

- Barika Project management software
  - Design Document

## **1.Introduction**

**Team:** Guillermo Barros, Luc Anthony Nkounkou

**Reviewer:** Hamit Pena

## **Overview**

This document explains the main structure and design of Barika project management software illustrated below.

## **2. Context**

Barika is Java (Spring MVC) based web application developed to help a local company in need of a new tool to manage multiple projects, tasks, and team members.

The tool will be used by different teams in the company which include administrators, project managers and team members. Each project should be created by an administrator and assigned to a project manager. The project manager will create tasks and assign them to team members.

### **Product Features:**

#### **Admin Module**

- Login {/ Logout} to the system
- Create {/ View / Edit} users
- Create {/ Edit / Delete} projects
- Add/Remove project members / leader



**Product Features:****Project Leader Module**

- Login {/ Logout} to the system
- Create {/ View / Edit} tasks
- Add/Remove project members
- Assign tasks to project's member
- List tasks
- List Members
- See Member statistics

**Product Features:****Project's Member Module**

- Login {/ Logout} to the system
- View tasks
- Change tasks state
- List tasks
- List Members
- See own statistics

**4. Goal & Non-Goal**

The user driven impact of the project. For the non-goal we clearly describe which problems we won't be fixing just to get everyone on the same page.

The goal of this project is to facilitate the collaboration and coordination of projects, tasks and members inside of the company. The functionalities are as follow:

Feature	Description	Access Restriction
Login / Logout to the system	In order to enter to the system the user must login to the system by providing his username and password.	Administrator / Leader / Member
Create/ Remove / Edit users	This page must allow the administrator to manage the information about users. Allowing him to add a new user, edit or delete an existing user	Administrator
Create/ Remove / Edit projects	This page must allow the administrator to manage the information about projects. Allowing him to create a new project, edit or delete an existing project. The information to manage about projects is: <ul style="list-style-type: none"> <li>Title</li> <li>Description</li> <li>Start Date</li> <li>End Date</li> </ul>	Administrator
Add/Remove project members	This page should allow the administrator to add or remove project's members	Administrator / Leader
Add/Remove project leader	This page should allow the administrator to add or remove project's leader	Administrator

Feature	Description	Access Restriction
ListTaks	This option must display the list of all tasks assigned to a member	Project members
View a Task	This page must allow the member to see the detail of a task	Project members
ChangeTask state	A page to modify a task's state	Project leader / members

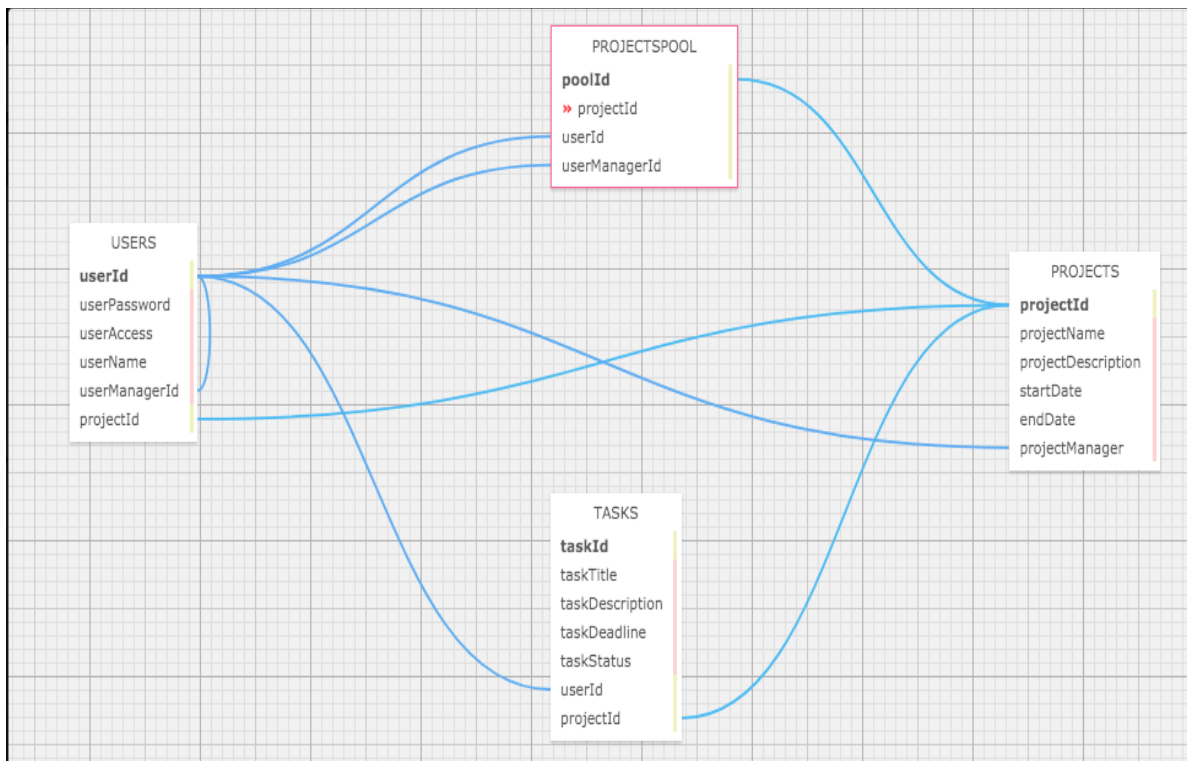
Feature	Description	Access Restriction
Create /View / Edit /Delete a Task	This page must allow the project leader to manage the information about tasks. Allowing him to create a new task, edit or delete an existing task. The information about tasks to be managed is: <ul style="list-style-type: none"> <li>Title</li> <li>Description</li> <li>Deadline</li> <li>State</li> </ul>	Project leader
List Taks	This option must display the list of all available tasks for a project	Project leader
Assign Task to a member	A page to assign a task to a member (s)	Project leader
List project's members	A page to show the list of all members in a given project	Project leader / members
Member's statistics	A page to show the detail of a member statistics. Including: <ul style="list-style-type: none"> <li>Assigned Tasks</li> <li>Completed Task (in time, out of time)</li> <li>Pending Tasks</li> </ul>	Project leader / members (only for themselves)

## 5. Our solution

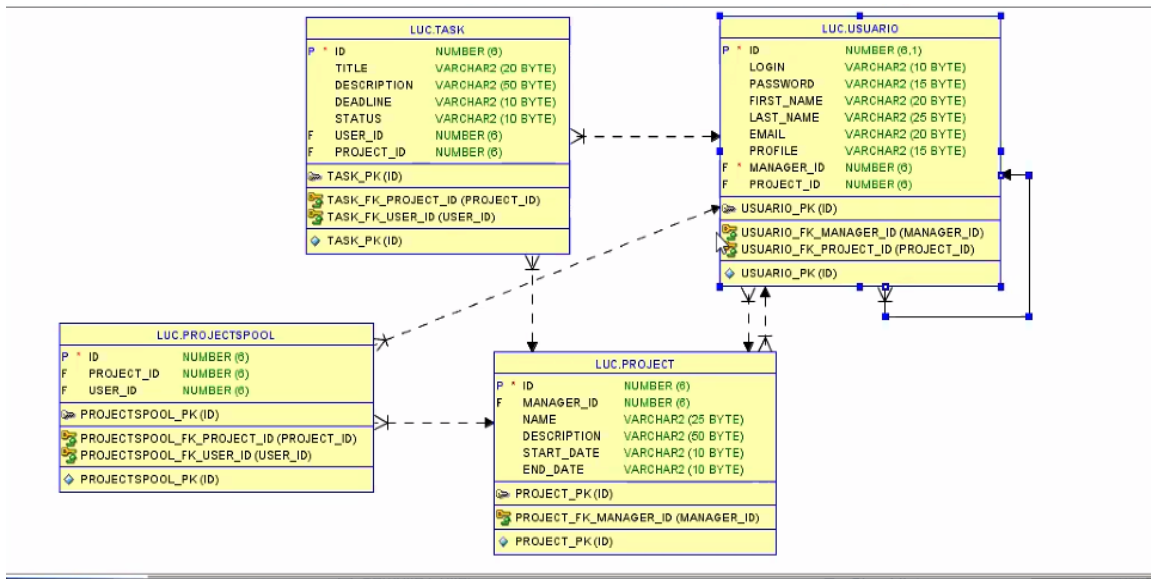
A full stack spring MVC web application.

### Database

#### a. DB Schema

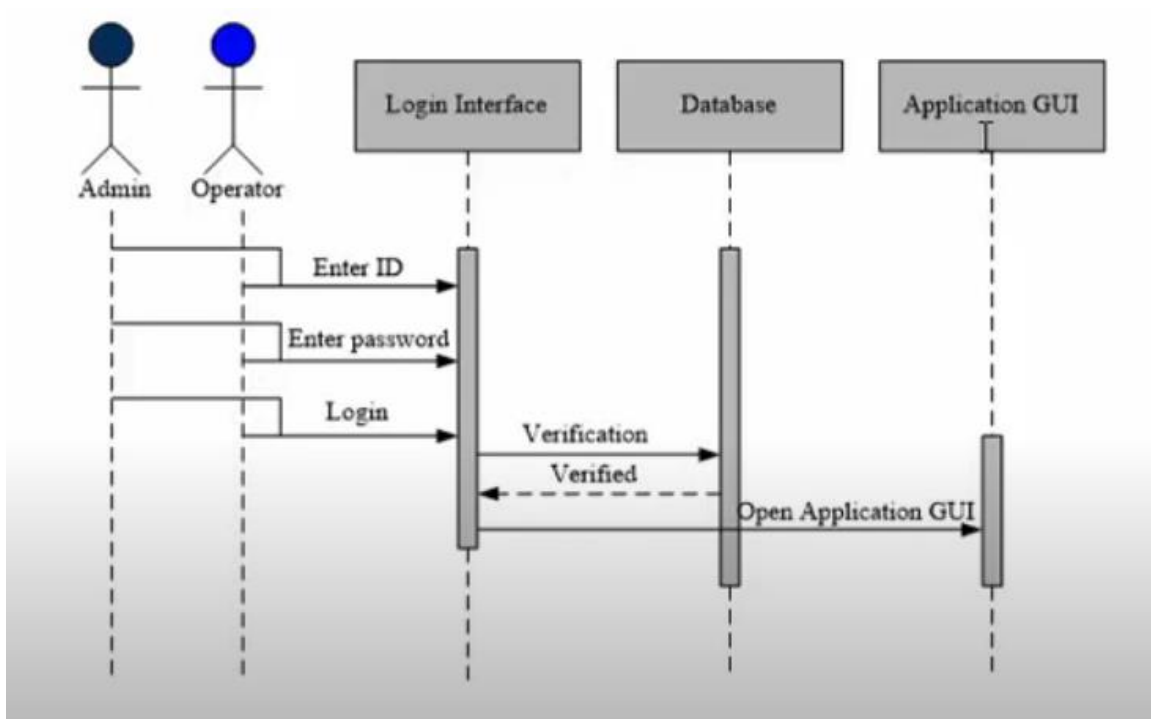


## b. MySQL database



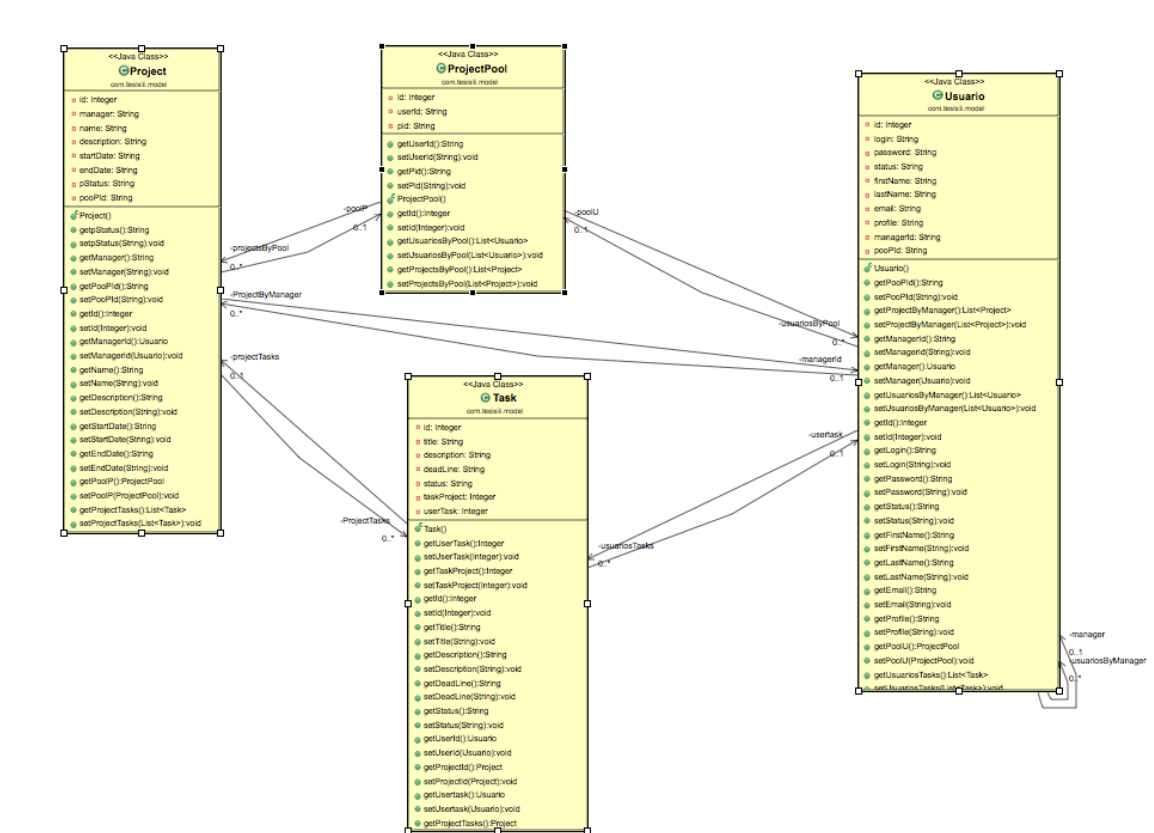
## Authentication

### c. System login [Spring Security module]

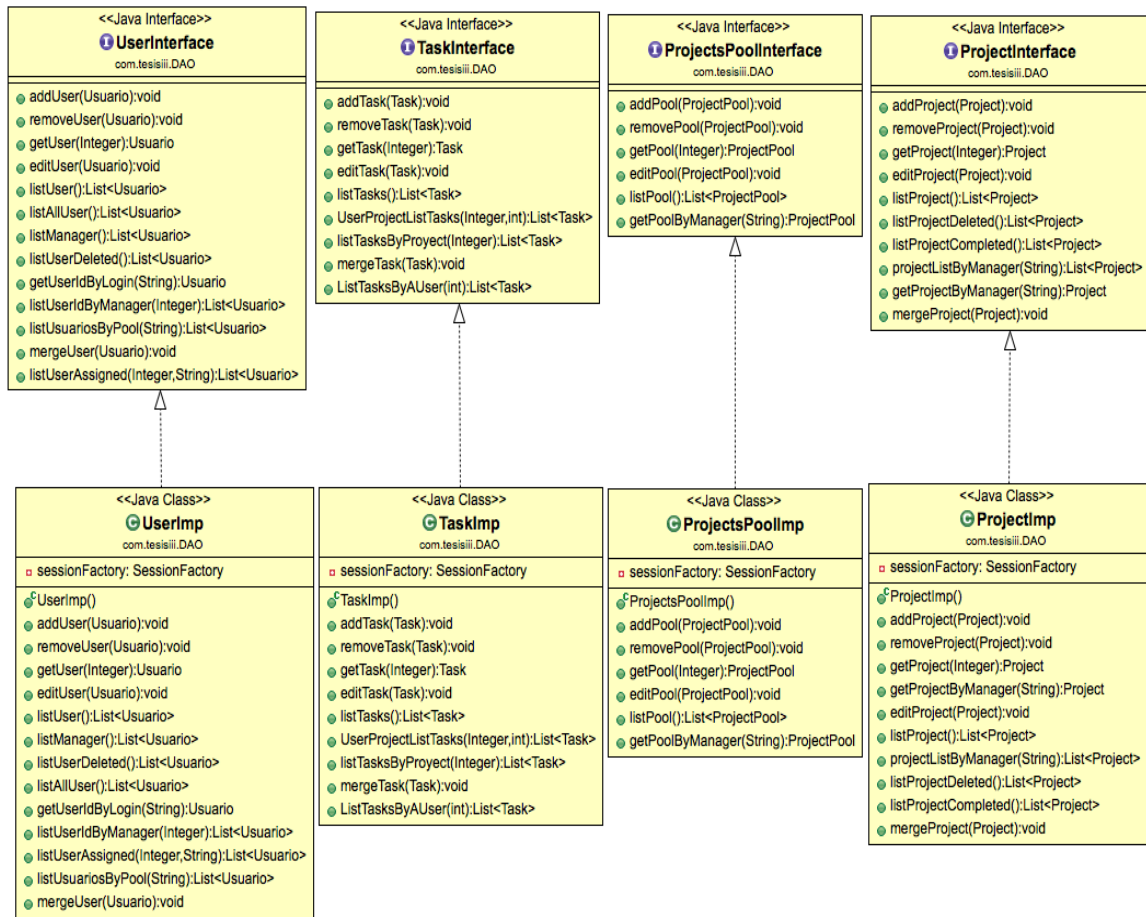


## System Architecture



#### d. Models







## e. DAO (Data Access Objects)





## f. Controllers

<<Java Class>>  <b>AdminController</b> com.tesisiii.controller	
serv: Services	
 AdminController() ● admin(Model):String ● currentUser(Authentication):String ● submitNewUser(Model):String ● removeUser(Usuario,Integer,BindingResult,Model):ModelAndView ● listAllUser(Model):String ● submitNewProject(Model):String ● editUser(Usuario,Integer,BindingResult,Model):ModelAndView ● editOrject(Project,Integer,BindingResult,Model):ModelAndView ● removeProject(Project,Integer,Integer,BindingResult,Model):ModelAndView ● editProject(Project,Integer,BindingResult,Model):ModelAndView ● editUserII(Usuario,Integer,BindingResult,Model):ModelAndView ● removeUserII(Usuario,Model):ModelAndView ● removeProjectII(Project,Integer,Model):ModelAndView ● submitNewProjectII(Project,BindingResult,Model):ModelAndView ● submitNewUser(Usuario,BindingResult,Model):ModelAndView ● AListAllProject(Project,BindingResult,Model):ModelAndView ● AListAllProjectCompleted(Project,BindingResult,Model):ModelAndView ● AListProjectDeleted(Project,BindingResult,Model):ModelAndView	

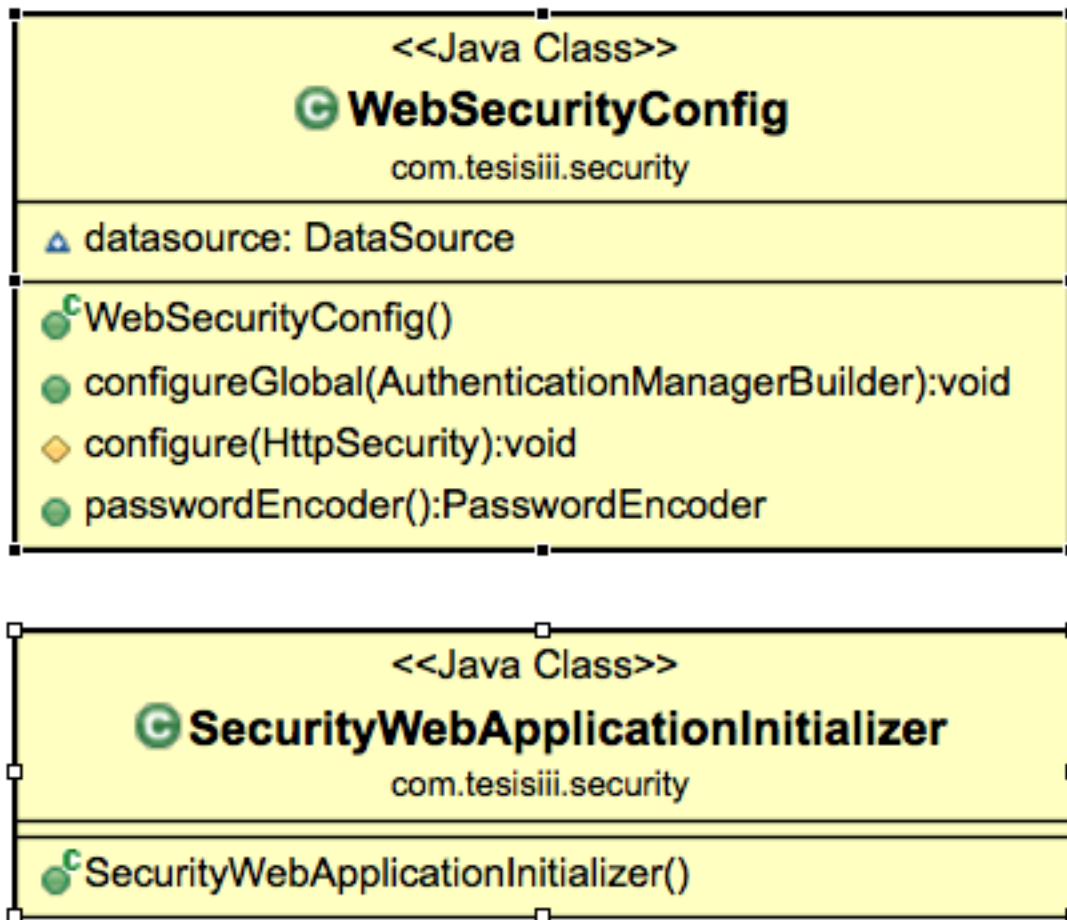
<<Java Class>>  <b>LoginController</b> com.tesisiii.controller	
 LoginController() ● getLoginView():String ● getDefaultAfterLogin(Authentication,Model):String	

<<Java Class>>  <b>PmController</b> com.tesisiii.controller	
serv: Services	
 PmController() ● pm(Authentication,String,Project,Model):String ● UsersByProject(Project,String,String,BindingResult,Model):ModelAndView ● pmAssigneUsersToProject(Project,Integer,String,BindingResult,Model):ModelAndView ● pmUnassaingUsersToProject(Project,Integer,String,BindingResult,Model):ModelAndView ● pmTasksUsersToProject(Task,String,int,int,String,BindingResult,Model):ModelAndView ● pmTaskByProject(Model,String,int):String ● pmTaskByProjectII(Task,Model,String,BindingResult):ModelAndView ● pmEditProject(Project,String,int,Model):ModelAndView ● editProjectII(Project,Integer,String,Model):ModelAndView ● pmEditTask(Task,String,int,int,Model):ModelAndView ● pmEditTaskII(Task,String,int,int,Model):ModelAndView ● pmRemoveTask(Task,String,int,Model):ModelAndView ● pmRemoveTaskII(Task,String,Model):ModelAndView	














<<Java Class>>  <b>UserController</b> com.tesisiii.controller	
serv: Services	
 UserController() ● user(Task,String,Model):String ● pmEditTask(Task,String,int,int,Model):ModelAndView ● pmEditTaskII(Task,String,int,int,Model):String	



g. Spring Security



## h. Configuration

<div>&lt;&lt;Java Class&gt;&gt;</div> <div> <b>HibernateConfig</b></div> <div>com.tesisiii.config</div>
<div> <b>HibernateConfig()</b></div> <div> sessionFactory():LocalSessionFactoryBean</div> <div> transactionManager(SessionFactory):HibernateTransactionManager</div> <div> dataSourceOracle():DataSource</div> <div>  hibernatePropertiesOracle():Properties</div>
<div>&lt;&lt;Java Class&gt;&gt;</div> <div> <b>ServletInitializer</b></div> <div>com.tesisiii.config</div>
<div> <b>ServletInitializer()</b></div> <div> onStartUp(ServletContext):void</div>
<div>&lt;&lt;Java Class&gt;&gt;</div> <div> <b>WebConfig</b></div> <div>com.tesisiii.config</div>
<div> <b>WebConfig()</b></div> <div> getViewResolver():ViewResolver</div>

## **8. Feedbacks & Discussion**

Any issue experienced with the app should be sent @: [lucbarika@gmail.com](mailto:lucbarika@gmail.com)

Also, new feature request can be submitted, and a technical evaluation will be done in order to decide if the feature can be implemented or not.

## **9. Project scope & timeline**

<b>Deliverables</b>	<b>Required Items</b>	<b>Duration</b>
Database Model	Database Schema Diagram	10 days
Architecture Model	Architecture Diagram	10 days
Project Documentation	User Stories, Architecture Model, Database Model	20 days