# Intro

Your client would like to create an appointment management system that focusses on the needs of individual practitioners. Currently available offerings create practitioners as part of a clinic/group setting. If they leave that group, they can lose easy or all access to client data. Or they are individuals that keep control of their client data but cannot collaborate with other practitioners.

Additionally, they would like the customers to be able to control access to the data that is stored and to be able to respond to requests for access. Access contracts between clients and users of their data would define the period that access was allowed for.

Extra Additionally, as new services are added customers accessing those services would be able to allow practitioners to access data that is held rather than having to renter details. If the stored data is missing information required for the new service, then available fields would be populated with flags for the missing info to be entered, saving the client time and improving the user experience.

The app should:

* Allow creation or registration, modification and in some cases deletion or archiving of:
  + Individual practitioners - Provider
  + Customers - Client
  + Locations - Clinic/group physical location
  + Workspace - Somewhere to provide a service in location
  + Appointments
  + Notes - Client, Appointment
  + Organisation - A group of locations
  + Forms - Contact details, medical intake etc
  + Resources - shared resources that need to be reserved or allocated
* Providers should have:
  + Contact Details
  + Business Type
  + Role
  + List of Appointments
  + List of Locations
  + Login details
* Providers should be able to:
  + access relevant info about their clients.
* Clients should have:
  + Contact Details
  + List of services used
  + List of Appointments
* Clients should be able to:
  + Access their own appointments
* Locations should have:
  + Name
  + Contact details
  + List of Workspaces
  + List of Providers
* Locations should be able to get a
  + List of workspaces
  + List of resources
  + List of Users
  + List of Clients
  + List of Appointments
* Workspaces should have a:
  + Name
* Workspaces should be able to get a:
  + List of available resources
  + List of appointments
  + List of resources required for an appointment
* allow them to be aggregated into group settings like clinics or gyms without losing their client data if they leave that setting.

Clinical, shop or other group settings would aggregate the practitioners, who exist in their own right out with the group setting. This allows practitioners who often work in multiple settings to have one diary and one place where they can take notes and manage their clients. An emphasis on flexible data sharing facilities in group settings to allow practitioners that share clients to update a common record, but keep ownership of their own data, especially if they leave the group setting. Other potential features could include:

* Diary system for appointments
* Take notes
* Client Details
* Online booking
* Take and record payments

MVP

|  |  |  |
| --- | --- | --- |
| Must Have | Should have | Could have |
| Front-end:   * Display appointments * Display/Create/Modify:   + Users   + Clients   + Appointments   + Notes * Allow users to be grouped | Front-end:   * Use Diary view (API React Big Calendar?) * Allow configuration of data sharing in groups * Allow sub-groups * Add features for users: Search/Filter | * User Login * User Authentication * Deploy to cloud * Client Login * Online booking for clients |
| Back- end:   * Full CRUD functionality * Restful routes * SOLID principles | Back- end:   * Create routes to support front-end |  |
| UI - Functional CSS | UI - Improved CSS |  |

# Tech Stack

UI - MUI, Bootstrap…

Front end - Reactjs - looking at API’s for displaying calendars (React Big-Calendar)

Back end - Spring Boot API looking at Firebase