Managing students' projects using cloud computing

System analysis and design in high level

Purpose: To understand and determine the scope of the system and its components and architecture.

Supervisors: Dr. Mouhib Alnoukari - Eng. Anas Abdulaziz

Students: Raghad al hossny – Kassem al Kelani

project goals and services:

make a system which will automate and organized all aspects of students' projects completing process in Syrian private university (Junior Project, Senior Project 1, Senior Project 2), which is:

- Organizing the projects suggestions process that happened in the beginning of each semester.
- Full Automat the project request and registration process.
- Managing and facilitation the projects evaluation process.

The main actors with their roles and responsibilities:

1. student:
    - View the available projects list.
    - Request and register a project “in the begging of the semester”.
    - View and track the projects evaluation plan, and execute its rules.
    - The ability to upload files.

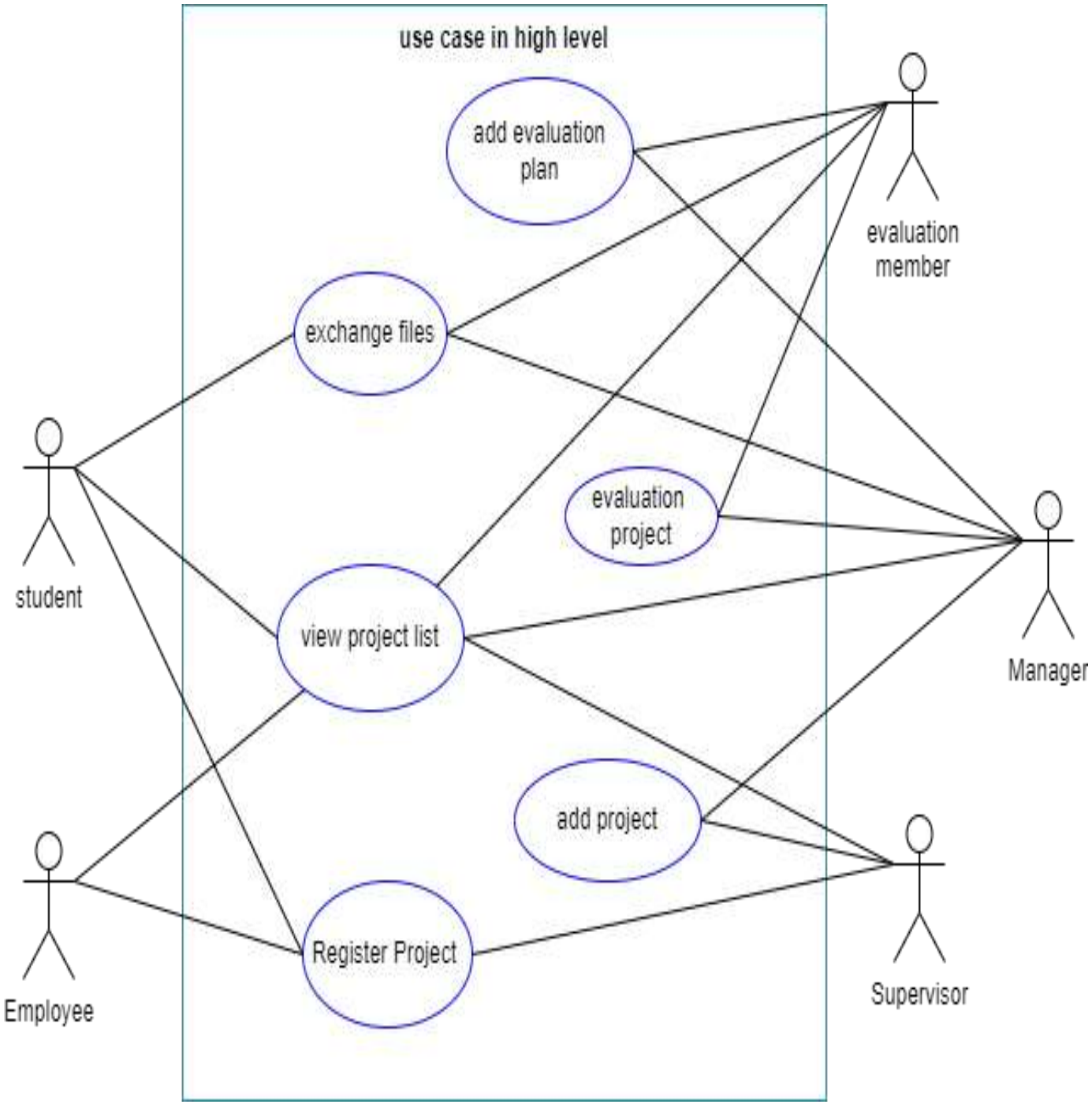
2. Supervisor:
    - Add a project suggestion “in the begging of the semester”.

3. Manager:
    - Approve or rejects projects suggestions.
    - Approve and edit evaluation plan.
    - Track the progress of any team.
    - View simple reports.

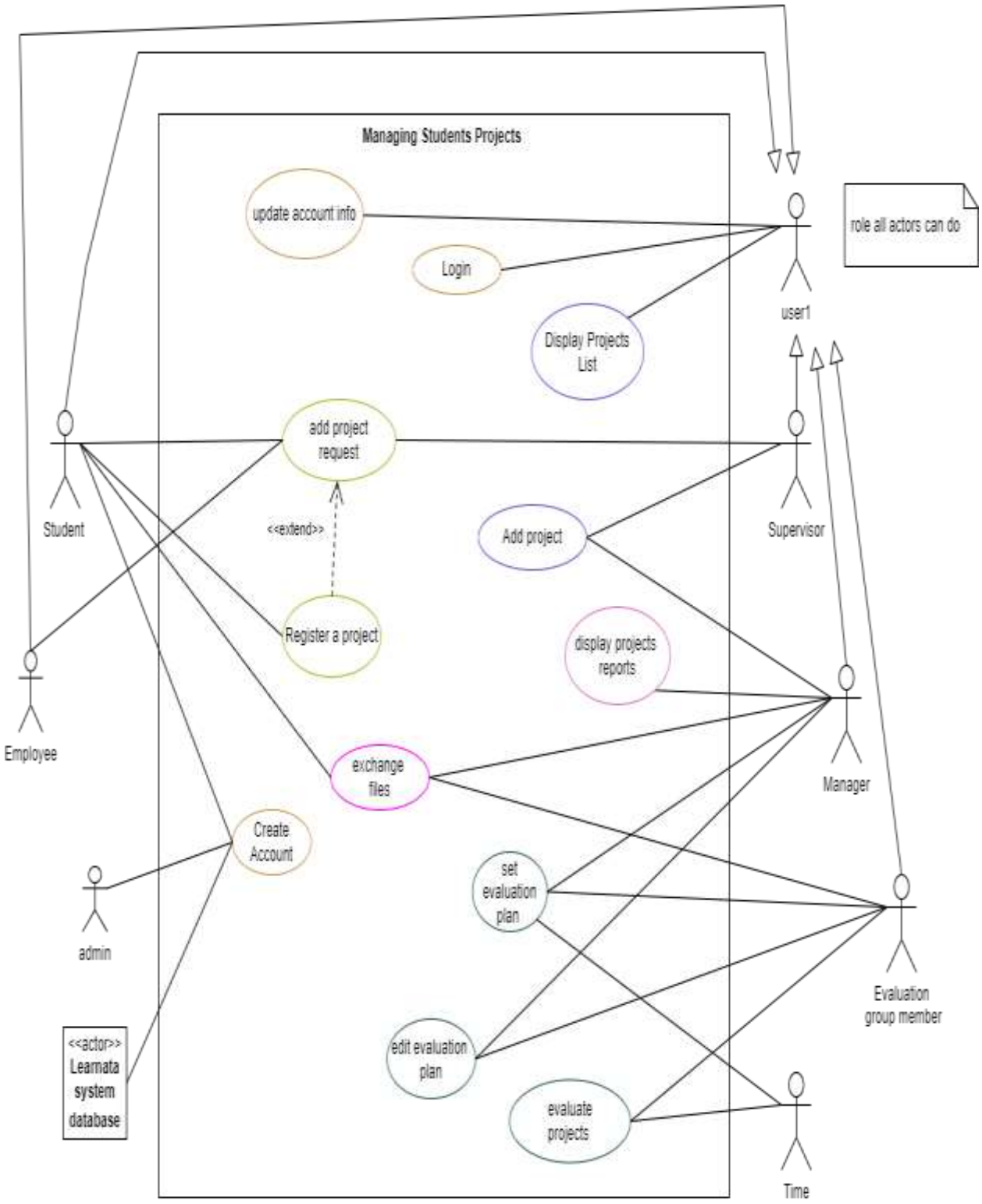
4. evaluation team member:
    - make evaluation plan for the projects.
    - Evaluating teams.
5. responsible employee:
    - receive registration requests from students.

6. The database of the university system “Learnata” (secondary actor):
    - to compare the ids (when user create his only account by his university id).
    - Check if the student information match with the project registration conditions (like: total number of his completed hours is more than or equal to 100).

Use case diagram in high level:



Detailed use case diagram:



The specification of top three important use cases:

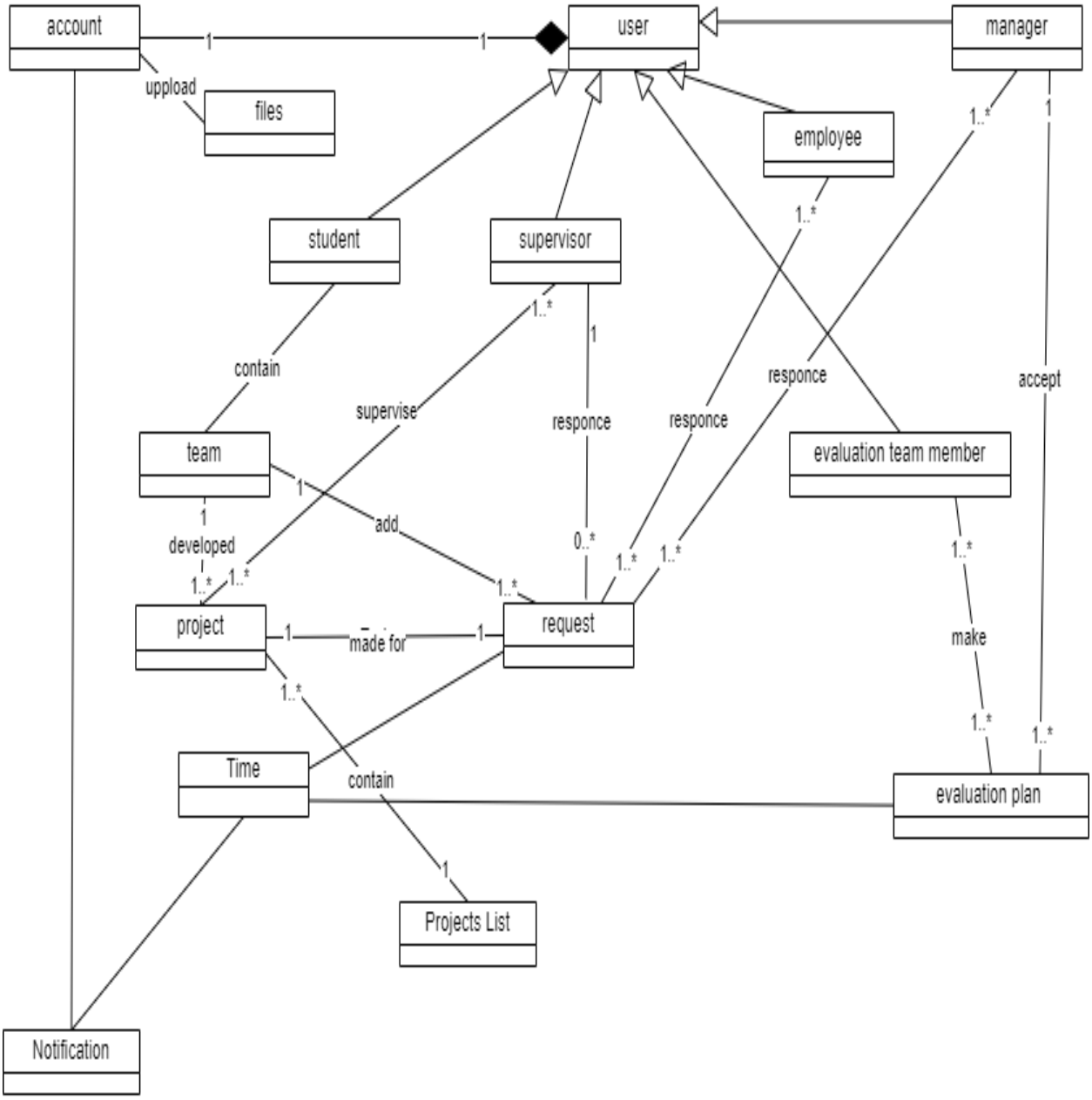
Use case name	add project.
Participating Actors	initiated by supervisor communicated with manager.
Flow of events	1. First supervisor will ask to “make project suggestion”. 2. The system will ask the supervisor to enter project title. 3. Supervisor will enter the title. 4. System will compare titles in the database and return to the supervisor acceptance if there is no such project in the history, and ask him to complete the other informations. 5. Supervisor will enter the other informations required to make a new project (goal, description, plan and key steps). 6. The system will accept to make the project after complete the project information, and send the suggestion to the manager. 7. the manager makes an approve for the suggestion project. 8. System will send the approval notification to supervisor, and add the project to the project list to be display to every on using the system.
Entry condition	the service is available (specific time determined by manager)
Exit conditions	project will be shown in the project list.

Use case name	make project request.
Participating Actors	initiated by student communicated with supervisor, employee.
Flow of events	1. After choosing an available project from project list, the student will take the option make request. 2. System will ask the student to add a team member. 3. Student will enter names and ids for his partners. 4. System will check the database for the ids and send notifications for partners asked them to make a make request acceptance on their accounts. 5. Other students(partners) will enter their accounts and approve to make that request. 6. After this system will send the request to the supervisor of the project. 7. If the supervisor accepted it 8. system will send the approvement to the student and, registration request with project information and student to the responsible employee. 9. If the employee registers the project (he will check some conditions related to university rules outside our system responsibilities) he will choice accepted and done. 10. The system will send the acceptance notification to the students, and the service will end.
Entry condition	the registration service is available (specific time), and

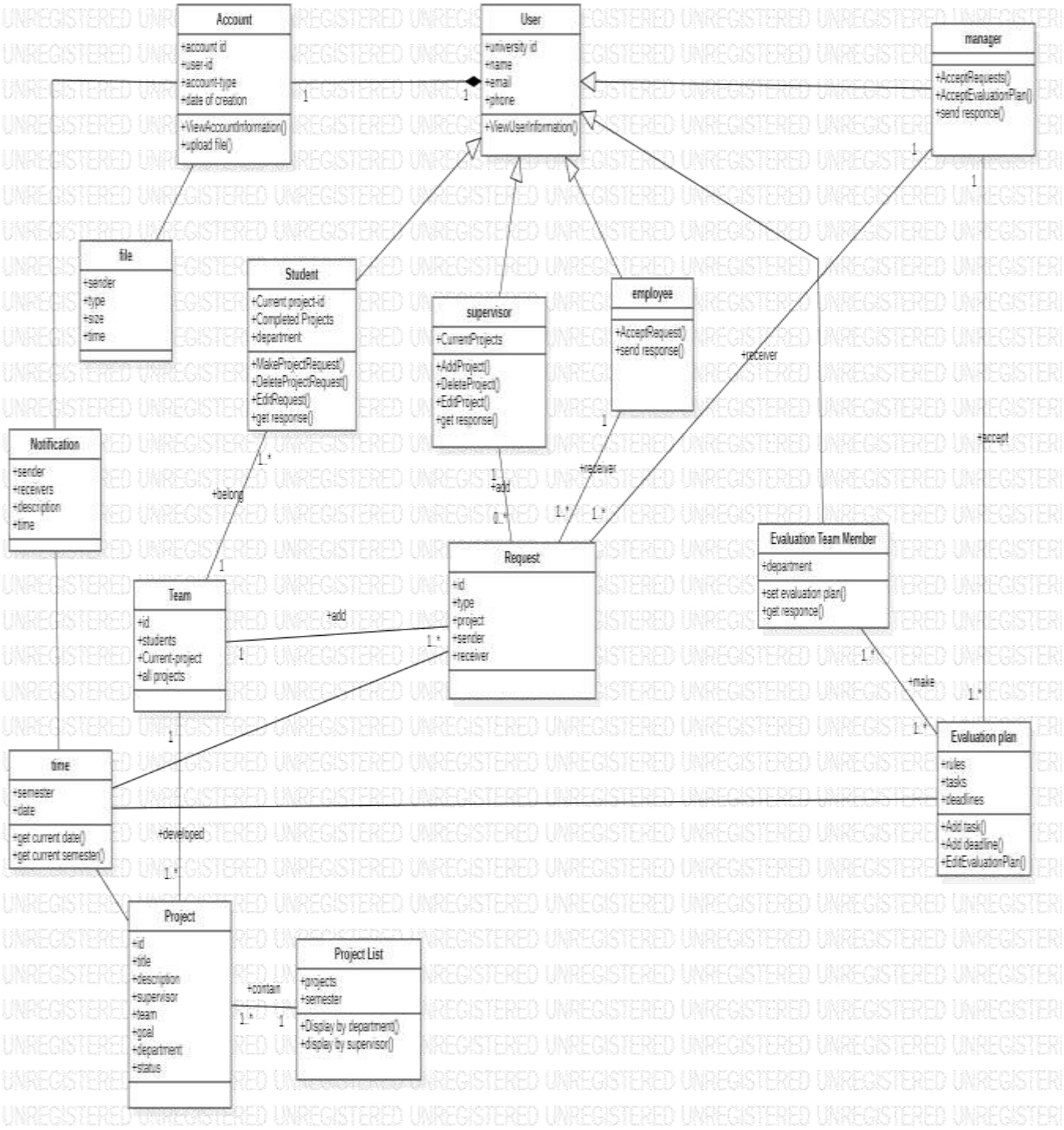
	all supervisors already added their projects (project list is available).
Exit conditions	student register their projects and have a communication page connects them with their team members.

Use case name	create student account.
Participating actors	initiated by student.
Flow of events	1. Students will enter the system(website) first and ask to make account. 2. System will ask for their university id and some informations name, number, email (all information must be match with his information in the university system database). 3. Student will enter the information. 4. The system will access the university database to check the id and other informations. 5. If the student is existed, the system will then check some conditions related to projects (number of completed hours more than or equal to 100, and some courses has to be complete), if all conditions are met the system will ask the user to enter a password for the account. 6. Student will enter a password. 7. System will check if that password met a determined security limit if that true system will make an account for the student and take him to it.
Entry condition	the account creator is student the student exist in the university and met some projects Conditions
Exit conditions	student has account and can start to use the system services.

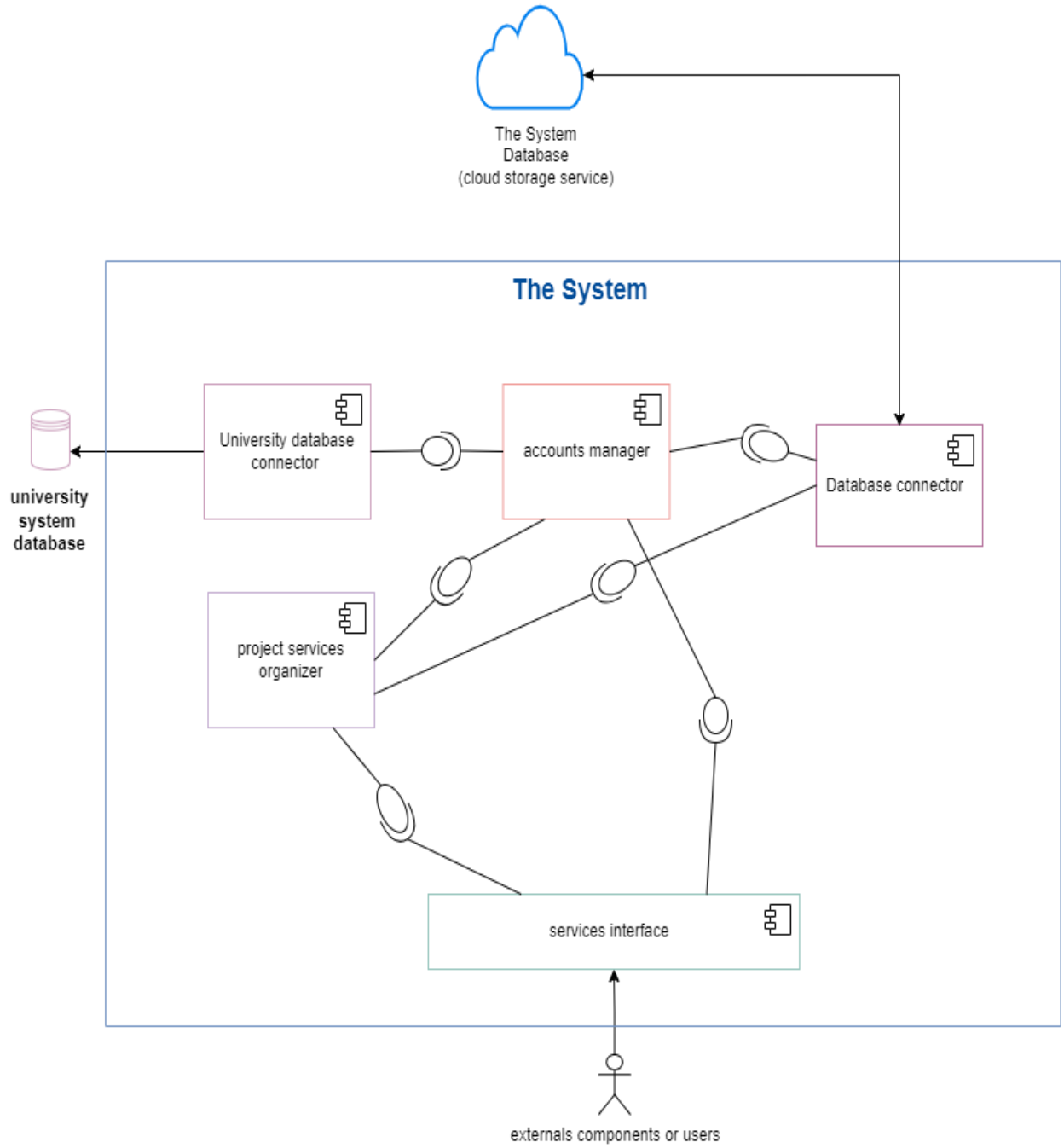
Class diagram:



Detailed class diagram:



Components Diagram:



Components responsibilities:

1. University database connector:
    - This component will access to the university database copy when a student wants to make an account to check its informations and whether he is a member in the university by his id.
    - Also, it will check whether a student met with the conditions of the project registration, that determent by the university.
    - And it will give the result to the “Account manager component” so it can make a new account for this member.

2. Account manager:
    - Made everything related to accounts management like make new account after taking the approve from “University database connector component”, Delete account, Edit account information.
    - It will give an interface for the “services interface component” only the output of its operations.
    - Also an interface to the “project services organizer”, before any system move it have to check account permissions.
3. Projects services organizer:
    - Handel and organize all aspects of projects logic from suggestion projects, adding projects, registration and requesting, to evaluation process.
    - It can interact with the database by the component “ database connector”.

- Also have an interface with the component “services interface”, which will connect the system with the outside components, and “account manager” that will also control account permissions.

4. Services interface:
    - This component will have an interface from all the last 2 components to have their services and presented to the user or external components so that they don’t have to know or communicate with all components otherwise only communicate with the “services interface component” and that will give a level of abstraction and encapsulation to all servicer.

5. Database connector:
    - This component responsible for communication with the cloud database, act as a bridge between other components and the database in the cloud, handling tasks such as establishing connections, executing database requests.
    - This gives a layer of security and more organization of how talking to the database.

Architectural Style:

Note: the system will be developed to be a web application.

MVC architecture:

1. view: is a visual representation of the MVC model. This level creates an interface to show the actual output to the user.

2. model: this level is very important as it represents the data, defines where the application's data objects are stored.

3. controller: is a level that acts as the brain of the entire MVC system. It receives user input from the view, processes it, and communicates with the model to perform data operations.

Why choosing MVC architecture:

- Faster development process.
- The modification does not affect the entire model.
- Ability to provides multiply views.
- Reusability.
- Separation of concerns.

Disadvantages of MVC architecture:

- Complexity.

Architectural Style: