

- **URL: /help**
 - This page requires that the visitor is either loggedIn via /login, or their email address has been validated from the form on the root URL page
 - This page has two containers
 - Container 1
 - Form
 - Select Dropdown Menu: How can we help you today? (required)
 - Values will be provided
 - Text Area: Please enter a brief description (optional)
 - Button 1: Initiate Chat Session
 - Button 2: Submit Ticket
 - When user clicks either button, validate the Select Dropdown as a value POST the form
 - Processing the posted form
 - Create a ticket using the Autotask API
 - Sample code in createATTicket.js
 - If form was posted using Button 1: redirect to /chat
 - If form was posted using Button 2:
 - If LoggedIn: redirect to /tickets
 - Else: redirect to /login
 - Container 2 (only display if user is NOT loggedIn from the /login page)
 - Link: Login (redirects to /login, pre-populates email field of login form)
 - <p>Paragraph of descriptive text below the Login link</p>

PLEASE QUOTE HOURS ABOVE THIS POINT AS A SEPARATE BLOCK OF TIME

- **Angular and Node.JS Design Review**
 - Review our web application as it is today. If any changes are required to update the routing, views or any other functionality so it uses Angular & Node.JS best practices for design and functionality, please allow for time to make updates so that all the work below and future projects aren't limited because of early design choices.
- **Global Configuration Variables**
 - We've been using a username/password combo PPierce@cntdemo.com / 12345 a lot for demo testing. Move this variables into a global config file of some kind that can be easily edited in one place and have the changes apply everywhere they're needed in the application.
 - Move other hard coded variables into the global config as well so all hard coded values are easy to edit in one place for the entire application.
- **Update Session Cookie and have the following stored:**
 - clearNorth: true/false (based on whether a clearnorthtech.com valid user logged in)
 - loggedIn: true/false (based on whether any successful login occurred)
 - resourceID: number (id of resource when ClearNorthTech resource logs in)
 - contactID: number (value of contactID from Contacts cache, only if ClientPortalUser)
 - accountID: number (value of accountID from Accounts cache, make sure to use parentID if applicable)
 - accountName: string (value of name from Accounts cache, make sure to use parentID if applicable)
 - securityLevel: number (value of securityLevel from ClientPortalUser cache)
 - firstName: string (value of firstName from Contacts cache, only if ClientPortalUser)
 - lastName: string (value of lastName from Contacts cache, only if ClientPortalUser)
 - email: string (value from login or email validation)
- **URL: /updatePassword**
 - Provide an "update password" function for users (only for users that don't have @clearnorthtech.com email addresses) when they're logged in. Please the link near the Logout link.
 - Allow the user to enter a new password that meets the requirements of our reset password function. Require them to enter the password twice. When they click, "Update Password", if the passwords they've entered match, send the updated password to Autotask API using code similar to the password

reset function. If the entered passwords don't match or don't meet the complexity requirements, display an appropriate error message

- **Real-time Web Chat**

Web based real-time chat using Socket.io library (supported by Azure)

- Do not specifically copy this code, but it provides some information on using a Socket.io chat app within Azure, <http://azure.microsoft.com/en-us/documentation/articles/web-sites-nodejs-chat-app-socketio/>
- All chat activity needs to be logged somewhere. Mondo? Redis? I'm open to input..
- All chat messages should be time-stamped, but also insert a timestamp into the chat display every 15 minutes (Use Central Time for the timezone)
- **URL: /chat (this is the client view to the Chat engine)**
 - **Require: Validated Email or Client Access Portal LoggedIn**
 - Display: Ticket Number: Ticket Title as page header, which will be returned by the Autotask Create Ticket API call
 - Display Status Message:
 - If there is no ClearNorthAgent in their chat session, display a status message above the chat window that says, "Waiting for Support Agent"
 - If a ClearNorthAgent joins the chat, change the status message to, "You are chatting with <AgentName>"
 - Allow the user and ClearNorthAgent to chat within the chat container
 - Provide an "End Session" button below the chat container
 - When user clicks this button, Display a message to ClearNorthAgent that the user has left the chat
- **URL: /chatDashboard (this is the clearnorthtech.com user view of the chat engine)**
 - **Require: Login using a clearnorthtech.com email address on /login**
 - Two containers
 - Left Container – narrow, maybe 200px
 - List all user initiated chats from /chat
 - List should be sorted in order of the initiated chat time, oldest on top
 - Each item in the list should contain the following:
 - The ticket number created when user clicked Button 1
 - The user's Name (returned with the ticket created)
 - The user's Company (returned with the ticket created)
 - How long ago (in minutes) the chat session was started
 - Active ClearNorthAgent in the chat (if there is one)
 - Otherwise display text to be clicked: "Join Chat Session"
 - As the timer gets closer and closer to being open for 15 minutes without a ClearNorthAgent joining the chat, the background gets brighter red (can be tiered in different colors)
 - When a ClearNorthAgent joins the chat, change the background color to green
 - If a chat has been joined, but hasn't been updated by a ClearNorthAgent for 5 minutes, change the background color to yellow
 - When a ClearNorthAgent clicks "Join Chat Session" link, open it in the right container and update the status
 - When either the user or the ClearNorthAgent close the chat, remove it from this list
 - Right Container – wider, maybe 800px
 - Tabs across the top for each open chat
 - Above the chat container display
 - Ticket number
 - User's name
 - User's company
 - Chat container that displays the chat session with the end user

- Below the chat container
 - `<p>Descriptive text</p>`
 - Button to “End Session”, clicking this sends a message to the User that the Agent has ended the session, it closes the tab, removes the chat from the left container, and sends a TimeEntry to Autotask that contains the chat transcript. Sample code in createATTimeEntry.js.
 - Button to “Leave the Session” but leave it open. Clicking this closes the tab, changes the chat box in the left container to display “Join Chat Session”, sends a message to the user that the agent has left the session, and sends a TimeEntry to Autotask that contains the chat transcript. Sample code in createATTimeEntry.js.
- URL: /ticket**
 - This page requires that the visitor is loggedIn via /login
 - At top of page
 - A box that links to the /help page, using text “Open a new Ticket”
 - If user security level > 6 (sample code in checkATsecurityLevel.js, this could be tested at Login time and stored in the session cookie)
 - A box that links to /changeRequest, using the text, “Initiate a Change Request”, this URL will display the original form we created in Phase 1.

PLEASE QUOTE HOURS BETWEEN THIS POINT AND THE PREVIOUS QUOTE BREAK POINT
AS A SEPARATE BLOCK OF TIME,

- Below the navigation boxes
 - If user security level < 6
 - Query Autotask API for a list of open tickets for the current user
 - Else
 - Query Autotask API for all open tickets for user’s company
 - Display the tickets returned in a grid with the following columns (I would like this grid to be a javascript style grid widget if possible, instead of a basic HTML table)
 - Ticket Number
 - Ticket Title
 - Contact
 - Priority
 - Status
 - Source
 - Last Updated
 - Make the ticket number a clickable link to a URL /ticket/<ticketID>
 - Note: the ticketID and ticketNumber are different values
 - Of the API call is slow to respond, display a spinner\loading GIF until the data is returned
 - Sample code in listATTickets.js
- URL: /ticket/<ticketID>**
 - This page requires that the visitor is loggedIn via /login
 - This page also requires that the user be the assigned contact on the ticket or that the user’s security level is > 6
 - Provide a link to return to /ticket, called, “Return to Ticket List”
 - `<h1>Ticket Title</h1>`
 - `<h2>Ticket Number</h2>`
 - List
 - Contact
 - Priority
 - Status

- Source
- Clear North Assigned Resources (if any)
- <p>Ticket Description</p>
- Text Area: Add a Comment to this ticket
- Button: Submit Comment
 - When clicked, the comment is added to the ticket as a Note, see sample code in createATNote.js
- List All Time Entries & Notes currently on the ticket, sorted in Descending time order, see sample code in listATTicketLog.js. If the entity is a Time Entry, identify it as a ClearNorth update.

PLEASE QUOTE HOURS ABOVE THIS POINT AND BELOW THE PREVIOUS QUOTE BREAK POINT
AS A SEPARATE BLOCK OF TIME, THIS MAY BE DONE AT A LATER DATE
