Assignment 1
Analysis and Design Document

Student: George Cimpoies
Group: 30432

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 6

5. Class Design 7

6. Data Model 7

7. System Testing 8

8. Bibliography 8

1. Requirements Analysis

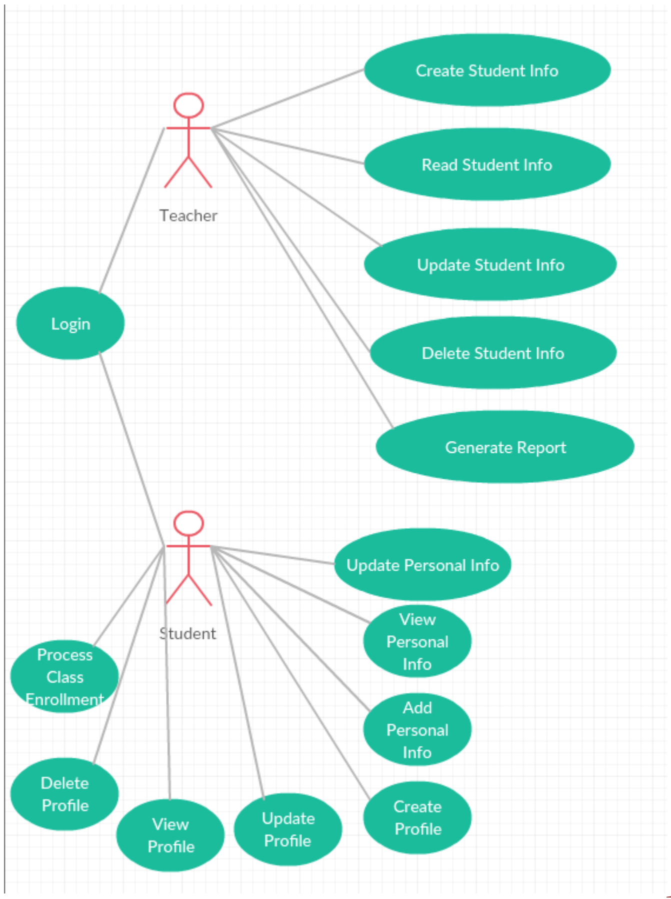
1.1 Assignment Specification

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

1.2 Functional Requirements

The application should have two types of users (student and teacher/administrator user) which have to provide a username and a password in order to use the application.
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.

2. Use-Case Model



]

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.

Use case: **Update Personal Information**
Level: User-goal level
Primary actor: Student
Precondition: The Student has to login
Main success scenario:
1. The student logins successfully.
2. The student chooses to update his personal information.
3. The student modifies the current information.
4. The updated information is saved.
Extensions: The user selects to cancel the operation.

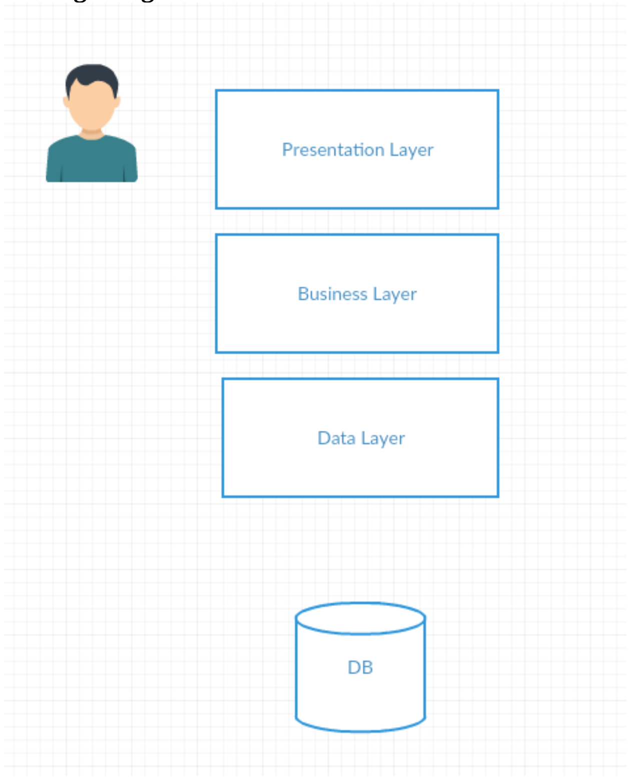
3. System Architectural Design

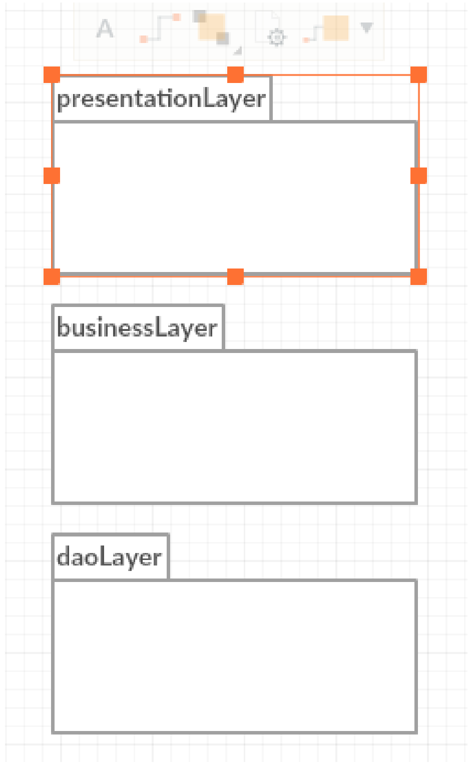
3.1 Architectural Pattern Description

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.

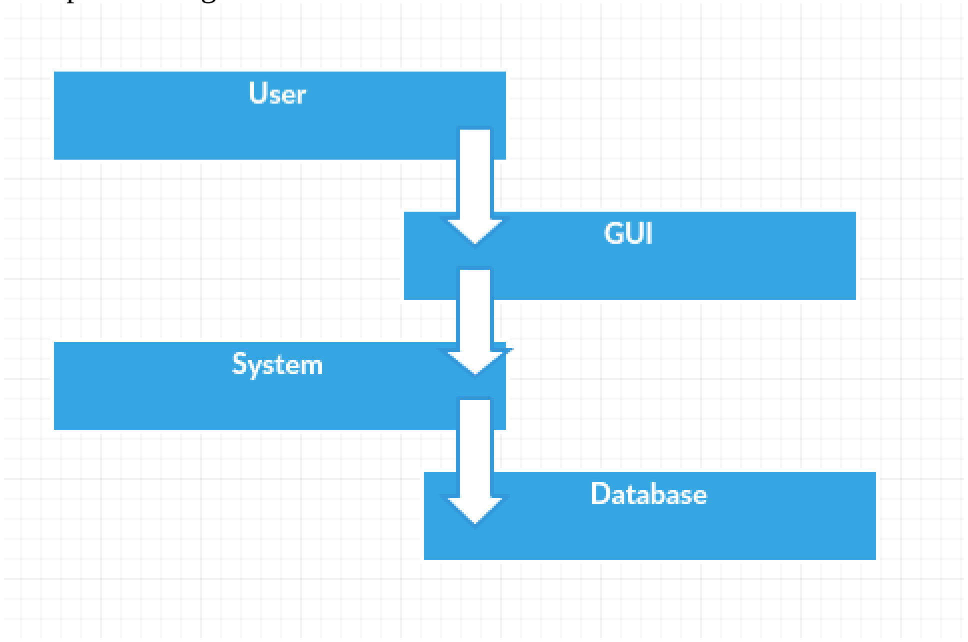
3.2 Diagrams

Package diagram

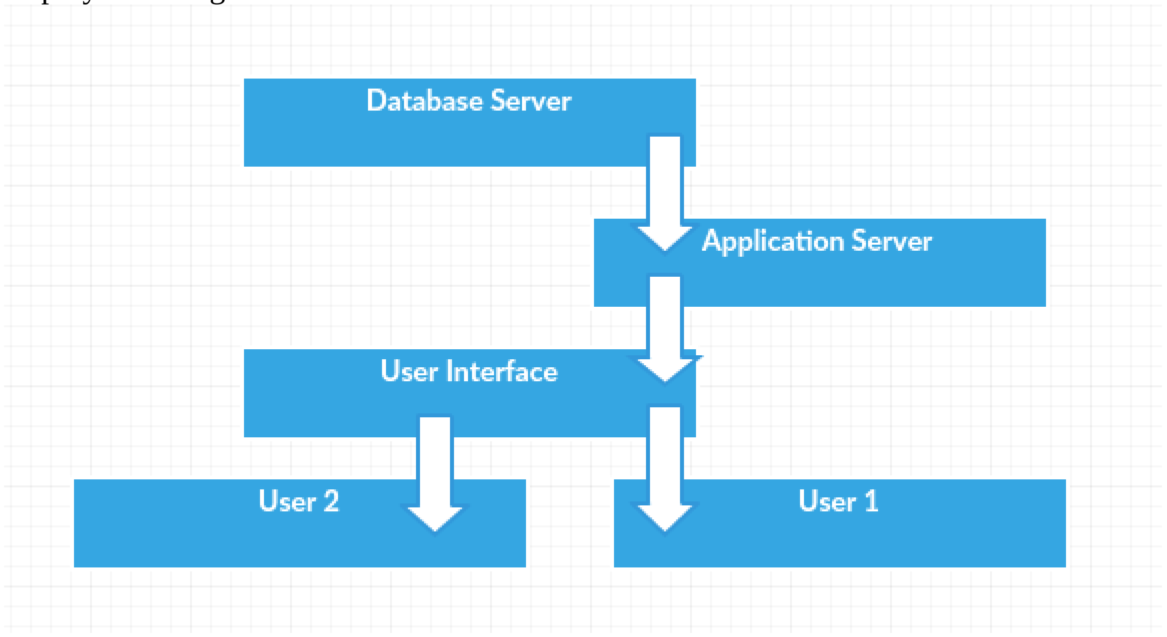




Component diagram:

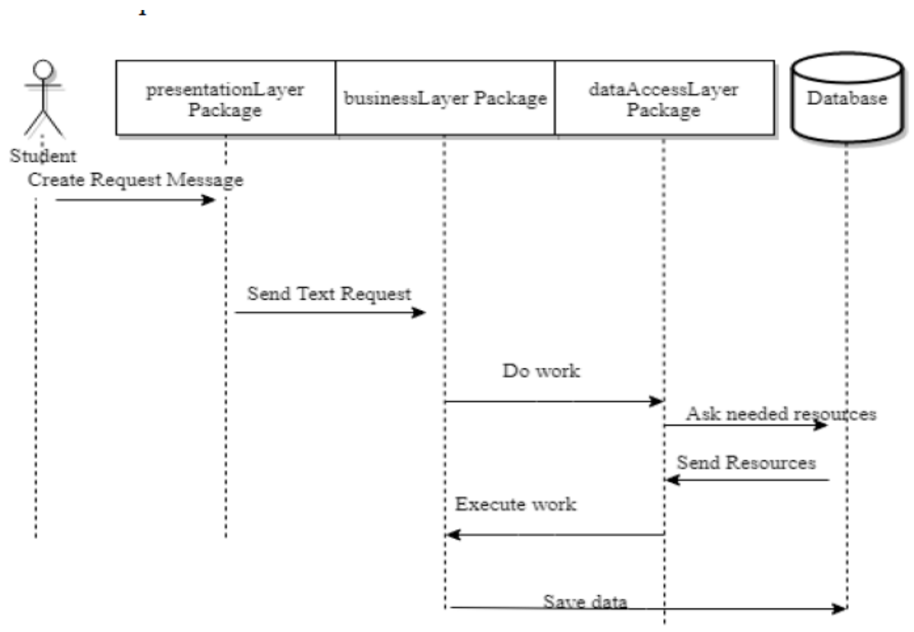


Deployment diagram:



4. UML Sequence Diagrams

Update Personal Info

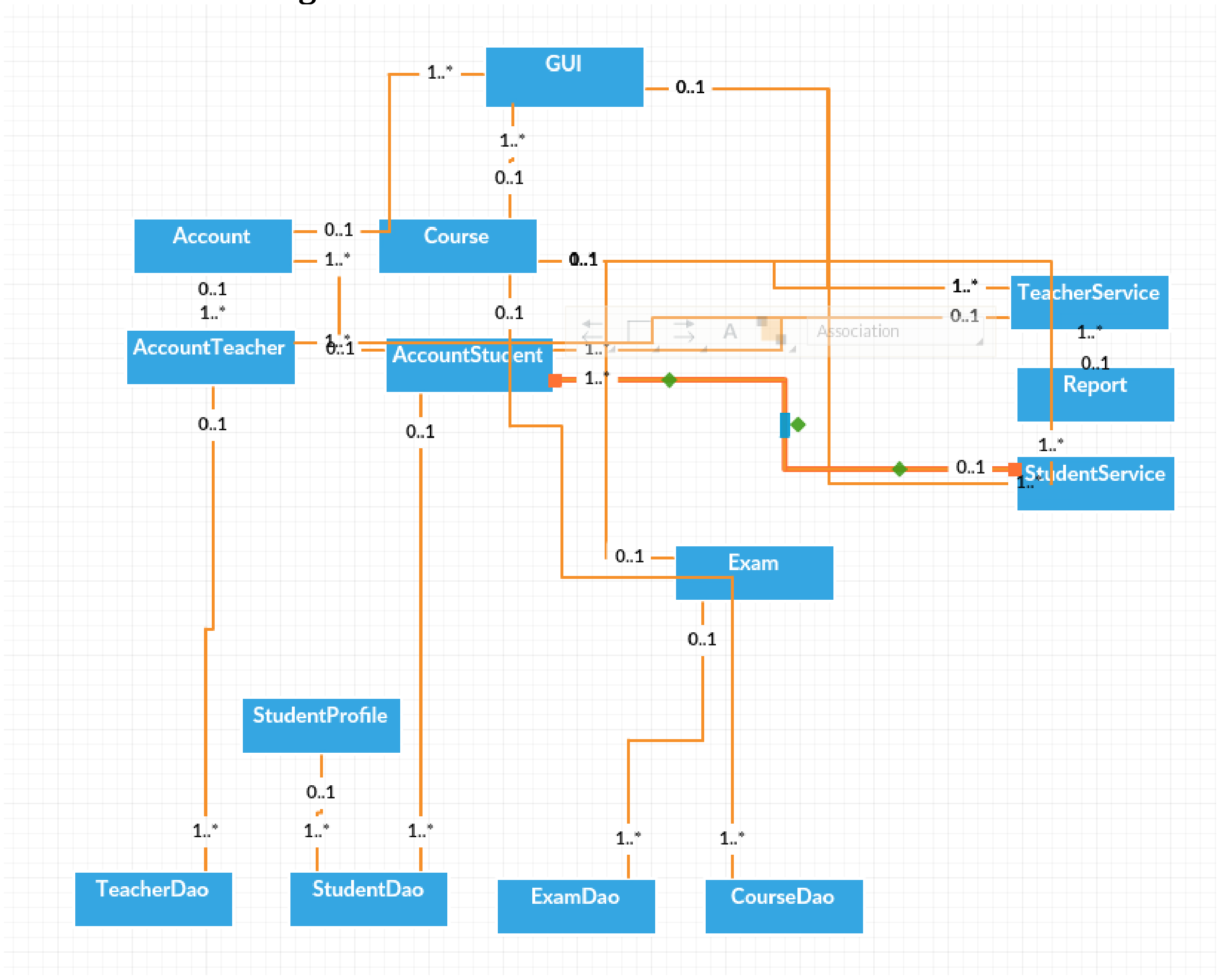


5. Class Design

5.1 Design Patterns Description

MVC – Model-View-Controller pattern – one of the most used design patterns to organize an application into three logical parts:
Models - data entities used throughout the application
View(s) – refers to the GUI side of the application
Controller – the "heart and brain" of the application

5.2 UML Class Diagram



6. Data Model

A data model consists of three different phases: Structural, Manipulative and Integrity.
Structural part: Consisting a set of rules
Manipulating part: Types of operations allowed, such as updating, retrieving, and changing the database.
Integrity part: Validate the accuracy of data.

7. System Testing

The application will be tested using Junit framework, specifically Junit 4.

8. Bibliography

https://dzone.com/articles/junit-tutorial-beginners
https://www.umsl.edu/~sauterv/analysis/Fall2010Papers/varuni/