Cleveland State University
EngagedScholarship@CSU

ETD Archive

2018

Drughelp.Care – a Web Application for the Discovery of Drug Addiction Treatment Facilities

Sachin Hiriyanna
Cleveland State University

Follow this and additional works at: https://engagedscholarship.csuohio.edu/etdarchive

Part of the Computational Engineering Commons, Medicine and Health Sciences Commons, and the Social and Behavioral Sciences Commons

How does access to this work benefit you? Let us know!

Recommended Citation

Hiriyanna, Sachin, "Drughelp.Care – a Web Application for the Discovery of Drug Addiction Treatment Facilities" (2018). *ETD Archive*. 1105.
https://engagedscholarship.csuohio.edu/etdarchive/1105

This Thesis is brought to you for free and open access by EngagedScholarship@CSU. It has been accepted for inclusion in ETD Archive by an authorized administrator of EngagedScholarship@CSU. For more information, please contact library.es@csuohio.edu.

DRUGHELP.CARE – A WEB APPLICATION FOR THE DISCOVERY OF DRUG ADDICTION TREATMENT FACILITIES

SACHIN HIRIYANNA

Bachelor of Technology in Information Technology

Vellore Institute of Technology

June 2016

Submitted in partial fulfilment of requirements towards the degree of

Master of Science in Software Engineering

at the

Department of Electrical Engineering & Computer Science

CLEVELAND STATE UNIVERSITY

May 2018

We hereby approve this thesis for

SACHIN HIRIYANNA

Candidate for the Master of Science in Software Engineering degree for the

Department of Electrical Engineering and Computer Science

and the CLEVELAND STATE UNIVERSITY'S

College of Graduate Studies by

Committee Chairperson, Dr. Wenbing Zhao

_____

Department of Electrical and Computer Science & Date

Committee Member, Dr. Lili Dong

_____

Department of Electrical and Computer Science & Date

Committee Member, Dr. Yongjian Fu

_____

Department of Electrical and Computer Science & Date

Student's Date of Defense: 05/07/2018

ACKNOWLEDGEMENT

I would like to express the deepest appreciation to my mentor and advisor, professor Dr. Wenbing Zhao, for continually helping me through this research and development. He has not only supported me technically but also morally by providing me with constant appreciation and confidence in what I'm doing and also by providing me with space and resources in his own laboratory. Without his guidance, support and invaluable advice this thesis would not have been possible. I will always admire his persistence and commitment towards technology and innovation.

I would like to thank my committee members Dr. Lili Dong whose work demonstrated to me that concern for Advanced Control Systems and Dr. Yongjian Fu whose tremendous researches are related to Web Technologies and Data Mining.

I would like to thank my family, especially my father, mother and sister for their unconditional love and trust in me. And for standing by and supporting me all the time both morally and financially to finish this course of study and research.

And finally to all my friends and relatives for having the faith in me and upholding the friendship all the time.

DRUGHELP.CARE – A WEB APPLICATION FOR THE DISCOVERY OF DRUG ADDICTION TREATMENT FACILITIES

SACHIN HIRIYANNA

ABSTRACT

One of the current and biggest problems in the system of emergency care for the drug overdose epidemic, is the failure of information delivery on nearby treatment facilities. Even though some initiatives on the internet tried to solve this issue, they either failed in delivering the information or in providing ease of usability. This thesis research aims to develop a web application which,

1. Delivers a highly accurate, easy-to-understand and targeted information in a timely manner for drug abusers and their well-wishers.
2. Provides an eco-system for the treatment facilities with an easy-to-use backend to constantly update their complex information along with automatic email reminders and data completion progress indicators.

In the first step of this thesis, a study was carried out on efficiency and ease of access to information by the drug abusers and by their well-wishers on nearby treatment facilities. In the second step, another study carried out in order to find out the ease of information delivery by treatment facilities about themselves. Finally, based on the established truth, a front and back-end of the web application are designed, developed using LAMP stack, deployed and tested. A search engine for the drug abusers and their well-wishers preserves the complete anonymity, which is very important to hide the identities of the drug abusers.

TABLE OF CONTENTS

Page

ABSTRACT iv

LIST OF FIGURES vii

ACRONYM ix

CHAPTER

I: THE STUDY 1

1.1 Motivation 1

1.2 Study on Drug Aabusers 2

1.2.1 The Problems 2

1.2.2 The Solutions 3

1.3 Study on Treatment Facilities (Agencies) 4

1.3.1 The Problem 4

1.3.2 The Solution 4

1.4 Related Work 5

1.5 Web Application Development 5

1.5.1 Bootstrap 6

II: REQUIREMENTS SPECIFICATIONS 7

2.1 System Requirements 7

2.1.1 Application Development 8

2.1.2 Web Hosting 8

2.2 User Requirements 9

2.2.1 Drug abusers 9

2.2.2 Treatment Facilities 10

III: SOFTWARE DESIGN AND PLANNING 11

3.1 User Interface Design 12

3.1.1 User Stories and brainstorming 12

3.1.2 Use of Accordion 13

3.2 Database Design 13

3.2.1 Tables 14

IV: SOFTWARE DEVELOPMENT 20

4.1 Environment Setup 20

4.2 Version Control using Git 20

4.3 Implementation 22

4.3.1 Homepage – www.drughelp.care 22

4.3.2 Search Page 23

4.3.3 Treatment facility – Account Signup 24

4.3.3 Treatment facility – Account Sign-in and Password Reset 27

4.3.4 Treatment facility – Dashboard 29

4.3.5 Treatment facility – Contact Person 33

V: CONCLUSION 35

REFERENCES 36

APPENDIX 38

LIST OF FIGURES

Figure Page

1: Ohio Overdose Death Rates by Age 2

2: Bootstrap 4 Supported Browsers on Mobile Devices 6

3: Base Architecture 11

4: User stories and brainstorming on a whiteboard 12

5: Use of Accordion 13

6: Database Relationship Model 14

7: Agency Table 14

8: Agency Table – Indexes 14

9: Service Parent Table 15

10: Service Parent Table – Indexes 15

11: Service Child Table 15

12: Service Child Table – Indexes 16

13: Registered Service Table 16

14: Registered Service Table – Indexes 16

15: Registered Service Offering Table 17

16: Registered Service Offering Table – Indexes 17

17: Registered Service Dataset Table 18

18: Registered Service Dataset Table – Indexes 18

19: Password Reset Table 18

20: Password Reset Table – Indexes 19

21: Contact Person Table 19

22: Contact Person Table - Indexes 19

23: GitHub desktop tool 21

24: My Contributions in Line of Code 21

25: Homepage 22

26: Search page 23

27: Search page's sorting and filtering options 24

28: Agency Account Creation 25

29: Agency Email Verification 26

30: Agency Email Verification Action 26

31: Agency Sign-in page 27

32: Agency account - Reset password page 28

33: Agency reset password email 28

34: Agency's Dashboard 29

35: Service Registration page 30

36: Service Offering Registration page 31

37: View Registered Services page 31

38: Service Details page 32

39: Service Details page - Example modal 32

40: Contact Person email 33

41: Contact Person - Availability page 34

42: Contact Person - Waitlist page 34

43: Screenshots of web application on Mobile phone 35

ACRONYM

a. CSS – Cascading Style Sheets

b. SMTP – Simple Mail Transfer Protocol

c. FK – Foreign Key

d. LAMP – Linux + Apache + MySQL + PHP (A development stack)

e. PHP – PHP Server side scripting language

f. MySQL – Database query Language

g. SFTP – Secured File Transfer Protocol

h. WWW – World Wide Web

CHAPTER I

THE STUDY

1.1. Motivation

More than 64,000 Americans lost their lives from drug overdoses in 2016 alone [1], and this number is increasing. Most of these lives could have been saved if either the victim or their friends or family members acted quickly to reach out to treatment facilities in a timely manner. The real problem lies in the fact that these treatment facilities are either non discoverable or not relevant. For example, some treatment facilities do not accept pregnant ladies and most of time this information is not known to the people unless they get to the facility physically.

There are several web apps that offer a searchable list of treatment facilities, however they lack in providing accurate information or updated information like information on availability, waitlist or any other service specific information.

Thus, it is much required to develop the web application which is having the data that is trustable and is updated dynamically by the treatment facilities themselves.

1.2 Study on Drug Abusers

In the state of Ohio, drug addiction, abuse, and overdose deaths is the most disastrous health issue. And the state of Ohio is the leader in the country for the drug overdose deaths per capita, and the rates are continuously rising, devastating the lives within families, communities, and also local governing bodies across the state of Ohio. Figure 2 shows the overdose deaths rates by age in the state of Ohio from 2007 to 2016 (Incomplete year). As we can analyze the trend, the rates are just increasing and it is very sad to know there are only a few initiates actually doing something useful to help the drug addicts.

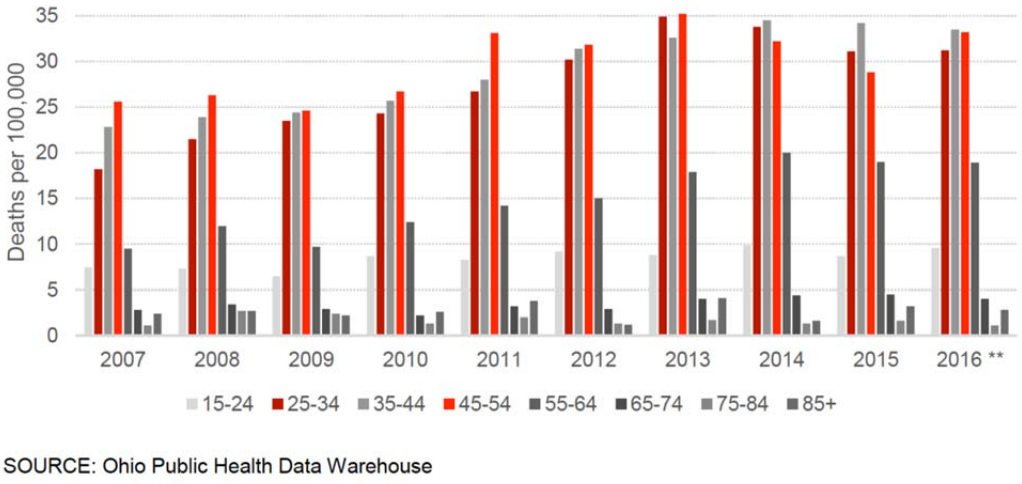


Figure 1: Ohio Overdose Death Rates by Age

1.2.1 The Problems: Although there a few initiatives who are trying to help drug abusers, the following are the problems that are facing by the end-users.

- The drug abuser wouldn't know the admission policies and whether or not the treatment facility would accept their condition. Ex: Treatment Center X doesn't admit any patients who are pregnant.
- The drug abuser wouldn't know the availability or waitlist of a particular treatment or service unless and until they make a phone call to every facility they are interested in.
- The drug abuser is very confused to navigate and look around the website since the design is very cluttered and maybe filled with ads or useless web components.
- Unable to get responsive and clean website on mobile phone or tablet.

1.2.2 The Solutions: To address the above mentioned problems, the following features are considered to develop within the web application as preferred solutions.

- To show the availability and waitlist for admission dynamically.
- To show the constrains or restriction for admission to particular treatment or service.
- To provide accurate information, verified and updated by treatment facilities themselves.
- To provide quick access buttons like call, directions or website.
- To provide clean web design to help intoxicated person to get help without any confusion.
- To provide options to filter and sort the list of treatment facilities based on various attributes including availability and waitlist.
- To provide responsive design which even supports mobile devices.

1.3 Study on Treatment Facilities (Agencies)

Although treatment facilities are willing provide real-time data on availability, waitlist and admission policies, it is unfortunate that nobody has developed a web application which will let them deliver that to the potential customers.

1.3.1 The Problem: The following points are the problems that treatment facilities currently have, that needs to be addressed to help drug abusers obtain trustable information on them.

- No dashboard or admin panel developed for the treatment facilities to administer their data.
- No option on the existing discovery platforms for the treatment facility to update their availability and waitlist for each and every drug related treatment services.
- No option to visualize the data entry progress for every service they register.
- No option for them to add contact persons to get scheduled reminders to update the data on treatment services.
- No option to register a treatment service with multiple time or location based sub-services.

1.3.2 The Solution: To address the above mentioned problems, the following features are considered to develop within the web application as preferred solutions for treatment facilities.

- To develop a dashboard for administrators.
- To provide option to resister, update and delete a service.

- To provide option to register multiple sub-services under a service based on location and hours of operation.
- To provide option to add multiple contact persons to a particular service.
- To send email reminders to constantly update the availability and waitlist of the treatment services.
- To show data entry progress indicator for every service they have registered.

1.4 Related Work

EmeraldJennyfoundation.com [2] was founded last year to solve the challenge of discovery of drug abusers treatment facility, it has the search feature that has filters for outpatient, residential and Medication Assisted Treatment options [3]. But this platform also has most of the problems that are listed above including failure to provide availability and waitlist information.

1.5 Web Application Development

Users tends to uninstall the mobile app at some point of time because of limited resources on a mobile device like storage space, battery life etc. [3]. Now by taking advantages of a good and efficient CSS Framework on our web application, we can deliver them with best experience possible on whatever the device they are accessing it on without having them to download or install an app. Because of these reasons, it decided that the best option to develop is a web application but not a mobile application.

1.5.1 Bootstrap: Bootstrap [4] is the most used and advanced CSS Framework to build Web applications. It not only is easy to learn and implement but is also very efficient in terms of the responsiveness of web components on various devices with different screen sizes. Hence we can take advantages of these features to build web application which works seamlessly on the mobile phone or tablet's browser without having to reach out to a browser on computer by the user.

Bootstrap follows the twelve column Grid system [5] and a series of containers, rows, and columns to layout and align the contents. It is built using flexbox [6] and is fully responsive to every screen sizes.

Below figure shows the supportiveness of Bootstrap version 4 on browsers of various mobile devices [7].

	Chrome	Firefox	Safari	Android Browser & WebView	Microsoft Edge
Android	Supported	Supported	N/A	Android v5.0+ supported	Supported
iOS	Supported	Supported	Supported	N/A	Supported
Windows 10 Mobile	N/A	N/A	N/A	N/A	Supported

Figure 2: Bootstrap 4 Supported Browsers on Mobile Devices

CHAPTER II

REQUIREMENTS SPECIFICATIONS

Usually, the requirements in any software development lifecycle is broadly classified into two main details; system and user requirements, where in this project, software development engineers need more technical specification on the web application whereas the end-users i.e. drug addicts and treatment facilities needs more of a high-level statements of specifications on the web application. [4]

2.1 System Requirements

The system requirements specifications mean more technically detailed description of the web application to be built and the operational conditions as to how it is going to be used, and development constrains like what programming languages to use, which server or database to use and also what architectural pattern to follow. Having the systems requirement ready is a crucial part to make sure the future software developers will get the clear picture as to what exactly to do to maintain the web application. Each sub-heading under this paragraph will give the detailed system requirements for the web application.

2.1.1 Application Development:

1. **Front-end development:** DrugHelp.Care web application front end may require constant design changes based on the feedback we get from the drug abusers. For this reason, it is important to use a CSS Framework which provides us the flexibility to reuse components and modify the design of entire application from a single file.
2. **Back-end development:** It is very important for us to build the application that is scalable and is easy to maintain. For this reason, we have to choose a development stack that is widely used and has good support on the internet.

2.1.2 Web Hosting:

1. **Web Server:** Since the web application deals with huge amount of data with complex database transaction and queries, it is required to choose a server that can handle good amount of computation and can handle the load when hundreds or thousands of user access the app concurrently. For this reason, it is wise to choose a serve that is auto-scaling, load-balancing and has the option to pay as per the usage.
2. **Email Server:** We need an email server that supports SMTP and message encryption techniques. And to avoid using blacklisted servers accidently from third parties, it always good to create a brand new instance of email server which gives us the confidence that the email will not end up in spam folders.

3. **Database:** It is important to choose a relational database which offers all the advanced features of query language like constrains, joins etc. also the one which provide the user interface to visualize the data in schema.

2.2 User Requirements

The main end-users of the web application are drug abusers, their friends or families and treatment facility administrators.

2.2.1 Drug abusers: Since the drug abusers will have short window of memory when they are intoxicated, the look and feel of the treatment search page needs to have clean and comfortable user interface design. The following are the design characteristics to consider while designing the user interfaces.

1. Minimal use of colors
2. Use of non-flashy colors
3. Loosely placed UI components
4. Enough room to perform mouse click
5. Less or zero auto changing contents
6. Easy navigation between components
7. Collapsible components to hide unwanted text
8. Highlighting the important buttons

2.2.2 Treatment Facilities: The treatment facilities will be dealing with inputting a massive amount of data as a process of signing up to the web application. For each of the treatment services and sub-services they are offering, they need to input the details accordingly, maintain and track the progress of data entry. Since the whole application depends on the accuracy and data completion progress by the treatment facilities, it is required to design efficient user interfaces to help treatment facilities understand each step of the signup. The following points briefly summarizes the expected and mandatory characteristics of Treatment Facilities' part of the application.

1. To easily register one or more services at a time.
2. To easily understand the complete flow of registration.
3. To show bird's eye view of huge dataset of services to select from.
4. To provide data completion progress indicators.
5. To provide feedback on every important steps.
6. To remind the contact persons to update the information periodically.
7. To provide easy and efficient interface to contact persons.
8. To easily manage and update each service details.
9. To easily navigate between the services.
10. To provide high security on the data that they are entering.
11. To secure the administrators account with authentication techniques.
12. To provide email notifications when an account information changes.
13. To provide all the utilities to completely authorize a contact person for a service.
14. To secure administrator's account from hacking by computer bots.

CHAPTER III

SOFTWARE DESIGN AND PLANNING

Software design is a process of translating or visualizing the user requirements, which will help the software developer to clearly understand the exact needs of end-users. Whereas software planning is the process of planning the development of software application completely by defining the time and cost of the Software application. Figure 3 shows the base architecture of the web application.

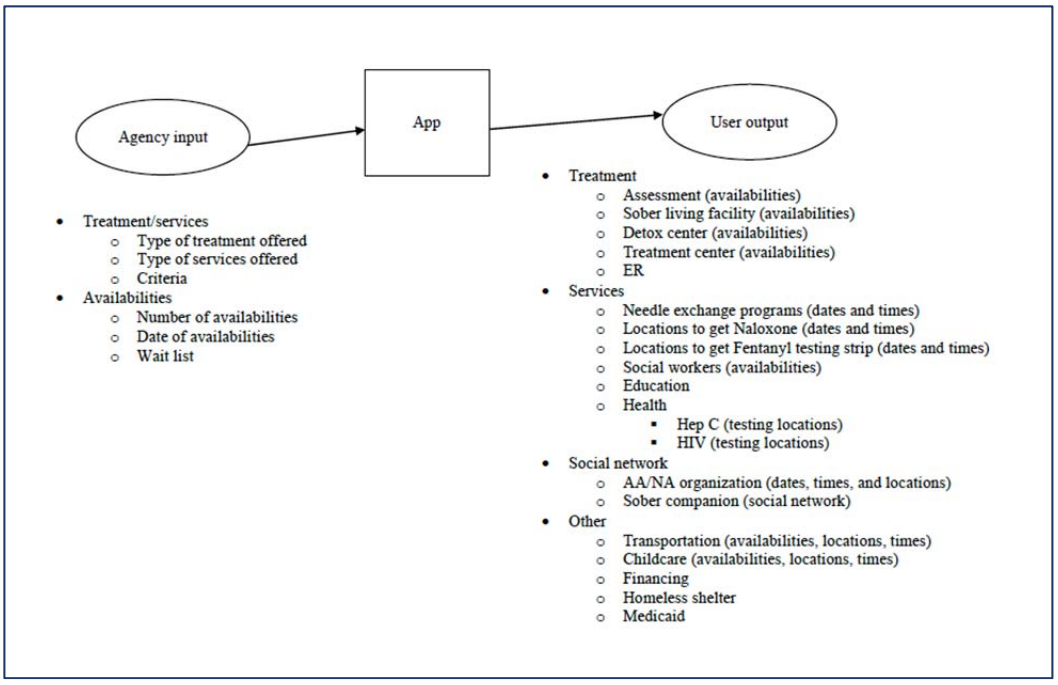


Figure 3: Base Architecture

3.1 User Interface Design

User interface is the frontend part of the application where the end-users can perform interactions with the backend through GUI components like buttons, text field etc. In this web application, drug abusers being one of the end-user, requires us to design the UI very carefully with clean design and properly placed web components.

3.1.1 User Stories and brainstorming: As per the requirements, we need to create user stories and brainstorm among the stake holders on a whiteboard to verify the correctness and usability before proceeding to the development of the front-end. Figure 4 shows the brainstorming of user stories on a whiteboard. This step very important as during the process of brainstorming, we will come across all the web UI components that we need to use during development. This helps us to think about challenges that may arise during the development.

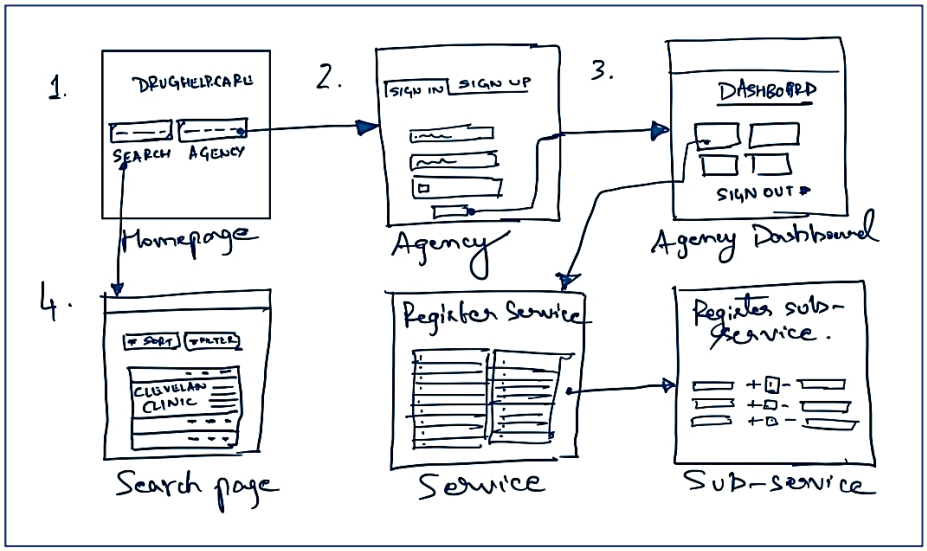


Figure 4: User stories and brainstorming on a whiteboard

3.1.2 Use of Accordion: Accordion is a web component that extends the Cards container [9] of Bootstrap in which organizes content within collapsible items. Accordion provides us the better way to manage the web view area, by allowing us to expand only one collapsible item at a time.

Figure 5 shows a simple example of the use of accordion using Bootstrap CSS Framework.

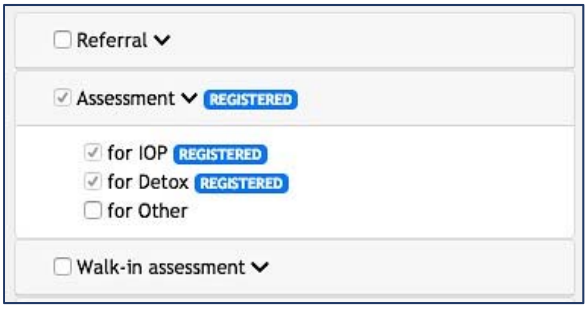


Figure 5: Use of Accordion

Working: When ‘Assessment’ tab is expanded with its children, every other i.e. ‘Referral’ and ‘Walk-in assessment’ tabs will have the children collapsed and hidden. This enables the better management of web area.

3.2 Database Design

Figure 6 shows the relationship model of database for the web application. The Registered Service table, Contact Person table and Password Reset table has a foreign key on Agency table. Whereas the Dataset table has a FK on Registered Service Offering table which in turn has FK on Registered Service table.

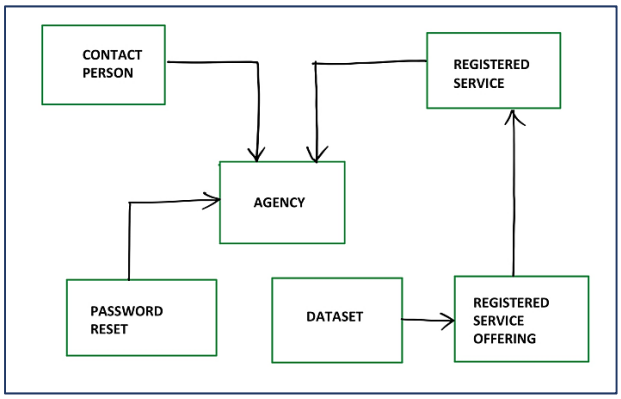


Figure 6: Database Relationship Model

3.2.1 Tables:

1. AGENCY TABLE: This table is used to store account information of Agencies. Figure 7 shows all the columns and data types. Figure 8 shows the Indexes of the Agency table.

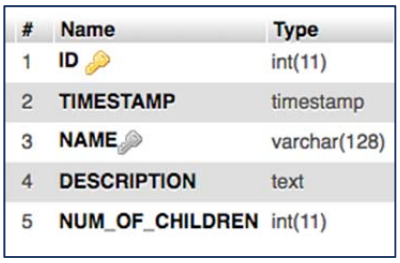
#	Name	Type
1	TIMESTAMP	timestamp
2	ID	int(11)
3	NAME	varchar(64)
4	WEBSITE	text
5	ADDRESS	text
6	CITY	varchar(64)
7	STATE	varchar(64)
8	ZIP	int(5)
9	PHONE	bigint(11)
10	EMAIL	varchar(128)
11	PASSWORD	varchar(32)
12	VERIFICATION_CODE	varchar(32)
13	IS_VERIFIED	bit(1)

Figure 7: Agency Table

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	6	A	No	
PHONE	BTREE	Yes	No	PHONE	6	A	No	
				EMAIL	6	A	No	
EMAIL	BTREE	Yes	No	EMAIL	6	A	No	
ID	BTREE	No	No	ID	6	A	No	

Figure 8: Agency Table – Indexes

2. SERVICE PARENT TABLE: This table is used to store set of all the services that agencies can choose from during registration. Figure 9 shows all the columns and data types. Figure 10 shows the Indexes of the Service Parent table.



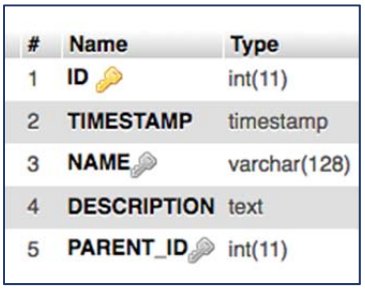
#	Name	Type
1	ID	int(11)
2	TIMESTAMP	timestamp
3	NAME	varchar(128)
4	DESCRIPTION	text
5	NUM_OF_CHILDREN	int(11)

Figure 9: Service Parent Table

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	30	A	No	
NAME	BTREE	Yes	No	NAME	30	A	No	

Figure 10: Service Parent Table – Indexes

3. SERVICE CHILD TABLE: This table is used to store set of all the sub-services that agencies can choose from during registration. Figure 11 shows all the columns and data types. Figure 12 shows the Indexes of the Service Child table.



#	Name	Type
1	ID	int(11)
2	TIMESTAMP	timestamp
3	NAME	varchar(128)
4	DESCRIPTION	text
5	PARENT_ID	int(11)

Figure 11: Service Child Table

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	18	A	No	
NAME	BTREE	Yes	No	NAME	18	A	No	
				PARENT_ID	18	A	No	
PARENT_ID	BTREE	No	No	PARENT_ID	18	A	No	

Figure 12: Service Child Table – Indexes

4. REGISTERED SERVICE TABLE: This table is used to store all the registered services of an agency. Figure 13 shows all the columns and data types. Figure 14 shows the Indexes of the Registered Service table.

#	Name	Type
1	TIMESTAMP	timestamp
2	ID	int(11)
3	AGENCY_ID	int(11)
4	NAME	varchar(256)
5	SERVICE_PARENT_ID	int(11)
6	SERVICE_CHILD_ID	int(11)

Figure 13: Registered Service Table

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	30	A	No	
SERVICE_PARENT_ID	BTREE	Yes	No	SERVICE_PARENT_ID	30	A	No	
				SERVICE_CHILD_ID	30	A	No	
				AGENCY_ID	30	A	No	
ID	BTREE	No	No	ID	30	A	No	
AGENCY_ID	BTREE	No	No	AGENCY_ID	10	A	No	
				NAME	30	A	Yes	
SERVICE_CHILD_ID	BTREE	No	No	SERVICE_CHILD_ID	30	A	No	

Figure 14: Registered Service Table – Indexes

5. REGISTERED SERVICE OFFERING TABLE: This table is used to store all the service offerings of an agency based on time or location. Figure 15 shows all the

columns and data types. Figure 16 shows the Indexes of the Registered Service Offering table.

#	Name	Type
1	TIMESTAMP	timestamp
2	ID	int(11)
3	REGISTERED_SERVICE_ID	int(9)
4	AGENCY_ID	int(9)
5	NAME	varchar(256)
6	AVAILABILITY_COUNT	int(9)
7	WAITLIST_COUNT	int(9)
8	AVAILABILITY_INFO	varchar(256)
9	WAITLIST_INFO	varchar(256)
10	AVAILABILITY_LAST_UPDATED_INFO	varchar(256)
11	WAITLIST_LAST_UPDATED_INFO	varchar(256)

Figure 15: Registered Service Offering Table

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	44	A	No	
AGENCY_ID	BTREE	Yes	No	AGENCY_ID	11	A	No	
				NAME	44	A	No	
ID	BTREE	No	No	ID	44	A	No	
REGISTERED_SERVICE_ID	BTREE	No	No	REGISTERED_SERVICE_ID	44	A	No	

Figure 16: Registered Service Offering Table – Indexes

6. REGISTERED SERVICE DATASET TABLE: This table is used to store all the data of every service offered by an agency. Figure 17 shows all the columns and data types. Figure 18 shows the Indexes of the Registered Service Dataset table.

#	Name	Type
1	ID	int(11)
2	TIMESTAMP	timestamp
3	REGISTERED_SERVICE_ID	int(11)
4	AGENCY_ID	int(11)
5	ADDRESS	text
6	DAYS_AND_HOURS	text
7	TYPES_OF_INTERVENTION	text
8	TYPES_OF_DRUGS	text
9	MENTAL_HEALTH_SERVICES	text
10	TYPES_OF_INSURANCE	text
11	ELIGIBILITY_CRITERIA	text
12	ACCOMMODATIONS	text
13	ADDITIONAL_NOTES	text
14	HAS_ATLEAST_ONE_CONTACT	bit(1)

Figure 17: Registered Service Dataset Table

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	10	A	No	
REGISTERED_SERVICE_ID	BTREE	Yes	No	REGISTERED_SERVICE_ID	10	A	No	
				AGENCY_ID	10	A	No	
AGENCY_ID	BTREE	No	No	AGENCY_ID	10	A	No	

Figure 18: Registered Service Dataset Table – Indexes

7. PASSWORD RESET TABLE: This table is used to store all the password reset requests with unique URLs. Figure 19 shows all the columns and data types. Figure 20 shows the Indexes of the Password Reset table.

#	Name	Type
1	TIMESTAMP	timestamp
2	ID	int(11)
3	AGENCY_ID	int(11)
4	RESET_ID	varchar(32)
5	IS_ACTIVE	bit(1)
6	RESET_URL	text

Figure 19: Password Reset Table

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	0	A	No	
AGENCY_ID	BTREE	Yes	No	AGENCY_ID	0	A	No	
				RESET_ID	0	A	No	

Figure 20: Password Reset Table – Indexes

8. CONTACT PERSON TABLE: This table is used to stores all the contact person's information of every service of an agency. Figure 21 shows all the columns and data types. Figure 22 shows the Indexes of the Contact Person table.

#	Name	Type
1	TIMESTAMP	timestamp
2	ID	int(11)
3	PERSON_NUM	int(2)
4	AGENCY_ID	int(9)
5	SERVICE_OFFERING_ID	int(9)
6	NAME	varchar(128)
7	UNIQUE_ACCESS_CODE	varchar(32)
8	EMAIL	varchar(256)
9	PASSWORD	text
10	VERIFICATION_CODE	int(6)
11	IS_VERIFIED	bit(1)

Figure 21: Contact Person Table

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	9	A	No	
SERVICE_OFFERING_ID	BTREE	Yes	No	SERVICE_OFFERING_ID	9	A	No	
				EMAIL	9	A	No	
PERSON_NUM	BTREE	Yes	No	PERSON_NUM	2	A	No	
				AGENCY_ID	9	A	No	
				SERVICE_OFFERING_ID	9	A	No	
				EMAIL	9	A	No	
ID	BTREE	No	No	ID	9	A	No	
CONTACT_PERSON_ibfk_1	BTREE	No	No	AGENCY_ID	9	A	No	

Figure 22: Contact Person Table - Indexes

CHAPTER IV

SOFTWARE DEVELOPMENT

Software development is the process of implementing the web application based on user and system requirements.

4.1 Environment Setup

The following tools will be used to develop the web application.

1. Atom IDE, will be used as an editor and debugger.
2. CyberDuck SFTP, will be used to transfer files to and from the server to the local computer.
3. DataGrip, will be used as database client to visualize the schema.
4. PHPMyAdmin, is used to access production database on the server through HTTP.

4.2 Version Control using Git

A Version control system is a tool that allows us to track the iterative changes we make to our code base. It is important tool for any software developer because it allows us to revert back to previous version of the code base anytime when required and also allows us to let more developer collaborate towards the app development. Moreover, we

can have different branches of the repository with different versions of code base, this enables us to compare and differentiate between the code base of different versions.

Figure 23 shows the preview all my contributions towards the development web app on a most popular version control software called GitHub.

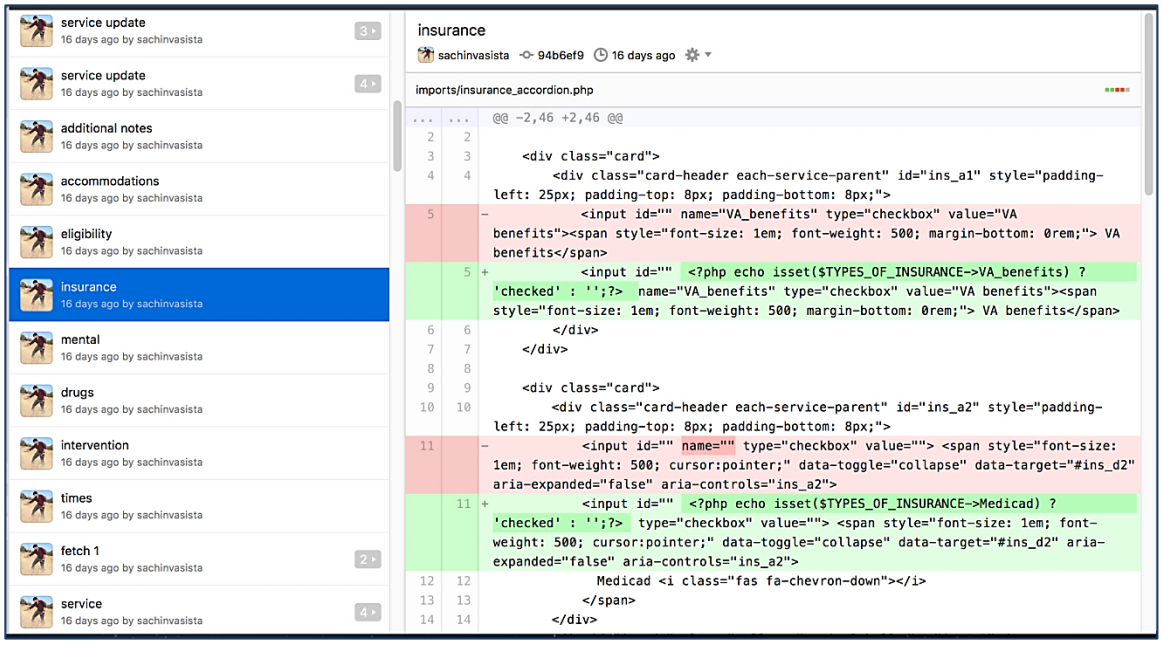


Figure 23: GitHub desktop tool

Figure 24 shows a graph generated by GitHub for my contributions towards web app development in Line of code over time.

Figure 24: My Contributions in Line of Code

4.3 Implementation

Each of the following heading showcases the screenshot of the live web application and their features and functionalities.

4.3.1 Homepage – www.drughelp.care: Figure 25 is the screenshot of web application's homepage. It has two main buttons, one for drug abusers and another for agencies. As we can see it the design is clean and clear and to the point. It is not cluttered and it is not confusing.

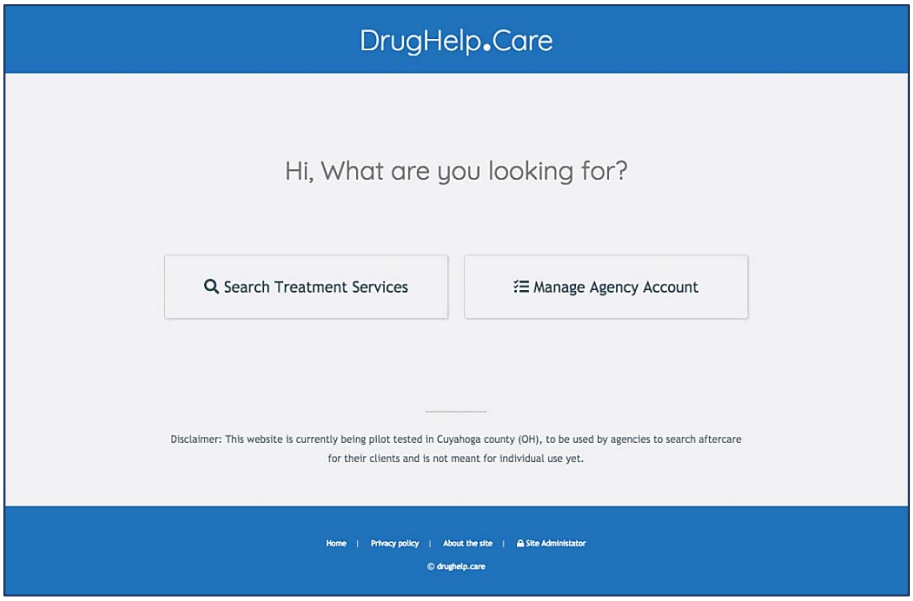


Figure 25: Homepage

4.3.2 Search Page: Search page is used by the drug abusers, their friends or family members. They can look for emergency care or treatment facilities in a timely manner with accurate and updated information. Figure 26 shows the screenshot of a search page, as we can see the space is intelligently managed by using accordion and collapsible cards and all the important buttons like call, web and map are highlighted and placed on top.

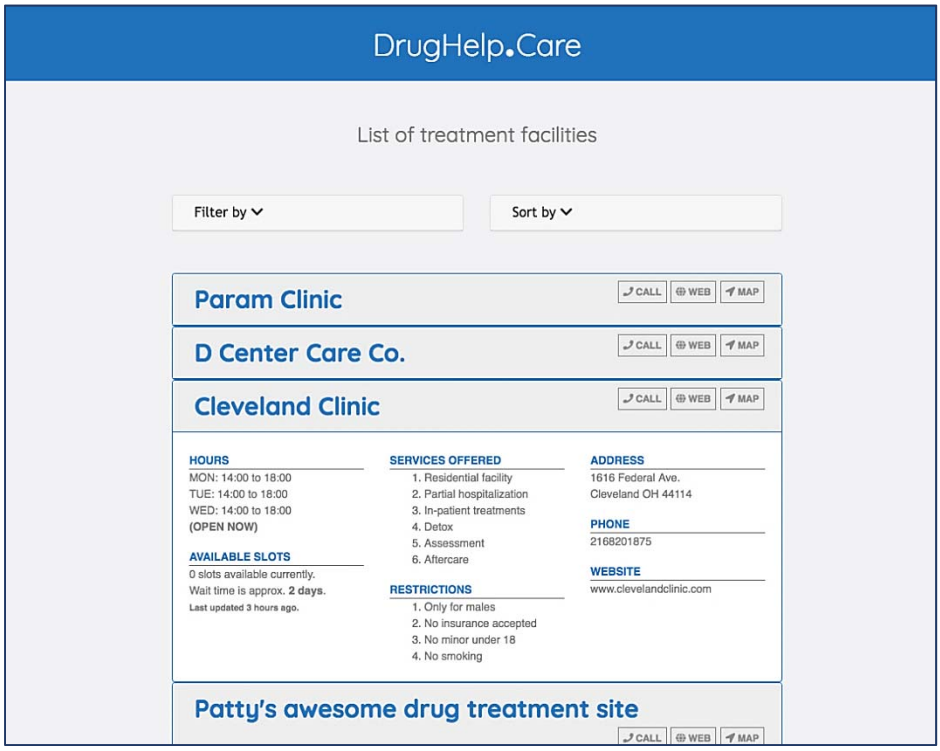


Figure 26: Search page

Figure 27 shows the screenshot of all the sorting and filtering options that a user can select from while looking for the facility.

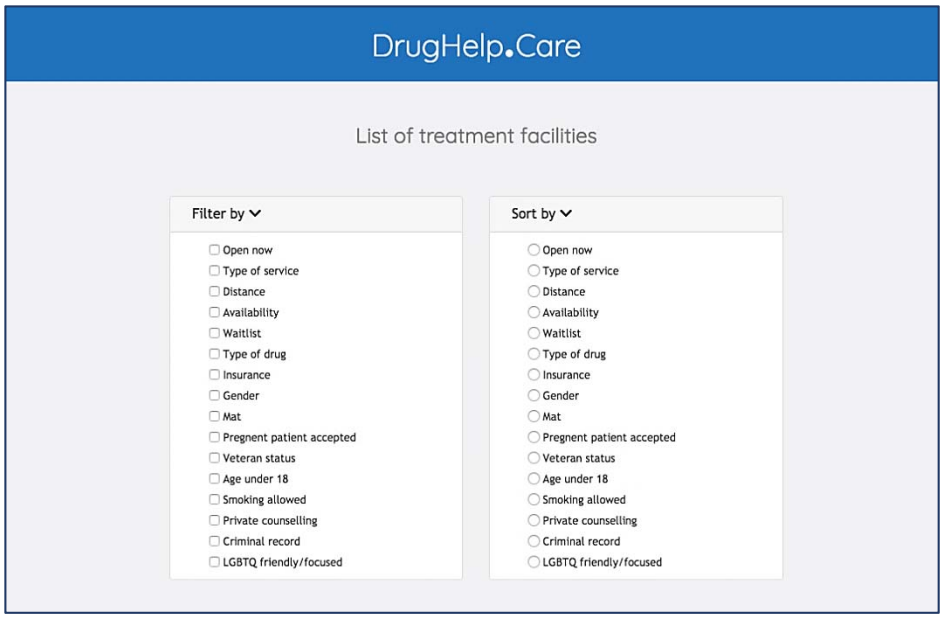


Figure 27: Search page's sorting and filtering options

4.3.3 Treatment facility – Account Signup: Agencies, also called as Treatment facilities will have an administrator who will sign-up by proving all the contact and location information including administrator's email id. Figure 28 shows the screenshot of the signup page for the Agencies.

Figure 29 is the screenshot of an email that is sent to administrator's inbox for the purpose of Email verification. This email will have a unique link generated by the backend.

Figure 30 is the page that opens when administrators click on the unique link. It shows a special checkbox, which ensures that the verification operation is being done by a human but not a software bot.

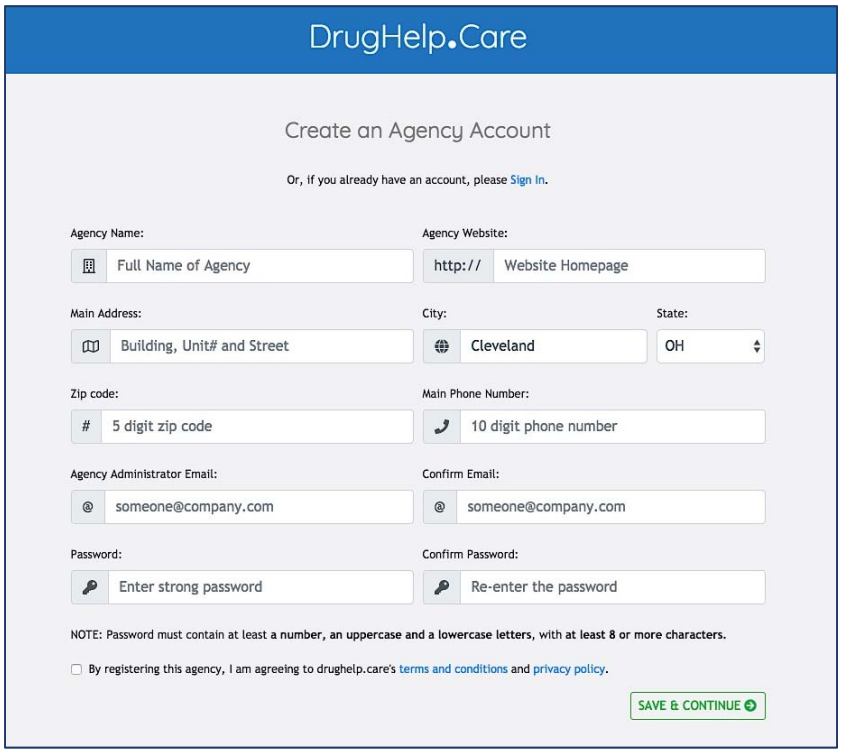


Figure 28: Agency Account Creation

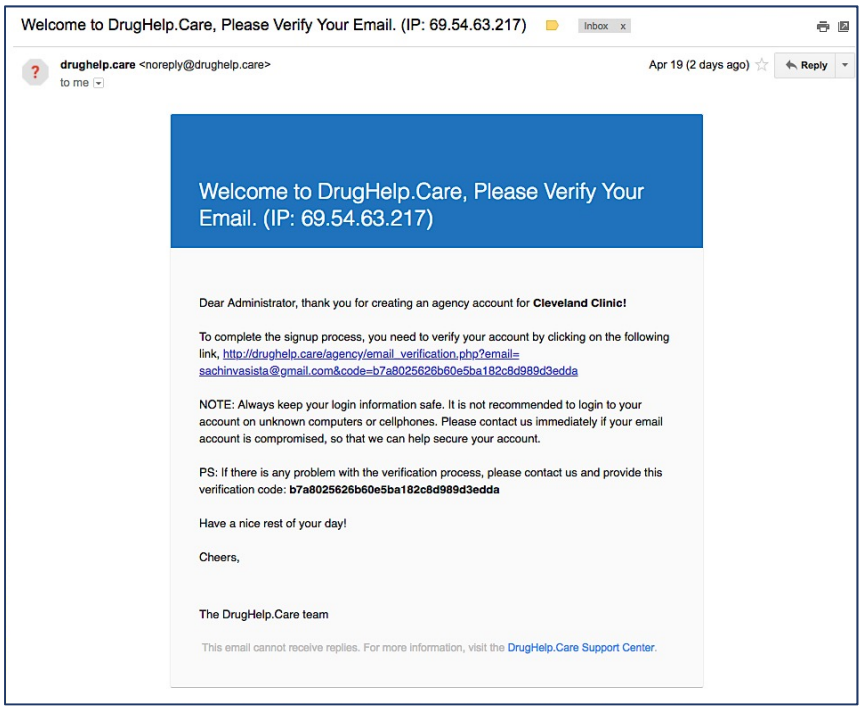


Figure 29: Agency Email Verification

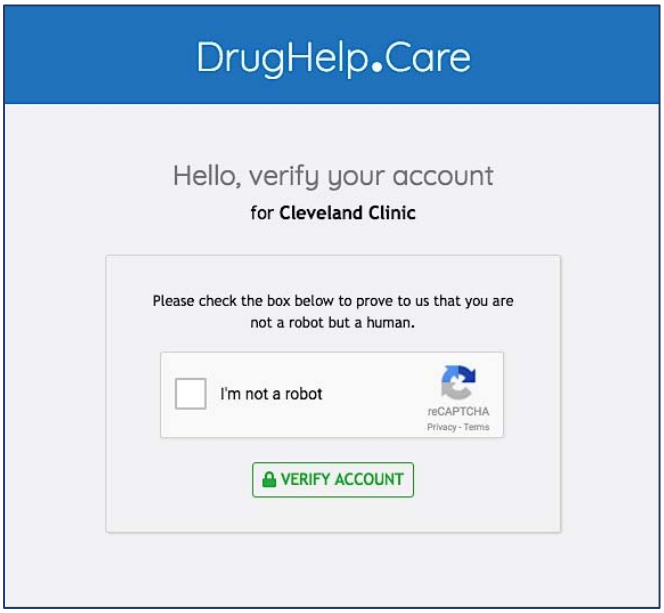


Figure 30: Agency Email Verification Action

4.3.3 Treatment facility – Account Sign-in and Password Reset: Figure 31 shows the screenshot of Agency's account sign in page. It shows a special checkbox, which ensures that the operation is being done by a human but not a software bot. The password is masked by bullets for privacy while the user typing it.

Figure 32 shows Reset Password page, where agency admins can request for a unique link to reset the password.

Figure 33 shows the email with a unique link to Reset the password, sent to agency admin's inbox.

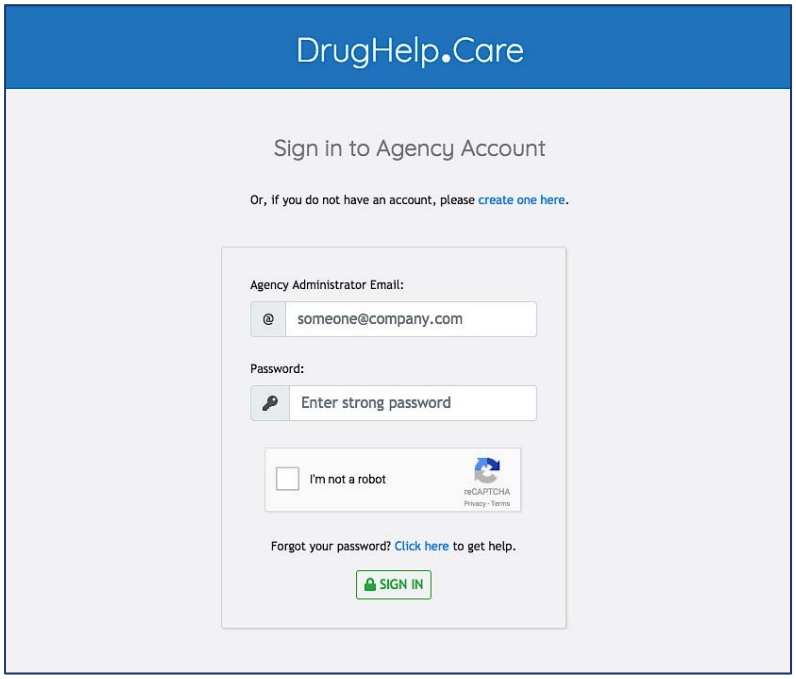


Figure 31: Agency Sign-in page

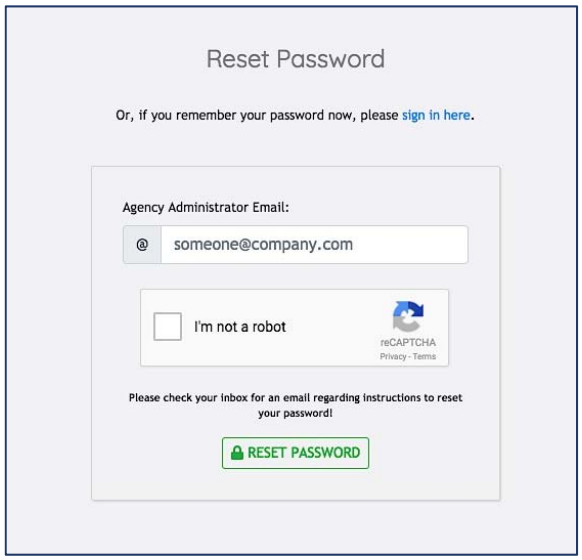


Figure 32: Agency account - Reset password page

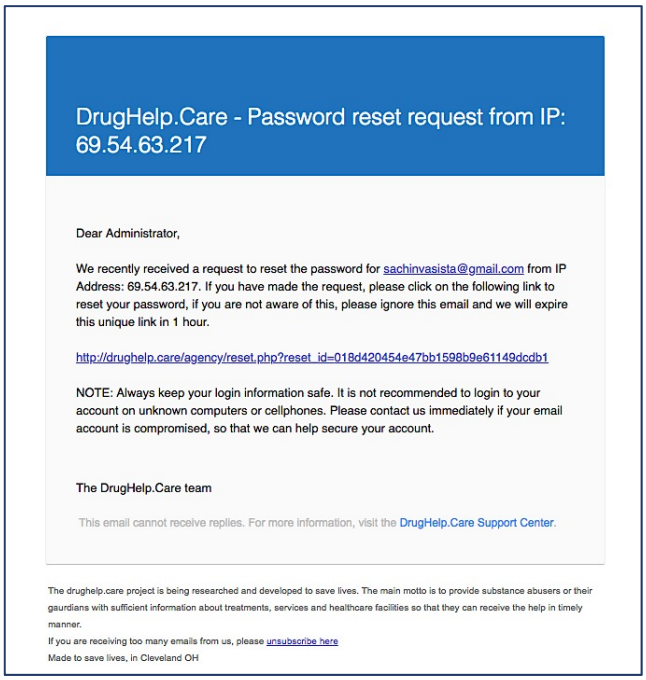


Figure 33: Agency reset password email

4.3.4 Treatment facility – Dashboard: Figure 34: Shows Agency's Dashboard, and it is simple and straight forward.

Figure 35: Shows the service registration page where admin can select one or more services and sub services and continue with registration.

Figure 36: Shows the service offering registration page where admin will add one or more offering for each service based on time or location.

Figure 37: Shows all the registered services with a data completion progress indicator for each of the services.

Figure 38: Shows a page where details can be entered for each service.

Figure 39: Shows a screenshot of how a modal form looks like in the process of data entry.

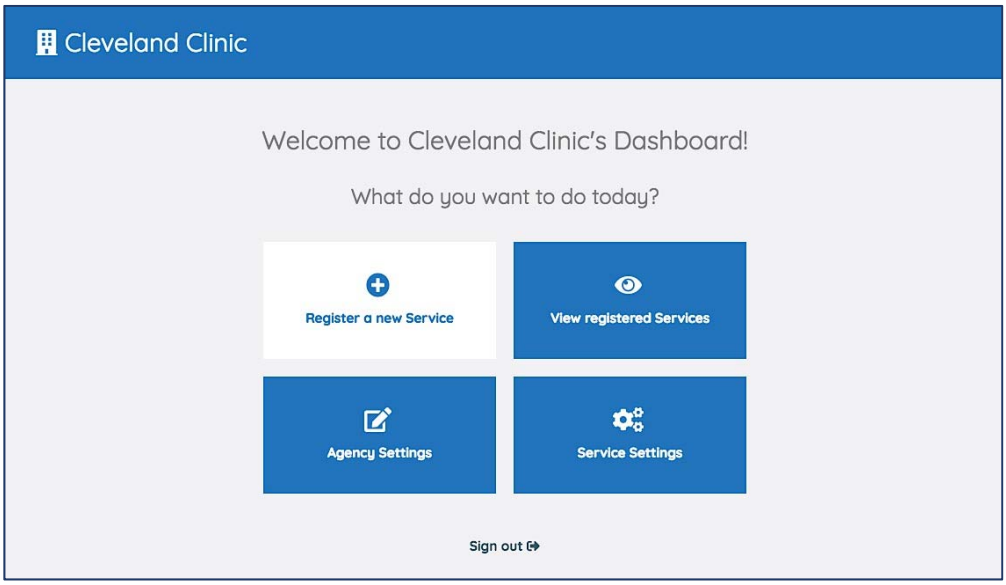


Figure 34: Agency's Dashboard

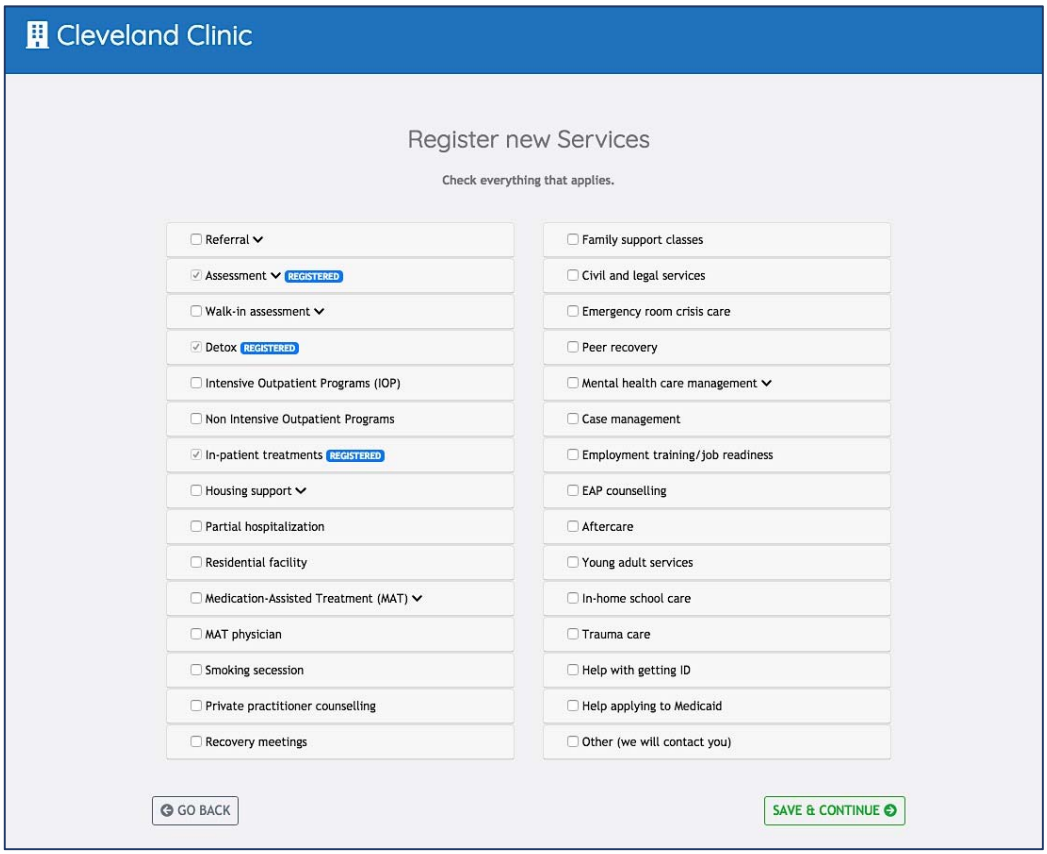


Figure 35: Service Registration page

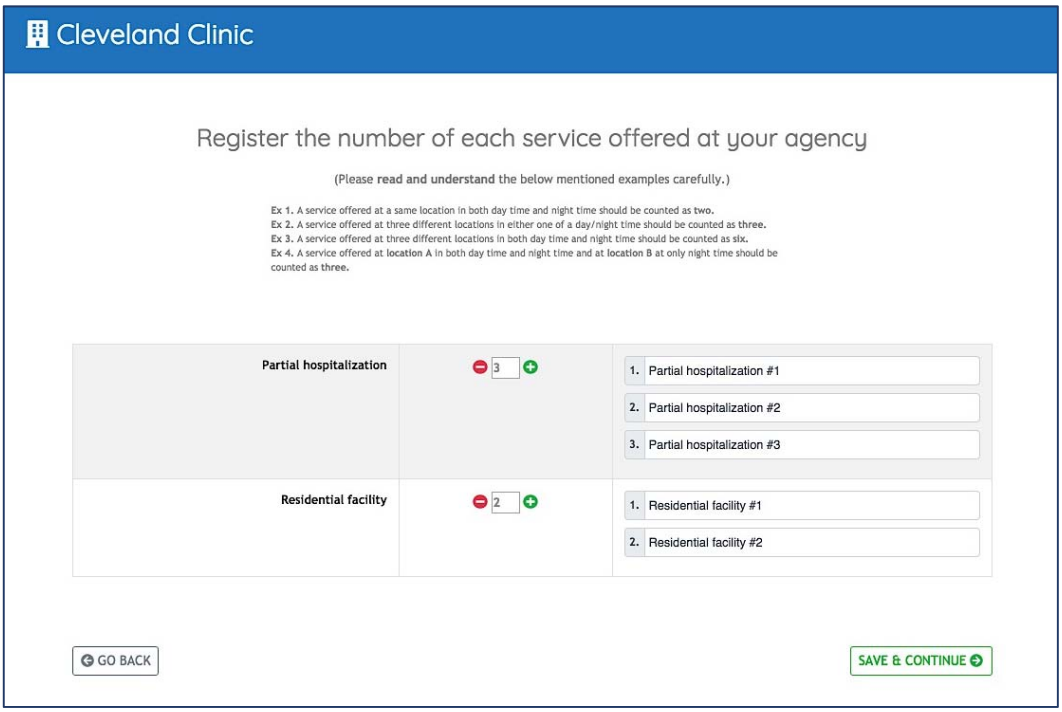


Figure 36: Service Offering Registration page

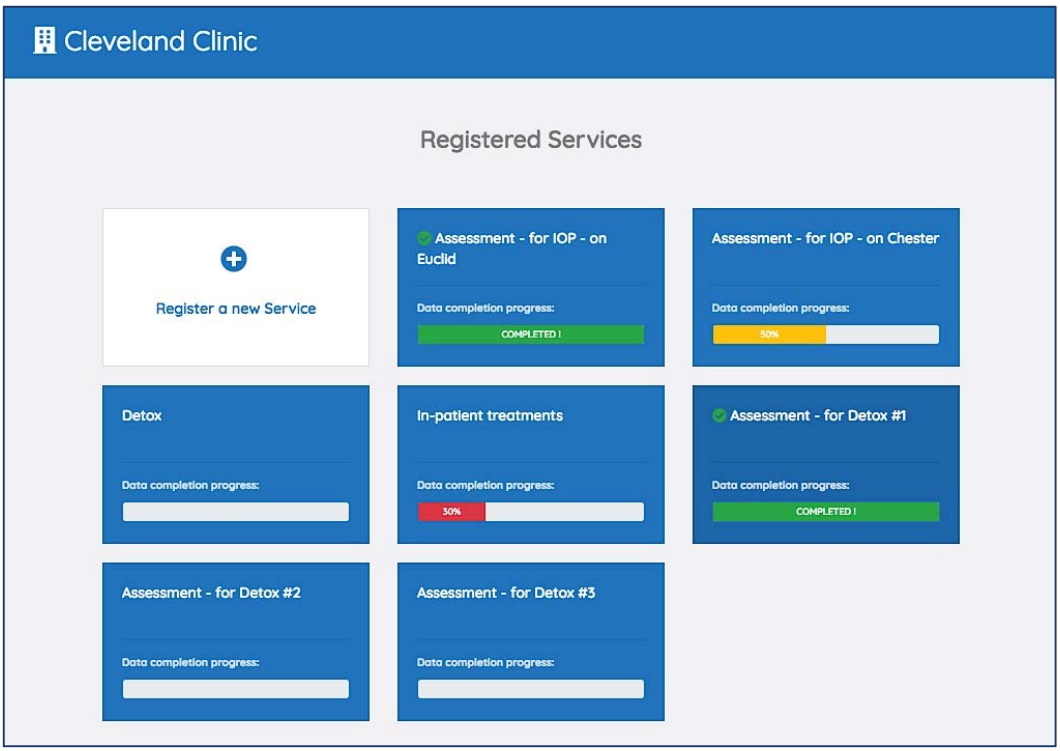


Figure 37: View Registered Services page

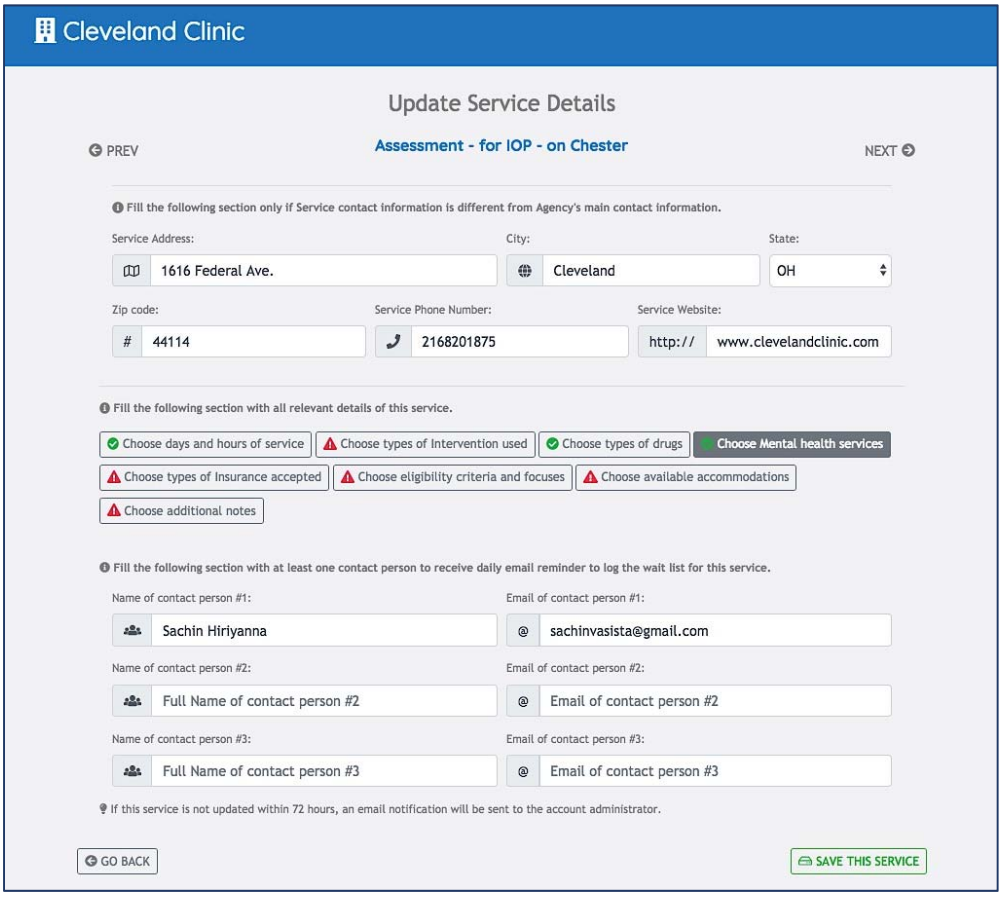


Figure 38: Service Details page

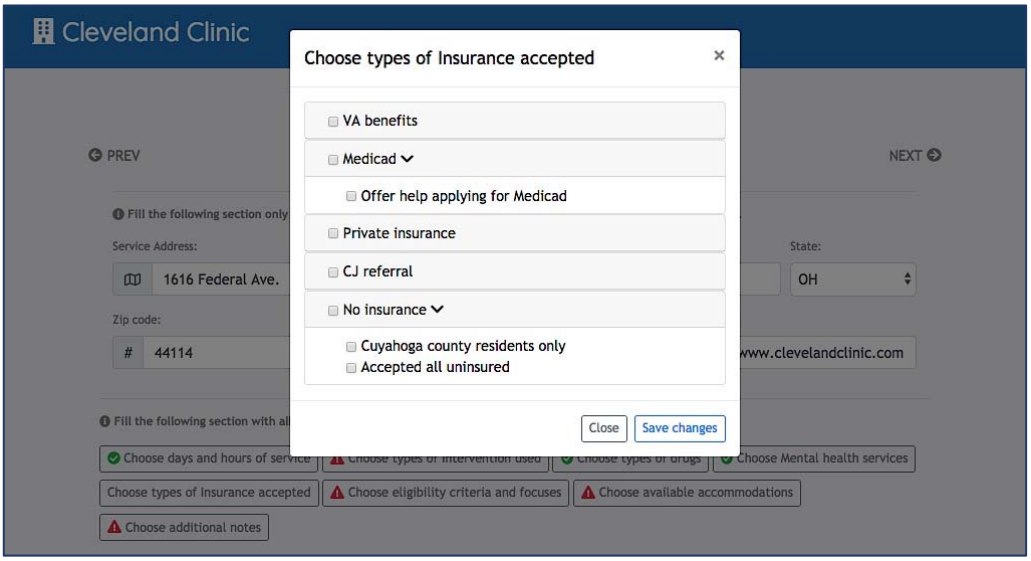


Figure 39: Service Details page - Example modal

4.3.5 Treatment facility – Contact Person: Figure 40: Shows a screenshot of an email that a contact person receives periodically to update the availability and waitlist of the assigned service.

Figure 41: Shows a page to update the availability of a service.

Figure 42: Shows a page to update the waitlist of a service.

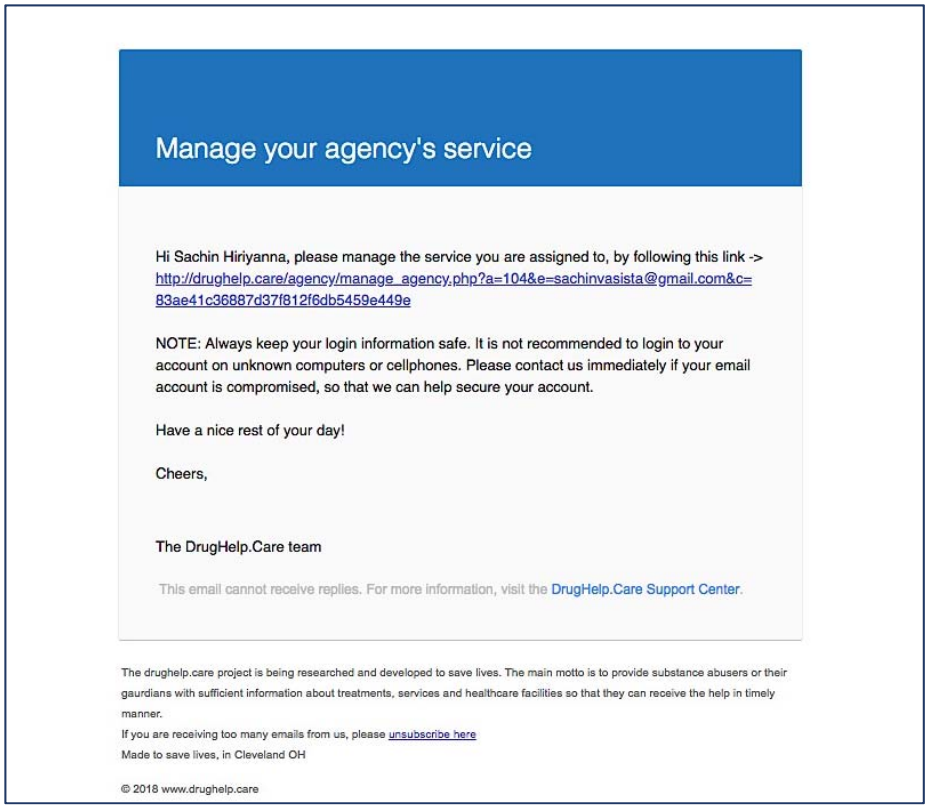


Figure 40: Contact Person email

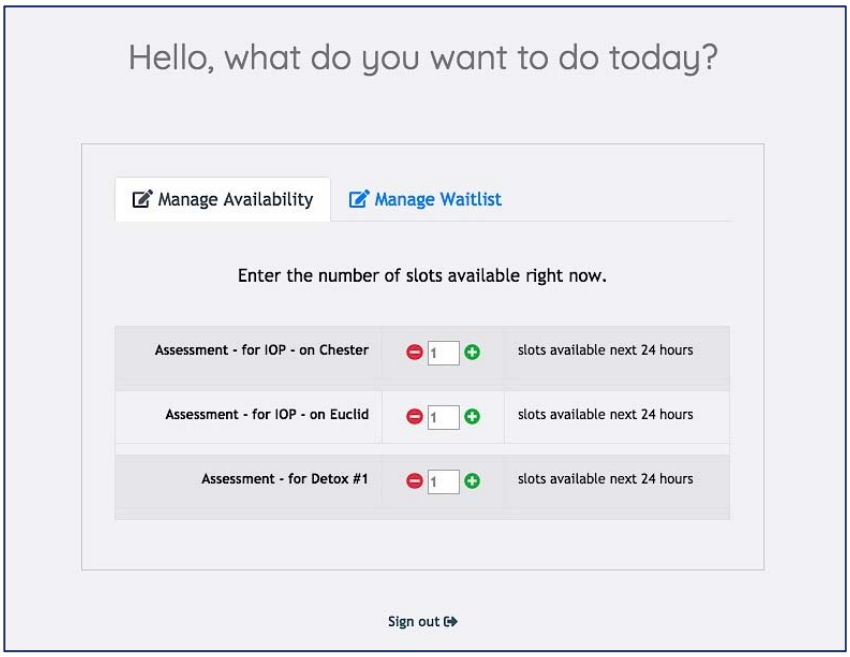


Figure 41: Contact Person - Availability page

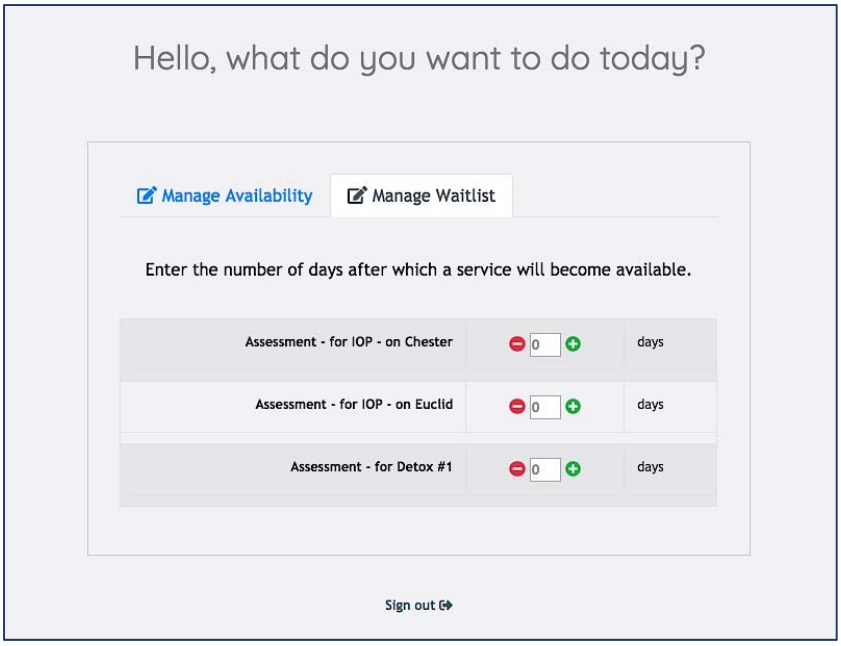


Figure 42: Contact Person - Waitlist page

CHAPTER V

CONCLUSION

In this thesis, we developed a web application that is used by drug abusers and treatment facilities. The major contributions of this thesis include:

1. We carried out a study on drug abusers and treatment facilities and understood the existing problems faced by each of them.
2. We listed the user and system requirements based on the existing problems.
3. We designed and planned the web application, including UI and database.
4. We developed the web application using Bootstrap CSS Framework and PHP in the backend with MySQL database and tested for the responsiveness on mobile devices. Figure 43 shows the actual screenshots of web application on mobile device.

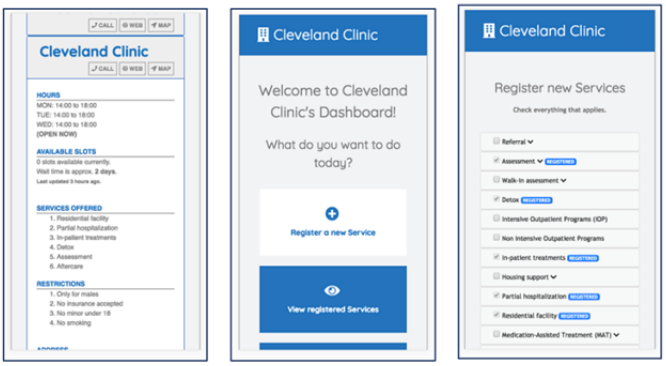


Figure 43: Screenshots of web application on Mobile phone

REFERENCES

[1] www.drugabuse.gov, "Overdose Death Rates," National Institutes of Helath (NIH), 09 2017. [Online]. Available: https://www.drugabuse.gov/related-topics/trends-statistics/overdose-death-rates. [Accessed 04 2018].

[2] P. K. G.-H. J. Ickin S., "Why Do Users Install and Delete Apps? A Survey Study," *Ojala A., Holmström Olsson H., Werder K. (eds) Software Business*, vol. 304, 2017.

[3] Bootstrap, "Introduction," Bootstrap, [Online]. Available: https://getbootstrap.com/docs/4.0/getting-started/introduction/. [Accessed 03 05 2018].

[4] Bootstrap, "Grid system," Bootstrap, [Online]. Available: https://getbootstrap.com/docs/4.1/layout/grid/. [Accessed 03 05 2018].

[5] Baltimora, "Basic concepts of flexbox," mozilla, 12 04 2018. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox. [Accessed 03 05 2018].

[6] "Browsers and devices," [Online]. Available: https://getbootstrap.com/docs/4.0/getting-started/browsers-devices/. [Accessed 03 05 2018].

[7] L. Wilson, "Emerald Jenny Foundation helps addicts, families connect with nearby resources," [Online]. Available: https://www.news5cleveland.com/news/e-team/emerald-jenny-foundation-helps-addicts-families-connect-with-nearby-resources. [Accessed 05 05 2018].

APPENDIX

IMPORTANT CODE SNIPPETS

1. AGENCY AUTHENTICATION PAGE

```
...

if ($Utilities -> captchaIsValid($CAPTCHA, $conn)) {
        if ($Utilities -> agencyExists($EMAIL, $conn)) {
            $RESPONSE = mysqli_query($conn, "SELECT * FROM AGENCY WHERE `EMAIL` =
'$EMAIL' AND `PASSWORD` = '$PASSWORD'");
            if (!$RESPONSE) {
                die('Error: ' . mysqli_error($conn));
                session_unset();
                session_destroy();
            } elseif (mysqli_num_rows($RESPONSE) > 0) {
                $RESPONSE = mysqli_query($conn, "SELECT * FROM AGENCY WHERE
`EMAIL` = '$EMAIL' AND `PASSWORD` = '$PASSWORD' AND `IS_VERIFIED` = 1;");
                if (!$RESPONSE) {
                    die('Error: ' . mysqli_error($conn));
                    session_unset();
                    session_destroy();
                } elseif (mysqli_num_rows($RESPONSE) > 0) {
                    $row = mysqli_fetch_assoc($RESPONSE);
                    $_SESSION['EMAIL'] = $EMAIL;
                    $_SESSION['ID'] = $row["ID"];
                    $_SESSION['LAST_ACTIVITY'] = time();
                    header('Location: dashboard.php');
                } else {
                    //Password doesn't match for the provided email, exit and
alert the user.
                    header('Location:
../agency/sign_in.php?alert=email_not_verified');
                    exit;
                }
            } else {
                //Password doesn't match for the provided email, exit and alert
the user.
                header('Location:
../agency/sign_in.php?alert=incorrect_password');
                exit;
            }
        } else {
            //Agency doesn't exist for the provided email, exit and alert the
user.
            header('Location: ../agency/sign_in.php?alert=agency_email_invalid');
            exit;
        }
    } else {
        //Captcha is not Valid, So exit.
        header('Location: ../agency/sign_in.php?alert=captcha_not_valid');
        exit;
    }

...
```

2. VIEW SERVICES PAGE WITH PROGRESS INDICATORS

```
...

$_SESSION['service_ids'] = array();
```

```
  $_SESSION['service_names'] = array();
  $Utilities = new Utilities();
    $sql = "SELECT *
    FROM  `REGISTERED_SERVICE_OFFERING`
    WHERE  `AGENCY_ID` =  ". $_SESSION['ID'] .
            " ORDER BY `REGISTERED_SERVICE_ID`";
    $result = $conn -> query($sql);
    if ($result->num_rows > 0) {
        $count = 0;

        while ($row = $result -> fetch_assoc()) {
            array_push($_SESSION['service_ids'], $row['ID']);
            array_push($_SESSION['service_names'], $row['NAME']);
            $progress = $Utilities->getDataCompletionProgress($row['ID'], $conn);
            $color = $progress <= 30 ? 'danger' : ($progress <= 60 ? 'warning' :
($progress < 100 ? 'info' : 'success'));
            $indicator = $progress == 100 ? '<i class="fas fa-check-circle text-
success"></i>' : '';
            $text = $progress == 100 ? 'COMPLETED !' : ($progress < 10? '' :
$progress."%");
            echo '
                <div class="col-md-4">
                  <div class="card animated zoomIn dark-blue-on-hover-service-
card" onclick="myhref(\'update_service_details.php?index='.$count.'&s_id=' .
$row['ID'] . '&s_name=' . $row['NAME'] . '\', false);">
                    <div class="card-body">
                      <div style="min-height:45px; font-size:1.2em"><p
class="card-title two-line">'.$indicator.' ' . $row['NAME'] . '</p></div>
                      <hr>
                      <div style="font-size:0.9em; font-weight:700; margin-
bottom:8px; opacity:0.8;">Data completion progress:</div>
                      <div class="progress" style="border: 1px solid #2173BD;
height: 23px;">
                        <div class="progress-bar bg-'.$color.'" role="progressbar"
style="font-size:0.8em; width: '.$progress.'%;" aria-valuenow="'.$progress.'"
aria-valuemin="0" aria-valuemax="100">'.$text.'</div>
                      </div>
                    </div>
                  </div>
                </div>
              ';
            $count++;
        }
    }
$conn->close();

...
```

3. GET COMPLETION PROGESS FUNTION

```
...

public function getDataCompletionProgress($s_id, $conn_)
        {
            $count = 0;
            $counts = mysqli_fetch_assoc(mysqli_query($conn_, "
                        SELECT
                        (SELECT Count(ID) FROM REGISTERED_SERVICE_DATASET WHERE
REGISTERED_SERVICE_ID = ".$s_id." AND ADDRESS IS NOT NULL) AS C1,
                        (SELECT Count(ID) FROM REGISTERED_SERVICE_DATASET WHERE
REGISTERED_SERVICE_ID = ".$s_id." AND DAYS_AND_HOURS IS NOT NULL) AS C2,
                        (SELECT Count(ID) FROM REGISTERED_SERVICE_DATASET WHERE
REGISTERED_SERVICE_ID = ".$s_id." AND TYPES_OF_INTERVENTION IS NOT NULL) AS C3,
                        (SELECT Count(ID) FROM REGISTERED_SERVICE_DATASET WHERE
REGISTERED_SERVICE_ID = ".$s_id." AND TYPES_OF_DRUGS IS NOT NULL) AS C4,
                        (SELECT Count(ID) FROM REGISTERED_SERVICE_DATASET WHERE
REGISTERED_SERVICE_ID = ".$s_id." AND MENTAL_HEALTH_SERVICES IS NOT NULL) AS C5,
                        (SELECT Count(ID) FROM REGISTERED_SERVICE_DATASET WHERE
REGISTERED_SERVICE_ID = ".$s_id." AND TYPES_OF_INSURANCE IS NOT NULL) AS C6,
```

```
                        (SELECT Count(ID) FROM REGISTERED_SERVICE_DATASET WHERE
REGISTERED_SERVICE_ID = ".$s_id." AND ELIGIBILITY_CRITERIA IS NOT NULL) AS C7,
                        (SELECT Count(ID) FROM REGISTERED_SERVICE_DATASET WHERE
REGISTERED_SERVICE_ID = ".$s_id." AND ACCOMMODATIONS IS NOT NULL) AS C8,
                        (SELECT Count(ID) FROM REGISTERED_SERVICE_DATASET WHERE
REGISTERED_SERVICE_ID = ".$s_id." AND ADDITIONAL_NOTES IS NOT NULL) AS C9,
                        (SELECT Count(ID) FROM REGISTERED_SERVICE_DATASET WHERE
REGISTERED_SERVICE_ID = ".$s_id." AND HAS_ATLEAST_ONE_CONTACT = 1) AS C10
                        FROM REGISTERED_SERVICE_DATASET"));

            $count = $counts['C1'] + $counts['C2'] + $counts['C3'] +
                      $counts['C4'] + $counts['C5'] + $counts['C6'] +
                        $counts['C7'] + $counts['C8'] + $counts['C9'] +
                          $counts['C10'];

            return (($count / 10) * 100); //There are 10 different coulumns to be
updated to 100% completion
        }

...
```

4. GET ALL SERVICES OF AGENCY FUNTION

```
...

public function getAllServicesOfAgency($ID, $conn_)
        {
            $SERVICES = array();
            $SQL = 'SELECT SERVICE_PARENT.NAME AS NAME
                    FROM REGISTERED_SERVICE
                    LEFT JOIN SERVICE_PARENT ON
REGISTERED_SERVICE.SERVICE_PARENT_ID = SERVICE_PARENT.ID
                    WHERE `AGENCY_ID` = '.$ID.' GROUP BY
REGISTERED_SERVICE.SERVICE_PARENT_ID ORDER BY SERVICE_PARENT.NAME DESC';
            $RESULT = $conn_->query($SQL);
            while ($SERVICE = $RESULT->fetch_assoc()) {
                array_push($SERVICES, $SERVICE);
            }
            return $SERVICES;
        }

...
```

5. GET ALL SERVICES FUNTION

```
...

public function getAllServicesToManage($AGENCY_ID, $EMAIL, $ACCESS_CODE, $conn_)
        {
            $SERVICES = array();
            $SQL = "SELECT REGISTERED_SERVICE_OFFERING.ID,
REGISTERED_SERVICE_OFFERING.NAME, REGISTERED_SERVICE_OFFERING.WAITLIST_COUNT,
            REGISTERED_SERVICE_OFFERING.AVAILABILITY_COUNT,
            REGISTERED_SERVICE_OFFERING.WAITLIST_INFO,
            REGISTERED_SERVICE_OFFERING.AVAILABILITY_INFO,
            REGISTERED_SERVICE_OFFERING.WAITLIST_LAST_UPDATED_INFO,
            REGISTERED_SERVICE_OFFERING.AVAILABILITY_LAST_UPDATED_INFO,
            REGISTERED_SERVICE_OFFERING.TIMESTAMP
            FROM `CONTACT_PERSON`
            INNER JOIN REGISTERED_SERVICE_OFFERING ON
CONTACT_PERSON.SERVICE_OFFERING_ID = REGISTERED_SERVICE_OFFERING.ID
            WHERE CONTACT_PERSON.AGENCY_ID = ".$AGENCY_ID." AND
CONTACT_PERSON.UNIQUE_ACCESS_CODE = '".$ACCESS_CODE."'";
              $RESULT = $conn_->query($SQL);
              while ($SERVICE = $RESULT->fetch_assoc()) {
                  array_push($SERVICES, $SERVICE);
```

```
            }
            return $SERVICES;
        }

...
```

6. PASSWORD RESET CLASS

```
class Password_Reset
{
    private $agencyID;
    private $agencyEmail;
    private $uniqueResetID;
    private $isActive;
    private $resetURL;
    private $conn_;

    public function __construct($c, $a, $e, $u, $i, $r)
    {
        $this->conn_ = $c;
        $this->agencyID = $a;
        $this->agencyEmail = $e;
        $this->uniqueResetID = $u;
        $this->isActive = $i;
        $this->resetURL = $r;
    }

    public function insert_data_to_table_and_send_email()
    {
        ...
    }

    private function sendEmail()
    {
       ...
    }

    public function delete_all_resetIDs()
    {
        ...
    }
}
```

7. PASSWORD RESET SEND EMAIL FUNCTION

```
...

private function sendEmail()
    {
        include("Mailer.php");

        //Compose Massage
        $body .= '<p style="text-align:left;">Dear Administrator,<br><br>We
recently received a request to reset the password for '.$this->agencyEmail.' from
IP Address: '.$_SERVER['REMOTE_ADDR'].'. If you have made the request, please
click on the following link to reset your password, if you are not aware of this,
please ignore this email and we will expire this unique link in 1 hour.';
        $body .= '<br><br><a href="'.$this->resetURL.'">'.$this-
>resetURL.'</a><br><br>
        NOTE: Always keep your login information safe. It is not recommended to
login to your account on unknown computers or cellphones. Please contact us
immediately if your email account is compromised, so that we can help secure your
account.</p>';

        //Send Mail
        $Mailer = new Mailer('noreply@drughelp.care', $this->agencyEmail,
'DrugHelp.Care - Password reset request from IP: '.$_SERVER['REMOTE_ADDR'],
$body);
```

```
        return $Mailer->send();
    }

...
```

8. PASSWORD RESET SEND EMAIL FUNCTION

```
...

//type = warning | success | danger | info | light | dark
private function showAlert($type, $head, $body)
        {
            return '
                <center>
                  <div class="alert alert-'.$type.' alert-dismissible fade show
animated flipInX" role="alert" style="max-width:770px; border: 1px #ccc solid;
text-align: left;">
                    <button type="button" class="close" data-dismiss="alert" aria-
label="Close">
                      <span aria-hidden="true">&times;</span>
                    </button>
                    <strong>'.$head.'</strong> '.$body.'
                  </div>
                </center>
            ';
        }

...
```

9. EMAIL SENDING FUNCTION

```
...

try {

$this->mail->addCustomHeader('MIME-Version: 1.0');
$this->mail->addCustomHeader('Content-type: text/html; charset=iso-8859-1');
$this->mail->addCustomHeader('X-Priority: 3');
$this->mail->addCustomHeader('Return-Path:
drughelp.care'.'<noreply@drughelp.care>');
$this->mail->addCustomHeader('Organization: Cleveland State University');
$this->mail->addCustomHeader('X-Mailer: PHP/' . phpversion());
$this->mail->addReplyTo('noreply@drughelp.care', 'drughelp.care');
$this->mail->setFrom($this->from, 'drughelp.care');
$this->mail->isHTML(true);
$this->mail->addAddress($this->to);
$this->mail->Subject = $this->subject;
$this->mail->Body = $this->body;
$this->mail->send();
            return true;
     } catch (Exception $e) {
            echo 'Message could not be sent. Mailer Error: ', $this->mail-
>ErrorInfo;
            return false;
     }


...
```

10. AGENCY SIGNUP PAGE

```
...

$EMAIL = strtolower(mysqli_real_escape_string($conn, $_POST['email_1']));
    $EMAIL_2 = strtolower(mysqli_real_escape_string($conn, $_POST['email_2']));
```

```
    if (strcmp($EMAIL, $EMAIL_2) != 0) {
        mysqli_close($conn);
        header('Location: ../agency/index.php?alert=emails_dont_match');
        exit;
    }

    $PASSWORD = md5($_POST['password_1']);
    $PASSWORD_2 = md5($_POST['password_2']);

    if (strcmp($PASSWORD, $PASSWORD_2) != 0) {
        mysqli_close($conn);
        header('Location: ../agency/index.php?alert=passwords_dont_match');
        exit;
    }

    if ($Utilities->agencyExists($EMAIL, $conn)) {
        mysqli_close($conn);
        header('Location: ../agency/index.php?alert=email_already_exists');
        exit;
    }

    $NAME = mysqli_real_escape_string($conn, $_POST['agency']);
    $WEBSITE = mysqli_real_escape_string($conn, $_POST['website']);
    $ADDRESS = mysqli_real_escape_string($conn, $_POST['address']);
    $CITY = mysqli_real_escape_string($conn, $_POST['city']);
    $STATE = mysqli_real_escape_string($conn, $_POST['state']);
    $ZIP = mysqli_real_escape_string($conn, $_POST['zip']);
    $PHONE = mysqli_real_escape_string($conn, $_POST['phone']);


    $VERIFICATION_CODE = $Utilities->generateUniqueID($EMAIL);

        $sql = "INSERT INTO  `drughelp_care`.`AGENCY` (
                              `TIMESTAMP` ,
                              `ID` ,
                              `NAME` ,
                              `WEBSITE` ,
                              `ADDRESS` ,
                              `CITY` ,
                              `STATE` ,
                              `ZIP` ,
                              `PHONE` ,
                              `EMAIL` ,
                              `PASSWORD` ,
                              `VERIFICATION_CODE` ,
                              `IS_VERIFIED`
                              )
                              VALUES (
                              CURRENT_TIMESTAMP , NULL ,  '$NAME',  '$WEBSITE',
        '$ADDRESS',  '$CITY',  '$STATE',  $ZIP,  $PHONE,  '$EMAIL', '$PASSWORD' ,
        '$VERIFICATION_CODE', 0
                              );";

        if ($conn -> query($sql) === true) {
                mysqli_close($conn);
        }

...
```