DOCUMENTATION

Introduction

ProjectX is a web application which supports the creation and management of software projects. Many choices and assumptions have been taken to achieve its development and they will be introduced in this document.

First of all it’s assumed that a project is divided in stages, which are macro sub-parts, and stages are divided in tasks: these are the actual activities to be carried out.

The users assume different roles in different projects :

* Project Manager
* Supervisor
* Developer

Every project is assigned to the project manager who created it, every stage is assigned to a supervisor to manage it and every task is assigned to one/more developers to be done.

There are bindings on the possibility of assume a role because it’s assumed that in a software company the employees are classified in three types:

* Project Manager
* Senior
* Junior

This distinction is important at the moment of the assignement of roles in a project since a Project Manager user can only be a Project Manager or a supervisor in a project, while a Senior user can only be a supervisor or a developer and a Junior can only be a developer.

ProjectX offers the possibility to:

* Sign up a user(ALL)
* Login (ALL)
* Access and visualize all projects which the user is working on(ALL)
* Create a new project(PM)
* Assign resources to a project(PM)
* Divide the project into stages(PM)
* Assign the stage to a supervisor(PM)
* Divide the stages into tasks(PM/SUPERVISOR)
* Assign developers to tasks(PM/SUPERVISOR)
* Manage and modify a project/stage/task(DEPENDING ON ROLE)
* Manage project delays(PM/SUPERVISOR)
* Search for similar project by subject area(ALL)
* Search for similar client by subject area(PM)
* Organize project meeting(PM)
* Organize stage meeting(PM/SUPERVISOR)

Moreover projectX keeps trace of the stages status and sends email to signal their start, finish or eventual delay. The actual email sending is simulated since it’s not supported by a SMTP server.

ANALYSIS

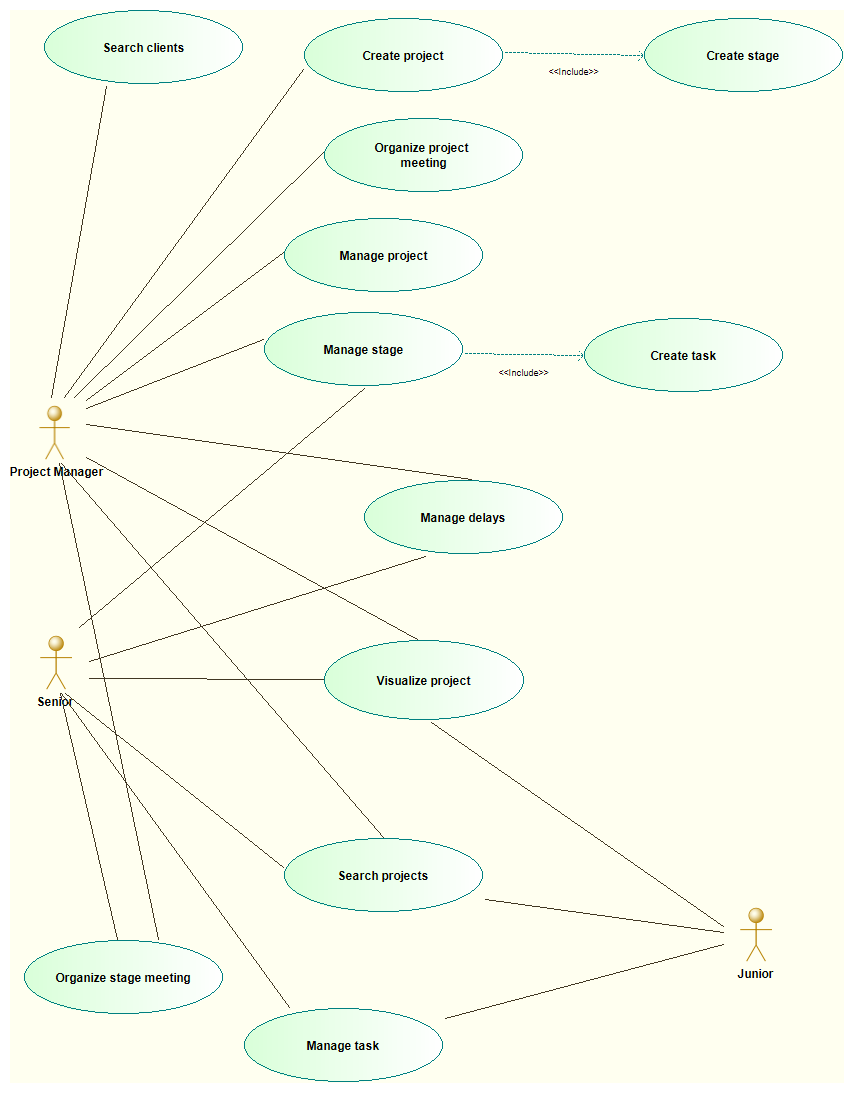
Requirements Analysis

The first stage of the application analysis was the outline of the principal requirements:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Type | Priority |
| 1 | The system should offer the possibility to create a new project | * Functional * UI * Project | M |
| 1.a | The system should offer the possibility to specify the requirements of the project | * Functional * UI * Project | M |
| 1.b | The system should offer the possibility to specify the goals of the project to be accomplished | * Functional * UI * Project | M |
| 1.c | The system should offer the possibility to establish a time estimation of the project | * Functional * UI * Project | M |
| 1.d | The system should offer the possibility to establish the costs of the project | * Functional * UI * Project | M |
| 2 | The system should offer a tool to manage the stages of the project | * Functional * UI * Stage | M |
| 2.a | The system should offer the possibility to specify the goals of the stage | * Functional * Stage | M |
| 2.b | The system should offer the possibility to establish the requirements of the stage | * Functional * UI * Stage | M |
| 2.c | The system should offer the possibility to check the results | * Functional * UI * Stage | M |
| 2.d | The system should notify the project manager about every stage of the project | * Functional * Stage | M |
| 3 | The system should offer the possibility to manage the human resources | * Functional * UI * Human resources | M |
| 3.a | The system should offer the possibility to manage the human resources of every project | * Functional * UI * Human resources * Project | M |
| 3.b | The system should offer the possibility to manage the human resources of every stages and to assign the roles of every person implied | * Functionality * Human resources * Stage | M |
| 4 | The system should offer the possibility to manage the physical resources | * Functional * UI * Physical resources | M |
| 5 | The system should offer the possibility to do outsourcing in case of lack of human and physical resources | * Functional * UI * Project * Stage * Resources | M |
| 6 | The system should offer the possibility to manage the project delays | * Functional * UI * Project | M |
| 7 | The system should offer the possibility to organize meetings | * Functional * UI * Meetings | M |
| 7.a | The system should offer the possibility to organize meetings among human resources of a stage | * Functional * UI * Meetings * Stage | M |
| 7.b | The system should offer the possibility to organize meetings with the client of the project | * Functional * UI * Meetings * Project * Clients | M |
| 8 | The system should keep trace of every clients who had/has interactions with it | * Functionality * Clients | M |
| 8.a | The system should offer the possibility to look for clients who had similar project request | * Functional * UI * Search * Clients | M |
| 9 | The system should offer the possibility to look for similar projects | * Functional * UI * Search * Project | M |
| 10 | The system should offer an algorithm for resources optimization | * Functional * Resources * Project | S |
| 11 | The system should manage the spread of human resources in various project with a security system | * Non Functional * Project | M |

Use Cases

The accurate requirements analysis led to the definitions of the use cases



Create Project Use Case - detailed:

|  |  |
| --- | --- |
| Title | Create project |
| Level | User |
| Primary actor | Project Manager |
| Preconditions | The user must be a Project Manager(PM); the pool of physical and human resources must have already been set by the IT manager.  The user must know all informations (duration of each stage, costs, physical resources etc…) about the project. |
| Postconditions | The new project is created, the creator is its Project Manager and he can manage every part of it. Each stage is defined and assigned to a Supervisor. |
| Main success scenario | 1. The system sets him as the PM by default 2. PM inserts the name of the project 3. PM inserts the requirements 4. PM inserts the goals of the project 5. PM inserts the budget and the estimated costs 6. PM inserts the start-day and the finish day 7. PM sets a pool of physical resources 8. PM creates a stage 9. PM names the stage 10. PM sets the goal and the requirements of the stage 11. PM sets the start and finish day of the stage 12. PM assignes a stage to a Supervisor 13. PM Repeats from point 8 until required 14. PM sets the precedences of the stage 15. The system calculates the critical stages 16. The system starts a thread that will keep trace of critical events(start/finish stage, delays…) in the project 17. PM saves the project |
| Extentions | * 1. The name of the project is already in use      1. The system displays an error message      2. The system asks to insert a different name   2. The system shows an amount of physical resources that the PM considers inadequate      1. The project manager indicates the company that will be the outsourcer and its contacts   3. The name of the stage is already in use in the project      1. The system displays an error message      2. The system asks to insert a different name   4. The system calculates an estimated duration that overcomes the deadline inserted by the PM      1. The system warns the PM that the deadline will never be respected |
| Frequency of use | It depends on the size of the company that uses the system. For instance, it will be once a month in big companies with lots of projects. |

Manage Project Use Case – detailed:

|  |  |
| --- | --- |
| Title | Manage Project |
| Level | User |
| Primary actor | Project Manager(PM) |
| Preconditions | The User must be the PM of the project |
| Postconditions | The project is modified |
| Main success scenario | 1. The PM enters in edit(manage) mode 2. Pm makes any modifications required to the project 3. Pm saves modifications |
| Extentions | * 1. PM needs to change the project name and the name is already in use      1. The system displays an error message      2. The system asks to insert a different name   2. During changes, some resources run out or the system shows a residual amount of human resources considerated inadequate by the PM      1. The project manager indicates the company that will be the outsourcer for that stage and its contacts |
| Frequency of use | Often (For example: 2/3 times per month) |

Manage Stage Use Case – detailed:

|  |  |
| --- | --- |
| Title | Manage stage |
| Level | User |
| Primary actor | Project Manager, Senior |
| Preconditions | The user must be the supervisor of the stage. |
| Postconditions | The stage is now divided in tasks, its human resources have been divided on every tasks. |
| Main success scenario | 1. The supervisor creates a task 2. The supervisor assigns the name to the task 3. The supervisor inserts the duration and the start day 4. The system return a list of people available for the task period 5. The supervisor assignes the task to people whose skills are considerated adequate to it 6. The supervisor repeats from point 4 until required 7. The supervisor saves the changes |
| Extentions | * 1. Supervisor modifies an existing task   2. The name of the task is already in use in the same stage      1. The system displays an error message      2. The system asks to insert a different name   3. There aren’t available human resources for the task or the number of human resources isn’t enough      1. The system displays an error message      2. The system offers the possibility to do outsourcing by notifing the PM that will contact an outsourcer company   4. During a stage the system shows an amount of human resources that the PM considers inadequate      1. The project manager indicates the company that will be the outsourcer for that stage and its contacts |
| Frequency of use | Often |

Manage Delays Use Case – detailed:

|  |  |
| --- | --- |
| Title | Manage delays |
| Level | User |
| Primary actor | Project Manager, Senior |
| Preconditions | The user must be the supervisor or the PM |
| Postconditions | The task which caused the delay is extended, the actual duration of the project is updated |
| Main success scenario | 1. The system notifies the supervisor that a non critical delay occurred in his stage 2. The supervisor extends the task which is causing the delay 3. The system shows an updated list of available human resources for the extension 4. The supervisor assigns the human resources 5. The supervisor saves the changes |
| Extentions | * 1. The system notifies the supervisor and the PM that a critical delay occurred in the supervisor’s stage      1. The PM moves forward the deadline of the project      2. The system updates the status of the project in the DB. |
| Frequency of use | Few times, in case of delay |

Manage Task Use Case – informal:

|  |  |
| --- | --- |
| Title | Manage a task |
| Primary Actors | Senior, Junior |
| Main Success Scenario | 1. The user enters into a task 2. The user reports that he completed his work 3. The system checks if all users involved in the task have completed their work 4. The system sets the task’s state as “completed “ 5. The system updates the stage’s rate which the task belongs to |
| Extentions | * 1. The total amount of work hasn’t been completed yet      + 1. The system reports to the user that not every users have completed their work   2. The stage’s rate achieves 100%      + 1. The system notify to the stage’s supervisor that it’s completed |

Organize Project Meeting Use Case – informal:

|  |  |
| --- | --- |
| Title | Organize projects meeting |
| Primary Actors | Project Manager(PM) |
| Main Success Scenario | 1. PM enters into a project 2. PM creates a new meeting 3. PM inserts the object and a message 4. PM submits the request of meeting 5. The system sends an email to the client of the project |
| Extentions | 1. The PM wants to invite supervisors too    1. PM signals that he wants to invite all supervisors    2. The system sends an email to every supervisor |

Organize Stage Meeting Use Case – informal:

|  |  |
| --- | --- |
| Title | Organize stage meeting |
| Primary Actors | Project Manager(PM), Senior |
| Main Success Scenario | 1. The user enters into a stage 2. The user creates a new meeting 3. The user inserts the object and a message 4. The user submits the request of meeting 5. The system sends an email to every person involved in a stage |
| Extentions | 1. The supervisor wants to invite the project manager too    1. The supervisor signals that he wants to invite the PM    2. The system sends an email to PM |

Search Projects Use case – informal:

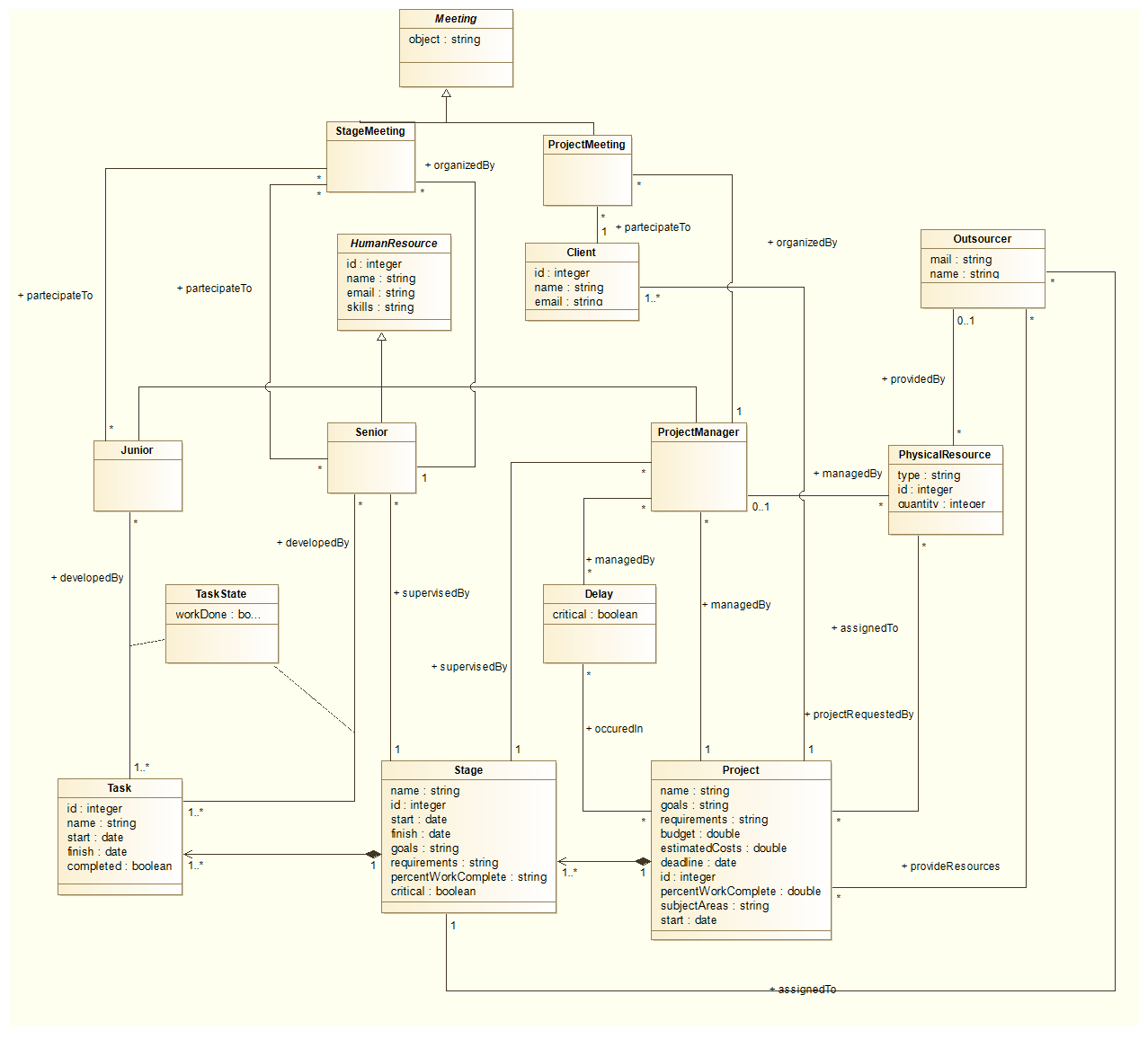
|  |  |
| --- | --- |
| Title | Search project |
| Primary Actors | Project Manager(PM), Senior, Junior |
| Main Success Scenario | 1. The user enters into the search mode 2. The user selects projects research 3. The user inserts the subject area to look for similar projects 4. The system returns a list of projects with the name of the project and its PM |
| Extentions | 1. There are no similar projects and the system shows a “Not Found” message |

Search Clients Use Case – informal:

|  |  |
| --- | --- |
| Title | Search Clients |
| Primary Actors | Project Manager(PM) |
| Main Success Scenario | 1. The user enters the search mode 2. The user selects client research 3. The user inserts the subject area to look for clients who asked for projects of that area 4. The system returns a list of clients with the requested profile and their contacts |
| Extentions | 1. There are no clients who has that kind of request and the system shows a “Not Found” message |

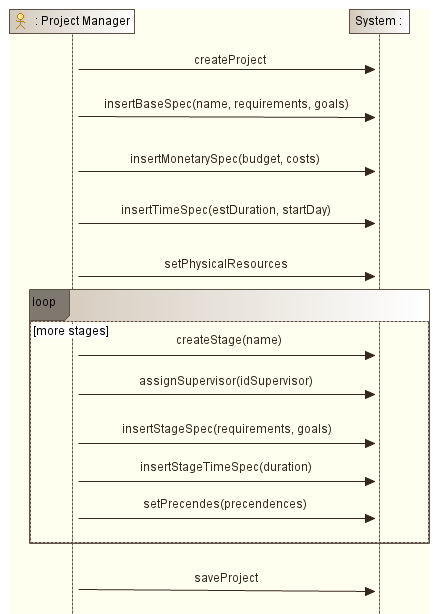
Domain Model

The domain model diagram offers a quick and immediate overview on the business logic of the application:

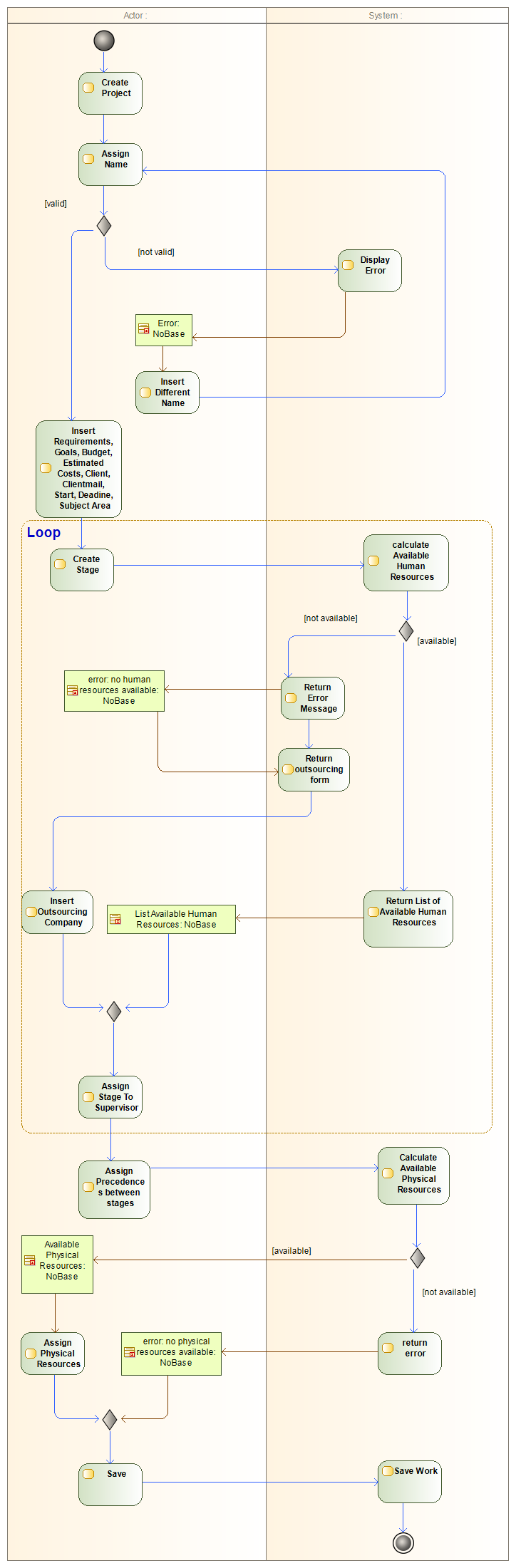


SSD Diagram

To better understand the flow of the “Create Project” use case, here’s its diagram:



Activity Diagram

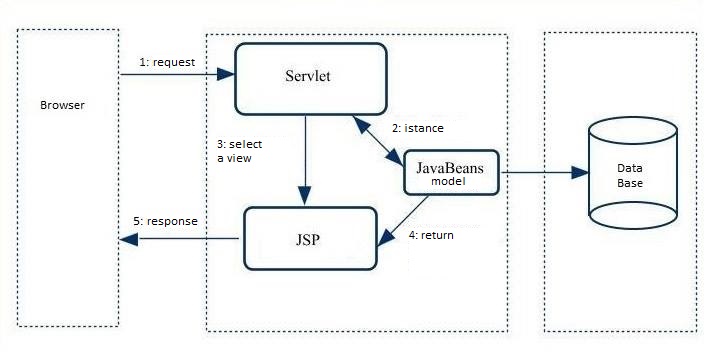
Another diagram to understand the complexity of “Create Project”:

DESIGN

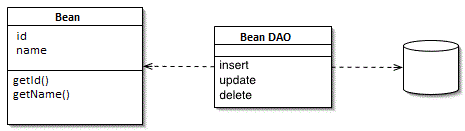
After the first analysis, project design started.

Architecture

ProjectX is a web application based on the architectural pattern MVC(Model-View-Controller) and JSP technology. Its persistence lays on a MySQL Database and the communication between the business logic classes and the DB is managed by the DAO (Data Access Object) architectural pattern.

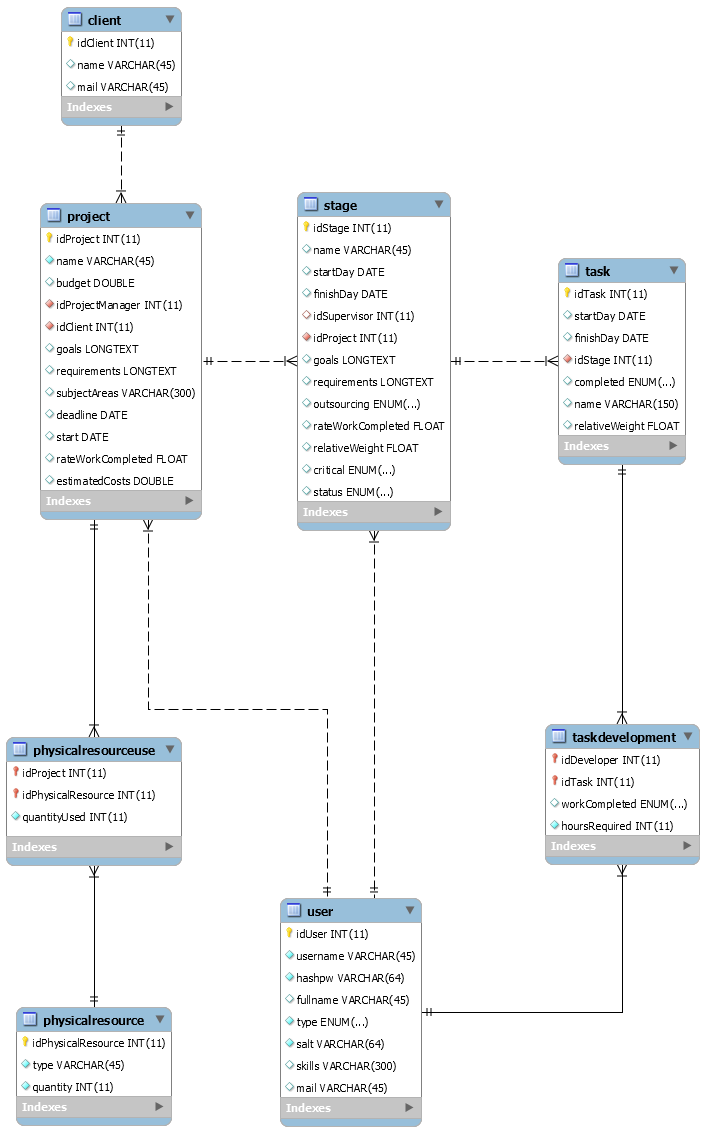


Data Mapper Diagram

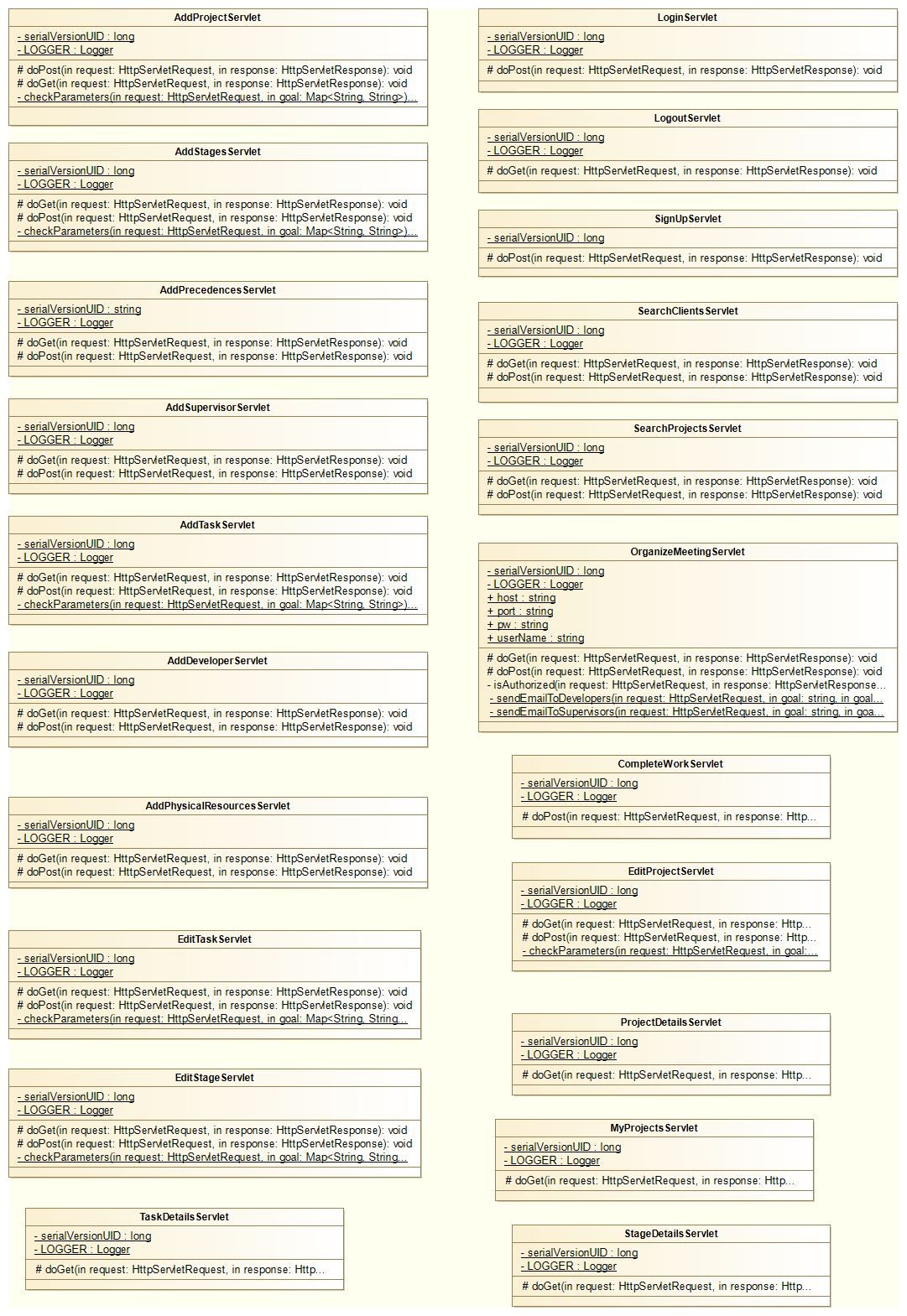


ER Diagram

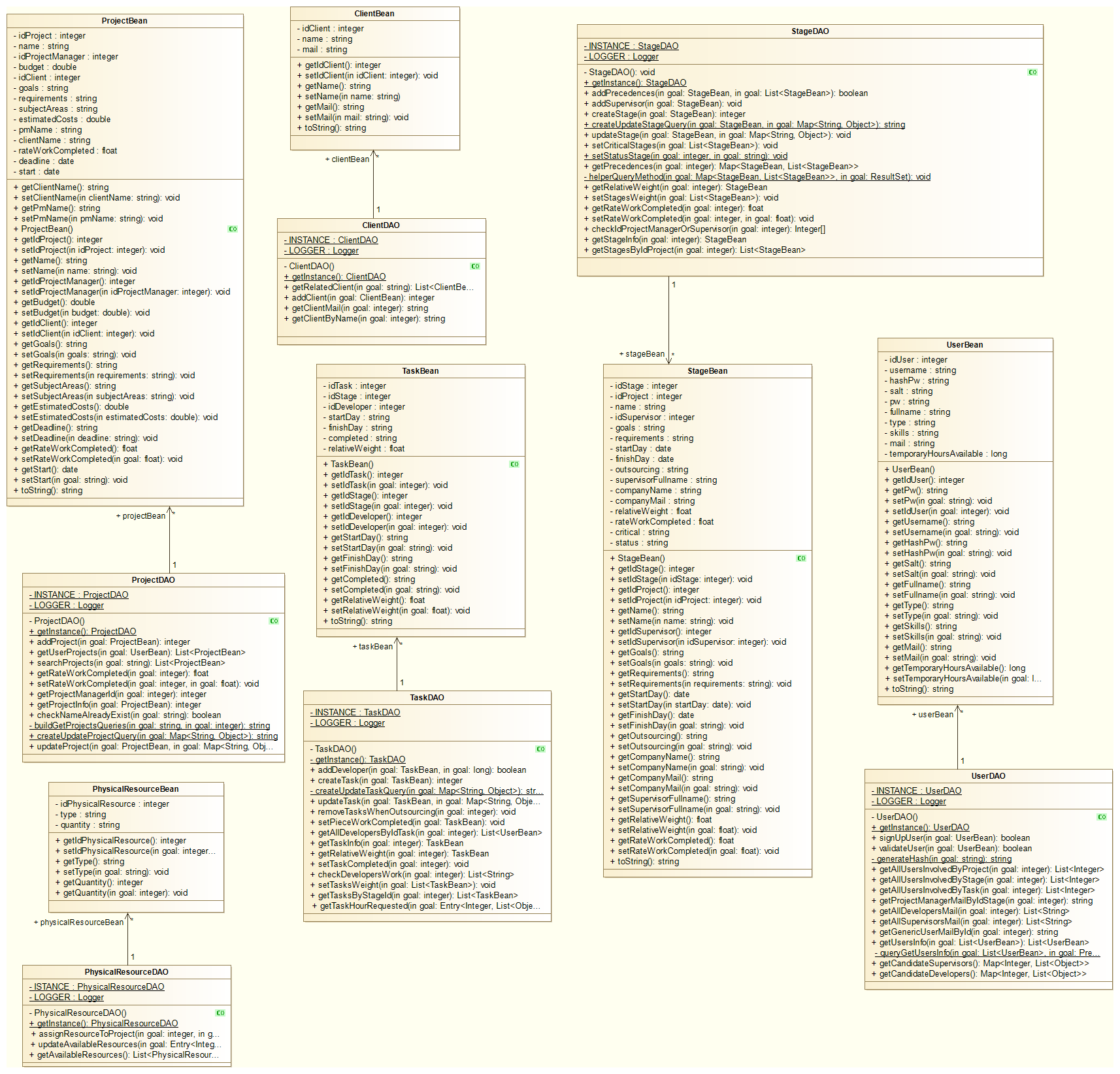
The entity-relationships diagram illustrates the structure of the relational database:



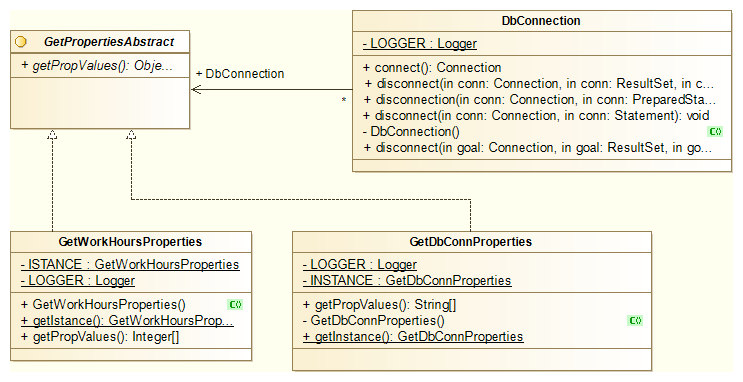
Design class diagrams

Package: controllers

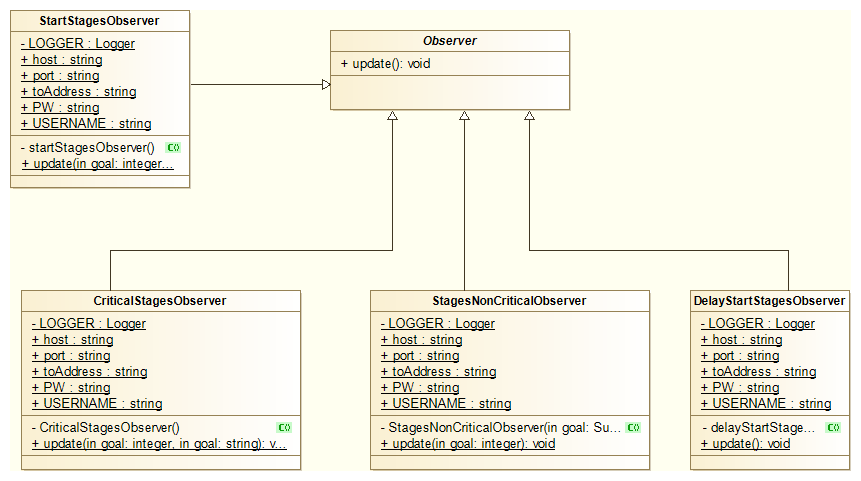
Package: models



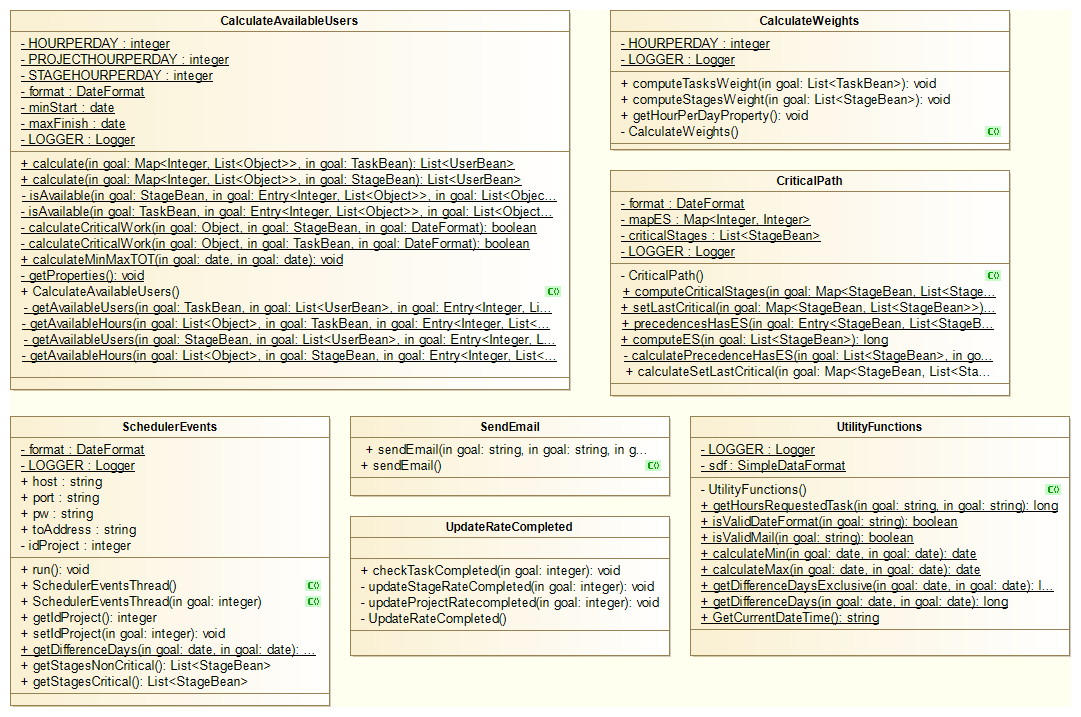
Package: utils



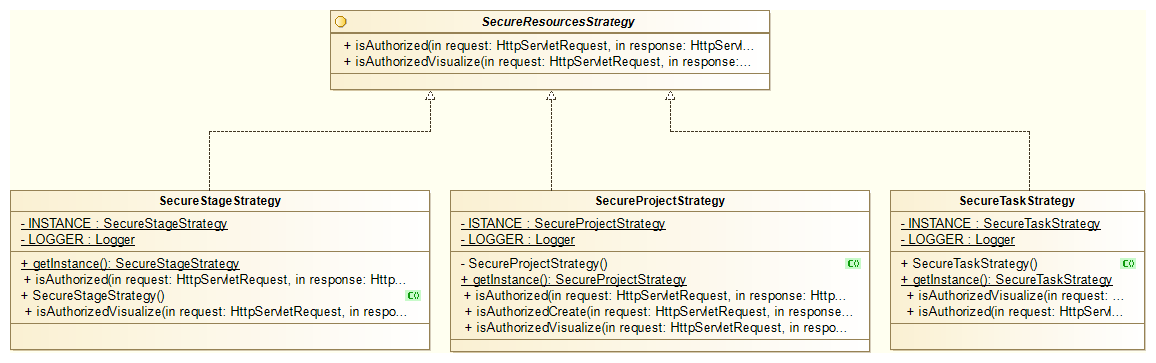
Package: context



Package: controllers.utils

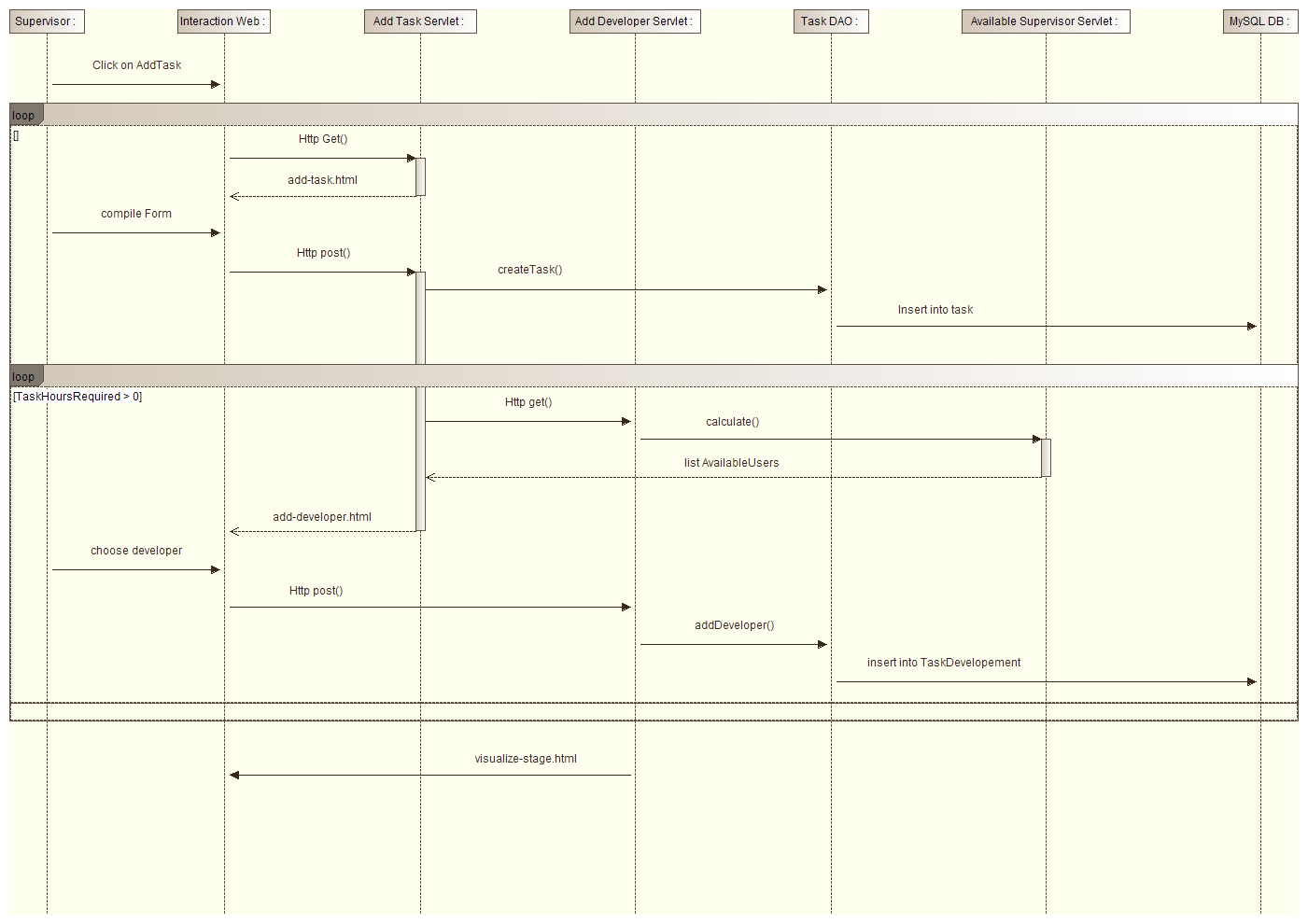


Package: controllers.utils.security



Sequence Diagram

This diagram is useful to better understand the creation of tasks(add task), included in the “Manage Stage” use case:



Design Pattern

During the development of the project it was useful to apply some design patterns.

**Singleton**:

Singleton was introduced in all DAO classes, to ensure the existence of unique instances and in all implementations of *SecureResourceStrategy* class.

**Strategy**:

The security system structure is based on Strategy pattern: depending on the action to secure(create Project/stage/task) etc..), the classes override the methods *isAuthorized* and *isAuthorizedVisualize.*

Computational Classes

In projectX some classes are purely computational. They support the management of core services as “Assign Developers to task” and the monitoring of delays. It follows the description of their logic:

*CalculateAvailableUsers*

This class calculates a list of users who have available hours and can be assigned to a new stage/task.