 **School Management System**



**Lerato Oanyane**

**Noxolo Madikizela**

**Nkosinathi Dlamini**

**Mahlatse Ramabopa**

**Table of Contents**

1. Overall Description................................................................................................. 1
2. Scope/DoD............................................................................................................ 1
3. Entity Relationship Diagram.................................................................................... 2
4. Table Descriptions.................................................................................................. 3
5. Class Diagram........................................................................................................ 4
6. User Stories........................................................................................................... 5
7. Use Case Diagram.................................................................................................. 6
8. UI Flow Diagram.................................................................................................... 7
9. Wireframes............................................................................................................ 8
10. Project Backlog Items............................................................................................. 9
11. Daily Sprint Backlog Items..................................................................................... 10
12. List of REST API Endpoints.................................................................................... 11
13. Security: 3 Username, Password, and Role............................................................. 12
14. logging framework................................................................................................ 14
15. Design Patterns.................................................................................................... 16
16. Technical Architecture/Technology Stack................................................................ 19
17. Additional Documentation...................................................................................... 20
18. Lessons Learned for/by each person....................................................................... 25

**1. Overall Description**

A School Management System is a software application that helps schools manage their daily operations. It is designed to automate and streamline administrative tasks.

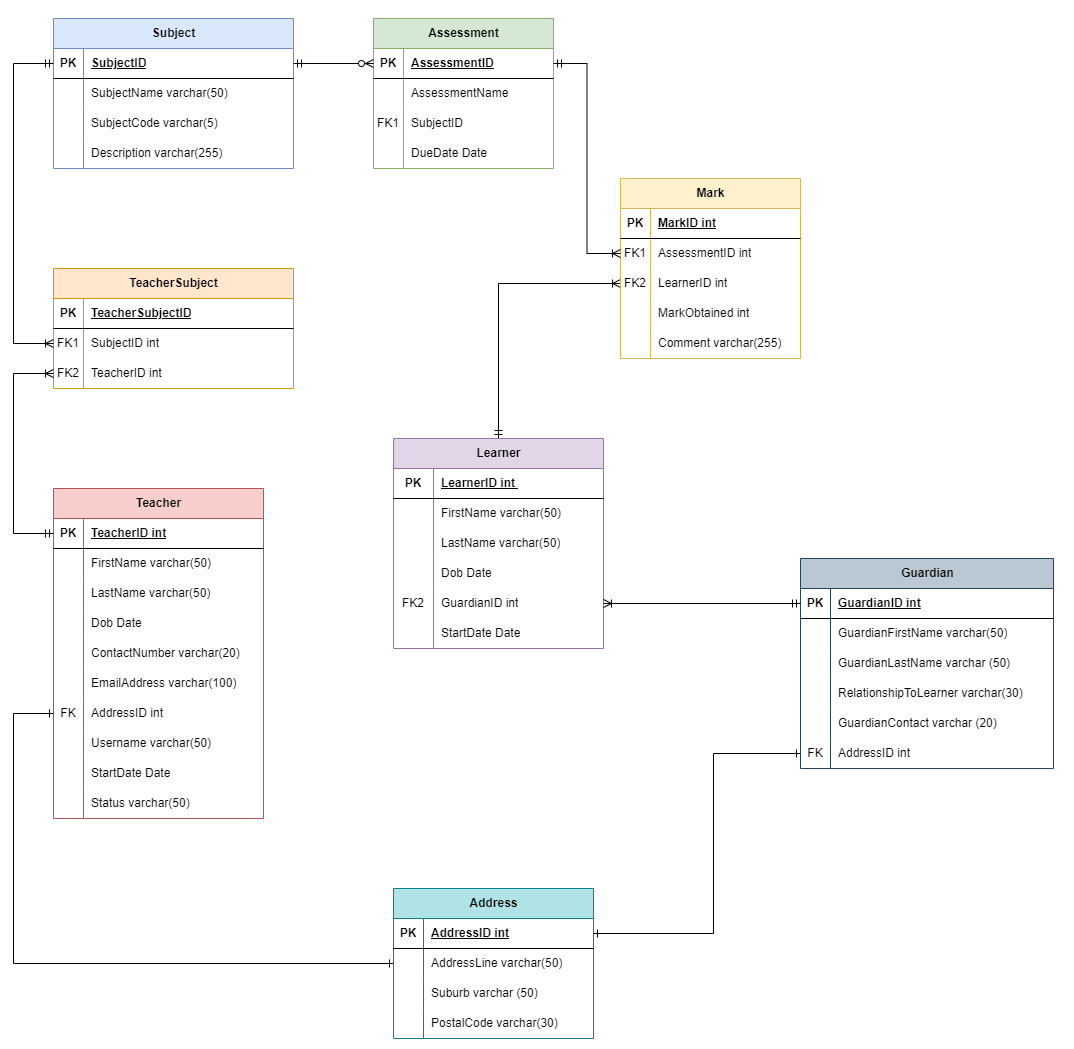
The system will be used by administrator/s, principal, and teachers to efficiently manage student records, assessments, grading and generating reports for the teacher to see the overall performance of the learners.

**2. Scope/DoD (Definition of Done)**

All users must be able to login.  
Code review completed and passed.

All testing is completed for both REST API and ASP.NET MVC.

**3. Entity Relationship Diagram**



**4. Table Descriptions**

|  |  |
| --- | --- |
| **Table Name** | **Description** |
| Address | * Contains the address line (street name and number), the suburb, and the postal code. * This table will be used to store teachers, learners, and guardians home address   Lerato |
| Teacher | * Contains the teacher’s first name, last name, date of birth, contact number, email address, address Id, username, and the start date * This table will be used to store all teacher's information   Noxolo |
| Learner | * Contains the learner’s first name, last name, date of birth, address Id, guardian Id, and the start date * This table will be used to store all learner's information   Lerato |
| Guardian | * Contains the guardian’s first name, last name, relationship with the learner, and the address Id * This table will be used to store all guardian’s information   Nkosinathi |
| Subject | * Contains the subject name, subject code and the subject description. * This table will be used to store all the subjects   Nkosinathi |
| Teacher Subject | * Contains the subject Id, and teacher Id * This table will be used to store information about the teacher and the subject they teach   Mahlatse |
| Assessment | * Contains the assessment name, subject Id, due date. * This table will be used all the store all the assessments associated to subject   Mahlatse |
| Mark | * Contain the assessment Id, learner Id, mark obtained, and the comment. * This table will be used to store the marks of the learners   Noxolo |
|  |  |

**5. Class Diagram**

**6. User Stories**

**User**

* As a user, I want to login in so that I can use the system’s features

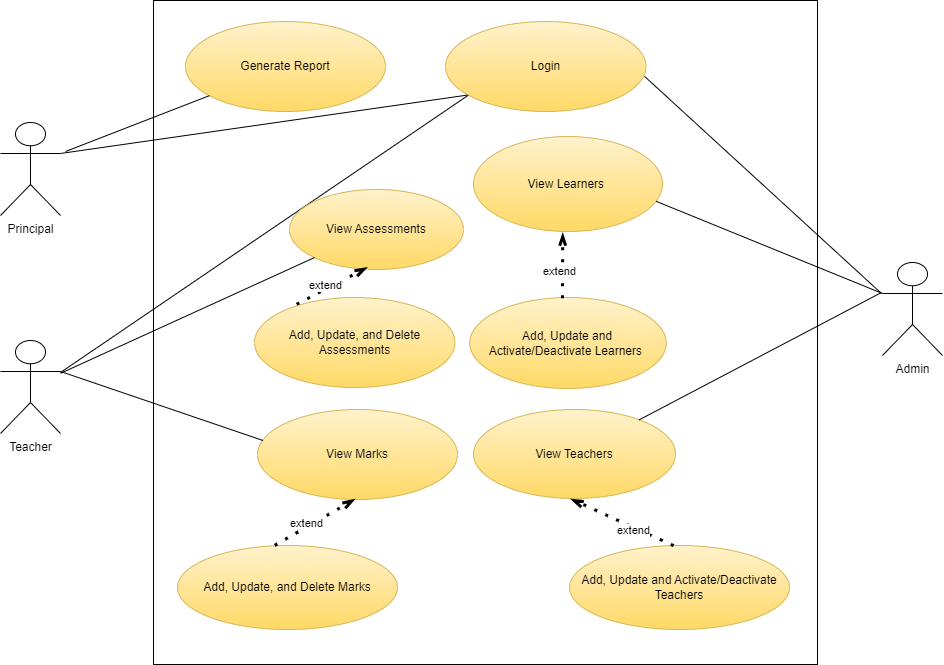
**Administrator**

* As an admin, I want to add new learner/s to the system so that I can maintain an up-to-date learner's database.
* As an admin, I want to view all the learners, so that I can see the list of all registered learners in the school.
* As an admin, I want to update learners' information so that I have accurate information about each learner.
* As an admin, I want to delete learners that are no longer with the school for more than 10 years so that we maintain an up-to-date database.
* As an admin, I want to add a new teacher, so that I can maintain an up-to-date teacher's database.
* As an admin, I want to view all the teachers, so that I can see the list of all teachers in the school.
* As an admin, I want to be able to update the teacher's information, so that I can have up-to-date information for each teacher.
* As an admin, I want to be able to activate and deactivate a teacher's account and information for a former teacher, so that I have information about existing teachers only.

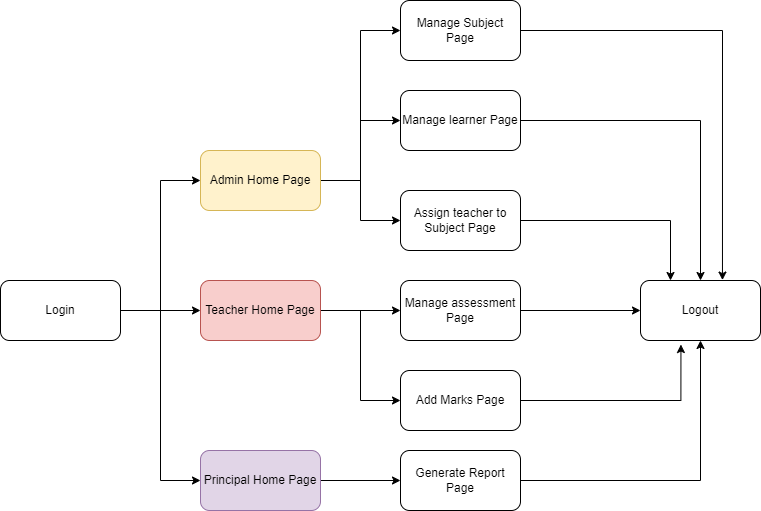
**Teacher**

* As a teacher, I want to create new assessments so that I can assign them to my learners.
* As a teacher, I want to add leaners marks, so that the system can compute totals and averages for me.
* As a teacher, I want to be able to update leaners marks, so that correct any incorrect learners' marks.

**7. User Case Diagram**

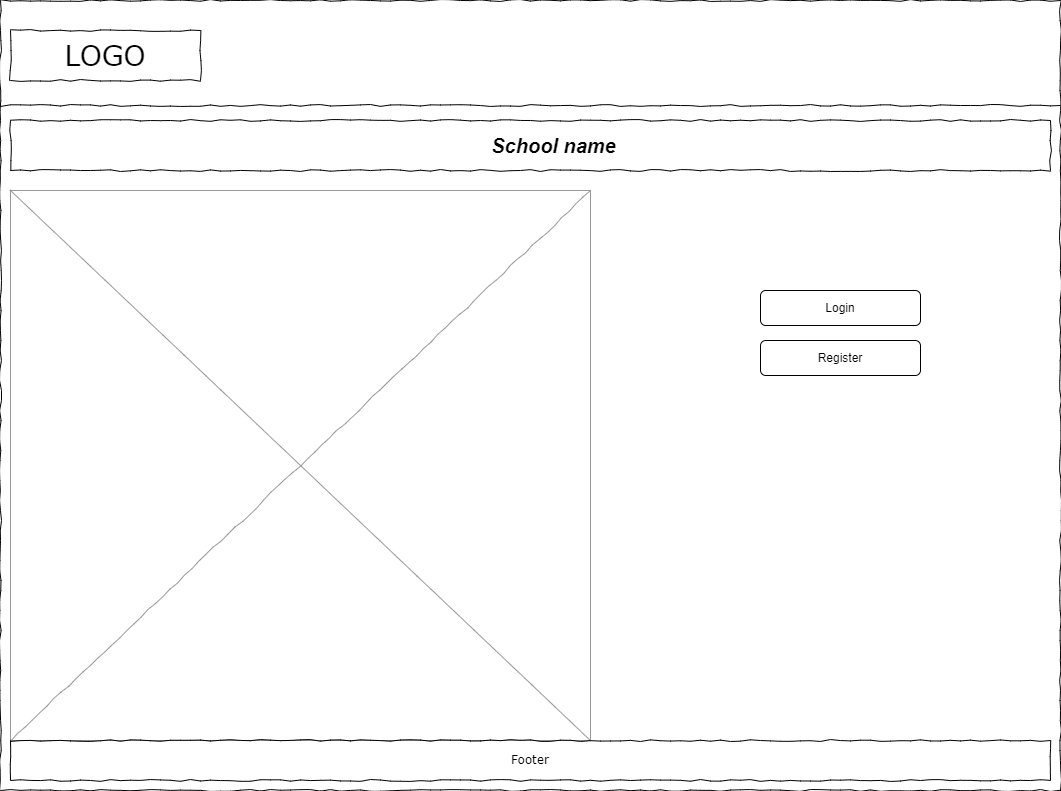


**8. UI Flow Diagram**



**9. Wireframes**

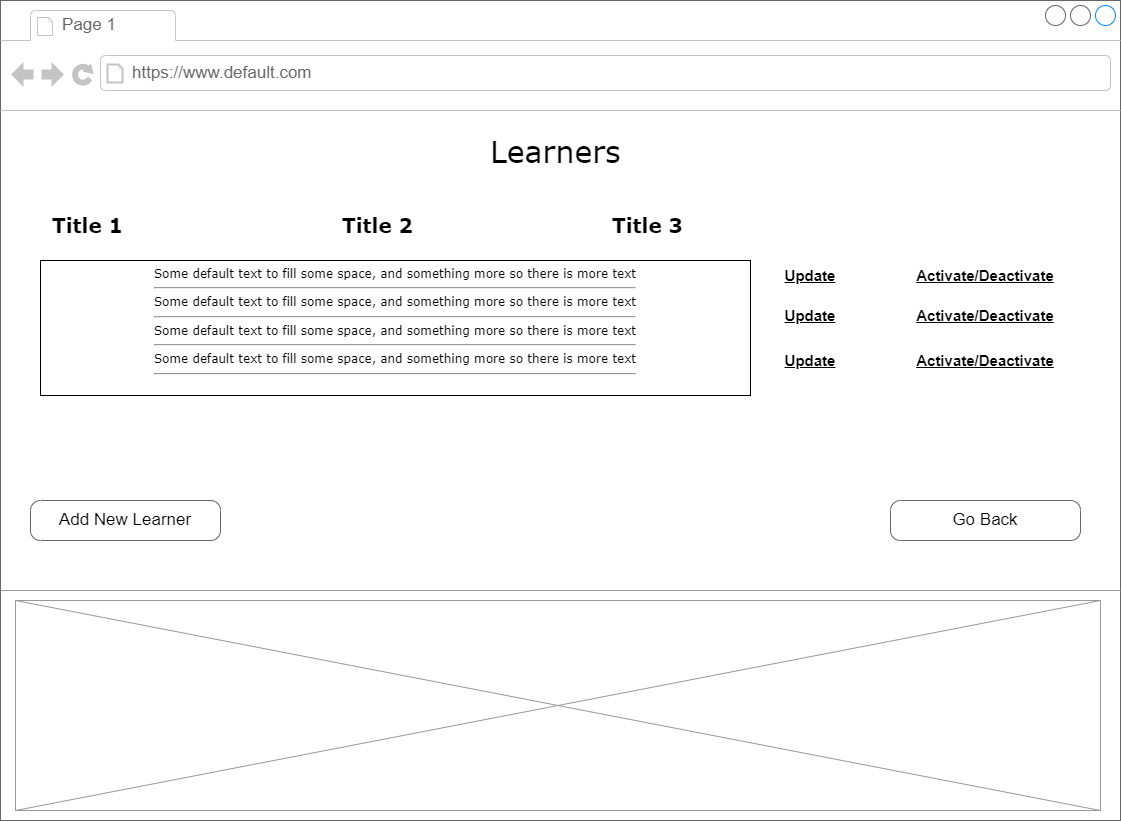
Home Page



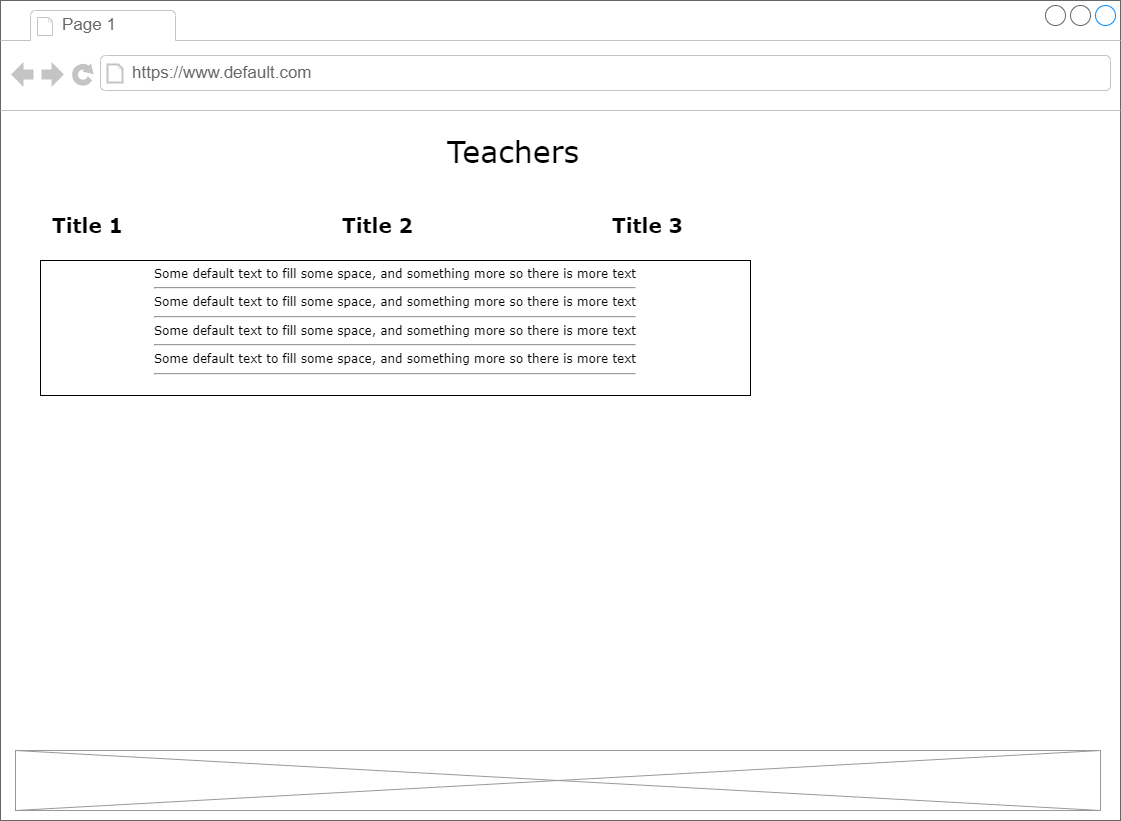
Admin Home Page



Admin – Learners Page



Admin – Teachers Page



**10. Project Backlog Items**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| School Management System - Product Backlog Items | | | | | |
| **ID** | **User** | **User Story** | **Assign To** | **Priority** | **Status** |
| 1 | User | As a user, I want to login in so that I can use the system’s features | Lerato | 1 | Done |
| 2 | Administrator | As an admin, I want to add new learners to the system so that I can maintain an up-to-date learner's database. | Mahlatse | 1 | Done |
| 3 | Administrator | As an admin, I want to view all the learners, so that I can see the list of all registered learners in the school. | Mahlatse | 1 | Done |
| 4 | Administrator | As an admin, I want to update learners' information so that I have accurate information about each learner. | Mahlatse | 2 | Not Started |
| 5 | Administrator | As an admin, I want to delete learners that are no longer with the school for more than 10 years so that we maintain an up-to-date database. | Mahlatse | 3 | Not Started |
| 6 | Administrator | As an admin, I want to add a new teacher, so that I can maintain an up-to-date teacher's database. | Noxolo | 2 | Not Started |
| 7 | Administrator | As an admin, I want to view all the teachers, so that I can see the list of all teachers in the school. | Noxolo | 1 | Done |
| 8 | Administrator | As an admin, I want to be able to update the teacher's information, so that I can have up-to-date information for each teacher. | Noxolo | 2 | Not Started |
| 9 | Administrator | As an admin, I want to be able to activate and deactivate a teacher's account and information for a former teacher, so that I have information about existing teachers only. | Noxolo | 2 | Not Started |
| 10 | Teacher | As a teacher, I want to create new assessments so that I can assign them to my learners. | Nkosinathi | 2 | Not Started |
| 11 | Teacher | As a teacher, I want to add leaners marks, so that the system can compute totals and averages for me. | Nkosinathi | 2 | Not Started |
| 12 | Teacher | As a teacher, I want to be able to update leaners marks, so that correct any incorrect learners' marks. | Nkosinathi | 2 | Not Started |
| 13 | Teacher | As a teacher, I want to view student's marks, so that I can ... | Nkosinathi | 2 | Not Started |
| 14 | Principal | As a principal, I want to view reports | Lerato | 2 | Not Started |

**11. Daily Sprint Backlog Items**

* Create a home page
* Create redirection for the home page
* Create the Administrator page
* Create the principal page

**12. List of REST API Endpoints**

## User Login and Registration

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **HTTP Method** | **Endpoint: User Logina and Register** | **Description** |
|  | POST | https://localhost:7021/api/ApplicationUser/Register | Register a new user. |
|  | POST | https://localhost:7021/api/ApplicationUser/Login | User login with registered username and password. |

## Administrator - Learners

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **HTTP Method** | **Endpoint: Learners** | **Description** |
|  | GET | <https://localhost:7021/api/AdminLearners/GetAllLearners> | Get all the learners. |
|  | GET | <https://localhost:7021/api/AdminLearners/GetLearnerById?id=5> | Get a learner by ID. Return learner. |
|  | POST | <https://localhost:7021/api/AdminLearners/CreateNewLearner> | Create a new learner. |
|  | PUT | <https://localhost:7021/api/AdminLearners/UpdateLearnerById?id=11> | Update a learner by Id. |
|  | DELETE | <https://localhost:7171/api/AdminLearners/DeleteLearnerById?id=20> | Delete a learner by Id. |

## Administrator - Teachers

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **HTTP Method** | **Endpoint: Teachers** | **Description** |
|  | GET | <https://localhost:7171/api/AdminLearners/DeleteLearnerById?id=20> | Get all the teachers. |
|  | GET | <https://localhost:7171/api/AdminTeachers/5> | Get a teacher by ID. Return teacher. |
|  | POST | <https://localhost:7171/api/AdminTeachers> | Create a new teacher. |
|  | PUT | <https://localhost:7171/api/AdminTeachers/11> | Update a teacher by Id. |
|  | DELETE | <https://localhost:7171/api/AdminTeachers/Deactivate?id=11> | Delete a learner by Id. |

## Principal

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **HTTP Method** | **Endpoint: Principal** | **Description** |
|  | GET | <https://localhost:7171/api/Principal/Learners> | Get all the learners. |
|  | GET | https://localhost:7171/api/Principal/Teacher/Status/active | Get all teachers by status. |
|  | GET | <https://localhost:7171/api/Principal/Learner/Detailed> | Get a detailed list learners. |
|  | GET | <https://localhost:7171/api/Principal/TeacherSubject> | Get all teachers and their subject/s. |
|  | GET | <https://localhost:7171/api/Principal/Learner/Subject> | Get all learners and their subject/s. |
|  | GET | <https://localhost:7171/api/Principal/Learner/Assessment> | Get all learner assessment/s. |

**13. Security: 3 Username, Password, and Role**

|  |  |  |
| --- | --- | --- |
| Username | Password | Role |
| AdminOne | Admin!1SE5 | Administrator |
| TeacherOne | Teacher$125 | Teacher |
| PrincipalOne | Principal$123% | Principal |

**14. logging framework**

* log4net

**15. Design Patterns**

* Repository Pattern
* MVC

**16. Technical Architecture/Technology Stack**

**Back-end:**

* Visual Studio, C#, Entity Framework, ADO.NET, ASP.NET Core Web API

**Front-end:**

* ASP.NET Core Web App MVC

**Database:**

* MS SQL Server

**REST-API Endpoints**

* Swagger UI.

**Testing**

* Moq Framework
* Nunit Framework
* xUnit Framework

**17. Additional Documentation**

**18. Lessons Learned for/by each person**

**Mahlatse**  
  
- Better understanding of the software life cycle.

- Requirement specification writing and practicing the SCRUM framework

- Using C# .NET to develop a Full Stack Web Application using different .NET frameworks.

**Noxolo**

-Experience working in a scrum team

-Better understanding of design patterns and their use

-Understanding of the MVC architecture

**Lerato**

**-** Time management and effective communication between the team members

- Better understanding of LINQ

- Better understanding of the MVC pattern