

Phase 1 & 2: Simplified, Maintainable Rails 7 Setup

Phase 1: Users & Access Control (Simplified)

Generate Devise User Model

```
rails g devise User
```

Add extra fields via migration

```
rails g migration AddFieldsToUsers role:integer status:integer  
phone_number:string two_factor_enabled:boolean failed_attempts:integer  
last_login_at:datetime
```

Migration example:

```
class AddFieldsToUsers < ActiveRecord::Migration[7.0]  
  def change  
    add_column :users, :role, :integer, default: 0, null: false  
    add_column :users, :status, :integer, default: 0, null: false  
    add_column :users, :phone_number, :string  
    add_column :users, :two_factor_enabled, :boolean, default: false  
    add_column :users, :failed_attempts, :integer, default: 0  
    add_column :users, :last_login_at, :datetime  
  end  
end
```

Generate Agent, Client, Admin Models (linked to User)

```
rails g model Agent user:references full_name:string agency_name:string  
license_number:string commission_rate:decimal status:integer  
onboarded_at:datetime core_system_id:string  
rails g model Client user:references agent:references full_name:string  
national_id:string date_of_birth:date address:text kyc_verified:boolean  
onboarding_source:integer core_system_id:string  
rails g model Admin user:references full_name:string role:integer  
department:string
```

Models & Associations

app/models/user.rb

```
class User < ApplicationRecord
  devise :database_authenticatable, :recoverable, :rememberable, :validatable

  enum role: { client: 0, agent: 1, admin: 2 }
  enum status: { active: 0, inactive: 1, suspended: 2 }

  has_one :agent
  has_one :client
  has_one :admin
end
```

app/models/agent.rb

```
class Agent < ApplicationRecord
  belongs_to :user
  has_many :clients
  has_many :reservations
  has_many :sales, through: :reservations
  enum status: { active: 0, suspended: 1 }
end
```

app/models/client.rb

```
class Client < ApplicationRecord
  belongs_to :user
  belongs_to :agent, optional: true
  has_many :kyc_documents
  has_many :applications
  has_many :reservations
  has_many :accounts
  has_many :transactions, through: :accounts
end
```

app/models/admin.rb

```
class Admin < ApplicationRecord
  belongs_to :user
  enum role: { super_admin: 0, kyc_officer: 1, finance_officer: 2,
```

```
support_officer: 3 }  
end
```

Namespaced Dashboards

```
rails g controller agent_portal/dashboard index  
rails g controller client_portal/dashboard index  
rails g controller admin_portal/dashboard index
```

Routes Setup

config/routes.rb

```
Rails.application.routes.draw do  
  devise_for :users  
  
  namespace :agent_portal do  
    root 'dashboard#index'  
  end  
  
  namespace :client_portal do  
    root 'dashboard#index'  
  end  
  
  namespace :admin_portal do  
    root 'dashboard#index'  
  end  
end
```

Devise Redirects Based on Role

app/controllers/application_controller.rb

```
class ApplicationController < ActionController::Base  
  before_action :authenticate_user!  
  
  def after_sign_in_path_for(user)  
    case user.role  
    when 'agent'  
      agent_portal_root_path  
    when 'client'  
      client_portal_root_path  
    when 'admin'
```

```
      admin_portal_root_path
    else
      root_path
    end
  end
end
end
```

Run Migrations

```
rails db:create db:migrate
```

Phase 2: Project & Stand Management

Generate Models & Migrations

```
rails g model Project core_project_id:string name:string location:string
description:text developer_name:string total_stands:integer status:integer
map_image_url:string
rails g model Stand project:references core_stand_id:string stand_number:string
area_sqm:decimal price:decimal status:integer reserved_at:datetime
sold_at:datetime latitude:decimal longitude:decimal
updated_from_core_at:datetime
```

Model Associations

app/models/project.rb

```
class Project < ApplicationRecord
  has_many :stands
  has_many :reservations, through: :stands
  has_many :accounts, through: :stands
  enum status: { active: 0, completed: 1, archived: 2 }
end
```

app/models/stand.rb

```
class Stand < ApplicationRecord
  belongs_to :project
  has_many :reservations
  has_one :current_reservation, -> { where(status: 'confirmed') }, class_name:
```

```
'Reservation'  
  has_one :account  
  enum status: { available: 0, reserved: 1, sold: 2 }  
end
```

Controllers & Namespaces

```
rails g controller admin_portal/projects index show new create edit update  
rails g controller admin_portal/stands index show new create edit update
```

Routes Setup

config/routes.rb addition:

```
namespace :admin_portal do  
  resources :projects do  
    resources :stands  
  end  
end
```

Views

- app/views/admin_portal/projects/index.html.erb
- app/views/admin_portal/projects/show.html.erb
- app/views/admin_portal/stands/index.html.erb
- app/views/admin_portal/stands/show.html.erb

(Use simple tables to list projects/stands, and forms for new/edit.)

Run Migrations

```
rails db:migrate
```

With this simplified setup: - **One Devise User model** ✓ - **Role-based portals** ✓ - **Shared resources (Projects, Stands)** ✓ - **Easy to maintain and extend** ✓