**Outline**

Contents

**1. Introduction**

1.1 Project Overview

1.2 Project Deliverables

1.3 Reference Materials

**2. Project Organization**

2.1 Process Model

2.2 Organizational Structure

2.3 Organizational Boundaries and Interfaces

**3. Managerial Process**

3.1 Management Objectives and Priorities

3.2 Assumptions, Dependencies, and Constraints

3.3 Risk Management

**4. Technical Process**

4.1 Methods, Tools, and Techniques

4.2 Software Documentation

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

5.1 Work Packages

## **1.0 Introduction**

### **1.1 Project Overview**

The client’s requirement is to develop a learning management system (LMS) to help a university IT department with their activities and improve their services, and for the management to track student’s basic information.

The product is supposed to simulate a Typical LMS, which includes Blackboard or Moodle. Our project is done by a team of 5 undergraduate students during an academic semester, in conjunction with lectures and other class activities. It is similar to Blackboard or Moodle LMS system.

### **1.2 Project Deliverables**

* Design Graphical User Interface
* Link the GUI to a database that holds all the information regarding academic records.
* Add use cases for both students and teachers with each to have a specific user view.
* The student would be able to enroll in classes, and view their grades, and view their GPA according to their grades.
* Teachers are referred to as Admin in our software, and they’re able to insert, update, and monitor the whole process.
* The students can log in via a username and password, and they can only view their information and not able to perform changes.
* Democratic Team

### 

### **1.3 Reference Materials**

The overall Document Structure was taken from the instructions sent by the professor.

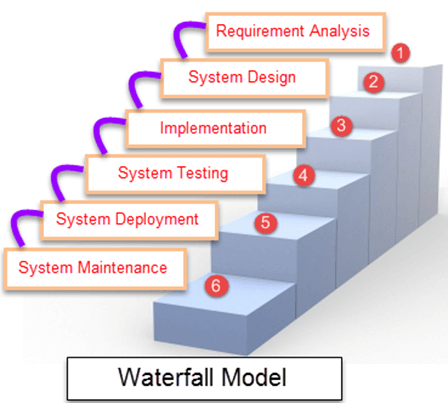
Sites Used:

[https://trello.com](https://trello.com/b/RgdjNxSC/software-engineering-bb-project)

[https://github.com](https://github.com/Daniyal2493/BlackBoard_Application)

## **2.0 Project Organization**

### **2.1 Process Model**



|  |  |
| --- | --- |
| **Different phases** | **Activities performed in each stage** |
| **Requirement Gathering stage** | * We gathered the requirements from the project description posted on bb.uhd.edu for Software Engineering Class. |
| **Design Stage** | * We decided to use two programming languages to create the software, then decide which application is more efficient: * First choice was using Visual Basics programming language in parallel with Microsoft Access as a database. * Second choice was using C# in .NET framework, in parallel with MySQL. |
| **Built Stage** | * At this Stage, we started coding the program, by dividing the work among team members. |
| **Test Stage** | * In this phase, you test the software to verify that it is built as per the specifications given by the Professor Chang. |
| **Deployment stage** | * Deploy the project in the built environment, by presenting it to class on Monday 11/25/2019. |
| **Maintenance stage** | * At this point, the software is complete, and we regulate maintenance period or apply changes per client’s request. |

### 

### **2.2 Organizational Structure**

Version Control: Register account on<https://github.com/> to study version control techniques for your team work on the project. A list of basic git commands is attached with this project assignment instruction.

We used trello to notify each member on what was due and when it was due.

### **2.3 Organizational Boundaries and Interfaces**

* **CRC Cards**

|  |
| --- |
| **CLASS**  Student Info |
| **RESPONSIBILITY**   1. Displays student name and ID 2. Displays GPA 3. Displays midterm and final grades |
| **COLLABORATION**   1. StudentCourse 2. Courses class |

|  |
| --- |
| **CLASS**  Student Grades |
| **RESPONSIBILITY**   1. Display the course name and degree category 2. Display professor of course 3. Display the building name and room number 4. Display the days of the week to attend the class 5. Display session time |
| **COLLABORATION**   1. Student Info class 2. Student course grade class |

|  |
| --- |
| **CLASS**  Student Management Window |
| **RESPONSIBILITY**   1. Displays student ID and class ID 2. Displays assignments and numeric grade received for that assignment 3. Displays the grade weight 4. Displays cumulative grade for the course |
| **COLLABORATION**   1. Student management class |

|  |
| --- |
| **CLASS**  Admin |
| **RESPONSIBILITY**   1. Administrators can view student ID and course ID 2. Administrators can modify the assignment & grade 3. Administrators can calculate cumulative grade for the course |
| **COLLABORATION**   1. Admin courses |

|  |
| --- |
| **CLASS**  Admin Course View |
| **RESPONSIBILITY**   1. Administrators can modify course name 2. Administrators can modify student ID and student grades |
| **COLLABORATION**   1. Modify course name |

## 

## **3.0 Managerial Process**

### **3.1 Management Objectives and Priorities**

* Relative priorities among functionality, schedule, budget[student hours worked], and quality.
* risk management procedures
* approach to acquiring third party software
* approach to modifying or using existing software

### **3.2 Assumptions, Dependencies, and Constraints**

User should be able to do all the following:

* **adding, deleting, inserting, and modifying a record (record could be a course and a student profile),**
* **logging in/out system,**
* **browsing all records,**
* **checking a student or all students’ GPA**

### **3.3 Risk Management**

Drew out an outline of what to be expected, such as a USE CASE scenario:

Use case name: **View student records/info**

Actor/User: Admin, Student

Steps:

1. User clicks on “student login” or “admin login”
2. User enter’s “student Username” and “Password” or if admin: User enter’s “admin Username” and “Password”
3. User clicks “login button”

Once logged on:

1. User views “grades”
2. User views “Courses”
3. User views “GPA”
4. User views “name & ID”

User can click exit to logout.

Use case name: **Entering a new Student**

Actor/User: Admin

Steps:

1. User clicks on “View Student” button.
2. System displays an entry screen with new user id generated and prompts the user to enter first name, last name, Student ID, Courses, Grades and calculate the GPA.
3. User clicks on “ADD Student” button.
4. System creates new student record in the database.

Use case name: **Updating an existing student**

Actor/User: admin

Steps:

1. User clicks on “Student number” button.
2. System displays the students information screen and prompts the user to input the updated information
3. Once updated the system allows the User to save by clicking “Save Student Data” button
4. User has the option to update other students as well by their enrollment number.
5. When finished user can use “exit” to logout.

Use case name: **Deleting an existing student**

Actor/User: admin

Steps:

1. User enter on the “enrollment #” to choose the targeted student .
2. User selects “ Delete Student Data” to delete the student.
3. System asks User if he is sure, User can select “yes”
4. Student is deleted from the database.
5. When finished user can use “exit” to logout.

## 

## 

## **4.0 Technical Process**

### **4.1 Methods, Tools, and Techniques**

* Discord
* GroupMe
* Trello
* SQL Server Management Studio
* LucidChart
* GitHub
* Visual Studio
* .NET Framework C#
* Google Documents
* Gmail
* Google.com
* MS Office Suite
* Teamviewer

### 

### 

### 

### 

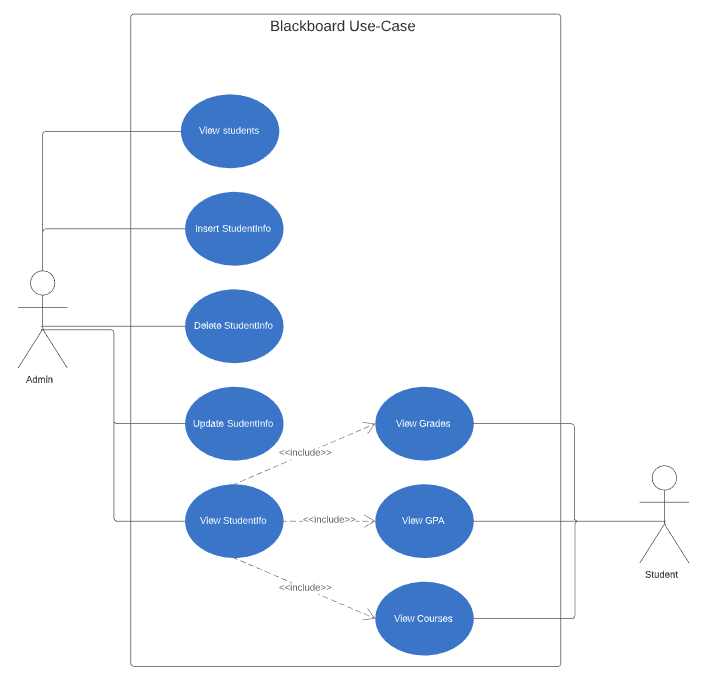
### 

### 

### 

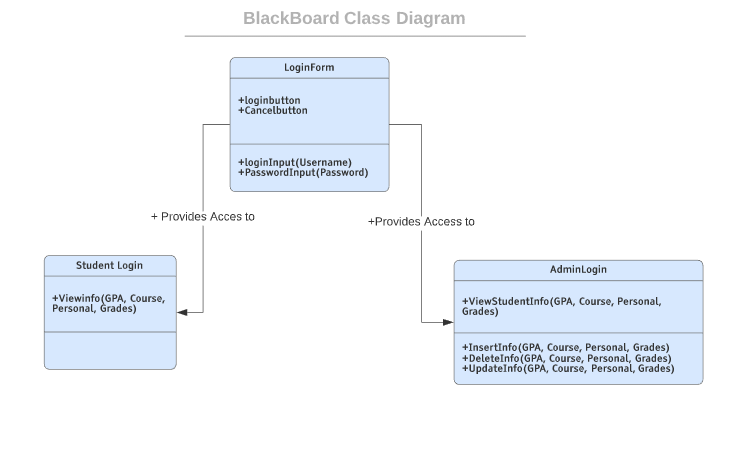
### **4.2 Software Documentation**

**Use Case Diagram:**



### 

**Class Diagram:**



## 

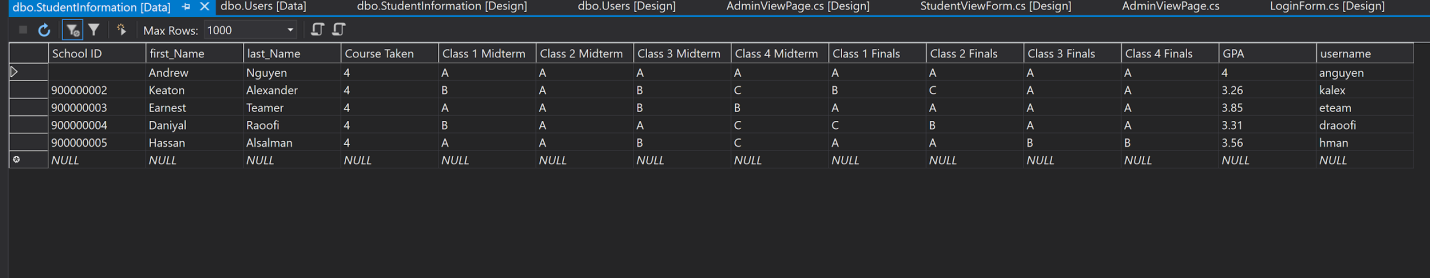
## 

## 

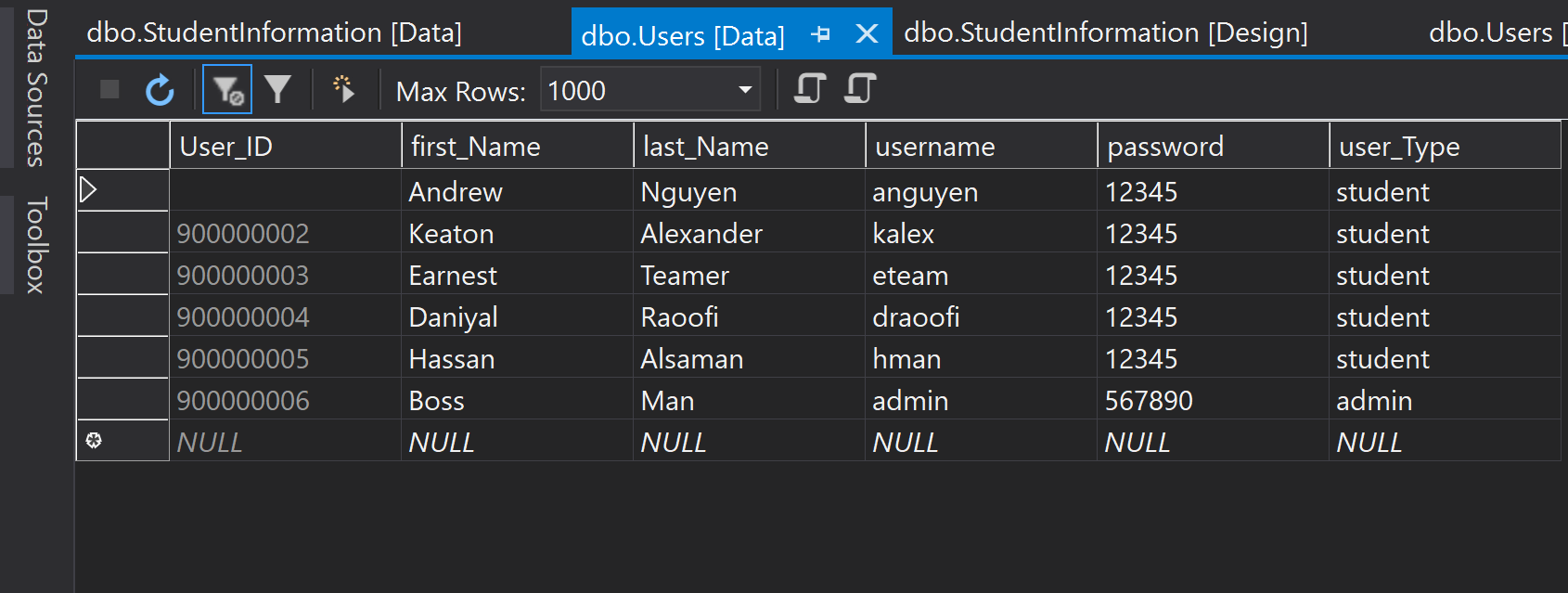
## **5.0 Work Packages, Schedule, and Budget**

### **5.1 Work Packages**

StudentInformation



User



We used SQL Server to hold all the necessary data for the project which can be edited by all members from visual studios