

Parakeet Social Support

Registry Application

Final Report

SDDEC22-08

Client: Dr. Katherine Rafferty

Advisor: Professor Matthew Wymore

Team Members:

Samantha Culver: Team Manager & Front-end Designer

Kellan Hulet: Front-end Developer, Front-end Manager

Siddharth Rana: Full Stack Developer, Website Manager

(Brayden Ruch: Back-end Developer, Client Correspondent

Amaranthia Willers: Back-end Developer, Back-end manager)

Email: sddec22-08@iastate.edu

Team Website: <http://sddec22-08.sd.ece.iastate.edu/>

User's Manual / Basic Functionality Guide	2
Introduction to Project	6
Project Plan	8
Design	15

User's Manual

CREATING AN ACCOUNT

1. Navigate to parakeet.care
2. Click the tab labeled "Login" in the top right of the navigation header
3. Select the "Create Account" tab

The screenshot shows the Parakeet website's account creation interface. At the top, a dark blue navigation bar contains the Parakeet logo on the left and links for 'About', 'Search', 'Account', 'Messages', and 'Login' on the right. Below the navigation bar, there are two tabs: 'Sign In' and 'Create Account'. The 'Create Account' tab is active and highlighted. An arrow labeled 'Step 2' points to the 'Login' link in the navigation bar, and another arrow labeled 'Step 3' points to the 'Create Account' tab. The 'Create Account' form includes the following fields: 'Email', 'Password' (with a visibility toggle), 'Confirm Password' (with a visibility toggle), 'Given Name', and 'Family Name'. A teal 'Create Account' button is positioned at the bottom of the form.

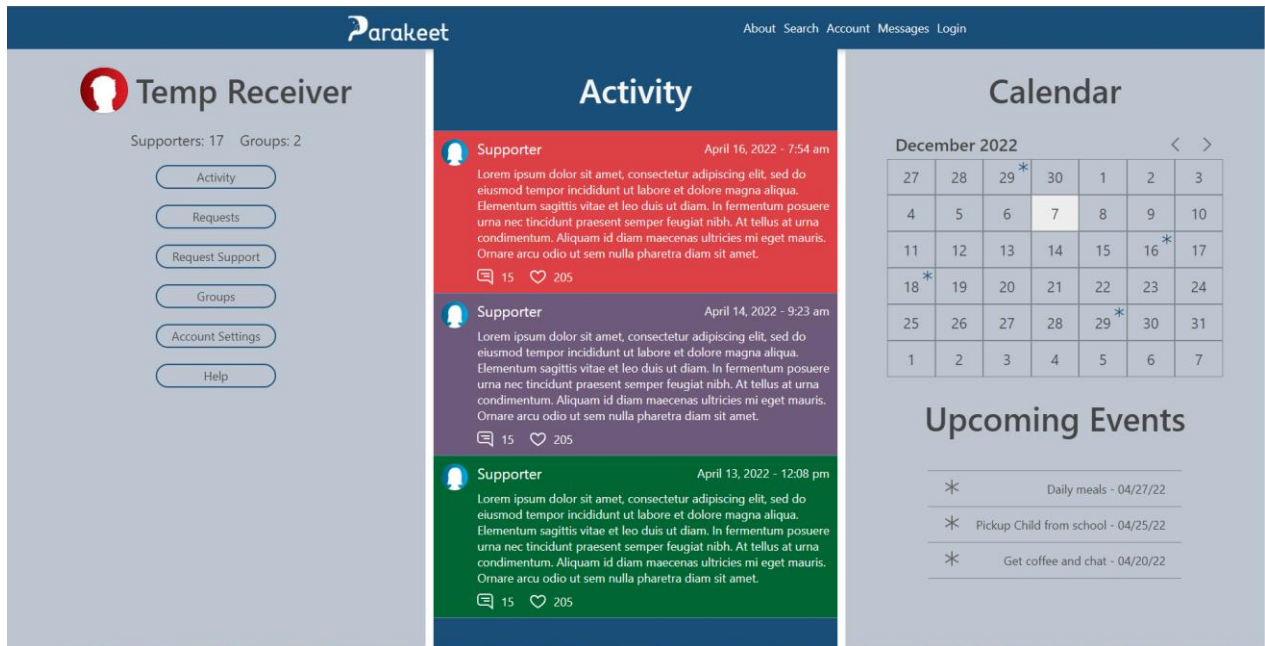
4. Fill in required information

The screenshot shows a verification page titled 'We Emailed You'. The text reads: 'Your code is on the way. To log in, enter the code we emailed to k***@g***. It may take a minute to arrive.' Below this text, there is a text input field with the placeholder 'Enter your code'. Underneath the input field is a teal button labeled 'Confirm'. At the bottom of the form is another text input field with the placeholder 'Resend Code'.

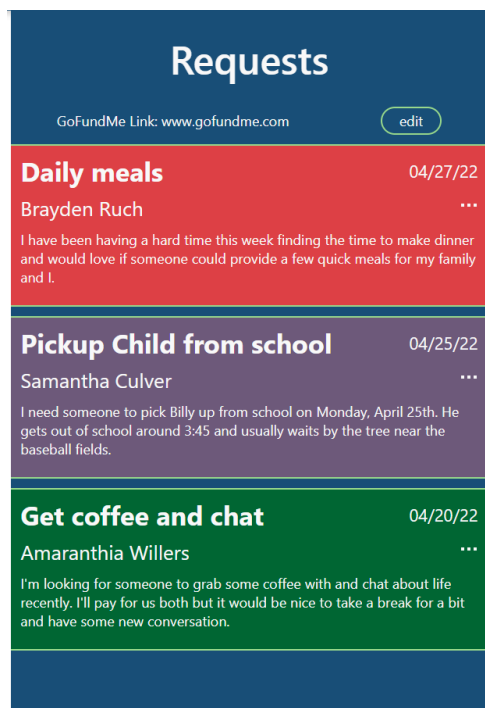
5. Navigate to given email to retrieve six-digit authentication code
6. Email is now verified, and account is created
 - a. User will be automatically moved to account page after successful account creation

NAVIGATING ACCOUNT PAGE

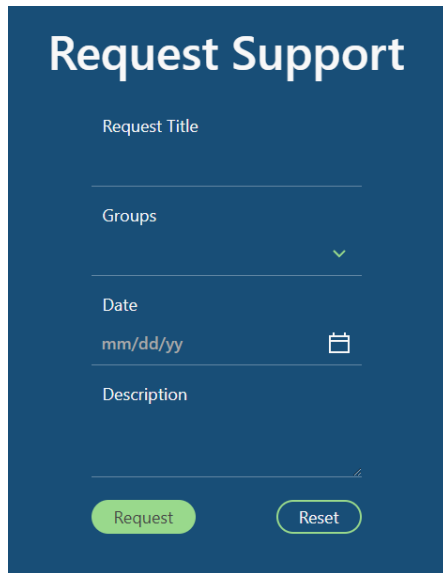
1. Activity page. The default screen for the account page. The middle section will change depending on which page is selected from the left side menu.



2. Request page. Users are able to view active and completed requests they have submitted.



3. Request Support page. Users are able to fill out the form to request support. Request is only available to users within selected group.



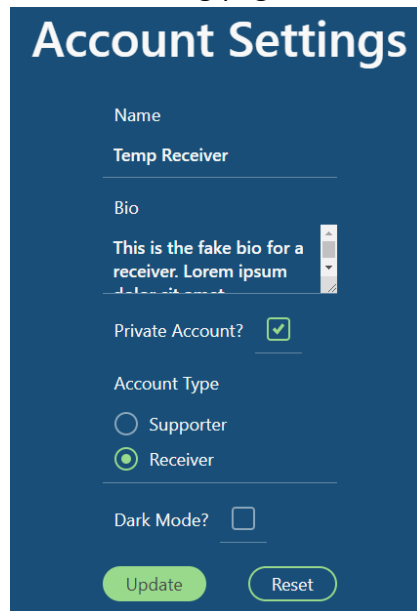
The image shows a 'Request Support' form on a dark blue background. The form has a title 'Request Support' at the top. Below the title are four input fields: 'Request Title', 'Groups' (with a dropdown arrow), 'Date' (with a calendar icon and placeholder 'mm/dd/yy'), and 'Description'. At the bottom of the form are two buttons: 'Request' (highlighted in light green) and 'Reset'.

4. Groups page. Displays all groups a user has created and contains the ability to edit users within the group.



The image shows a 'Groups' page on a dark blue background. The title 'Groups' is at the top. Below the title are three group cards, each with a title, a list of members, and an 'Edit' button. The first card is titled 'Friends' and lists 'Amaranthia Willers', 'Samantha Culver', and 'Brayden Ruch'. The second card is titled 'Work' and lists 'Work friend 1' and 'Work friend 2'. The third card is titled 'Community' and lists 'Community friend 1', 'Community friend 2', and 'Community friend 3'.

5. Account Setting page. Users are able to change settings within their account.



MESSAGING

1. Login to account / create account if none exist
2. Navigate to messages from navigation header
3. Select user to message, type message, and hit send



Introduction

PROBLEM STATEMENT

There are many individuals who face medical crises daily. Frequently, it is quite difficult for these people to get the support they need from their community due to societal pressure, the stress of the crisis causing them to forget who offered support, and a variety of other reasons as well. With the development of the Social Support Registry web application parakeet.care, team 8 is trying to make it easier for individuals to get the types of support they need from their community in an organized and simple manner.

REQUIREMENTS & CONSTRAINTS

- Functional requirements:
 - Request support
 - Fill requests for support
 - Bio/About to tell people about their needs
 - Search feature
 - Create an account
 - Public and Private account options
 - Access a calendar
 - Messaging feature
- Resource requirements: Server to run the frontend and backend of application once launched
 - Total budget is \$1000 (constraint)
- Aesthetics requirement: The app will have a blue palette for the theme. The goal is to have a calm and relaxing feel.
- Economic/market requirements: There are a few social support applications but none that aim to organize supporters.
- Time requirement: Completing as much of the development for the project as possible before the end of the second semester.

ENGINEERING STANDARDS

802E-2020: Privacy Considerations-how to mitigate privacy threats

828-2012: Configuration Management minimum standard requirements

829-Software Testing Documentation Standard

802.15.8-Peer to peer communication standard for chat feature

830-1998: Software Requirement Specifications Documentation for passing the work on to other maintainers/developers after our part of the project ends

X.509-SSL-defining the format of public key certificates

23026-2015: The standard for the life cycle of websites

7002: Data Privacy Standard for personal information

INTENDED USERS AND USES

- 1) People who are in health crises will benefit because they can more easily organize their needs for their fellow community members to offer help for
- 2) The community around people in need cares about the existence of the website because they can use it to offer aid in the ways that they are most able to help their friends and family in a crisis
- 3) Healthcare providers appreciate the existence of Support Registries because when individuals facing a health crisis get emotional and tangible support for loved ones it helps ease their stresses and improves the chances of a positive recovery.

Project Plan

PROJECT MANAGEMENT/TRACKING PROCEDURES

Team SDDEC22-08 is adopting the agile project management style. This style is demonstrated by our frequent communication with our client, the team's efforts to make the project easy to adjust should new goals arise later in the development process, and how thus far our team goals have been fluctuating between development goals and design goals for the project quite fluidly. To track the team progress on the project, we are using Gitlab, Jiro, and Discord to maintain progress and team communication.

TASK DECOMPOSITION

For the project, the overall objective is to design a website that allows users to communicate their needs to their community and have their community sign up for tasks to help users in need with their requests. Breaking this website down is rather simple.

For the Back-end development:

1. Database for User Information
 - a. Registration
 - b. Login
 - c. Profile
 - d. Privacy Settings
 - e. Need Support/Giving Support
2. Creating the Calendar feature for each user
 - a. Requires being able to pull from support jobs, calendars of other users/communities, and have notifications of scheduling conflicts
3. Chatting between users for specific tasks will mainly be a backend problem, since the front end really only needs to send the correct requests to the server for the messages to be displayed/sent
 - a. Task Chats
 - b. Community Chats
 - c. Public Chats
4. Relations between Users
 - a. Following
 - b. Friends/privacy settings
 - c. Communities

d. Searching

For the Front-end development:

1. We need to first create the main page from which every other page can be reached:
 - a. This will be the home page. It will have the option to link to a sign-in page or a registration page for new users. It will also allow users to go to the 'about' page to read about the purpose of the website.
2. Then, a sign-in page should be created, with communication set up with the server to validate a user's name and password.
 - a. This page should send a request to the server to verify login information. If invalid login, it should notify the user that either they have an incorrect username or password entered.
 - b. IFF the username and password are both valid, the website should be approved to link to that user's home page.
3. From there, the about page should be developed.
 - a. This page will mostly detail the purpose of the website and site resources for the research behind the reason for the website's development. It will also provide a summary of the developers who worked on the website. Since this page will be mostly text, it will be one of the easier pages.
 - b. This page should have the ability to link back to the homepage (either the non-signed in page or the user's personal homepage if the user was logged in at the time the about page was viewed)
4. User's homepage (similar to 1)
 - a. This homepage should provide the user with the option to logout, the ability to look at their user profile
 - b. User should be able to interact with their messages
 - c. They should be able to access their calendar
 - d. And users can search for other users from this page
5. Registration page
 - a. Users should be able to mark whether they are in-need or helper user
 - b. Users should be able to rank their support needs if they marked in-need
 - c. Users should be able to Provide a summary of their situation along with any photos they'd like to share
 - d. Users should have the option to specify their general location (by zip-code)
 - e. Users should be able to setup email or phone notifications for when someone messages them or add things to their calendar or signs up for support tasks
 - f. Users should be able to add support needs if they aren't provided as one of the preset suggestions: they should also be able to give a detailed description of what their needs would entail (whether a preset need or an added need)
 - g. Users should be able to specify days of the week where some needs are more important than others.

- h. Users should be able to set a username and password for login purposes (as well as a recovery email)
 - i. Users should be able to set their account to either public, private (only access user page if provided a link), or hidden (hidden = only able to find if searched for by specific name)
- 6. Profile page
 - a. Profile should contain most of the information provided from registration
 - b. Profile should provide the option to modify any and all information within it
 - c. Profile page should offer a shareable link for users to send to loved ones and people in their community to make it easy for them to find their page.
- 7. Calendar page
 - a. Should allow user to modify tasks on their calendar
 - b. Should be able to send user reminders for upcoming tasks on their calendar (either phone or email reminders or both)
- 8. Search page
 - a. Can search by usernames to find specific people you're looking for
 - b. Can copy user link from the search result to share with own friends/family
 - c. Can Click on user to take to specific user's page for more details
 - d. Can send user a message from the search page
 - e. Can request friend status from specific user to do higher-trust support tasks
- 9. Messaging page
 - a. Can view all conversations had with other users
 - b. Sorted by inbox and outbox
 - c. Can open specific message chain to continue conversations with users
- 10. Other users' page
 - a. Should allow user to view other users' needs
 - b. User should be able to sign up for needs requested by another user, which would add that task to user's own calendar
 - c. Can read user profile description to understand their story
 - d. Can send user a message to start a message chain
 - e. Can request friend status to do higher-trust support tasks

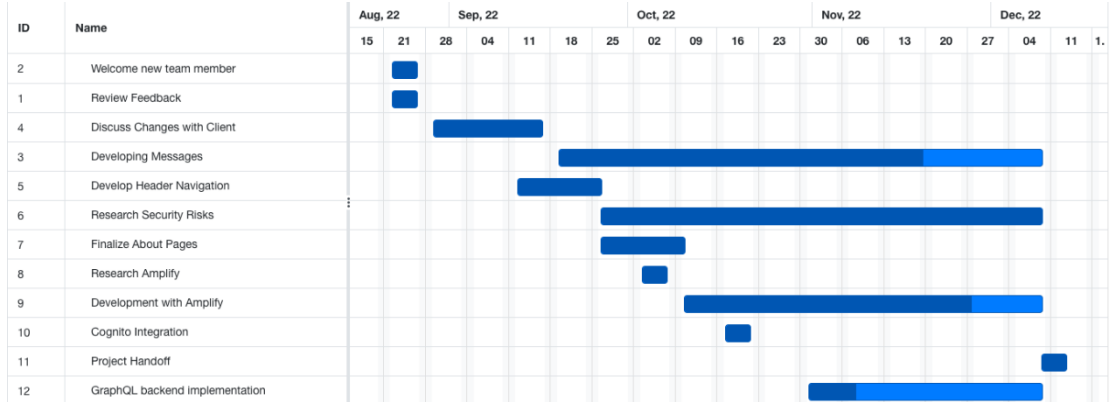
PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

- 1) It is necessary to first create the main page from which every other page can be reached:
 - a. Welcoming to new users
 - b. Each to navigate
 - c. Efficiently designed to allow users to navigate to their profile or sign up

- 2) Then, a sign-in page should be created, with communication set up with the server to validate a user's name and password.
 - a. Secure sign in
 - b. Allow browser to cache user data for easier sign in
 - c. Optionally keep users signed in on given device
- 3) From there, the about page should be developed.
 - a. Show developer bios
 - b. Explain the idea behind the website
 - c. Show relevant research to support website
- 4) User's homepage (similar to 1)
 - a. Dependent on type of user, show information relevant to them
 - i. ie: if supporter, show upcoming obligations/status of accounts user supports
 - ii. ie: if user is receiving support, easily ask for more or check status of requested support
 - b. Link to calendar and account settings are easily accessible
- 5) Registration page
 - a. User is able to easily sign up for an account
 - b. Secure registration
 - c. Auto sign in after?
- 6) Profile page
 - a. Show information about the user that is readable but also looks good
 - b. Able to follow user or request to follow
 - c. Able to message user or request to message
- 7) Calendar page
 - a. Display all time sensitive support in an easy to view calendar
 - b. Could be linked with Google Calendar
- 8) Search page
 - a. Search for users or communities
 - b. Filter by type of user or by text input
 - c. Results are easily filtered and viewable
- 9) Messaging page
 - a. Receive messages from users
 - b. Show messages from newest to oldest
- 10) General front end
 - a. Able to communicate with the backend quickly and display the loading state to the user.
 - b. Handle invalid inputs or bugs without app crashing
 - c. Overall aesthetic is calming and looks professional
 - d. Users can navigate the app easily and intuitively
- 11) Communities

- a. Creating A Community
 - b. Community Support Tasks
 - c. Community Chats
- 12) Mobile App Portability
- a. Format Web Application so that it will work as a mobile application

PROJECT TIMELINE/SCHEDULE



RISKS AND RISK MANAGEMENT/MITIGATION

Risks	Category	Probability	Impact	RMM (number)
Personal information leaks	cybersecurity	.2	High	001
Lack of Support Response	market	0.5	High	002

RMMS:

Risk Information Sheet			
Risk ID: 002	Date: 03/13/2022	Probability: 0.5	Impact: High

Description: Lack of response from support networks of users	
Refinement/Context: may not know about application, may not check the application often enough, there may not be enough people that can provide the needed support using the application to keep track of the support needs of people	
Mitigation/Monitoring: check with users for feedback, make a way to spread from different users to their support networks, make incentives for checking in more frequently(?)	
Management/Contingency Plan/Trigger: <ol style="list-style-type: none"> 1. Advertise to user base 2. Notifications/reminders for users 3. Consistent Usability of product 4. Feedback from Users 	
Current Status: monitoring, planning, and feedback	
Originator: Amy Willers	Assigned: all

PERSONEL EFFORT REQUIREMENTS

Front-end Development time estimates for project by task:

Tasks	Estimated Time to complete (in hours)
1) Initial homepage	4
2) Sign-in page	9
3) About page	2
4) User homepage	15
5) Registration page	40
6) Profile page	15
7) Calendar page	25
8) Search page	5

9) Messaging page	5
10) Other user page	5
Total time for Front-end Development:	125 hours

Back-end Development time estimates for project by task:

Tasks	Estimated Time to complete (in hours)
User Information	15
Relations between Users	30
Calendar Information	10
Chat Features	50
Communities	15
Total time for Back-end Development:	120

OTHER RESOURCE REQUIREMENTS

Design of the Logo and selecting a color scheme are the primary other requirements.

Design

DESIGN CONTEXT

This product is being developed to assist communities with individuals that are experiencing a health crisis and need several types of support from their communities. These communities are impacted by our design, and our project addresses the societal need for effective ways to provide support to individuals in need of assistance.

Area	Description	Examples	Relevant Considerations
Public health, safety, and welfare	How does your project affect the general well-being of various stakeholder groups? These groups may be direct users or may be indirectly affected (e.g., solution is implemented in their communities)	Increasing/reducing exposure to pollutants and other harmful substances, increasing/reducing safety risks, increasing/reducing job opportunities	Increasing accessibility to emotional, physical, and other types of support for people who are in a health crisis.
Global, cultural, and social	How well does your project reflect the values, practices, and aims of the cultural groups it affects? Groups may include but are not limited to specific communities, nations, professions, workplaces, and ethnic cultures.	Development or operation of the solution would violate a profession's code of ethics, implementation of the solution would require an undesired change in community practices	Development of the solution would allow the groups it impacts to better provide the support to those in need that they want to, helping them to support their communities.

Environmental	What environmental impact might your project have? This can include indirect effects, such as deforestation or unsustainable practices related to materials manufacture or procurement.	Increasing/decreasing energy usage from nonrenewable sources, increasing/decreasing usage/production of non-recyclable materials	Increasing carpooling to school/medical appointments, minor increase in usage of non-recyclable materials used for servers/computers
Economic	What economic impact might your project have? This can include the financial viability of your product within your team or company, cost to consumers, or broader economic effects on communities, markets, nations, and other groups.	Product needs to remain affordable for target users, product creates or diminishes opportunities for economic advancement, high development cost creates risk for organization	Due to the fact that the website does not raise money to support itself, it will be an endeavor that operates at a loss for the client to support the needs of several communities.

USER NEEDS

Users in crisis need a way to ask for assistance and clarify the type of support they need through their crisis, because currently it is extremely difficult for people to ask for assistance and get the support they need (as evidenced by several research studies provided by the client) more easily.

Users wanting to support others need an easier way to understand what their community members need to be able to provide appropriate assistance, because currently it is extremely difficult for supporters to really know the best way to help their loved ones in crisis situations.

Communities need a straightforward way to organize who in the community will provide support to the individuals in crisis at any given time to spread out the support for an individual in need, so they receive a balance of support and not sporadic support because currently, there is no simple way to organize when people are going to provide their support to their community members.

TECHNICAL COMPLEXITY

The project has significant technical complexity due to the level of security needed to protect user information. Because there are accounts associated with users, and people can register to support one another in ways including picking up children from school, it is necessary to ensure that the users' information is secure and protected. This involves engineering security for the coding base and determining the best way to secure user information is a challenge that must match the current industry standards for data security.

DESIGN EXPLORATION

Design Decisions

One primary design decision that was made was to have the navigation of the website be located at the top of the web page rather than in a dropdown bar or some other format. A second design decision was to allow for users to message one another. This was a decision the team came to when it was realized that to provide support to someone, it may be necessary to have a conversation with them to arrange a time that they can provide the support. It would therefore be easiest if the same website where users schedule their time to support an individual allows them to communicate with others on when works best. A third design decision was to decide on a logo and web name for the project. The team allowed the client to decide, but the group produced several options to choose from, and suggested ideas for logos that could go along with each of our name ideas.

Ideation

For the navigation of the website, the team elected to have the navigation system be on the top of the website because it was like many other sites, and therefore more likely to be familiar to the user base and more intuitive to use and navigate than if the design were to change the navigation method to something more unique. The team spent some time thinking of different navigation options. The pros and cons of each idea were weighed, and it was determined that the navigation at the top of the web page would be best. Some other options that had been considered included:

- Navigation along the left side of the web page
- a drop-down menu
- navigation along the right side of the web page
- navigation along the bottom
- and the navigation at the top of the page

The team chose to have the navigation be at the top of the page.

Decision-Making and Trade-Off

Choosing to have the navigation to be in a header instead of a dropdown was due to the accessibility and ease of use it provided. A sidebar can be convenient for a minimalist look but having access to most of the website in an easy to find and navigate header seemed to be the best option. Allowing users to have access to a messaging feature was something we felt

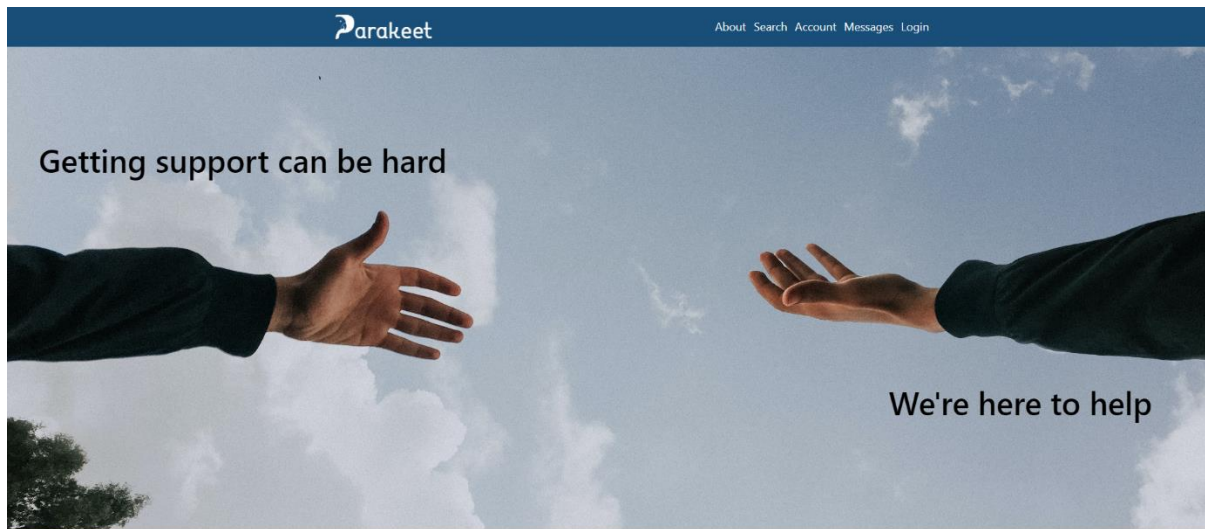
important to have because of the level of support it could provide. Being able to talk to someone, whether to set up times to meet or to provide support through text, is an important part of the website. We decided against not having an messaging feature and possibly providing a phone number or contact info because we want the app to be fully available and usable by anyone. The team decided to let the client decide with our help because it is the client's idea and the group's goal is to help create their idea into a reality. The team wanted the client to have the final say to keep the work aligned with their idea.

PROPOSED DESIGN

On the front end, the design starts on a default homepage that welcomes users in and tells them a little description of the website. The site has a header that displays the website's name on the right and has five links to different pages. The first page is an about page and a mission statement. Those pages are stand alone and only link to the home page. There is an account page where users can monitor their account as well as access their calendar, account settings, and request support pages. The search page allows users to search through various filters for accounts. There is a messages tab that will allow users to communicate with one another and finally a login page that prompts a user to login or allows a user to register a new account. Our current color scheme is a blue palette, using mostly navy blue as the brand color with light gray and lime green accent colors. The palette can be found in the next section. The front end is being developed via React.JS using Grommet and Visual Studio Code as the IDE. The site will be deployed for testing on an ISU testing server until the official server (provided through the client) is ready to be used to deploy the server-side code on. The components communicate with the back end through a series of routers and websockets that communicate data entered from the front end to send to the back end and vice versa.


On the back end, the design will be handled using AWS Amplify data UI and GraphQL api calls. AWS Amplify provides a visual editor for creating data structures which can be automatically input into a graphql schema and imported into the frontend. The data structures will include structures for accounts, setting, activity feeds, messaging, requests, support acceptance, etc. Amplify automatically uses GraphQL to create the API calls and creates a DynamoDB database to store the data.


Design Visual and Description




As shown by the visual above, the home page of the website hosts the name of the site in the top left corner. In the top right, there are redirects to different web pages, including to the about page, the search functionality, the account information page, the message center, and the ability to login.

How it works...

 To use this website, it is important to first register your account. It can easily be done by clicking the button below which will take you through the process of registration. There are two primary user types within this website, though users can fall under both user categories simultaneously if they so choose.

 The first user type is the user in-need. These users created an account on this website to help bring attention to their community about support they seriously need. Often, there are people who are in crisis who seriously need emotional and tangible support who don't effectively get what they need. This website serves to help organize their needs in one place and allow their support network to more effectively figure out what their loved ones need most in their most stressful times.

 The second primary user type is the supporter category. These users are community members who want to assist their loved ones as well as individuals who simply want to support others in need. These users are able to search for specific people in need to follow, so they can more easily organize when people need help as well as what kind of help they need. The supporters can also interact with other users' calendars to sign up for different support tasks, find out other users' availability for emotional support networks, and the like. They can also message other users to communicate plans and determine the best way to help those in need.

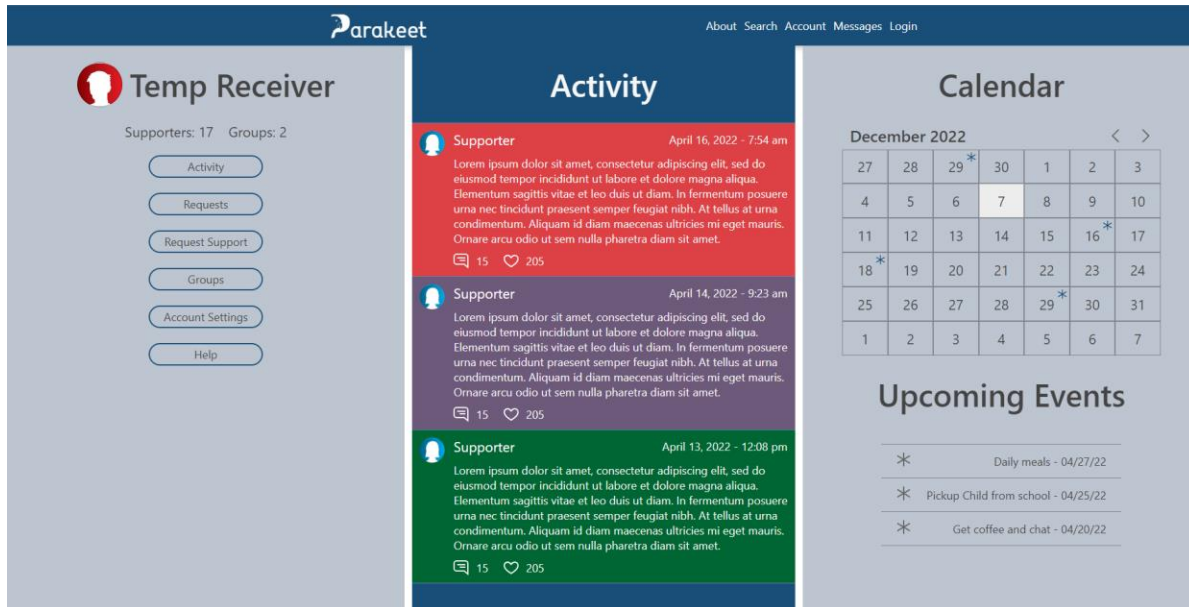
[Register now to start receiving support](#)

Our mission:

[Full Mission Statement](#)

- * The goal of Parakeet is to help individuals get the type and amount of support they need, when it's most needed.
- * Many people struggle to get the support they need, despite loved ones frequently offering to 'be there for whatever' they need.
- * This website is to help users organize those offers of aid, and get the help that they most need in an efficient and effective manner.

Scrolling down on the home page better details the website's purpose and functionality and allows users to register a new account. The search features allow users to filter between searching for other users, different groups on the site, and different organizations associated with the website.



The above image shows the account page, which is the main page of the application. Users are able to access their activity feed which will show any activity regarding their requests. On the left side, users are able to navigate to different screens to view current requests, create a request, edit and view groups, change account settings, and get help regarding the website.

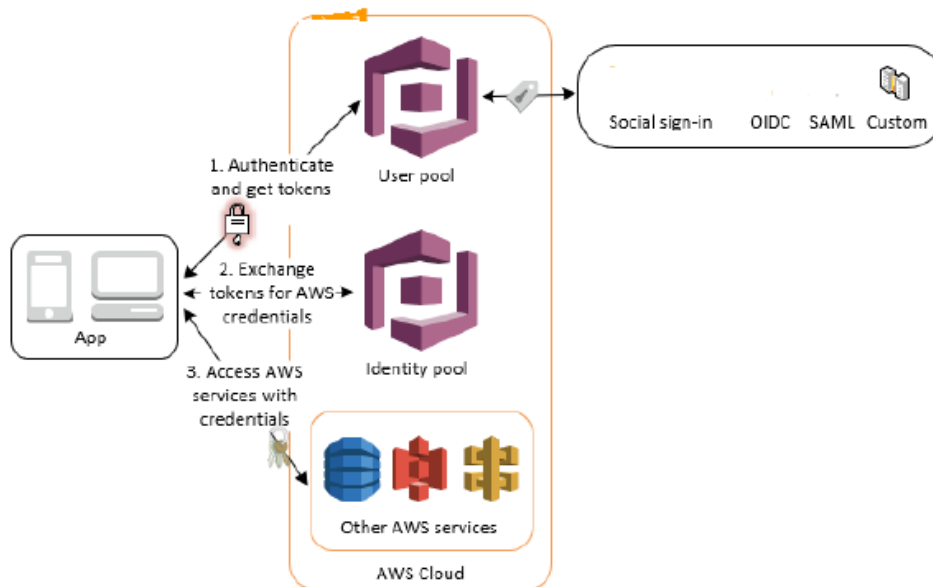
FUNCTIONALITY

The design is intended to function as a middle ground to receive support. The goal is that the user could use the app similarly to social media, where they can check it as they please and request support at any time, with supporters being able to help at any time as well. The app also would help with organizing support, allowing users to see when agreed upon assistance will take place with the calendar, link to GoFundMe for monetary support, or organize supporters for certain days, for example in the case of food delivery. The current design is functional for demo purposes using test data built into the frontend. The goal was for the client to have an app to show to potential users and demonstrate what an account looked like that would be receiving support and an account that would be supporting other accounts. However, there were some functionalities we had hoped to be completed that were unable to be finished. These included the search/discover feature and a majority of the backend functionality.

TECHNOLOGY CONSIDERATIONS

One technology we are utilizing is AWS Cognito. The service is easy to set up and use within the React application, and we got a functioning registration of a user demonstrated. Cognito can make account registration secure and encrypted, which is important to our application since we

are dealing with mostly medical issues. One of the trade-offs of Cognito would be the pricing since if we were to hit an active monthly user count of greater than 50,000 users, we would have to pay per user. On the other hand, the service is free until we reach that amount. The relationship will be the frontend will query AWS Cognito when a user registers or logs in and Cognito will return a token, which will be again passed to Cognito and will return the user's data. This data will be used between the frontend and backend to load user data. Below is a diagram of how Cognito integrates into an application.



For the frontend, we are using ReactJS framework, which is an easy way to create applications. React is easily expandable as well so if in the future, the client wants to add onto the application, it can be done quickly and easily. A downside is that only Kellan Hulet had experience in React before the project so there is a learning curve to overcome.

DESIGN ANALYSIS

For the final design, the front-end of the website has been determined to be mostly functional. The messaging features were not able to be completed, along with a few of the smaller features. The client is satisfied with the design thus far and hopes to continue development on the website with other teams in the future to complete the unfinished design.

TESTING

Testing the product involves protecting data for users, ensuring proper data storage, testing data updates that properly update every applicable section of the website without altering

unnecessary pages, and several other tests. Because the product is a website that functions similarly to many social media sites in combination with many fundraising/service sites, testing will involve primarily tests that are like those performed on social media and fundraising/servicing websites.

UNIT TESTING

Website pages- loading the page, organization of information, navigation away. Main tools will be browsers and error reporting for bugs.

Server responses- PostMan for checking without specific website page navigation, creating a new profile, changing a current profile, searching for a specific user profile, multiple requests at once, multiple chats at the same time, multiple messages from a user, multiple messages to a user, multiple support job requests, error message testing for the calendar, websites not loading, chat permissions for users not allowed to view the chat, community calendar, loading multiple web pages

Database storage- multiple open chats at the same time, multiple messages, calendars, additional templates, public chats with multiple users, feedback from users. Tools will include PostMan, web browsers, and some unit testing within MySQL and Node.js for bugs and errors.

INTERFACE TESTING

The application will be based around each individual user. The personalized interfaces must be tested to ensure each user sees a correct version of the page. For example, each user account page will be personalized to them, similarly to social media. The team will want to make sure that the data displayed on the account page would be the same as displayed on the calendar page and to other users looking at this user's profile. To test this, with each pull request, the group will have manual tests to ensure the data is displaying as expected. A specific example will be the user profile page and calendar. The profile page will have a smaller version of the calendar and upcoming events. These events and smaller calendars must match the calendar page with any changes made. The team will be using Jest and Enzyme to create automated tests that can be run for every new pull request made to make sure the functionality remains the same.

INTEGRATION TESTING

A few critical integration paths in our design involve the search feature and how it interacts with accounts and groups. For starters, a user's account will have a public or private setting with a public account being able to be supported by anyone and a private account requiring accepting followers. One implementation may have only public accounts show up on the search function and a private account can be found through a link.

SYSTEM TESTING

System testing will be done using Jest as well. We want to create an in-depth suite of tests that will cover the basic functionalities of the app, such as creating an account, following a user, adding a request for support, etc. We want to make sure that the app will handle errors correctly without crashing and display a useful error message to the user. We will also have tests that make sure the users' data is being displayed correctly across pages and that changes will be handled correctly as well.

REGRESSION TESTING

To ensure new additions do not break the program, we are working on several different branches of git. This prevents work conflicts and allows us to test new code, while still going back to older versions if the new code does not work. Critical features that cannot break include our server communication and our database storage.

ACCEPTANCE TESTING

Our website will include a feedback page for users to be able to make additional requests, explain what they like about the web design, and what is not working with the product. Over the summer, the client released a beta version of the website to users to be able to test it and interact with the feedback page, so the following semester we used that feedback to improve the design.

SECURITY TESTING

Our plan for security testing is to design with security in mind and take steps to ensure user data security. One example of this was to use AWS Cognito as our user login and registration system. Since AWS handles encryption and data security, we do not have to worry about creating our own encryption system which would be far less secure than the one from AWS. Cognito also lets us set up two factor authentication for added account security.

RESULTS

For acceptance testing results, the team received lots of feedback on what the website was doing right and ways to improve. Some of the more key features that needed to be improved were the event interaction, messaging, and calendar features. Unit testing showed that each individual component of the front-end was working as expected and communicating properly with corresponding back-end components on AWS.

IMPLEMENTATION

Implementation began early in the first semester in order to show the client potential color themes as well as a basic design for the website. By the end of the semester, a working prototype was finished which included several test accounts to provide an example of how the website might function. There was little development done on the backend in the first

semester, outside of setting up a server to run the application over the summer. During the second semester, development continued on the frontend with feedback gathered over the summer being implemented. Due to unfortunate circumstances involving team members no longer continuing with the project, the backend development had to be moved to AWS Amplify in order to attempt to speed up development. Due to these factors, many functionalities were not completed, which included: groups, search/discover, messaging, activity page algorithm, and the ability to save changes to an account.

CONCLUSION

Throughout the project development, the team encountered many setbacks out of our control. These included one teammate leaving partly through the second semester for unknown reasons (the team has been unable to get ahold of this team member for much of the semester, despite the professor's attempts to reach out to him as well), a new teammate joining the group and needing to be filled in on the progress made as far as well as what was left to be done, and lastly a team member needing to withdraw from school due to a health crisis. Due to these setbacks, it is our impression as a team that there is plenty of content that still needs to be done to complete this website to our client's requirements. We therefore recommend the continuation of this project by a future development team, using the documentation provided to the client by our team to pick up where we left off.

APPENDICES

Appendix 1: User's Manual / Basic Functionality Guide 2-5

Appendix II: Introduction to Project 6-7

Appendix III: Project Plan 8-14

Appendix IV Design 15-24