Fair Home Pro (FHP)*

Justin Harris
Department of Computer Engineering
University of Idaho, USA
harr1433@vandals.uidaho.edu

Jeremy Wisecarver
Department of Computer Science
University of Idaho, USA
wise9314@vandals.uidaho.edu

ABSTRACT

SQL is a domain-specific language that is used for managing data in relational database management systems. In this document, we focus on the development of a Fair Home Pro project given to us in CS360 (Database Management) at the University of Idaho. The project goal was to design a system from the ground up that uses MySQL as the backend, a front-end web system, and a middle-end system to query the MySQL database.

FHP CONCEPTS

• MySQL; Web-Based *Graphical user interface (GUI); RDMS -> Relation Database Management System*;

KEYWORDS

SQL; PHP; MySQL; JavaScript; HTML; Database; PHP;

---

1 INTRODUCTION

The scope of this paper is to describe the inner workings of our Fair Home Pro project. From here on out FHP will refer to Fair Home Pro. The main goal of the FHP project is to create a website where a homeowner can get a service contract with a service professional at the lowest price possible. The FHP project has two major components to it. The front end, also known as the user interface (UI) is the first component. The UI is talked about in section 3. The back end, also known as the Relational Database Management System (RDMS) is the second component. The RDMS is mentioned in section 1.1, 2, and 4.

1.1 RDMS using MySQL

For our RDMS, we are using MySQL to host our database schema "fairhomepro". MySQL is one of the most popular open-source SQL databases and was the primary query language taught in CS360. Since MySQL is widely used, it has a lot of forum and community support to assist us when creating our database. Although MySQL lack good performance scaling for larger databases, it was the correct choice for the FHP project's database.

1.2 Tools used in Development

Through-out the development of this project, multiple tools were utilized by each member of the project. The tools used in our development were Visual Studio Code (VS code), phpMyAdmin, Github and Github Desktop, Discord and XAMPP. VS Code was the primary IDE used to write, compile, test, and review source code. Each team members had multiple extensions installed, which they were allowed to choose themselves, in VS Code to make the development process as easy as possible. All team members used phpMyAdmin to create, manage, and test the FHP database. The management of the source code for this project was handled through GitHub and GitHub Desktop. Using GitHub allowed us to create a centralized master repository of our source code. From the master repository, team members would then create a separate branch off the master repository to develop individual portions of the overall project. Once the individual portions were finished, team members could then create a pull request to merge the branch back into the master repository. This enabled the team to compartmentalize the project into smaller portions, allowing team members to stay focused on one task at a time to piece the whole project together. All team communications and meetings were handled through Discord. We would regularly post references, progress updates, and questions in Discord which allowed us to stay up to date on the overall development of the project. XAMPP was just to install and run the PHP and MySQL services on each team members development device.

1.3 Languages used in Development

The languages used to create this project include PHP, HTML, MySQL, JavaScript (JS) and Cascading Style Sheets(CSS). Since this project mainly involved sending and retrieving data to and from a database, using MySQL alongside PHP was an obvious choice for us especially since MySQL was the primary language used throughout the CS360 course. Using PHP enabled us to easily connect directly to our database, as well as query, update, and insert data without the need for additional API packages or routing options that other languages like JavaScript would require. Since PHP was chosen as our primary language to run the back end of our project, HTML was the best option to use for this project's front end. PHP and HTML work together simply by declaring the PHP source file as an HTML type of document inside of the source file. Then, if we needed to use PHP within the HTML declaration, we would just use PHP tags to identify those specific sections as PHP code. The JS and CSS were used in tandem with the HTML to create additional functionality and styling for the project. Using JS were able to allow additional windows to be displayed on top of our webpage instead of always redirecting to another page. CSS was used to add additional styling using inline CSS directly coded into the HTML sections of code and global style sheets from Bootstrap. Additionally, some CSS style sheets were created by team members to customize additional styling. Some of the biggest challenges that we faced was learning how to use each of these coding languages separately and using them together. Neither member had used any of the languages used in this project prior to the assignment of this project, so there was a steep learning curve concerning how to use the syntax of each of these languages.

2 FairHomePro DATABASE

The core functionality of this project is built into the *fairhomepro* database schema, which was made according to the FHP requirements. The schema consists of 13 different tables that has some interaction between itself and another table. These tables will be discussed further in this section.

2.1 Homeowners Table

	HO_name	HO_email	password	HO_phone	HO_creditcard	HO_bankaccount
Edit Copy Delete	Barry Barnes	bbarn@uidaho.edu	asdfg	2147483647	9856987456321523	2531485698563214
Edit Copy Delete	Jeremy Wisecarver	jeremy@email.com	789456	2147483647	20363025587416 95	20369854785210 36
Edit Copy Delete	Justin Haris	jmharris7755@gmail.com	123	2088184087	123	123
Edit Copy Delete	Keller Lawson	klaws@hotmail.com	asdf.lkj	2147483647	1236554785201236	1236547852145789
Edit Copy Delete	Puala Freds	pfreds@duckgo.com	qwerty	2147483647	74856254100320 21	60129856474123 02
Edit Copy Delete	Winter Campbell	wecamp@gmail.com	hello	2085555555	45617897456778 94	123012301230123

Figure 1: Homeowners Table

As shown in Figure 1, the Homeowners table is defined by six columns. The *HO_name* column is the homeowners' full name. The *HO_email* is the homeowners unique identifier. The *password* is the password for the homeowner's account. The *HO_Phone* is the homeowner's phone number, and the *HO_creditcard* and *HO_bankaccount* are the homeowner's credit card and bank account information. An important note here is that the *HO_creditcard* and *HO_bankaccount* are allowed to be a null value, and are not required unless the homeowner wants to purchase a service contract, which is talked about later on.

2.2 Homes Table

	home_ID	street	city	state	zip	constr_type	floors	h_sq_ft	y_sq_ft
Edit Copy Delete	1	123 Hedgehog Lane	Post Falls	AR	83815	55+	Laminate	2222	2222
Edit Copy Delete	2	789 Test Street	Spokane	WA	99602	Log Home	Vinyl	444	444
Edit Copy Delete	3	123 Test Street	Post Falls	CA	88888	55+	Marble	1111	2222
Edit Copy Delete	4	1001 Happy Ave	Lucky Town	CO	88523	Production Home	Vinyl	800	100
Edit Copy Delete	5	555 Pillow Street	Sleep Town	GA	44566	Custome Home	Hardwood	4000	1000
Edit Copy Delete	6	908 Barnes Ave	Santa Clara	CA	99052	Timber Frame	Porcelain	523	210
Edit Copy Delete	7	603 Gerund Blvd	New York	NY	88750	Modular Building	Ceramic	500	500

Figure 2: Homes Table

As shown in Figure 2, the Homes table consists of nine columns. The *home_ID* column is the a unique integer identifier assigned to each home that a homeowner registers to their account. The *street, city, state,* and *zip* columns are the home's address that the homeowner registers. The *constr_type* column indicates the construction type of the home. The *floors* column is the primary type of flooring that the home has. The *h_sq_ft* and *y_sq_ft* columns are the square footage of the home and yard respectively. An important note on this section is that the *home_ID* is not a value known to the homeowner. This field is used internally so that the FHP back end can identify which homeowner own which homes.

2.3 Owns Table

home_ID	HO_email
1	jmharris7755@gmail.com
2	jmharris7755@gmail.com
3	wecamp@gmail.com
4	jeremy@email.com
5	pfreds@duckgo.com
6	bbarn@uidaho.edu
7	klaws@hotmail.com

Figure 3: Homes Table

The Owns table contains only two columns, the *home_ID* and the *HO_email*. The *home_ID* column is the same *home_ID* from the Homes table, and the *HO_email* column is the same *HO_email* column from the HJomeowners table. Both of these fields are foreign keys to the Owns table, which uses these values to correctly route a homeowner to each of their homes.

2.4 Plant_Types Table

plant_ID	plant_type
1	Begonias
2	Fuchsia
3	Geraniums
4	Abutilon
5	Caladium
6	Rose Bushes
7	Boxwood and Myrtle

Figure 4: Plant_Types Table

The Plant_types table consists of two columns. The purpose of this column is to have an available list of plants that a homeowner can

add to each home. The first column is the *plant_ID* column. This is an incremented integer that assigns each plant type a unique ID. The second column is the *plant_type* column. This column holds types of plants that homeowners are able assign their homes.

2.5 Has_Plant Table

home_ID	plant_ID
1	6
2	3
1	2
2	6
1	4
2	5
3	5
3	2

Figure 5: Has_Plant Table

This table contains two columns that are both foreign keys from other tables. The *home_ID* column, which correspons to the Homes Table's *home_ID*, and the *plant_ID* column corresponding to the Plant_Types *plant_ID* column. This table associates which homes have what types of plants.

2.6 Service_Pros Table

Business_Name	SP_email	SP_password	SP_phone	SP_creditcard	SP_bankaccount
Blue Cross Blue Shield	bluecross@gmail.com	123	123	123	123
Jim's Windows Cleaning	jwindows@hotmail.com	Happy.Jim	5556982360	1023658965412365	6452369856478521
Lawn Service Guys	lsgw@gmail.com	123456	5559865201	123456789	123456789
Moes Mowers	moemower@moemowers.com	Moemower	4256985233	4568713256981200	10023658874569 32
Windows Xpress	winxpress@outlook.com	Winexpress	8062365489	6201589645893256	6201589645893256

Figure 6: Service_Pros Table

Service_Pros keeps track of the Service Professional's business information. The *Business_Name* column holds the name of the business. The *SP_email* is the unique identifier for the Service_Pros table, much like the Homeowners Table. The *SP_password* field stores the Service Professional's password, which would normally be encrypted, but for testing purposes these were left unencrypted. Finally, the *SP_phone*, *SP_creditcard,* and *SP_bankaccount* columns hold the phone number, credit card and bank account number for each Service Professional.

2.7 Service_Types Table

service_ID	service
1	Lawn Mowing
2	Window Cleaning
3	Hedge Trimming
4	Garden upkeeping

Figure 7: Service_Types Table

This table was designed to hold a list of available services that a Service Professional can offer. The *service_ID* column is a unique auto-incremented ID number for each unique service in the *service* column.

2.8 Services Table

SP_email	s_type	s_price
bluecross@gmail.com	Lawn Mowing	3
jwindows@hotmail.com	Window Cleaning	100
winxpress@outlook.com	Window Cleaning	50
moemower@moemowers.com	Lawn Mowing	18

Figure 8: Services Table

Once a Service Professional registers the types of services that they offer, they are stored into the Services Table. In this table we track the unique identifier of the *SP_email* column to the *s_type* and *s_price* columns, where *s_type* is the service type and *s_price* is the service price.

2.9 Specialties Table

SP_email	service_ID	specialty_ID
bluecross@gmail.com	1	1
jwindows@hotmail.com	2	2
winxpress@outlook.com	2	3
moemower@moemowers.com	1	4

Figure 9: Specialties Table

Here every Service Professional can register a service as a service that they specialize in, or a specialty. The *SP*_email column is the Service Providers email used to identify which service (*service_ID*) is the specialty service (*specialty_ID*).

2.10 Transaction Table

transaction_ID	HO_email	SP_email	service_type	price	contract_ID	transaction_date
1	jmharris7755@gmail.com	bluecross@gmail.com	Lawn Mowing	3	1	2021/12/09
2	jmharris7755@gmail.com	bluecross@gmail.com	Lawn Mowing	3	2	2021/12/09
3	wecamp@gmail.com	bluecross@gmail.com	Lawn Mowing	3	3	2021/12/09
4	wecamp@gmail.com	bluecross@gmail.com	Lawn Mowing	3	4	2021/12/09

Figure 10: Transation Table

The Transaction Table stores information about service purchased between a Service Professional and a Homeowner. This table has a unique identifier of a *transaction_ID*, which is an integer assigned to each transaction. The *HO_email* and *SP_email* columns store the Homeowner and Service Professionals email address, to identify which Homeowner purchased a service from which Service Professional. In the *service_type* and *price* columns, we can see what service was purchases for what prices. The *contract_ID* column references the *contract_ID* relating to the Contracts table, and the *transaction_date* is the day, month, and year that the transaction was made.

2.11 Contract Table

contract_ID	contract_date	service_type	price	SP_email	HO_email
1	2021/12/09	Lawn Mowing	3	bluecross@gmail.com	jmharris7755@gmail.com
2	2021/12/09	Lawn Mowing	3	bluecross@gmail.com	jmharris7755@gmail.com
3	2021/12/09	Lawn Mowing	3	bluecross@gmail.com	wecamp@gmail.com
4	2021/12/09	Lawn Mowing	3	bluecross@gmail.com	wecamp@gmail.com

Figure 11: Contract Table

This table stores all of the contracts made between homeowners and Service Professionals. This contains all of the same fields as the Transaction Table, except for the unique identifier of the *transaction_ID*. The Contract table is used by both homeowner and service providers to track contracts that they have made with each other for service to be performed.

2.12 Complaints Table

complaint_ID	contract_ID	contract_date	complaint_date	complaint_text	SP_email	HO_email
1	1	2021/12/09	2021/12/09	no like	bluecross@gmail.com	jmharris7755@gmail.com
2	3	2021/12/09	2021/12/09	Not Satisfactory	bluecross@gmail.com	wecamp@gmail.com
3	4	2021/12/09	2021/12/09	Rude employees	bluecross@gmail.com	wecamp@gmail.com

Figure 12: Complaints Table

The Complaints Table keeps track of complaints that either the homeowners or the service professionals make about each other. The unique filed of the Complaints Table is the *complaint_ID* column. This is an incremented integer to track the complaint. Contained in the Complaints table is the Service Professionals email address (*SP_email)*, the homeowners email address (*HO_email*), the contract date(*contract_date)*, and the date that the complaint was made(*complaint_date)*. Upon a Homeowner making a complaint, they should be issued a full refund for the paid service.

2.13 Licenses Table

SP_email	licenses

Figure 14: Licenses Table

Finally, the Licenses Table. This table was mentioned last because it was the only table that was not fully implemented. This table would go in the sections where the Specialties and Services tables were talked about if it had. However, the concept behind the Licenses table is that a service professional would be able to specify what types of licenses they had for various services. If a Home Improvement service was offered, the Service Provider could then list that they had a General Contracting License. The type of license would be identified by the *SP_email* and *licenses* category together as a tuple.

3 FHP USER INTERFACE

The FairHomePro User interface was created using a variety of web-development languages mentioned above, such as HTML, JS and CSS. The main goal of the User Interface for the Homeowner is to be able to create an account to get a bid for a service at the lowest price it is offered. For a Service Professional, the goal of the User Interface is to create an easy-to-use GUI to register their business, automatically receive contracts and payments from customers, and add what types of services they offer.

3.1 FHP Landing Page

Upon entering the FHP site, any visitor will be greeted with the FHP Landing page. The user is greeted with a soft purple-fade background, which is the general theme that we went with for our site. Here they will get an introduction to the website and have options to create an account as a Homeowner or a Service professional. If the user already has an account, they are provided with an option to sign in.

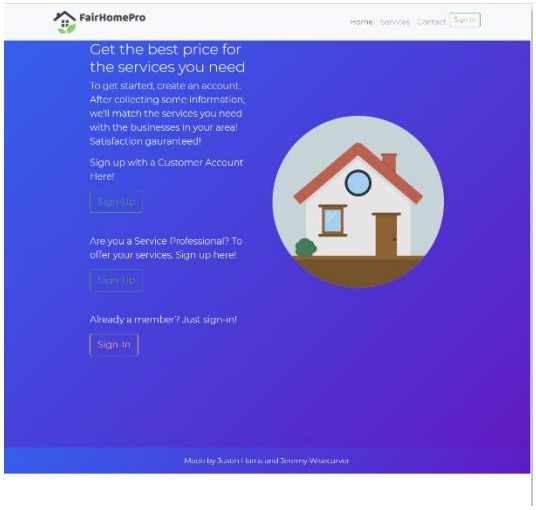


Figure 15: FHP Landing Page

If a new visitor is curious about the site, they are able to navigate around in a limited capacity. Outside of the options to sign up or sign in, a new visitor is only able to click on the Services or Contact options on the Navigation bar. If a new user navigates to the Service tab they are greeted by a page similar to the home page, that instructs the user to sign in or create an account to purchase services.

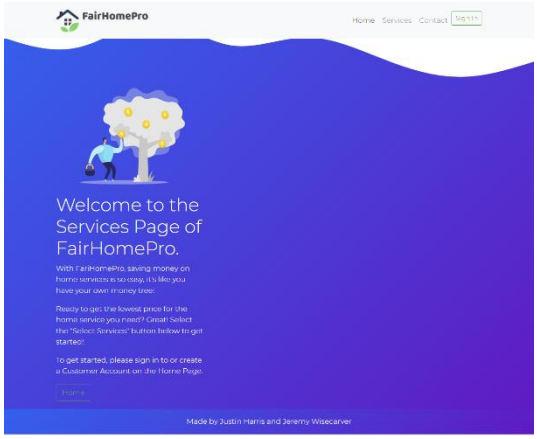


Figure 16: FHP Services Page Not logged in

The FHP Contact Page is the same for all visitors to the webpage, whether they are not signed in, signed in as a Homeowner, or signed in as a Service Professional. It consists of a simple for where the visitor can enter in the email address that they want to contact us at and send a comment.

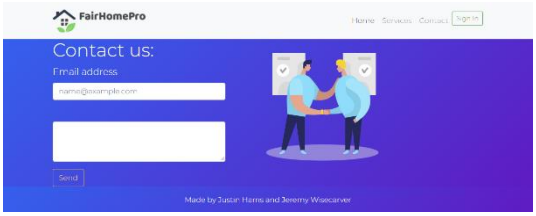


Figure 17: FHP Contact Page

3.2 FHP Homeowner Pages

If the user decided to sign up for an account as a Homeowner they will greeted by a simple signup form.

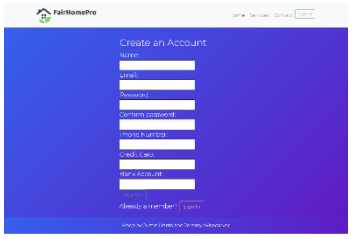


Figure 18: Homeowner Sign-up Page

This Sign up page allows homeowners to create an account with their full Name, email address, phone number, credit card and bank account numbers. There is additionally a two fields for the homeowner to enter in their password. This is to have some verification of the what the homeowner wants to set their password to. After signing up, the homeowner will be redirected to adding in information about their home.

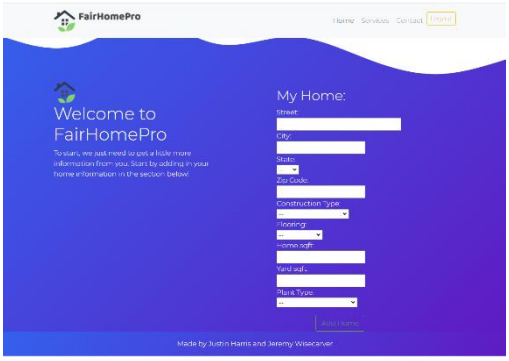


Figure 19: Add A Home Page

The fields for the Homeowners home are left empty for them to fill out. Here the Homeowner will type in the home's Street, City, Zip Code, Home square footage, and Yard square footage. Drop down menus are provided for the home's State, Construction Type, Flooring and Plant Type that the home has. After filling out this information, the homeowner is redirected to a new homepage.

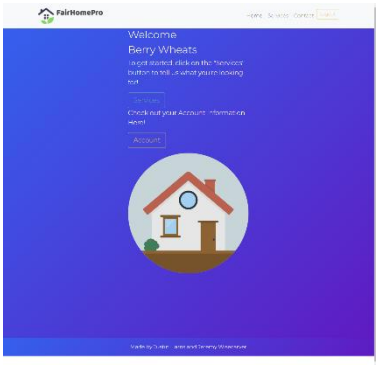


Figure 20: Customer Landing Page

On this page, this Homeowner is greeted by the name that they typed registered to the Account. From here, the Homeowner can check out their Account Page, or view the Services Page.

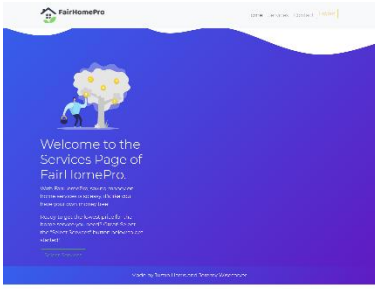


Figure 21: Customer Services Page

On this page, the homeowner is greeted with a warm welcome and instructed to select the Select Services button to continue.

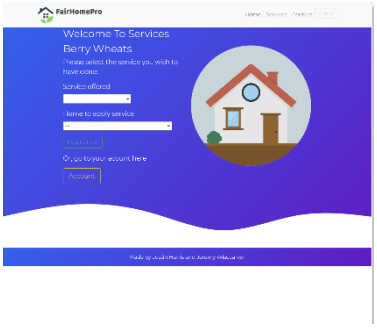


Figure 22: Customer Services Page

On this page, the Homeowner can find two dropdown menus. One drop-down menu is for the Homeowner to select which service they want, and the other is to select which home the service will be getting done on. Once the Homeowner has selected these, they can request a quote by clicking on the Request bid button. After this a hidden field will show up that confirms that the Homeowner wants get the service they selected done on the home that they selected, as seen in Figure 23. Then, if the Homeowner selects that they did want this service, they will be redirected to the Payment Page, seen in Figure 24.

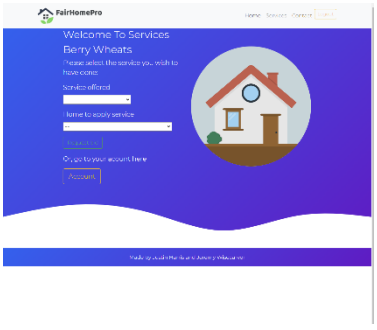


Figure 23: Customer Services Page

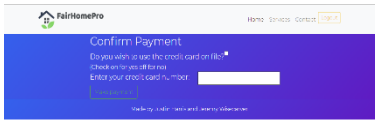


Figure 24: Payment Confirmation Page

On the Payment Confirmation Page, the customer can either enter a new credit care to use, or select a check box to use on that is on file. After setting up the payment information and selecting the Make Payment Button, they will be redirected to their account.

3.3 Homeowner Account Page

In the Homeowner Account Page, the homeowner can view and edit the information that they registered the account with, except for the email address. They can also add and edit the homes that they've registered, add additional plants to homes, or file a complaint about an service contract that they purchases.

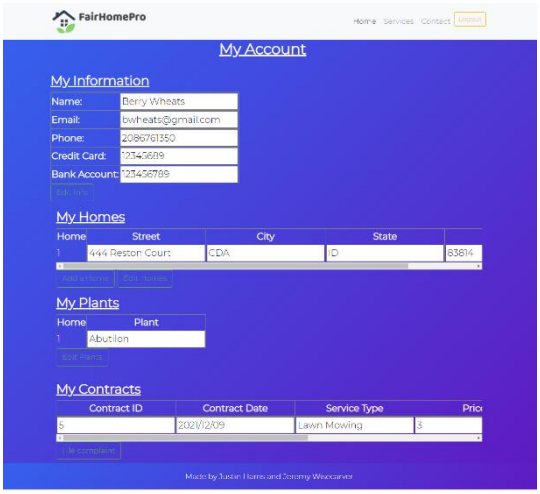
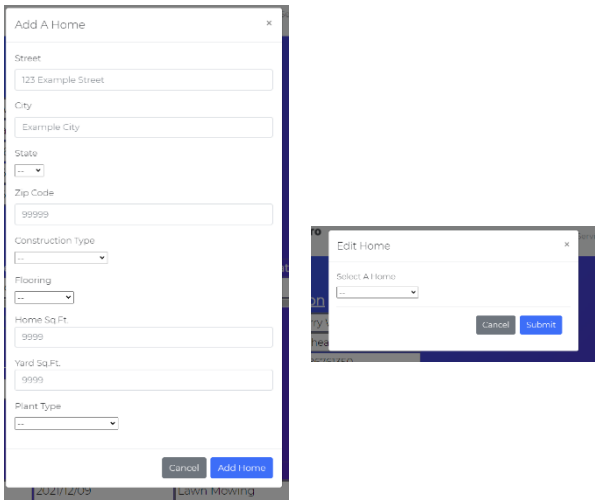


Figure 25: Homeowner Account Page

If the homeowner wants to add a home to their account, they can click the Add a Home button. From here a modular pop-up window will come up that has the whole form for the Homeowner to add a new home with. Once the form closes, the home will be added to the customer's account. Similarly, clicking the Edit Homes button will bring up a pop-up windows where the Homeowner can select which home they want to edit. This can be seen in Figures 26 and 27.



Figures 26 and 27: AddHome and editHome Modals

After selecting a home to edit from the Edit home Modal, the Homeowner will be redirected to the Edit Home Page. Here the information for the home that they selected will be prefilled out. After editing the home details and selecting Submit, the homeowner will be redirected back to the Account page.

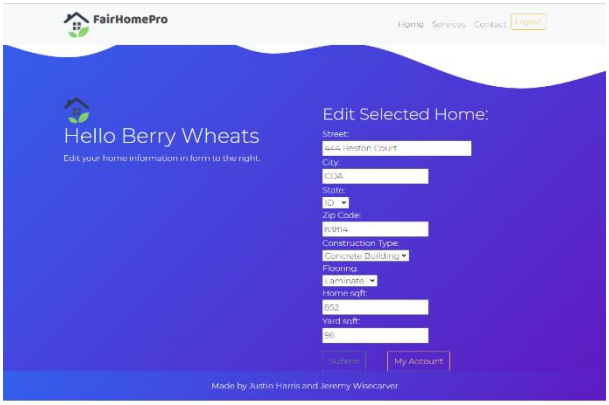


Figure 28: Edit Selected Home Page

Back from the Account Page, if the Homeowner selects the Add a Plant button, they will be redirected to the Add A Plant Page.

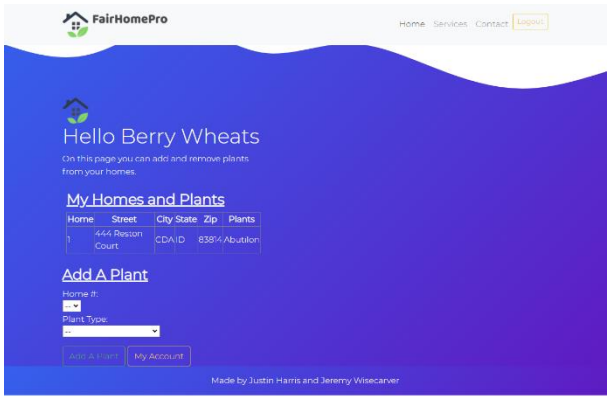


Figure 28: Edit Selected Home Page

On this page the Homeowner can see all the plants that each home has. To add a plant type to a home, the homeowner simply needs to select the Home# drop down to specify which home, according to the table above, and then select the plant type that they want to add. Selecting the Add A Plant button will add the plant to the table in the middle of the screen.

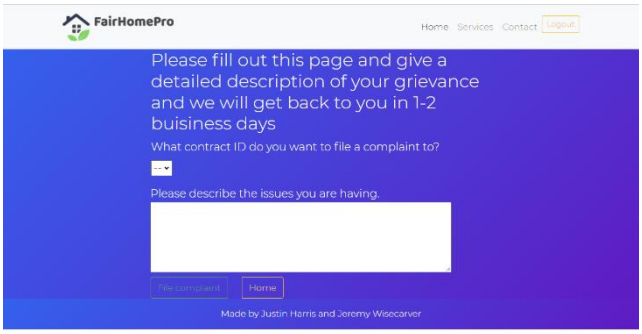


Figure 29: Complaint Page

The last thing that a Homeowner can do is to file a complaint about a service that they have purchased. All of the Homeowner's contracts can be seen at the bottom of their account page. By clicking the button to make a complaint, they will be redirect to the for in Figure 29.

3.4 Service Professional Pages

If a visitor is a Service Professional that wants to offer their services through FHP, they can sign up as a Service Professional. The form is very similar to the customer account creations, except that the Name field is replaced with the Business name field.

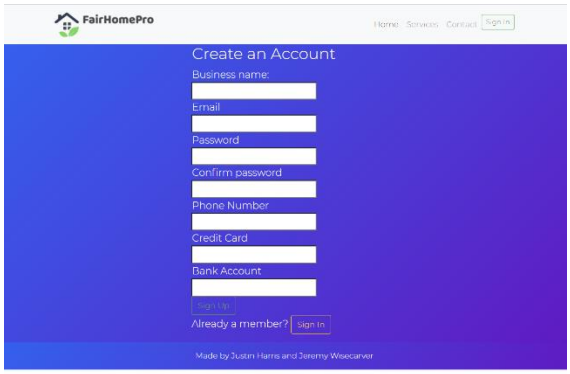


Figure 30: Service Professional Sign up

From here the Service Professional will be redirected to a specific homepage just for them.

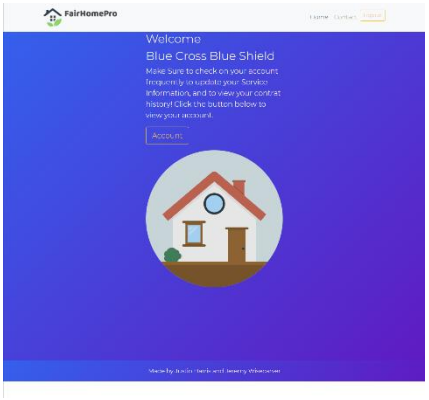


Figure 31: Service Professional Landing page

As it can be seen in Figure 31, there is not a lot to do from the Service Professionals Landing Page. All of the Service Professionals actions are handled through the Service Professionals Account. Even the "Services" link in the navigation bar is not available, because this is also handled through the Account.

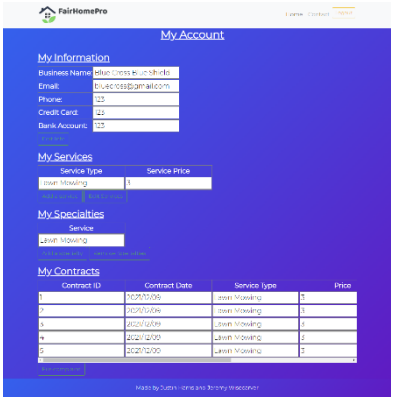


Figure 32: Service Professional Account Page

From the Service Professionals Account Page Service professionals are able to edit their business information, including adding services that they offer, editing current services, adding specialty services, and removing specialty services, and filing complaints about Homeowners.

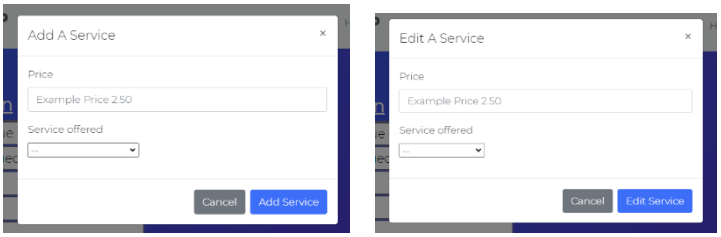


Figure 33: Add and edit service Modals

The pop-up windows to add and edit services work similarly but have slight differences to them. The Add A Service Modal returns the full list of services that Service Professionals are able to offer and lets them set a price for the service. While the Edit a Service Modal only shows service that the Service Professional has already registered and allows them to adjust the pricing on the service.

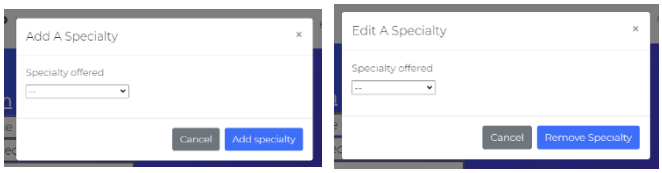


Figure 34: Add and edit Specialties

Another two pop-up modals that operate similarly are the Add A Specialty and Edit A Specialty Modals. The Add A Specialty Modal allows a Service Professional to select a Service from the list of services that they offer to register as a Specialty. Similarly, using the Edit A Specialty windows, the Service Professional can remove any service that they have registered as a Specialty, and make it just a normal service again.

Finally, the Service Professionals can make complaints about Homeowners if necessary. However, unlike the Homeowners, Service Professionals are not offered any sort of monetary compensation for having problems with a Homeowner.

4 MYSQL QUERIES

For this project, we ended up creating around 30 queries for out database that are executed situationally, from multiple pages, depending on what the user selects to do. To see detailed information on our queries, please see reference [2] for a link to the server.inc.php file hosted on our Github. Since all our queries are used situationally, there isn't one particular query that we particularly used the most. However, one of the favorite queries inside of a PHP function called account_create_home_table(). The query itself is about 80 lines of code, so it's too much to put into a paper. However, this query really tied together all of PHP, HTML, and MYSQL we learned this semester. The basic rundown of the query is that it queries the database for the homeowners address information and dynamically creates the tables on the page from within the PHP code. It also incorporated adjusted column names for each column field to give them a cleaner look.

5 Conclusion and Future Improvements

Throughout the duration of this project, there has been much time to reflect on the shortcomings. As noted in Section 2.13, there were quite a few features that we were unable to implement. Part of this was due to time mismanagement on the project. Implementing a stricter Agile workflow would have greatly helped us stay on track, implement our features, and accomplish our goals for this project. Additionally, a Bootstrap theme was mainly used for the styling of this project. Unfortunately, the styling chosen had very limited options for color scheme and most of them are very hard to see against the background of the Bootstrap theme. Another change that we would have made would be to change the way that the Homeowners receive their services. Giving Homeowners additional options to search for a service based on off if a Service Professional has certain Licenses or Specialties would provide a more thorough search to meet the needs and wants of each Homeowner. Even further improving on our pricing. We did not get prices to print out by square foot or yard. We could have the price per square foot or yard of the service, and then multiply it by the square footage of the home or yard to give the homeowner a complete cost for their home, as well as the general pricing.

References

[1] 2021. GitHub Repository CS360- https://github.com/jmharris7755/FairHomePro/ :December 9, 2021.

[2] 2020. GitHub Repository CS360, server.inc.php - https://github.com/jmharris7755/FairHomePro/blob/New-Master/bsphp/components/server.inc.php: December 9, 2021.