**Document: Software Design Document**

**Written By: Casey Boatman & Justin Andras**

**Submission Date: April 26th, 2017**

**Team Name: Gold Team**

1. **Introduction**
   1. **Purpose**

This SDD provides the design details of the Gold Team’s semester project for the Above Average University Portal. The expected audience are the students in the Software Engineering course, Professor Zhang, and the Gold Team that developed and maintained the project.

* 1. **Scope**

This project was built for educational purposes in our Software Engineering course, it will not be for sale. AAU is a fictional university. The architecture was chosen as a learning opportunity. We did not want to go the direction of implementing this in a way that any of us have in previous courses (such as using JSPs like the majority of us did in Database Systems).

* 1. **Definitions, acronyms, and abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| AAU | Above Average University |
| Gold Team | Team of developers working on this project. Includes:  Phil Parker  Ryan Peters  Jacob Hertl  Aaron Tonkovich  Casey Boatman  Justin Andras  Joshua Potrawski  Jonathan Nowak |
| Repo | Repository. Type of class in our project. |
| SDD | Software Design Document |

1. **References**
2. **Decomposition description**

**3.1 Module decomposition**

Since our project only has one “package”, this SDD will break it down into types of classes

* 1. **Entry Point (.java file)**

The AauEntryPoint file is the heart of the project. Without this main file, nothing else will work. This is where everything for the project is setup, including things such as setting environment variables, instantiating the repo’s, created the JSONParser and sql2o objects, and more. This also defines all of the pages that are to be/can be navigated to.

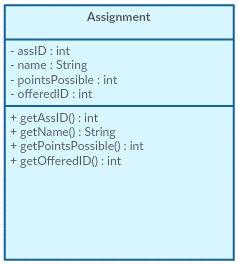
* 1. **Models (.java files)**

The Model classes are what hold our data. Data is returned from our database using a sql2o object, and is placed into the corresponding model. Section 3.3 of this document shows each model’s database counterpart.

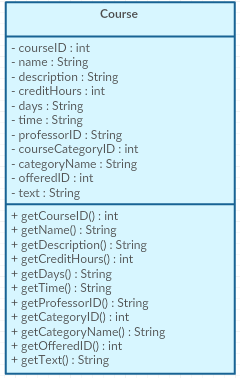
* + 1. **Announcement**



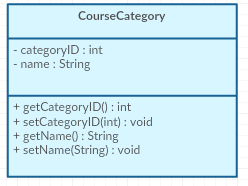
* + 1. **Assignment**



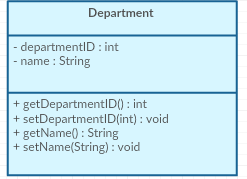
* + 1. **Course**



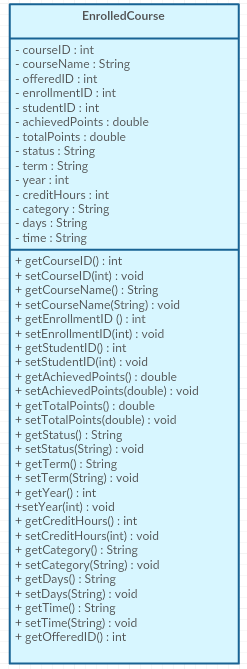
* + 1. **Course Category**



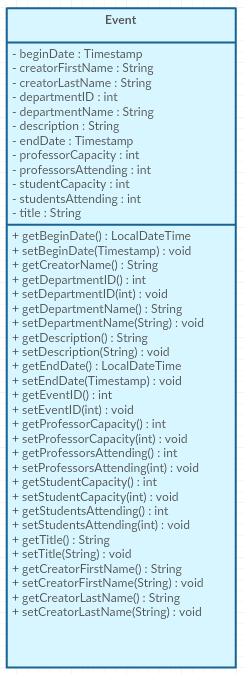
* + 1. **Department**



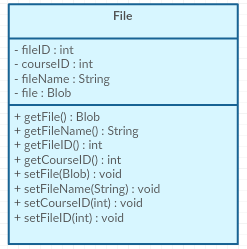
* + 1. **EnrolledCourse**



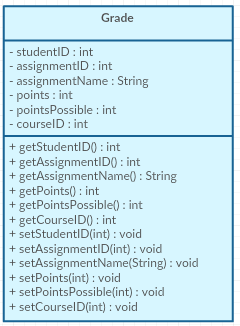
* + 1. **Event**



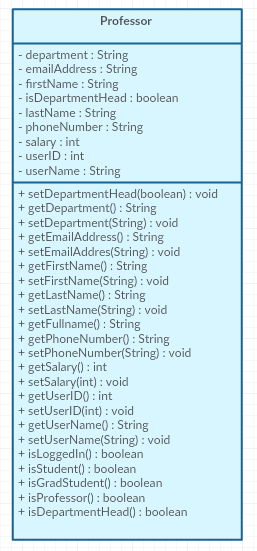
* + 1. **File**



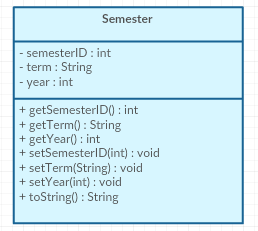
* + 1. **Grade**



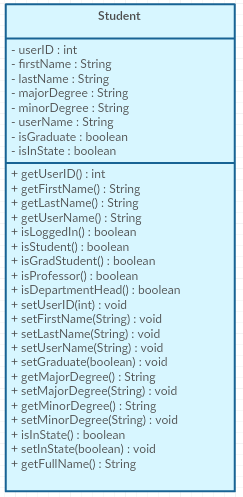
* + 1. **Professor**



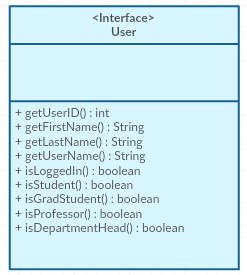
* + 1. **Semester**



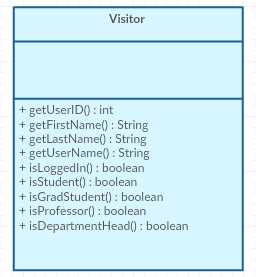
* + 1. **Student**



* + 1. **User**



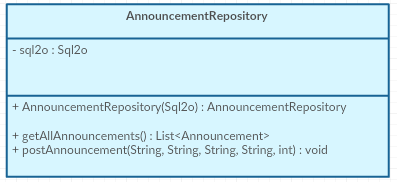
* + 1. **Visitor**



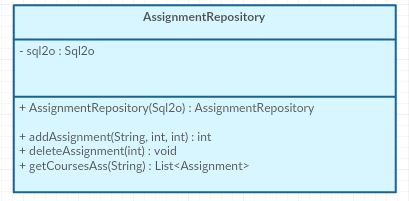
* 1. **Repos (.java files)**

The Repository classes are what fetch data from the database that are then put into the model classes. Repositories house multiple get methods to grab the data different ways.

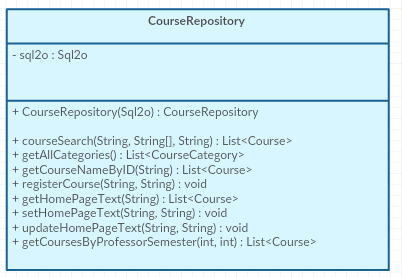
* + 1. **AnnouncementRepository**



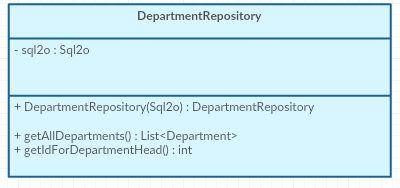
* + 1. **AssignmentRepository**



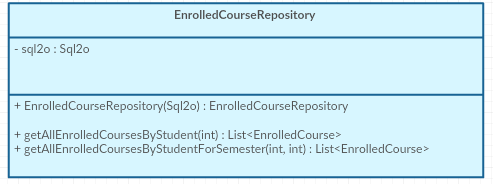
* + 1. **CourseRepository**



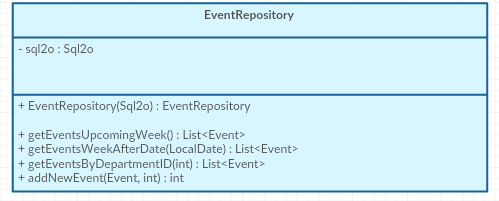
* + 1. **DepartmentRepository**



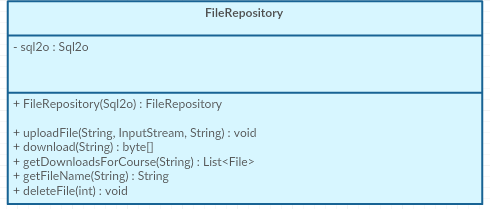
* + 1. **EnrolledCourseRepository**



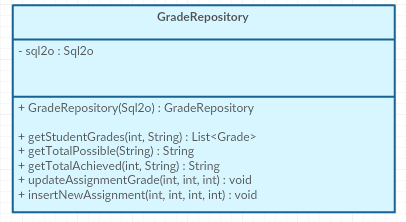
* + 1. **EventRepository**



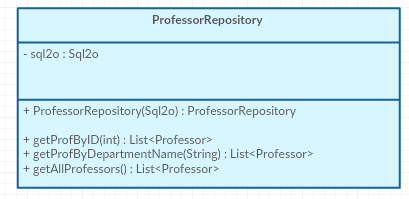
* + 1. **FileRepository**



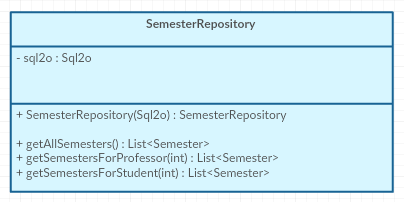
* + 1. **GradeRepository**



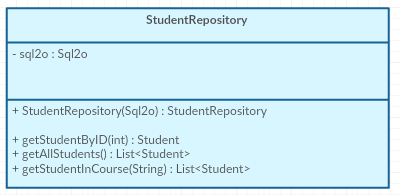
* + 1. **ProfessorRepository**



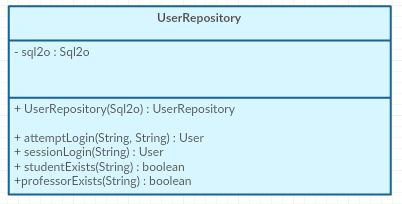
* + 1. **SemesterRepository**



* + 1. **StudentRepository**



* + 1. **UserRepository**



* 1. **Templates (.ftl files)**
     1. **Base**
     2. **Course-details**
     3. **Course-search**
     4. **directorySearch**
     5. **faculty**
     6. **homepage**
     7. **studentPortal**

**3.2 Concurrent process decomposition**We have a backend process that spawns a new thread for each http request to render output data.

**3.3 Data Decomposition**

We use repositories that depend on the model classes they export.

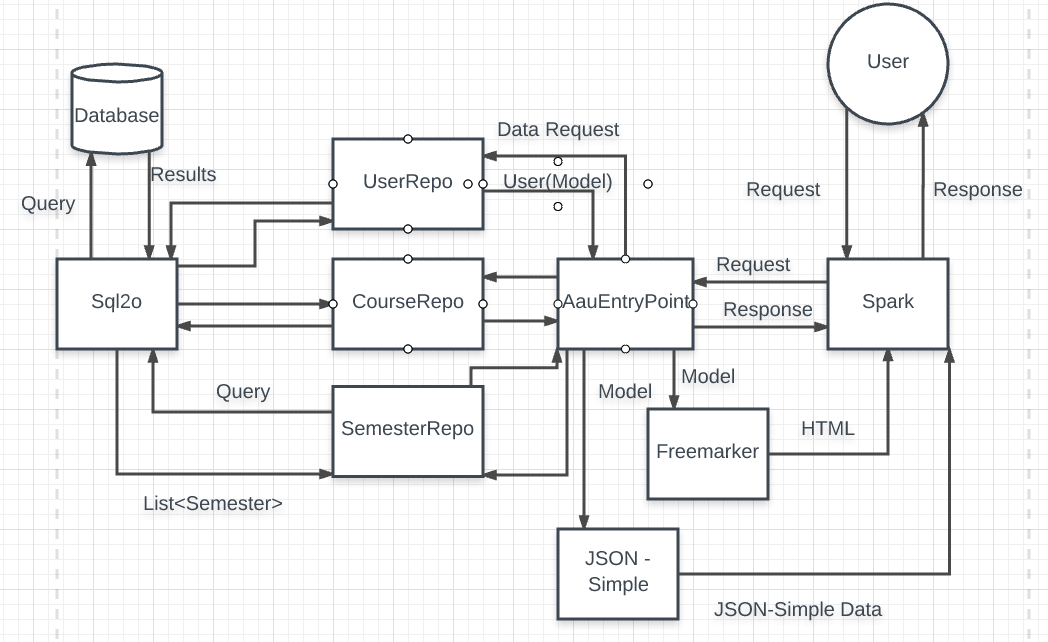
|  |  |
| --- | --- |
| **Model** | **Database Counterpart** |
| Announcement | Faculty\_Announcements |
| Assignment | Assignments |
| Course | Courses |
| CourseCategory | Course\_Categories |
| Department | Departments |
| EnrolledCourse | Student\_Enrollment |
| Event | Events |
| File | Course\_Files |
| Grade | Grades |
| Professor | Professors |
| Semester | Semesters |
| Student | Students |
| User | Professors & Students |
| Visitor | Just used to replace null references |

1. **Dependency description**

**4.1 Intermodule Dependencies**

**Our project only has one package, so it is only dependent among each breakdown of classes which was explained above**

**4.2 Data Dependencies**



1. **Interface description**
2. **Detailed design**