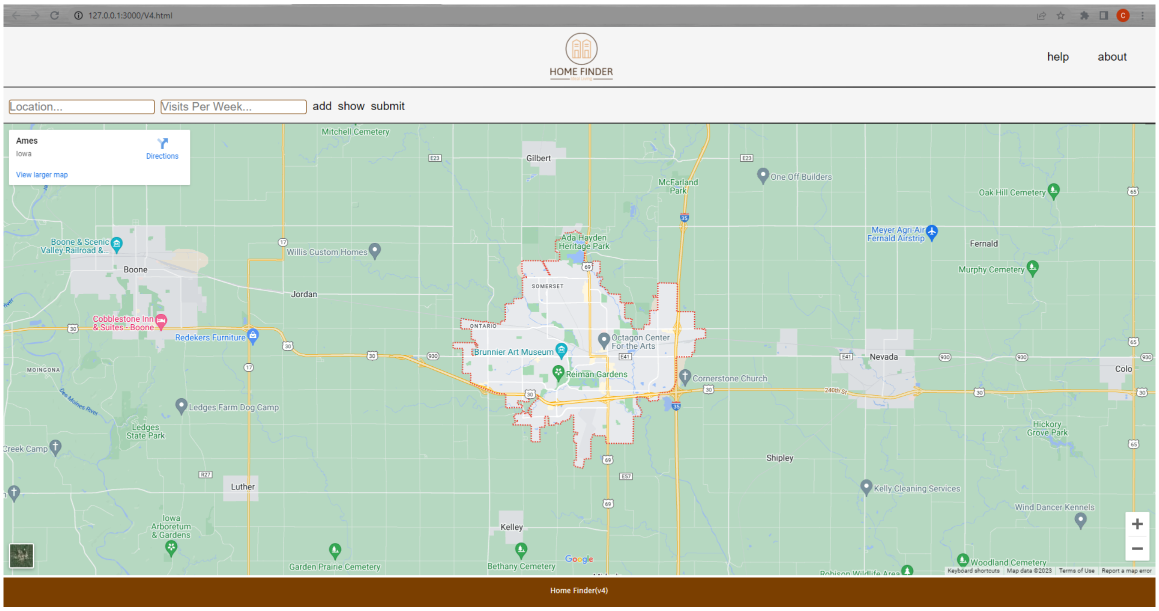
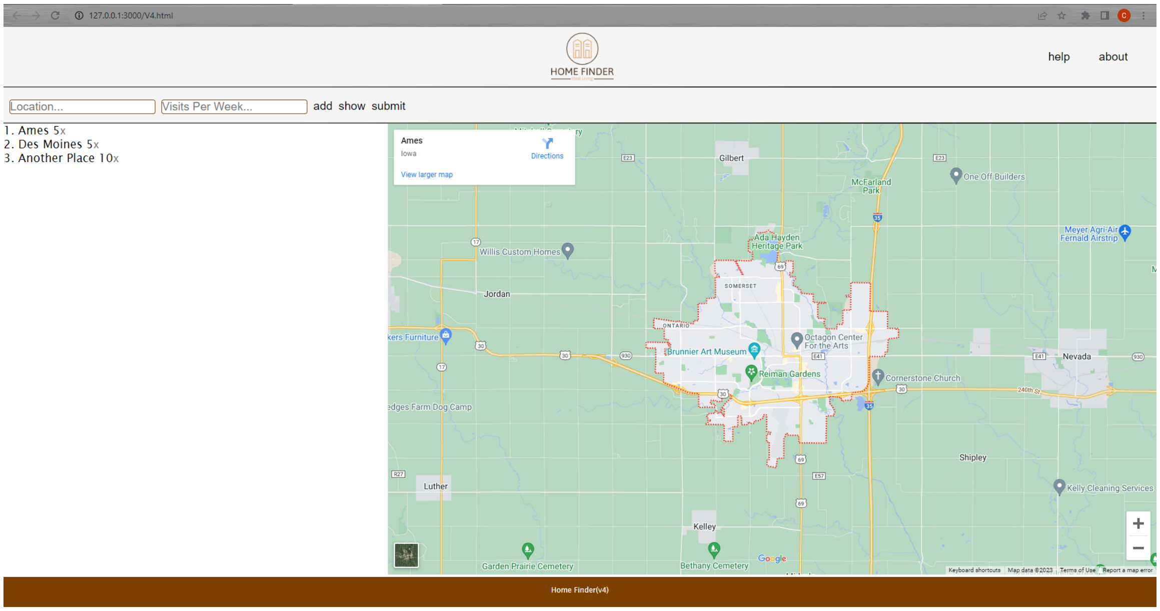


Figure 4.3.11 Current Progress (1)



1. Ames 5x
2. Des Moines 5x
3. Another Place 10x

Figure 4.3.12 Current Progress (2)

1 Team

1.1 Team Members

Christian Boughton

Michael Wieland

Daniel Chrisman

Ella Knott

Lith Almadani

1.2 Required Skill Sets for Your Project

- A working knowledge of front-end languages (i.e., JavaScript, CSS, and HTML)
- A working knowledge of backend technologies and frameworks.
- The ability to allow communication between the client side and server side
- The ability to design a user-friendly graphic interface (webpage)
- The ability to produce a comprehensive cyber security analysis
- The ability to utilize API calls

1.3 Skill Sets covered by the Team

● Visual Design:	Christian Boughton, Michael Wieland
● Back-end coding:	Lith Almadani, Daniel Chrisman
● Front-end coding:	Ella Knott, Michael Wieland
● Database Management:	Lith Almadani, Michael Wieland
● Error Handling:	Christian Boughton, Lith Almadani
● Data Integration:	Christian Boughton, Daniel Chrisman
● Functions and API calls:	Christian Boughton, Ella Knott
● Security Testing:	Ella Knott

1.4 Project Management Style Adopted by the team

We are using agile as a software project management style. We are using agile because it is more dynamic and decentralized. With our project expected to take two semesters and our advisor changing in the second semester, the agile framework will allow us to procure pieces of the project in every step early on for review. If, at any point, the client decides to add, remove, or change a part of our implementation, this can be done easier than with a waterfall style. The agile project management style is more inclusive to our highly motivated team. Because each piece is being worked on simultaneously, the team is more involved and able to make each section their own, versus a centralized waterfall design that employs a single team lead to run the entire project.

1.5 Initial Project Management Roles

- Christian Boughton - API and functionality Research
- Ella Knott - Hosting platform Research

- Daniel Chrisman - Website creation platforms and languages
- Lith Almadani - Website integration Research
- Michael Wieland - Design research and documentation

2 Introduction

2.1 Problem Statement

Searching for a new home can be exhausting and time-consuming. It requires endless hours of browsing numerous listings and websites to find the ideal area for residency. The process can be even more overwhelming if you are unfamiliar with the area. This is where our platform can make things easier. Our platform simplifies the home search process by allowing you to input your preferences, including your frequently visited places and the maximum desired travel time between them. Our algorithm generates a heat map for your desired location, making it easier for you to identify houses within the desired area. You will be able to observe your average daily commute and plan your busy schedule more effectively. Additionally, the platform can help you locate an expansion for your private business. Ideal for daycares, physicians and dentists, and recurring customer base entities. This is done by calculating the average commute time for your recurring clientele through saved addresses. An additional feature will include the business finder side of the application. This side of the application will allow businesses to find ideal locations for physical properties by analyzing traffic information, zoning laws, etc,.. Our project will prevent wasted time and undo frustration and stress by integrating many popular applications with our unique algorithm to create a visually pleasing and natively intuitive heat map.

2.2 Requirements & Constraints

Functional requirements:

- The application will be fully accessible on web-connected mobile phones, desktops, and laptops.
- The server side will process inputted information within 5 seconds of form submission (constraint)
- Creating a heat map based on a list of data input by users like locations, frequency, and time.
- Creating a distance calculator that takes an input of locations and calculates the distance.
- The ability to create accounts and log in to existing ones.
- The ability to create, save, access, and delete records.

Resource requirements:

- Ability to access a web-based browser for viewing the application.
- A hosting platform will be required to display our web page and allow interaction.

Qualitative aesthetic requirements:

- The application needs to be visually accessible to users, including text size, font, color contrast, etc.
- The overall layout of the site should also be compatible with its corresponding real-estate site to display both the heat map of applicable neighborhoods and the available housing on the market.

Economic requirements:

- All members of the team are required to have access to a web-enabled device for program sharing.

UI requirements:

- The application will be designed and tested with the user experience at the forefront of development. By focusing on what users might find unnecessary and stripping much of the visual frills associated with other housing applications, our application will be easier to use, understand, and benefit from.

2.3 Engineering Standards

IEEE 802.11 ac, g, b, a, n are required for universal wifi access to our application. Because our application will process information server-side to limit user resource use, the 802.11 suite of standards will be required for access.

IEEE 802.3 defines the Ethernet standard. wired or ethernet-based connectivity will be required for our application to be used by the maximum number of individuals.

IEEE 4003. This standard defines the ability for data from constellations of satellites to be used in navigation. This is crucial to our application as the global positioning system, governed under the Global Navigation Satellite System (IEEE 4003), identifies locations, aids in calculating commute times, and provides the data necessary to create the ultimate goal of this project: the weighted heat map of ideal areas to live.

2.4 Intended Users and Uses

Our project is intended for people in the market for new housing who may not want to disrupt their normal routine. Currently, many real estate applications just show a map of available housing and leave users to calculate their daily commutes independently. With our application, users will be provided a heat map of neighborhoods that have the least commute time based on their frequented locations and their level of importance (which users can also alter if necessary)

Some example use cases for our application include:

1. A family of three is expecting to add a new member to their family. They need to upgrade to a larger home, but they don't want to pull their child out of their current school system. With HomeFinder, they can find neighborhoods close to the school and home that matches their size needs.
2. A businessman is promoted to a position that requires frequent travel. They decided they would like to move closer to the airport to reduce their commute time to the airport. This user can input the airport as an important location, and HomeFinder will highlight more neighborhoods surrounding the airport.
3. A college student is having trouble with her current roommate situation. She decides she would like to move to a different apartment, but she doesn't want to give up the convenience of her living arrangement. HomeFinder considers that the student frequently visits the central campus, the mall, and her favorite pho restaurant and generates a heat map of neighborhoods located short distances between these locations. The student finds an apartment that cuts her weekly commute time by almost a third.

3 Project Plan

3.1 Project Management/Tracking Procedures

We use agile as a software project management style. We are using agile because it is more dynamic and decentralized. With our project expected to take two semesters and our advisor changing in the second semester, the agile framework will allow us to procure pieces of the project in every step early on for review. If, at any point, the client decides to add, remove, or change a part of our implementation, this can be done easier than with a waterfall style. The agile project management style is more inclusive to our highly motivated team. Because each piece is being worked on simultaneously, the team is more involved and able to make each section their own, versus a centralized waterfall design that employs a single team lead to run the entire project.

The goals of our project are to; produce a functioning product that fits the description of the project proposal, learn to work as an engineering team would in the field, and ensure each team member contributes to the actual engineering of the final product. The agile methodology was selected to better accomplish these goals by providing greater flexibility to our final implementation, providing a more dynamic role for each team member, and splitting the engineering aspects of the project so that each member would practice their trade.

Our team uses a combination of GitHub and Discord to track our progress in this project. GitHub allows us to create a repository of research and code we have been working on. Communication, announcements, and planning take place on Discord as our team has a few available open spots in our schedule to meet. As the semester progresses, if our current tracking method proves failable, we may move to ProofHub as it combines the features of many other tracking software such as time management, calendars, repositories, and communication

3.2 Task Decomposition

Refer to image 3.2 for sprint decomposition

Linear Breakdown of all tasks

- ObtainHosting platform
- Obtain API access
    - Obtain Google Maps API access
        - Create management account and project to receive keys
        - Learn how to use API through documentation
        - Test functionality of APIs
    - Obtain Zillow API access
        - Email R&D department for access to their internal data APIs
        - Learn how to use API though provided documentation
        - Test functionality of API
- Develope a Front-End-Only product for the first iteration
    - Design the visual appearance of the website
        - Begin with pencil and paper
        - Get feedback from our client
        - Alter design if necessary and resubmit for feedback
    - Create graphic interface using HTML, and CSS
    - Add functionality with JS
        - Ensure button clicks correspond with desired actions
        - Take input from the text box and save internally
        - Delete imputed locations from internal storage
    - Process inputs and utilize APIs
        - JS functions retrieve user input from internal storage
        - API converts the locations into coordinate points
        - JS function calculates the max/min latitude and longitude to make a variable size grid encompassing all points
        - API calculates drive time from each entered location to each point on the grid
        - JS function averages the drive times taking user entered weights into account (DT = DT *(1 - (weight * .10))
        - JS separates coordinates by the percent max drive time and uses predetermined ranges to assign heatmap color codes
        - API generates heatmap from coordinate points and color codes
    - Display heatmap to screen
    - Use the same drive time grid to pull listings from zillow within a variable radius of the lowest percentile coordinates
    - API drops pins on the locations where the listings are overlaid on the heat map
    - Pins link to the Zillow website listing when clicked
- Develope a Front-End Back-End version
    - Modify graphic interface to modularize the application
    - Modify functionality to support forums
    - Add account creation to store and recall previous uses of this application
- Create database

- Identify requirements
    - Understand the requirements of the system or application that the database will support
    - Identify data that needs to be stored
    - create organization of stored data
- Design the schema
    - Define tables, columns, and relationships between the tables in the database
- Choose database management system

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Sprint 1; The use of the hosting platform and APIs will be at no cost to the development team. The hosting platform will be able to support front-end only, and UI/UX web pages

Sprint 2; The visual design will be pleasing to PC and Mobile users The visual design will score 3.5 or higher out of 5 in a random campus review

Sprint 3; Input processing time will be negligible to the user A maximum of 2 bugs in the styling implementation for this first iteration. (i.e., button does not highlight when hovered over)

Sprint 4; Total input processing time, including API usage, should be no greater than 800ms

Sprint 5; No more than 1000ms may elapse from the start of processing information to the final heatmap generated.

Sprint 6; No bugs present in the program No more than 1000ms may elapse from the start of processing information to the final heatmap being generated.

Performance: Our backend should be able to handle high traffic volumes and respond quickly, with low latency and high throughput. Scalability: Our backend should be able to scale up or down based on demand.

Security: Our backend and database should be secure and protected against common threats like SQL injection.

Flexibility: our backend has to be adaptable and handle a variety of frameworks, languages, and platforms.

Maintainability: Our backend should have clear and well-organized code and documentation.

3.4 Project Timeline/Schedule

Refer to figure 3.4 for Gant chart

3.5 Risks And Risk Management/Mitigation

Sprint 1:

- Hosting platforms typically employ a maximum request limit
    - Obtain a platform that will not constrict the web application

Sprint 2:

- Web applications that require too many inputs may not scale to small platforms
    - Use a minimalistic approach to designing the web application and a grid layout with viewport conditions to better adapt to the constraints of the user

Sprint 3:

- Overcomplicating functionality may lead to broken links and inoperability
    - Small modular programs will be utilized with internal data storage in mutable arrays to provide simplicity and error checking

Sprint 4;

- API use may be restricted or denied
    - Backup APIs that provide the same functionality are readily available
- API utilization may be monetarily costly
    - During testing the application will be limited to manually imputed information instead of creating an API call when unnecessary to limit costs
    - Alternative APIs are available with differing price schemas

Sprint 5;

- System testing may accrue large monetary costs from large API request generations
    - We are trying to contact the API host to see if we can run tests without cost

Sprint 6;

- Security risks including data breaches, unauthorized access, and hacking
    - Complete cybersecurity review using Kali linux's suite of penetration testing tools
    - Use a reliable hosting platform that has cybersecurity policies in place
- Scalability risks for variable client traffic
    - Use a reliable hosting platform that can handle variable traffic
- Integration risks from compatibility issues, data loss, and other bugs
- Performance risks from requiring HTTPS requests through forms to connect to the backend
    - APIs have been chosen due to their low turnaround time
- Database integrity
    - testing for accurate and reliable consistencies in our stored data

3.6 Personnel Effort Requirements

Refer to figure 3.6 for Person-Hours table

- Obtain Host Platform & APIs (Tasks: 1-3)

    - Basic acquisition such as obtaining API keys
- Develop the front-end-only product for the first iteration (Tasks: 4-8, 10-11)
    - These are the first steps in the design process
        - Visual appearance - 5 hours
        - Graphic interface with functionality - 10 - 15 hours
            - Create Graphical interface
            - Button & Input functionality/processing
        - Add API interfacing - 3 hours
        - Modify outputs for results - 2 hours
    - Overall these tasks will initially require a high input of hours, but as features are designed and implemented other micro-tasks of this type will be easier and quicker to complete.
- Develop a traditional front-end UI/UX version (Tasks: 13-14)
    - Modify the graphic interface - 10 hours
    - Create Backend - 10 hours
        - Process inputs from forums with PHP to be used by APIs - 10 - 15 hours
        - Modify outputs and send them back to the user - 15 - 20 hours
    - After testing initial front end work, a good amount of time will be spent fixing and improving features and the overall design. Along with this work, we will start the creation of the backend. Backend work will continue after the improvements are implemented.
- Testing - 20 hours (Tasks: 9 & 12 & 15)
    - Tasks that require a lot of time invested in upscaling the quality and function of the site.

3.7 Other Resource Requirements

- Monetary Considerations; Our project is all web-based, so we will not need material except purchasing of domains or request limit extensions for APIs if needed.

- Programming Languages; Our group is using HTML, CSS, PHP, JS, and an undetermined back-end object-oriented programming language..

- Databases: Our group will need to store data for our web application in databases like MySQL, MongoDB, or SQLite.

- Cloud services: The application will use cloud services for hosting, storage, and other infrastructure for online applications, provided by cloud services like Amazon Web Services (AWS).

- APIs: The application includes functions using APIs offered by third-party services like auto-complete addressing, distance matrices, maps, and others.

4 Design

4.1 Design Context

4.1.1 Broader Context

Our design project is intended to be used by any member of the population that is searching for an ideal area to live in. The application can subsequently find ideal areas for businesses with recurring client models. The community impact of this application offers many benefits to all users: however, it affects lower-income and disabled users the most. The tools provided allow users with limited transportation or need to be within a certain distance from a care facility to better find housing that matches their needs. Our goal for this project is to solve the societal issue of locating housing based on the user's needs.

List relevant considerations related to your project in each of the following areas:

Area	**Description**	**Examples**
Public health, safety, and welfare	Low-income and disabled users will be empowered to locate housing that matches their needs.	Low-income users can find housing near public transportation or their worksite, helping with transportation costs. Disabled users can find housing near care centers or resources in case of emergencies. The direct users of our application are likely to benefit from the convenience and time-saving aspect of the tool. Using our application, users can find a living location that is most convenient for them in terms of the time spent on daily trips, which can lead to less stress, more free time, and a better quality of life. Communities, where our application is used, may also indirectly benefit from reduced traffic congestion and associated

		pollution levels, which can lead to improved public health and safety. Our project could positively affect public health and safety by reducing commuting time. By reducing the time spent on the road, there is a potential reduction in the likelihood of accidents, which can lead to fewer injuries and fatalities.
Global, cultural, and social	This application is not geared toward any social community. It is designed to be usable for any member of the population.	No code of ethics will be violated with this application. Specific communities and ethnic cultures may have different values, preferences, and lifestyle choices that influence their perception of an optimal living location. Our project may need to consider these differences and provide customized features that reflect the values, practices, and aims of the cultural groups it affects. For example, a community that values walking and biking may prefer a more walkable or bike-friendly location.
Environmental	This is a software-only project that produces no environmental impact	Because this project is meant to replace current applications, no extra energy will be spent. Our project could contribute to reducing the overall carbon footprint by encouraging users to find living locations closer to their

		workplaces or daily destinations, therefore reducing the need for long commutes and associated carbon emissions. By reducing the time spent on daily trips, our project could also reduce traffic congestion, which can lead to improved air quality and reduced noise pollution. Our project may rely on data centers and servers, which can have high energy consumption and a carbon footprint.
Economic	Private: Users will be empowered to find ideal living locations that will ease their day-to-day spending. Commercial: Users will be able to expand or relocate to areas that are more convenient for their clients	Private: Employees can live closer to their employer, saving on transportation costs. Commercial: Small businesses with a dedicated clientele can move locations and maintain the same average commute for customers (i.e., dentists, day-cares, and family docs. Our project could contribute to reducing transportation costs for individuals and families by helping them find living locations that are closer to their workplaces or daily destinations. This could result in substantial savings in fuel costs, vehicle maintenance, and public transportation fees. Our project could also lead to increased economic activity in

		certain areas by encouraging people to move to certain neighborhoods or cities. However, it also may have a negative economic impact on certain neighborhoods or cities if it encourages people to move away from those areas

4.1.2 User Needs

Home buyers need a way to search for ideal areas to live and see the properties available in those areas because current applications fail to provide adequate search customization.

Businesses need a way to ensure they are relocating or expanding to an area their clients will be satisfied with because long commutes or inopportune areas may cause undue hardships for their clientele or revenue loss.

Commuters need a way to find living locations closer to their workplaces or daily destinations because they want to reduce the amount of time and money spent on daily transportation and improve their quality of life.

Families with children need a way to find living locations in safe and convenient neighborhoods with good schools and parks because they want to provide their children with good education and a safe environment.

University students need a way to find living locations close to their campus and other facilities because they want to reduce the time and money spent on transportation and improve their academic performance and social life.

Retirees need a way to find living locations in a calm, pleasant neighborhood with access to healthcare services and social activities because they want to enjoy their retirement and maintain their physical and mental well-being.

Real estate developers need a way to identify areas with high demand for housing and potential for profitable investments because they want to maximize their profits.out

4.1.3 Prior Work/Solutions

Finding the best places to live based on daily travel times has been the subject of numerous research projects and web applications.

1. Walk Score: Walk Score is a web application that calculates the walkability of a given location by analyzing its closeness to nearby amenities such as grocery stores, restaurants, schools, and parks. One of the advantages of Walk Score is that it considers a wide range of factors that can affect the livability of a location.
   https://www.walkscore.com/
2. Trulia: Trulia is a real estate website that provides a heat map of neighborhoods based on factors such as crime rates, schools, and commute times. One of the advantages of Trulia is that it provides a broad view of different neighborhoods and their livability factors.
   https://www.trulia.com/

Our web application has several advantages compared to these existing products and research studies:

- First, it uses a data-driven approach to generate personalized heat maps that reflect individual users' commuting patterns and preferences.

- Second, it provides a user-friendly interface that allows users to easily explore different living locations and compare them based on commute time.

- Third, it can be integrated with other real estate websites and services to help users get a better experience.

One potential disadvantage of our web application is that it may require a large amount of data and computational resources to generate accurate heat maps for a wide range of users and locations.

Applications such as Zillow provide some customization for calculating commute times; however, they lack the ability to generate ideal areas to live in, then connect the user to listings in those areas.

Comparative Pros	Comparative Cons
Visually identify ideal home areas	Larger datasets leading to increased costs
Works with existing applications instead of competing with them. Listings are generated through APIs that link back to the existing site. Our goal is to provide customization. This allows the best [parts of other sites to be used while ensuring competitors do not box us out.	
Provides area searches and listings	

4.1.4 Technical Complexity

1. User Interface; Our application requires a user-friendly interface that allows for account creation, input handling, and map generation with modifications. The design of such an interface requires knowledge of user experience design principles and front-end web development skills such as HTML, CSS, and JavaScript.
2. Data Storage; Our application will need to store user data securely and efficiently. This requires knowledge of database management principles, such as SQL and NoSQL databases.

3. Data Processing; Our application requires a custom algorithm to overlay a variable-sized grid and perform complex calculations to create the data necessary for the generation of the heatmaps. This requires an understanding of data structures, geometry, and general mathematics.

4. Heatmap Generation: Our application will generate heatmaps based on user inputs. This requires knowledge of data analysis and visualization, as well as back-end development skills such as Python and Google Maps APIs.

5. Optimization Algorithms: Our application will also need to implement optimization algorithms to find the optimal living location for each user based on their daily trips. This requires knowledge of mathematical optimization principles.

   In general, the project's technical complexity is sufficiently challenging, requiring a broad range of technical skills and expertise, including UX design, front-end, and back-end development, database management, data analysis and visualization, and mathematical optimization. Also, the project's requirements match current solutions and industry standards, such as location-based applications like Google Maps or Waze.

4.2 Design Exploration

4.2.1 Design Decisions

1. Pleasing visual web design that is minimal to ensure understanding and ease of use.

2. Functional front end for processing user-inputted information coded in HTML, CSS, and JS

3. Functional back end for storing data, processing data, and connecting to APIs

4.2.2 Ideation

Our team used the brainstorming method with a storyboard technique to ideate a solution for structuring our web application.The storyboard gave us a visual representation of our ideas that we could build on and reference.

1. Front end only design

2. Front end with UI/UX design

3. Design built on public applications such as wix.com

4. Design hosted on a private server

5. Design hosted on a cloud infrastructure

4.2.3 Decision-Making and Trade-Off

We chose to identify the pros and cons of each idea through lists in our storyboard. We chose to first start with a front-end-only design to ensure functionality before moving to a cloud-hosted front-end UI/UX design for the final implementation. Our final implementation will allow greater user access with less overhead for our client.

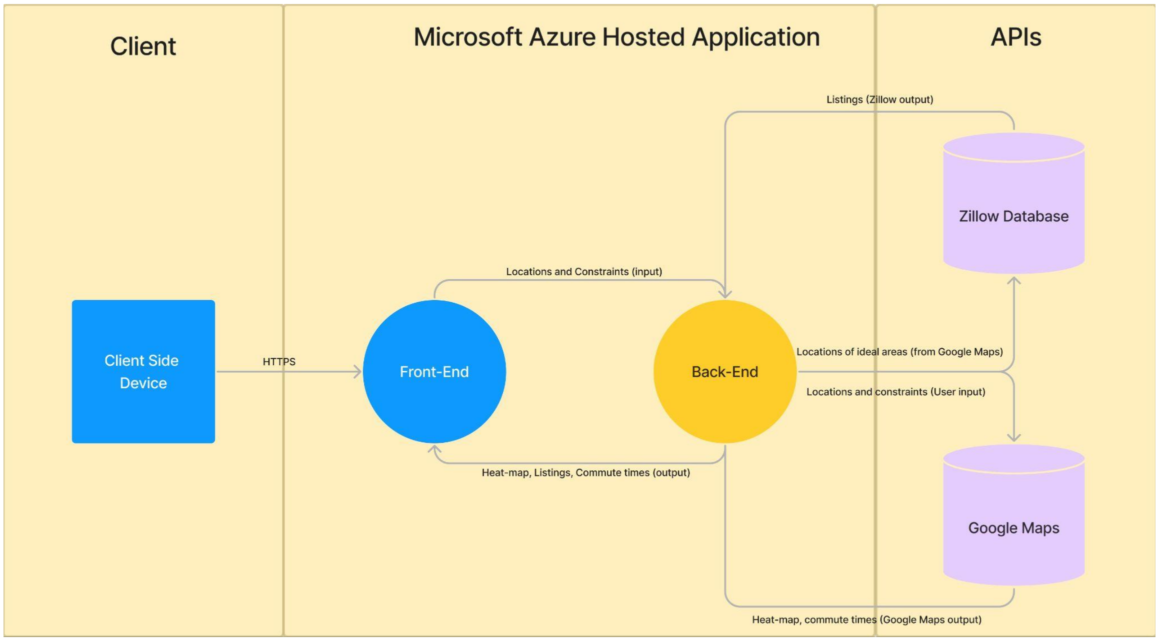
4.3 Proposed Design

Our group has created the front-end structure and functionality, barring connection to the APIs.

The web application can scale to the viewport's size and run in common browsers. The functionality of the web application has been completed; the application can take user inputs, store them in an internal array, modify the array, and process the data into our final data set used by the end API to generate a heatmap. The APIs have not been connected to the web application at this point, as we are testing the return types of different APIs to ensure data integrity.

4.3.1 Design Visual and Description

Refer to figures 4.3.11 and 4.3.12 for examples of our web application progress


Client
Microsoft Azure Hosted Application
APIs
Listings (Zillow output)
Zillow Database
Locations and Constraints (input)
Client Side Device
HTTPS
Front-End
Back-End
Locations of ideal areas (from Google Maps)
Locations and constraints (User input)
Heat-map, Listings, Commute times (output)
Google Maps
Heat-map, commute times (Google Maps output)

Our design is a front-end UI/UX website accessed over HTTPS, The user side, box 1, is responsible for generating requests by entering in location and constraint information. This information is formatted by the front end and sent to the back end through PHP (box2). The backend then processes the information and uses the Google Maps API to generate ideal locations. The ideal locations are broken into subdivisions on the back end and sent to the Zillow API to retrieve listings in those areas (box3). The heat map and listings, and commute times are then sent back to the front end for user viewing.

4.3.2 Functionality

Our design is intended to be used by home seekers as a first or last resort to aid in the home-buying process. The normal use of our application is a user connects to the web-based application and enters the locations most commonly visited and the time constraints for traveling to those locations. Our application will then process the imputed information and respond with a heatmap of the ideal living areas that fit the constraints for the entered locations. After generating the heatmap, users can view the ideal areas and select from listings within those areas. Selecting a listing links the user to the listing's primary hosted site, where more information can be found.

Our current design takes user input and processes it into the data required for the API functions. We currently do not have the APIs integrated with the application to save on costs so we have substituted a Haversine formula for calculating distances across the Earth using coordinates instead of drive times.

4.3.3 Areas of Concern and Development

Our primary concern is the cost associated with large user requests. At a predefined number of user requests to the APIs, fees will start to accrue. Our team could not design the application without the use of the APIs because satellite and GPS information is needed to calculate drive times with traffic considerations.

To address this issue, our team will integrate the API calls last and present the functionality to our clients, allowing them to make a final decision based on their needs. continued use of this application will require an access key under our client or another account. The application will be designed to easily input a new access key for simplicity for the client.

4.4 Technology Considerations

We are using JavaScript for the frontend and Google APIs for map generation and distance calculations.

Strengths:

1- JavaScript is widely used for web development, which makes it easy to find resources and support.

2- Google Maps API is a powerful tool for generating maps and manipulating location-based data, making it way easier to implement our main functionalities and add more features.

Weaknesses:

1- JavaScript is a client-side language, which means that it can only execute code on the client's browser. This can lead to performance issues if the application is handling a lot of data, as the browser may need help to keep up with the calculations required for generating heat maps.

2-The use of Google Maps API comes with a cost, as the API requires charges based on usage.

Trade-offs:

1- The use of JavaScript and Google Maps API will allow us to have a high degree of customization and flexibility in the design, but it will also require us the more technical expertise to implement and maintain.

2- Using a third-party API like Google Maps means that the application relies on that provider for its core functionality. This can create dependencies and potential issues if the provider makes changes.

Possible solutions and design alternatives:

1- To improve performance, our application could utilize server-side processing to handle the generation of heatmaps. This would allow for more efficient handling of large datasets and reduce the strain on the client's browser.

2- To reduce costs, our application could explore alternative map providers that offer similar functionality at a lower cost or for free. Examples include OpenStreetMap or Mapbox.

3- To reduce dependencies on third-party providers, our application could explore the use of open-source libraries for map generation and location-based services. Examples include Leaflet or D3.js.

4.5 Design Analysis

The proposed design from 3.3 worked because the interface is the exact same as previously designed. However, prior to figuring out the final design, we drew out multiple different ways to assemble the website until we found one that made sense as a developer and a user. Furthering into the future, we plan to have a beta version of the website for real estate agents to test out and send feedback about all the different aspects of the website.

Regarding modifying or iterating over the design, the website needs to become more user-friendly. Our "rough draft" is a little finicky. A more user-friendly site ultimately means a more professional site. This can be achieved by improving the flow, navigation, and accessibility of the website. It is important to keep in mind the users' perspectives and preferences when designing the website.

4.6 Design Plan

Requirements: The website should have a user-friendly interface allowing users to input their preferences, such as location, work location, and any other frequently visited place. The website should also provide relevant information, such as current houses in the market and not ones that have already been sold.

Modules: The design plan should include separate frontend and backend modules. The frontend module would be responsible for presenting the user interface and collecting user input, while the backend module would process the input and provide the relevant information to the user.

Module diagram: The frontend module would depend on the backend module, as it needs to receive information from the backend to present it to the user. The backend module would also be concurrent, as it needs to process multiple requests from different users at the same time.

Interfaces: The frontend module would communicate with the backend module through an API, defining the input and output parameters and their formats.

Module constraints: The website should be able to handle a large number of users and provide accurate and up-to-date information. the backend module will be designed to handle different types of data and formats and use appropriate algorithms and databases to process the user input and provide relevant information.

5 Testing

Testing is an essential part of any software development project. It is particularly crucial for a web application like ours that aims to help users find the optimal living location by creating heatmaps based on their daily trips. And to ensure that our application functions correctly and meets user requirements, we need a comprehensive testing strategy that covers all aspects of our system's design. One of the unique challenges to testing our system is the reliance on third-party APIs, particularly Google APIs, to generate the heatmaps. Since these APIs are external to our system, they can be subject to unexpected changes, impacting our application's functionality. So, we need to ensure that our testing strategy includes detailed testing of the integration with these APIs. Another challenge in testing our system is the diverse range of user inputs our application needs to handle, including daily trip frequency and timing. This makes it essential to perform extensive testing of the input validation and processing functions to ensure that the application can handle various inputs and produce accurate heatmaps. We will address these challenges and ensure the overall quality of our application by implementing a testing strategy that includes a range of testing techniques, including unit testing, integration testing, interface testing, and user acceptance testing.

5.1 Unit Testing

1. User Account Creation and Management: We will create test cases for user account creation, login, logout, and password reset functionalities. Which can be done using tools like JUnit.

2. Input Validation Functions: We will test the functions responsible for validating user inputs, including input data types, format, and range. Which can be done using tools like JUnit or NUnit.

3. Heatmap Generation: We will test it by creating test cases for generating heatmaps based on user inputs and verifying that the heatmaps are generated correctly. Which can be done using testing frameworks like JUnit, NUnit, or TestNG. Output heat maps can also be verified by comparing the weights to hand-generated versions. Hand-generated versions will be calculated using google maps to find the commute time from the start to each location and take the average. This should match the heat map weight.

4. API Integration: we will test the functions responsible for integrating with Google APIs to generate heatmaps based on user inputs. Which can be done using testing frameworks that support API testing, like Postman.

5. User Authentication and Authorization: we will test the functions responsible for user authentication and authorization. Which can be done using frameworks like Spring Security.

When testing each unit, we will consider testing both the expected behavior and the boundary conditions, which will include testing both valid and invalid inputs and edge cases that test the limits of our application. And to perform these tests, we will use various testing frameworks and tools, such as JUnit, NUnit, or TestNG. These frameworks can be integrated with our build and deployment process to automate testing.

5.2 Interface Testing

1. User Interface: This interface allows users to create accounts, provide input for daily trip frequency and timing, view the generated heatmaps, and manage their accounts. We need to test the user interface of our web application to ensure that it works as expected. We will use tools like Selenium, Cypress, or TestCafe. We will also deploy a beta version of this application for review by realtors. Our client has approved the review process.

2. Google API Interface: This interface enables our web application to integrate with Google APIs to generate heatmaps based on user inputs. We need to test the integration between our web application and Google APIs. We will use tools like Postman or SoapUI to create API tests and ensure that the API calls return the expected results.

3. Finally, we must test the interaction between the user interface and Google API interfaces to ensure that our web application works as expected. We can do this by monitoring data results from the integrated API and comparing them to the expected results.

5.3 Integration Testing

The first critical path in our design is the input and output streams to and from the used APIs. This is because we do not have full control over how the data will be manipulated and what the results will be. We will test this by first using a bare-bones framework to run several API requests and monitor output. Requests will be generated based on normal and edge case inputs (i.e., a single location or a very large amount of locations spread over a great distance). We will use tools such as JUnit and NUnit to test for exceptions by comparing the output structure to the expected structure. After ensuring the edge case and normal tests match our expected structure for data returns, we will then generate heat maps from these and compare them to the hand-created ones described above.

The second critical path is the user interface and input stream on the client end.

5.4 System Testing

will be done using the BrowserStack suite of tools. These tools will allow us to view the website across 300 platforms, including mobile-based ones. BrowserStack will also allow us to test the website's responsiveness to ensure we maintain usability across screen and device types. Overall functionality will be tested by generating heatmaps on each of the 300 browsers and comparing them manually or through image recognition to determine if any meaningful differences arise. The BrowserStack application will allow us to test our design on each platform concurrently by simulating each request from a different browser type.

A beta version of this application will be reviewed by currently practicing realtors to ensure functionality meets the demands of our users.

5.5 Regression Testing

We have created our site as modularly as possible and used general CSS styles for buttons to ensure future additions will adhere to our current design. The critical features implemented in this program are the integration of the APIs. The modularity allows us to make modifications to the functions without disrupting the current progress. Front-end visual or structural changes will need to be tested as stated above on 4.4 for browser support and responsiveness. We will reuse the BrowserStack suite of tools to ensure any additions to the visual or structural design in the front end do not hamper the end result.

Tests created using the BrowserStack testing tools will be automated and used to determine whether new functionality breaks old functionality.

5.6 Acceptance Testing

Our client has been involved in our acceptance testing through periodic meetings. We have met once a week to update progress made. While we display these updates, we are receiving feedback on design choices. This will help make sure that non-functional requirements are met. As development progresses, these meetings will continue. As functional features are added, we will test them and then receive feedback from the client.

Additionally, we have received information that there is interest in our product from members of the housing and real estate industry. We plan to hold a showcase once the product is further developed to receive feedback.

5.7 Security Testing

Our project will be dealing with user information, including their frequented locations as well as listed housing information; because of this, we need to ensure that our user’s sessions and data are properly encrypted and that our site is safe from injections and other common vulnerabilities. We will test for various potential web-based security issues using various penetration testing tools, including Burp Suite, SQLMap, Nikto, and more. We plan to use the OWASP Top Ten list of web-application vulnerabilities for a baseline of vulnerabilities to test for using these tools. This list highlights broken access control, cryptographic failures, site injections, and request forgeries as some of the most common vulnerabilities on the web.

5.8 Results

N/A. We have not reached the point of testing in our project yet. The most testing was to make sure the front end allowed the user to type words.

6 Implementation

Next semester, our team will separate the application into a front-end back-end system that stores previously created heatmap data under user profiles to limit the number of API calls made and allow users to recall data.

7 Professionalism

This discussion is with respect to the paper titled "Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Responsibility

Public Health and Safety: The IEEE Code of Ethics addresses the responsibility by stating that engineers shall "hold paramount the safety, health, and welfare of the public and the protection of the environment." This statement emphasizes the importance of ensuring public safety and environmental protection in engineering projects.

Global and Social Impact: The IEEE Code of Ethics addresses this responsibility by stating that engineers shall "seek, accept. and offer honest criticism of technical work, to acknowledge and correct errors, and to properly credit the contributions of others." This statement emphasizes the importance of being honest and transparent in technical work.

Communication: The IEEE Code of Ethics states engineers shall "be truthful and realistic in stating claims or estimates based on available data." This shows the importance of being honest and accurate in communicating technical information.

Professional Developments: The IEEE Code of Ethics states "advance the integrity and reputation of the profession consistent with the public interest." This emphasizes the importance of maintaining a high level of professionalism and ethical behavior in the engineering profession.

Ethical Leadership: The IEEE code of ethics addresses this responsibility by stating that Engineers shall "Avoid conflicts of interest and shall not exploit or misrepresent the work of others for personal or professional gain." This statement emphasizes the importance of avoiding conflicts of interest and maintaining ethical behavior in all professional interactions.

Sustainability: The IEEE code of ethics addresses this responsibility by stating that engineers shall "Consider this social and environmental impact of their professional activities." This statement emphasizes the importance of considering the long-term social environmental impact of engineering projects.

Professional Responsibility: The IEEE code of ethics addresses this responsibility by stating that engineers shall "uphold and enhance the honor, integrity and dignity of the engineering profession." This statement emphasizes the importance of maintaining the highest standards of professionalism and ethical behavior in the engineering profession.

Compared to the NSPE code of ethics, the IEEE code of ethics is more comprehensive in addressing each of the seven professional responsibilities. While the NSPE code of ethics also covers similar areas, the IEEE code of ethics provides more specific guidance on how engineers should behave in different professional situations additionally, the IEEE code of ethics places a greater emphasis on the social environmental impact of engineering projects.

7.2 Project Specific Professional Responsibility Areas

Public health and safety: Yes, this applies to our project because the safety of the users' information must be kept in an area where it cannot be exploited by any hackers or easily accessible to the public. We are performing this at a high caliber by using reputable sites to host our site to ensure peak security.

Global and social impact: Yes, this is applicable to our project as our website is directed towards pretty much anybody so we want honest criticism in order to produce the greatest result.

Communication: Communication is very important to our project as it is what is going to create the greatest result. Without ideal communication, it will be almost impossible to complete our project.

Professional developments: This is also applicable to our website as professional development is something that we want to strive for. Professional development means user-friendly as well as an almost perfect website.

Ethical leadership: This is applicable because everybody must be held responsible for their portion inputted into a project. Leadership is something that can be very beneficial because it keeps everybody on the same task, working together towards the same goal which is very important in a project that spans two semesters.

Sustainability: Sustainability does not pertain to our project as we are only a website so there's not much sustainability to go into it.

Professional responsibility: Going hand in hand with professional development this is also applicable to our website. In order to efficiently complete this task we must all be responsible and complete our individual tasks.

7.3 Most Applicable Professional Responsibility Area

The most applicable professional responsibility area within our project is the amount of data from users that we take in. It's going to take a lot of responsibility to make sure this data is secure and not messed around with by anyone within our group or anybody outside of our group.

8 Closing Material

8.1 Discussion

Our current progress for this project is a partially implemented front-end-only version of our website. For our current iteration, users can input the locations they frequently visit and the number of times they visit a location per week. Users can also view an iframe map and scroll around the map; however, this is merely a placeholder for the generated map. In the Javascript portion of the front-end code, we have included our algorithm to compute coordinate points and weights for input into the heat map generation API. We currently like the layout of this website design and are going to continue to use it and add to it in the future. Our project has met the requirements for a web-based application that is usable across the most common device types and browsers. Additional functionality to meet the required goals of this project are planned to be implemented next semester. These goals are to generate a heatmap and recreate the application as a full-stack development. To do this, we only need to connect the API functionality for location autofill, readable address-to-coordinates, a distance/drivetime matrix, and a heatmap generation API.

8.2 Conclusion

At this point in the project we have a semi-functional front-end version of the website that meets the responsiveness goal of ensuring as many users are able to utilize the product as possible. Beginning next semester we will add API functionality to complete the front end and distribute a beta version to a group of realtors for testing. After receiving feedback we will make the necessary changes, and implement this project as a full-stack development. Our goal is for our users to be able to input any location they frequent and ease the process of home searching for them by generating a heatmap of the most ideal areas to live in, without the stress that accompanies today's version of home buying. For this semester, our goal was to understand how this project could come to life and to see whether or not it is possible to create such an app. All areas of our research proved the application was possible for us to build and test before the end of our senior year. The APIs, platforms, and functionality worked perfectly and showed us that this project could be done. If we were to go back and change one thing, we would begin our web design process using a grid layout in CSS. This layout has proved to be the most responsive and allows the greatest manipulation of our webpage. In all, our project has progressed faster than we anticipated and should have a completed product that meets all requirements listed by the client before the end of the fall 2023 semester.

8.3 References

8.4 Appendices

8.4.1 Team Contract

Team Members:

1) Christian Boughton 2) Michael Wieland

3) Daniel Chrisman 4) Lith Almadani

5) Ella Knott 6) ______________________

Team Procedures

Day, time, and location (face-to-face or virtual) for regular team meetings:

Group Meetings: Thursdays, 1:30 pm. ECPRE Lobby

Client Meetings: Thursdays, 2:00 pm. Advisors Office

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):

Discord - quick communication between team members, reminders, scheduling, etc
E-mail - communication with Advisor, more formal announcements

3. Decision-making policy (e.g., consensus, majority vote):

Majority vote with considerations for experience levels in relevant areas

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):

Minutes shared and archived through discord notes taken in rotations

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

Members are expected to attend all meetings, if a team member is not able to meet or plans to be late to the meeting, they should inform the rest of the team via Discord.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

Everybody is expected to participate in team assignments and to respect and follow timelines, and deadlines. However, it is understandable that our schedules are different and our participation rate will be different from one assignment to another.

3. Expected level of communication with other team members:

On a weekly basis and everybody should communicate their absence on any team meeting or assignment

4. Expected level of commitment to team decisions and tasks:

Team members are expected to respect and commit to team decisions that were made through majority voting.

Team members are welcome to share their opinion concern and debate as long they commit to team decisions

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):

Client Interaction: Ella Knott

Testing: Lith Almadani

Team Organization: Christian Boughton

Documentation Organization: Michael Wieland

Development lead: Daniel Chrisman

2. Strategies for supporting and guiding the work of all team members:

Servant-leader roles are played by each individual in their field of expertise. They will use the abilities they have to focus on the growth of the group and well-being of the members. SMEs in their respective fields will lend their expertise to struggling group members in a respectable fashion for the betterment of each member.

3. Strategies for recognizing the contributions of all team members:

Attentiveness, timeliness and genuineness. Peer-to-peer recognition for work completed. Individual accomplishments in this project are posted on the group discord page for all members to see. Team accomplishments are included in lab reports and sent to the client/advisor. During our weekly meetings team members are encouraged to share their accomplishments in person.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the

team.

Christian Boughton -10 years of experience in the private field as a manager/supervisor allows me to see the big picture and day-to-day operations and how they intertwine to achieve an end goal. Previous experience in data integration and large data analysis.

Michael Wieland - Front-end design and testing, project management, and documentation experience from coursework. Currently in an internship/co-op working on front-end development that uses a geo-mapping service (MapBox). Light experience with database management.

Ella Knott - 3 years experience interning in cybersecurity. Primary focus on Vulnerability Management, light experience in front-end design, and API integration. I typically fill in the role of mediator and communicator in previous projects.

Lith Almadani- Developing, testing, and maintaining firmware for crop drying and monitoring systems. Developed Python testing scripts that use the CAN library to communicate with different modules.Experience using PSoC creator to design firmware for arm microcontroller to read, filter, and transmit different sensor data. Experience using CAN bus protocol to create a smooth network to transfer sensor data.

Daniel Chrisman - Front-end design and integration with the backend. Project management and documentation background from previous courses. Have worked with app development and work with data security. A background in field experience helps drive my communication skills within the team to ensure we are all on the same page when it comes to individual tasks to complete a larger-scale image.

2. Strategies for encouraging and support contributions and ideas from all team members:

Setting and sharing goals individually and as a team with announcements when they are reached. Healthy working environments that pay respect to each member's ideas.

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will

a team member inform the team that the team environment is obstructing their

opportunity or ability to contribute?)

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

Secure access to all necessary APIs

Secure a web hosting platform

Deepen understanding of necessary tools/languages

Design the graphic interface to our website

2. Strategies for planning and assigning individual and teamwork:

Individual portions of the project will be decided during the Tuesday meeting time and will serve to accomplish the team's weekly goal.

3. Strategies for keeping on task:

Claim tasks bit by bit and work on them throughout the week. If you get stuck or overwhelmed, let the team know sooner than later.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

Communicate the infractions with the team first, and try to solve the root of the issue. If necessary, ask the advisor for help.

2. What will your team do if the infractions continue?

Communicate with Professor Shannon/Dr. Daniels depending on the semester about how to handle the uncooperative teammate.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the

consequences as stated in this contract.

1) Daniel Chrisman	DATE 2/19/2023
2) Ella Knott	DATE 2/19/2023
3) Michael Wieland	DATE 2/19/2023
4)Lith Almadani	DATE 2/19/2023
5) Christian Boughton	DATE 2/19/2023