Helpers on Hand

System Requirements Documentation

by: Jennifer Rose

Table of Contents

Customer Problem Statement

Individuals may find themselves in need of assistance completing domestic chores in or around their homes, such as changing light bulbs, hanging pictures, hooking up new electronic equipment, etc. Meanwhile, able-bodied individuals with free time on their hands and a need to stay busy sometimes find themselves looking for things to do. This system will connect the two. A system manager will track available volunteer helpers and pair them with chore requests from clients.

Glossary of terms

- Availability calendar – an interactive calendar displaying the days and times helpers are available for specific chores
- Chore – a household task helpers can complete for clients upon request; usually a task requiring manual labor or experience navigating complex systems such as calling insurance companies or health care offices
- Chore request – a ticket that is created when a client logs in to the system and selects a chore they need completed by a helper
- Chore chart – a dynamic report that shows which helpers are assigned to perform chores for clients including the day and time they are assigned
- Client – a user of the system that submits requests for helpers to complete chores for them
- Helper – a user of the system that volunteers to complete chores for clients
- Manager – a user of the system that coordinates chore assignments and maintains system code tables
- Transaction id – a unique identifying number issued when a chore assignment is made connecting a client, chore, and helper(s)

System Requirements

Functional Requirements

No.	Priority Weight	Description
FR-001	High	Helpers should be able to choose multiple values from a list of chores they can complete
FR-002	High	Clients should be able to choose multiple values from a list of chores they need completed
FR-003	High	Helpers should be able to select multiple days and multiple times in a day they are available to perform the chores they are able to complete

No.	Priority Weight	Description
FR-004	High	When helpers choose days and times they can perform certain chores, those days and times should show in the calendar for clients to select from
FR-005	Low	Each chore should have a color or icon assigned to it
FR-006	Medium	Users in the system should be able to hold more than one role
FR-007	Medium	System should display a chore chart dynamically
FR-008	High	Users should create a unique username with a password
FR-009	High	Managers should be able to see all the users and chores
FR-010	High	Managers should be able to edit all the users and chores
FR-011	High	The manager should be able to create tasks to assign to helpers
FR-012	Medium	The clients should be able to see the status of their chore requests
NR-001	Low	Website color scheme should include tones that are proven to elicit calm vibes
NR-002	High	The forms on the website should be very simplistic with basic fields and very little direction needed
UR-001	High	Radio buttons should appear when the login button is clicked allowing the user to choose which role they are logging into
UR-002	Low	Helper and Client users should see a list of their past chore tickets

Nonfunctional Requirements

- Functionality: The system is not intended for large scale usage. It is designed for use in small communities such as neighborhoods, apartment complexes, extended families, non-profit organizations, etc. It requires login ids/usernames and passwords.
- Usability: The user interface has pleasing tones and simple fields.
- Reliability: This is a web-based system which will be as reliable as the user's internet connection.
- Performance: This is a web-based system which will perform as quickly as the user's internet connection allows.
- Supportability: The system is supportable as long as the administrator agrees to run it.

Stakeholders

Helpers on Hand is a program that will appeal to small groups and communities such as homeowners' associations and retirement neighborhoods. The users of the system will be folks that need a little help here and there with small chores around the home. Other users will be the helpers that volunteer to do the chores. Some folks will be helpers and clients. For instance, Damon may volunteer to help someone with their laundry, then ask for help from someone else with his television antennae. People will benefit from this program by sharing the chores with others without feeling burdensome by asking family and friends for help when they have perfectly capable and eager neighbors nearby ready and willing. The helper role will typically be filled by someone that enjoys doing simple tasks, staying busy, feeling accomplished, and/or helping others. The manager of the system will likely be on the board of the association or the property manager or another well-meaning individual looking for ways to give back to their community.

Actors and Goals

Primary Actors

- Client: This actor can select chores they need completed and choose days and times from an availability calendar.
- Helper: This actor can select chores that they can perform on specific days and times.
- Manager: This actor can add, remove, or update *chores, helpers, clients,* and *transactions.*

Secondary Actors

- Chore: Chores are the item *helpers* can request completed and *clients* can complete.
- Chore chart: This is a report that can be pulled by a *manager*. The displayed details and criteria of the report may be altered or filtered by a *manager*.
- Chore request: This item is created by a *client* when they submit a request for a *chore* to be completed for them on a specific day and time.
- Request queue: This is where the *chore requests* accumulate until a *manager* assigns a *chore* to a *helper*.
- System: This is the functional database where *chores* are defined and information about *helpers* and *clients* are held. It is a tool the *manager* uses to pair *helpers* with *clients* and *chores*. It contains the details of all *chore requests* and *chores completed*.
- Transaction: This ticket is created by a *manager* and corresponds to a *client chore request*. It assigns a *helper* to the *chore* and sets the day and time for completion

Use Cases

Manager (total: 16)

- Access database: log into SQL database (4)
- Create records, clients: create records for clients (2)
- Create records, helpers: create records for helpers (2)
- Create records, chores: create records for chores (2)
- Assign transactions: review the request queue and assign chores to helpers for clients (2)
- Populate chore chart: create a report that shows which helpers are completing which chores for the clients for a given period of time (2)
- Run reports: write queries to display data (2)

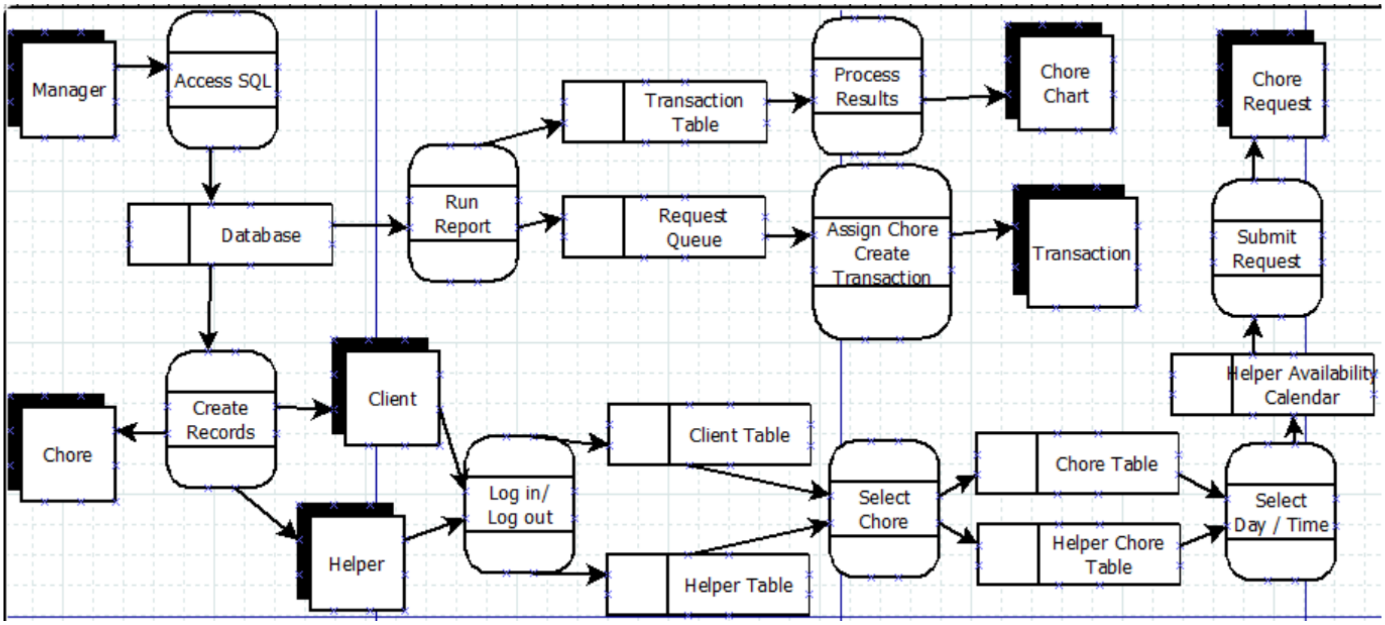
Helper (total: 18)

- Log in/Log out: log into and out of the site (4)
- Choose chore: choose which chore or chores they can complete (2)
- Choose day: choose the day they are available to complete these chores (2)
- Choose time: choose the time they are available to complete these chores (2)
- Submit availability: sends the days and times the helper is available for each chore they can do, so it will appear as available to the clients (6)
- View history: see a list of past transactions completed (2)

Client (total: 16)

- Log in/Log out: log into and out of the site (4)
- View history: see a list of past requests and statuses (2)
- Choose chore: choose which chore or chores they need completed (2)
- Choose day: choose the day they would like to have the chore completed (2)
- Choose time: choose the time they would like to have the chore completed (2)
- Submit request: send a chore request to a queue for the manager to retrieve and assign (4)

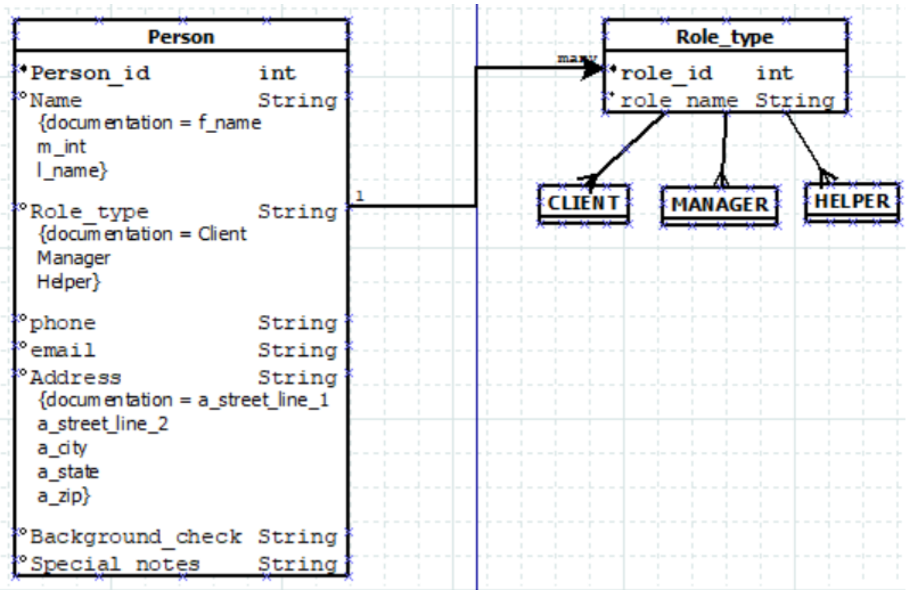
Use Case Diagram



Class Diagram

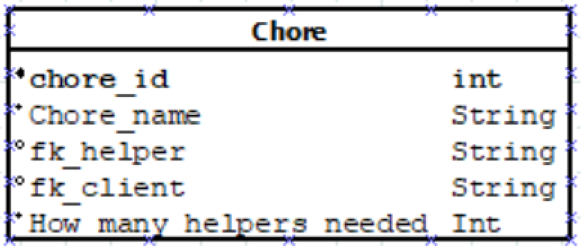
Person and Role

We have a person record which contains contact data including name, email address, phone number, and mailing address. The person will have a person id number which will be used in other areas of the system. A person can hold a role of client, helper, or manager. A person can hold more than one role.



Chore

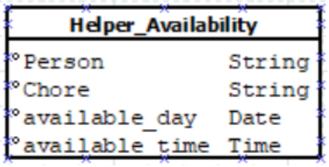
A chore is an object containing details about a domestic task that can be requested by a client and performed by a helper. Some chores may require more than one person to complete, therefore chores have an attribute related to how many helpers are needed to complete it.



Chore	
◆chore_id	int
◆Chore_name	String
○fk_helper	String
○fk_client	String
◆How many helpers needed	Int

Helper Availability

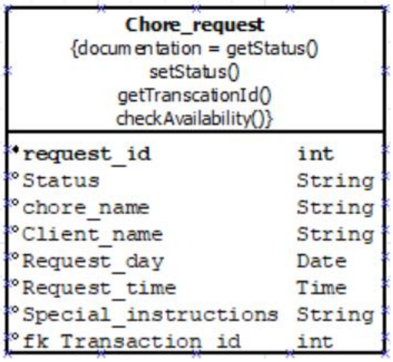
An availability entry is an object created to let the system know the day and time a helper is available for specific chores. It will be used to connect chore requests with transactions.



Helper_Availability	
○Person	String
○Chore	String
○available_day	Date
○available time	Time

Chore Request

A chore request is an object created when a client chooses a day and time they need help with a domestic task. It contains the day and time they would like the chore completed and what type of chore is needed.

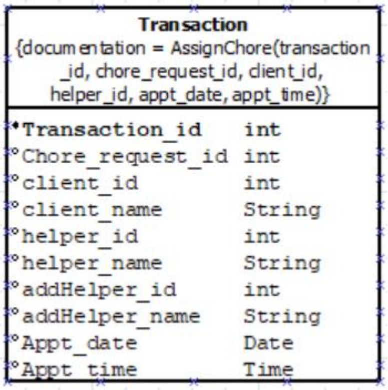


Chore_request
{docum entation = getStatus()
setStatus()
getTranscationId()
checkAvailability()}

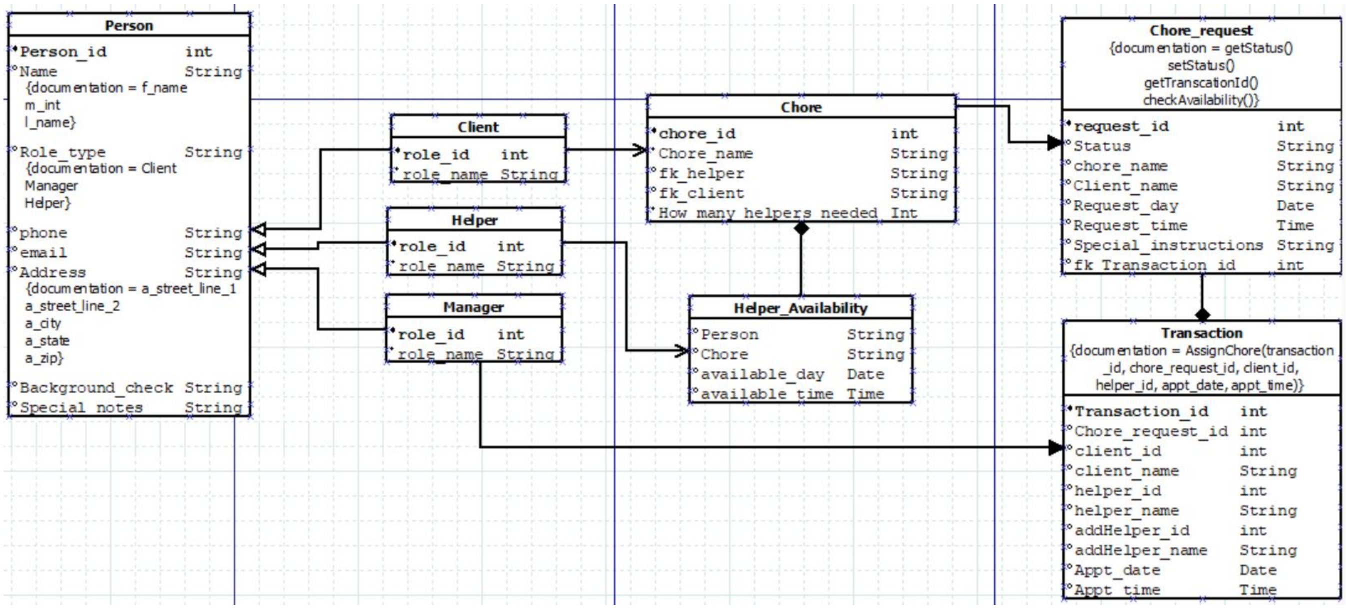
◆request_id	int
○Status	String
○chore_name	String
○Client_name	String
○Request_day	Date
○Request_time	Time
○Special_instructions	String
○fk Transaction id	int

Transaction

A transaction is an object that a manager creates when they review the chore requests and the helper availability to assign a helper to a chore. The Transaction will include the names of the client and all the helpers that will be scheduled to complete the chore, as well as, the day and time the helpers should complete them.

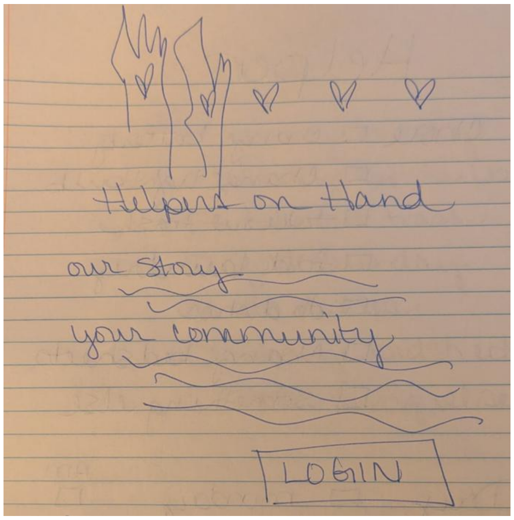


Data Types and Operation Signatures

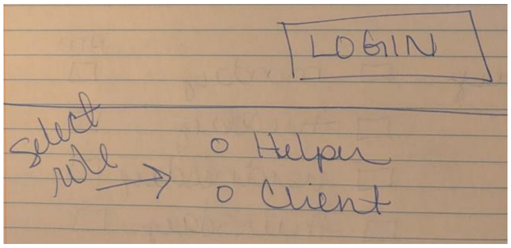


User Interface Requirements

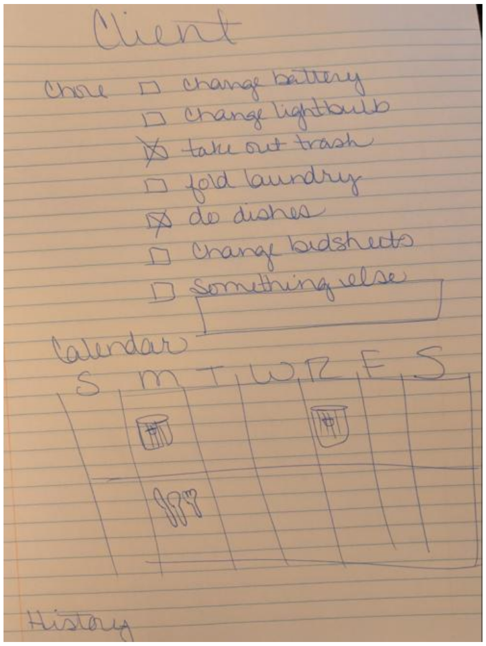
- Home Page



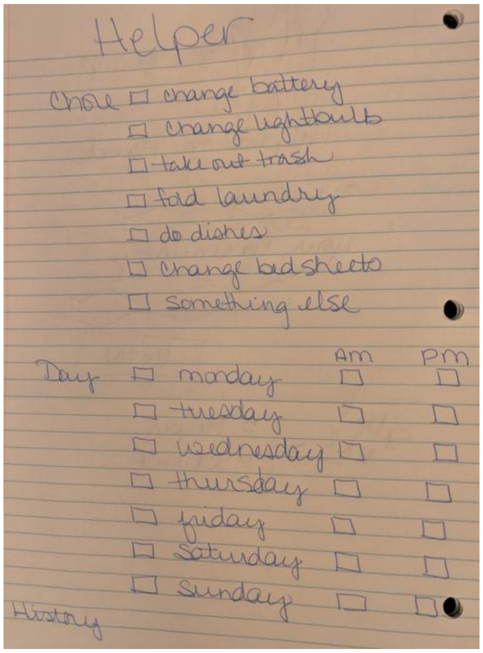
- Login Screen - choice of roles



- Client - Landing page with chore dropdown menu
- Client - Dynamic calendar populated with availability for chores
- Client - List of chore transaction history



- Helper - Landing page with chore dropdown menu
- Helper - Empty calendar for Helper to choose their availability
- Helper - List of completed chore transaction history



Helper

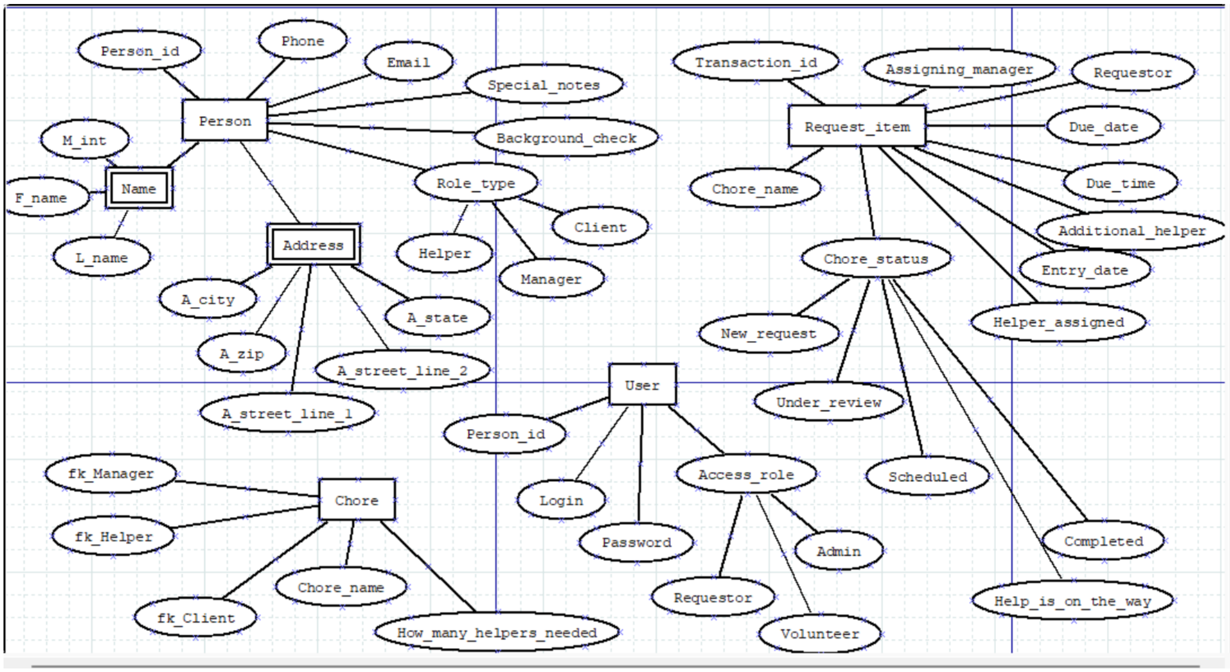
Chore □ change battery
□ change lightbulb
□ take out trash
□ fold laundry
□ do dishes
□ change bed sheets
□ something else

		Am	Pm
Day	□ monday	□	□
	□ tuesday	□	□
	□ wednesday	□	□
	□ thursday	□	□
	□ friday	□	□
	□ Saturday	□	□
	□ Sunday	□	□

History

Plan of Work

Complete



- ✓ Database created using *phpMyAdmin*
- ✓ 7 Tables created
  - ○ Address
  - ○ Chore
  - ○ Chore_status
  - ○ Name
  - ○ Person
  - ○ Request
  - ○ Users
- ✓ Stub web page built using Html and PHP

Helpers on Hand Home Page

Plan of Work

Scheduled

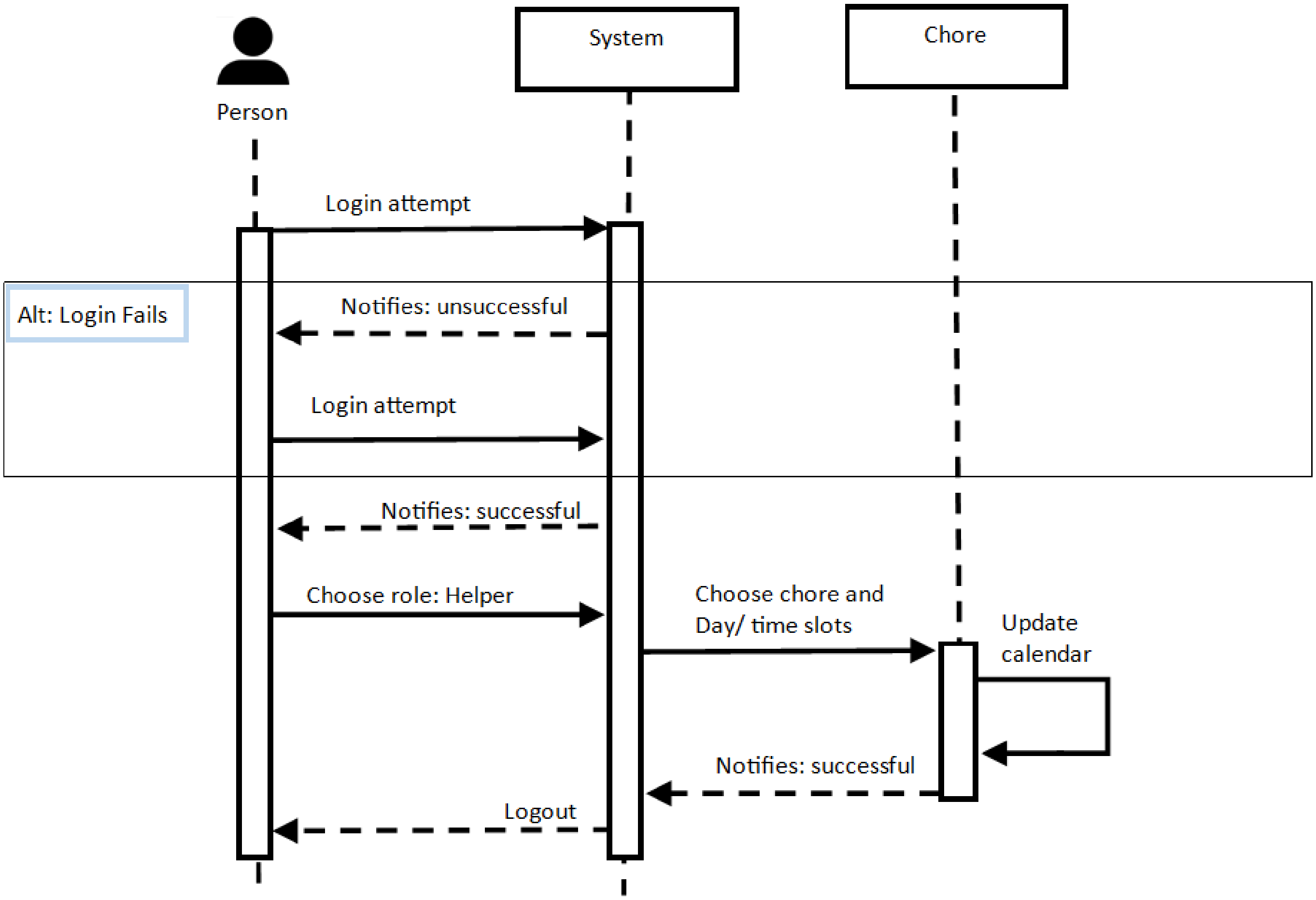
- □ Build test users
- □ Continue building HTML and PHP forms for helpers and clients
- □ Test data flow
- □ Dress up the interface, continue testing
- □ Fine tune and test
- □ Soft launch; record demo for midterm
- □ Allow user testing, make adjustments based on feedback
- □ Make adjustments, continue testing
- □ Testing, Testing, Testing
- □ Testing, Testing, Testing
- □ Fine tune and test
- □ Fine tune and test
- □ Record final presentation

Sequence Diagram 1

Helper Volunteers for Chores
Actor: Helper
Object: Person, Helper, Chore

Steps of the volunteer process:

1. Person logs into the system
2. Person chooses Helper role
3. Helper chooses the types of chores they can perform
4. Helper marks the days and times they are available to do each type of chore
5. Helper submits the availability form

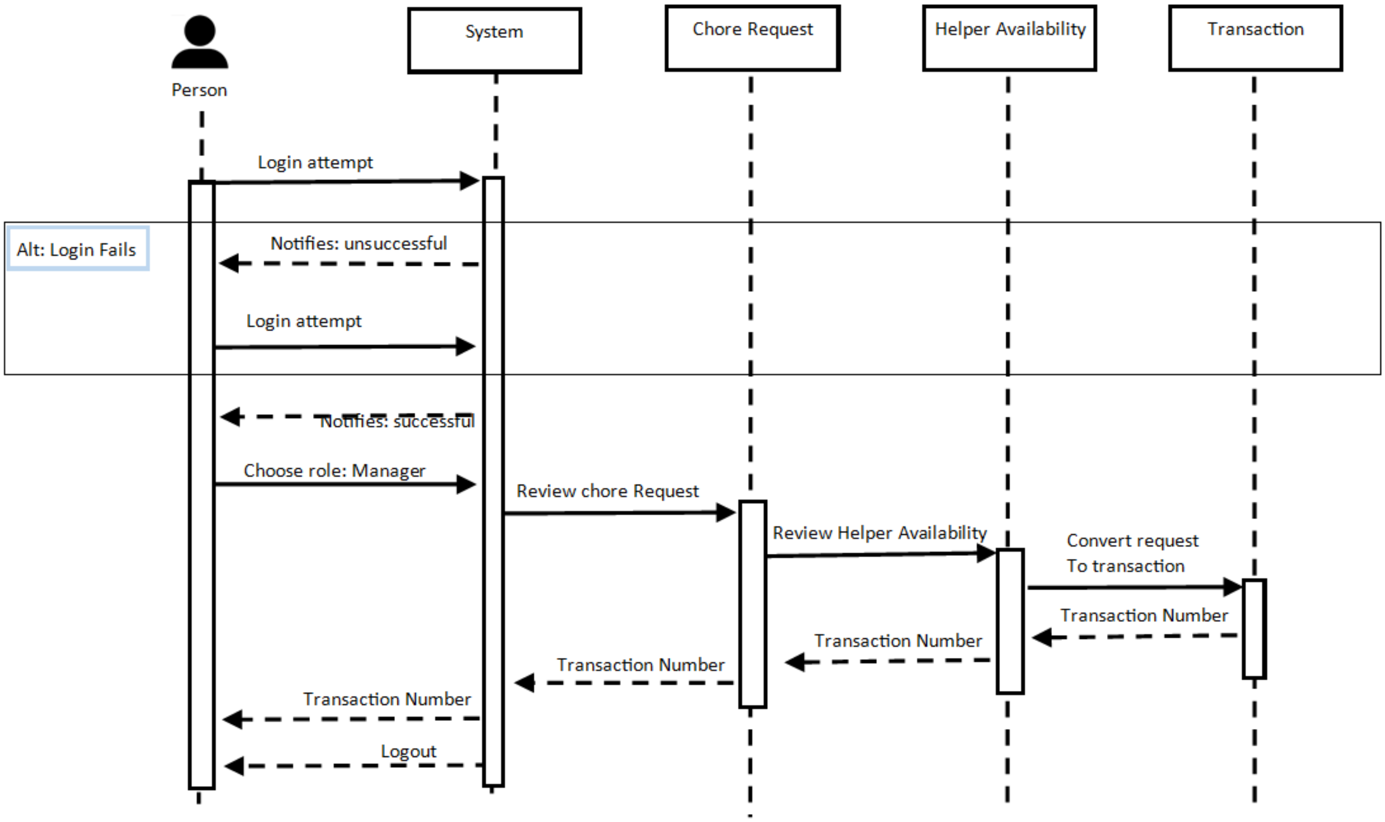


Sequence Diagram 2

Manager Creates Transaction from Chore Request
Actor: Manager
Object: Person, Manager, Chore Request, Helper Availability, Transaction

Steps of the assignment process:

1. Person logs into the system
2. Person chooses Manager role
3. Manager views chore request queue
4. Manager compares requests with Helper availability
5. Manager converts request into transaction and assigns to Helper



Activity Diagram

Client submits a chore request

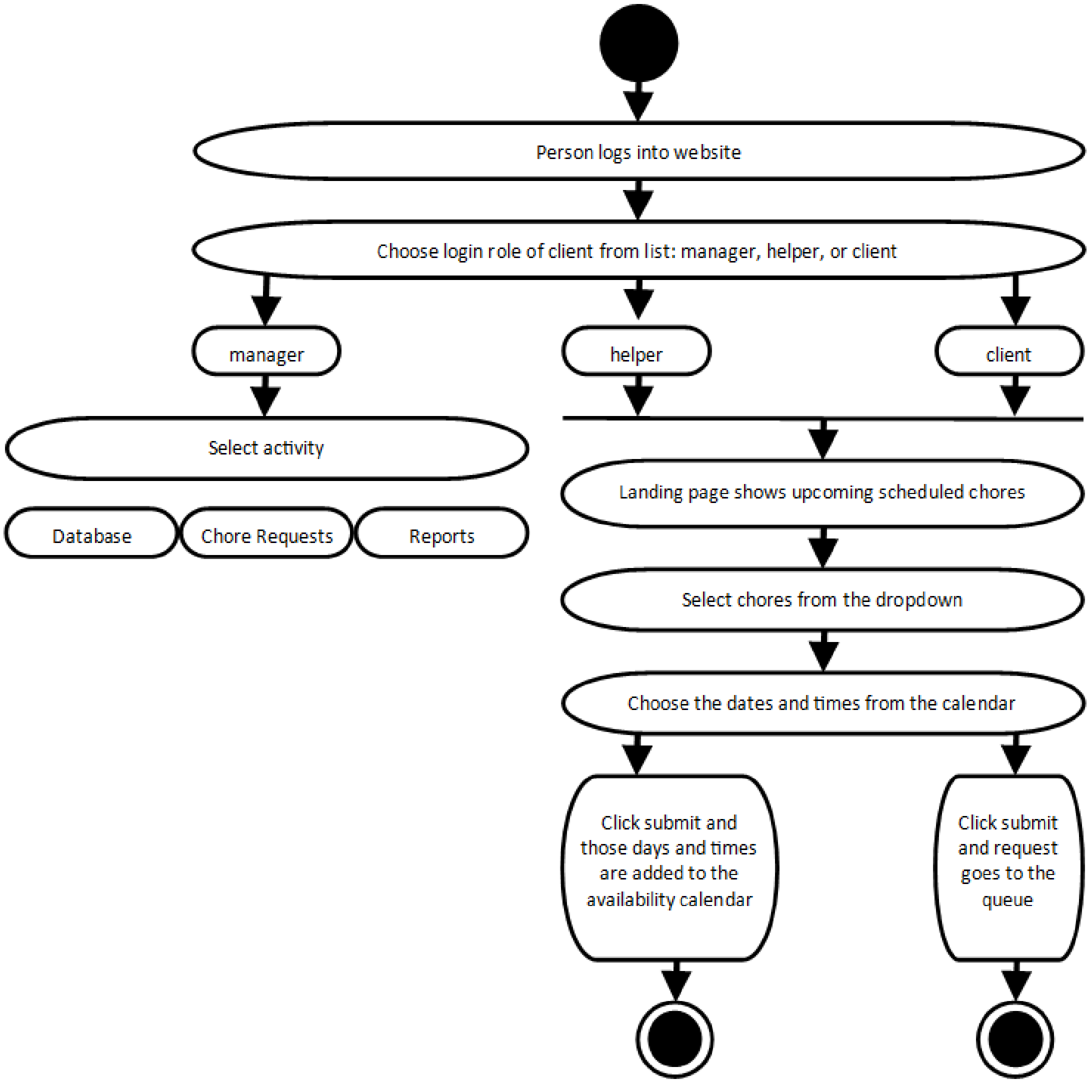
- Initial State: Person logs into the system

- Final State:

1. Client submitted a request for help with one or more chores
2. Client cancelled a previous chore request
3. Client did not submit a request for help

Actions

The person logs into the website as a client. They view their upcoming chores. They initiate a new chore request. Choose the chores they need and select from the days and times shown as available to them.

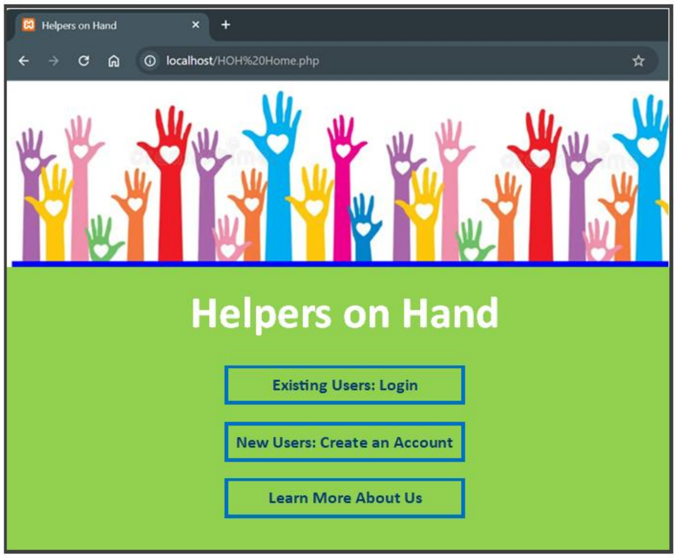


Person logs into website
Choose login role of client from list: manager, helper, or client
manager
helper
client
Select activity
Database
Chore Requests
Reports
Landing page shows upcoming scheduled chores
Select chores from the dropdown
Choose the dates and times from the calendar
Click submit and those days and times are added to the availability calendar
Click submit and request goes to the queue

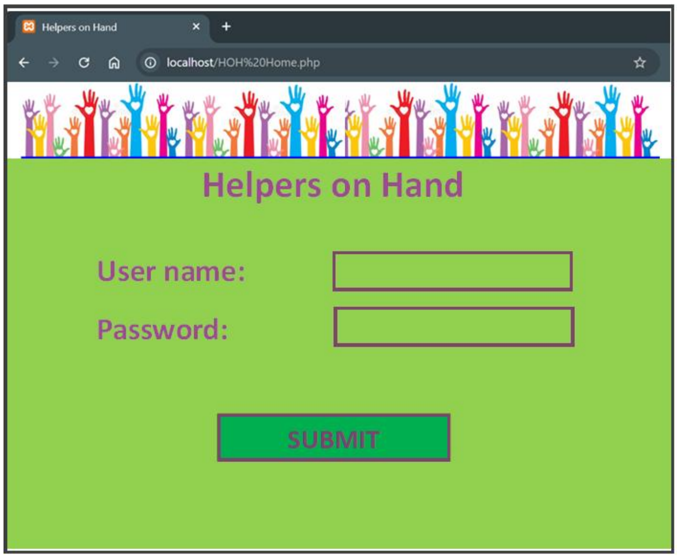
Helpers on Hand System User Interface

Use Case 1: Existing User Requests Task

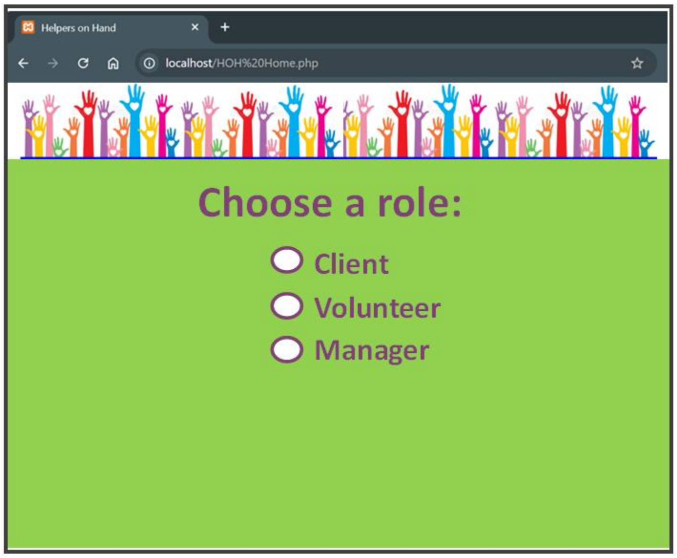
From the landing page, a user clicks the button associated with logging in as an existing user.



Then they will enter their username and password in a form accepting data types of text and password. Then click a submit button



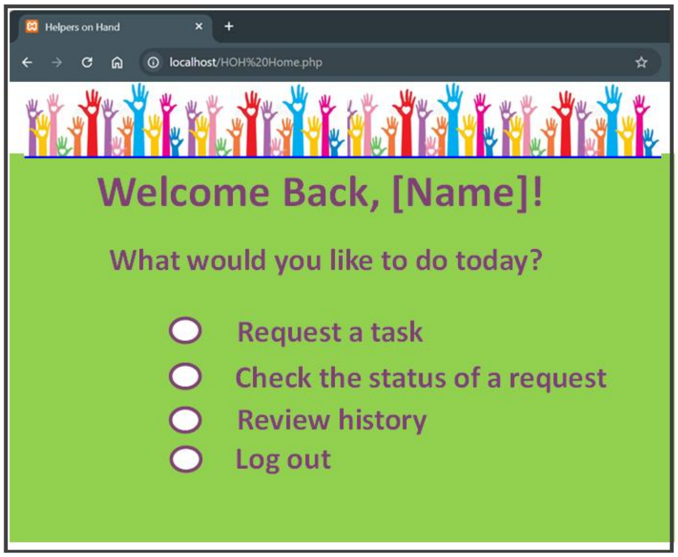
They will choose the role in which they are currently acting. One user may hold more than one role. Clicking the radio button will move the user to the next screen. They will click Client.



Choose a role:

○ Client
○ Volunteer
○ Manager

The user will land on a page welcoming them by name. They will click a radio button to Request a task which will advance them to the next screen.

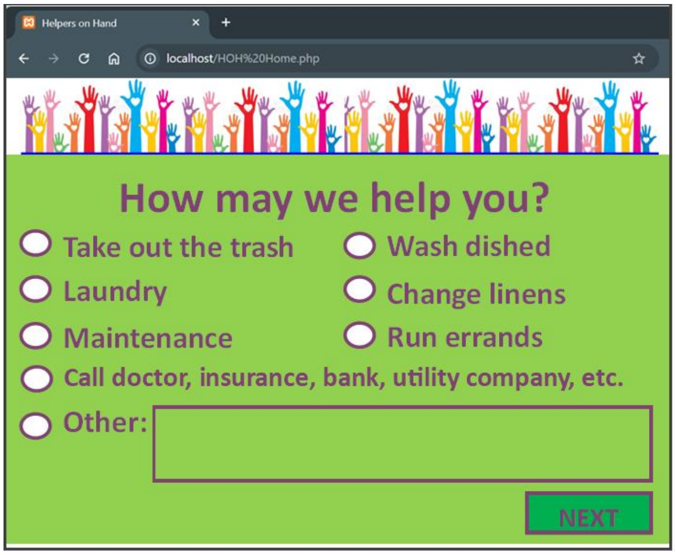


Welcome Back, [Name]!

What would you like to do today?

○ Request a task
○ Check the status of a request
○ Review history
○ Log out

The user may select multiple radio buttons associated with chores. There will be a free text field for clients to request chores that are not listed. Then they will click the NEXT button to move to the next screen.



How may we help you?

○ Take out the trash ○ Wash dished

○ Laundry ○ Change linens

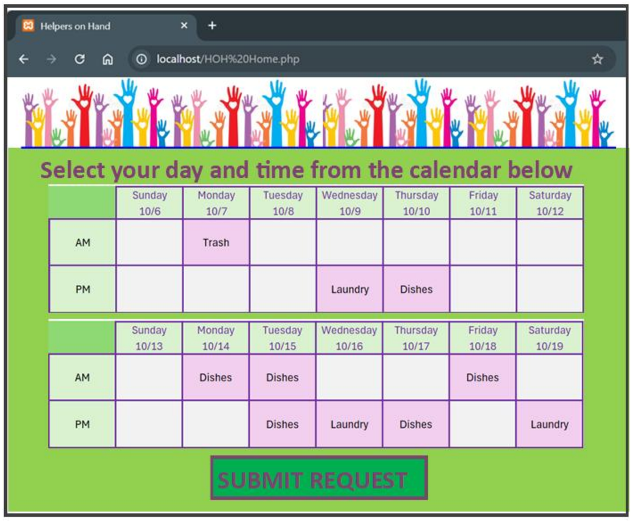
○ Maintenance ○ Run errands

○ Call doctor, insurance, bank, utility company, etc.

○ Other:

NEXT

A calendar will appear showing the days those chores are available to be completed in either the AM or PM. When the user clicks on the blocks pertaining to their choices, the boxes will appear darker and have a yellow outline. Then the user will click a button to SUBMIT REQUEST.

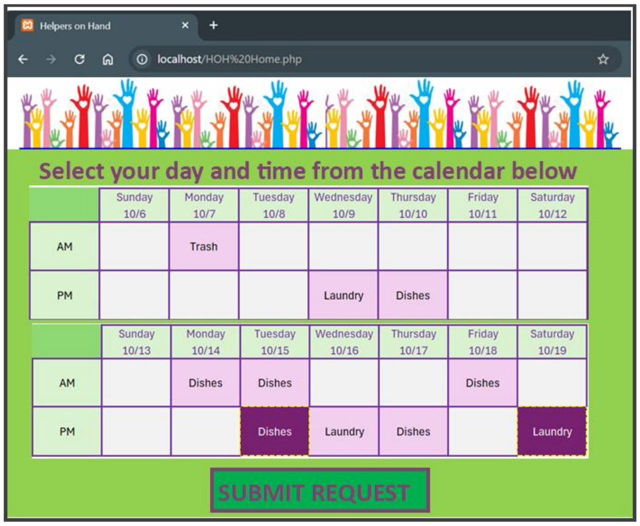


Select your day and time from the calendar below

	Sunday 10/6	Monday 10/7	Tuesday 10/8	Wednesday 10/9	Thursday 10/10	Friday 10/11	Saturday 10/12
AM		Trash					
PM				Laundry	Dishes		

	Sunday 10/13	Monday 10/14	Tuesday 10/15	Wednesday 10/16	Thursday 10/17	Friday 10/18	Saturday 10/19
AM		Dishes	Dishes			Dishes	
PM			Dishes	Laundry	Dishes		Laundry

SUBMIT REQUEST



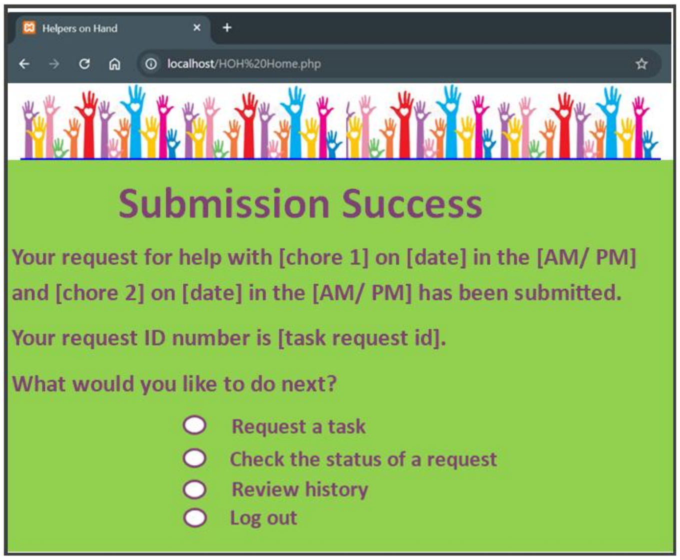
Select your day and time from the calendar below

	Sunday 10/6	Monday 10/7	Tuesday 10/8	Wednesday 10/9	Thursday 10/10	Friday 10/11	Saturday 10/12
AM		Trash					
PM				Laundry	Dishes		

	Sunday 10/13	Monday 10/14	Tuesday 10/15	Wednesday 10/16	Thursday 10/17	Friday 10/18	Saturday 10/19
AM		Dishes	Dishes			Dishes	
PM			Dishes	Laundry	Dishes		Laundry

SUBMIT REQUEST

When the task request is successfully submitted, a screen will display a summary of the requested chore(s), the date(s) and time(s) of expected completion, and an automatically generated request ID number. Next will be a list of options to continue. They will be radio buttons and will take the user to the action chosen.



Submission Success

Your request for help with [chore 1] on [date] in the [AM/ PM] and [chore 2] on [date] in the [AM/ PM] has been submitted.

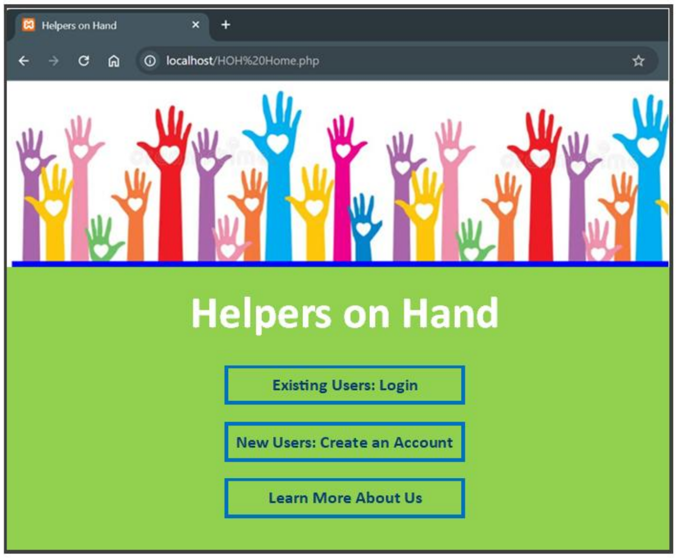
Your request ID number is [task request id].

What would you like to do next?

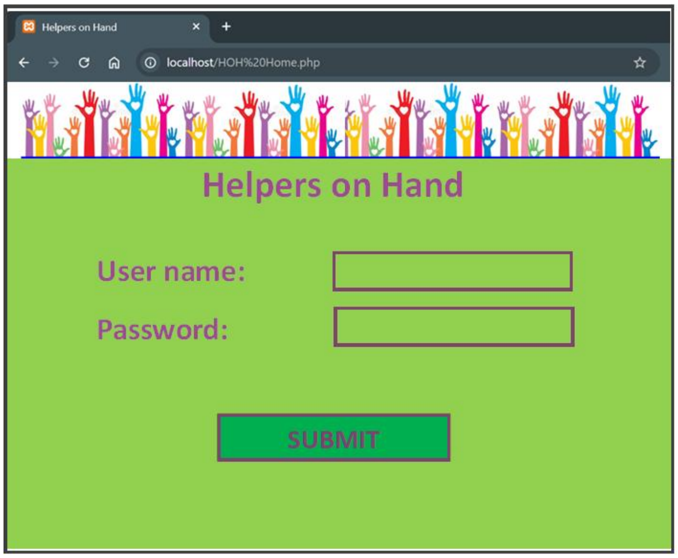
○ Request a task
○ Check the status of a request
○ Review history
○ Log out

Use Case 2: Existing User Checks Request Status

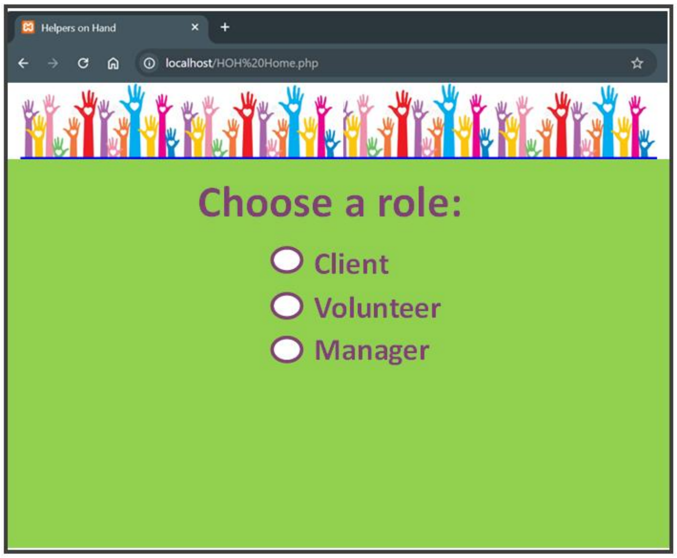
From the landing page, a user clicks the button associated with logging in as an existing user.



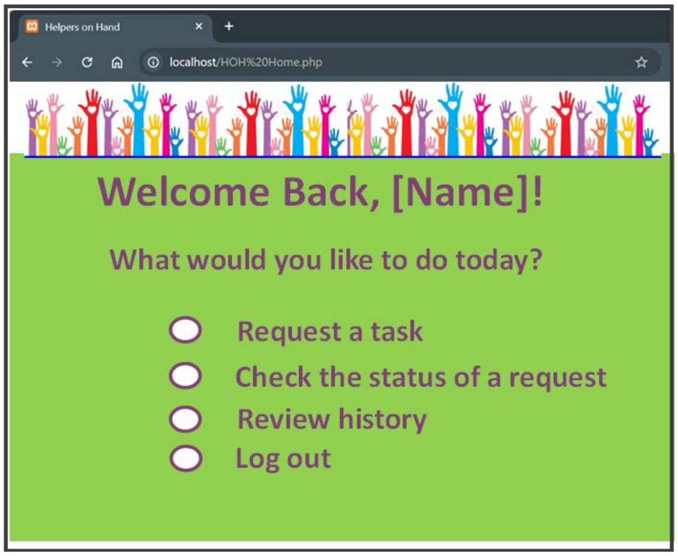
Then they will enter their username and password in a form accepting data types of text and password. Then click a submit button



They will choose the role in which they are currently acting. One user may hold more than one role. Clicking the radio button will move the user to the next screen. They will click Client.



The user will land on a page welcoming them by name. They will click a radio button to Check the status of a request which will advance them to the next screen.

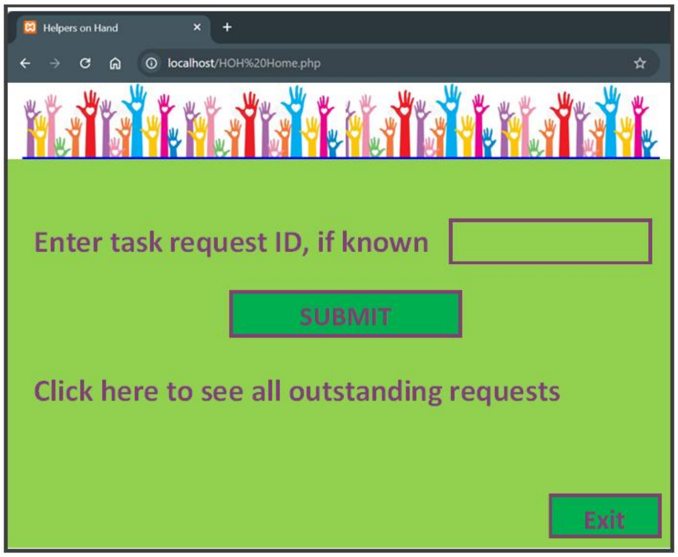


Welcome Back, [Name]!

What would you like to do today?

- ○ Request a task
- ○ Check the status of a request
- ○ Review history
- ○ Log out

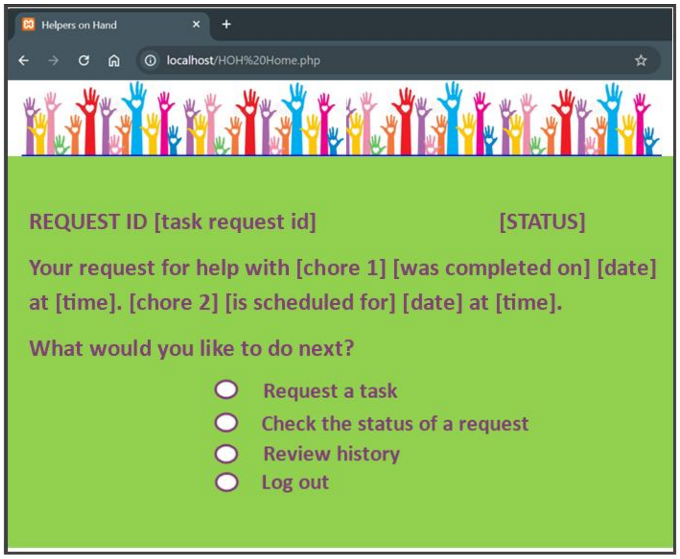
They will enter a task request ID number in an integer field, then click Submit to search for their request. If they do not have a request ID number, they can click the sentence 'Click here to see all outstanding requests', or they can click Exit to take them back to the initial landing page.



Enter task request ID, if known

Click here to see all outstanding requests

This screen will show them the results of their search including a summary of the request with the current status of the request which could be: Submitted, Scheduled, Cancelled, or Complete. Then it will display radio buttons with options of what to do next.



REQUEST ID [task request id] [STATUS]

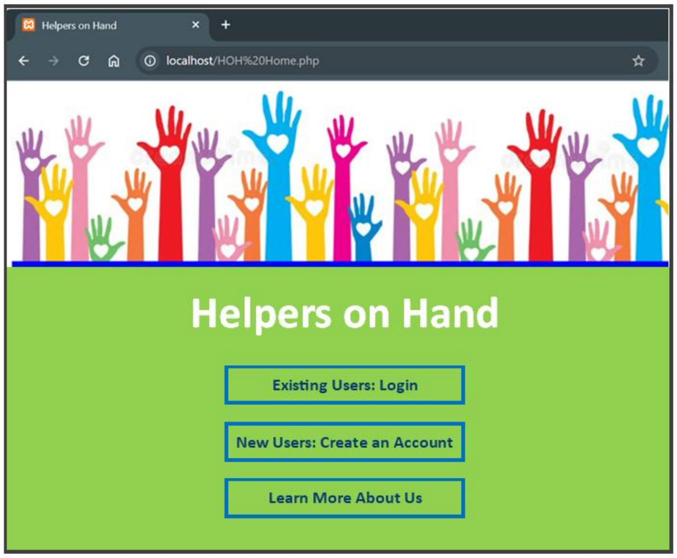
Your request for help with [chore 1] [was completed on] [date] at [time]. [chore 2] [is scheduled for] [date] at [time].

What would you like to do next?

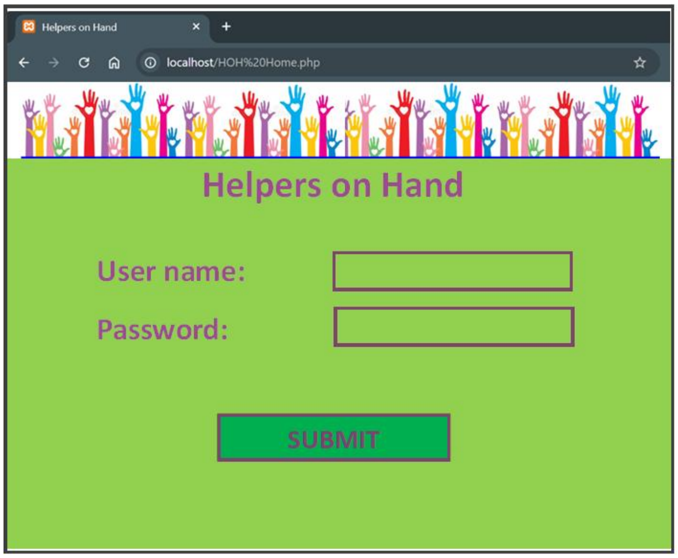
- ○ Request a task
- ○ Check the status of a request
- ○ Review history
- ○ Log out

Use Case 3: Existing User Reviews history

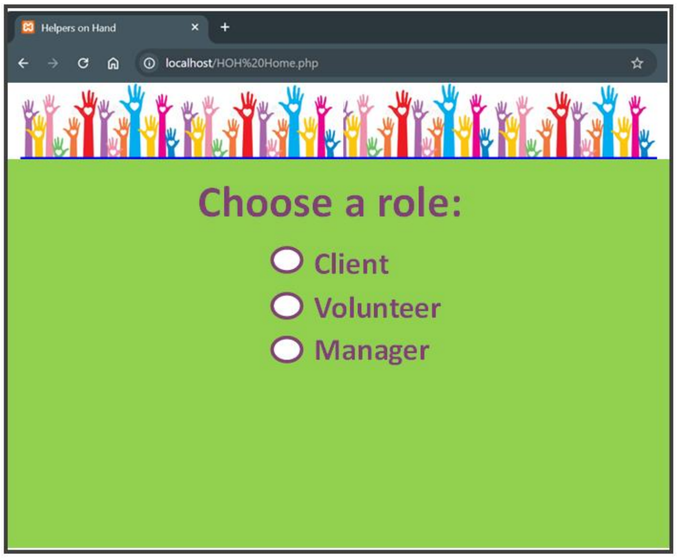
From the landing page, a user clicks the button associated with logging in as an existing user.



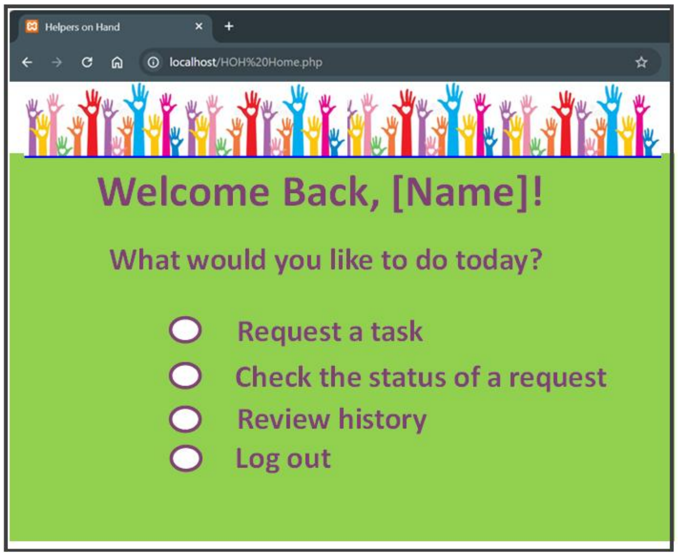
Then they will enter their username and password in a form accepting data types of text and password. Then click a submit button



They will choose the role in which they are currently acting. One user may hold more than one role. Clicking the radio button will move the user to the next screen. They will click Client.



The user will land on a page welcoming them by name. They will click a radio button to Review history which will advance them to the next screen.

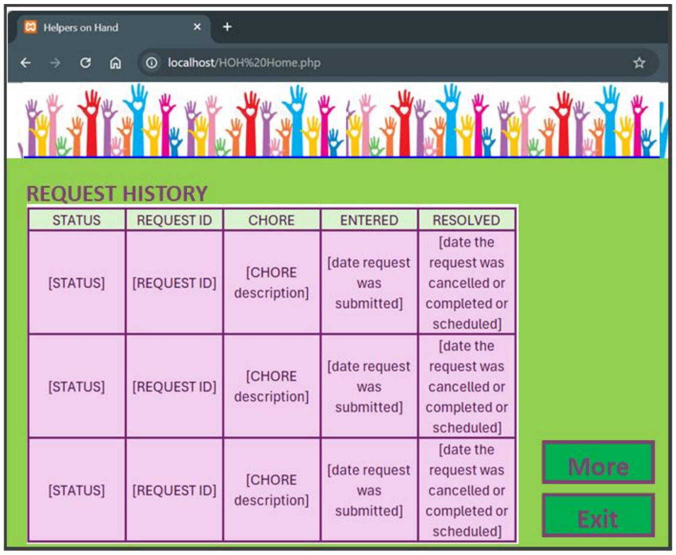


Welcome Back, [Name]!

What would you like to do today?

○ Request a task
○ Check the status of a request
○ Review history
○ Log out

This screen will be a table populated with all of the request this client has ever requested with details about the requests and their statuses. This will only show them the requests where they are the requester/ client. If they are also a volunteer, those tasks will not appear here. They will need to switch roles on the home page.

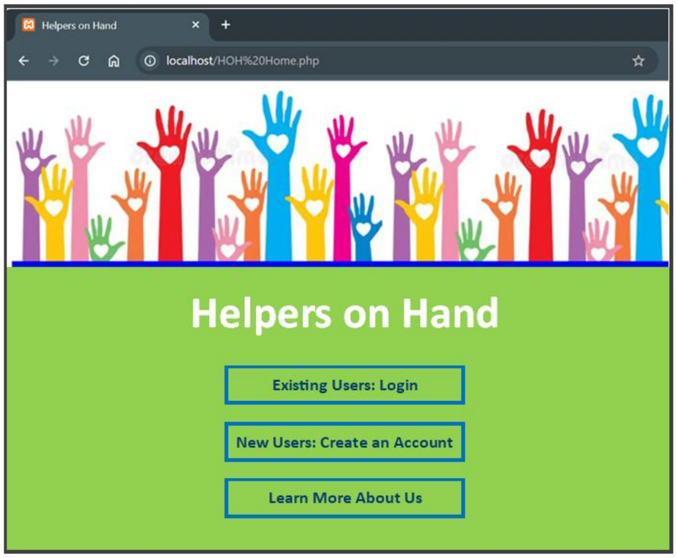


REQUEST HISTORY

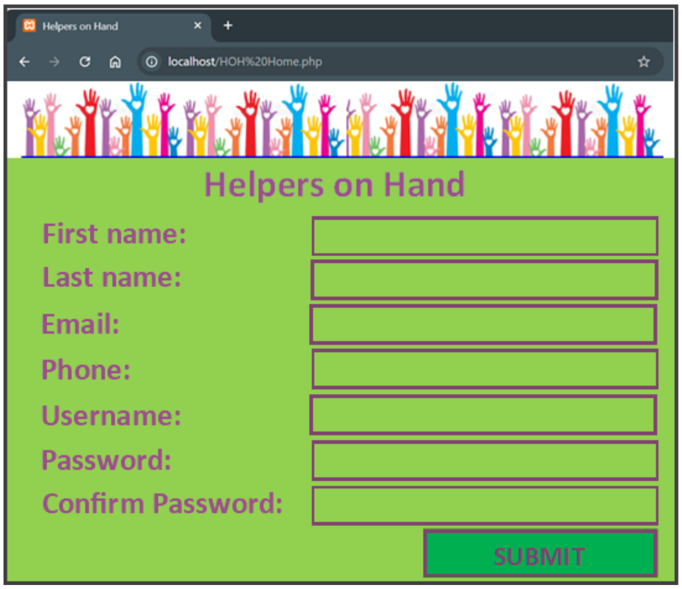
STATUS	REQUEST ID	CHORE	ENTERED	RESOLVED
[STATUS]	[REQUEST ID]	[CHORE description]	[date request was submitted]	[date the request was cancelled or completed or scheduled]
[STATUS]	[REQUEST ID]	[CHORE description]	[date request was submitted]	[date the request was cancelled or completed or scheduled]
[STATUS]	[REQUEST ID]	[CHORE description]	[date request was submitted]	[date the request was cancelled or completed or scheduled]

Use Case 4: New User Creates Account

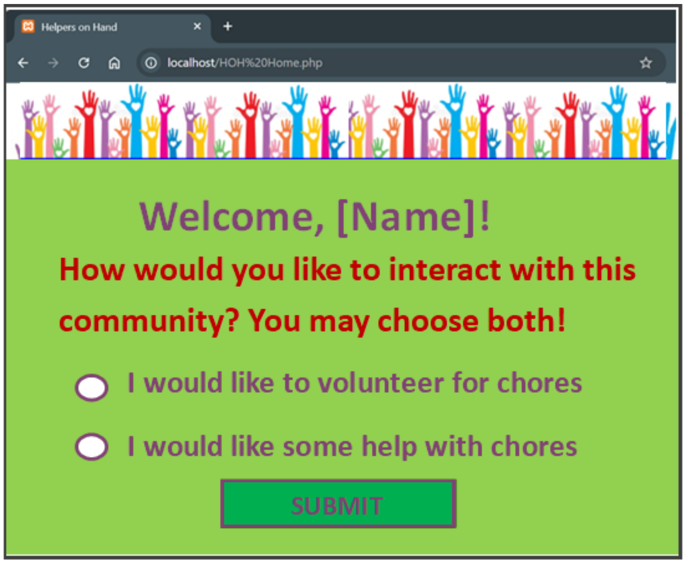
From the landing page, a user clicks the button associated with New Users to create an account.



The next screen asks for basic information in plain text fields. When the data is entered and the submit button is pushed, the Username field will run against the database to make sure it is unique. The fields for Password and Confirm Password will have a method comparing the input to be sure they are exact. If both methods run without failing, the account is created.



If the account creation was successful, this screen will appear asking the user in what capacity they would like to use the site, either as a volunteer helper, or a client, or both. When they click the submit button it will take them to the next screen.



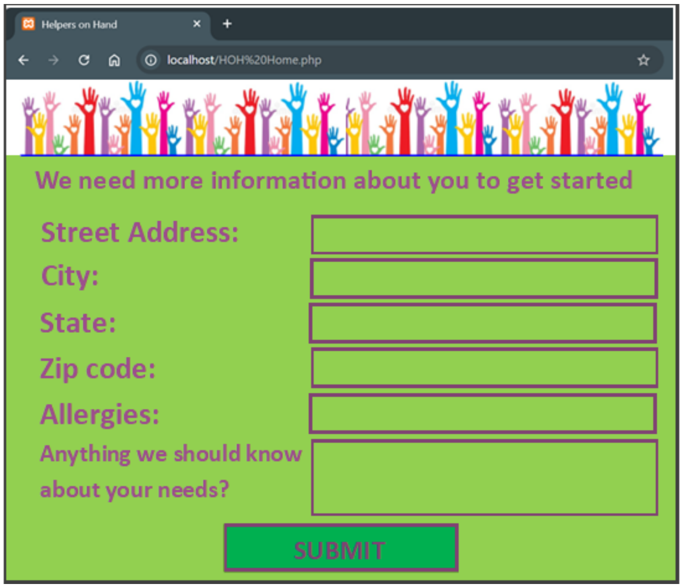
Welcome, [Name]!

How would you like to interact with this community? You may choose both!

○ I would like to volunteer for chores

○ I would like some help with chores

On this screen we ask the new user for more information.



We need more information about you to get started

Street Address:

City:

State:

Zip code:

Allergies:

Anything we should know about your needs?

Project Plan

Traceability Matrix

System Requirements

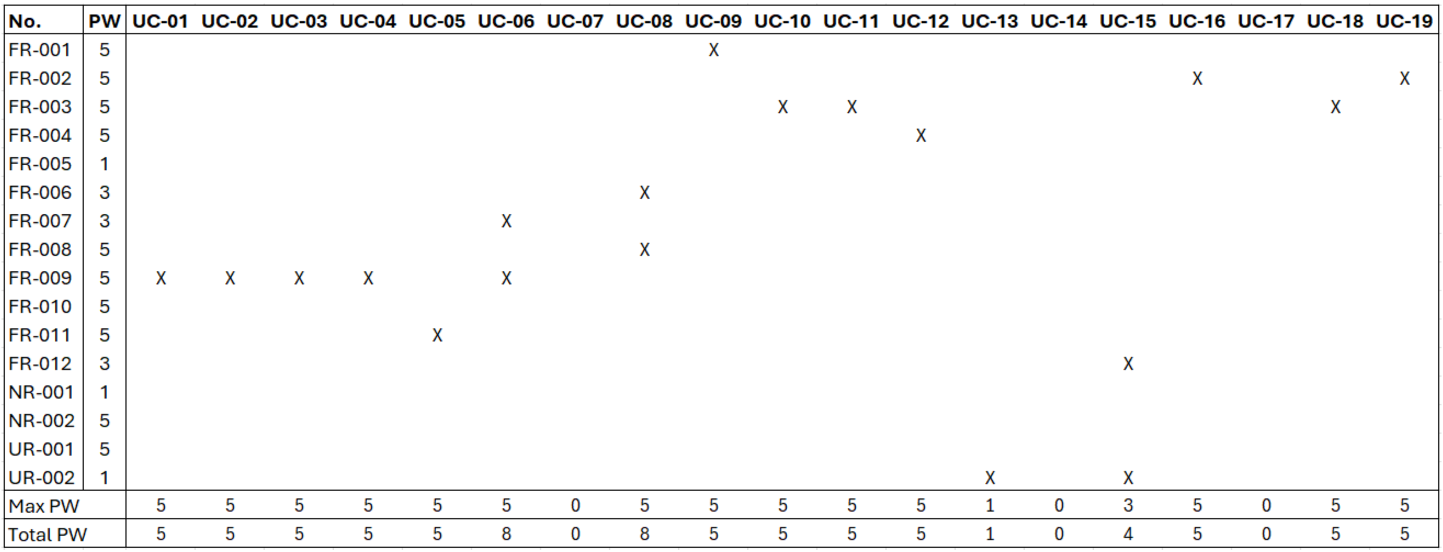
No.	Priority Weight	Description
FR-001	5	Helpers should be able to choose multiple values from a list of chores they can complete
FR-002	5	Clients should be able to choose multiple values from a list of chores they need completed
FR-003	5	Helpers should be able to select multiple days and multiple times in a day they are available to perform the chores they are able to complete
FR-004	5	When helpers choose days and times they can perform certain chores, those days and times should show in the calendar for clients to select from
FR-005	1	Each chore should have a color or icon assigned to it
FR-006	3	Users in the system should be able to hold more than one role
FR-007	3	System should display a chore chart dynamically
FR-008	5	Users should create a unique username with a password
FR-009	5	Managers should be able to see all the users and chores
FR-010	5	Managers should be able to edit all the users and chores
FR-011	5	The manager should be able to create tasks to assign to helpers
FR-012	3	The clients should be able to see the status of their chore requests
NR-001	1	Website color scheme should include tones that are proven to elicit calm vibes
NR-002	5	The forms on the website should be very simplistic with basic fields and very little direction needed
UR-001	5	Radio buttons should appear when the login button is clicked allowing the user to choose which role they are logging into
UR-002	1	Helper and Client users should see a list of their past chore tickets

Use Cases

No.	Description
UC-01	Access database: log into SQL database
UC-02	Create records, clients: create records for clients
UC-03	Create records, helpers: create records for helpers
UC-04	Create records, chores: create records for chores
UC-05	Assign transactions: review the request queue and assign chores to helpers for clients

No.	Description
UC-06	Populate chore chart: create a report that shows which helpers are completing which chores for the clients for a given period of time
UC-07	Run reports: write queries to display data
UC-08	Log in/Log out: log into and out of the site
UC-09	Choose chore: choose which chore or chores they can complete
UC-10	Choose day: choose the day they are available to complete these chores
UC-11	Choose time: choose the time they are available to complete these chores
UC-12	Submit availability: sends the days and times the helper is available for each chore they can do, so it will appear as available to the clients
UC-13	View history: see a list of past transactions completed
UC-14	Log in/Log out: log into and out of the site
UC-15	View history: see a list of past requests and statuses
UC-16	Choose chore: choose which chore or chores they need completed
UC-17	Choose day: choose the day they would like to have the chore completed
UC-18	Choose time: choose the time they would like to have the chore completed
UC-19	Submit request: send a chore request to a queue for the manager to retrieve and assign

Traceability Matrix



No.	PW	UC-01	UC-02	UC-03	UC-04	UC-05	UC-06	UC-07	UC-08	UC-09	UC-10	UC-11	UC-12	UC-13	UC-14	UC-15	UC-16	UC-17	UC-18	UC-19
FR-001	5									X										
FR-002	5																X			X
FR-003	5										X	X							X	
FR-004	5												X							
FR-005	1																			
FR-006	3								X											
FR-007	3						X													
FR-008	5								X											
FR-009	5	X	X	X	X		X													
FR-010	5																			
FR-011	5					X														
FR-012	3															X				
NR-001	1																			
NR-002	5																			
UR-001	5																			
UR-002	1													X		X				
Max PW		5	5	5	5	5	5	0	5	5	5	5	5	1	0	3	5	0	5	5
Total PW		5	5	5	5	5	8	0	8	5	5	5	5	1	0	4	5	0	5	5

Reference