**Student - Mentor Guidance System (SMGS)**

**Software Requirements Specification**

|  |  |  |
| --- | --- | --- |
| **Date** | **Description** | **Author** |
| 16/09/2021 | First Draft to document functional requirements for SMGS | Chetan Badgujar  Bhushan Chopade |

**Table of Contents**

1.Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

2. Overall Description

2.1 Application Perspective

2.2 Application Functions

2.3 Assumptions and dependencies

2.4 Apportioning of requirements

2.5 Technology Stack

3. Specific Requirements

3.1 Functionality

3.1.1 Admin functionality

3.1.2 Mentor functionality

3.1.3 Student functionality

3.1.4 Use-case diagram

3.1.4.1 System Use-case diagram

3.1.4.2 Admin Use-case diagram

3.1.4.3 Student Use-case diagram

3.1.4.4 Mentor Use-case diagram

3.1.5 Activity diagram

3.1.5.1 Admin Model

3.1.5.2 Student Model

3.1.5.3 Mentor Model

3.1.6 DFD

3.1.7.1 Level 1 - DFD - Student Model

3.1.7.2 Level 1 - DFD - Mentor Model

3.1.7.3 Level 1 - DFD - Admin Model

3.2 Usability

         3.3 Reliability

3.4 Performance

3.5 Supportability

**Software Requirements Specification**

**1. Introduction**

This section gives a scope description and overview of everything included in this SRS document. Also,

the purpose of this document is described and a list of abbreviations and definitions is provided.

**1.1 Purpose**

The purpose of this document is to give a detailed description of the requirements for the

“Student - Mentor Guidance System” software. This software is intended to provide additional functionality of assigning students to the mentor. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interface, and interactions with other external applications

**1.2 Scope**

The “Student - Mentor Guidance System” is a web-based application that helps people to gather and analyze data related to students and mentors and efficiently assigns students to the mentors depending upon the total number of students and available mentors. The application provides functionality to maintain relationships between the two.

Users can provide their personal details as well as course enrolled information using web portal. This

information will act as the basis for the assignment process.

All system information is maintained in a database. The application interacts with MySQL database and performs insertion, update as well as deletion as directed by the user.

**1.3 Definitions, Acronyms and Abbreviations**

SMGS: Student - Mentor Guidance System.

User: Someone who interacts with web-portals.

Admin: Administrator- the person who maintains the entire web application

Student: Person who enrols for the course.

Mentor: Person who guides the student.

I/U/D: Insertion/Update/Deletion

DFD: Data Flow Diagram

ER: Entity-Relationship Diagram

Web Portal: A web application UI which presents special facilities for interacting with users.

**1.4 References**

1. StackOverflow: <https://stackoverflow.com/>

2. Java Docs: <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>

3. Spring Boot Docs: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

4. ReactJS Docs: <https://reactjs.org/docs/getting-started.html>

**1.5 Overview**

The remainder of this document includes three chapters. The second one provides an overview of the

system functionality and system interaction with other systems. This chapter also introduces user’s

interaction with the system. Further, the chapter also mentions the system constraints and assumptions

about the system. The third chapter provides the requirements specification in detailed terms and a

description of the different system interfaces.

**2. Overall Description**

The system mainly consists of SMGS Online system containing application server and web server. SMGS interacts with the database to get student and mentor information. Secondary information like courses and queries are stored in the database.

**2.1 Application Perspective**

The system will consist of a web portal. The web portal will be used to maintain reports regarding the performance of mentor and student. This software is to be used as a grading tool to analyze performance of students.

**2.2 Application Functions**

SMGS mainly supports following high level functions:

2.1.1 Mentors will get assigned to the students depending upon the number of students.

2.1.2 Mentors with more rating will be given preference while allotment process,

2.1.3 Students can request another mentor and the admin will handle the request.

2.1.4 Students will be able to give rating and feedback to their respective mentors.

2.1.5 Mentors will be able to evaluate students and update their marks.

2.1.6 Students will be able to generate certificates depending upon their performance.

**2.5 Technology Stack**

2.5.1 Backend:

|  |  |
| --- | --- |
| **Category** | **Technology Name** |
| Framework | Spring Boot |
| Database | MySQL |
| Language | Java |

2.5.2 Frontend:

|  |  |
| --- | --- |
| **Category** | **Technology Name** |
| Library | ReactJS |
| Language | HTML, CSS, JavaScript |

**3. Specific Requirements**

This section contains all of the functional and quality requirements of the system. It gives a detailed

description of the system and all its features.

**3.1 Functionality**

The system will consist of a web portal. The web portal will be used to maintain reports regarding the performance of mentor and student. This software is to be used as a grading tool to analyze. performance of students.

**3.1.1 Admin Functionality:**

3.1.1.1The super user, admin class represents complete authority over the system

3.1.1.2 Registers both mentor and student

3.1.1.3 View the list of students and mentors who have successfully registered in the system.

3.1.1.4 Deletion of the accounts of mentor and student.

3.1.1.5 View the progress of the course which has been selected by students. The progress consists of tasks performed for each milestone by students related to the course.

3.1.1.6 Manually assign mentors to the students.

3.1.1.7 Login and logout for each session.

**3.1.2 Mentor Functionality:**

3.1.2.1 Guide class represents a user who is responsible for guiding the students through their course phase.

3.1.2.2 A Guide is registered by admin. Upon receiving the login credentials into the system, a guide can perform various functionalities. These include,

3.1.2.2.1. View the list of students who have successfully registered under them.

3.1.2.2.2. Update the number of students which can be handled (Size of batch).

3.1.2.2.3. Update the marks of the student after evaluation.

3.1.2.2.4. Login and logout for each session.

3.1.2.2.5. Signup for the registration.

3.1.2.2.6. Update the personal information registered during signup.

**3.1.3 Student Functionality:**

3.1.3.1Login and logout for each session.

3.1.3.2 Signup for the registration.

3.1.3.3 Update the personal information registered during signup.

3.1.3.4 Check assigned mentor.

3.1.3.5 Give feedback to the mentor.

3.1.3.6. Access the obtained marks from the mentor.

3.1.3.7 Generate Certificate.

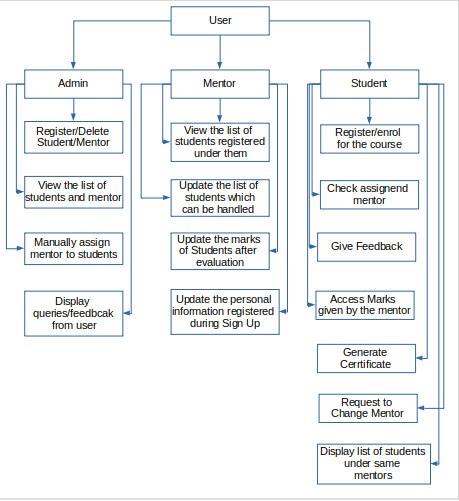
3.1.3.8 Request to change the mentor.

3.1.3.9 List of other students mapped under the same batch/mentor.

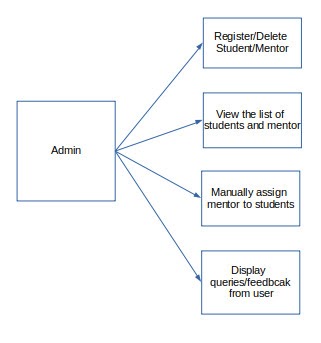
**3.1.4 Use-Case diagram:**

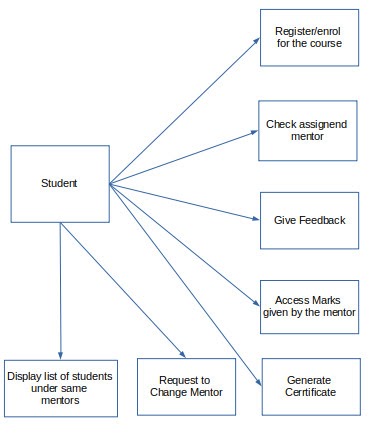
3.1.4.1 System:

Following diagram describes the entire flow of the system:

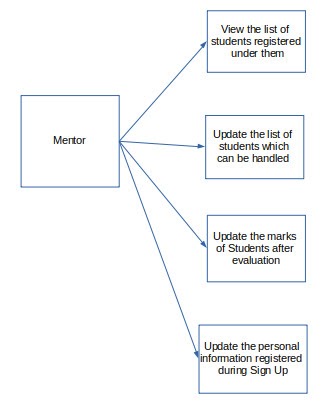
****

3.1.4.2 Admin Model:

3.1.4.3 Student Model: ****



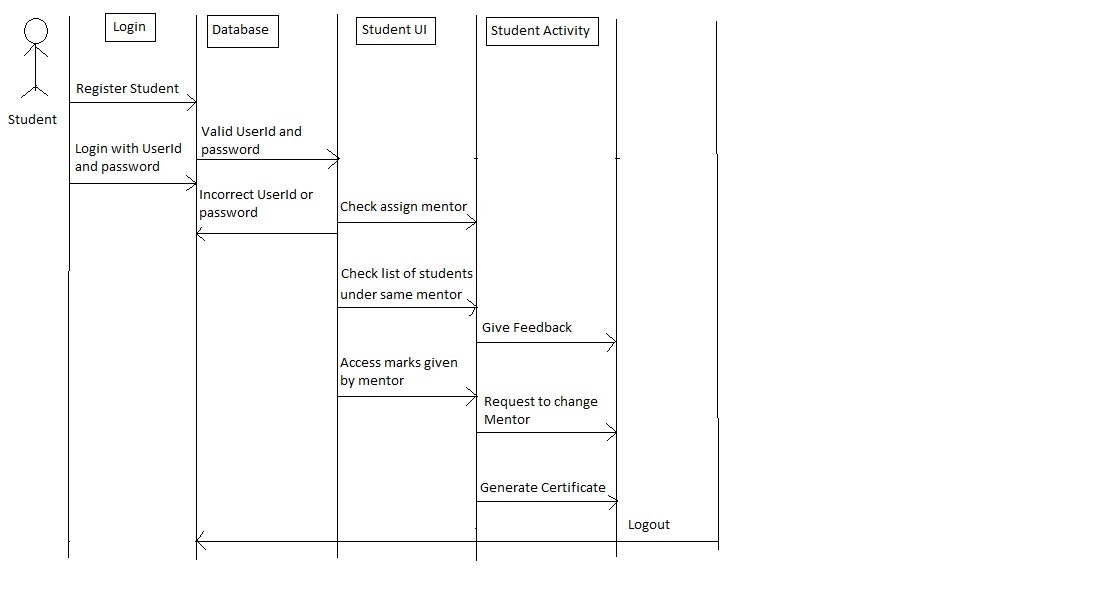
3.1.4.4 Mentor Model:



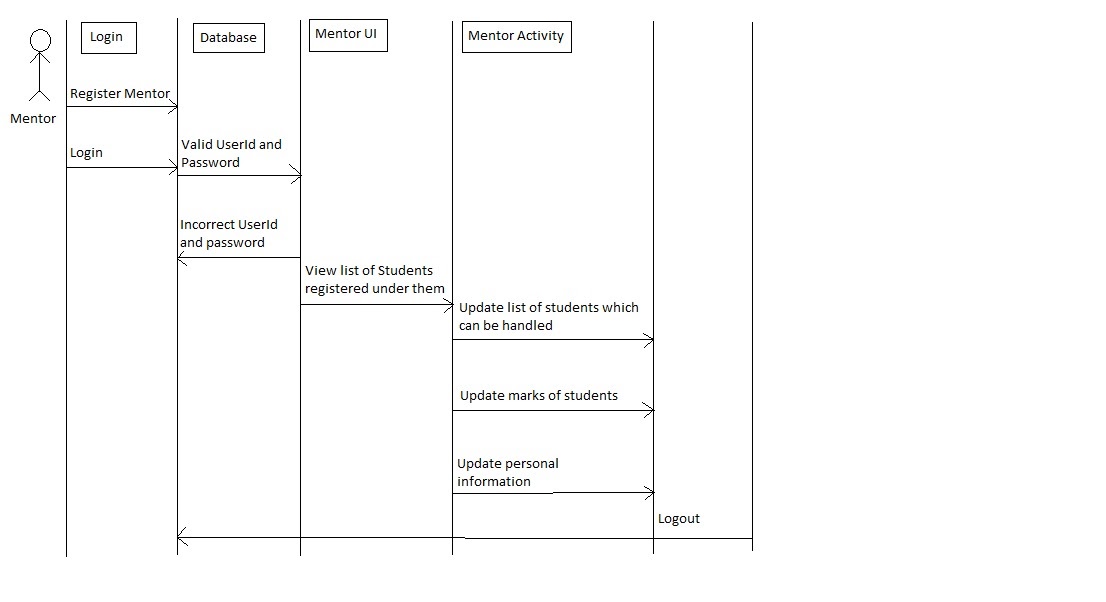
**3.1.6 Sequence Diagram:**

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place.

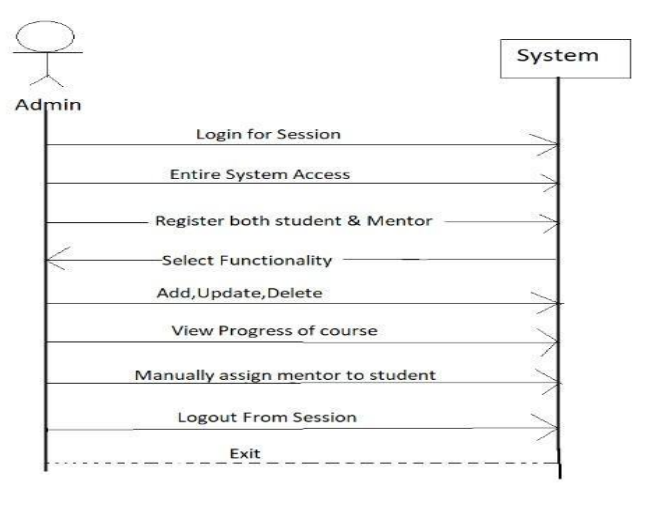
**Student Model:**



**Mentor Model :**



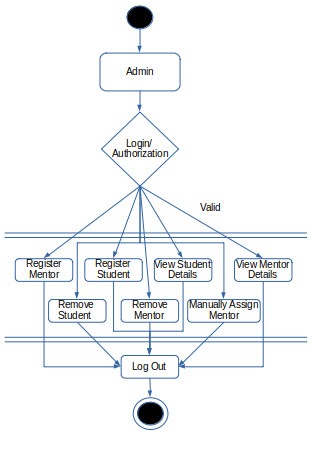
**Admin Model:**

****

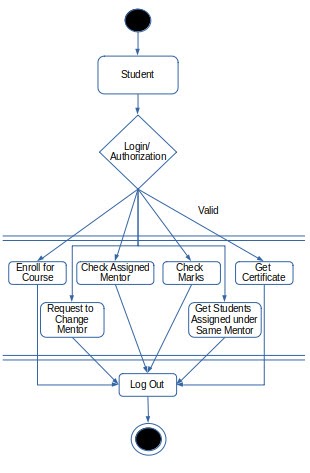
**3.1.5 Activity Diagram:**

An activity diagram portrays the control flow of SMGS from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

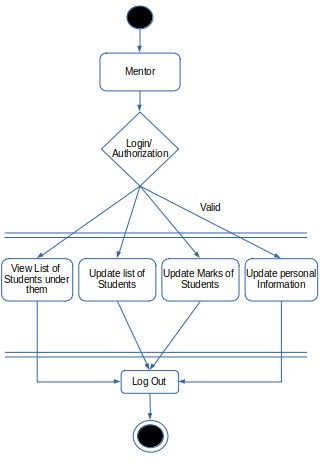
3.1.5.1 Admin Model:



3.1.5.2 Student Model:



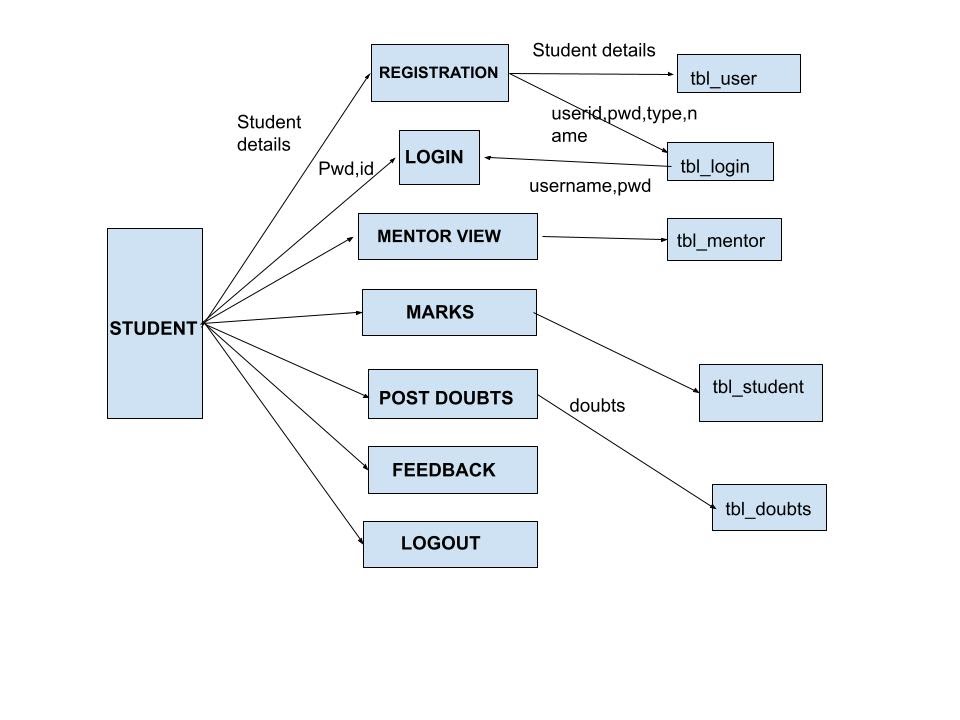
3.1.5.3 Mentor Model:



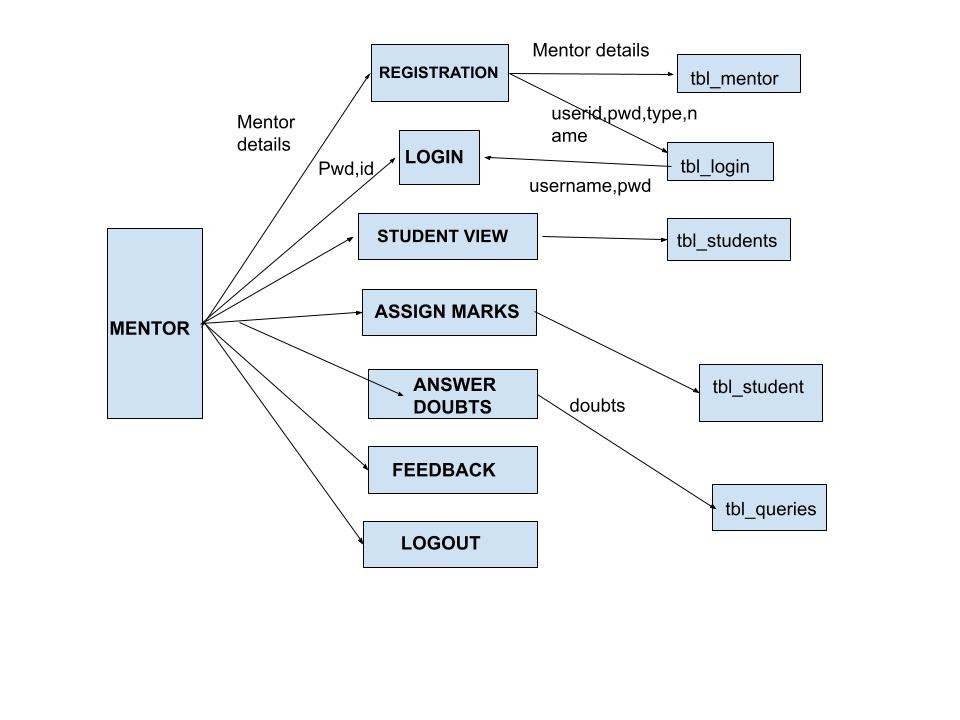
**3.1.6 DFD:**

Data Flow Diagram represent detailed and well explained diagram of system components

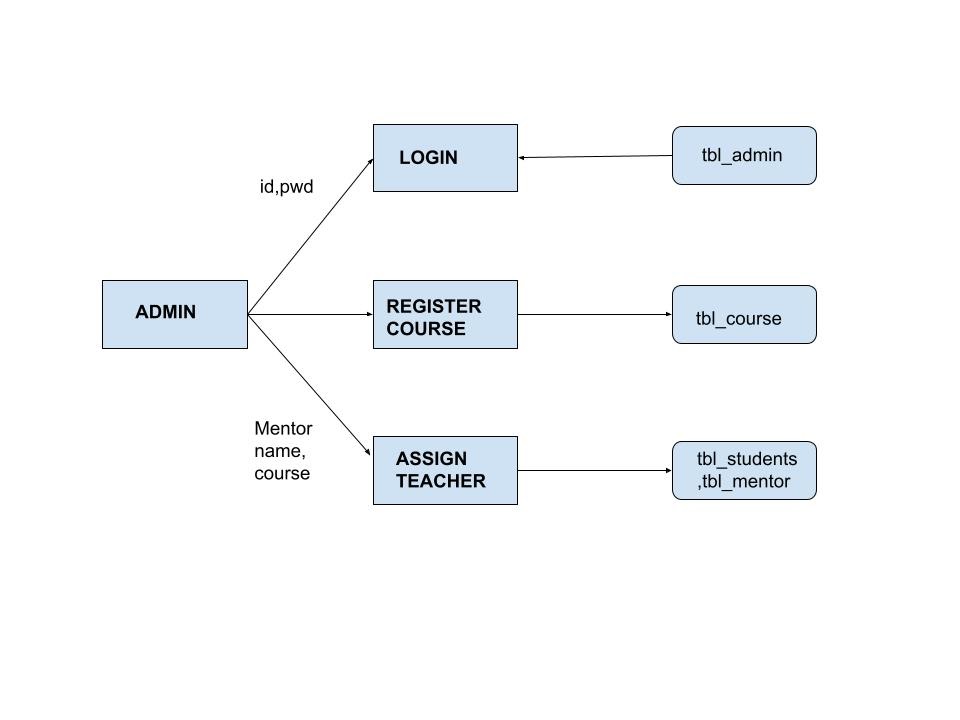
**Level1  - DFD Student Model:**



**Level 1 - DFD Mentor Model:**



**Level 1 - DFD Admin Model:**



**3.2 Usability**

The main purpose of developing the SMGS is to create a system so that,

3.2.1 Student - mentor relationship can be maintained in a more efficient way.

3.2.2 Mentors can keep track of the total number of students he/she is guiding.

3.2.3 Admin can see the assigned mentors and queries/feedback from the students.

**3.3 Reliability**

TItle: System Reliability

Desc: The reliability of the system.

Scale: The reliability that the system gives the right result on an operation.

**3.4 Performance**

The requirements in this section provide a detailed specification of the user interaction with the

software and measurements placed on the system performance.

3.4.1. Effective use of colours

Title: Colour Usage

Desc: Usage of colour coding to explicitly separate various subsections for enhanced readability and using different color codes for various courses.

3.4.2 Allowing multiple deletion

Title: Multiple deletion of various users

Desc: Multiple deletion is allowed simultaneously to ease out the work of the user ensuring consistency of data to be maintained effectively.

3.4.3 Guiding user to enter valid data

Title: Preventing illegal states

Desc: With the usage of alert boxes, guiding user to prevent entering any illegal data and thus

ensuring data to be consistent as well as non redundant.

3.4.4 Providing necessary and reliable options

Title: Options Reliability

Desc: Using javascript to hide and show visibility of reliable options at various interval of time

when necessary and at same time making it simple and smooth to run.

**3.5 Supportability**

The following supportability and maintainability features should be supported,

3.5.1 SMGS should adopt standards based integration for extensibility and scalability.

3.5.2 All code artifacts should have proper documentation.