



Politecnico di Milano

Facoltà di Ingegneria dell'Informazione

Engineering Software 2 Project

MPH

Manage Project Homework
Part II: Design Document

2011/2012

Prof. Di Nitto

Andrea Gandini
Andrea Mambretti
Guendalina Milano

Index

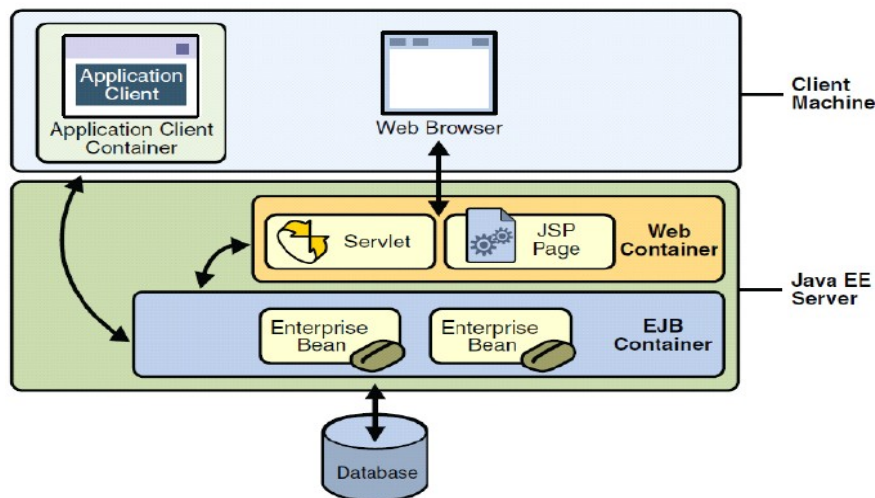
1Introduction.....	4
1.1RASD changes.....	4
2Data tier.....	5
2.1Conceptual design.....	5
2.2Logical design.....	6
3Business tier.....	7
3.1Boundary-Control-Entity Model.....	7
3.1.1Person.....	7
3.1.2Student.....	7
3.1.3Professor.....	9
3.1.4Administrator.....	10
3.2Entity beans.....	11
3.2.1Person.....	11
3.2.2Student.....	12
3.2.3Professor.....	12
3.2.4Group.....	13
3.2.5Project.....	14
3.2.6Deliverable.....	15
3.2.7File.....	15
3.3Session beans.....	15
3.3.1SessionGroup.....	15
3.3.2SessionProject.....	16
3.3.3SessionPerson.....	16
3.3.4SessionUpDown.....	16
3.4 Functionalities diagrams and their interaction.....	18
3.4.1Login and registration.....	18
3.4.2Creation of a new group.....	21
3.4.3Upload of a file.....	22
3.4.4Creation of a new project.....	23
3.4.5Enabling a professor.....	23
3.4.6Enabling a group to display other group profile.....	24
3.4.7Marking of a project.....	24
3.5Sequence diagrams.....	25
3.5.1Login and registration.....	25
3.5.2Creation of a new project.....	26
3.5.3Accepting the subscription to a group.....	26
4Web tier.....	27
4.1UX model.....	27
4.1.1Professor.....	27
4.1.2Student.....	27
4.1.3Administrator.....	28
4.2Deployment.....	29

1 Introduction

To realize MPH project was used JEE 5 (Java Enterprise Edition), that permits to have a clear distinction to the front-end and the back-end. It used the MVC (Model View Controller) model to define this structure.

JEE 5 permits to separate the project in two macro areas: the client machine and the Java EE Server. In the client, because we choose to create a web project, we have only web browser that manage the page send by the server.

Java EE Server manage the request from the client: it permits to distinct web container and EJB container. The first one manage the request and represent the logic business of the application. It used also to creates dinamic web page. The second one permits to have an access to the database and create query to retrieve data storage in the database.



The design choices are the sequent:

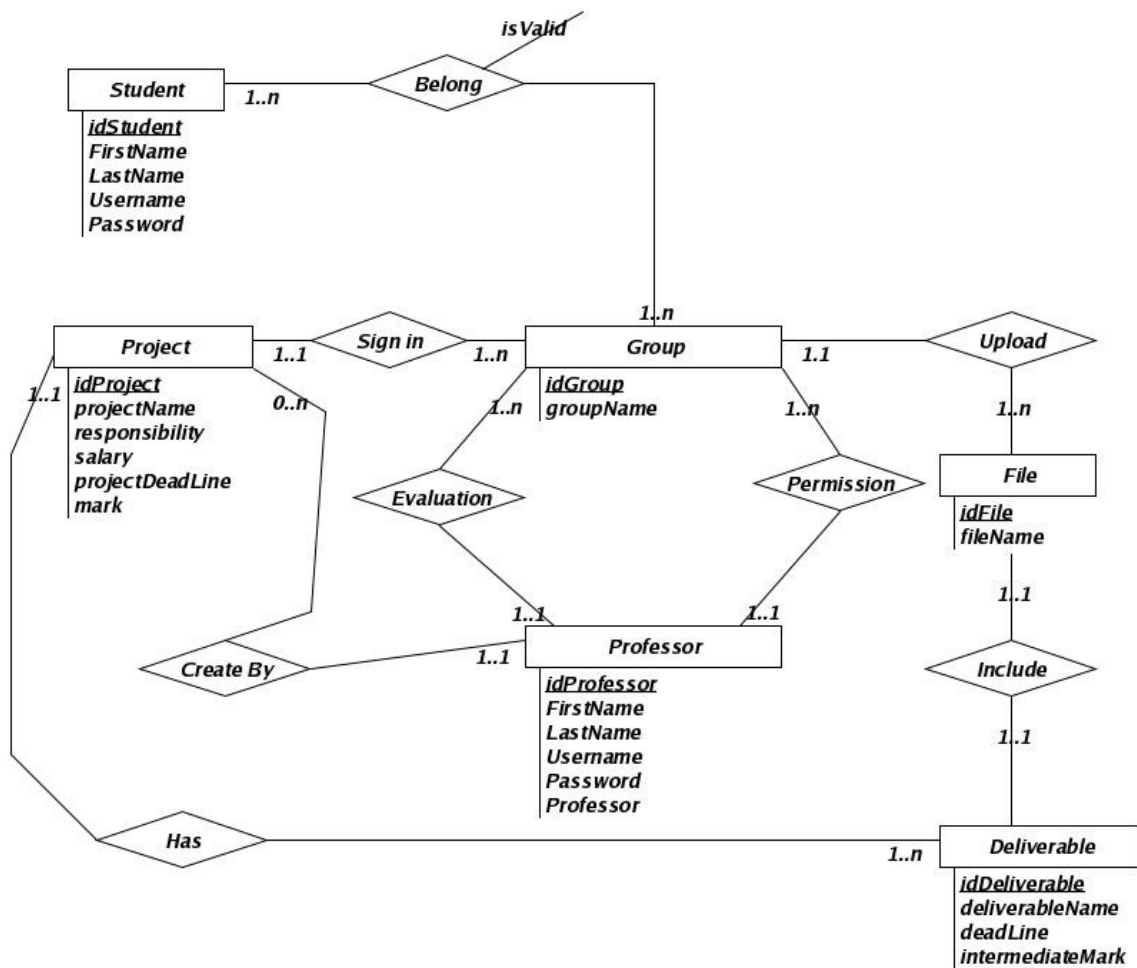
- Web Application
The system are web based, so it isn't necessary for the user to install the application on his machine. He must have only a web browser. This choice pretends updates are provide by Servlet and AJAX technology.
- Application Server
The application server used is JBoss, open source and constantly updates system. The version used for the project is the 5.0.
- DBMS
the DBMS used is MySQL, a widespread and open source database.

1.1 RASD changes

From the previous RASD made, we choose to implement the possibility for a student to accept or deny the invitation to a group from his personal page. It permits to have more control to the group who are subscribed. We also decide to permit to subscribe like an administrator and, in this case, your registration must confirmed from an administrator with the correct permissions.

2 Data tier

2.1 Conceptual design



2.2 Logical design

Entity:

- 1 Student: This entity contains all information about a generic student like the idstudent, firstname, lastname and also credits for the system as username and password.
- 2 Professor: It contains information about a professor such as idprofessor, firstname, lastname and also username and password for enter into the system.
- 3 Group: It is the entity for store information about groups and it contains the id of group as under idgroup, the name.
- 4 File: It contains the id of the file uploaded and the name of that.
- 5 Deliverable: It contains information about the deadline chosen by professor. The idnumber, the deliverableName and the date of the deadline.
- 6 Project: the information stored here are idproject, nameProject, responsibility, salary, mark.

Relation:

- 1 Belong: Association between student and group. It represents that a student belong to one or more groups and also into the groups there are one or more students.
- 2 Sign in: Association between group and project. It means that a group can join to a project and that into a project can be one or more groups.
- 3 Create By: It represents that a professor creates groups (from zero to n) and that a project can be proposed by one professor.
- 4 Has: This relation between project and deliverable means that into a project we have one or more deliverable and that one deliverable belongs just one project.
- 5 Evaluation: It means that all groups are evaluated by a professor and one professor get evaluation to groups of him project.
- 6 Permission: This relation between professor and group says that the professor give to group permission to see files of other group and that one group receives this permission from its professor for that project.
- 7 Upload: This relation say that a group can upload some file for their deliverable and that one file can be uploaded by only one group
- 8 Include: This Association show us that one file is for one deliverable and that one deliverable has one file to be uploaded

LOGIC SCHEMA

Student (idStudent, firstName, lastName, username, password)

Belong(idStudent, idGroup, isValid)

Administrator(idAdministrator, firstName, lastName, username, password)

Group (idGroup, groupName, idProfessor, idProject)

Project(idProject, projectName, responsibility, salary, projectDeadLine, mark, idProfessor)

Professor(idProfessor, FirstName, LastName, Username, Password, Professor)

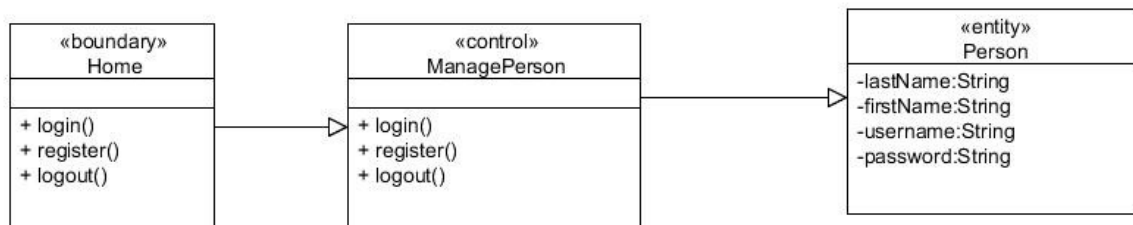
Deliverable(idDeliverable, deliverableName, deadLine, idProject)

File(idFile, fileName, idDeliverable)

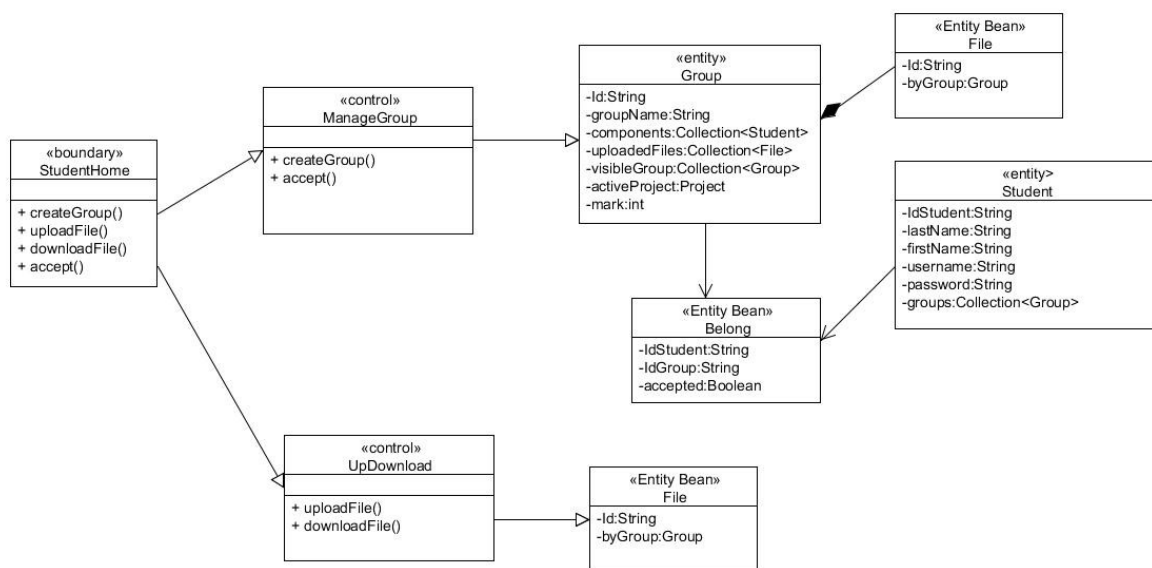
3 Business tier

3.1 Boundary-Control-Entity Model

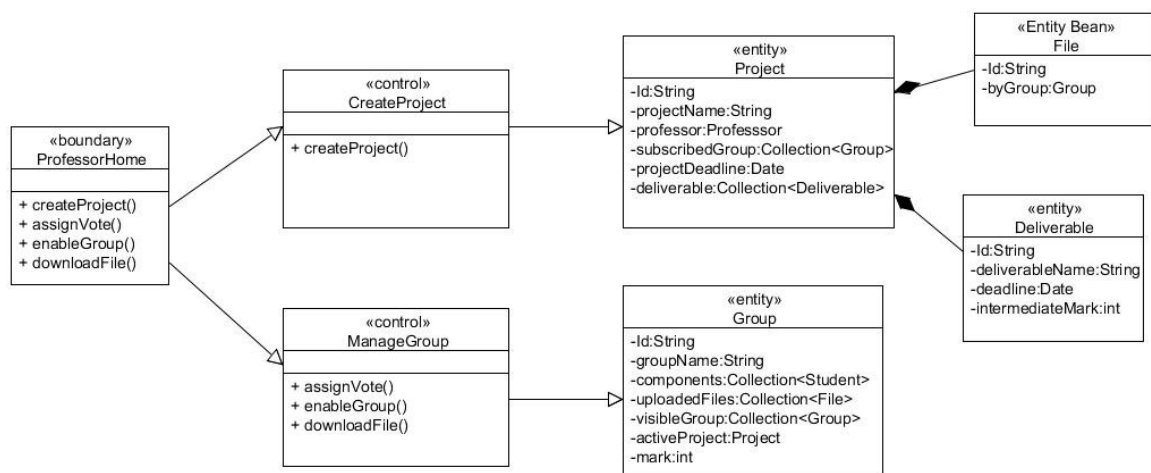
3.1.1 Person



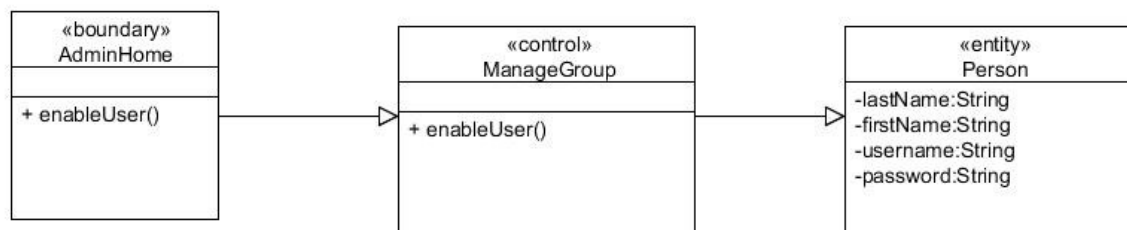
3.1.2 Student



3.1.3 Professor

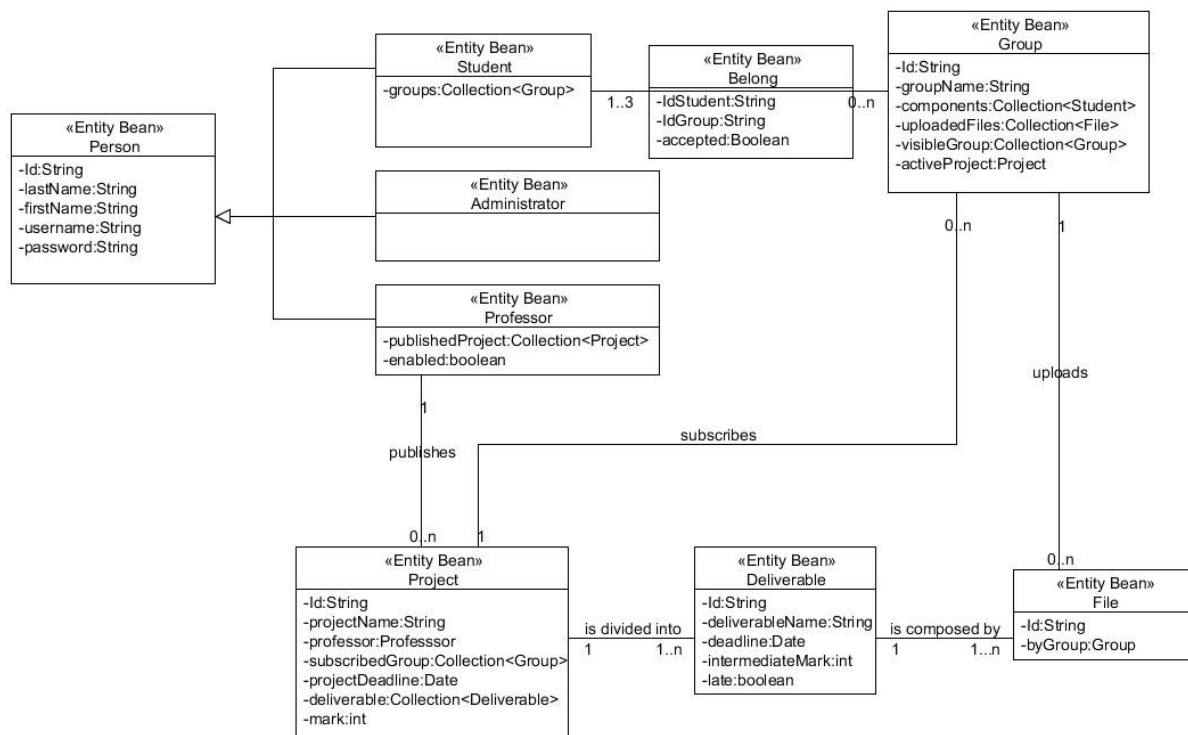


3.1.4 Administrator



3.2 Entity beans

In this chapter they are shown the entity beans we are intended to use in the application. Here it is a global view.

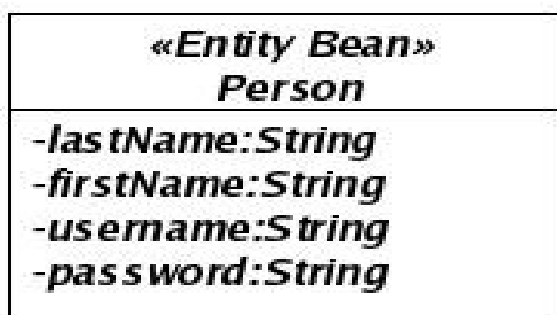


Subsequently for each entity beans it will be explain the meaning of each field and the relationship among the other entities.

In every entity an **id** field is present, it permits to refer to a specific object univocally.

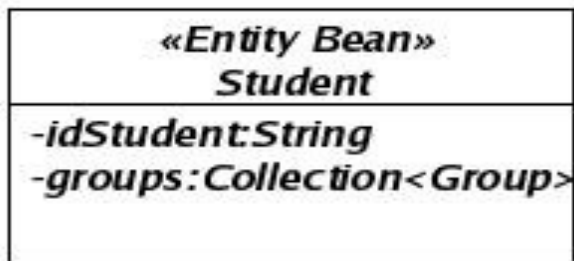
From now on all the fields with an obvious meaning won't be explain more in detail.

3.2.1 Person



This entity beans is used to manage the creation of a new account and to provide for all the information about a generic user once he/she had registered. **firstName** represents the name of person, **lastName** represents the surname, **username** is the nick name of the person and the **password** is the secret key for accessing to the system.

3.2.2 Student

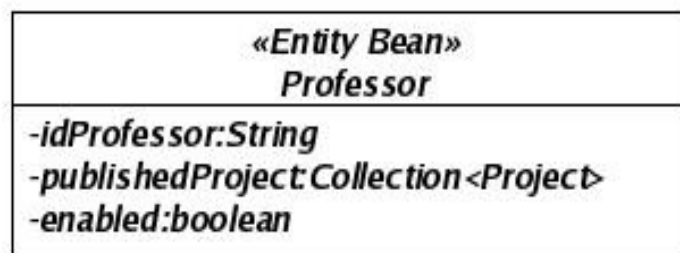


This entity get all the attribute from person and add the student identifier (idStudent) and keep the list of groups where the student is signed in (groups) .

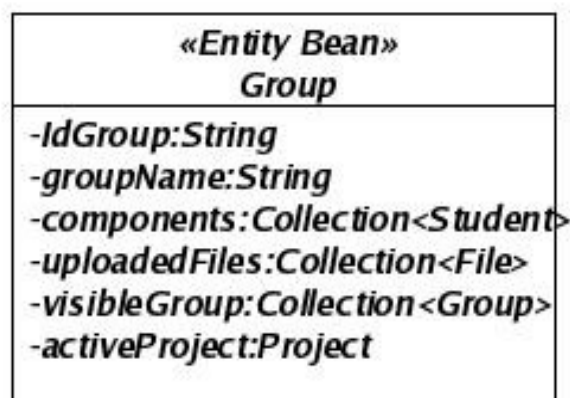
3.2.3 Professor

The field **publishedProject** represents the collection of all projects the professor published .

Enable states whether an administrator had already enabled the professor or not. Also there are the fields of person and the professor identifier (idProfessor).



3.2.4 Group



The group is composed by an idGroup number, the name of group (groupName), the collection with student that are into the group, the collection of files uploaded from members of the group, the collection of groups that can be see from this group and the id of activate project in active project.

3.2.5 Project

«Entity Bean» Project
-IdProject:String -projectName:String -professor:Professor -subscribedGroup:Collection<Group> -projectDeadline:Date -deliverable:Collection<Deliverable> -mark:int

3.2.6 Deliverable

«Entity Bean» Deliverable
-IdDeliverable:String -deliverableName:String -deadline:Date -intermediateMark:int -late:boolean

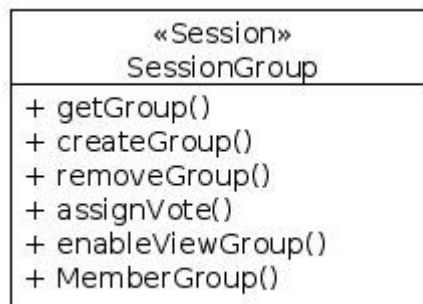
3.2.7 File

«Entity Bean» File
-Id:String -byGroup:Group -uploadDate:Date

ByGroup field represents the group who uploaded the file.

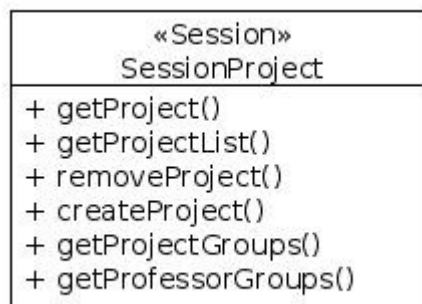
3.3 Session beans

3.3.1 SessionGroup



This session contains all methods helpful to manage groups. It permits to create groups, share information between group (option valid only for a professor), retrieve the list of all groups and remove groups. It is also useful to assign vote to a group and contains the method invoked by a professor to access at the ViewGroup.

3.3.2 SessionProject



This session manage the request referring at projects. It permits to a professor to create projects, remove it; group to subscribes to a specified project. Is useful also to retrieve the list of the project and the specific description of a specific project.

It manage also the deliverables and the files referring of a specific project. This permit to manage all control of file referring to a project in only one class. Groups can visualize the deliverables of project in their page and upload file directly from there.

3.3.3 SessionPerson

«Session Stateful» SessionPerson
+ logstate()
+ register() + login() + logout() + enableProf() + getPerson()

In this session we have specified all method used to manage a user. It permits to create a user (student, professor or administrator), login or retrieve a specific user from database. It is necessary to complete login/logout operation.

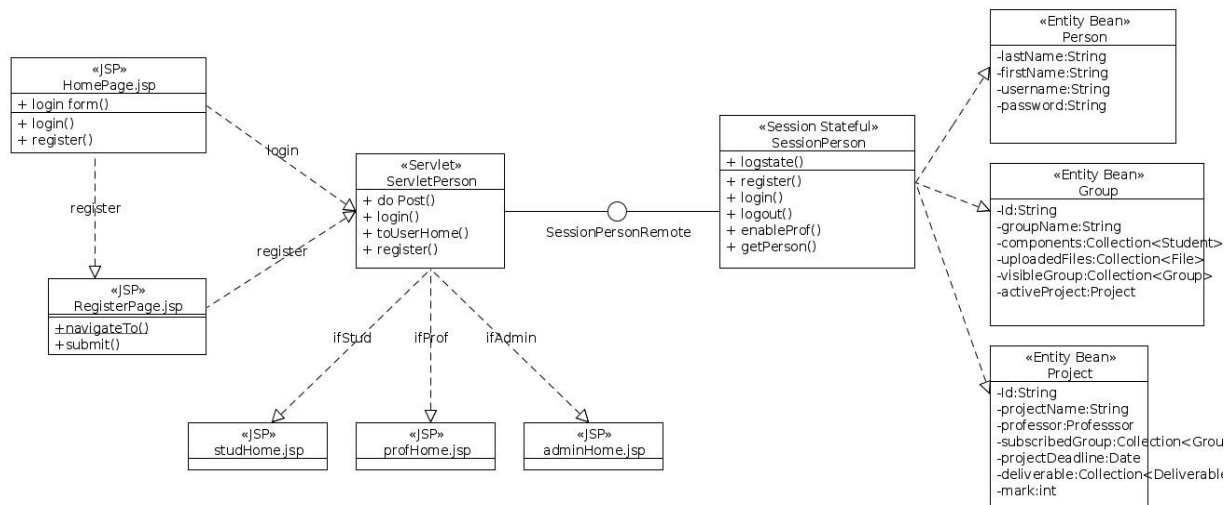
3.3.4 SessionUpDown

«Session» SessionUpDown
+ downloadFile() + uploadFile() + removeUploadFile() + downloadGroupAll() + downloadTypeAll()

This session has been identified cause method inside it are critical for the whole system. It permits to manage upload and download of all file committed by groups. Inside this class are enable the method offer from the platform Jboss that permit to create a .rar/.zip of collection of file uploaded. Professor can chooses to download all file uploaded by Group12 and the request is manage by this session.

3.4 Functionalities diagrams and their interaction

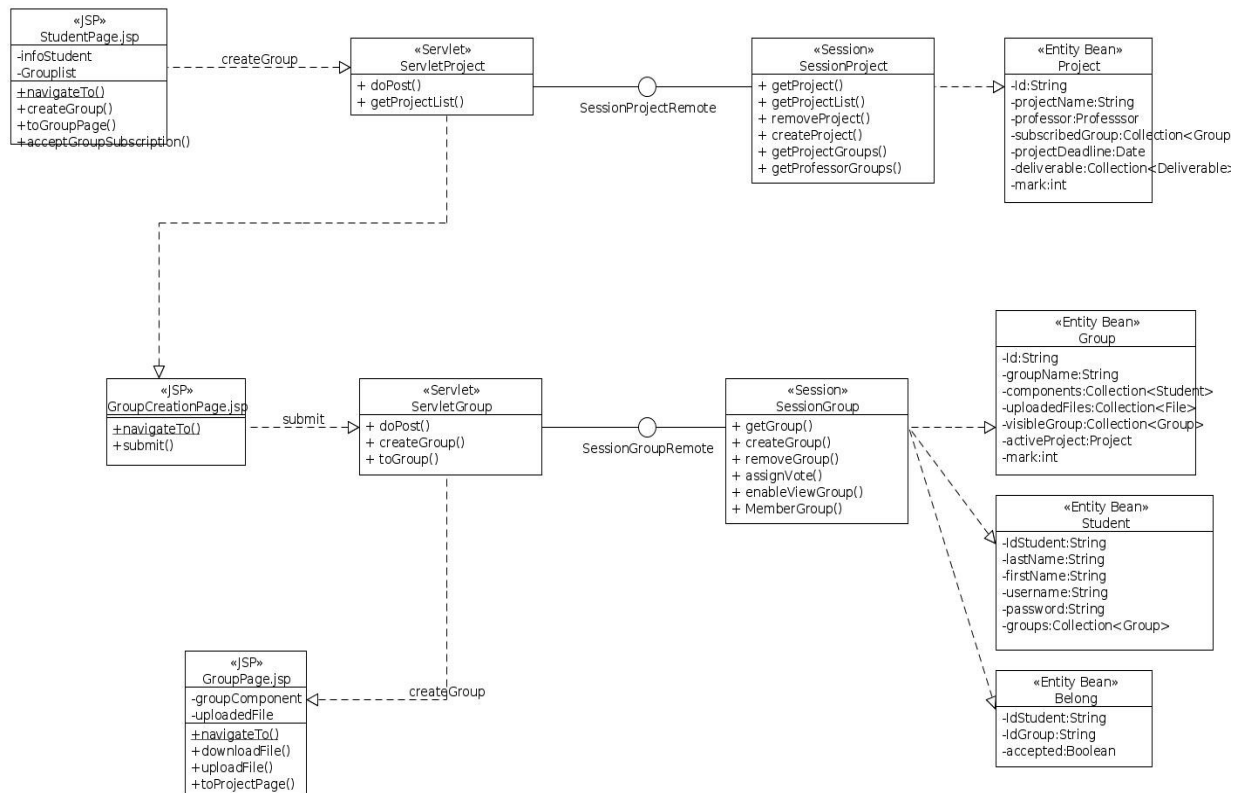
3.4.1 Login and registration



User, when access the web site, are redirect by default to HomePage. Here user can be choose to login or register itself to the site. If chooses the first option, he fills the form and submits the request to a servlet which deals with the person method. Here theres a control with database and information inserted by user: if correct, user are redirected to the UserPage correctly by his registration (student are redirected to StudentPage, etc.); if ecnounter some error, user are redirected to HomePage.

Second option are register: user are redirected to a register page where can insert all information (like username, password) and send request. Same servlet used before occupies session to register user on database and, if all is correctly, user are redirected to UserPage.

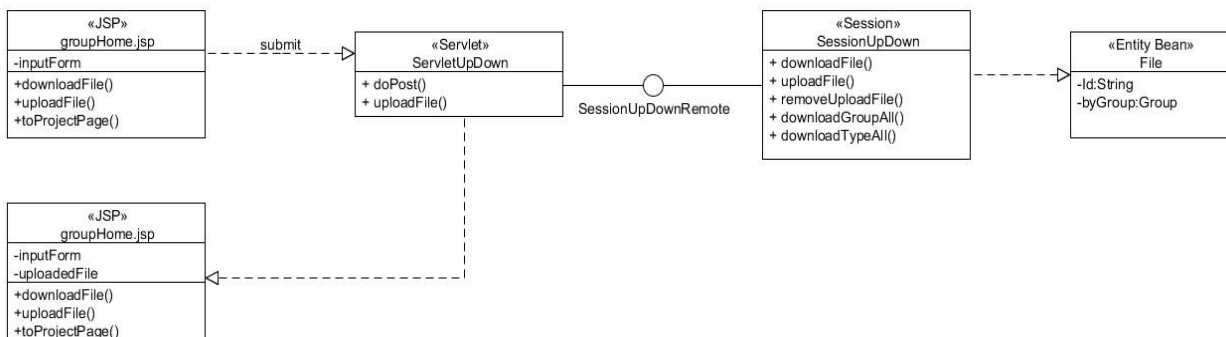
3.4.2 Creation of a new group



From StudentPage a student sends a request to create a group. Servlet takes the request and retrieves information about projects available. User can now fill the form: user who wants to invite, select the group name and chooses the project. Student can now submit his request. Another servlet takes the information input by student and persists it into the various entity. After the operation, student is redirected to group page.

Note ServletGroup communicates with a session that retrieve/persist the entities group, student and belong. This is necessary to have a list of the student invited to participate at the group just created: these ones can accept or deny the invitation.

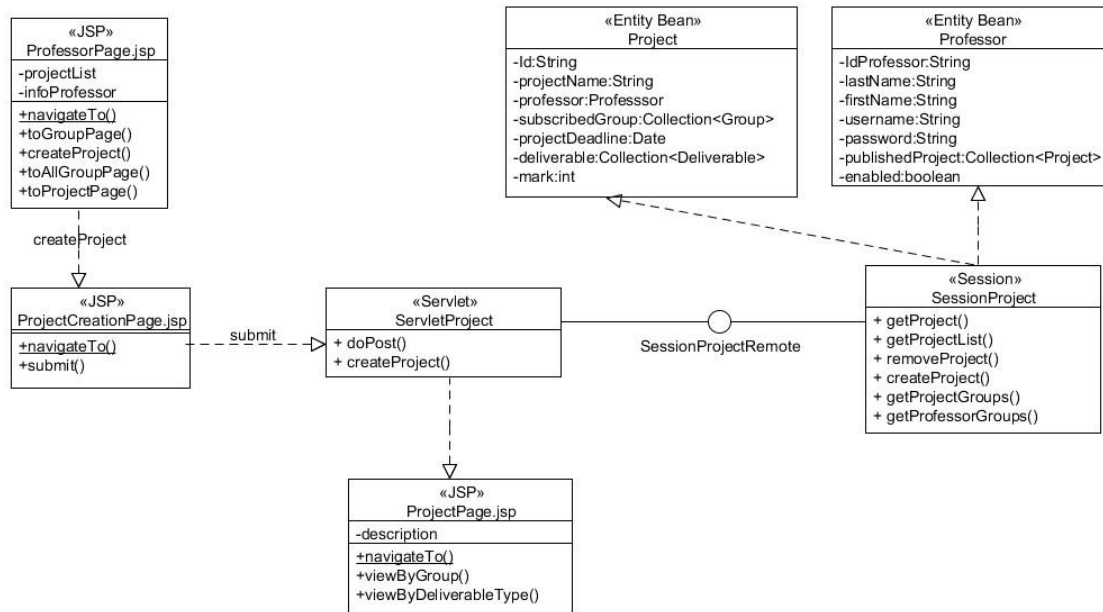
3.4.3 Upload of a file



Each component of the group can upload one or more file in his/her group homepage through an input form. ServletUpDown receives the request and SessionUpDown saves the uploaded file.

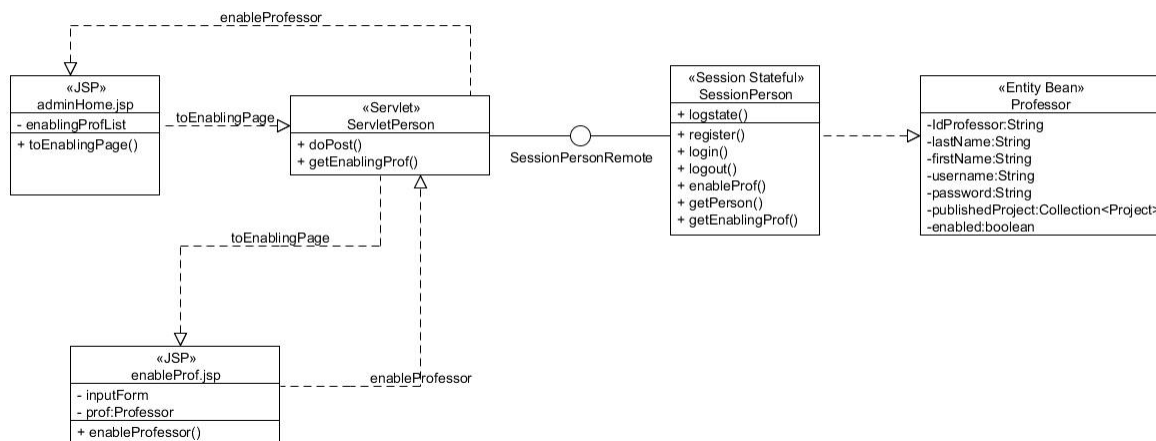
After that the student is redirected to the group homepage.

3.4.4 Creation of a new project



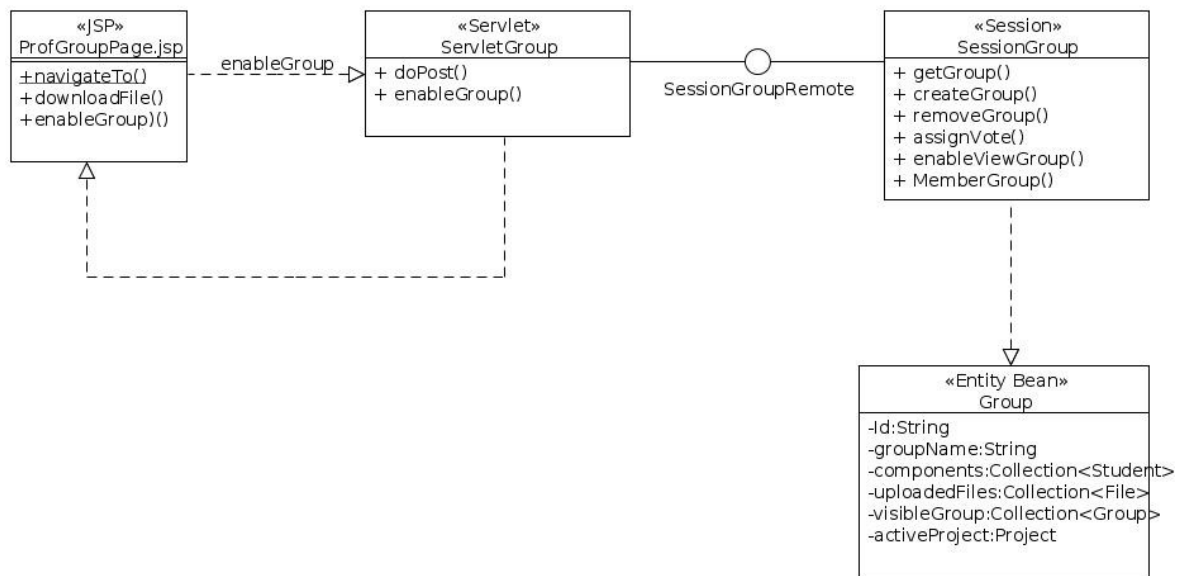
Professor goes on ProjectCreationPage by a simple link, cause the destination page don't need any information from database. Here he inputs all information (name, deliverable, deadline) and sends, by a submit, the request to the servlet that manages the method about projects. Here a session create the project, map it to the professor who sends request and this one is redirected to the project page, where the information are displayed.

3.4.5 Enabling a professor



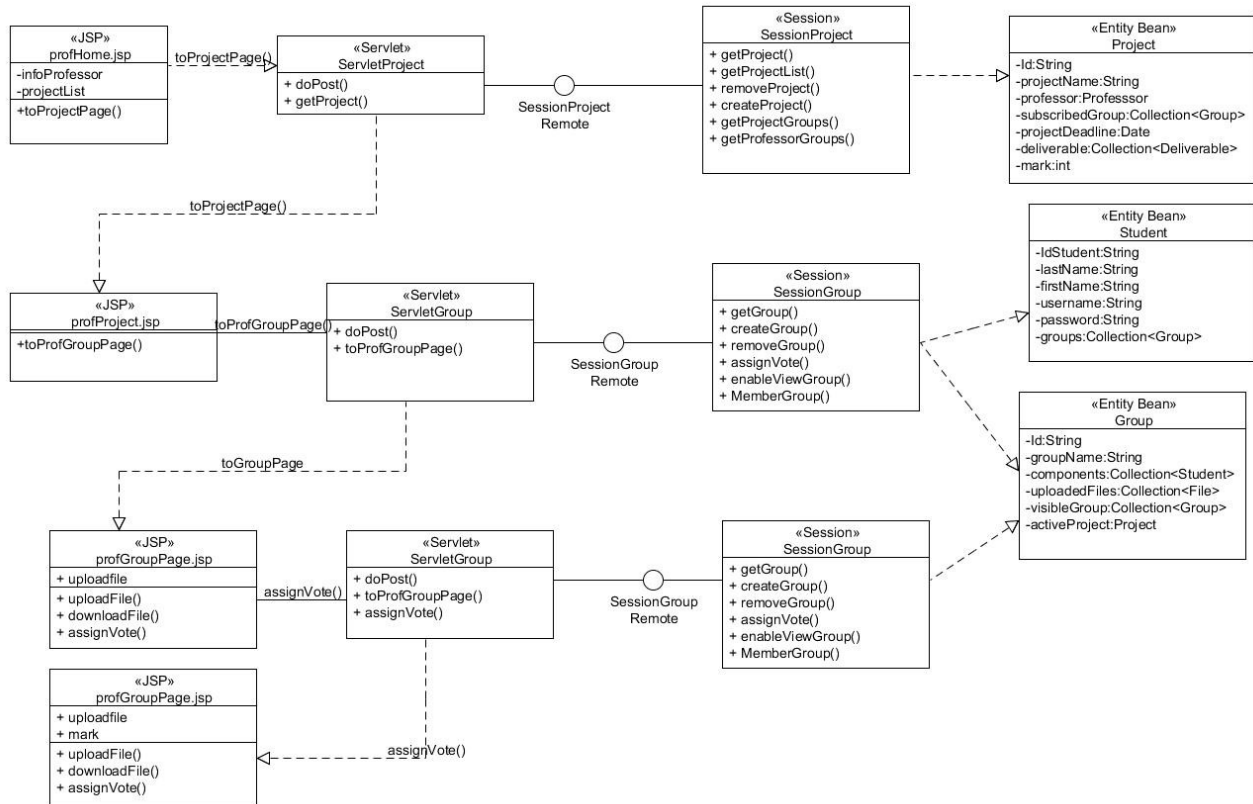
Only administrator can enable a request for registration made by a professor. From adminHome administrator selects one of the professor he hasn't enabled yet. ServletPerson manages the request and SessionPerson retrieves the selected professor, enableProf.jsp is displayed. Administrator fills an input form and the new state of professor is updated by SessionPerson and is redirected to his homepage.

3.4.6 Enabling a group to display other group profile



Where a professor view a ProfGroupPage (the group page view for a professor), he can select from a list the groups that share information with the selected group. So, a professor select the group(s) and send the request to the business tier, who registers who group share info. Professor remains on the ProfGroupPage.

3.4.7 Marking of a project



From his homepage the professor selects a specific project, the ServletProject and the corresponding session bean are invoked and provides the retrieval of the selected project and forwards to profProject.jsp.

From this new page the professor selects the group he wants to evaluate, ServletGroup and SessionGroup manages the request and retrieves all data about the corresponding group and redirect to profGroupPage.jsp.

From this page the professor assign a vote to the group through a simple input form.

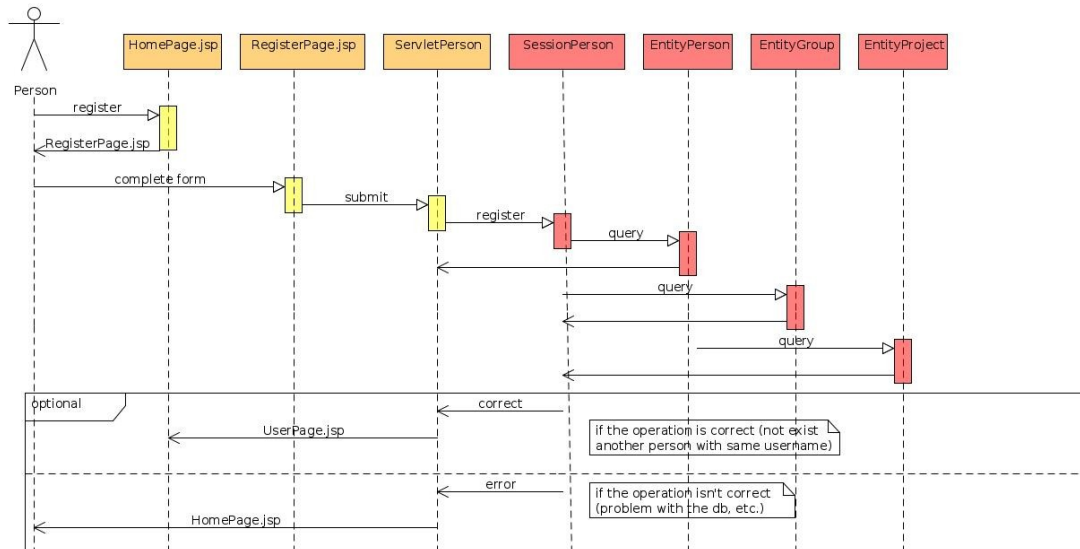
ServletGroup and Session group manages the request and save the new situation.

The navigation flow is redirected to profGroupPage.jsp

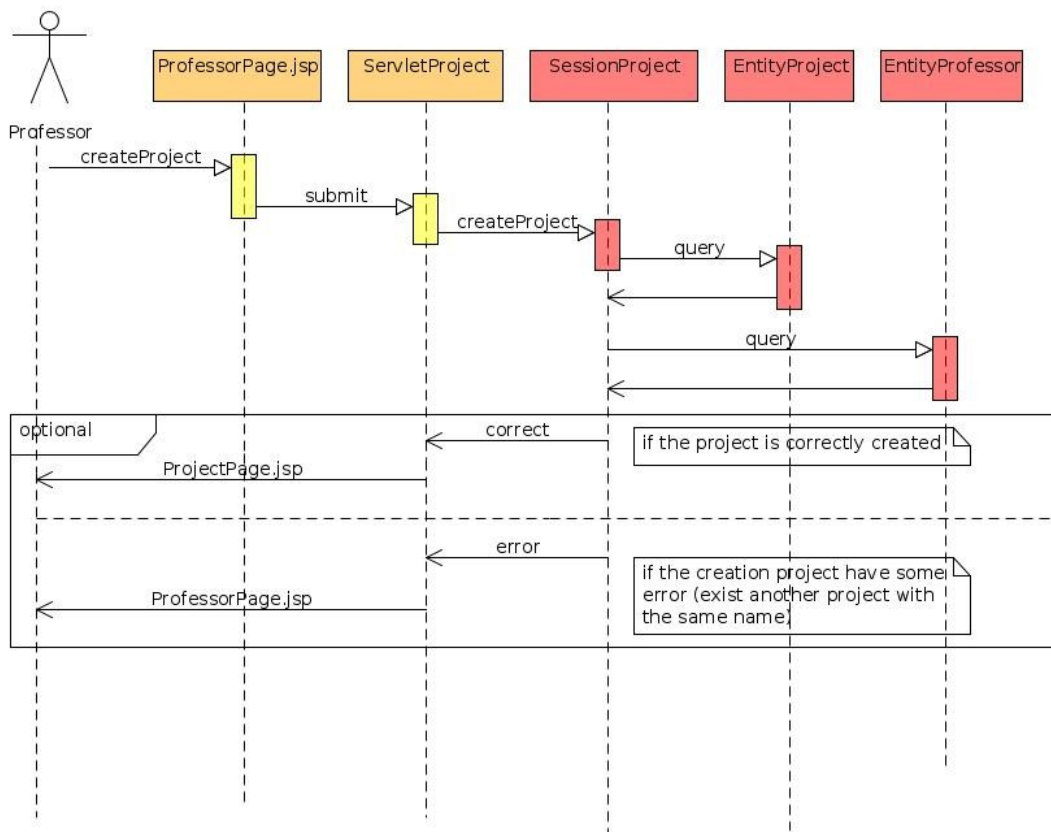
3.5 Sequence diagrams

Following are presented some sequence diagram to explain relationship between servlet, session and entity.

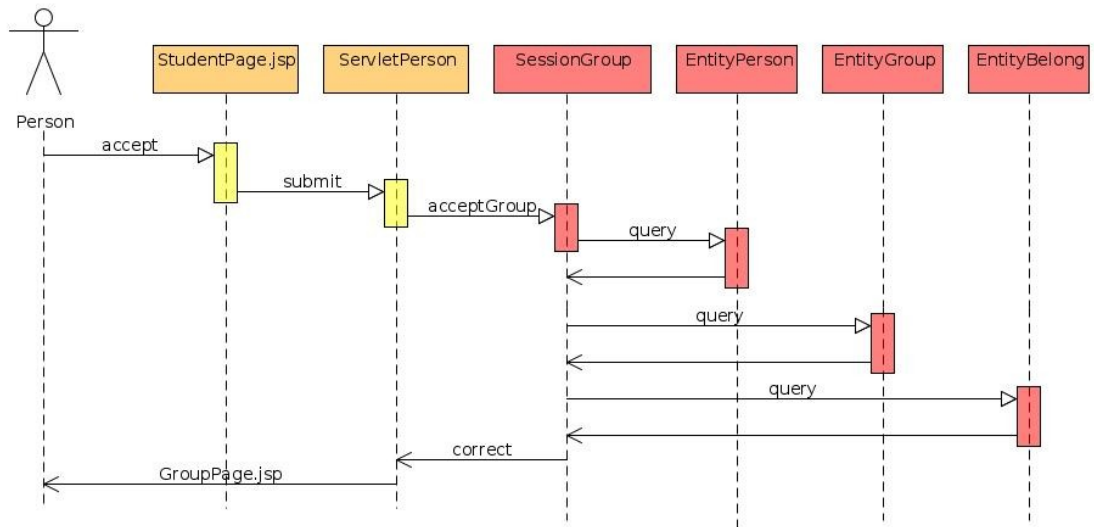
3.5.1 Login and registration



3.5.2 Creation of a new project



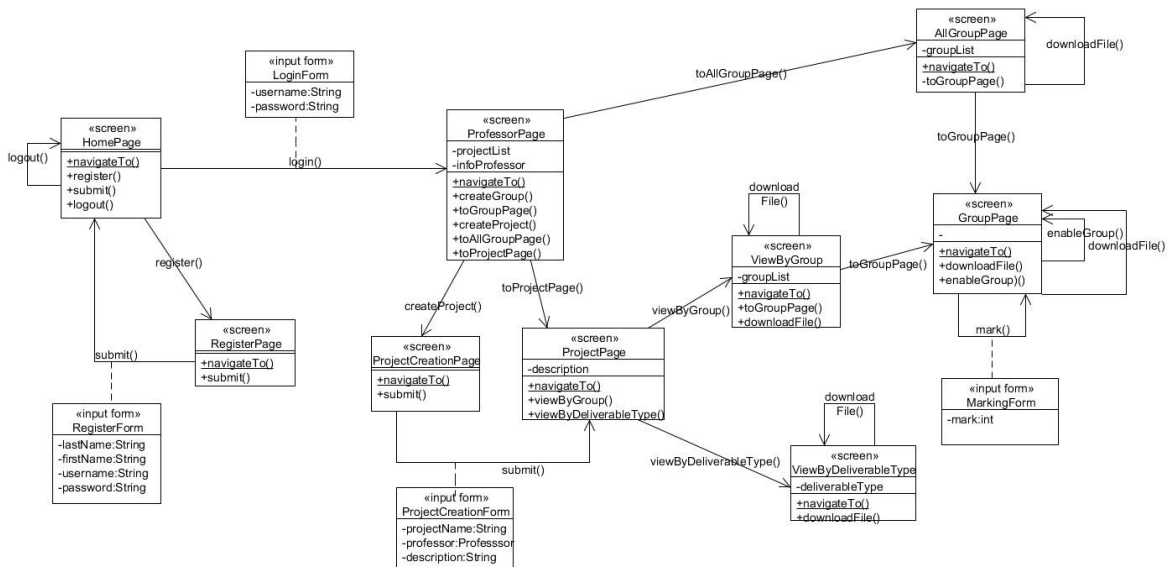
3.5.3 Accepting the subscription to a group



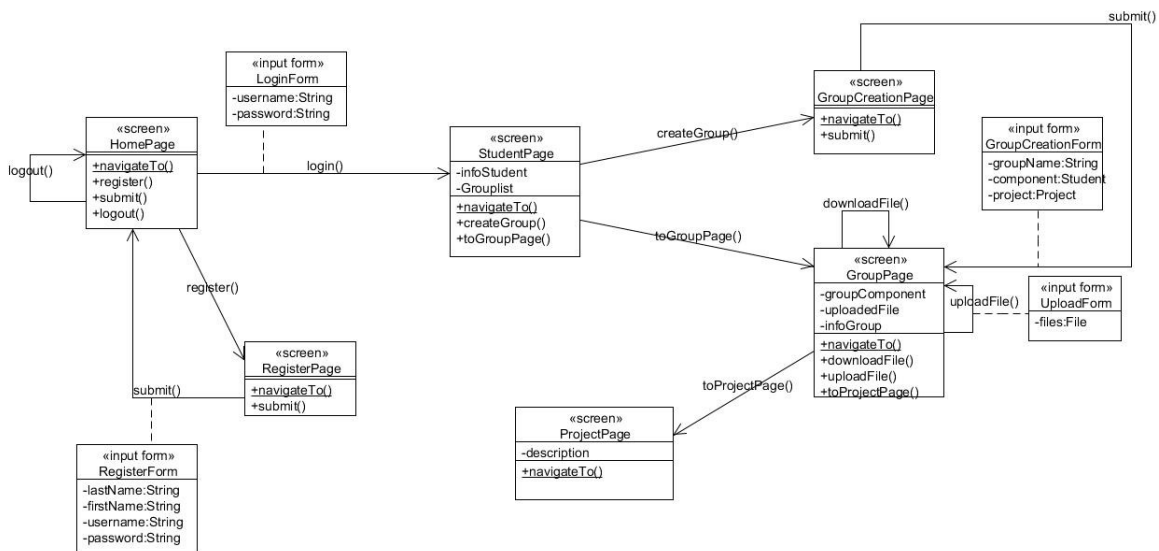
4 Web tier

4.1 UX model

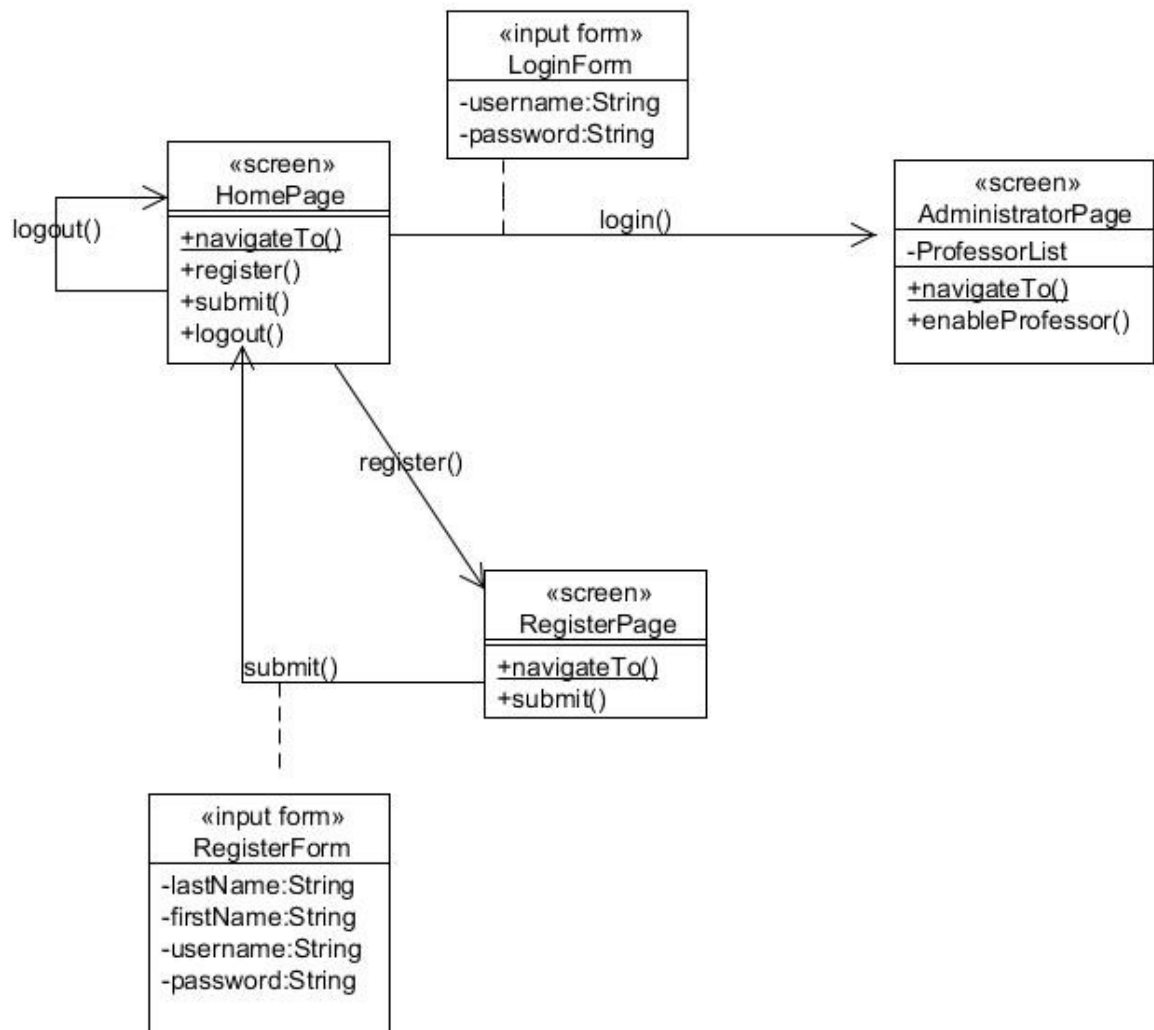
4.1.1 Professor



4.1.2 Student



4.1.3 Administrator



4.2 Deployment

The archive of the application is represented below:

