Student: Andreea-Sabina Lazaroiu

**Group: 30432**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

2. Use-Case Model 3

3. System Architectural Design 3

4. UML Sequence Diagrams 3

5. Class Design 3

6. Data Model 3

7. System Testing 3

8. Bibliography 3

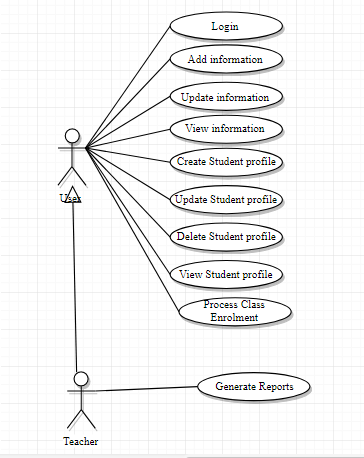
1. Requirements Analysis

# Assignment Specification

Design and implement a Java application for the management of students in the CS Department at TUCN.

# Functional Requirements

2. Use-Case Model



Use case: Update Information

Level: User-goal level

Primary actor: User (Regular User)

Precondition: The User has to login

Main success scenario:

1. The user logins successfully.
2. The user chooses to update his personal information.
3. The user modifies the current information.
4. The updated information is saved.

Extensions: The user selects to cancel the operation.

Use case: Generate report

Level: User-goal level

Primary actor: Teacher

Precondition: The teacher has to login and to select a student.

Main success scenario:

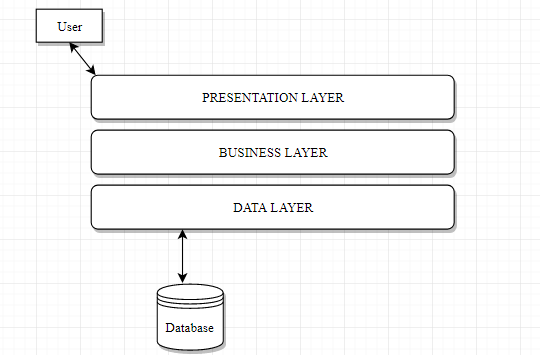
1. The teacher logins successfully.
2. The teacher selects a student.
3. The teacher selects a period.
4. The teacher generates a report based on the student’s account information.

3. System Architectural Design

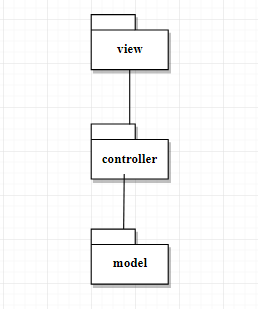
**3.1 Architectural Pattern Description**

The architectural pattern that will be used for this application will be the Layered Architecture Pattern. With this pattern we will logically group our components into separate layers that will communicate with each other. The logical separation will be done using packages.

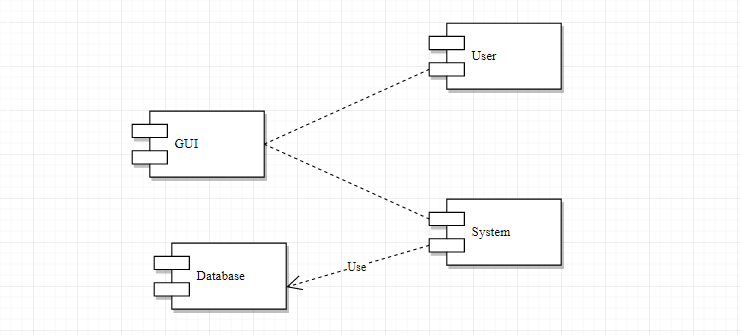
**3.2 Diagrams**



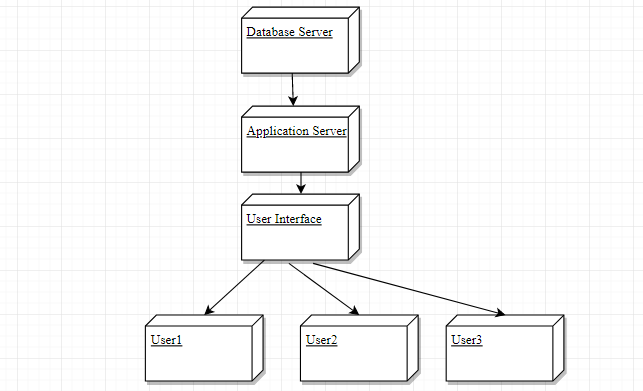
Package Diagram: We build the Layered Architecture using the Packages. We will have three packages: model, controller and view. In the model package we extract from the database the necessary information. In the controller package we satisfy the needs that come from the view package where the users ask requests through a graphical user interface.



Component Diagram:



Deployment Diagram:



**3.2 Diagrams**

*[Create the system’s conceptual architecture; use architectural patterns and describe how they are applied. Create package, component and deployment diagrams]*

4. UML Sequence Diagrams

*[Create a sequence diagram for a relevant scenario.]*

5. Class Design

**5.1 Design Patterns Description**

*[Describe briefly the used design patterns.]*

**5.2 UML Class Diagram**

*[Create the UML Class Diagram and highlight and motivate how the design patterns are used.]*

6. Data Model

*[Present the data models used in the system’s implementation.]*

7. System Testing

*[Present the used testing strategies (unit testing, integration testing, validation testing) and testing methods (data-flow, partitioning, boundary analysis, etc.).]*

8. Bibliography