Application Planning – Wild Oasis

# Customer

* Small boutique hotel with 8 luxurious wooden cabins
* They need a custom-built application to manage everything about the hotel:
  + Bookings
  + Cabins
  + Guests
* It’s going to be an internal application used inside the hotel to check in guests when they arrive
* They need the API
* Customer facing website (Later)

# Project Requirements

* Users of the app are hotel employees. They need to be logged in into the application to perform tasks
* New users can only be signed up inside the applications
  + Guarantee that only the hotel employees can get accounts
* Users should be able to
  + Upload an avatar
  + Change their name
  + Change their password
* App needs a table view with all cabins, showing a photo of the cabin, name, capacity, price and current discount
* Users should be able to update or delete a cabin, and to create new cabins (including uploading a photo)
* App needs a table view with
  + Bookings
  + Arrival and departure date
  + Status
  + Paid amount
  + Cabin data
  + Guest data
* The booking status can be
  + Unconfirmed
    - Booked but not checked in
  + Checked in
  + Checked out
* Table should be filtrable by this status
* Other booking data includes
  + Number of guests
  + Number of nights
  + Guest observation
  + Whether they booked breakfast
  + Breakfast price
* Users should be able to
  + Delete booking
  + Check in booking
  + Checkout out booking
* Bookings may not have been paid at guest arrival. Therefore, on check in, users need to accept payment (outside the app) and then conform the payment has been received (inside the app)
* On check in, the guest should have the ability to add breakfast for the entire stay, if they hadn’t already
* Guest data should contain
  + Full name
  + Email
  + National ID
  + Nationality
  + A country flag for easy identification
* The initial App screen should be a dashboard to display information for the last 7, 30, 90 days
  + List of guests checking in and out on the current day
* Users should be able to perform these tasks
  + Statistics on recent bookings, sales, check ins, occupancy rate
  + A chart showing all daily hotel sales, showing both ‘total’ sales and ‘extra’ sales (breakfast)
  + A chart showing statistics on stay durations
* Users should be able to define a few application-wide settings:
  + Breakfast price
  + Min / max nights/ booking
  + Max guests/ booking
* App needs dark mode

# 2 + 3. Features + Pages

* **Authentication**
  + Users of the app are hotel employees. They need to be logged in into the application to perform tasks
  + New users can only be signed up inside the applications
    - Guarantee that only the hotel employees can get accounts
  + Users should be able to
    - Upload an avatar
    - Change their name
    - Change their password
* **Cabins** 
  + App needs a table view with all cabins, showing a photo of the cabin, name, capacity, price and current discount
  + Users should be able to update or delete a cabin, and to create new cabins (including uploading a photo)
* **Bookings**
  + App needs a table view with
    - Bookings
    - Arrival and departure date
    - Status
    - Paid amount
    - Cabin data
    - Guest data
  + The booking status can be
    - Unconfirmed
      * Booked but not checked in
    - Checked in
    - Checked out
  + Table should be filtrable by this status
  + Other booking data includes
    - Number of guests
    - Number of nights
    - Guest observation
    - Whether they booked breakfast
    - Breakfast price
* **Check In/ Check Out**
  + Users should be able to
    - Delete booking
    - Check in booking
    - Checkout out booking
  + Bookings may not have been paid at guest arrival. Therefore, on check in, users need to accept payment (outside the app) and then confirm the payment has been received (inside the app)
  + On check in, the guest should have the ability to add breakfast for the entire stay, if they hadn’t already
* **Guests**
  + Guest data should contain
    - Full name
    - Email
    - National ID
    - Nationality
    - A country flag for easy identification
* **Dashboard**
  + The initial App screen should be a dashboard to display information for the last 7, 30, 90 days
    - List of guests checking in and out on the current day
  + Users should be able to perform these tasks
    - Statistics on recent bookings, sales, check ins, occupancy rate
    - A chart showing all daily hotel sales, showing both ‘total’ sales and ‘extra’ sales (breakfast)
    - A chart showing statistics on stay durations
* **Settings**
  + Users should be able to define a few application-wide settings:
    - Breakfast price
    - Min / max nights/ booking
    - Max guests/ booking
  + App needs dark mode

## Pages

* Bookings
  + /bookings
* Cabins
  + /cabins
* Guests
* Dashboard
  + /dashboard
  + This is the homepage
* Check in Check out
  + /checkin/:bookingID
* App settings
  + /settings
* Authentication
  + /login
* User sign up
  + /users
* Account settings
  + /account

# Tech Decisions

## Client Side Rendering (CSR) vs Server Side Rendering (SSR)

### CSR with Vanilla React

* Used for SPA’s
* 100% rendered on the client
* JS needs to be downloaded before app is starring
  + Bad for performance
* **BEST for internal tools used by companies**
  + Hidden behind a login screen
  + Not needed with SEO

### SSR with Framework (Next.js or Remix)

* Used to build **Multi-Page apps (MPAs)**
* Some of the HTML can be rendered on the server
* The **React team is advocating more and more for this**

## We are going to use CSR

**Routing**

* React Router
  + Gold standard for SPA’s

**Styling**

* Styled Components
  + Very popular react library

**State management**

* **Remote State management** (our app will implement an easy to use interface that lives on a server)
  + React Query
    - Best way of managing remote state
    - Many features like caching, automatic re-fetching, offline support
* **UI** 
  + There will be very little UI state management
  + Context API

**Form Management**

* React Hook Form
  + Handling bigger forms can be a lot of work. This library will simplify all this