Student: Breenghia Alexandra-Stefana

**Group:30641**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-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

The applicatin is designed to be used in CS department in TUCN, in order to better manage te students enrolled, courses and the grades from the exams. The application will have two types of users (student and teacher/administrator user) which have to provide an username and a password in order to use the application.

# Functional Requirements

*The regular user can perform the following operations: - Add/update/view client information (name, identity card number, personal numerical code, address, etc.). - Create/update/delete/view student profile (account information: identification number, group, enrolments, grades). - Process class enrolment (enroll, exams, grades). The administrator user can perform the following operations: - CRUD on students information. - Generate reports for a particular period containing the activities performed by a student*

# Non-functional Requirements

* *The data will be stored in an SQL server database*
* *The Layers architectural pattern will be used to organize the application. Data Access Layer will use SQL statements*
* *All the inputs of the application will be validated against invalid data before submitting the data and saving it in the database*

2. Use-Case Model

*Use case: Student login*

*Level: user-goal level*

*Primary actor: Student*

*Main success scenario: the user opens the application, introduce the credentials, and the specific page is opened (student main form)*

*Extensions: if the database is inaccessible, a message will popup. If the credentials introduced are wrong, the user will be informed.*

*Use case: Generate report*

*Level: sub-function*

*Primary actor: Administrator*

*Main success scenario: the user logins into the application, selects the student for witch will generate the report and the application generates it.*

*Extensions: In success, the report will be generated. If there is a failure (db connection went off for example) a message will be displayed.*

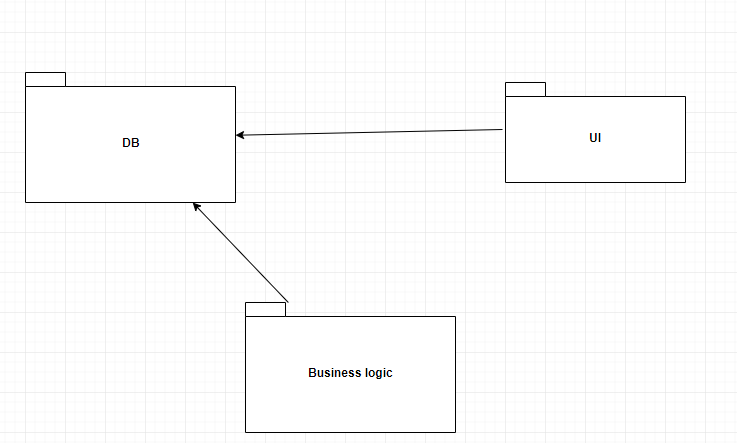
3. System Architectural Design

**3.1 Architectural Pattern Description**

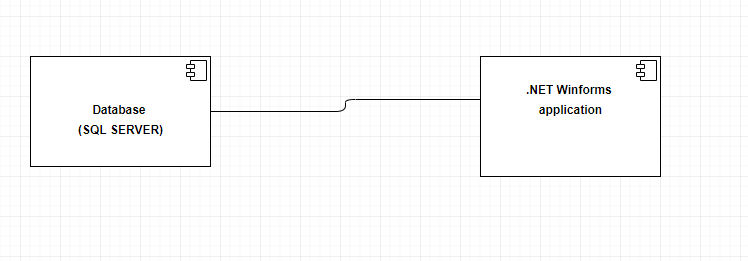
*Components within the layered architecture pattern are organized into horizontal layers, each layer performing a specific role within the application (e.g., presentation logic or business logic). Although the layered architecture pattern does not specify the number and types of layers that must exist in the pattern, most layered architectures consist of four standard layers: presentation, business, persistence, and database. In some cases, the business layer and persistence layer are combined into a single business layer, particularly when the persistence logic is embedded within the business layer components. Thus, smaller applications may have only three layers, whereas larger and more complex business applications may contain five or more layers.*

**3.2 Diagrams**

***Package diagram***



*Component Diagram*



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