Addis Ababa University
Faculty of Computer and Mathematical Sciences
Department of Computer Science

Software Engineering (Cosc3071) Course Project
Semester 2, Year 2015/16

Software Design Document

School Management System specific to Grading and Registration

For

Saint Joseph School, Addis Ababa, Ethiopia

Submitted by:
Kirubel Girma/ NSR-1572-07\
Nahom KidaneMariam/ NSR/4556/07
Melaku Habtu/ NSR/7979/07
Mickyas Getachew/ NSR/6748/07
Nathan Damtew/ NSR-0950-07
Wakigari Garoma/ NSR-1572-07

Submitted to:
Dr. Dagmawi Lemma

29/12/2016
Addis Ababa
Ethiopia

Revision History

Author/Reviewer	***Version***	***Status***	***Date***	***Changes***
Wakigari Garoma	1.0	DRAFT.	1/2/2017	• Original Document

Table of Contents

1. Introduction

1.1. Overview

This design document is made to specify the design details of the system. We will look at an overview of the current system and how it works. We will be looking at the design goals we have in terms of performance, dependability, cost, maintenance and End user criteria. We will decompose the new system into subsystems and components and will look at the datatypes and variables used in the classes. The relating techniques and architecture type of the new system will be defined as well.

1.2. Definitions, Acronyms, and Abbreviations

Definitions

Example:

Term/Phrase	**Definition**
Admin	A staff person who comments on attendances and can generate reports
Student	An end user of the system, who takes assignments and exams
Secretary	A person who types exams, takes attendance and generates reports
Cashier	A person who issues receipts to parents/Guardians who have paid the registration fee

Acronyms and Abbreviations

Term/Phrase	**Definition**
UI	User Interface
DB	Database
Waterfall Model	A model whereby each step in the SDLC is completed before moving to the next step
SDLC	Software Development Life Cycle

1.3. References

- Bernd Bruegge, Object Oriented Software Engineering with UML, Patterns and Java
- Requirement Specification Document of this School Management System

2. System Overview

The current system works in the following manner. A student comes to register to the registrar office. The student will be subjected to different requirements depending on the grade he/she wants to enter (either Grade 1 or 7 or 9).

If a student is applying for first grade he has to fulfill the following requirements

- Age should be between 5 and 7
- Student should have a valid birth certificate to prove age
- Should take exam and pass, and due to the large amount of applying students, he/she should draw lots and win.

If the student is applying for either Grade 7 or 9, then

- Student should have a GPA greater than 75
- Student should have conduct A
- Student should pass exam

Assuming the student gets in, the student is assigned a section (randomly assigned by the secretary of his/her department, i.e. High School, Elementary or Junior). There are four quarters, and on each quarter the student takes tests, assignments and final exams. The teacher is responsible for generating the above assessments.

Attendance is taken every day by a student representative chosen by the home room teacher of the section. This representative then takes it to the secretary every morning to be filled into the database. If a student is late or absent, he or she should bring a valid reason or a sick leave document. If this happens 5 times in a year, the student should bring his or her parents.

The secretary is responsible for generating report cards and attendance reports. She uses Excel and Access to store grades submitted by the teachers, and attendance lists. Excel will automatically generate the reports.

Payment report is generated by another office, using Access. As new students are registered into the school, they are added into the database. As the school year goes on, the list of students is reviewed and the program generates a list of outstanding students (students who have not paid semester fees). The officer notifies the appropriate secretaries, who in notify the director. The director goes to class, calls out the above listed students and tells them to complete their payments. The database is updated as the student makes the payment, and finally a report is generated, useful for auditing purposes.

The purpose of the new system is automation, so the conceptual model of the two systems will be the same.

3. Design Considerations (Design Goals)

We will look at design goals through the following criteria's

- **Performance**

The registration subsystem to be used by the registrar office should have a faster response time, as parents/guardians form queues and do not want to be kept waiting. The database does not include much more than the records of students and teachers, so it an average desktop hard drive should suffice. The report generating system should be able to handle generating reports of all students at once.

- **Dependability**

The system needs to be fairly dependable. System failure will not endanger human lives in any way, but failure should be avoided at all

costs. It should be available as long as a user needs to use it. It should recognize invalid commands and show appropriate messages. The system should be secure in that unauthorized users should not access unauthorized data.

- **Cost**

Upgrading the system will cost time as records need to be copied to the new system. There will be minimum cost of training. Enhancements and bug fixes shall be done as necessary.

- **Maintenance**

Should the system fail or should any problems arise, it should be addressed by the developers in the appropriate time. The system should be designed to be robust and not fail. It is easy to adapt to the interface and only basic training will be necessary. There should be clearly defined cohesive modules, implemented as simply as possible to add functionalities or classes. The system shall be richly-commented to allow readability.

- **End user criteria**

The whole point of the system was automation, meaning the users will have an easier time doing their jobs. End user satisfaction is a big factor when designing the system. It should be efficient enough to be faster than the previous system, while secure enough to maintain integrity of sensitive data.

3.1 Assumptions and Dependencies

- **Performance**

Hardware requirements are:

a) Processor: Minimum Pentium 2 266 MHz processor
b) Internet Explorer 9 or above, Firefox, Chrome.
c) OS: Windows 7/8/8.1/10
d) 2GB RAM or greater

The above are minimum requirements assumed to comfortably run an application using Java.

- **Dependability**

Necessary resources like PC's are assumed to be delivered from the school.

- **Maintenance**

All code should be written in accordance with the Camel Case standard.

- **End user**

We assume the end users have basic computer knowledge to use the system.

3.2 Development Methods

We have implemented the waterfall methodology. We have progressively moved through the steps of project planning, requirement analysis and design. We decided to use this approach because the client did not want to change the scope of the project, and thoroughness was a more important goal than speed, unlike Agile Development. For more details on Waterfall Methodology, follow this link.

https://en.wikipedia.org/wiki/Waterfall_model

4. Proposed Software Architecture

We have divided the system into a couple of subsystems based on their functionality.

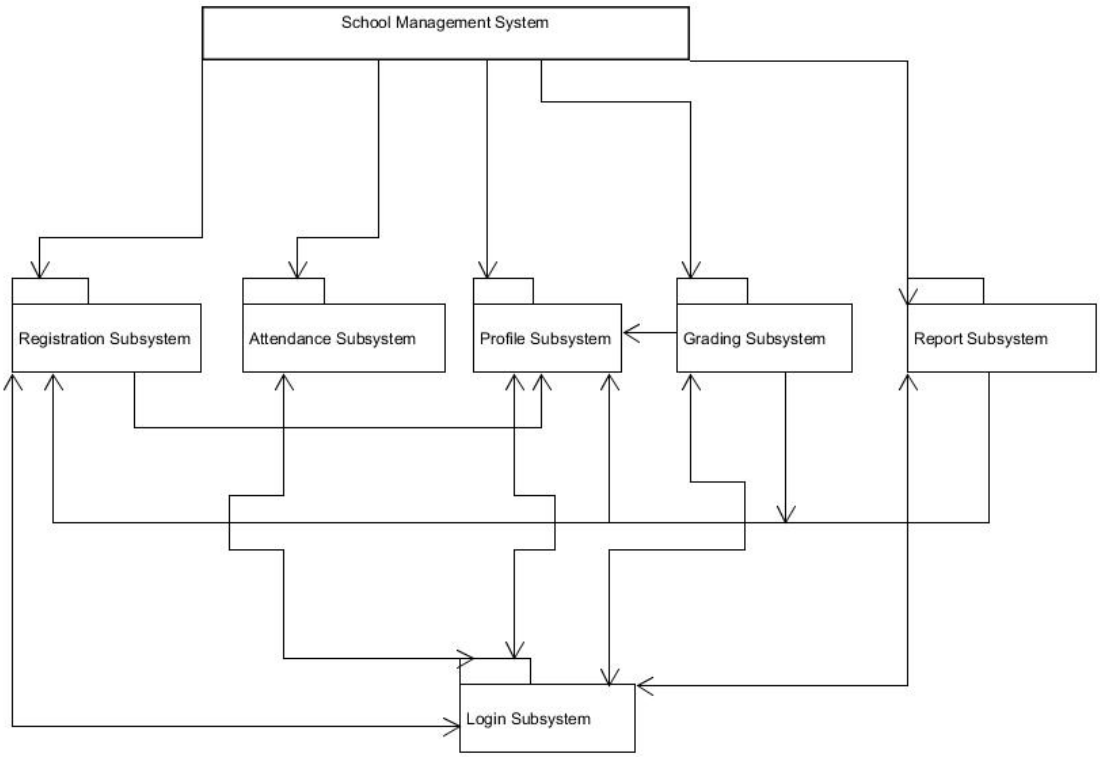
The registration subsystem handles the incoming of new students, the requirements that they should meet (according to previously defined business rules) and adding a registered student to the database.

Attendance subsystem has different classes for teacher and student attendance, and another class records the day and the status (isAbsent, isLate) of the student or the teacher. This is also written into the database. The profile subsystem records the profiles of new students, teachers, admins and secretaries. It has methods allowing it to view the existing staff and student profiles by searching using ID.

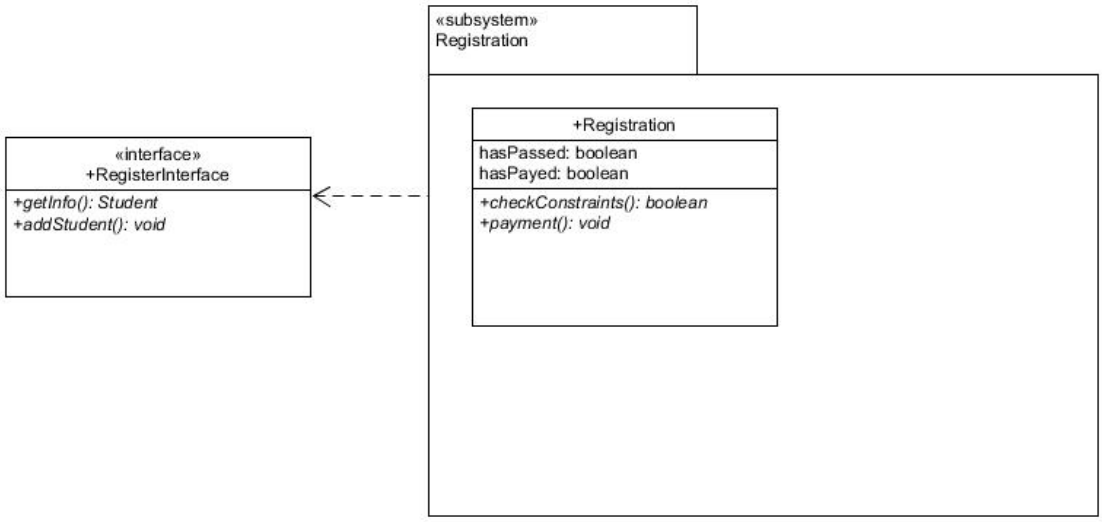
The grading subsystem allows uploading assignment files, test dates and assignment dates and the description of the assessment (how much marks it holds, what chapters are included, etc.) for each subject and corresponding teacher. The report subsystem is where the attendance, registration fee payment, mark list and report cards are made. For example, for report cards, an array of subjects and grades for each subject are passed for a particular student, and is finally displayed in a tabular manner.

The User Interface subsystem basically handles login authentication and implementation of interfaces.

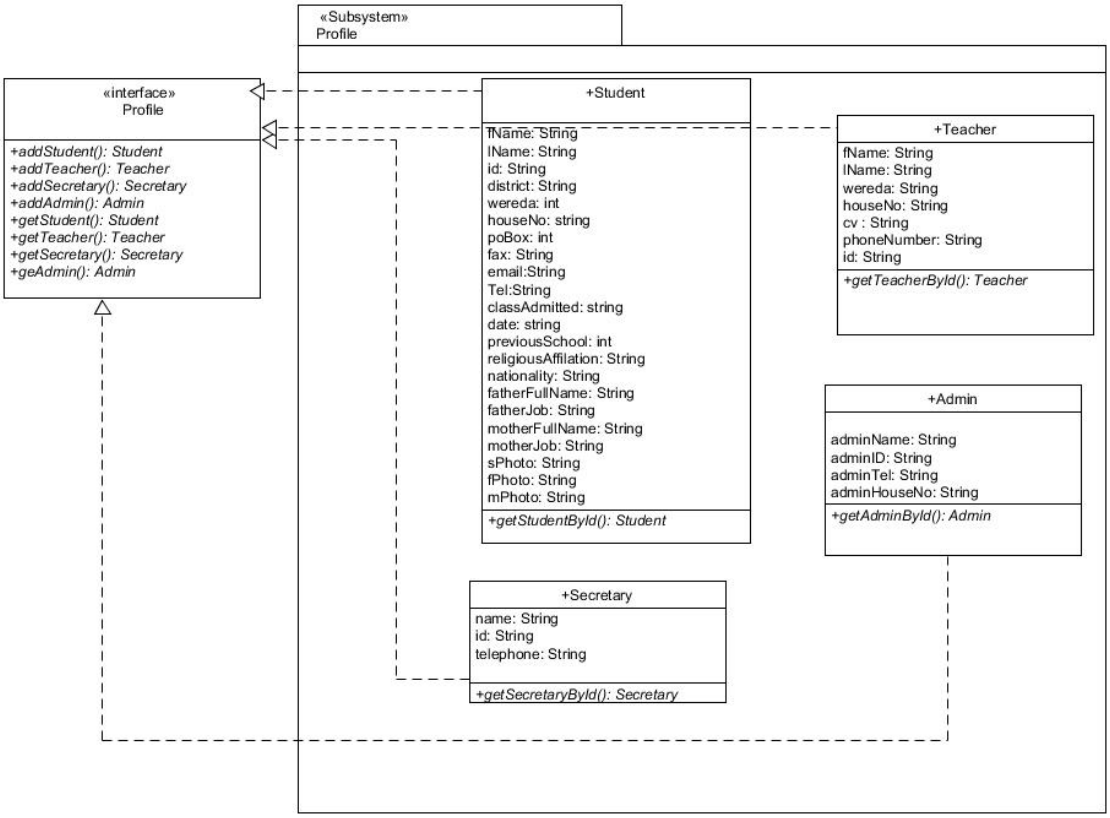
4.1. Subsystem decomposition



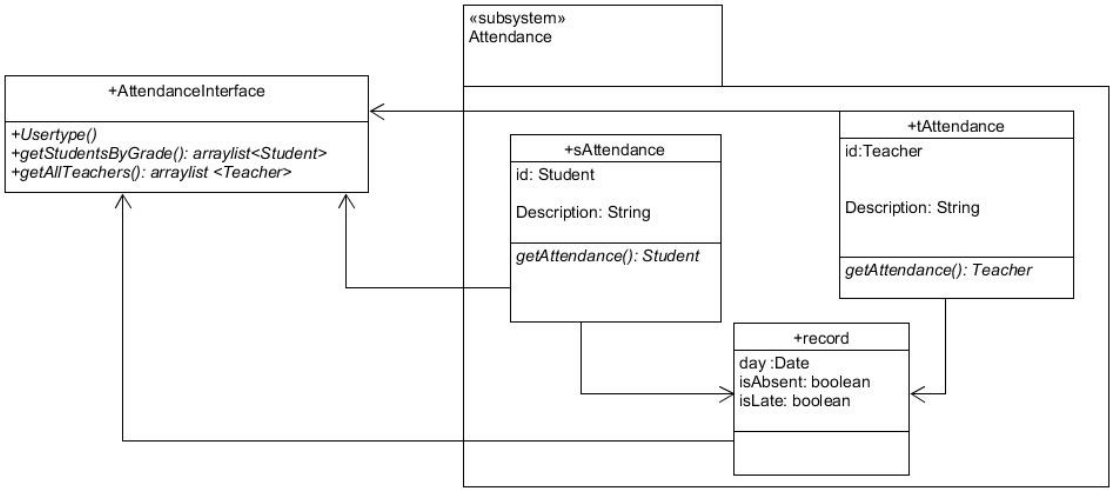
4.1.1. Registration Subsystem



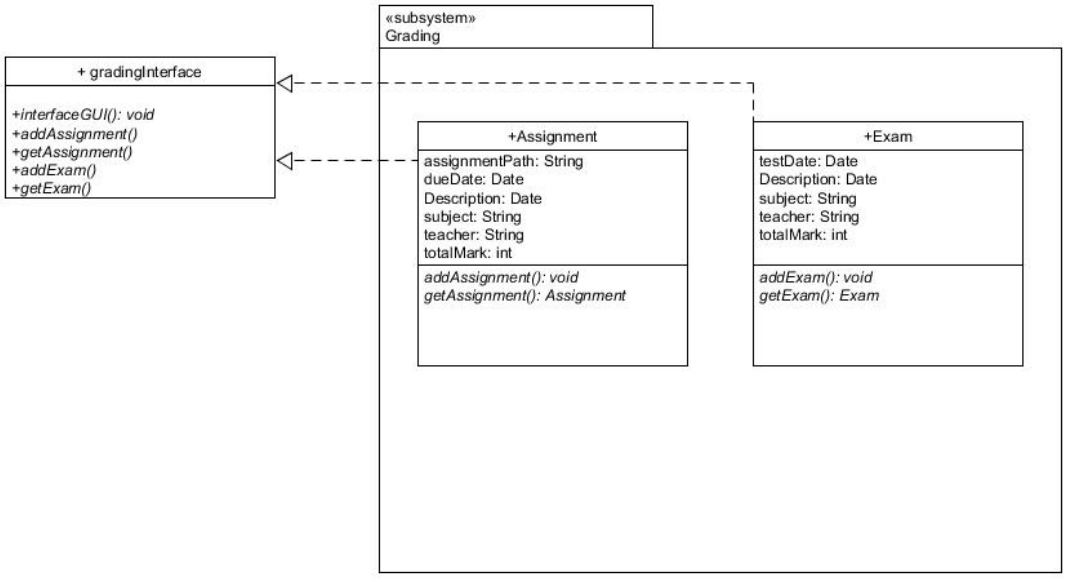
4.1.2 Profile Sub System



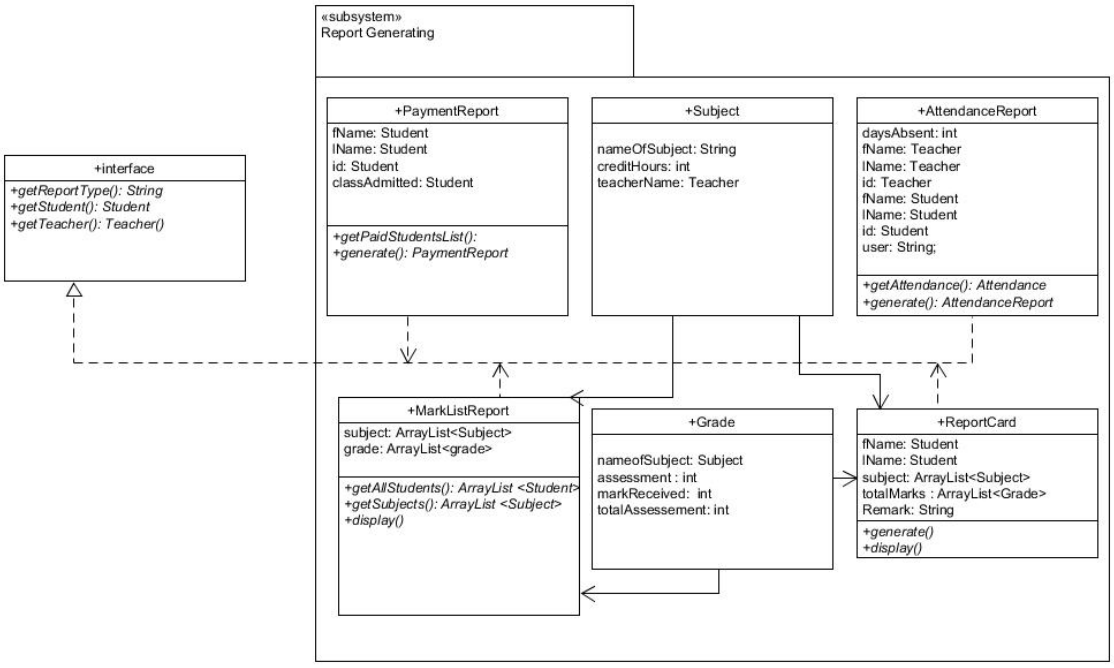
4.1.3 Attendance Sub System



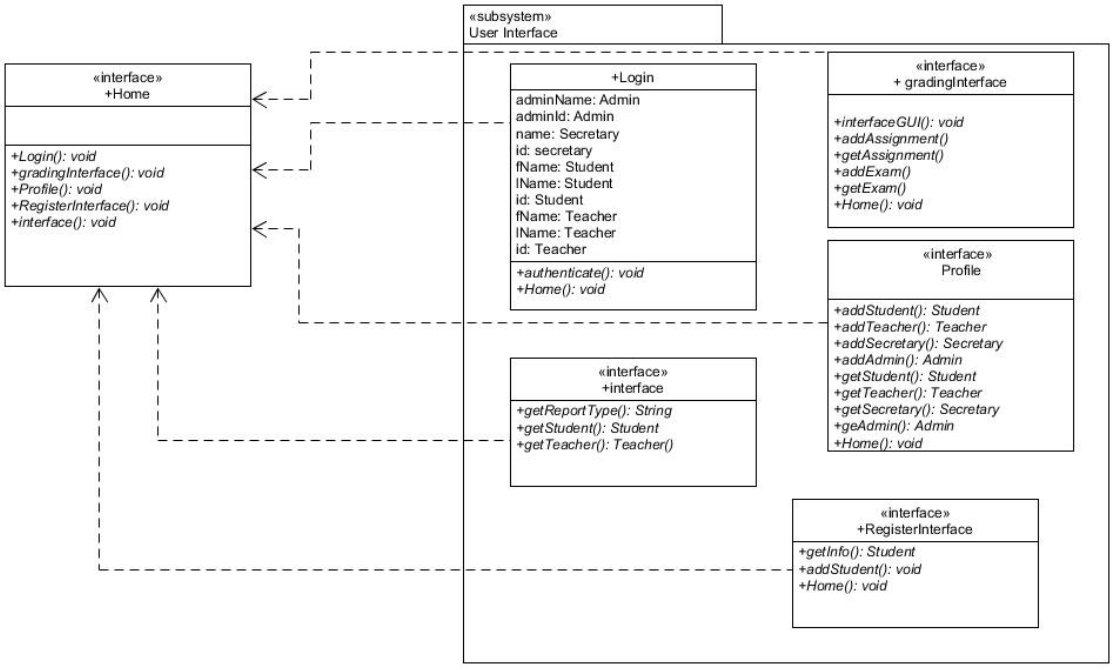
4.1.4 Grading Sub System



4.1.5 Report Subsystem



4.1.6 User Interface Sub System

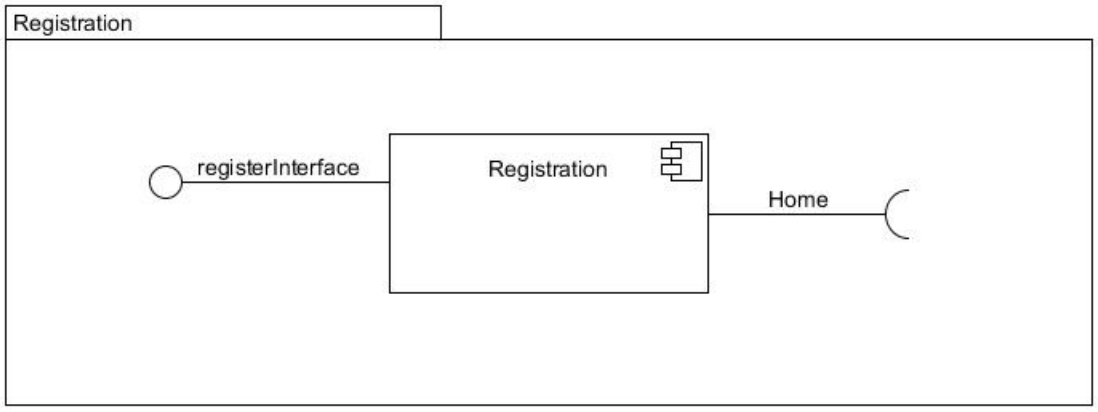


4.2 Subsystems Relating Technique

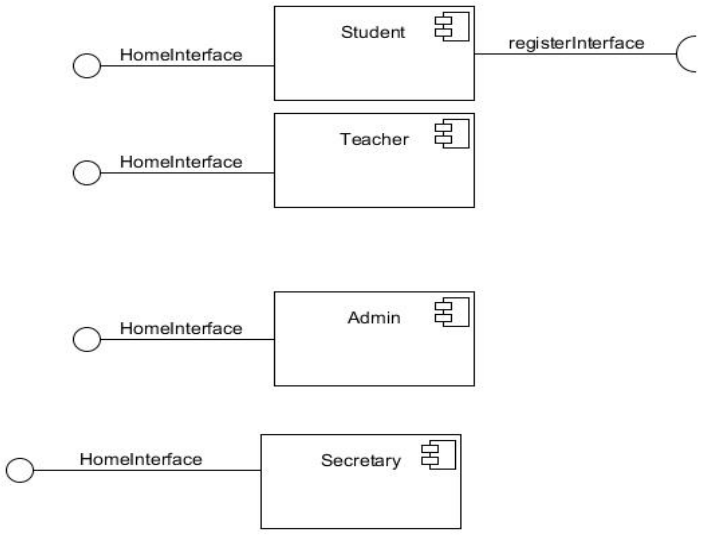
The above described subsystems will be related to each other in partitions, to allow for communications among subsystems on the same level. Most subsystems which are on the same level are interconnected. They use classes directly below them as well as next to them; for example, all interfaces of subsystems call the User Interface Subsystem and this subsystem calls subsystems on the same layer as it is. We also adopted an open architecture because classes should not be limited by only one layer above or below them.

4.3 Components

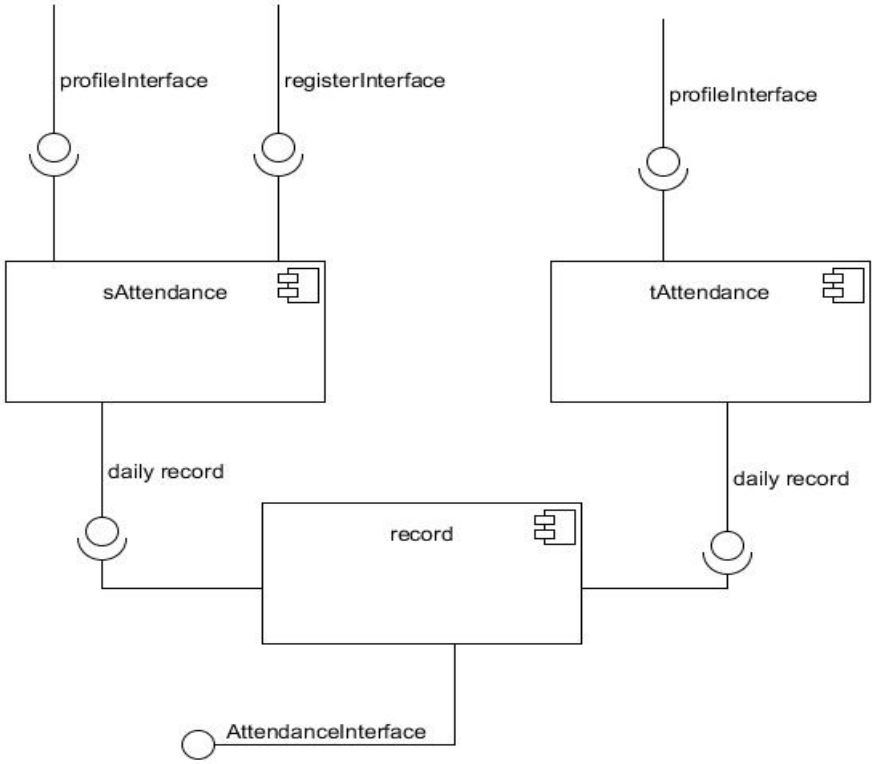
4.3.1 Registration Component



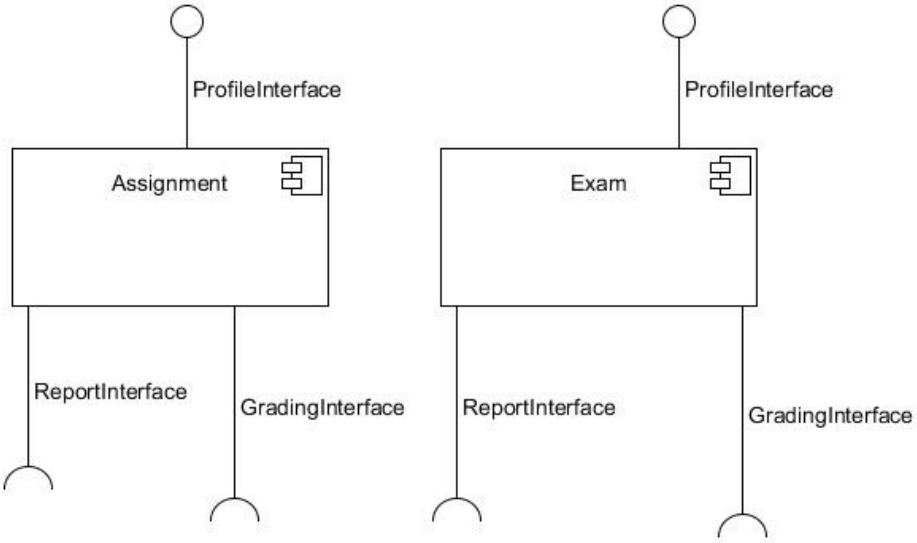
4.3.2 Profile Component



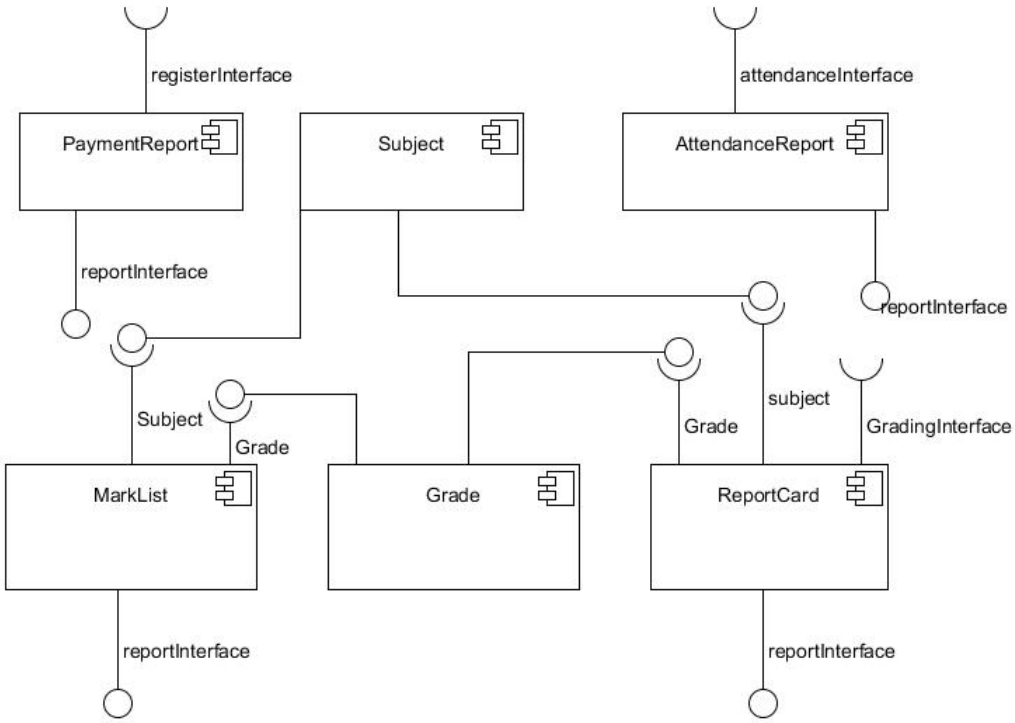
4.3.3 Attendance Component



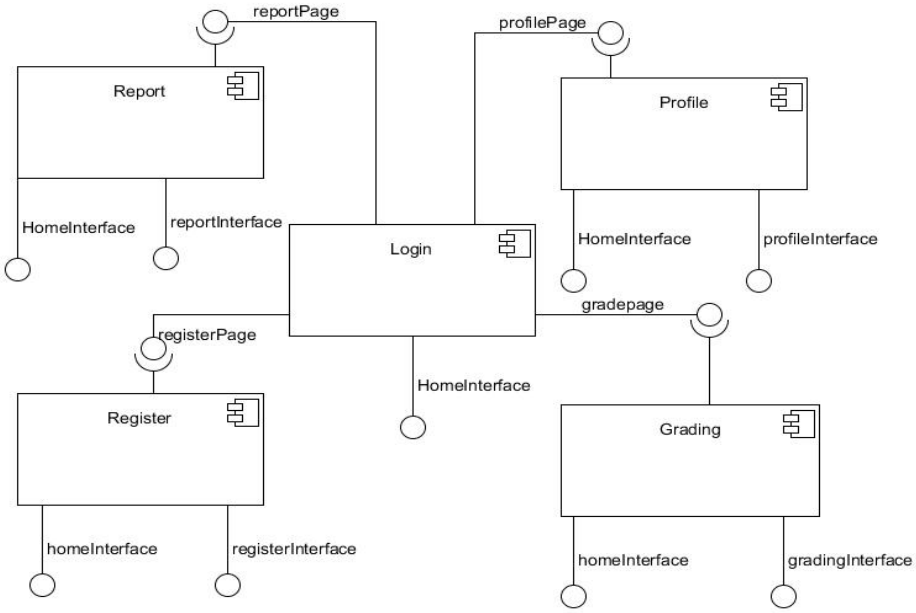
4.3.4 Grading Component



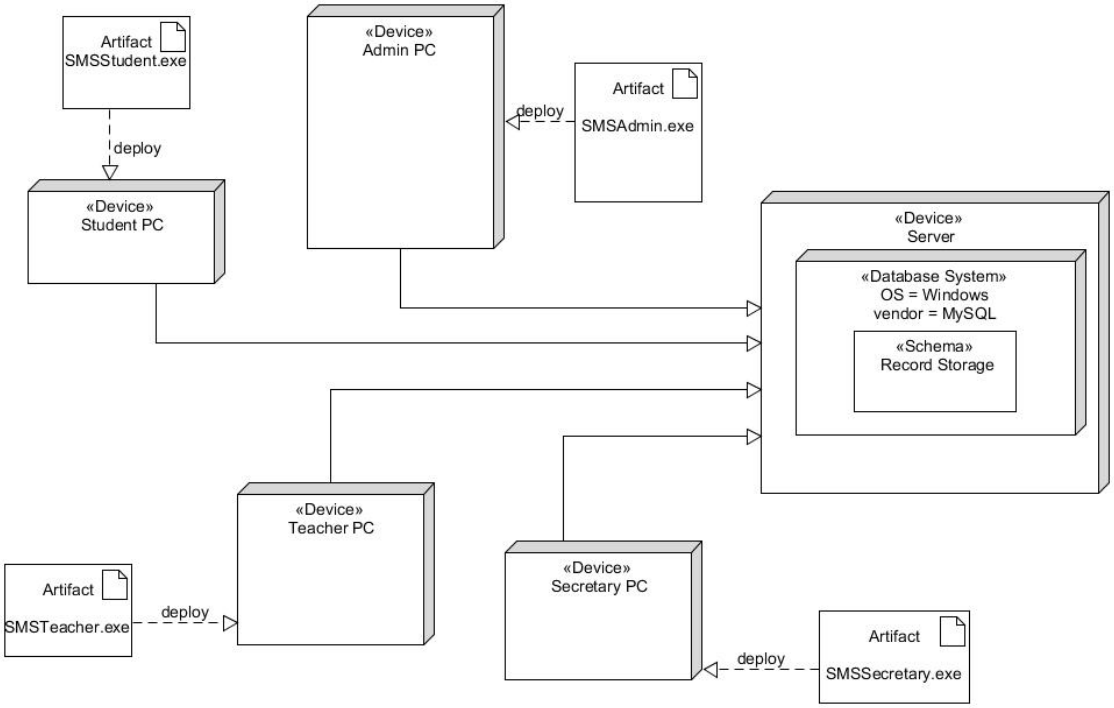
4.3.5 Report Component



4.3.6 User Interface Component



4.5 Hardware/Software Mapping



4.6 Persistent Data Management

We decided to use database as storage instead of Files because of the level of detail access required. Databases will enable us to query reports, insert and remove records easily. MySQL is open source, and the free version has all the functionalities that we need to implement in our system. It is faster than MS SQL Server, and about equally secure, so it would be an important asset for the registrar system.

The following business rules are configurable and the information shall be stored in a file called **schoolmanagementsystem.ini.** It is a text file that shall be read upon system initialization. Each configurable part of the business rule shall precede a comment describing the purpose and the meaning of the configuration attribute.

Business rule: *All Grade 1 students must of age 5-7*

Representation in the **schoolmanagementsystem.ini:**

- *studentAge < 7 && studentAge > 5*

Business Rule: *All students registering for 7th and 9th grade should have a GPA>75 and have conduct A*

- *gpa>5 && conduct = 'A'*

Business Rule: *students who have not paid cannot register*.

- *Receipt? Register() : display ("Not paid!");*

4.7 Access Control

The system is authenticated using a username and a password (different from publicly known ID). The user interface asks for the user's permission type and provides a layout to enter login information.

Annex

Annex 1. Web Server Specification

Server = Apache server

Version = 2.4.23

Annex 2. Database Specification

Vendor = MySQL

Version = 5.7.14