



Universidad Nacional de Colombia

Workshop 2

Design Artifacts and System Modeling

Software Engineering II

Teacher: Carlos Andres Sierra Virguez

David Santiago Velasquez Gomez

Stiven Aguirre Granada

Juan Felipe Hernandez Ochoa

Sergio Alejandro Reita Serrano

David Andres Camelo Suarez

1. CRC Cards

Class: User	
Responsibility	Collaboration
<ul style="list-style-type: none">• Register on the website• Log in and log out• Edit profile: personal information associated startups• Recover password• Create, edit, or delete your own startups.• Comment and vote on startups	<ul style="list-style-type: none">• Startups• Comment• Vote• AuthManager

Class: Startup	
Responsibility	Collaboration
<ul style="list-style-type: none">• Store startup information• Display associated comments• Display associated vote calculation	<ul style="list-style-type: none">• User• Comment• Vote• SeachEngine

Class: Admin	
Responsibility	Collaboration
<ul style="list-style-type: none">• Delete startups or comments that violate rules• Monitor content• Manage users	<ul style="list-style-type: none">• User• Startup• Comment

Class: AuthManager	
Responsibility	Collaboration
<ul style="list-style-type: none">• Validate user credentials• Manage active sessions• Send password recovery emails• Encrypt passwords and verify their authenticity	<ul style="list-style-type: none">• User• EmailService

Class: Comment	
Responsibility	Collaboration
<ul style="list-style-type: none">• Create, edit, and delete comments• Display comments associated with a	<ul style="list-style-type: none">• User• Startup

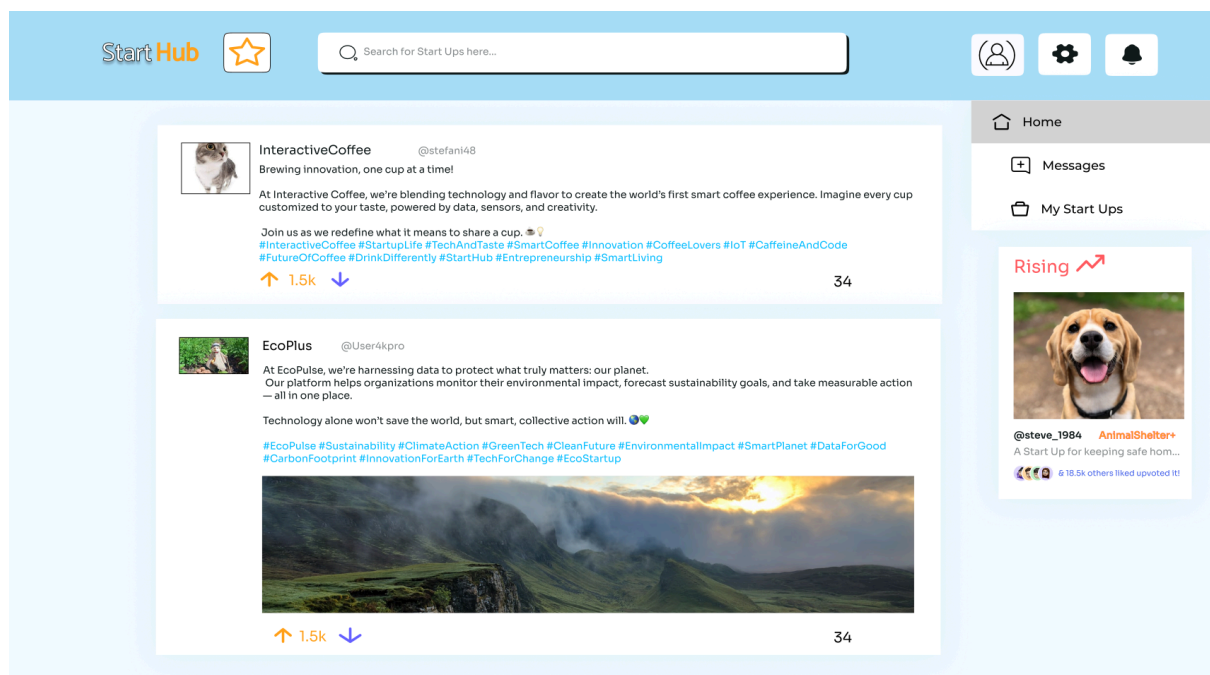
startup <ul style="list-style-type: none"> Verify permissions 	<ul style="list-style-type: none"> Admin
--	---

Class: Vote	
Responsibility	Collaboration
<ul style="list-style-type: none"> Record positive or negative votes Prevent duplicate votes from the same user Update the total vote count 	<ul style="list-style-type: none"> User Startup

Class: SearchEngine	
Responsibility	Collaboration
<ul style="list-style-type: none"> Search startups by name, category, or keyword Filter results by date, relevance, or votes Show thematic subforums 	<ul style="list-style-type: none"> UserInterface Startup Vote

2. Mockups

https://drive.google.com/file/d/1tqwh98fxnFF_OD7KPOQL0DZHGLSXi2Zs/view?usp=sharing



StartHub

Search for Start Ups here...

Aditi Agarwal

@AgarwalAdi004

2nd Year EKE Student at Arrancar University

210

Followers

200

Following

Edit Profile

Share Profile

Profile

StartUps

Awards

About

Hello! I am currently an Electronics and Computer Engineering student at Arrancar University, where I am eagerly immersing myself in the world of technology and innovation. With a keen interest in automotive engineering, UI/UX designing, data science, and machine learning.

...See More

My Resume

View

Education

Arrancar University > Bachelors Degree , Electronics and Computer Engineering

Community Awards

Star Platinum Club

Verified Pro - UI/UX

Lone wolf award

Arrancar University Hackathon 2024

NIT Trichy Research Presentation award

+8 More...

Home

Messages

My Start Ups

Personal settings

Security data

Log out

StartHub

Search for Start Ups here...

Home

Messages

My Start Ups

↑

1.5k

↓

EcoPlus

@User4kpro

#EcoPulse #Sustainability #ClimateAction #GreenTech #CleanFuture #EnvironmentalImpact #SmartPlanet #DataForGood #CarbonFootprint #InnovationForEarth #TechForChange #EcoStartup

34

5

Edit Information

See members

↑

8.9k

↓

NextPlay Studios

@Hideo Kojumbo

#GameDev #IndieGameDev #VideoGameDesign #GameDevelopment #IndieGames #GamingCommunity #PlayToCreate #GameDesign #Unity3D #UnrealEngine #GameArt #GameProgramming #LevelDesign #GameplayDesign

256

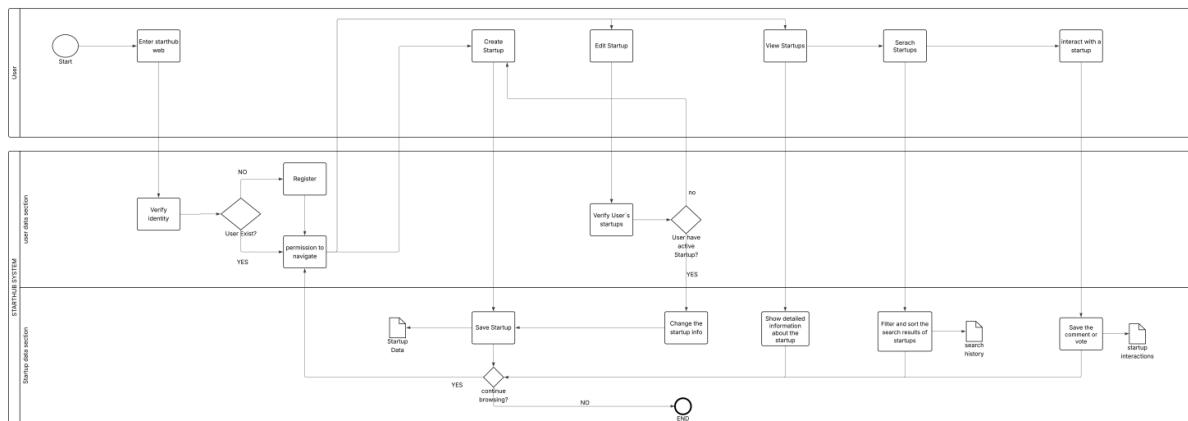
33

You're a member!

Create a new Start Up

3. Business Model Processes:

<https://drive.google.com/file/d/1Snm3OK66WsCv8irQJqFyVd5sWB2dQbZ1/view?usp=sharing>



The BPMN diagram represents the general workflow of the StartHub platform, illustrating how users interact with the system to manage and explore startups. The process begins when a user accesses the StartHub website, where the system verifies their identity to determine whether they are an existing user. If not, the user proceeds to register and obtain permission to navigate through the platform. Once authenticated, the user can perform multiple actions such as creating, editing, or viewing startups.

When a startup is created, its data is stored in the system's database, allowing future modifications and updates. Users can also search for startups by name, category, or keywords, and the system filters and sorts the search results to enhance discoverability.

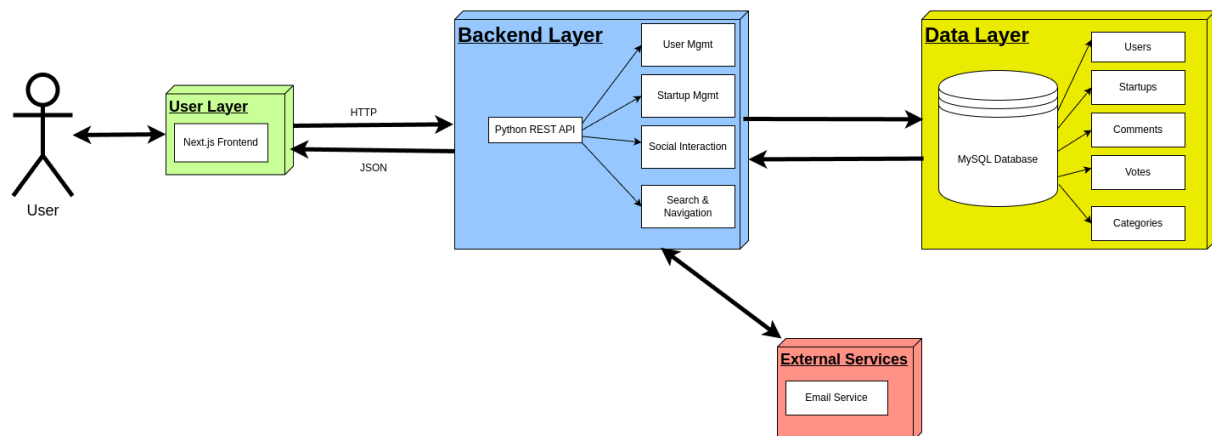
Additionally, users can interact with startup profiles by voting or posting comments, which are then saved and linked to the corresponding records. This BPMN process plays a central role in the application as it defines the logical and operational flow of user interaction, ensuring that authentication, data management, and social engagement occur in a structured and efficient manner.

It provides a clear visualization of how the system integrates user actions with backend processes, maintaining consistency, security, and a smooth user experience throughout the platform.

4. Architecture Diagram

https://drive.google.com/file/d/1NHdsd_serIMqO6XS1suBQFyjQJnXnCfg/view?usp=sharing

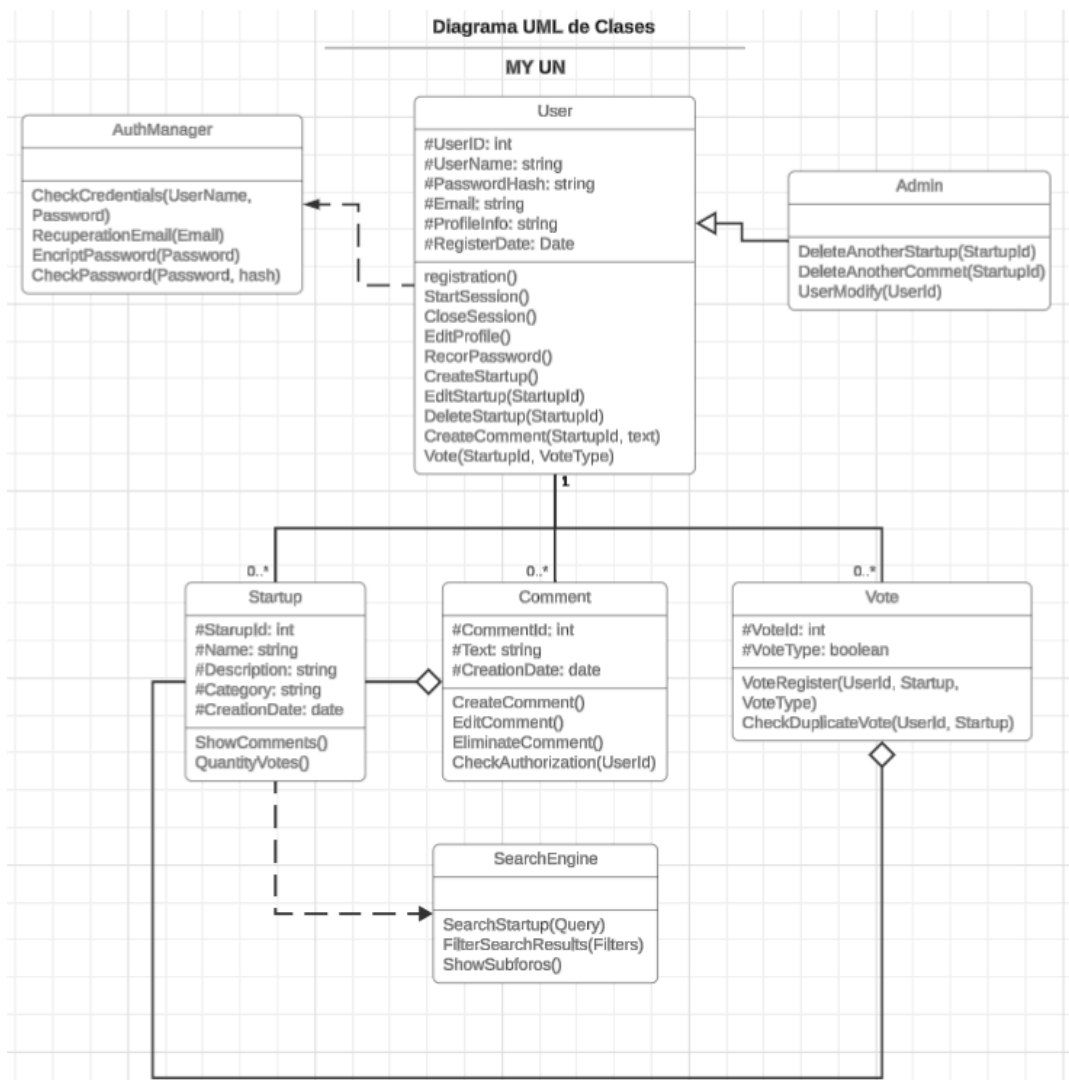
StartHub Architecture Diagram



The diagram illustrates the three main layers of the StartHub system architecture.

- The **User Layer** (Next.js frontend) handles user interaction through HTTP and JSON requests.
- The **Backend Layer** (Python REST API) manages business logic such as user management, startup management, social interaction, and search functions.
- The **Data Layer** (MySQL database) stores structured information about users, startups, comments, votes, and categories.
- The **External Services** like email notifications are integrated to enhance communication and automation.

5. Class Diagram



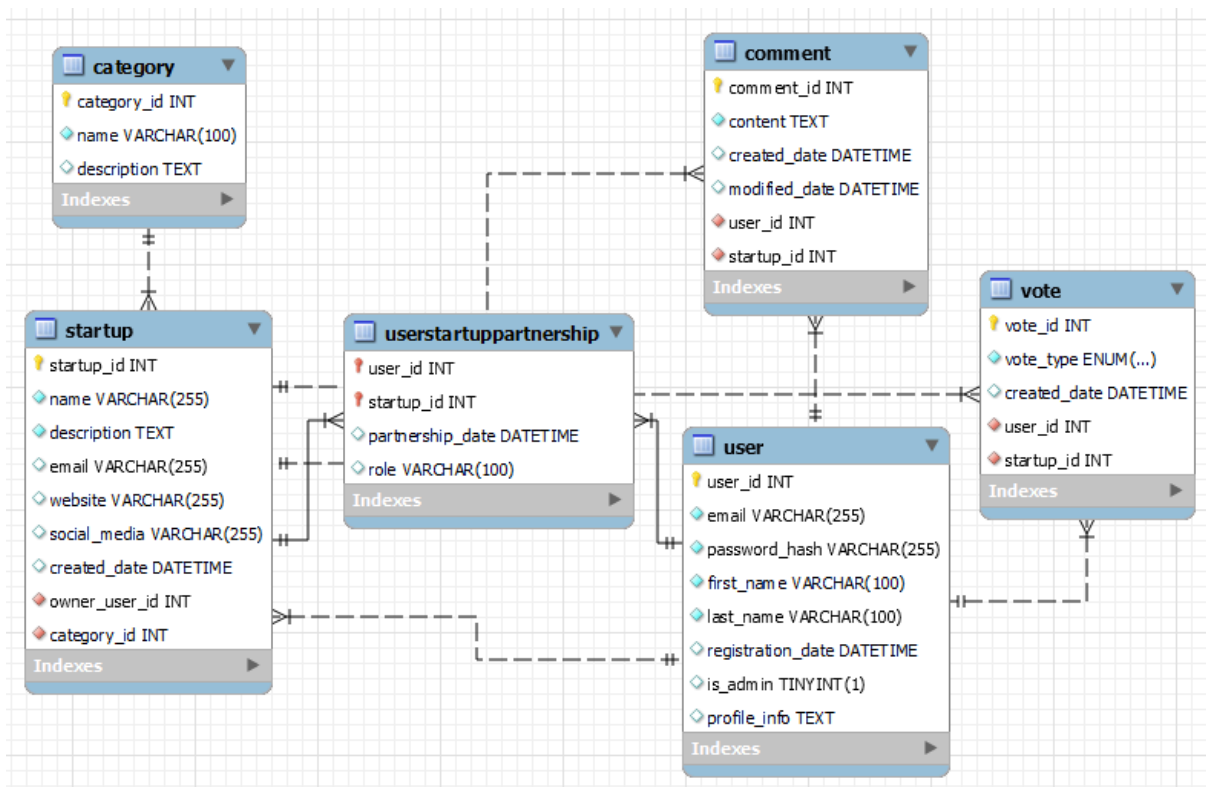
The diagram illustrates the main components, detailing the relationships, attributes, and methods of each class:

- **User:**
 - Handles registration, session management, password recovery, and profile editing.
 - Allows users to create, edit, and delete startups.
 - Enables interaction through comments and votes on startups.
- **Admin:**
 - Extends user capabilities with administrative permissions.
 - Can delete startups, remove comments, and modify user information.

- **Startup:**
 - Represents an entrepreneurial project within the system.
 - Stores key details such as name, description, category, and creation date.
 - Provides methods to display comments and count votes.
- **Comment:**
 - Manages user comments associated with startups.
 - Includes methods for creation, editing, deletion, and authorization checks.
- **Vote:**
 - Records user votes on startups.
 - Supports registering votes and checking for duplicates.
- **AuthManager**
 - Responsible for user authentication and password security.
 - Includes methods for credential verification and password encryption.
- **SearchEngine:**
 - Provides search and filtering functionalities for startups.
 - Allows users to browse and refine search results.

This class structure ensures **clear organization, security, and efficient user interaction** within the system.

6. Relational Database Model



Main Entities

1. User

This entity stores information about all system users.

The attributes include:

- **Unique identifier:** `user_id` (primary key)
- **Access credentials:** `email` and `password_hash` (encrypted)
- **Personal information:** `first_name`, `last_name`
- **Other fields:** `registration_date` (registration date) and `is_admin` (administrative permission flag)

This table implements the system requirements for registration, login, and profile management.

2. Category

Represents the sectors or thematic categories of startups (e.g., technology, health, education, energy).

It contains the following attributes:

- **Primary key:** `category_id`
- **Main attributes:** `name` (unique) and `description`

This separation complies with the **third normal form (3NF)**, eliminating transitive dependencies and facilitating the classification of startups by sector.

3. Startup

This is the **core entity** of the system, responsible for storing complete information about each registered startup.

Its main attributes are:

- **Primary key:** `startup_id`
- **General attributes:** `name`, `description`
- **Contact information:** `email`, `website`, `social_media`
- **Other fields:** `created_date` (creation date)

Foreign keys:

- `owner_user_id` → references **User** (startup owner)
- `category_id` → references **Category** (sector or theme)

This structure implements efficient **one-to-many** relationships between users, categories, and startups.

4. Comment

Stores comments made by users on startup posts.

Main attributes:

- **Primary key:** `comment_id`
- **Content and control:** `content`, `created_date`, `modified_date`

Foreign keys:

- `user_id` → references **User**
- `startup_id` → references **Startup**

This entity enables social interaction between system users and registered startups.

5. Vote

Implements the positive and negative voting system (**upvote** / **downvote**) for startups.

Relevant attributes:

- **Primary key:** `vote_id`
- **Vote type:** `vote_type` (ENUM('upvote', 'downvote'))
- **Date:** `created_date`

Foreign keys:

- `user_id` → references **User**
- `startup_id` → references **Startup**

Unique constraint: `UNIQUE(user_id, startup_id)` ensures that each user can only vote once per startup.

This entity supports the **startup rating module** within the platform.

6. UserStartupPartnership

A **junction table** that implements a **many-to-many (M:N)** relationship between users and startups, representing the “*partner startups*” described in the requirements.

Main features:

- **Composite primary key:** (`user_id`, `startup_id`)
- **Additional attributes:** `role`, `partnership_date`

Foreign keys:

- `user_id` → references **User**
- `startup_id` → references **Startup**

This table models collaborations, co-founders, or support relationships between users and startups.

Model Relationships

1. User → Startup (1:N)

A user can create multiple startups, but each startup belongs to a single owner.

Implemented through the foreign key **owner_user_id** in the **Startup** table.

2. Category → Startup (1:N)

A category groups multiple startups, but each startup belongs to only one category.

This relationship enables thematic navigation within the system.

3. User → Comment (1:N)

4. Startup → Comment (1:N)

A user can write multiple comments, and a startup can receive multiple comments.

Both relationships are implemented using foreign keys in the **Comment** table.

5. User → Vote (1:N)

6. Startup → Vote (1:N)

A user can cast multiple votes across different startups but only one per startup.

This is enforced by the **UNIQUE(user_id, startup_id)** constraint in the **Vote** table.

7. User ↔ Startup (M:N) via UserStartupPartnership

Allows users to be partners in multiple startups and vice versa.

Implemented through the **UserStartupPartnership** junction table containing foreign keys to both entities.

References

Olarte, L. M., & Sierra, C. A. (2025). *Systems design: Software engineering II* [PowerPoint slides]. Computer Engineering Program, School of Engineering, Universidad Nacional de Colombia.

[https://github.com/EngAndres/unal_public/blob/main/Software%20Engineering%20Morning%20\(G3\)/slides/1_SoftwareDesign.pdf](https://github.com/EngAndres/unal_public/blob/main/Software%20Engineering%20Morning%20(G3)/slides/1_SoftwareDesign.pdf)

Eseme, S. (2025, 29 agosto). *GraphQL vs REST: Todo lo que Necesitas Saber*. Kinsta®.

<https://kinsta.com/es/blog/graphql-vs-rest/>