FOREM
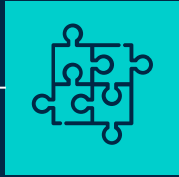
Community For Everyone

Group 05

01076024 Software Architecture Design
Semester 1, 2021

TABLE OF CONTENTS

Project Explanation

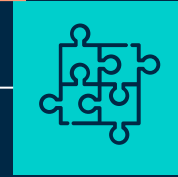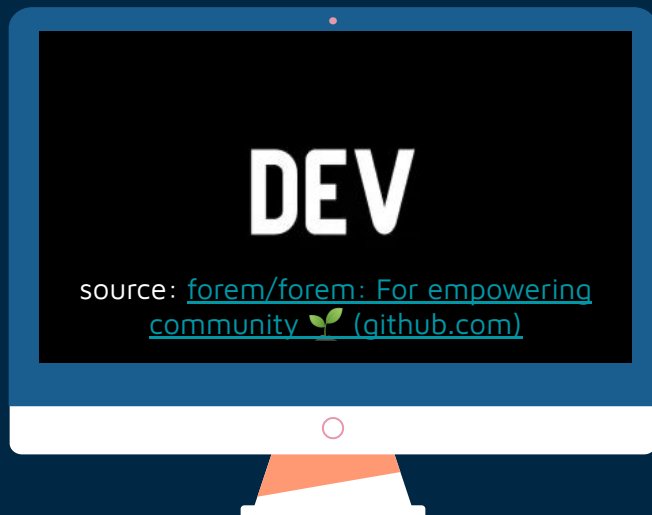01

Project Explanation

Forem is open source software for building communities. Communities for your peers, customers, fanbases, families, friends, and any other time and space where people need to come together to be part of a collective

source: forem/forem: For empowering community 🌱 (github.com)

Who use Forem?

It is a community of software developers who write articles, take part in discussions, and build their professional profiles.

source: forem/forem: For empowering community 🌱 (github.com)
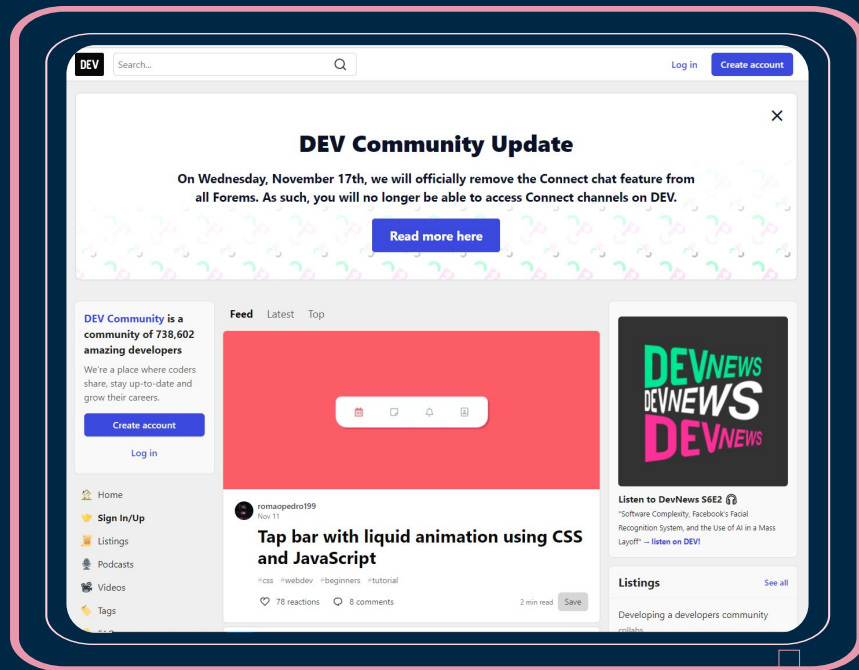
Software Architecture & UML

02

Architecture of Software Product

Model View Controller (MVC)

Architecture of Software Product

Quality Attribute

03

Quality Attribute

Usability

Scalability

Singleton

Weakness and Tactics

Usability

Supporting reason

We have methodically built certain important features while also rushing some things and throwing some things at the wall to see if they stick. That's the nature of building a software project, a community, and a company all at the same time.

Tactics

Forem builds various modules to use the Website from the very beginning.

Proofs

For Empowering Community – DEV Community (What does Forem do?)

Testability

Supporting reason

Forem can support many types of testing such as Acceptance Test Integration, Test Unit Test etc.

Tactics

Forem uses Rails's test system in Ruby language, and uses jest in Javascript and can also test some parts of the system for users to follow.

Proofs

Acceptance Testing

Compatibility

Supporting reason

Forem to be highly compatible with one another in order to build a cohesive ecosystem where members of multiple Forem can move seamlessly between community apps – even if those apps do not share any data about the individual.

Tactics

Forem builds its own class base, allowing models built on the platform to easily migrate between applications created by Forem.

Proofs

For Empowering Community – DEV Community (What is Forem?)

Weakness & Tactics

Weakness :

Performance

Description

Forems base on Ruby on Rails and Handle usage to finish by yourself that has issues when the website is heavily loaded.

Tactics

Add other support in Architecture such as static file service, load balance, database.

Design Pattern

04

Design Pattern

Decorator

Strategy

Compatibility
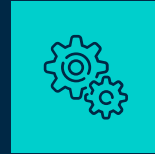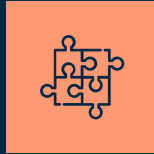
Builder

1. Decorator
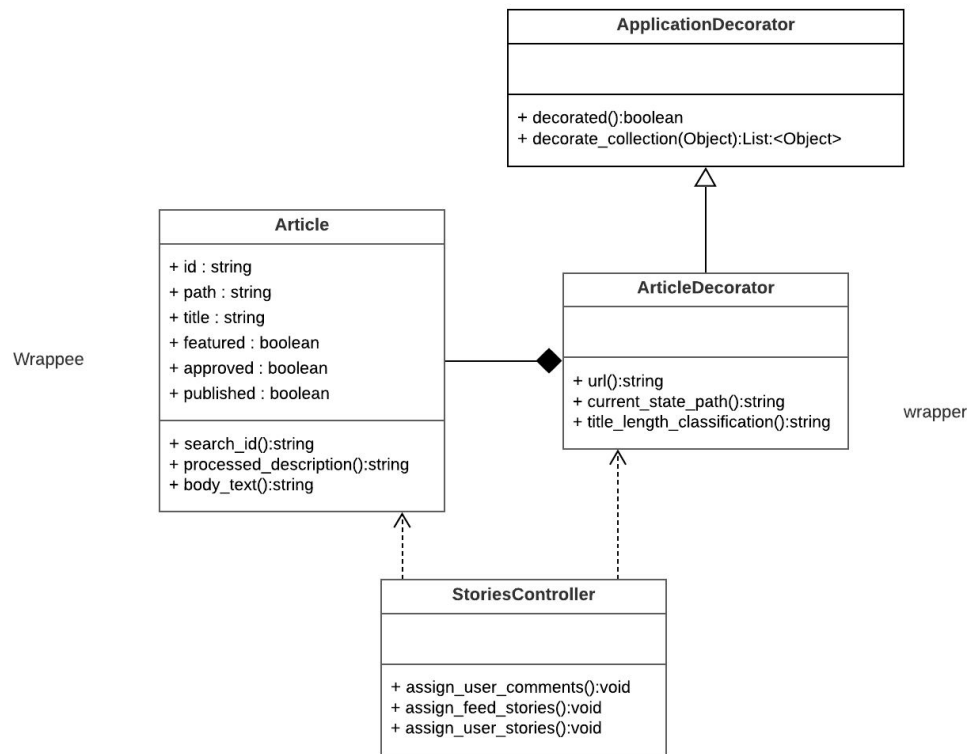
1. Decorator

```ruby
class ApplicationDecorator
  include ActiveModel::Serialization
  include ActiveModel::Serializers::JSON

  delegate_missing_to :@object

  # ActiveModel compatibility
  delegate :to_param, :to_partial_path, :current_page, :total_pages,
           :limit_value, :total_count, :entry_name, :offset_value,
           :last_page?, to: :@object

  attr_reader :object

  def initialize(object)
    @object = object
  end

  def decorated?
    true
  end

  def self.decorate_collection(objects)
    objects.map(&:decorate)
  end
end
```

1. Decorator

```ruby
class ArticleDecorator < ApplicationDecorator
  LONG_MARKDOWN_THRESHOLD = 900

  def current_state_path
    published ? "/#{username}/#{slug}" : "/#{username}/#{slug}?preview=#{password}"
  end

  def processed_canonical_url
    if canonical_url.present?
      canonical_url.to_s.strip
    else
      url
    end
  end

  def comments_to_show_count
    cached_tag_list_array.include?("discuss") ? 75 : 25
  end

  def cached_tag_list_array
    (cached_tag_list || "").split(", ")
  end
```

1. Decorator

```
163    def handle_organization_index
164      @user = @organization
165      @stories = ArticleDecorator.decorate_collection(@organization.articles.published
166        .limited_column_select
167        .order(published_at: :desc).page(@page).per(8))
168      @organization_article_index = true
169      set_organization_json_ld
170      set_surrogate_key_header "articles-org-#{@organization.id}"
171      render template: "organizations/show"
172    end
```

1. Decorator

Source file

Class ApplicationDecorator - Base class for decorator
- github.com/forem/forem/blob/main/app/decorators/application_decorator.rb

Class ArticleDecorator - Wrapper
- github.com/forem/forem/blob/main/app/decorators/article_decorator.rb

Class Article - Wrappee
- github.com/forem/forem/blob/main/app/models/article.rb

Usage
- github.com/forem/forem/blob/main/app/controllers/articles_controller.rb

- github.com/forem/forem/blob/main/app/controllers/stories_controller.rb
  (Line number : 165, 251, 307)

2. Strategy

ApplicationPolicy

+ show():boolean
+ create():boolean
+ updaet():boolean
+ edit():boolean
+ manage():boolean
+ destroy():boolean

Admin::ApplicationController

+ authorize_admin():void

InternalPolicy

+ access():boolean

ArticlePolicy

+ update():boolean
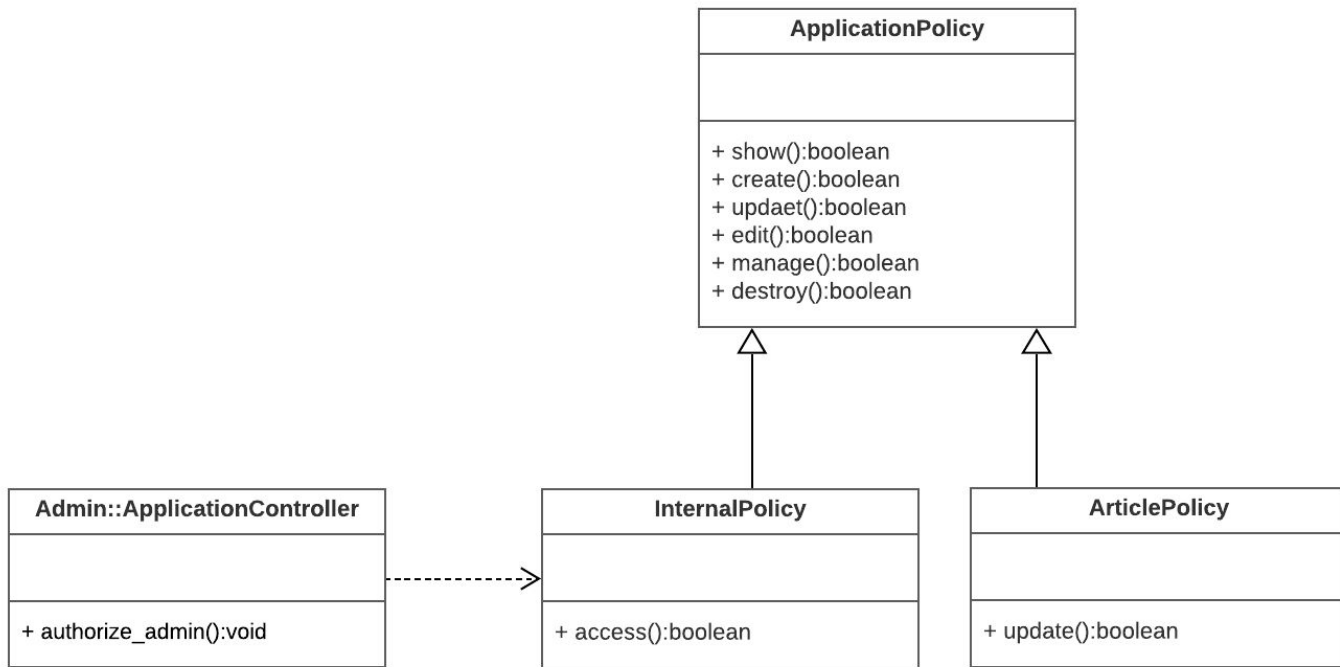
2. Strategy

```ruby
1   class ApplicationPolicy
2     attr_reader :user, :record
3
4     def initialize(user, record)
5       raise Pundit::NotAuthorizedError, "You must be logged in" unless user
6
7       @user = user
8       @record = record
9     end
10
11    def index?
12      false
13    end
14
15    def show?
16      scope.exists?(id: record.id)
17    end
18
19    def create?
```

2. Strategy

```ruby
class InternalPolicy < ApplicationPolicy
  def access?
    user.has_any_role?(
      { name: :single_resource_admin, resource: record },
      :super_admin,
      :admin,
    )
  end
end
```

2. Strategy

```ruby
class InternalPolicy < ApplicationPolicy
  def access?
    user.has_any_role?(
      { name: :single_resource_admin, resource: record },
      :super_admin,
      :admin,
    )
  end
end
```

2. Strategy

```
31    def authorize_admin
32      authorize(authorization_resource, :access?, policy_class: InternalPolicy)
33    end
```

2. Strategy

Source file

Class ApplicationPolicy - Base policy
- github.com/forem/forem/blob/main/app/policies/application_policy.rb

Class InternalPolicy
- github.com/forem/forem/blob/main/app/policies/internal_policy.rb

Usage
- github.com/forem/forem/blob/main/app/controllers/admin/application_controller.rb   (Line number : 33)

- github.com/forem/forem/blob/main/app/policies/article_policy.rb

3. Singleton

Campaign

- instacnce: Campaign

+ current(): Campaign
+ show_in_sidebar():boolean

SidebarsController

+ get_latest_campaign_articles() : void

Settings::Campaign

+ sidebar_image : string
+ sidebar_enabled : boolean

+ show_in_sidebar() : boolean
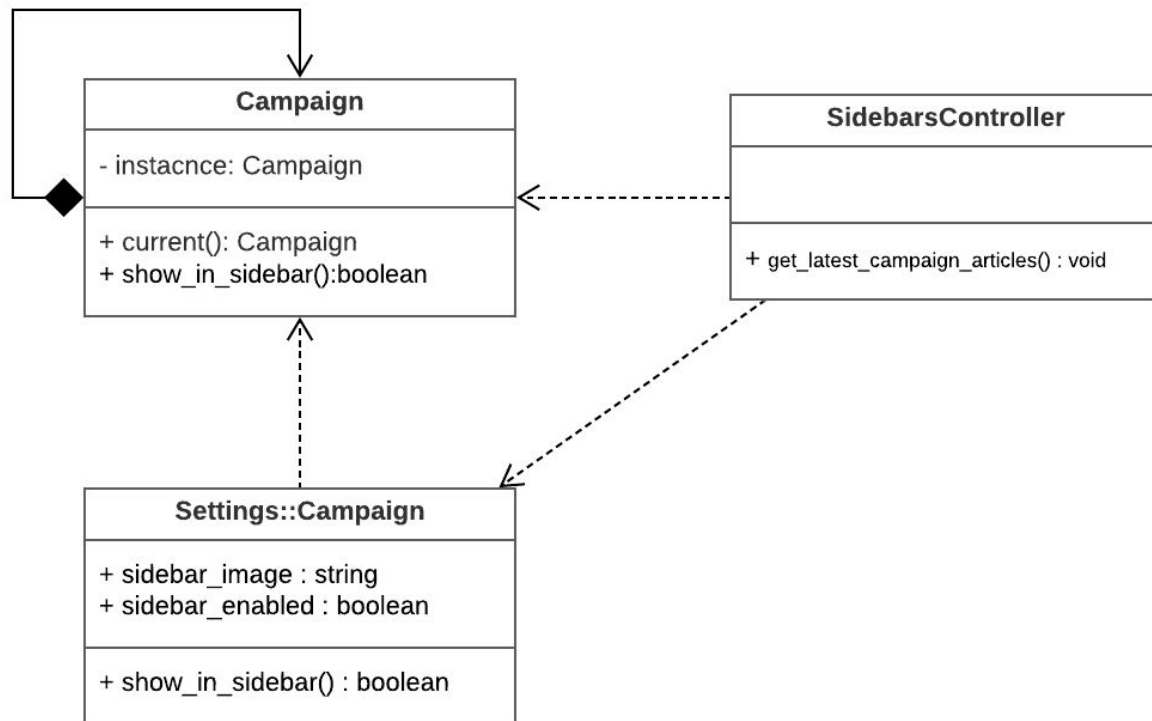
3. Singleton

```ruby
class Campaign
  include Singleton

  # Ruby's singleton exposes the instance via a method of the same name, but we
  # prefer a friendlier name.
  def self.current
    instance
  end

  METHODS = %w[
    articles_require_approval?
    call_to_action
    featured_tags
    hero_html_variant_name
    sidebar_enabled?
    sidebar_image
    url
  ].freeze

  delegate(*METHODS, to: Settings::Campaign)

  def show_in_sidebar?
    sidebar_enabled? && sidebar_image.present?
  end
end
```

3. Singleton

```ruby
module Settings
  class Campaign < Base
    self.table_name = :settings_campaigns

    # Define your settings
    setting :articles_expiry_time, type: :integer, default: 4
    setting :articles_require_approval, type: :boolean, default: 0
    setting :call_to_action, type: :string, default: "Share your project"
    setting :featured_tags, type: :array, default: %w[]
    setting :hero_html_variant_name, type: :string, default: ""
    setting :sidebar_enabled, type: :boolean, default: 0
    setting :sidebar_image, type: :string, default: nil, validates: { url: true }
    setting :url, type: :string, default: nil
  end
end
```

3. Singleton

```ruby
class SidebarsController < ApplicationController
  layout false
  before_action :set_cache_control_headers, only: %i[show]

  def show
    get_latest_campaign_articles
    set_surrogate_key_header "home-sidebar"
  end

  private

  def get_latest_campaign_articles
    campaign_articles_scope = Article.tagged_with(Campaign.current.featured_tags, any: true)
      .where("published_at > ? AND score > ?", Settings::Campaign.articles_expiry_time.weeks.ago, 0)
      .order(hotness_score: :desc)

    requires_approval = Campaign.current.articles_require_approval?
    campaign_articles_scope = campaign_articles_scope.where(approved: true) if requires_approval

    @campaign_articles_count = campaign_articles_scope.count
    @latest_campaign_articles = campaign_articles_scope.limit(5).pluck(:path, :title, :comments_count, :created_at)
  end
end
```
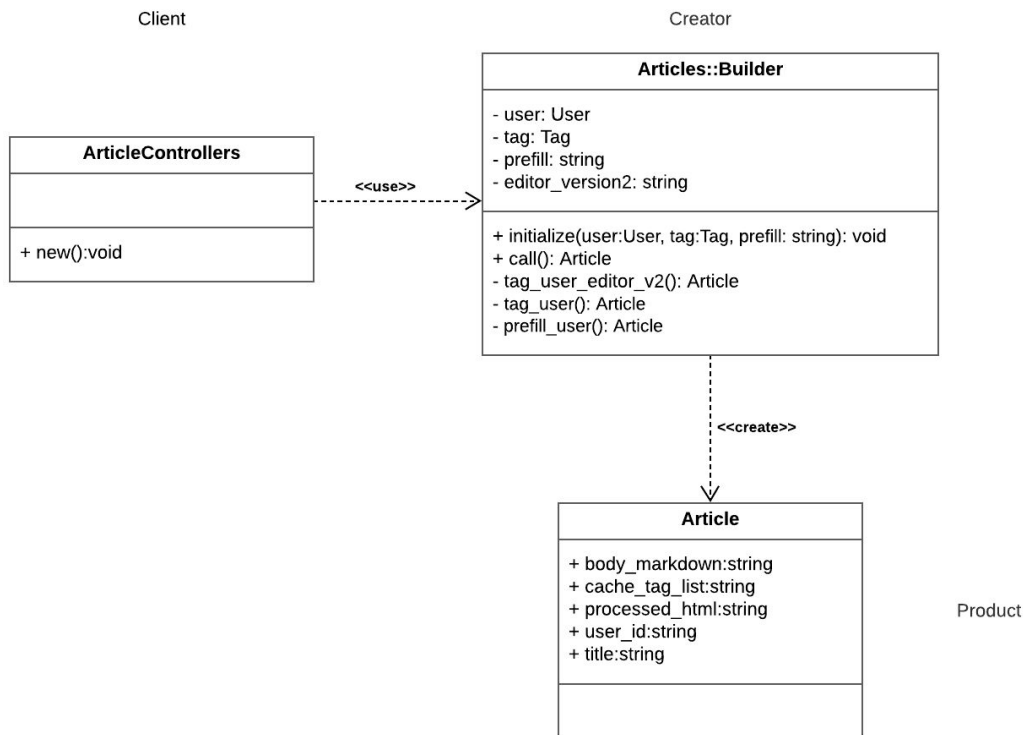
3. Singleton

Source file

Class Campaign - Singleton Class
- github.com/forem/forem/blob/main/app/models/campaign.rb
- github.com/forem/forem/blob/main/app/models/settings/campaign.rb

Usage
- github.com/forem/forem/blob/main/app/controllers/sidebars_controller.rb
  (Line number : 13-21)

4. Builder

Client

Creator

ArticleControllers

+ new():void

<<use>>

Articles::Builder

- user: User
- tag: Tag
- prefill: string
- editor_version2: string

+ initialize(user:User, tag:Tag, prefill: string): void
+ call(): Article
- tag_user_editor_v2(): Article
- tag_user(): Article
- prefill_user(): Article

<<create>>

Article

+ body_markdown:string
+ cache_tag_list:string
+ processed_html:string
+ user_id:string
+ title:string

Product

4. Builder

```ruby
54    def new
55      base_editor_assigments
56
57      @article, needs_authorization = Articles::Builder.call(@user, @tag, @prefill)
58
59      if needs_authorization
60        authorize(Article)
61      else
62        skip_authorization
63        store_location_for(:user, request.path)
64      end
65    end
66
```

4. Builder

```ruby
19    def call
20      return [tag_user_editor_v2, true] if tag && editor_version2
21      return [tag_user, true] if tag&.submission_template.present? && user
22      return [prefill_user_editor_v2, true] if prefill.present? && editor_version2
23      return [prefill_user, true] if prefill.present? && user
24      return [tag_article, false] if tag
25      return [user_editor_v2, false] if editor_version2
26
27      [user_editor_v1, false]
28    end
29
30    private
31
32    attr_reader :user, :tag, :prefill, :editor_version2
33
34    def tag_user_editor_v2
35      submission_template = tag.submission_template_customized(user.name).to_s
36
37      Article.new(
38        body_markdown: submission_template.split("---").last.to_s.strip,
39        cached_tag_list: tag.name,
40        processed_html: "",
41        user_id: user.id,
42        title: normalized_text(submission_template, "title:"),
43      )
44    end
45
46    def tag_user
47      Article.new(
48        body_markdown: tag.submission_template_customized(user.name),
49        processed_html: "",
50        user_id: user.id,
51      )
52    end
```

4. Builder

Source file

Class Builder - Creator
- github.com/forem/forem/blob/main/app/services/articles/builder.rb

Class Article - Product
- github.com/forem/forem/blob/main/app/models/article.rb

Usage
- github.com/forem/forem/blob/main/app/controllers/articles_controller.rb
  (Line number : 57)

Members

KORNTAWAT WITCHUVANIT

—61010023
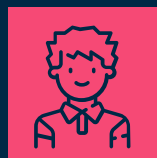

KITTISAK PHORMRAKSA

—61010092

NARAWIT TRIPROM

—61010556

KLANARONG TOMKOTR

—61010047

PIYATHIDA CHOOJAN

—61010656

PIYADA THONGTEM

—61010821

THANKS