|  |  |
| --- | --- |
|  | Design Document |
|  |  |
|  | Student ID  [Course title]  [Date] |

Contents

[Introduction: 2](#_Toc45664001)

[Requirements Analysis: 2](#_Toc45664002)

[System Requirements: 2](#_Toc45664003)

[Functional Requirements: 2](#_Toc45664004)

[Non- Functional Requirements: 4](#_Toc45664005)

[GUI Design: 4](#_Toc45664006)

[1. User Login: 5](#_Toc45664007)

[2. Main Menu: 5](#_Toc45664008)

[3. Manage Classes: 6](#_Toc45664009)

[4. Homework Setup: 6](#_Toc45664010)

[5. Homework marking: 7](#_Toc45664011)

[Testing: 7](#_Toc45664012)

[Methodology: 9](#_Toc45664013)

# Introduction:

the project is about the development of a school management system using dog data and cloud technologies. We have designed and developed system for teachers’ section. A teacher can manage classes, homework and can communicate with parents or administration using our system. We have developed a desktop application for selected user type. As we have designed a desktop application, so it can only be accessed by the specific group of people. In our case only those teachers who are part of school can use our application.

# Requirements Analysis:

## System Requirements:

The hardware and software requirements for the smooth running of our system are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Software | Visual Studio 2015 | Hardware | RAM 8 GB |
| SQL Server | CPU core i5 |
|  | HD up to 2 GB |
|  |

## Functional Requirements:

Using our system, the user can perform following functions:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function Id | Name | Description | Pre-condition | Post Condition |
| FR- TE- 01 | Set up Homework | The user can set up homework online and offline. | User must be log in into the system. | The homework will be created successfully and will be visible to the students. |
| FR- TE- 02 | Mark Homework | The user can mark the assignment of an individual student. | User must be log in into the system. | The marks of the students will be updated. |
| FR- TE- 03 | Manage Classes | The user can add or delete a student also he can see the list of the students | User must be log in into the system. | The user will be able to see list of students and will add or delete a student successfully. |
| FR- TE- 04 | Assess performance | The user can assess the performance of a student and can share it with parents or administration. | User must be log in into the system. | The record will be shared successfully. |

## Non- Functional Requirements:

* Access will be provided to the authorized users only.
* Multiple users can use the website at a time.
* System ensures privacy of one’s personal data.

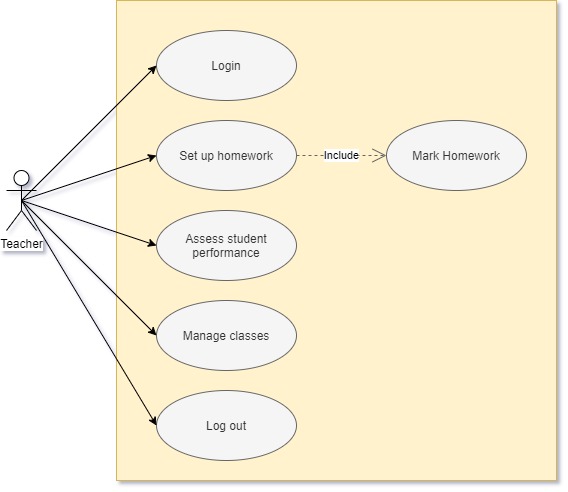


Figure Use Case Diagram

# GUI Design:

As pour system will be run on Desktop so we have designed a windows form application using C sharp language. the application is able to perform all the actions along with an interactive interface that is easy to use and understand.

## User Login:

As the administrated has already address the data pf all the teachers who can use their system so for Login the user will only provide his username and password, the one stored in the database.

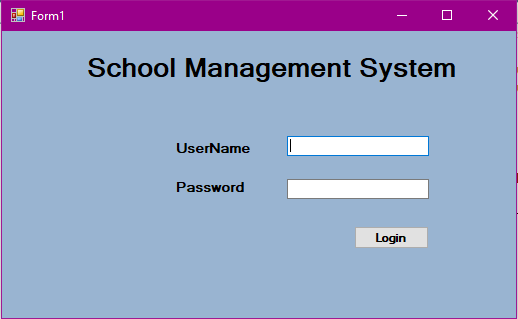


Figure User Login

## Main Menu:

After successful login the main Manu will show the list of functions that a user can perform.

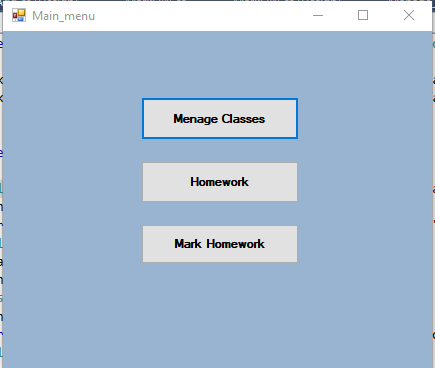


Figure main Menu

## Manage Classes:

This will allow user to view list pf all the students, add or delete a student.

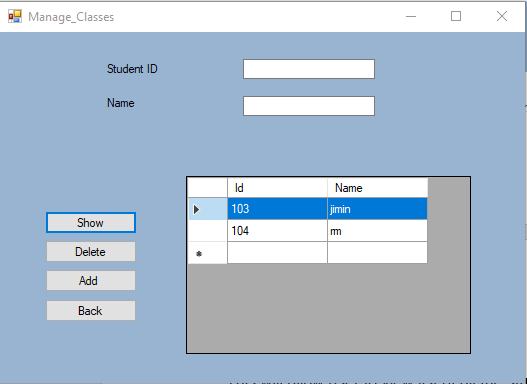


Figure Manage Students

## Homework Setup:

The user will be able to set up a new homework for the students, also he can add a detail description along with the HW starting and ending date.

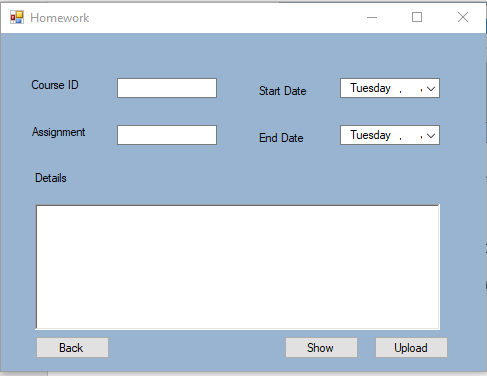


Figure Home work set up

## Homework marking:

The user will be able to mark homework of all the students who have submitted their assignments. He can also update the scores.

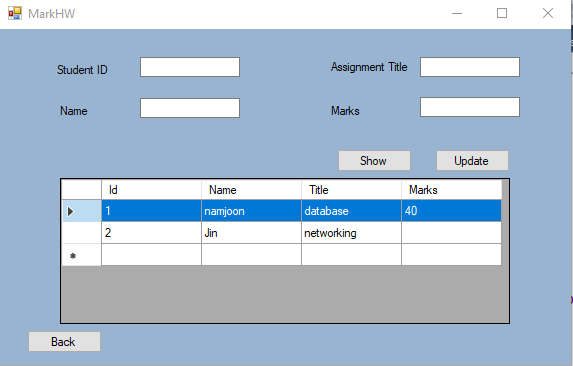


Figure Mark Homework

# Testing:

For integration testing, black box testing technique is used. as in black box texting a tester is not provided access to the code, he can test the system blindly for all the functional requirements.

For the testing of the system following test cases has been designed and system is tested accordingly:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case title** | | Login | |
| **Test case description** | | Teacher will login to the system through its id and password from a desktop computer | |
| **Execution Steps:** | | | |
| **S.no** | **Action** | | **Expected output** |
| 1 | Launch desktop application | | A login page is displayed |
| 2 | Enter incorrect email and password and login | | The email address you entered is incorrect |
| 3 | Enter correct email and incorrect password for login | | The password you entered is incorrect |
| 4 | Enter correct email and password | | Login successful |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case title** | | Manage homework | |
| **Test case description** | | Allocate homework to students whether online or offline and also mark them | |
| **Execution Steps:** | | | |
| **S.no** | **Action** | | **Expected output** |
| 1 | Launch application | | Homepage is displayed |
| 2 | Go to manage homework tab and click on add new homework button | | A new page will open for selecting homework file and setting status (online/ offline) |
| 3 | Go to manage homework tab and click on any homework link to update marks | | Student marks will be updated. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case title** | | Assess student performance | |
| **Test case description** | | Mark the performance of the students depending upon homework or other designated tasks | |
| **Execution Steps:** | | | |
| **S.no** | **Action** | | **Expected output** |
| 1 | Go to student’s tab | | A list of students enrolled in your course will be displayed |
| 2 | Select any student | | Previous record of student will be displayed |
| 3 | Click on update record and make changes | | Will allow to make changes in student profile and will save them |
|  | | | |
| **Test case title** | | Manage classes | |
| **Test case description** | | Manage classes as well as adding or removing a respective student from the class | |
| **Execution Steps:** | | | |
| **S.no** | **Action** | | **Expected output** |
| 1 | Go to manage class tab in your application | | A list of previous and upcoming classes as well as student enrolled will be displayed |
| 2 | Click on add a student and provide correct student id | | The student has been enrolled in your class |
| 3 | Click on add a student and provide incorrect student id | | Invalid student id |
| 4 | Select any enrolled student and click on remove | | Student successfully removed from your course |

# Methodology:

The SDLC methodology that we have adapted for the design and development of our project is water fall. As all the project functional requirements were already defined and we just need to focus on the design and development part so water fall model was the best model to use.

All the requirements of the system were analyzed properly, after requirement gathering and analysis system interface was designed. The all the functional requirements were coded in C sharp language and tested accordingly.

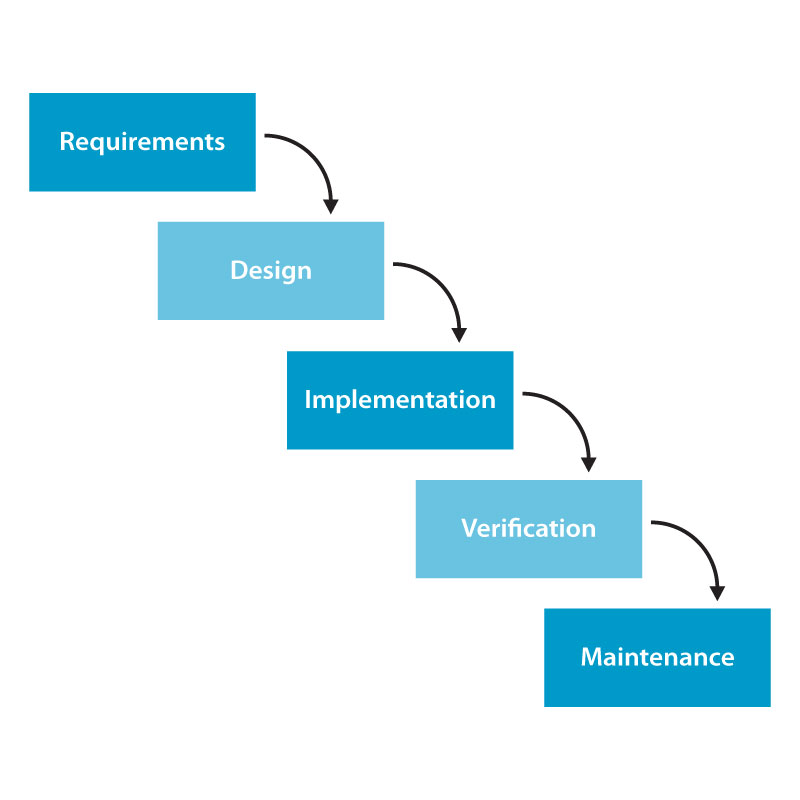


Figure Water Fall model Life cycle