

CSCI 6333/6315 Database Design and Implementation

Part C: Home Needs Services

Website:

<http://ec2-18-234-191-251.compute-1.amazonaws.com/homeneedsservice/>

Group 1: DataBuddies

Mariana Martinez 20286238

Christian Narcia 20324290

Sergio Soto 0375494

Maricruz Flores 20267250

Index:

1. Introduction
 - 1.1. Problem Statement
2. Database Design
 - 2.1. ER Design
 - 2.2. Relational Database Design
3. System Design
4. Dependency Description
5. Interface Description
6. Implementation
 - 6.1. Database Implementation
 - 6.1.1. Account_Creation
 - 6.1.2. Account_Login
 - 6.1.3. Calendar
 - 6.1.4. Customer
 - 6.1.5. Ints Description
 - 6.1.6. Needs
 - 6.1.7. Offers
 - 6.1.8. Participants
 - 6.1.9. Provider_Availability
 - 6.1.10. Provider_Details
 - 6.1.11. Reserv_Details
 - 6.1.12. Service
 - 6.1.13. Service_Provider
 - 6.1.14. Service_Reserv
 - 6.1.15. Service_Review
 - 6.1.16. Transaction
 - 6.2. Server-side Implementation
 - 6.2.1. Homepage
 - 6.2.2. Login/Signup Page
 - 6.2.3. Service Provider Page
 - 6.2.4. Customer Page
7. Usability
 - 7.1. Homepage
 - 7.2. Login/Signup Page
 - 7.3. Service Provider Page
 - 7.4. Customer Page
8. Testing
 - 8.1. Arrive at Home Page
 - 8.2. New User Sign Up
 - 8.3. Customer Home Page
 - 8.4. View Services By Filtering>Selecting Service
 - 8.5. View All Service Providers
 - 8.6. View A Service Providers Available Times
 - 8.7. Schedule A service

- 8.8. Cancel A Schedule Service
- 8.9. User Log In
- 8.10. User Sign Out
- 9. Contributions
- 10. Conclusions
- 11. References

1. INTRODUCTION

1.1. Problem Statement

There is demand for a platform that provides ease of access to information and resources for local based services and prospective customers. Our services project, HomeNeedsService.com, needs an online Marketplace that arranges home needs service offers to prospective customers. In order to provide a productive and functioning platform for someone who will be offering specific services we must provide a login option so they can have an area to adjust the information they would like others to see, offer control, and privacy. The login function will identify the service provider with their services ID. Like the service providers, customers will have the opportunity to have their unique account. The website should be able to provide current and new customers the ability to sign up and login to their accounts. This service ID specifies a service provider's background information such as address and name. The provider has the opportunity to identify their services, connect their prices, available times, and personalize their preferences. These preferences will allow a service provider to easily connect with a customer. This gives the opportunity to connect the right service to the right customer. The purpose is to create a system that provides the best result to both the service provider and customer. A customer will be able to find the services they need and be able to schedule, see or cancel their services. In addition to connecting services to customers, the service provider and customer will have the opportunity to adjust their service commitment. Each one will have the opportunity to view and cancel the designated service at any or for any desired time. The problem with the current website is the lack of communication between services provided and customers needing help. By having a place where both customer and service provider can report back to, both can have a more trustworthy experience where they can choose according to their needs and preferences whether it be cost, proximity or abilities.

2. DATABASE DESIGN

2.1. ER Design

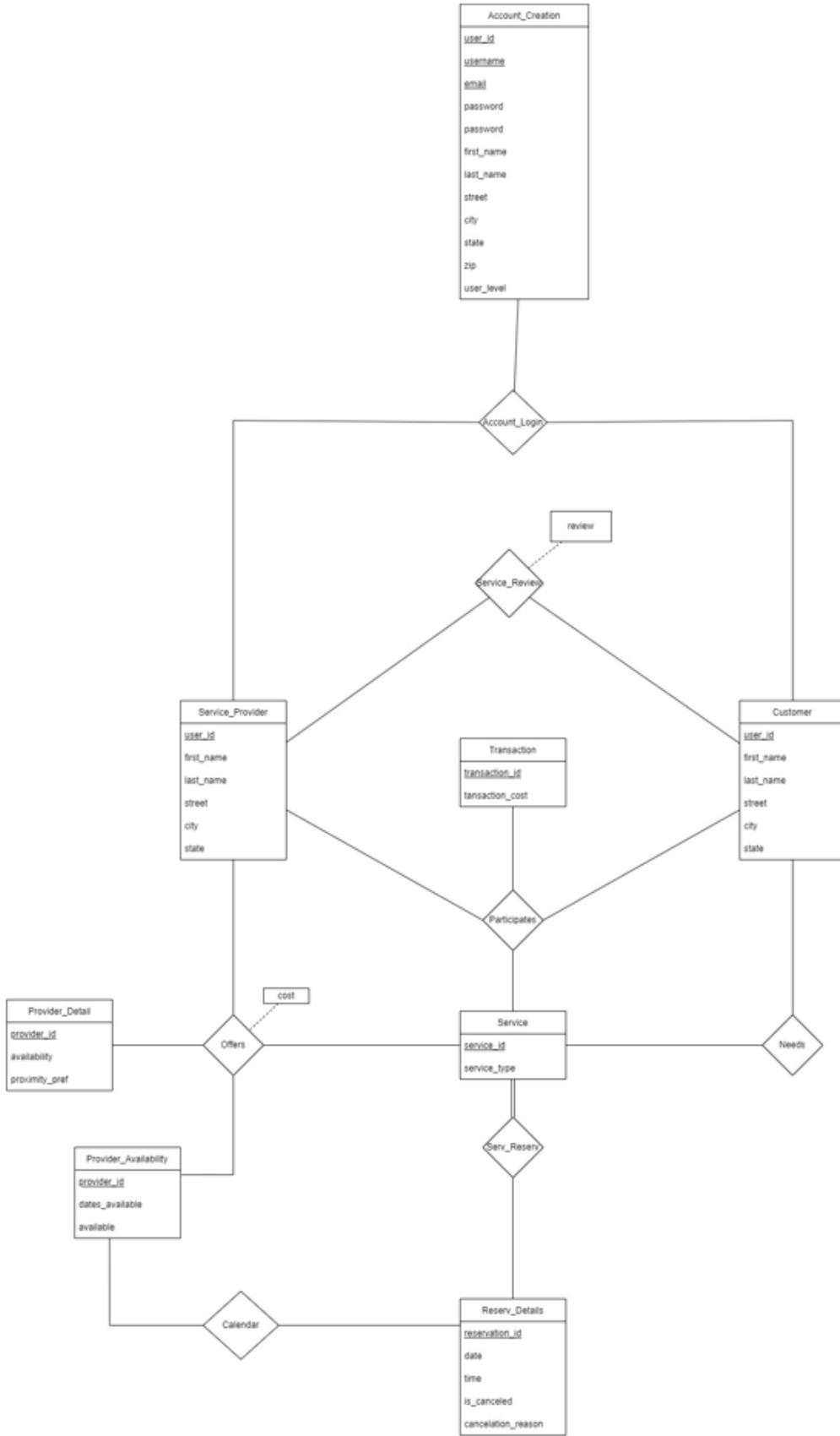


Figure 1.1 : ER Model

2.2. Relational Database Design

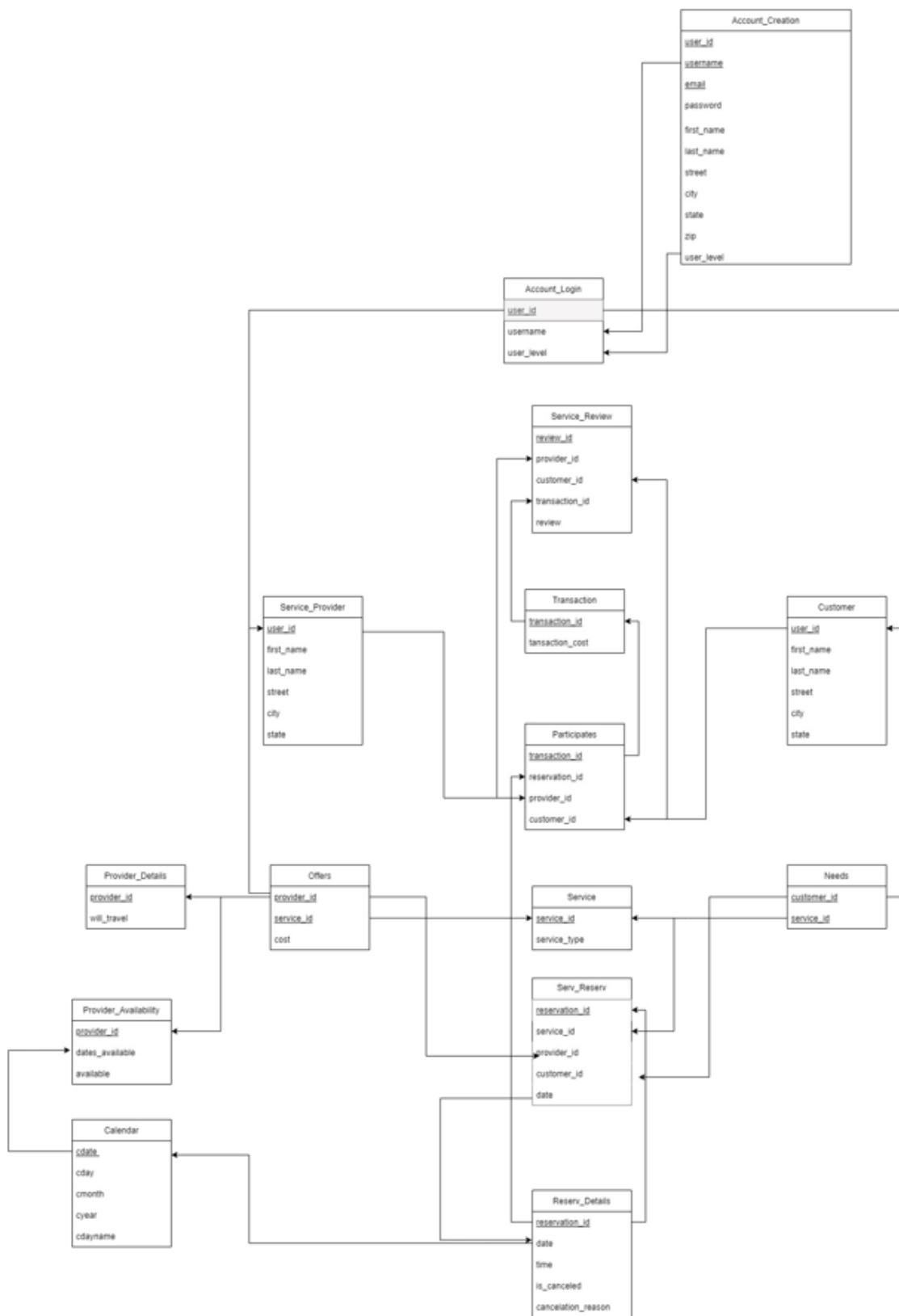


Figure 2.1 : Relational DB Design

3. SYSTEM DESIGN

3.1 Decomposition description

The HomeNeedService website consists of 3 modules, DataStore, DataProcessing, and UserInterface Modules.

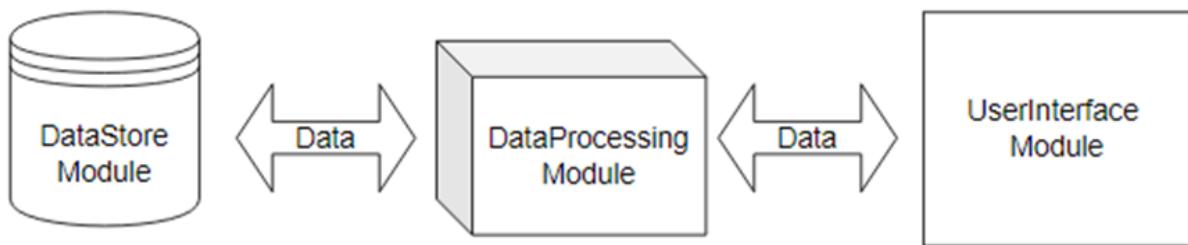


Figure 3.1 : Structure chart of HomeNeedService software framework

3.2 DataStore Module description

The DataStore Module for HomeNeedsService consists of a collection of tables that are relational to one another (Relational Data Model). For a complete description, refer to the ER model included in Figure 1.1.

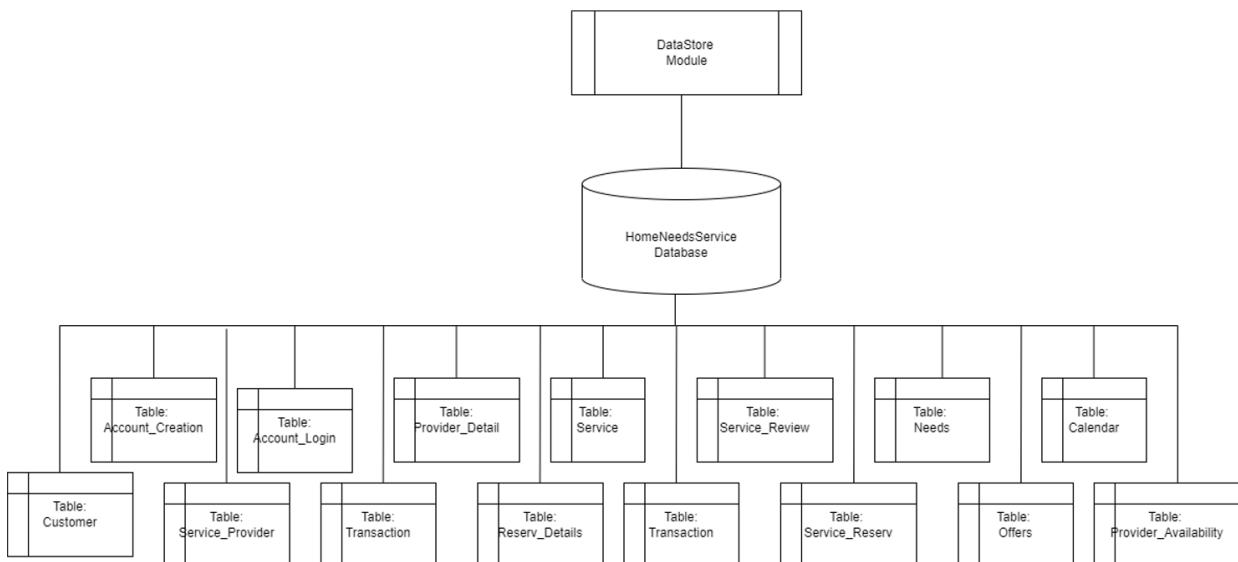


Figure 3.2 : The Structure of DataStore Module

3.3 DataProcessing Module Description

The DataProcessing module has the operations required for the HomeNeedService to provide the needs for all users. This 6-function set module will control all interactions and generate all required data for the user. The function sets are, User Authentication Function Set, Data Update Function Set, Data Retrieval Function Set, Report Generating Function Set, Computation Function Set, and Database Connection Function Set.

The User Authentication Function Set consists of all functions required to authorize users and manage sessions. Data Update Function Set consists of all required functions to update, modify, and create data for all users. Data Retrieval Function Set consists of all required functions to fetch all required information. Report Generating Function set consists of functions required to generate data for users and the webpage. Computations Function Set include all sub-functions that are needed to perform all computations and validations for the user and webpage. Database Connection Function Set has the functions needed to connect to the required database server.

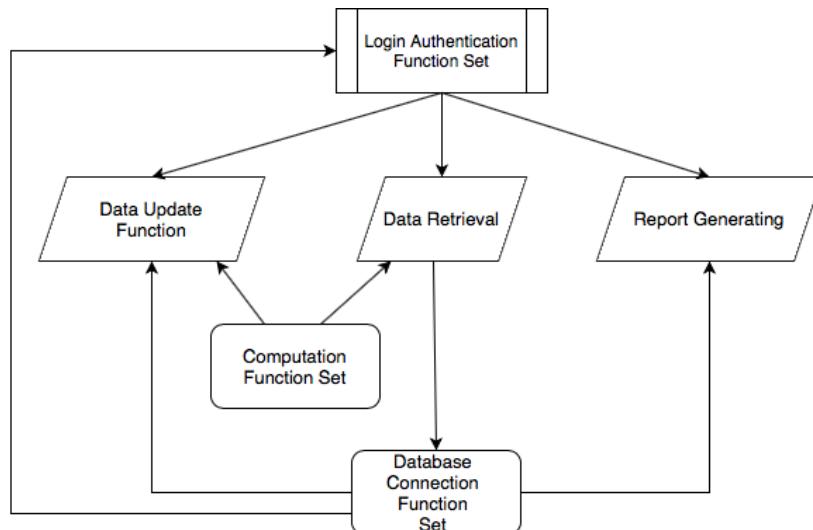


Figure 3.3 : Function Call Relationship Diagram among Function Sets

3.4 UserInterface Module Description:

We use the UserInterface Module to keep all our information concise for our system. It also allows the creation of an interface for users. Displayed in Figure 3.4 you will see the hierarchical decomposition of our DataProcessing module.

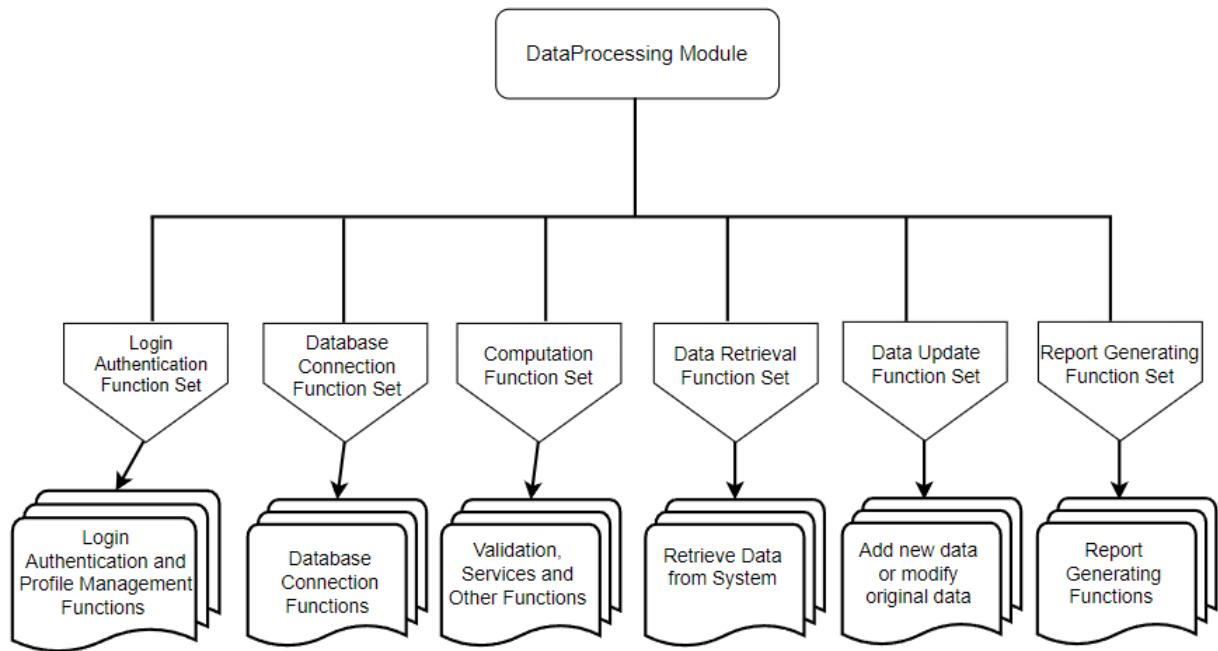


Figure 3.4 : Hierarchical Decomposition Diagram of Data Processing Module

4. DEPENDENCY DESCRIPTION

4.1 Inter-module dependencies

4.1.1 Database

The RDBMS will reside in a MariaDB server (DB server) with the name: *HomeNeedsService*.

The MariaDB server is embedded and administered by phpMyAdmin, an open source software that handles the administration of MariaDB and MySQL servers over the Web.

Table 1 lists the resources on which the DataStore Module depends.

Table 1 Database Server Resources

Hardware	Software

<p>AWS EC2: SSD Volume Type.t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)</p>	<p>Amazon Linux 2 AMI (HVM) - Kernel 5.10</p> <p>Maria DB: Server: 127.0.0.1 via TCP/IP Server type: MariaDB Server connection: SSL is not being used Server version: 10.4.22-MariaDB - mariadb.org binary distribution Protocol version: 10 Server charset: UTF-8 Unicode (utf8mb4)</p>
---	--

4.1.2 DataProcessing Module

The DataProcessing Module is hosted and executed in an AWS EC2 instance (Cloud Computing Service) that connects to an Apache Web Server with PHP through phpMyAdmin. This module accesses DataStore Module and exchanges data with UserInterface Module through phpMyAdmin and AWS EC2. The DataProcessing Module uses PHP language to program and access the data from/to the DataStore Module and UserInterface Module.

Table 2 lists the resources on which the DataProcessing Module depends.

Table 2 Web Server Resources

Hardware	Software
<p>AWS EC2: SSD Volume Type.t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)</p>	<p>Amazon Linux 2 AMI (HVM) - Kernel 5.10</p>

4.1.3 UserInterface Module

The UserInterface Module resides on the same Web Server machine as the DataProcessing Module and shares the same resources. It generates HTML web pages for the users to interact with the system. The web pages can execute on any client PC that connects to the Web Server through the Internet, and can be accessed by any client PC to Web Server via a web browser.

4.2 Inter-process dependencies.

The functions of the DataProcessing Module and static web pages of the UserInterface Module are integrated into PHP and Python files that are requested by the users from the client machines via web browsers and executes as process entities from in the Web Server.

5. INTERFACE DESCRIPTION

5.1 Module Interface

5.1.1 DataProcessing Module Interface description

The DataProcessing Module accesses the database using Python and PHP. The DataProcessing Module updates the Interface generating HTML code and can do http requests through the internet.

5.1.2 DataStore Module interface description

The SQL Server provides access to the DataStore Module. The connection is set up between the database server and the Web Server using a username and password. The database server uses MariaDB type language to carry out operations for database access.

5.1.3 User Interface Module interface description

The web browser is used as an interface for users to interact with the User Interface Module. The Information required by the users is presented through the web browser interface. All input data and requests from the users are sent to the DataProcessing Module through HTTP request.

6. IMPLEMENTATION

6.1 Database Implementation

Various relations were made to maintain, gather, update, and fetch information pertaining to service providers, clients, services, and service reservations.

6.1.1 Account_Creation

The Account_Creation relation contains user information as they create an account. This information includes username, password, email, first name, last name, street, city, state, and zip code. As a user registers, their information will be stored here. Additionally, every user will be assigned a user ID automatically as they create the account..

			user_id	username	email	password	first_name	last_name	street	city	state	zip	user_level	
<input type="checkbox"/>				1	alaunder0	alaunder0@blinklist.com	Upjhj8tz	Alexis	Launder	860 Miller Way	South Padre	Texas	78597	1
<input type="checkbox"/>				2	rwhibley1	rwhibley1@bloglines.com	5jnPoMmh7G	Ruddie	Whibley	65 Melody Terrace	Harlingen	Texas	78550	1
<input type="checkbox"/>				3	ottimash2	ottimash2@webeden.co.uk	VggQrnGq2E	Osborn	Timmarsh	9 Village Point	Edinburg	Texas	78504	1
<input type="checkbox"/>				4	jdwelly3	jdwelly3@yahoo.com	gwd9dOU	Jase	Dwelly	0327 Kim Hill	Brownsville	Texas	78520	1
<input type="checkbox"/>				5	gginger4	gginger4@pagesperso-orange.fr	Jn8VGzg3twN	Gardner	Ginger	1 Jay Alley	South Padre	Texas	78597	1
<input type="checkbox"/>				6	bthomas05	bthomas05@google.com	Qo8lqY9gAK	Beilul	Thomason	89831 Bowman Center	Harlingen	Texas	78550	1
<input type="checkbox"/>				7	cpenvarden6	cpenvarden6@paypal.com	0xE0X9	Carlotta	Penvarden	64061 Briar Crest Lane	Harlingen	Texas	78550	1
<input type="checkbox"/>				8	vdran7	vdran7@privacy.gov.au	Xd65Eqhx	Veronique	Dran	507 Bonner Center	Brownsville	Texas	78520	1
<input type="checkbox"/>				9	nstammler8	nstammler8@dell.com	7BMUD353BF7L	Nate	Stammler	4 2nd Place	San Benito	Texas	78586	1

Figure 6.1

6.1.2 Account_Login

The Account_Login relation contains the user's usernames and references it to its user's ID. It also contains an attribute called user_level that holds a variable 0 if the user is a customer or 1 if the user is a provider.

			user_id	username	user_level
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	1 alaunder0 0
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	2 rwhiley1 0
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	3 otitmarsh2 0
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	4 jdwelly3 0
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	5 gginger4 0
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	6 bthomason5 0
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	7 cpenvarden6 0
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	8 vdran7 0
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	9 nstammler8 0
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	10 lcane9 0
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	11 ServiceTest 1
	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	12 pstubbings1 1

Figure 6.2

6.1.3 Calendar

The Calendar relation contains a service provider's availability dates for service reservations. The information includes day, month, year, and day of the week. As noted, it represents the current calendar that can be reserved. The calendar contains the dates from current date to 12-31-2023 for each provider.

	<input type="checkbox"/>				cdate	cday	cmonth	cyear	cdayname
	<input type="checkbox"/>				2022-01-01	1	1	2022	Saturday
	<input type="checkbox"/>				2022-01-02	2	1	2022	Sunday
	<input type="checkbox"/>				2022-01-03	3	1	2022	Monday
	<input type="checkbox"/>				2022-01-04	4	1	2022	Tuesday
	<input type="checkbox"/>				2022-01-05	5	1	2022	Wednesday
	<input type="checkbox"/>				2022-01-06	6	1	2022	Thursday
	<input type="checkbox"/>				2022-01-07	7	1	2022	Friday
	<input type="checkbox"/>				2022-01-08	8	1	2022	Saturday
	<input type="checkbox"/>				2022-01-09	9	1	2022	Sunday
	<input type="checkbox"/>				2022-01-10	10	1	2022	Monday
	<input type="checkbox"/>				2022-01-11	11	1	2022	Tuesday
	<input type="checkbox"/>				2022-01-12	12	1	2022	Wednesday

Figure 6.3

6.1.4 Customer

The Customer relation contains customer account information that includes first name, last name, street, city, state, and zip code.

	<input type="checkbox"/>				user_id	first_name	last_name	street	city	state
	<input type="checkbox"/>				1	Alexis	Lauder	860 Miller Way	South Padre	Texas
	<input type="checkbox"/>				2	Ruddie	Whibley	65 Melody Terrace	Harlingen	Texas
	<input type="checkbox"/>				3	Osborn	Titmarsh	9 Village Point	Edinburg	Texas
	<input type="checkbox"/>				4	Jase	Dwelly	0327 Kim Hill	Brownsville	Texas
	<input type="checkbox"/>				5	Gardner	Ginger	1 Jay Alley	South Padre	Texas
	<input type="checkbox"/>				6	Beilul	Thomason	89831 Bowman Center	Harlingen	Texas
	<input type="checkbox"/>				7	Carlotta	Penvarden	64061 Briar Crest Lane	Harlingen	Texas
	<input type="checkbox"/>				8	Veronique	Dran	507 Bonner Center	Brownsville	Texas
	<input type="checkbox"/>				9	Nate	Stammler	4 2nd Place	San Benito	Texas
	<input type="checkbox"/>				10	Leshia	Cane	785 Hermina Circle	South Padre	Texas
	<input type="checkbox"/>				56	Sally	Jenkins	3405 Samson	Bay View	Texas

Figure 6.4

6.1.5 Ints

The Ints relation was only used to create the Calendar relation.

Statement used for the creation of the Calendar relation is as follows:

```
+ Options
1   INSERT INTO Calendar (cdate, cday, cmmonth, cyear, cdayname)
2     SELECT cal.date AS cdate, DAY(cal.date) AS cday, MONTH(cal.date) AS
3       cmmonth, YEAR(cal.date) AS cyear, DAYNAME(cal.date) AS cdayname
4     FROM (
5       SELECT '2022-01-01' + INTERVAL d.i*1000 + c.i* 100 + a.i * 10 + b.i
6         DAY AS date
7       FROM ints a JOIN ints b JOIN ints c JOIN ints d
8     ORDER BY d.i*1000 + c.i*100 + a.i*10 + b.i) cal
9     WHERE cal.date BETWEEN '2022-01-01' AND '2023-12-31' ;
```

Figure 6.5

6.1.6 Needs

The Needs relation contains a customer's specific service need. It will match a user's ID to the service specific ID.

		customer_id	service_id		
<input type="checkbox"/>	Edit	Copy	Delete	1	1
<input type="checkbox"/>	Edit	Copy	Delete	2	2
<input type="checkbox"/>	Edit	Copy	Delete	3	3
<input type="checkbox"/>	Edit	Copy	Delete	4	3
<input type="checkbox"/>	Edit	Copy	Delete	5	4
<input type="checkbox"/>	Edit	Copy	Delete	6	4
<input type="checkbox"/>	Edit	Copy	Delete	7	5
<input type="checkbox"/>	Edit	Copy	Delete	8	5
<input type="checkbox"/>	Edit	Copy	Delete	9	1
<input type="checkbox"/>	Edit	Copy	Delete	10	4

Figure 6.6

6.1.7 Offers

The Offers relation contains a service provider's designated service and the cost for the service they are providing. The service provider's ID will be matched to the service specific ID.

	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	provider_id	service_id	cost
						11	1	35.00
						12	1	50.00
						13	2	30.00
						14	3	25.00
						15	1	20.00
						16	4	15.00
						17	4	25.00
						18	1	35.00
						19	5	40.00
						20	2	40.00

Figure 6.7

6.1.8 Participants

The Participants relation contains the service provider, customer, and reservation. The service provider's ID will be matched to the user's ID and the reservation ID.

	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	transaction_id	reservation_id	provider_id	customer_id
						1	5	14	3
						2	2	12	9
						3	4	19	7
						4	1	11	1
						5	3	13	2

Figure 6.8

6.1.9 Provider_Availability

The Provider_Availability relation contains a service provider's availability dates to provide services. Each availability date will be matched to the service provider's ID. After a reservation is made, the available attribute must be changed from its default = 'Yes' to 'No'.

			provider_id	dates_available	available	
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	2022-04-01	Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	2022-04-02	Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	2022-04-03	Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	2022-04-04	Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	2022-04-05	Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	2022-04-06	Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	2022-04-07	Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	2022-04-08	Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	2022-04-09	Yes

Figure 6.9

6.1.10 Provider Details

The Provider_Details relation contains a service provider's preference on whether they are willing to travel to a different city to provide a service. Each service provider's ID will be matched to their travel preference.

			provider_id	will_travel
<input type="checkbox"/>	 Edit	 Copy	 Delete	11 No
<input type="checkbox"/>	 Edit	 Copy	 Delete	12 No
<input type="checkbox"/>	 Edit	 Copy	 Delete	13 Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	14 Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	15 No
<input type="checkbox"/>	 Edit	 Copy	 Delete	16 Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	17 No
<input type="checkbox"/>	 Edit	 Copy	 Delete	18 Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	19 Yes
<input type="checkbox"/>	 Edit	 Copy	 Delete	20 Yes

Figure 6.10

6.1.11 Reserv Details

The Reserv_Details relation contains the reservations that have been made and whether the reservation was canceled by a client. Each reservation's details will be matched to its corresponding reservation ID.

	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	reservation_id	date	time	is_canceled	cancellation_reason
	<input type="checkbox"/>				1	2022-05-24	11:00:00	No	NULL
	<input type="checkbox"/>				2	2022-04-28	13:00:00	No	NULL
	<input type="checkbox"/>				3	2022-05-15	17:00:00	No	NULL
	<input type="checkbox"/>				4	2022-06-01	10:00:00	No	NULL
	<input type="checkbox"/>				5	2022-05-16	09:00:00	No	NULL
	<input type="checkbox"/>				6	2022-08-05	01:00:00	No	NULL
	<input type="checkbox"/>				8	2022-09-04	11:00:00	No	NULL
	<input type="checkbox"/>				10	2022-05-25	17:00:00	No	NULL
	<input type="checkbox"/>				11	2022-05-07	16:35:00	No	NULL
	<input type="checkbox"/>				12	2022-05-27	17:36:00	No	NULL

Figure 6.11

6.1.12 Service

The Service relation contains the list of available service types that are available. Each service type will be matched with an ID to be represented throughout the database.

	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	service_id	service_type
	<input type="checkbox"/>				1	Plumbing
	<input type="checkbox"/>				2	Home Cleaning
	<input type="checkbox"/>				3	Cooling/Heating
	<input type="checkbox"/>				4	Electrical Service
	<input type="checkbox"/>				5	Appliance Repair Service

Figure 6.12

6.1.13 Service_Provider

The Service_Provider relation contains service provider's account information that includes first name, last name, street, city, state, and zip code. Each service provider will be assigned with an ID to be represented throughout the database.

	<input type="button" value="←"/> <input type="button" value="→"/>	<input type="button" value="▼"/>	user_id	first_name	last_name	street	city	state
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11	Warren	John	67 Comanche Park	San Benito Texas
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12	Patten	Stublings	00 Homewood Way	South Padre Texas
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	13	Bella	Goodby	705 Sunbrook Junction	South Padre Texas
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	14	Del	Strathe	37573 Vera Park	Harlingen Texas
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	15	Akim	Walliker	0 Main Pass	Brownsville Texas
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	16	Felipe	Chillistone	82638 La Follette Crossing	Brownsville Texas
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	17	Britni	Archbould	036 Northview Avenue	Edinburg Texas

Figure 6.13

6.1.14 Service_Reserv

The Service_Reserv relation contains the service provider, customer, service type, and date of a service reservation. Each reservation date will be matched to its corresponding service provider's ID, customer ID, service ID, and reservation ID.

	<input type="button" value="←"/> <input type="button" value="→"/>	<input type="button" value="▼"/>	reservation_id	service_id	provider_id	customer_id	DATE
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	1	11	1 2022-05-24
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	1	12	9 2022-04-28
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	2	13	2 2022-05-15
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4	5	19	7 2022-06-01
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5	3	14	3 2022-05-16
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	6	2	13	2 2022-08-05
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	10	1	12	1 2022-05-25
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11	1	12	1 2022-05-07
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12	1	18	1 2022-05-27
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	16	4	16	32 2022-05-12
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	17	1	18	32 2022-05-07

Figure 6.14

6.1.15 Service_Review

The Service_Review relation contains the service provider's customer review. Each review will be matched to its corresponding customer ID, service provider ID, and review ID.

	<input type="button" value="←"/> <input type="button" value="→"/>	<input type="button" value="▼"/>	review_id	provider_id	customer_id	transaction_id	review
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	11	1	4149 Beware! This person is very deceiving. Ended up pa...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	12	9	1521 Did an excellent job! Very friendly and was on tim...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	13	2	4825 Great experience! Recommend it 100%!
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4	19	7	2425 I will rate the service a 3/5. Not what I expect i...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5	14	3	123 Extremely knowledgeable. My issue was solved in a t...

Figure 6.15

6.1.16 Transaction

The Transaction relation contains the cost of a service transaction. Each transaction ID will be matched to the corresponding cost of the transaction..

			transaction_id	transaction_cost	
<input type="checkbox"/>	 Edit	 Copy	 Delete	123	60.00
<input type="checkbox"/>	 Edit	 Copy	 Delete	1521	150.00
<input type="checkbox"/>	 Edit	 Copy	 Delete	2425	80.00
<input type="checkbox"/>	 Edit	 Copy	 Delete	4149	120.00
<input type="checkbox"/>	 Edit	 Copy	 Delete	4825	75.00

Figure 6.16

6.2 Server-side Implementation

6.2.1 Homepage

The homepage for HomeNeedsServices.com welcomes the user at the landing page which users can go back to at any time by clicking the logo on the left hand corner.

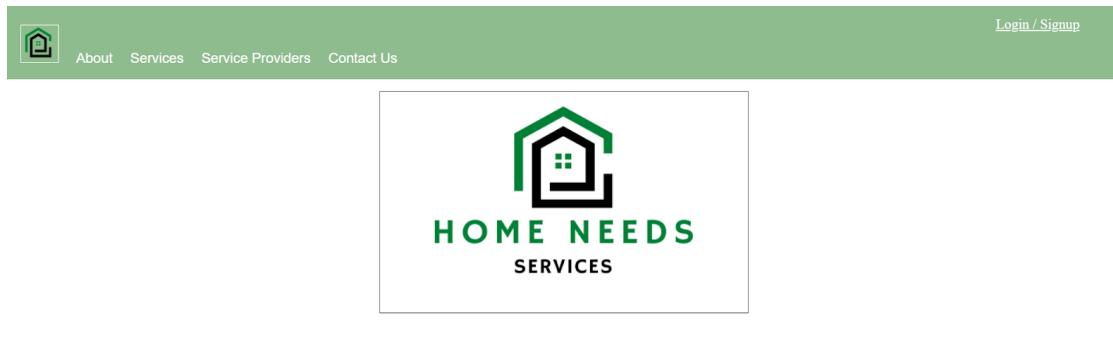


Figure 6.17

Next, we have a list of services available based on the service providers signed up.

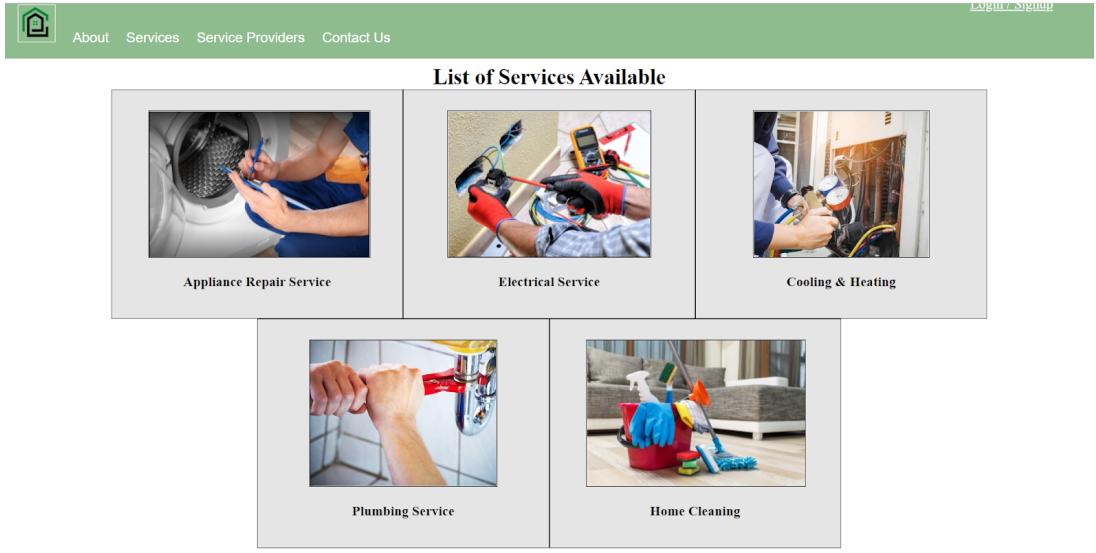


Figure 6.18

On the service providers page, we show the names, service and reviews each of the servicers have on their profiles. The customers looking to hire a service must log in in order to get more information about the availabilities and distance which they can do by clicking on the top right corner to either sign up or log in.

The screenshot shows a green header bar with a house icon and links for 'About', 'Services', 'Service Providers', and 'Contact Us'. In the top right corner, there is a 'Login / Signup' button. Below the header, the title 'List of Service Providers:' is centered, followed by the sub-instruction '(login for more information)'. A grid of service provider profiles is displayed:

Name: Warren John 	Name: Patten Stubbings 	Name: Bella Goodby 	Name: Sarina Kirkwood
Services: Plumbing Reviews: Beware! This person is very deceiving. Ended up paying more than the hourly wage that appears in the website	Services: Plumbing Reviews: Did an excellent job! Very friendly and was on time!	Services: Home Cleaning Reviews: Great experience! Recommend it 100%!	Services: Appliance Repair Service Reviews: I will rate the service a 3/5. Not what I expect it
Name: Del Strathe 	Name: Akim Walliker 	Name: Gigi Wilcockes 	Name: Fowler Britzius

Figure 6.19

By clicking on login/signup, the users would then be redirected to a separate page where they can login using their credentials, or create a new account by following the prompts. All of the coding information for the homepage can be found on the Github link on the Appendix, along

with the style.css file for the styling elements to make the page user friendly and javascript file script.js for visualization and table generation based on the database for the Service provider tab.

6.2.2 Login/Signup Page

To create a new account, we ask users to create a username and provide their email addresses, full name and residential address to be able to match their location to the service providers close to them. Lastly, they are required to create a password which is then encrypted. This information is then sent to the database for the account creation which is stored to be accessed at the next login time if the information provided matches the requirements and there are no errors. The code for this part can be found on our Github as Signup.php.

The screenshot shows a web-based sign-up form. At the top, there are two buttons: 'Login' and 'Sign up'. The 'Sign up' button is highlighted with a green background. Below the buttons, the title 'Sign-up' is centered. The form consists of several input fields arranged vertically:

- Username:** A text input field containing the placeholder 'Username'.
- Email:** A text input field containing the placeholder 'example@example.com'.
- First Name:** A text input field.
- Last Name:** A text input field.
- Street:** A text input field.
- City:** A text input field.
- State:** A text input field.
- Zip:** A text input field.
- Enter Password:** A text input field containing the placeholder 'Password'.
- Confirm Password:** A text input field containing the placeholder 'Password'.

At the bottom right of the form area, there is a 'Signup' button.

Figure 6.20

For login, the users would input their previously created username and password set which is sent to the database to confirm they are active users and redirect them to their respective homepage. If the user cannot log in, they will remain on the login page. The code for this implementation is found as Login.php in the Appendix and the loginStyle.css file for styling elements for the user on the Github.

Figure 6.21

We have implemented some warning measures for incorrect login information, incorrect signup attempts for when a user has already been registered and for when the passwords created do not match.

Lastly, the users have the ability to log out of their sessions by clicking on the logout button and this ends their session and returns them to the homepage.

6.2.3 Service Provider Page

Once a user has logged in, they will be presented with the service provider home page. Within the page, the service provider will be able to view the scheduled services with its customer. The schedules will also include time information. Additionally, the service provider has the opportunity to select their own preferences which include selecting “Inside City Limits” and “Times.” Lastly, the page will include the option to log out of their account. The code for the page is located as servHome.php. We also implement servHomepg.css that provides the user friendly UI. We implement script.js to control the homepage to hide and show the reference. We then implement the makeTable.js shown in the Appendix to make a table for service provider times. All of these code files can be found on Github, linked below.

When discussing how the service provider page connects with the server database we see the following. To get a specific service provider’s preferences we use getPref.php. To update those service provider’s preferences we implement updPref.php. The implementation also provides a message if the update has failed. Additionally, we implement resTime.php to fetch reservations that have been scheduled for the service provider from our database to display.

6.2.4 Customer Page

Once a user has logged in, they will be presented with the customer home page. Within the customer homepage, the available services and service providers will be displayed. This information is also accompanied by the available times that a service provider is open for reservation. The page will also display the time and information that a customer has scheduled with a service provider. The customer is also provided with the option to cancel the service, which is shown on file cancelRes.php in the Appendix. The code for the page is located as cutHome.php. We also implement cusHomepg.css which provides the user-friendly UI. We implement script.js to control the homepage to hide and show the reference. We then implement the makeTable.js to make a table for service provider times.

When discussing how the customer page connects with the server database we see the following. The five services that are being provided are appliance repair, cooling and heating, electrical service, home cleaning, and plumbing. To get the list of service provider available for each service we use the following code files respectively; getServAppRepair.php, getServCoolHeat.php, getServElectr.php, getServHomeClean.php, and getServPlumbing.php. When fetching the information, we will also get the service provider's name and review. Additionally, we use reservation.php to get the reservations from our database when a user selects a service provider, date and time. We also use resTimes.php to fetch and display the scheduled reservations for a client from our database. To get the dates that are available for a client we use timeStamps.php, all files are available on Github.

7. USABILITY

7.1 Homepage

To make the homepage easy for the users to navigate, we incorporated a button with our logo on the left hand corner which can be used to return to the homepage after going through some of the sections we created.

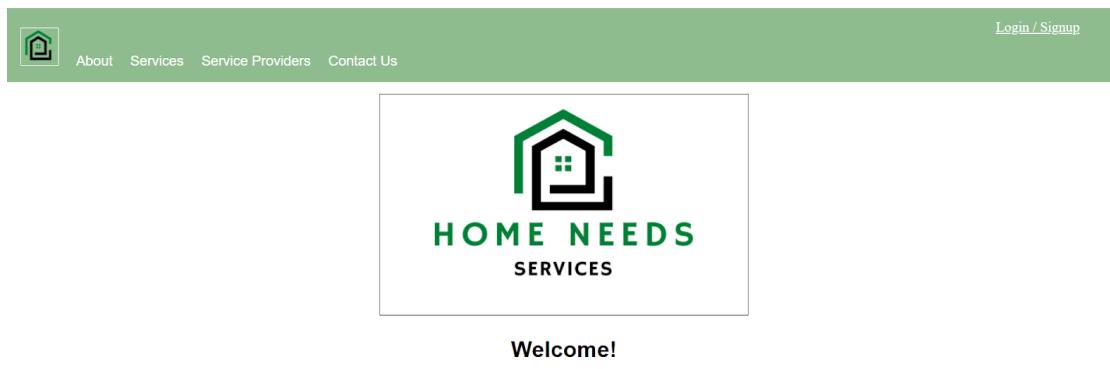


Figure 7.1

We made sure we had a clickable menu at the top of the page where we could have information about the website on the About info page, as well as the services and service providers available overall.



Figure 7.2

We also have a contact page on the same menu which allows the users to get customer service, for the purpose of this project, we have used a fake email and phone number.



Contact Us

If you have any issues or concerns, don't hesitate to contact us. We are here to help
customerservice@HomeNeedsServices.com
1-800-123-4567

Figure 7.3

The Login/Signup button is setup on the top right corner and clicking on it redirects users to a new page.

7.2 Login/Signup Page

For the login and signup page, we implemented the same color scheme used on the main homepage to add continuity. The new account creation was made straightforward, asking only the necessary information needed. The password generation is encrypted to ensure safety to the users. After the creation of a new account, users are redirected directly to their homepage without the need to login on the same page.

For returning users, the process is straightforward by just asking for their previously made credentials. If the user cannot log in, they will remain on the login page and receive the

corresponding error message. To logout, users can end their sessions by clicking on the logout button and this returns them to the main homepage.

7.3 Service Provider Page

We created a separate homepage for service providers so they could also have a place to manage their accounts. This page contains their scheduled services with its customer and time information. The service provider also has the ability to select their own preferences which include selecting “Inside City Limits” and “Times.”

7.4 Customer Page

Similar to the service providers, customers looking for a service also have their own homepage. Once the user has logged in they can see the available services and service providers based on their area and time availability. On this homepage, they see all services scheduled as well as the cancel button to give them the choice to cancel their appointments.

8. TESTING

8.1 Arrive at Home Page

The Home Services home page will provide a guest with information pertaining to the system. The user is able to view an About summary of Home services as well as the contact information. As an overview to the system, a user is also able to view services and service providers available. In order to schedule a service, a user must signup or login.

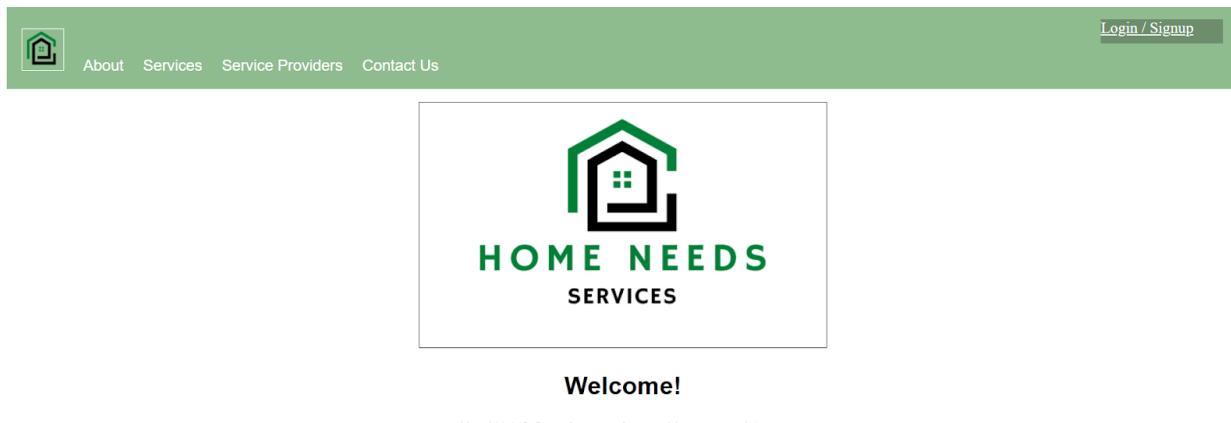


Figure 8.1

8.2 New User Sign Up

As a new user, you will first need to sign up. They may do so by selecting the “Login/Signup” option on the home page.

The image shows a user interface for a login page. At the top, there are two tabs: "Login" and "Sign up". The "Login" tab is highlighted with a green background. Below the tabs, the word "Login" is centered in bold black font. There are two input fields: the first is labeled "Username:" and contains the placeholder text "Username"; the second is labeled "Enter password:" and contains the placeholder text "Password". In the bottom right corner of the main area, there is a "Login" button.

Figure 8.2

The login option will be presented first, the user must select the “Sign Up” tab to continue registering. You must enter a username, email, first and last name, Address, and password.

The image shows a user interface for a sign-up page. At the top, there are two tabs: "Login" and "Sign up". The "Sign up" tab is highlighted with a green background. Below the tabs, the word "Sign-up" is centered in bold black font. There are several input fields: "Username:" with placeholder "JaneDoe"; "Email:" with placeholder "janedoe@test.com"; "First Name:" with placeholder "Jane"; "Last Name:" with placeholder "Doe"; "Address Line 1:" with placeholder "123 Main St." and "Address Line 2:" with placeholder "Edinburg"; "State/Zip:" with two fields: "Texas" and "78541"; "Enter Password:" with placeholder "*****"; and "Confirm Password:" with placeholder "*****". In the bottom right corner, there is a "Signup" button.

Figure 8.3

If the user does not input one of the designated input requirements or the passwords do not match, an error message will appear.

8.3 Customer Home Page

If a user successfully signs up, they will be redirected to the customer home page. Within the customer home page, they will receive a welcome message and be presented with their current

schedule times. As a new user, this area will be noted as “No Times Scheduled.” The user will also be able to sign out and view the services and service providers available.

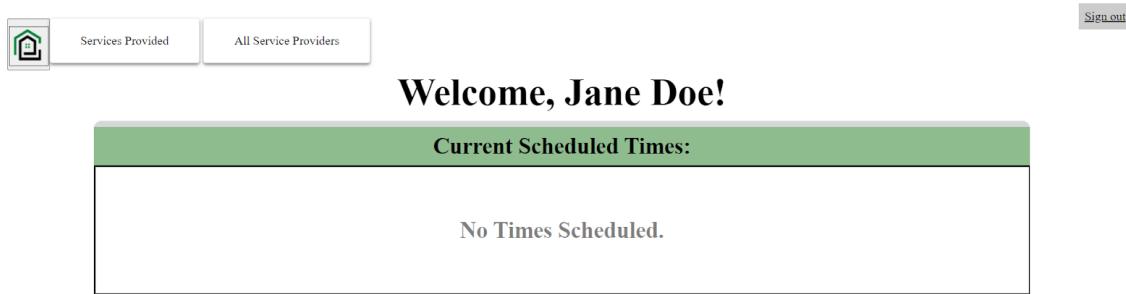


Figure 8.4

8.4 View Services By Filtering>Selecting Service

In continuation with the customer home page, they are able to hover over the “Services Provided” list to view services available.

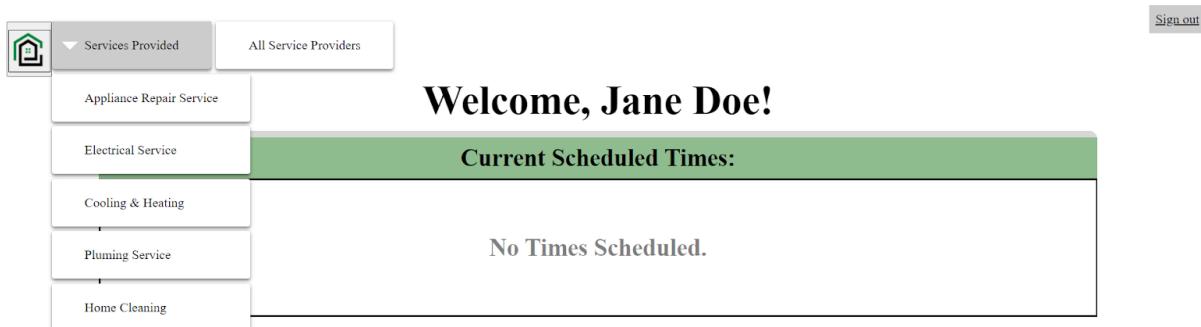


Figure 8.5

After selecting a specific service, the service provider list will only display the service providers that provide the selected service. For each service provider an image, service provider name, service name, and a review will be displayed.

The screenshot shows a user interface for a service provider search. At the top, there is a navigation bar with icons for home, services provided, all service providers, and sign out. The main heading is "Welcome, Jane Doe!". Below this, a section titled "Service: Home Cleaning" displays two service providers:

- Name:** Bella Goodby
- Services:** Home Cleaning
- Reviews:** Great experience! Recommend it 100%!

- Name:** Fowler Britzus
- Services:** Home Cleaning
- Reviews:** No Review Available.

Below this section is a header for "Current Scheduled Times:".

Figure 8.6

8.5 View All Service Providers

If a client would like to view all available service providers, they may click the “All Service Providers” tab. As noted, this will present all the service providers without a service filter.

The screenshot shows a user interface for viewing all service providers. At the top, there is a navigation bar with icons for home, services provided, all service providers, and sign out. The main heading is "Welcome, Jane Doe!". Below this, a section titled "All Service Providers Available" displays several service providers:

Name	Services	Reviews
Bella Goodby	Home Cleaning	Great experience! Recommend it 100%!
Sarina Kirkwood	Appliance Repair Service	I will rate the service a 3.5. Not what I expect it
Del Strathe	Cooling/Heating	Extremely knowledgeable. My issue was solved in a timely manner. Thank you so much!
Gigi Wilcockes	Plumbing	No Review Available.
Fowler Britzus	Home Cleaning	No Review Available.
Felipe Chillstone	Electrical Service	No Review Available.
Britni Archbould	Electrical Service	No Review Available.

Figure 8.7

8.6 View A Service Providers Available Times

To view a service provider's available times, they can click on a service provider's icon.

Available times for: **Bella Goodby**

Service: **Home Cleaning**

Select Time (8:00 AM - 8:00 PM): **--:--**

<input type="radio"/> Saturday 2022-05-07
<input type="radio"/> Sunday 2022-05-08
<input type="radio"/> Monday 2022-05-09
<input type="radio"/> Tuesday 2022-05-10
<input type="radio"/> Wednesday 2022-05-11
<input type="radio"/> Thursday 2022-05-12
<input type="radio"/> Friday 2022-05-13

Current Scheduled Times:

Figure 8.8

8.7 Schedule A service

In continuation with the previous view, the client may select the time and date desired to schedule an appointment by clicking on the circle for the time and selecting “Reserve Time.” If they no longer would like to schedule an appointment with the service provider, they may select “Cancel” to go back to the service provider table.

Available times for: **Bella Goodby**

Service: **Home Cleaning**

Select Time (8:00 AM - 8:00 PM): **10:00**

<input type="radio"/> Saturday 2022-05-07	20	01
<input type="radio"/> Sunday 2022-05-08	21	02
<input type="radio"/> Monday 2022-05-09	22	03
<input type="radio"/> Tuesday 2022-05-10	23	04
<input checked="" type="radio"/> Wednesday 2022-05-11	00	05
<input type="radio"/> Thursday 2022-05-12	01	06
<input type="radio"/> Friday 2022-05-13		

Current Scheduled Times:

Figure 8.9

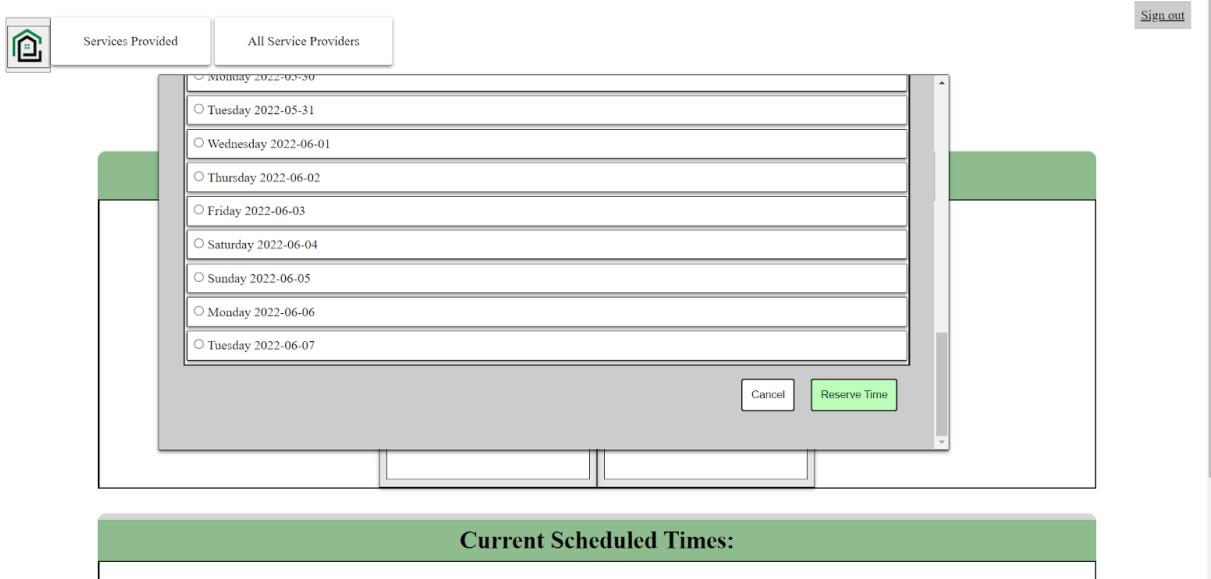


Figure 8.10

8.8 Show Scheduled Times

Once the user schedules a service, they will be redirected to their homepage where their scheduled times are displayed



Figure 8.11

Users can schedule as many services as they need and they will all show on their homepage.

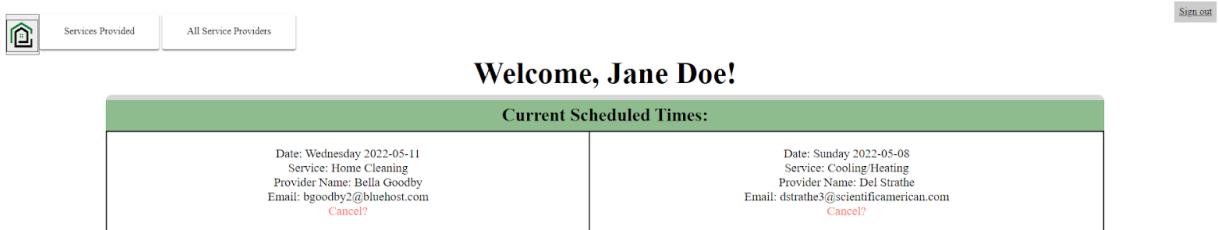


Figure 8.12

8.9 Cancel A Schedule Service

Under the “Current Scheduled Times” box, a user has the ability to cancel their appointment.

Current Scheduled Times:	
Date: Wednesday 2022-05-11 Service: Home Cleaning Provider Name: Bella Goodby Email: bgoodby2@bluehost.com Cancel?	Date: Sunday 2022-05-08 Service: Cooling/Heating Provider Name: Del Strathe Email: dstrathe3@scientificamerican.com Cancel?
<input type="checkbox"/> Cancel Reservation Update	

Figure 8.13

8.10 User Sign Out

If a user would like to sign out, they may select the “Sign Out” button on the top right hand corner.

Current Scheduled Times:	
Date: Wednesday 2022-05-11 Service: Home Cleaning Provider Name: Bella Goodby Email: bgoodby2@bluehost.com Cancel?	

Figure 8.14

If successful, they will be redirected to the home page.

Home Needs Services

Welcome!

Need Help? One of our service providers can assist you.

Figure 8.15

8.11 User Log In

If a user would like to log back into their account, they will need to select the “Login/Signup” option in the right hand top corner. The user will be presented with the “log in” tab. The user must enter all requirements, if not, they will be presented with a small error message.

The screenshot shows a login form with a green header containing 'Login' and 'Sign up' buttons. The main area is titled 'Login'. It has two input fields: 'Username:' with 'JaneDoe' entered and 'Enter password:' with '*****' entered. A 'Login' button is at the bottom right. The entire form is contained within a light gray box.

Figure 8.16

If they enter the incorrect username or password, they will be presented with an error message.

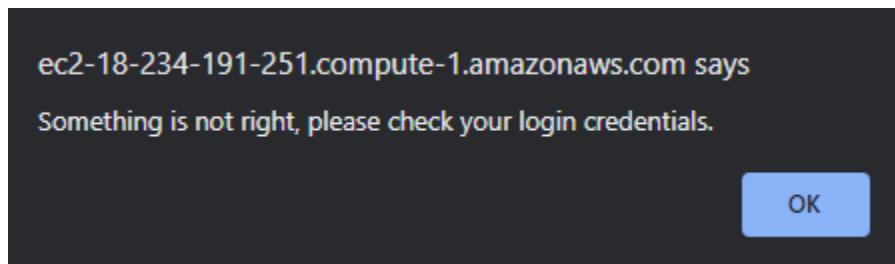


Figure 8.17

If they enter the correct information, they will be directed to the client home page.

8.12 Service Provider

If A service provider logs in, they are directed to their own Homepage. Whether a Customer or a Service provider is logging in is determined based on the user's level in the database. In the provider's homepage, the service provider is first presented with a welcome message and the list of “Current Scheduled Times”.

Welcome, Warren John !

 Sign out

Current Scheduled Times:			
	Date: Tuesday 2022-05-24 Service: Plumbing Customer Name: Alexis Launder Customer Email: alauder0@blinklist.com Address: 860 Miller Way City: South Padre Cancel?	Date: Monday 2022-05-09 Service: Plumbing Customer Name: Christian Narcia Customer Email: cinarcia1@gmail.com Address: 123 street City: Brownsville Cancel?	

Figure 8.18

Here, the service provider has the option to cancel any current scheduled times by clicking “Cancel” and then confirming that they want to cancel as shown below.

Welcome, Warren John !

 Sign out

Current Scheduled Times:			
	Date: Tuesday 2022-05-24 Service: Plumbing Customer Name: Alexis Launder Customer Email: alauder0@blinklist.com Address: 860 Miller Way City: South Padre Cancel?	Date: Monday 2022-05-09 Service: Plumbing Customer Name: Christian Narcia Customer Email: cinarcia1@gmail.com Address: 123 street City: Brownsville Cancel?	<input checked="" type="checkbox"/> Cancel Reservation Update

Figure 8.19

Once the box is checked and the user clicks “Update”, the reservation will be canceled.

On the top right of the homepage, there are two buttons, the gear button opens the pop up to update the preferences for the service provider currently logged in, as shown below, and the Logout button to end the users session, which will send the user to the homepage.

Welcome, Warren John !

 Sign out

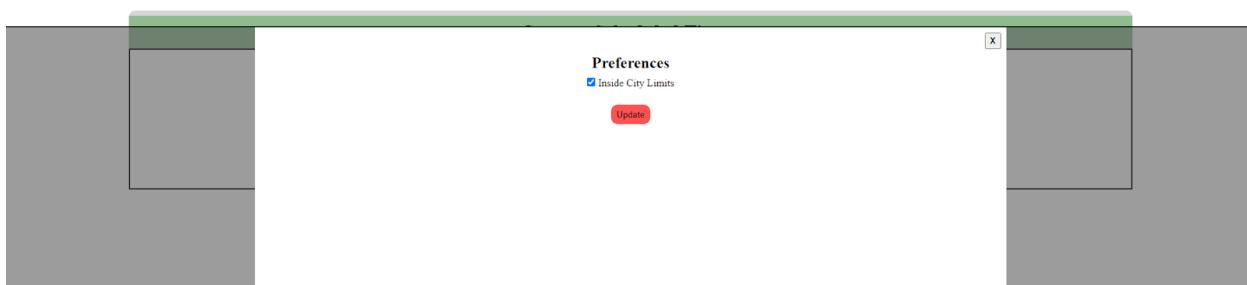


Figure 8.20

Here, if the service provider wishes to only work for clients within the City they live in, they check the box and click “Update”. Once they click update, their services will no longer be offered to clients who don’t live in the same City. The inverse is also true, unchecking the box and clicking “Update” will add the service provider to the result of customers outside the city.

9. CONTRIBUTIONS

Mariana Martinez :

Team Captain

Styling

Homepage

Website Testing

Christian Narcia :

Database Implementation on website end

Homepage Services, Service Provider Tab

Login/Sign up Front end

Customer/Service Provider Home Page (Front end/Back end)

Connected Web Pages

Sergio Soto :

Database Creation and Management

Created SQL Queries to fetch required information for website

AWS Website Hosting

AWS Database Hosting

Maricruz Flores:

Sign up/Login Back end

Website Testing

10. CONCLUSION

To present a platform that provides ease of access to information and resources for local based services and prospective customers we implemented an interactive website named Home Needs Services. As noted, the platform is a marketplace where a service provider may offer its services to prospective customers. Customers have the opportunity to join the platform.

For ease of use, we created a home webpage that provides the user with an introduction to the platform. A user is able to learn about our purpose, the service providers, and their respective service. In order to allow a user to participate in the market, they will have the opportunity to sign up for an account with its unique identification within the system. This unique ID will allow for the interaction and linkage within the service and service provider system. The customer will be provided an interface that is easy to use. There are no excessive pop-ups or page redirections. The customer is able to view the services being provided and its service providers. If needed,

they may filter the service provider list by selecting the service they desire. The user can easily schedule or cancel their service appointment. The simplicity of interacting within the system is to provide an accessible and user friendly platform.

Similarly, a service provider will have their own unique identity within the system that allows it to interact within its environment. A service provider may designate their service, the times and dates they are available, service reviews, and preferences. A provider will also be able to view all their scheduled appointments within a designated area, allowing for the continuous ease of use for all of its users.

All of these components are maintained by the usage of our database. The database maintains client, service provider, and service information. It is all connected and processed by assigning unique identification values for its users and tasks. With the usage of unique IDs, allows for the ease of access of information, the ability to easily update or change information, and connect appropriate information to each other.

We tested the functionality of the website and successfully registered a new user, allowed them to schedule an appointment, cancel the appointment, and navigate through the information provided throughout the website. Through the final test/simulation, we were met with no errors. For future work we could improve upon the automatization of the relations by creating more TRIGGER functions in the database.

11. REFERENCES

1. Tor Y, Chen Z. ANTES System Requirements Specification. 2003.

APPENDIX A

Github Link: https://github.com/xTheworst1x/Database_Website

***Fetches Login information from login form compares to database information
Login.php***

```

login.php
1 <?php
2
3     //CHECK FOR ACTIVE SESSION
4     if(session_status() == PHP_SESSION_NONE){
5         session_start();
6     }
7
8     $dbservername = "ec2-18-234-191-251.compute-1.amazonaws.com";
9     $dbusername = "admin";
10    $dbpassword = "localhost";
11    $dbname = "HomeNeedsService";
12
13    //create connection
14    $conn = new mysqli($dbservername, $dbusername, $dbpassword, $dbname);
15    //check connectin
16    if($conn->connect_error){
17        die("Connection failed: " . $conn->connect_error);
18    }
19
20    $attempt = 0;
21
22    $username = $_POST['username'];
23    $password = $_POST['password'];
24    $password = md5($password);
25
26    $sql = $conn->prepare("SELECT c.first_name as first_name, c.last_name as last_name, c.user_id as userid, l.user_level as userlvl
27    FROM Account_Login as l
28    INNER JOIN Account_Creation as c
29    ON l.username = c.username
30    WHERE l.username = ? AND c.password = ?");
31    $sql->bind_param("ss", $username, $password);
32    $sql->execute();
33
34    //$result = $conn->query($sql);
35    $result = $sql->get_result();
36
37    $result = $result->fetch_array();
38
39    //echo json_encode($result);
40    if ($result) {
41        $_SESSION['name'] = $result["first_name"] . " " . $result["last_name"];
42        $_SESSION['userID'] = $result["userid"];
43        $_SESSION['username'] = $username;
44        $_SESSION['userlvl'] = $result["userlvl"];
45
46        $userlvl = $result["userlvl"];
47
48        if($userlvl == 0){
49            header("location: Customer/custHome.php");
50            exit();
51        }else if($userlvl == 1){
52            header("location: ServiceProvider/servHome.php");
53            exit();
54        }else{
55            header("location: userloginPg.php");
56        }
57    }else{
58        //echo 'Error: ' . mysql_error();
59        $attempt += 1;
60        $_SESSION['attempt'] = $attempt;
61    }
62
63    header("location: userloginPg.php");
64
65 ?>

```

Creates table for current times the service provider is available

MakeTable.js

```

131  /* Fill Service Provider Times available table */
132  function servProvTimeList(myVar){
133      document.cookie = "avaServid =" + myVar;
134      //call ajax
135      var ajax = new XMLHttpRequest();
136      var method = "GET";
137      var url = "timeStamps.php";
138      var asynchronous = true;
139
140      ajax.open(method, url, asynchronous);
141
142      //sending ajax request
143      ajax.send();
144
145      // receiving response from php
146      ajax.onreadystatechange = function()
147      {
148          if(this.readyState == 4 && this.status == 200){
149              data = JSON.parse(this.responseText);
150              //alert(data);
151              createAvailableTable(data);
152          }
153      }
154  }
155

```

timeStamps.php

```

<?php
    //session_start();
    //Creating Servername variable
    $dbservername = "ec2-18-234-191-251.compute-1.amazonaws.com";
    $dbusername = "admin";
    $dbpassword = "localhost";
    $dbname = "HomeNeedsService";

    $servProvName = $_COOKIE['avaServid'];

    //create connection
    $conn = new mysqli( $dbservername, $dbusername, $dbpassword, $dbname);
    //check connectin
    if($conn -> connect_error){
        die("Connection failed:".$conn->connect_error);
    }

    //sql = "SELECT username, email FROM accounts where username != 'Admin'";
    $sql = "SELECT dates_available, DAYNAME(dates_available)
    FROM Service_Provider as p
    INNER JOIN Offers as o
    ON p.user_id = o.provider_id
    INNER JOIN Provider_Availability as a
    ON a.provider_id = o.provider_id
    WHERE a.provider_id = '$servProvName'
    AND a.available = 'Yes'
    AND DATE(dates_available) BETWEEN NOW()- INTERVAL 2 DAY AND (NOW() + INTERVAL 30 DAY);";

    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            $table[][]= $row["DAYNAME(dates_available)"]." ".strVal($row["dates_available"]);
        }
        echo json_encode($table);
    }
}

```

Cancel Reservation for Service Service Provider

cancelRes.php

```
ServiceProvider > cancelRes.php
1  <?php
2  |   if(session_status() == PHP_SESSION_NONE){
3  |       session_start();
4  |
5  |   //session_start();
6  |   $reservation_id = $_GET['resID'];
7  |
8  |   //Creating Servername variable
9  |   $dbservername = "ec2-18-234-191-251.compute-1.amazonaws.com";
10 |   $dbusername = "admin";
11 |   $dbpassword = "localhost";
12 |   $dbname = "HomeNeedsService";
13 |
14 |   //create connection
15 |   $conn = new mysqli($dbservername, $dbusername, $dbpassword, $dbname);
16 |   //check connectin
17 |   if($conn -> connect_error){
18 |       die("Connection failed:". $conn->connect_error);
19 |   }
20 |
21 |   try{
22 |       $sql = "DELETE FROM Service_Reserv WHERE Service_Reserv.reservation_id = '$reservation_id'";
23 |       $result = $conn->query($sql);
24 |   }catch(Exception $err){
25 |       echo "<script>alert($err)</script>";
26 |   }
27 |   header("location: ./servHome.php");
28 ?>
```