**LMS Project Report**

**Members (5):** Jason Brunelle, Fabian Hernandez Nieto, Tom Phan, Tristen Stacks, Vi Ta

**Team Model**

Our team used a Democratic Team Model. We found this to be beneficial since it does not enforce any formal team hierarchy. A Democratic team structure is appropriate for less understood problems. It allows the group to invent better solutions rather than using a chief programmer team. This project didn’t require many engineers so our group of 5 worked well with this model. This model allowed us to encourage egoless programming and allowed our team to share and review each others work.

**Test Cases**

The test cases we conducted have passed most, if not all, of our requirements as of our last tests.

|  |  |  |
| --- | --- | --- |
| **Test** | **Expectation** | **Pass/Fail** |
| Login (as governor, headmaster, professor, and student) | Display appropriate dashboard content if successful login. If login is unsuccessful, message is shown. | **Pass** |
| Enroll Courses | Student can select which courses to enroll in and add them to registered courses. Will update info in database. | **Pass** |
| Drop Courses | Student can deselect enrolled courses and update the info to remove course from student’s registered courses. | **Pass** |
| Create Assignments | Professor can create assignments and assign to course(s). | **Pass** |
| Delete Assignments | Professor can delete assignments and remove associated grades if turned in already. | **Pass** |
| GPA calculation | GPA is calculated for each student with given grades and can be viewed by students. Cannot be edited by lv 1, 2 ,or 3 users. | **Pass** |
| Add/Disable Users | Governor can add and disable users (student, professor, or headmaster). New users can login and disabled users can no longer login. | **Pass** |
| Grade Assignments | Professor can grade assignments for students enrolled in their courses. | **Pass** |
| View Grades | Students can view assignment grades. Cannot modify grades. | **Pass** |
| View Assignments | Students can view assignments for enrolled classes. | **Pass** |
| Submit Assignments | Students can submit assignments for a given coursework. | **Pass** |
| Logout | User returns to login screen and is required to login again to use system. Login flag is off. | **Pass** |

Updated Test Cases

**Use Cases**

Use Case 1 (**Add/Disable Users**)

* Gilderoy Lockhart did not work out as a professor. Albus Dumbledore logs in and **disables** Gilderoy Lockhart’s account.
* Dumbledore **adds** a new teacher: Remus Lupin.
  + Username: Remus.Lupin
  + Password: RLPass
  + Role: Teacher
* System discovers what will be his unique id in the system.
* All IDs start with 900 followed by role (teacher is 3) followed by 2 digits which is one more than last entry.
* Dumbledore Enrolls Remus Lupin in all 7 years of Defense Against the Dark Arts so he can teach the class.
* Dumbledore Logs off.

Use Case 2 (**Students enroll in classes**)

* Harry Potter, Hermione Granger, and Ron Weasley all **enroll** in the following classes:
  + Charms Year 3
  + Defense Against the Dark Arts Year 3
  + Divination Year 3
  + Potions Year 3
  + Transfiguration Year 3
* Fred Weasley, George Weasley, and Angelina Johnson all **enroll** in the following classes:
  + Charms Year 5
  + Herbology Year 5
  + Divination Year 5
  + Potions Year 5
  + Transfiguration Year 5
* All users log out when they are done.

*Note: only classes listed are the ones that match their "year" entry in their enroll-in-courses table.*

Use Case 3 (**Create, Turn-in, and Grade Assignment(s) and View Grades**)

* Severus Snape **assigns** the following assignment then logs out to the 5th year potions class:
  + 1 roll of parchment on the properties of moonstones. Total point 10.
  + Angelina logs in and sees the assignment is due. She **turns it in**.
  + Fred Weasley logs in and sees the assignment is due. He **turns it in**.
  + George Weasley fails to turn in the assignment.
* Seversus Snape logs in and **records** their grades: Angelina: 6, Fred: 3, George: 0
* Angelina, Fred and George log and **view** their grades.
* All users log off.

Use Case 4 (**Assign Professor to Class and** **Delete Assignment (and associated grades)**)

* Snape subs for Lupin because he is sick. Dumbledore has to "enroll" Snape as a teacher of Defense Against the Dark Arts.
* Snape assigns the 3rd year defense against the dark arts class an essay on the qualities of the werewolf.
  + Hermione turns in the assignment.
  + Harry and Ron do not.
* Lupin recovers, logs in and **cancels** the assignment.
* Hermione, Ron, and Harry all log in to see the assignment is no longer there.

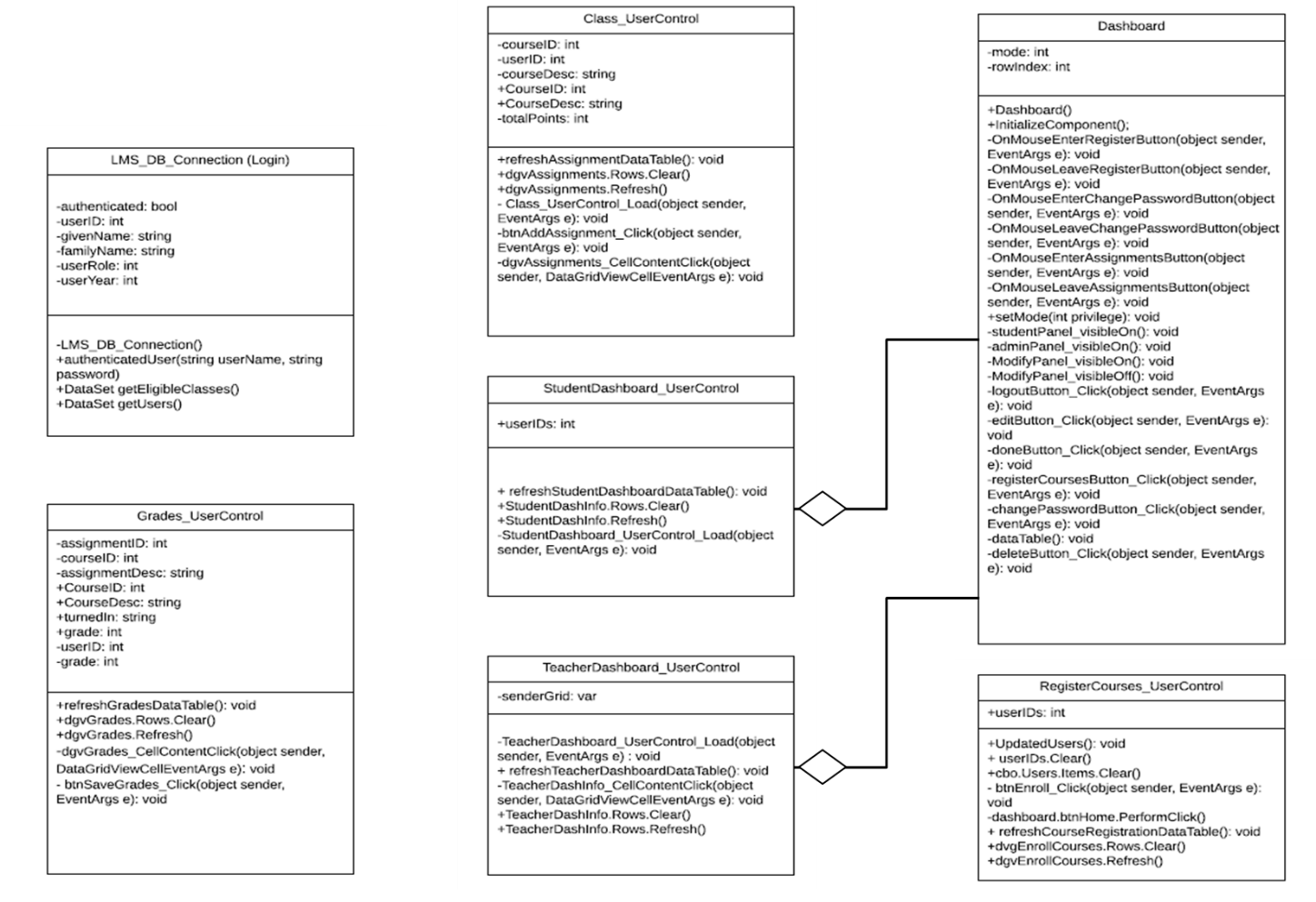
Use Case 5 (**Reassign Headmaster/Headmistress**)

* Delores Umbridge replaces Albus Dumbledore as Headmistress of the school
* Lucius Malfoy **adds** a new Headmistress: Delores Umbridge
  + Username: Delores.Umbridge
  + Password: DUPass
  + Role: Headmistrss
* Lucius Malfoy deactivates user Albus Dumbledore

Use Case 6 (**Deactivate User Account**)

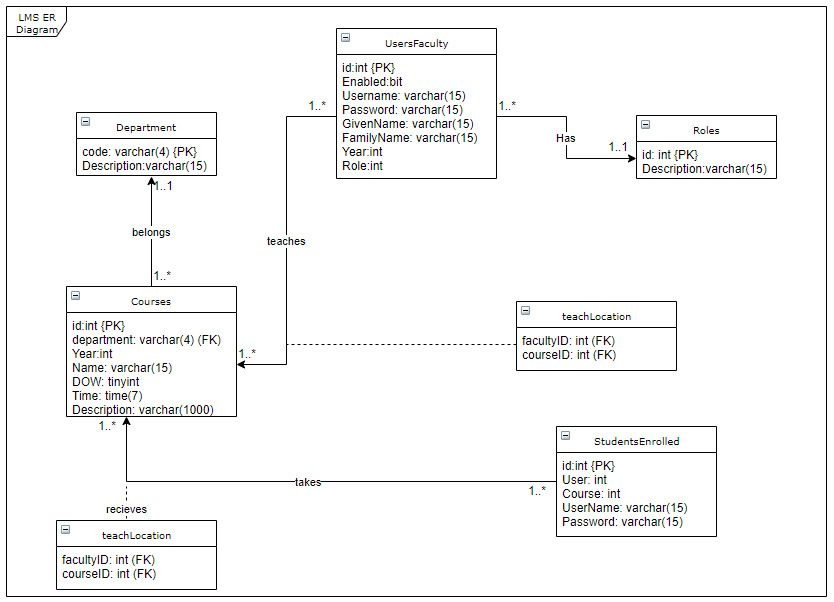
* Fred Weasley and George Weasley leave the school.
* Delores Umbridge **deactivates** their accounts.
* Fred and George can no longer log in.

**UML Diagrams**

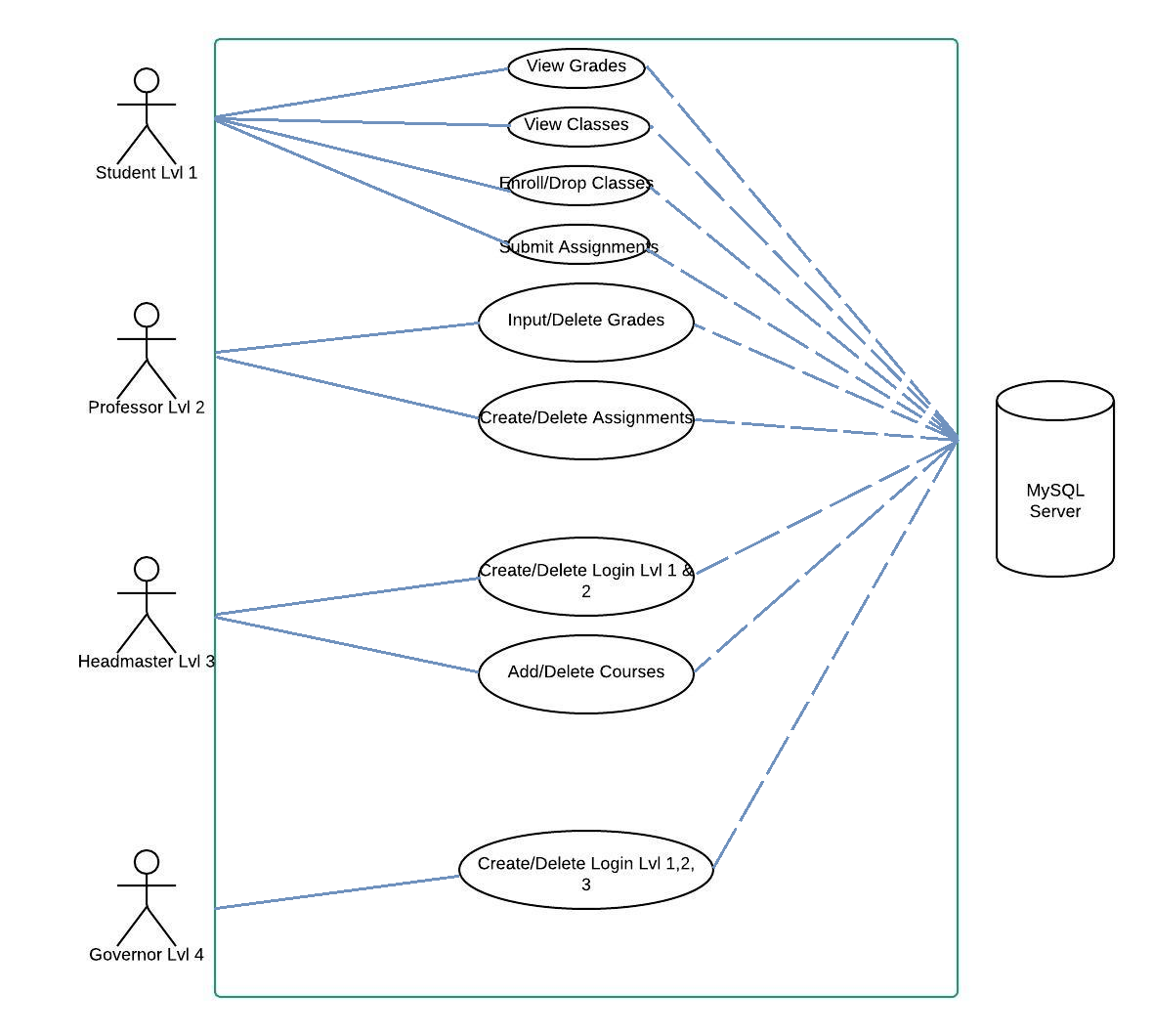


Class Diagram

…



ER Diagram

****

UML Diagram

**Artifacts**

|  |
| --- |
| **CLASS** |
| **User class** |
| **RESPONSIBILITY** |
| 1. Access Username 2. Access Password 3. Privilege (if student, professor, headmaster, or governor) |
| **COLLABORATION** |
| **Login class** |

|  |
| --- |
| **CLASS** |
| **Login class** |
| **RESPONSIBILITY** |
| 1. Verify input (w/i login class) 2. Check privilege (if student, professor, headmaster, or governor) 3. Load home page Dashboard class |
| **COLLABORATION** |
| **Dashboard class** |

|  |
| --- |
| **CLASS** |
| **Student class** |
| **RESPONSIBILITY** |
| 1. Display name 2. Display ID 3. Enroll/Drop courses 4. Exam scores for each course 5. Calculate GPA 6. View GPA 7. Turn in assignments 8. Log out (go to login panel) |
| **COLLABORATION** |
| **Dashboard class**  **Student class** |

|  |
| --- |
| **CLASS** |
| **Dashboard class** |
| **RESPONSIBILITY** |
| 1. Display name of user 2. Modify buttons **on** (if role < 4) send request to **student class** 3. Modify buttons **off** (if role == 4) 4. Display student info. (if headmaster) 5. Display assignments/courses (if role >= 2)    * Edit Assignments (If professor)    * Edit Courses (if headmaster) 6. Add/Remove Users (if governor) 7. Log out (go to login form) |
| **COLLABORATION** |
| **Login class**  **Student class** |

CRC Cards

**SPMP**

# **Software Project Management Plan**

**Introduction**

This document serves as the project plan for the Hogwarts Learning Management Plan. It will explain the functions the Learning Management System is capable of. We will provide an outline of our deliverables. We will also discuss the tools used to complete the project.

**Project overview**

The Hogwarts Learning Management System will establish an online system that provides four access levels for users. It will allow students, level 1, to view courses, view grades, enroll and drop form courses, and submit assignments. It will allow professors, level 2, to input and delete grades, and input and delete assignments. It will allow the headmaster, level 3, to create and delete logins for level 1 and level 2 users and allow them to add and delete courses. The governor, level 4, will be able to create and delete logins for user levels 1, 2, and 3.

**Project deliverables**

1. First Project Plan 2018.09.15

2. Analysis [ER Diagram, UML, CRC Cards, Class Diagram] 2018.10.28

3. Testing 2018.11.01 – 2018.11.25

4. Final Product and Demo 2018.11.26

5. Product Ready for Client 2018.12.01

**Evolution of this document**

This document will be updated as the project progresses. Updates should be expected in the following sections:

1. **Organization Structure -** this section will be revised appropriately as the requirements and artifacts are completed.
2. **Schedule –** as the project progresses, the project deliverables will be updated accordingly

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Updated By** | **Update Comments** |
| 2018.09.23 | Vi Ta | First Draft |
| 2018.10.07 | Vi Ta | UI Design |
| 2018.10.21 | Jason Brunelle | Database created and linked to project |
|  |  |  |
| 2018.10.22 | Jason Brunelle, Vi Ta | Project Overview updated, created 4 levels of users |
| 2018.10.28 | Vi Ta | CRC Cards added |
| 2018.10.30 | Fabian Hernandez | ER Diagram Created for database |
| 2018.11.01 | Tristen Stacks | Class Diagram and UML added |
| 2018.11.10 | Tom Phan | Testing |
| 2018.11.20 | All Members | Final Testing and Code Fixes |

**References**

Team Github <https://github.com/Tavi66/3321_LMS>

**Project organization**

**Process model**

We used a Democratic Team Model. We found this to be beneficial since it does not enforce any formal team hierarchy. It allowed the group to invent better solutions rather than using a chief programmer team. This project didn’t require many engineers so our group of 5 worked well with this model. This model allowed us to encourage egoless programming and allowed our team to share and review each other’s work.

**Organizational structure**

**Team Members –**

* 1. Jason Brunelle
  2. Vi Ta
  3. Fabian Hernandez
  4. Tristen Stacks
  5. Tom Phan

|  |  |  |
| --- | --- | --- |
| **Week/Deliverable** | **Team Member** | **Deliverable Description** |
| 1 | All Members | Project Plan |
| 2 | Tristen Stacks, Vi Ta, Fabian Hernandez, Tom Phan | Analysis |
| 3 | Jason Brunelle | Architecture Spec |
| 4 | Vi Ta, Jason Brunelle | Source Code |
| 5 | All Members | Testing |
| 6 | All Members | Final Deliverable |

## **Organizational Interfaces**

Every two weeks we coordinated team meetings to discuss updates, communications, and team deliverables.

**Technical process**

* 1. **Methods, tools, and techniques**  
      The project will be implemented utilizing Code and Fix. Tools to be used are Visual Studio for the IDE, coding in C#, mySQL for database storage, and GitHub for version control. Object-oriented programming will be used to successfully complete the project.