



Gira Application

Spring Fundamentals Exam

GiraApp is a software application that allows teams to create and track issues and automate workflows.

There are several requirements you must follow in the implementation:

1. Database Requirements

The **Database** of the **Gira** application needs to support **3 entities**:

User

- **Id** – Accepts **UUID-String** or **Long** values
- **Username**
 - The **length** of the **values** should be **between 3 and 20** characters long (both numbers are **INCLUSIVE**)
 - The values should be **unique** in the database
- **Password**
 - The **length** of the **values** should be **between 3 and 20** characters long (**INCLUSIVE**)
- **Email**
 - The values should contain a '@' symbol)
 - The values should be **unique** in the database

Task

- **Id** – Accepts **UUID-String** or **Long** values
- **Name**
 - The **length** of the **values** should be **between 3 and 20** characters long (both numbers are **INCLUSIVE**)
 - The values should be **unique** in the database
- **Progress**
 - Option between **(OPEN, IN_PROGRESS, COMPLETED)**
- **Due Date**
 - A **date**, that cannot be in the past
- **Classification**
 - Many tasks can have one specification
- **Assigned to (User)**
 - Many tasks can be assigned to one user

Classification

- **Id** – Accepts **UUID-String** or **Long** values
- **Classification name**
 - an option between **(BUG, FEATURE, SUPPORT and OTHER)**
- **Description**
 - Fell free to add some description to every classification

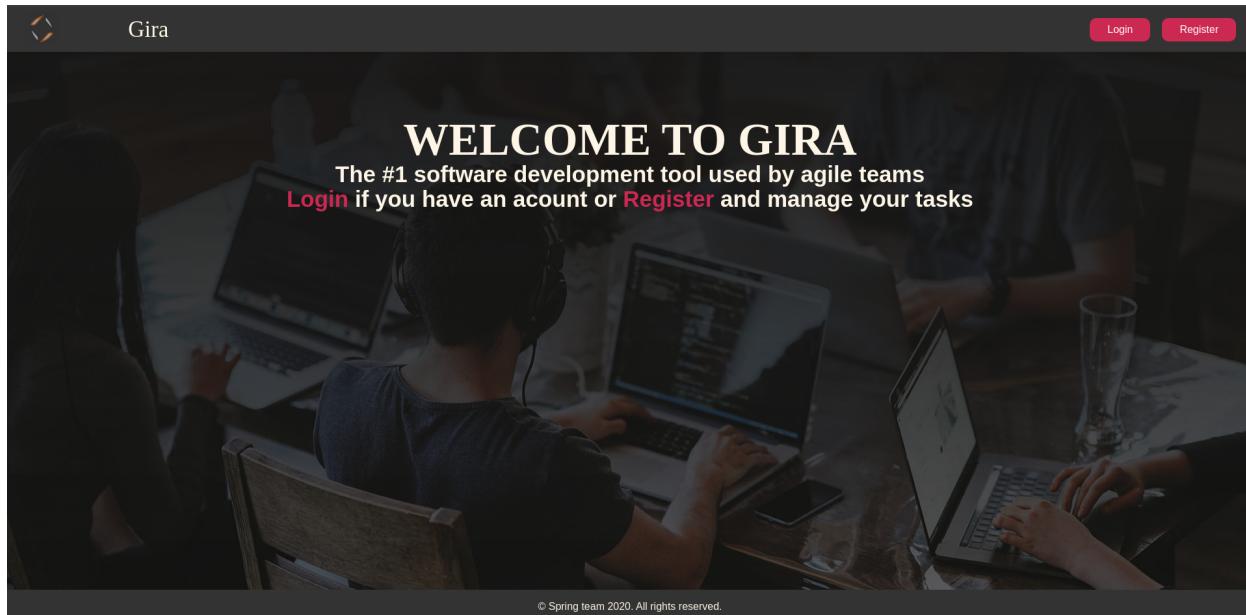
Nullable/Empty/Blank values are not allowed unless explicitly mentioned. Implement the entities with the **correct data types** and implement the **repositories** for them.

2. Initialize categories

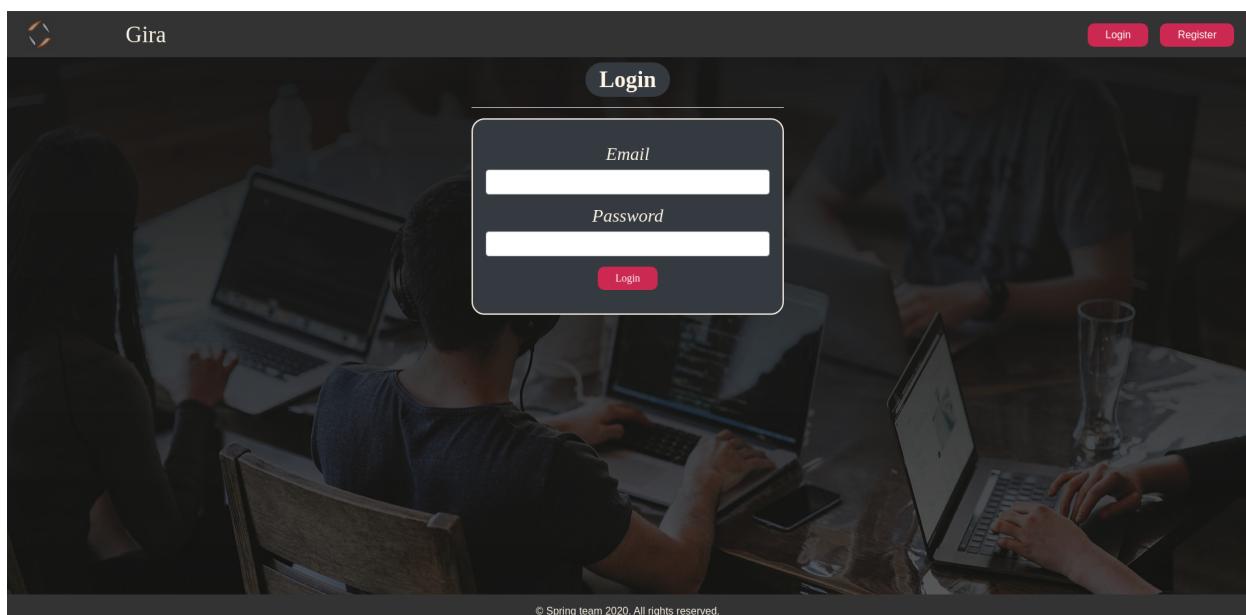
- Implement a method that checks (when app started) if the database does not have any category and initialize them
 - You are free to do this in some different ways
 - You can skip the description if you want

3. Page Requirements

Index Page (logged out user)



Login Page (logged out user)



Login Page validations

The image displays two side-by-side screenshots of a login interface. Both screenshots feature a dark background with a light gray header containing the word "Login".

Screenshot 1 (Left): Shows validation for incorrect email or password. The "Email" field contains "ivan@gmail.com" and has a red border. The "Password" field is empty and has a red border. A red error message at the top left says "Incorrect email or password!".

Screenshot 2 (Right): Shows validation for empty fields. Both the "Email" and "Password" fields have red borders. Red error messages at the top right say "Email can't be empty!" and "Password length must be between 3 and 20 characters!".

Registration Page (logged out user)

The image shows a registration form overlaid on a blurred background of people working on laptops. The registration form is titled "Register" and includes fields for "Username", "Email", "Password", and "Confirm Password", each with its own input field. A "Register" button is located at the bottom of the form. In the top right corner of the main window, there are "Login" and "Register" buttons. The top left corner features a logo with orange and white geometric shapes and the word "Gira".

© Spring team 2020. All rights reserved.

Registration Page validations

Register

Username

Username must be between 3 and 20 characters!

Email

Email cannot be empty!

Password

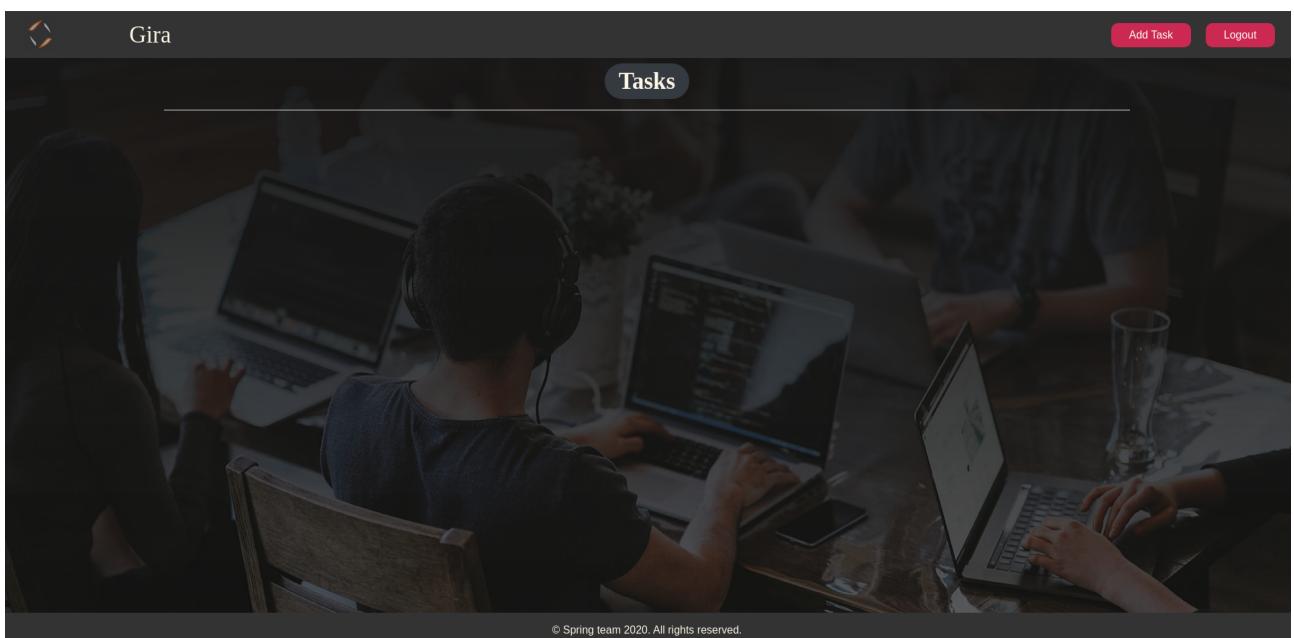
Password must be between 3 and 20 characters!

Confirm Password

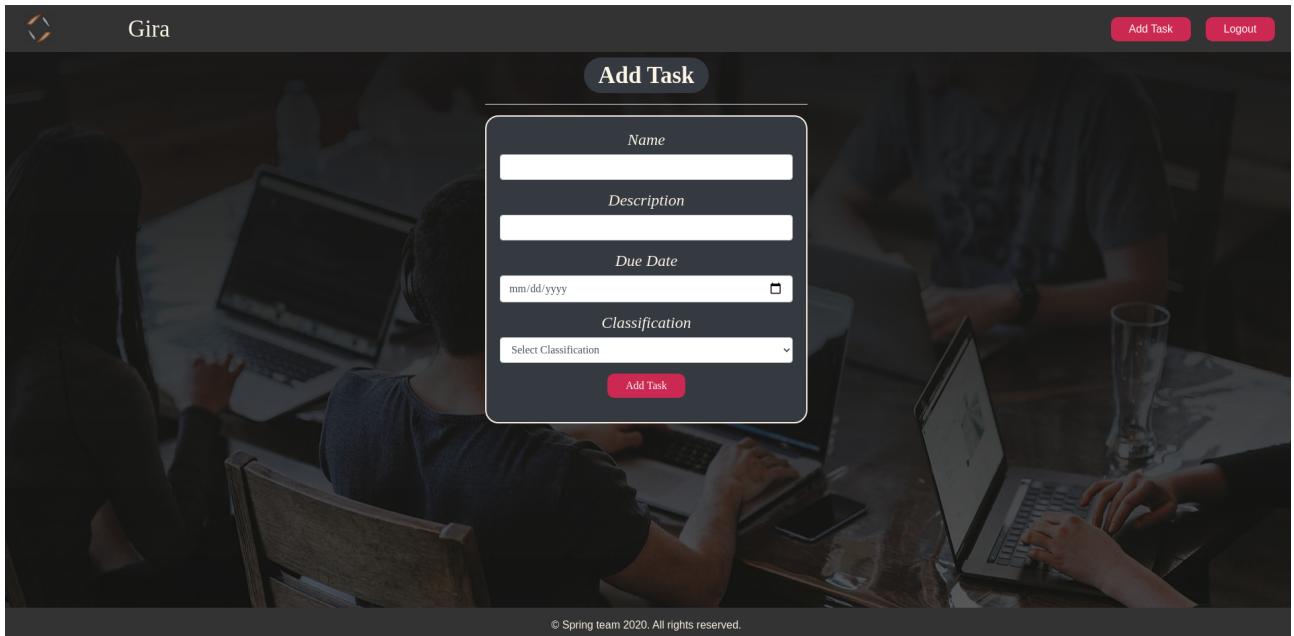
Register

Home Page (without having any tasks)

- Note: The home page should visualize **all of the tasks** from the database



Add Task



Gira

Add Task

Name

Description

Due Date
mm/dd/yyyy

Classification
Select Classification

Add Task

© Spring team 2020. All rights reserved.

This screenshot shows the 'Add Task' form on a web application. The form fields are: Name (empty), Description (empty), Due Date (06/30/2020), and Classification (empty). The 'Add Task' button is visible at the bottom. The background is a blurred image of people working on laptops.

Add Offer Validation

Add Task

Name

Name length must be between 3 and 20 characters characters!

Description

Description length must be more than 5 characters!

Due Date

06/30/2020

The date cannot be in the past!

Classification

Select Classification

Classification cannot be null!

Add Task

This screenshot shows the 'Add Task' form with validation errors. The 'Name' field is empty and highlighted in red. The 'Description' field is empty and highlighted in red. The 'Due Date' field contains '06/30/2020' and is highlighted in red, with an error message below it. The 'Classification' dropdown is empty and highlighted in red, with an error message below it. The 'Add Task' button is visible at the bottom.

Home Page (with tasks)

Tasks				
Name	Assigned to	Classification	Due Date	Progress
Dark mode	ivan	FEATURE	2021-01-01	OPEN Progress
New Laptop	ivan	SUPPORT	2020-08-30	IN_PROGRESS Progress
Login breaks	ivan	BUG	2020-08-14	COMPLETED Progress
Restock apples	ivan	OTHER	2020-08-07	OPEN Progress

The templates have been given to you in the application skeleton, so make sure you implement the pages correctly.

NOTES:

- The templates should look **EXACTLY** as shown above
- The templates do **NOT require additional CSS** for you to write. Only **bootstrap** and the **given CSS** are enough.

4. Functional Requirements

The Functionality Requirements describe the functionality that the application must support. The application should provide Guest (not logged in) users with the functionality to log in, register and view the Index page.

The application should provide **Users** (logged in) with the functionality to **log out**, **add a new Task (Add task page)**, **view all tasks (Home page)**. When creating a task, the "Assigned to" field should have **the person's name who created the task** and the initial progress should be "**OPEN**".

The **progress button** changes the progress column of the task:

- If the progress is "**OPEN**", the progress becomes "**IN_PROGRESS**".
- If the progress is "**IN_PROGRESS**", the progress becomes "**COMPLETED**".
- If the progress is "**COMPLETED**", **delete the task from the database**.

Gira in navbar should redirect to appropriate **URL depending** on that if the user is logged in.

The application should provide **functionality for adding tasks** with **classification (BUG, FEATURE, SUPPORT, OTHER)**.

The **task's classification** should have different colours. (The colours and the classes are in the end of the style.css file).

The **appropriate task's classification** should have the **appropriate style class**. The names are the same for the **css classes** and the **classification names**.

The application should store its data in a MySQL database.

5. Security Requirements

The Security Requirements are mainly access requirements. Configurations about which users can access specific functionalities and pages.

- **Guest** (not logged in) users can access:
 - **Index** page;
 - **Login** page;
 - **Register** page.
- **Users** (logged in) can access:
 - **Home** page;
 - **Add Task** page;
 - **Logout** functionality.

6. Scoring

Database – 10 points

Pages – 25 points

Functionality – 35 points

Security – 5 points

Validations – 15 points

Code Quality – 10 points