# 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 usually 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.

The app should:

* Allow creation of
  + Individual practitioners (Users)
  + Customers (Clients)
  + Organization (A group of locations)
  + Locations (Clinic/group physical location)
  + Workspace
  + Resources
  + Appointments
  + Notes - Client, Appointment
* Users should have:
  + Contact Details
  + Business Type
  + Role
  + Ability to access relevant info about their client.
  + Login (extension)
* Clients should have:
  + Contact Details
  + List of services used
  + List of details for each service
* Organizations should have:
  + Name
  + List of Users
  + List of Locations
  + List of Workspaces
  + List of Resources
  + List of Clients
  + List of Appointments
* Locations should have:
  + Name
  + Contact details
  + List of workspaces
  + List of resources
  + List of Users
* 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