Christopher Blank, Nathan Frazier, Paul Nixon, and Catherine Todd

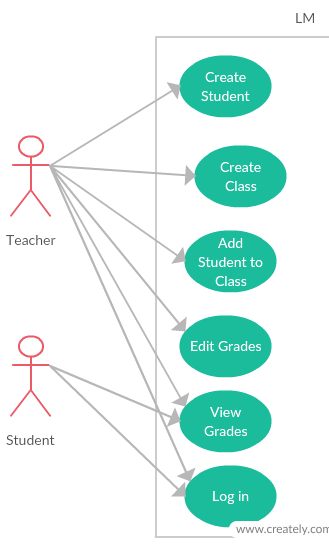
CS 3321

Chang

Due: 12/01/2018

**Generic Learning Platform Report**

1. **Team Model**
   1. We took a democratic approach to the project.
   2. Nathan Frazier- Lead Programmer
   3. Christopher Blank- Programmer
   4. Paul Nixon-Programmer
   5. Catherine Todd-Programmer/Manager
   6. We implemented a rapid prototyping as our lifecycle model
2. **Project Details**
   1. Programming Language: C#
   2. IDE: Visual Studio
   3. Data Storage: Text docs
   4. Version Control: GitHub
3. **Use Case Diagram**



1. **Use Cases**

Actor: Teacher

Use Case: Log in

Enter username

Enter password

Use Case: Add Grades

Use Case: Edit Grades

Use Case: Create Class

Specify Class Name

Specify Class ID

Use Case: Create Student

Enter Student Name, first and last

Enter Student ID

Assign Temporary Password

Use Case: Add Student to Class

Specify Class Name

Specify Class ID

Specify Student Name

Actor: Student

Use Case: Log in

Enter username

Enter password

Use Case: View Grades

1. **Data Dictionary**

User Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Caption | Data Type | Field Size | Notes |
| username |  | string |  |  |
| password |  | string |  |  |
| firstName |  | string |  |  |
| lastName |  | string |  |  |
| classes |  | string |  |  |
| isTeacher |  | bool |  |  |

Grade Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Item | Caption | Data Type | Field Size | Notes |
| grades |  | List<int> |  |  |

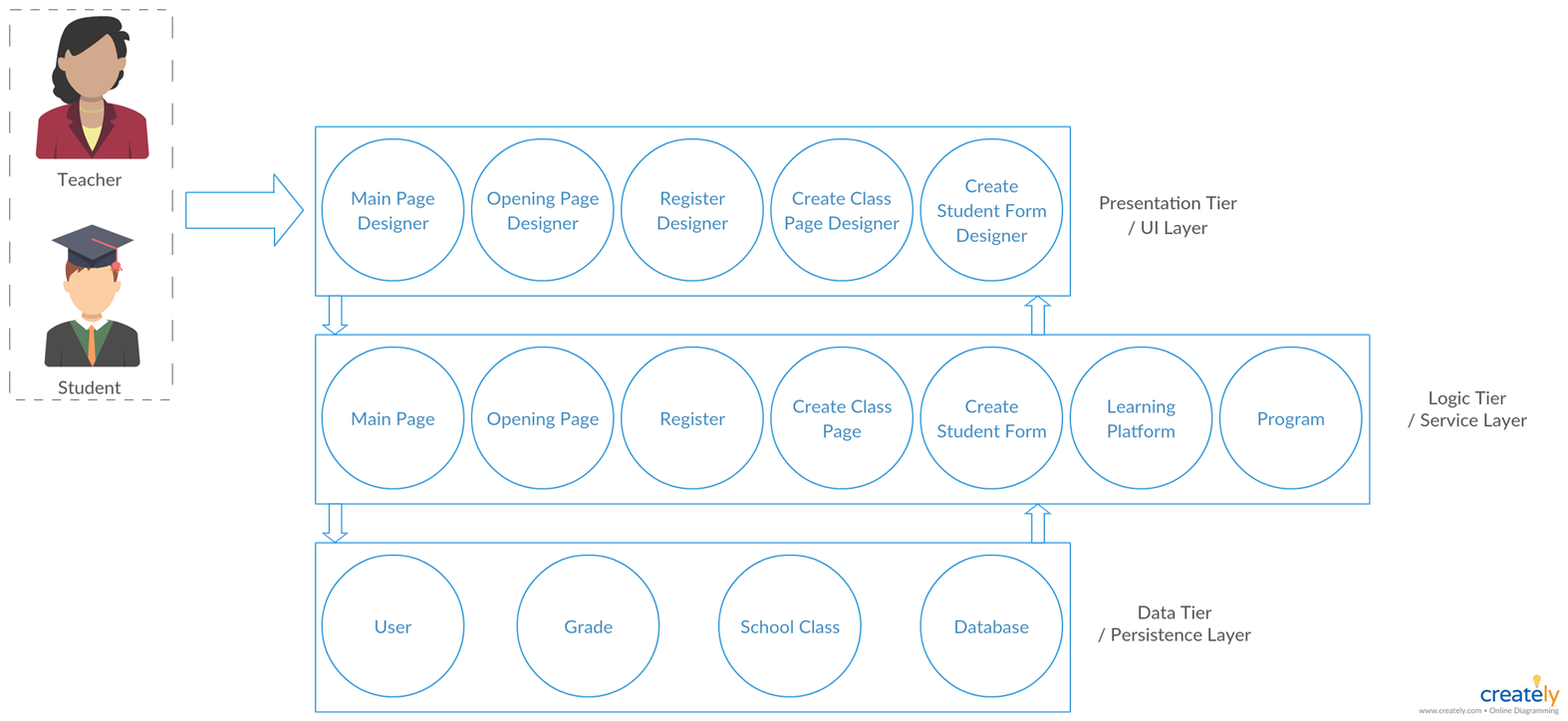
SchoolClass Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Item | Caption | Data Type | Field Size | Notes |
| className |  | string |  | Name of Class |
| classID |  | string |  | ID of Class |
| teacher |  | User |  | Name of Teacher |
| students |  | List<User> |  |  |
| grades |  | Dictionary<User, Grade> |  |  |

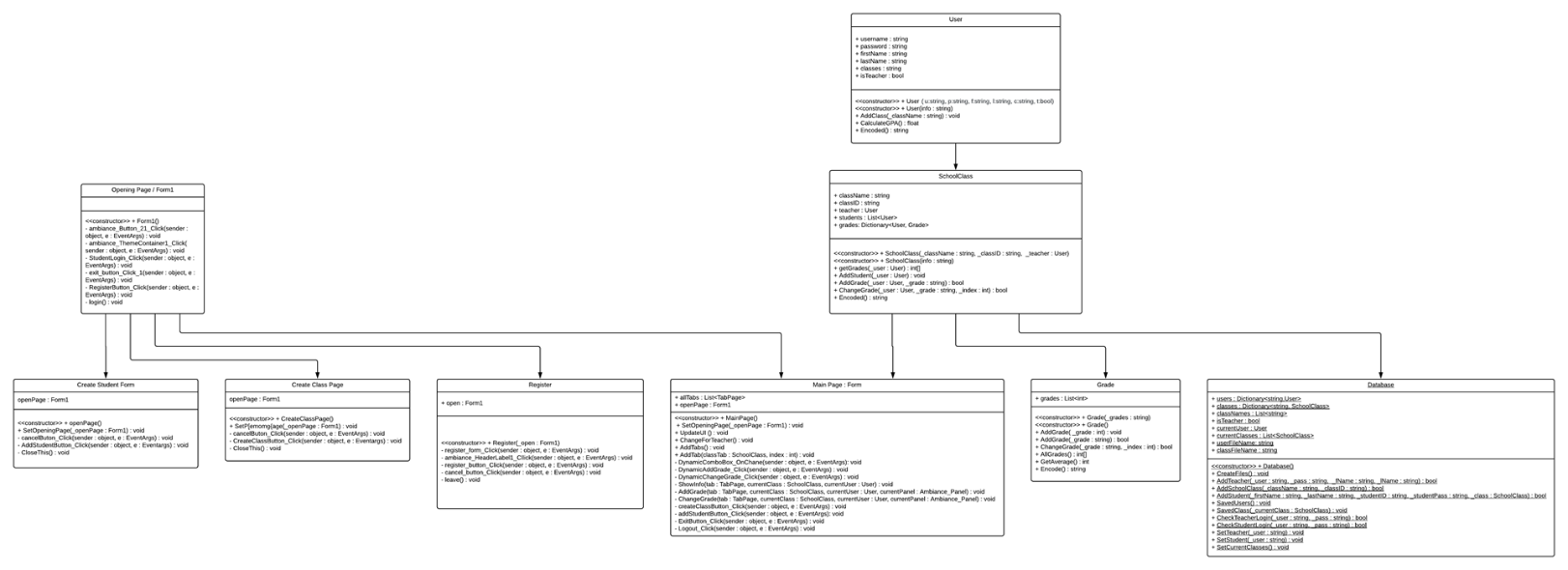
Database Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Item | Caption | Data Type | Field Size | Notes |
| isTeacher |  | bool |  |  |
| currentUser |  | User |  |  |
| currentClasses |  | List<SchoolClass> |  |  |
| users |  | Dictionary<string, User> |  |  |
| classes |  | Dictionary<string, SchoolClass> |  |  |
| classNames |  | List<string> |  |  |

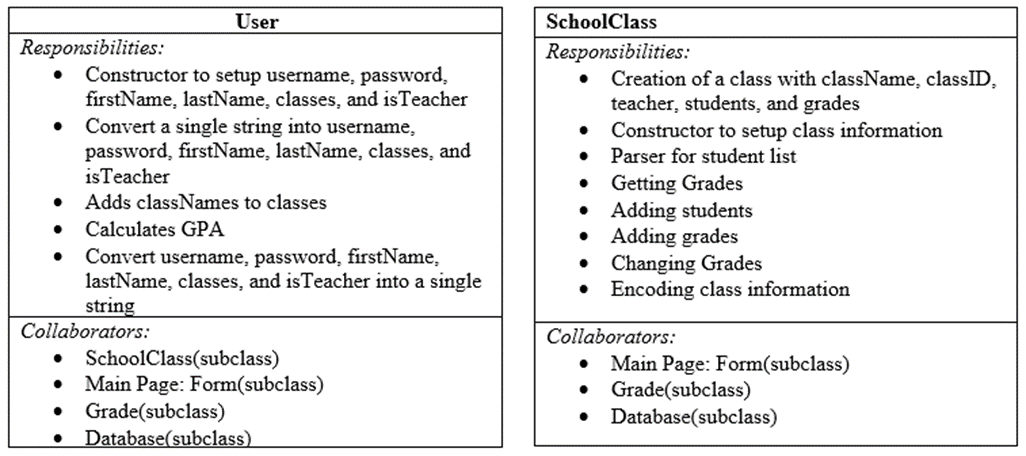
1. **Project Architecture**

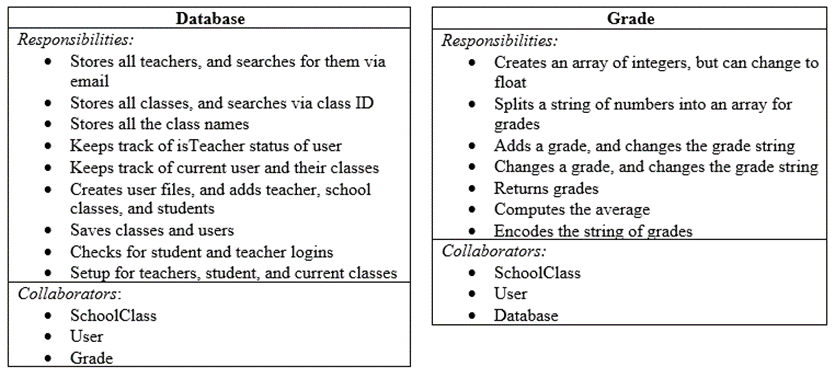


1. **UML**



1. **CRC Cards**





1. **Unit Tests**
2. Register as Teacher
   1. Registering New Teachers
      1. Registering a new teacher works fine
      2. Newly registered teacher is saved in database
3. Login as Teacher
   1. Logging in & out works fine
4. Create Class
   1. Creating a new class works fine
   2. Class name & ID are saved in database
   3. If I put a registered class name with a different ID, it’s still accepted
5. Add Student
   1. Adding a new student works fine
   2. Added students can be selected in Current Student
   3. A student can be added to another class just fine
   4. Grades for student can be seen
6. Editing Scores
   1. First test added starts with ‘0’ instead of ‘1’
   2. GPA changes if new grades are added and/or values are changed
7. Login as Student
   1. Logging in works fine
   2. Can only see grades, not change them
   3. Classes that the student is registered in and GPA are visible
8. **SPMP**

**1. Introduction**

1.1 Project Overview

Per the Rubric:

This project develops a learning management system (LMS) to help a university IT department their activities and improve their services, and for the management to track student’s basic information.

Generally speaking, Learning Management System deals with all kind of student details, academic related reports, college details, course details, curriculum, batch details and other resource related details too. It tracks all the details of a student from the day one to the end of his course which can be used for all reporting purpose, tracking of attendance, progress in the course, completed semesters years, coming semester year curriculum details, exam details, project or any other assignment details, final exam result etc. **This project doesn’t need to cover all the features and functions as a Learning Management System.**

* 1. Requirements

Per the Rubric:

The software system stores and retrieves students’ partial information in the current semester and other basic information including **student’s name**, **student’s ID**, **registered courses in the current semester**, **each exam’s score in one course**, **GPA calculation in the current semester**. Use the strategies studied in the lectures to accomplish the requirement artifacts.

The goal system has two types of accessing modes, administrator and user. Student information management system is managed by an administrator. It is the job of the administrator to insert update and monitor the whole process. When a user log in to the system. He/she would only view details of the student. He/she can't perform any changes.

1.2 Project Deliverables

Per the Rubric:

Answers the following questions:

How many members are you in your team? List all team members.

What type of team model is used in the project?

UML diagrams you used in the project. Some important diagram are expected to be included like architecture diagram, use case diagram, and class diagram.

All artifacts you used in the project. These artifacts include source code, UML diagrams, SPMP, version control documentation, test cases, data storage files, and other necessary artifacts studied in the class.

1.3 Definitions and Acronyms

LM: Learning Module

**2. Project Organization**

2.1 Process Model

Rapid Prototyping.

2.2 Organizational Structure

Democratic power structure influenced by the Chief Engineer model.

2.3 Organizational Boundaries and Interfaces

GitHub

2.4 Project Responsibilities

Nathan Frazier – Lead Programmer

Paul Nixon –Programmer

Catherine Todd – Programmer and Manager

Christopher Blank – Secretary / Librarian / Documentarian

**3. Managerial Process**

3.1 Management Objectives and Priorities

Complete the assignment as assigned in the rubric.

Make the finished product aesthetically pleasing.

3.2 Assumptions, Dependencies, and Constraints

No one will ever use this product, so it does not need to be scalable or perfectly reliable.

The program reads data from a text file, and is not configured to work with a database.

3.3 Monitoring and Controlling Mechanisms

The team communicates through GroupMe, GitHub, and personal meetings.

We share information at our own personal discression.

3.4 Staffing Plan

We formed a group with classmates seated

**4. Technical Process**

4.1 Methods, Tools, and Techniques

GitHub and Visual Studios 2015.

Programming conducted in C#.

4.2 Software Documentation

Handled on GitHub.

**5. Work Packages, Schedule, and Budget**

5.1 Work Packages

NA

5.2 Dependencies

The program has no external dependencies other than the text file used to store its information.

5.3 Resource Requirements

The program should run without issue on any personal computer produced in the last decade.

5.4 Budget and Resource Allocation

NA

5.5 Schedule

We operate via rough guidelines, trusting that all members will complete their chosen sections in a reasonably timely manner.