ChatGPT

Decidim Volunteer Scheduler Component

1. Generate the Component

First, scaffold the engine using Decidim's generators. In your Decidim Rails app, run:

```
rails generate decidim:component volunteer_scheduler
```

This creates the basic engine structure and gemspec (following Decidim's official pattern [1] [2]), including an isolated namespace (`Decidim::VolunteerScheduler`) and placeholder files.

2. Engine Structure and Registration

Fill in the generated files to define the gem and engines:

- `decidim-volunteer_scheduler.gemspec`: Define the gem with dependencies on Decidim and Rails.

```
Gem::Specification.new do |spec|
  spec.name          = "decidim-volunteer_scheduler"
  spec.version       = Decidim::VolunteerScheduler::VERSION
  spec.summary       = "Volunteer scheduling and referral coordination for
Decidim"
  spec.description   = "A Decidim component for managing volunteer tasks,
XP leveling, and a 5-level referral system."
  spec.authors       = ["Your Name"]
  spec.email         = ["you@example.com"]
  spec.homepage      = "https://github.com/yourorg/decidim-
volunteer_scheduler"
  spec.license       = "AGPL-3.0"

  spec.files         = Dir["{app,config,db,lib}/**/*", "MIT-LICENSE",
"Rakefile", "README.md"]
  spec.add_dependency "rails", ">= 6.0"
  spec.add_dependency "decidim-core", ">= 0.30"
end
```

  Dependencies: at minimum `rails` and `decidim-core` (and any other required gems).

- `lib/decidim-volunteer_scheduler.rb`: Main require file for the gem.

```
require "decidim-volunteer_scheduler/engine"
```

- `lib/decidim/volunteer_scheduler/version.rb`: Version constant for the gem.

```
module Decidim
  module VolunteerScheduler
    VERSION = "0.1.0"
  end
end
```

- `lib/decidim/volunteer_scheduler/engine.rb`: Defines the public (user-facing) engine. Use `isolate_namespace` and add any Decidim initializers (e.g. menu integration, assets). Example:

```
module Decidim
  module VolunteerScheduler
    class Engine < ::Rails::Engine
      isolate_namespace Decidim::VolunteerScheduler

      # Precompile any additional assets (e.g. Webpacker packs)
      initializer "decidim.volunteer_scheduler.assets.precompile" do
        Decidim::VolunteerScheduler::Engine.config.assets.precompile +=
["decidim/volunteer_scheduler/*.svg"]
      end

      # Add to Decidim menus (e.g. user menu, admin menu)
      initializer "decidim.volunteer_scheduler.menu" do
        Decidim.menu :user_menu do |menu|
          menu.add_item :volunteer_dashboard,
            I18n.t("menu.volunteer_dashboard", scope:
"decidim.volunteer_scheduler"),
            :decidim_volunteer_scheduler,
            position: 2.5,
            if: proc { current_user&.volunteer? }
        end

        Decidim.menu :admin_menu do |menu|
          menu.add_item :volunteer_management,
            I18n.t("menu.volunteer_management", scope:
"decidim.volunteer_scheduler.admin"),
            :decidim_volunteer_scheduler_admin,
            position: 7,
            active: :inclusive,
            if: proc { allowed_to?(:enter, :space_area,
```

```
space_name: :admin) }
        end
      end
    end
  end
end
```

Notes: Uses Decidim menu API [3] ; menu names should have corresponding I18n keys (e.g. in `config/locales/en.yml`).

- `lib/decidim/volunteer_scheduler/admin_engine.rb`: Defines the admin (back-office) engine.

```
module Decidim
  module VolunteerScheduler
    class AdminEngine < ::Rails::Engine
      isolate_namespace Decidim::VolunteerScheduler::Admin

      # Admin-specific initializers (if any) can go here.
    end
  end
end
```

Notes: This engine is isolated under `Decidim::VolunteerScheduler::Admin` and is mounted in the admin UI (as registered below).

- `lib/decidim/volunteer_scheduler/component.rb`: Registers the component with Decidim. Use the DSL as in Decidim's docs [4] . For example:

```
Decidim.register_component(:volunteer_scheduler) do |component|
  component.engine = Decidim::VolunteerScheduler::Engine
  component.admin_engine = Decidim::VolunteerScheduler::AdminEngine
  component.icon = "decidim/volunteer_scheduler/icon.svg"
  component.name = "volunteer_scheduler"

  # Example lifecycle hooks:
  component.on(:create) do |component_instance|

Decidim::VolunteerScheduler::CreateDefaultTemplates.call(component_instance)
  end
  component.on(:destroy) do |component_instance|

Decidim::VolunteerScheduler::CleanupComponentData.call(component_instance)
  end
```

```
  # (Optional) settings, exports, etc. can be defined here.
end
```

Dependencies: none beyond Decidim. Ensure the icon file exists in `app/assets` if referenced.

- `lib/decidim/volunteer_scheduler/test_utils.rb` (optional): Helpers for specs. Can define factory or helper methods. Example placeholder:

```
module Decidim::VolunteerScheduler::TestUtils
  def create_volunteer_profile_for(user, referrer: nil)
    # implementation for specs
  end
end
```

3. Component Registration in Decidim

The `Decidim.register_component` block (in `lib/decidim/volunteer_scheduler/component.rb`) wires up the component per Decidim conventions [4]. It specifies the main and admin engines, an icon and name, and can hook into lifecycle events. We've shown a basic example above. All literal strings should be translated via I18n (e.g. locale keys `decidim.volunteer_scheduler.menu.volunteer_dashboard`, `decidim.volunteer_scheduler.admin.menu.volunteer_management` in `config/locales/*.yml`).

4. Database Migrations (Base Models)

Create migrations for the core tables. For example:

- `db/migrate/20250101000001_create_task_templates.rb` (TaskTemplate):

```
class CreateTaskTemplates < ActiveRecord::Migration[6.1]
  def change
    create_table :decidim_volunteer_scheduler_task_templates do |t|
      t.references :component, null: false, foreign_key: {
to_table: :decidim_components }
      t.string   :title,          null: false, limit: 150
      t.text     :description,    null: false
      t.integer  :level,          null: false, default: 1
      t.integer  :frequency,      null: false, default: 0
      t.integer  :category,       null: false, default: 0
      t.integer  :xp_reward,      null: false, default: 10
      t.decimal  :scicent_reward, precision: 10, scale: 2, default: 0
      t.boolean  :active,         null: false, default: true
      t.datetime :available_from
```

```
      t.datetime :available_until
      t.integer  :max_assignments
      t.text     :requirements
      t.jsonb    :instructions,    default: {}
      t.jsonb    :metadata,        default: {}
      t.timestamps
    end
    add_index :decidim_volunteer_scheduler_task_templates, :component_id
    add_index :decidim_volunteer_scheduler_task_templates,
[:component_id, :level], name: "index_templates_on_component_and_level"
  end
end
```

Notes: Follows the schema in the spec [5]. Adds indexes for performance.

- `db/migrate/20250101000002_create_task_assignments.rb` (TaskAssignment):

```
class CreateTaskAssignments < ActiveRecord::Migration[6.1]
  def change
    create_table :decidim_volunteer_scheduler_task_assignments do |t|
      t.references :task_template, null: false, foreign_key: {
to_table: :decidim_volunteer_scheduler_task_templates }
      t.references :assignee,      null: false, foreign_key: {
to_table: :decidim_users }
      t.references :reviewer,                   foreign_key: {
to_table: :decidim_users }
      t.integer    :status,        null: false, default: 0
      t.datetime   :assigned_at,   null: false
      t.datetime   :due_date
      t.integer    :xp_earned,     null: false, default: 0
      t.decimal    :scicent_earned, precision: 10, scale: 2, default: 0
      t.text       :feedback
      t.timestamps
    end
    add_index :decidim_volunteer_scheduler_task_assignments, :assignee_id
    add_index :decidim_volunteer_scheduler_task_assignments, :status
  end
end
```

- `db/migrate/20250101000003_create_volunteer_profiles.rb` (VolunteerProfile):

```
class CreateVolunteerProfiles < ActiveRecord::Migration[6.1]
  def change
    create_table :decidim_volunteer_scheduler_volunteer_profiles do |t|
```

```
      t.references :user,     null: false, foreign_key: {
to_table: :decidim_users }
      t.references :referrer,             foreign_key: {
to_table: :decidim_users }
      t.string     :referral_code, null: false
      t.integer    :level,         null: false, default: 1
      t.integer    :total_xp,      null: false, default: 0
      t.decimal    :total_scicent_earned, precision: 10, scale: 2,
default: 0
      t.datetime   :last_activity_at
      t.timestamps
    end

add_index :decidim_volunteer_scheduler_volunteer_profiles, :referral_code,
unique: true
    add_index :decidim_volunteer_scheduler_volunteer_profiles, :user_id,
unique: true
  end
end
```

- `db/migrate/20250101000004_create_referrals.rb` (Referral):

```
class CreateReferrals < ActiveRecord::Migration[6.1]
  def change
    create_table :decidim_volunteer_scheduler_referrals do |t|
      t.references :referrer, null: false, foreign_key: {
to_table: :decidim_users }
      t.references :referred, null: false, foreign_key: {
to_table: :decidim_users }
      t.integer    :level,      null: false
      t.decimal    :commission_rate, precision: 5, scale: 4, null: false
      t.boolean    :active,     null: false, default: true
      t.decimal    :total_commission, precision: 10, scale: 2, default: 0
      t.timestamps
    end
    add_index :decidim_volunteer_scheduler_referrals,
[:referred_id, :level, :active], name:
"index_referrals_on_referred_level_active"
    add_index :decidim_volunteer_scheduler_referrals, :referrer_id
  end
end
```

- `db/migrate/20250101000005_create_scicent_transactions.rb` (ScicentTransaction):

```
class CreateScicentTransactions < ActiveRecord::Migration[6.1]
  def change
    create_table :decidim_volunteer_scheduler_scicent_transactions do |t|
      t.references :user,   null: false, foreign_key: {
to_table: :decidim_users }
      t.references :source, polymorphic: true
      t.integer    :transaction_type, null: false
      t.decimal    :amount, null: false, precision: 10, scale: 2
      t.integer    :status, null: false, default: 0
      t.timestamps
    end
    add_index :decidim_volunteer_scheduler_scicent_transactions,
[:user_id, :transaction_type, :status, :created_at], name:
"index_transactions_on_user_type_status_date"
  end
end
```

Each migration includes proper foreign keys and indexes (following Decidim's conventions [6] [5]). Adjust column types, constraints, and defaults as needed.

5. Core Models

Implement the models under `app/models/decidim/volunteer_scheduler/`, namespaced `Decidim::VolunteerScheduler`. Include associations and validations:

- `app/models/decidim/volunteer_scheduler/task_template.rb`:

```
module Decidim
  module VolunteerScheduler
    class TaskTemplate < ApplicationRecord
      belongs_to :component, class_name: "Decidim::Component"

      has_many :task_assignments, dependent: :destroy

      enum level:     { level1: 1, level2: 2, level3: 3 }
      enum frequency: { daily: 0, weekly: 1, monthly: 2, one_time: 3 }
      enum category:  { outreach: 0, technical: 1, administrative: 2,
creative: 3, research: 4, mentoring: 5 }

      validates :title,       presence: true, length: { maximum: 150 }
      validates :description, presence: true
      validates :xp_reward,   numericality: { greater_than: 0, less_than:
1000 }
      validates :scicent_reward, numericality: { greater_than_or_equal_to:
```

```
0 }
    end
  end
end
```

Dependencies: none extra. This follows the specification [7] . It automatically uses the table `decidim_volunteer_scheduler_task_templates` because of Rails naming in the engine.

- `app/models/decidim/volunteer_scheduler/volunteer_profile.rb`:

```
module Decidim
  module VolunteerScheduler
    class VolunteerProfile < ApplicationRecord
      belongs_to :user,     class_name: "Decidim::User"
      belongs_to :referrer, class_name: "Decidim::User", optional: true

      has_many :task_assignments, foreign_key: :assignee_id,
primary_key: :user_id
      has_many :referrals_made,  class_name:
"Decidim::VolunteerScheduler::Referral", foreign_key: :referrer_id,
primary_key: :user_id

      validates :referral_code, presence: true, uniqueness: true
      validates :level, inclusion: { in: 1..3 }

      # Constants for XP thresholds/capabilities (as per spec)
      LEVEL_THRESHOLDS = { 1 => 0, 2 => 100, 3 => 500 }.freeze
      LEVEL_CAPABILITIES = {
        1 => %w[basic_tasks],
        2 => %w[basic_tasks team_creation mentoring intermediate_tasks],
        3 => %w[basic_tasks team_creation mentoring intermediate_tasks
advanced_tasks team_leadership admin_tasks]
      }.freeze

      # Example method to handle leveling up
      def level_up_if_needed!
        new_level = calculate_level_from_xp
        return unless new_level > level

        update!(level: new_level)
        LevelUpNotificationJob.perform_later(user_id)
        unlock_capabilities(new_level)
      end

      def can_access_capability?(capability)
        LEVEL_CAPABILITIES[level].include?(capability.to_s)
```

```
      end

      # (Implementation of calculate_level_from_xp, unlock_capabilities
omitted for brevity)
    end
  end
end
```

Dependencies: none beyond `decidim-user`. Follows the spec [8] [9]. Ensure user model has `has_one :volunteer_profile` to link this (via a concern or patch).

- `app/models/decidim/volunteer_scheduler/referral.rb`:

```
module Decidim
  module VolunteerScheduler
    class Referral < ApplicationRecord
      belongs_to :referrer, class_name: "Decidim::User"
      belongs_to :referred, class_name: "Decidim::User"

      validates :level, inclusion: { in: 1..5 }
      validates :commission_rate, numericality: { greater_than: 0,
less_than_or_equal_to: 1 }

      # Commission rates per referral level (immutable)
      COMMISSION_RATES = { 1 => 0.10, 2 => 0.08, 3 => 0.06, 4 => 0.04, 5
=> 0.02 }.freeze

      def self.create_referral_chain(referrer, referred)
        transaction do
          current = referrer
          lvl = 1
          while current && lvl <= 5
            create!(
              referrer: current,
              referred: referred,
              level: lvl,
              commission_rate: COMMISSION_RATES[lvl],
              active: true
            )
            current = current.volunteer_profile&.referrer
            lvl += 1
          end
        end
      end
    end
```

```
  end
end
```

Dependencies: none. Implements the 5-level referral system as described [10] [11].

Each model is placed in the isolated engine namespace (`Decidim::VolunteerScheduler`) so that Rails and Decidim route queries to the correct tables (e.g. `decidim_volunteer_scheduler_referrals`). Write corresponding RSpec model tests under `spec/models/decidim/volunteer_scheduler/` as placeholders (e.g. `spec/models/decidim/volunteer_scheduler/task_template_spec.rb`), following Decidim's testing conventions.

6. Next Steps

With the basic engine, registration, migrations, and core models in place, the next deliverables include:

- **Remaining Models:** Implement `TaskAssignment` and any team/Scicent models.
- **Controllers & Views:** Add public and admin controllers (e.g. `TaskTemplatesController`, `AssignmentsController`, etc.) and associated views or cells for displaying task cards and dashboards.
- **Decidim Integration:** Register these controllers/routes in `config/routes.rb` (mounted under `decidim_volunteer_scheduler` and `decidim_volunteer_scheduler_admin`), and add permission checks (`decidim/volunteer_scheduler/permissions.rb`).
- **Background Jobs & Events:** Set up jobs (e.g. for commission calculations, level-up notifications) and Decidim Events (`task_assigned`, `level_up`, etc.) with notifications as sketched in the spec.
- **Settings & Localization:** Define component settings (e.g. referral levels, XP rewards) using `component.settings` hooks, and add I18n keys in `config/locales/*.yml` for any user-facing strings.
- **Testing:** Create RSpec factories and tests for each model, controller, and feature (matching Decidim's patterns).

Each step should follow Decidim's architecture and practices. For example, follow the pattern of existing modules (like Decidim Pages) when adding routes, menu entries, or permission scopes [4]. This phased approach ensures a clean, maintainable component aligned with Decidim's framework.

Sources: The above structure and code follow Decidim's official component patterns [4] [1] and the provided technical specification.

---

[1] [5] [6] [7] [8] [9] [10] [11] decidim_volunteer_scheduler_spec (5).md
file://file-7JxiMBEnHD8nLwsYGwnKLG

[2] [4] Components :: Decidim Docs
https://docs.decidim.org/en/develop/develop/components.html

[3] Menu :: Decidim Docs
https://docs.decidim.org/en/develop/customize/menu.html