|  |  |  |
| --- | --- | --- |
| **Objective** | **Requirements** | **Solution** |
| **User should be able to Register and/or Sign in** | A user not signed in needs to be able navigate to the site (publicly accessible page or “Landing Page”) and have the options to:   * Sign in as an existing user or * Register as new user | **LandingPage**   * Call to action buttons for:   + Sign in   + Register |
| On a user navigating to the Landing Page, the options presented to them should be based on their signed in status:   * If signed in – they should be prompted to go to the platform * If not signed in - they should be prompted to either sign in or register   Should not show any buttons until the users signed in state is determined | **AuthService**   * Able to determine user authentication status.   **AuthGuard**   * Able to determine if a user is signed in   **LandingPage**   * Dynamic enough to handle each case:   + Sign in   + Not signed in   + Undetermined * Call to action buttons for:   + Dashboard (signed in)   + Sign in (not signed in)   + Register (not signed in) |
| In a user is not signed in and they try to navigate to an in-platform route, they need to be navigated to a Landing Page | **AuthService**   * Able to determine user authentication status.   **AuthGuard**   * Able to determine if a user is signed in   **RoutingModule**   * With AuthGuard applied to protect in-platform routes from users not authenticated. |
| On signing in or registering the user should be taken into the platform to a logical location | **SignInModal**   * With navigation post-authentication.   **RegistrationModal**   * With navigation post-authentication.   **AuthService**   * Able to   + Authenticate the sign in attempt.   + Register a new user   **AuthGuard**   * Able to:   + Protect in platform routes, but   + Allow authenticated users through |
| Registration form needs to capture the following details of the user:   * First name * Last name * Email * Password (hidden) * Confirm password (hidden)   Action buttons required   * Register 🡪 Attempts to register user * Sign in 🡪 Navigates to the sign in form * Cancel 🡪 Takes user back to Landing Page | **RegistrationModal**   * With following fields:   + Text input - First name   + Text input - Last name   + Text input – Email   + Password input – Password   + Password input – Confirm Password * With following action buttons:   + Register (Primary)   + Sign in (Secondary)   + Cancel (Secondary) |
| Sign in form needs to capture the following details of the user:   * Email * Password (hidden)   Action buttons required   * Sign in 🡪 Attempts to sign user is * Register 🡪 Navigates to the register form * Cancel 🡪 Takes user back to Landing Page | **SignInModal**   * With following fields:   + Text input – Email name   + Password input – Password * With following action buttons:   + Sign in (Primary)   + Register (Secondary)   + Cancel (Secondary) |
| The sign in and new registration forms need to validate as much as possible to prevent preventable error attempts being made to database.   * (Note – Firebase only allows passwords 6 letters or greater) * If there are errors – user is to be shown a validation error on the form/modal   User should not be able to submit the form until full validated | **RegistrationModal**   * With following fields validations:   + First name - Required   + Last name - Required   + Email - Required and Email   + Password – Required, Min 6 letters   + Confirm Password – Required, Min 6 letters * On blur from the password or confirm password fields – should so a matching validation (prior to form submission) * Register action button disabled until form is fully validated   **SignInModal**   * With following fields validations:   + Email - Required and Email   + Password – Required, Min 6 letters * Register action button disabled until form is fully validated |
| If the sign in or registration attempt fails (sever-side, post form validation), the users should:   * Be informed of this * Provided details of the issue (where possible) and * Prompted to correct these issues and try again | **AuthService**   * Able to:   + Try sign in user and able to confirm is that attempt was success or failed   + Try register user and able to confirm is that attempt was success or failed   **ErrorService**   * Able to:   + Receive the http/BE error and converts it into an error that would be user friendly (a config to pass to an error modal)   **DynamicErrorModal**   * Able to:   + Receive a error config from the ErrorService and render and error modal   + Should be an overlay over the screen content – which has no impact on the screens content – so when dismissed the former screen content remains |
| Should the user have forgotten their password – there needs to be a means for the user to address this. | **SignInModal**   * With following action buttons:   + Forgot Password (Secondary)   **ForgotPasswordModal**   * With following fields:   + Text input – Email * With following fields validations:   + Email - Required and Email * With following action buttons:   + Reset Password (Primary)   + Cancel (Secondary) |
| Users should be authenticated against a user database | **AuthService**   * Able to determine user authentication status. |
| **User should be able to set and edit an appointment** | **Signed in users**   * Should be able to navigate to an appointments page with in the platform – via an appointments menu option of the Navbar * Should also be able to navigate directing to the appointments page with a URL route   **Users not signed in**   * Should not have an appointments menu option available to them * On attempting to navigate directly to the appointments page – should be presented from going so and navigated back to the LandingPage from where they can sign in |  |
| Default view the appointments page should be:   * A calendar   + Set to today   + Showing all appointments for the day |  |
| The user in the calendar should be able to change the day being viewed  By 1 day back or forwards  Or a specific date using a date picker  User should be able to add an appointment (open a new appointment modal)  by clicking a new appointment button  clicking a time slot in the calendar  New appointment modal “New appointment”  Should allow user to enter  The client  The appointment duration  Appointment start time  Date  Where the user has opened the modal by clicking the add appointment button  Date should be set to the day that the user was viewing  Rest should be empty  Where the user has opened the modal by clicking a time slot on the calendar  Date should be set to the day that the user was viewing  Appointment start time should be the time slot the use clicked to open the modal  Rest should be empty  Action buttons  Cancel  Add appointment – once form is completed properly  On clicking an existing appointment “View appointment”  Should be able to view  The client  The appointment duration  Appointment start time  Date  Invoice number (if the appointment has been invoiced) - Can be clicked to take user to invoice detail  At this time the details should not be editable  Action buttons  Close (primary)  Edit/Cancel appointment  Clicking the appoint Edit/Cancel appointment  Should make the modal in “View appointment” mode  Similar to “New appointment” but  All fields will be populated with the appointment details  Invoice number will appear – BUT will not be editable – Can be clicked to take user to invoice detail  Action buttons  Save changes – disabled until changes are made  Close – revert back to “View appointment” mode  Cancel appointment – deletes the appointment  If changes are made and save – warn user of changes  If appointment has invoice and tried to cancel – warn user of invoice needing to be voided – if user confirms – void invoice  If cancelled (no invoice) – warn user  If close – but changes are made – warn user  Limits (for now) - Exclusions  User can only see their own calendar  Appointments cannot be drag and dropped  Appointments cards will all be uniform in size  Appointment durations will be fixed to 15, 30, 45 minutes and 1 hour There will be no appointment types  Calendar view will be limited to 1 day (No week / month etc) |  |
| **User should be able to invoice a client** |  |  |
| **User should be able to record a client payment** |  |  |
| **User should be able to generate a client statement** |  |  |