School of Electrical and Information Engineering, University
of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

ELEN 4009 - Software Engineering

Student Marks/Records Management Software

Front end pair : Khosa Masana (559990) and Londiwe Ngema (448871)

Back end pair : Tshegofatso Misapitso (600313) and Sanele Gcaba (459380)

April 11, 2016

.

Contents

Document Status Sheet

Document Title	Software Requirements
Authors	M. Khosa, S. Gcaba, T. Misapitso, L. Ngema
Version	0.01
Document Status	Draft

Version	**Date**	**Author(s)**	**Summary**
0.0.1	25-02-2016	M. Khosa, S. Gcaba, T. Misapitso, L. Ngema	Document Creation.

.

1. INTRODUCTION

1.1 Problem Statement

University of the Witwatersrand school of electrical engineering and Information Engineering aims on implementing a software to manage provisional students marks as they complete assignments, tests, laboratory work, etc., throughout the year. The software to be implemented will have three distinct users which are the course coordinator, the school administrator and the student. Each user has different levels of access to the database. The course coordinator of each course, offered by the school, must be able to enter marks obtained by students for all forms of assessment like tests, assignments, labs, exams and many more depending on the structure of a particular course. Students must be able to retrieve and view their assessment marks obtained so far, for all forms of assessment on each course. This allows students to track their progress throughout the year as they continuously check their marks on the system. The school administrator must be able to retrieve relevant summaries of marks for each course in a spread-sheet format which will be analysed by the course coordinator and the head of school and then be sent to the official mark system of the faculty. All the marks stored, for each course, in an academic year must be kept in the database for 10 years. The users must be able to interact with the database through a web browser and a smart mobile application.

1.2 Project Objectives

The objective of this project is to design, implement and develop a student marks software that is user-friendly and cost effective. The software has only three types of users which are the course coordinator, school administrator and a student. The users must interact with the database through a web browser. The software must be able to allow the course coordinator, of each course offered by the school, to input marks obtained by the students for all forms of assessments. The students must be able to access the software and retrieve their marks throughout the year. The school administrator must be able to retrieve relevant summaries of marks for each course in a spreadsheet format. The project will be divided into two parts namely the back-end and front-end. The front-end of the software is going to be developed using HTML, CSS, JavaScript and PHP. The back-end is developed using PHP and mySQL. The two parts are thereafter combined into one student record/mark system. There is no budget allocated to this project since it can be designed, implemented and developed using readily available resources such as computers. The server used is an open source Apache server. The time allocated to complete this project is 2 months and 3 days, From 9th of February to the 11th of April.

1.3 Stakeholders

Figure 1 shows the stakeholders for the project.

1.4 Deliverables

At the end of this project the application website is expected to atleast enable users to:-

- Register themselves.
- Identify which group among the three mentioned above the user belongs to.
- If the user is a student, the most important thing is for them to track their progress by having all their marks on the system.
- It is important for the Admin to retrieve a table of student names and their results.
- Another important task for the Course Co-ordinator is for him/her to add various forms of assessment and also input the student marks.

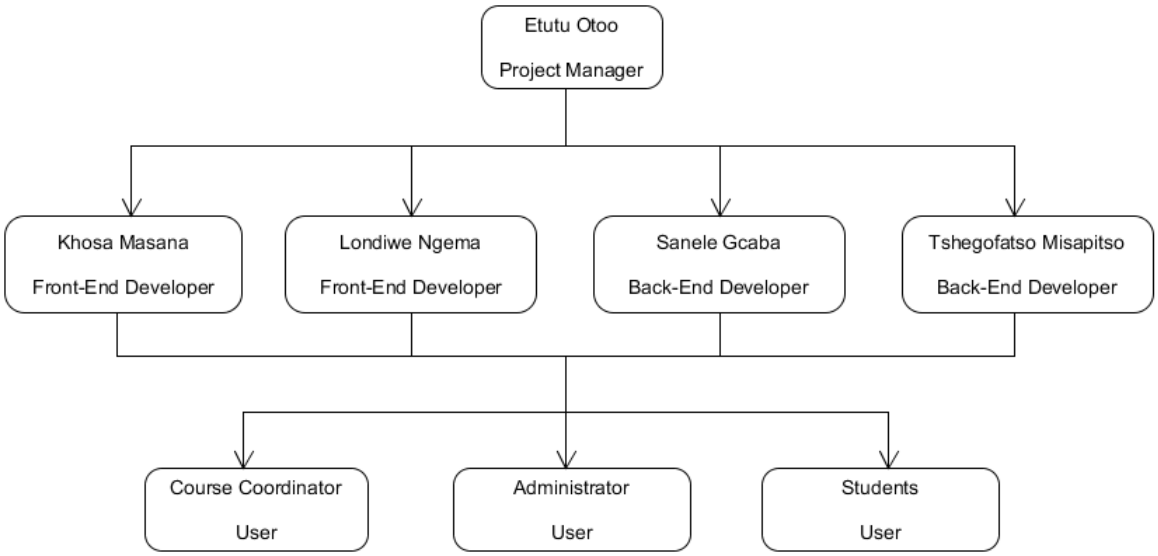


Figure 1 : Project Stakeholders

2. Requirements Gathering

2.1 Methodology

The System Development Life Cycle (SDLC) methodology to be used for the project follows the Agile Model, this model breaks down the product into a cycle, it quickly delivers a working product and is a more realistic methodology. Ongoing releases are produced with small incremental changes and it depends heavily on customer interaction [1]. Specifically the Scrum Agile Model will be adopted for the project, Scrum comprises of short sprints and it enables the software team to be able to prioritize on what matters most. The Scrum Model is basically about delivering more often and responding to feedback regularly offered[2].

2.2 Purpose

The purpose of this document is thus to present a detailed description of the requirements, design and implementation of the student marks/record system (SMMS). It will state the purpose and features of the system, the interfaces of the system, what the system will do, the limitations under which it must operate, define inputs and the expected reaction of the system (that is the outputs of the system) and present the designed and implemented web application back-end and front-end subsystems.

Student Marks/Records Management Software provides on-line services that enable students to view their marks as they complete assignments, tests and laboratory work, the software system allows staff such as course coordinators the right to add/edit marks obtained by the student under different forms of assessment. It is a convenient way for students to have access to their marks in a safe and confidential way as opposed to accessing them on notice boards where everyone else can publicly see them. The system ensures that each student can track his/her tentative progress throughout the year,it also helps in establishing errors in the record as early as possibly.

2.3 Project Scope

- There are three basic users of the system - Students, Course coordinators and School administrator.
- The primary function of the application is to allow students and staff to log-in using their details (Student/Staff number and password) and be able to access and view student records, the records include student marks for all forms of assessments for all courses registered for.
- The system database stores user profiles and student marks records. Marks records are retained for a period of 10 yrs.
- The software program should have domains assigned, i.e. each user can be able to access relevant information and they are allowed to view/edit/add based on what their recognised domain.
- The system would be accessed online via a Browser.

Below is a list of privileges per user as specified on the project brief [3].

The Course Coordinator should be able to:-

- Register himself/herself.
- Add various assessment method for the course and weighting for each assessment.
- Enter student marks on a user-friendly interface.
- Display or print out the table of students and their marks.
- Generate a summary statistics of the performance of the students - maximum, minimum, average, standard deviation or variants of each assessment.
- View projected pass rate based on the assessment marks accumulated students in class.

The School Administrator should be able to:-

- Register himself/herself.
- Display or print out table of students and their marks.
- Generate a summary statistics of the performance of the students.
- Generate a comparative chart of the assessment marks of selected courses being taken by students of a particular group.
- Histogram of assessment marks of all courses taken by a specific student.
- Any recorded offences (e.g. plagiarism) for a student.
- The performances in the same course across different years may be compared.

The Student should be able to:-

- Register himself/herself.
- Display assessment marks for a course and statistics for that assessment.
- Display assessment marks for all the courses
- Based on current assessment marks, give what performance goals are needed to pass the course.

2.4 List of Definitions and abbreviations

2.4.1 Definitions

Term	**Definition**
Database	A collection of records stored within the system
Table	A collection of related data consisting of columns and rows
Client	A requesting program or a user

2.4.2 Abbreviations

SMMS - Student Marks/record management system

HTTP - Hypertext Transfer Protocol

HTML - HyperText Markup Language

RDBMS - Relational Database Management System

SQL - Structured Query Language

2.5 Tools used

Web server - Apache2

The Apache web server is the worlds most used web server software program, it uses HTTP to serve files that form web pages to users in response to their requests. It is an open source program [4, 5].

Development tools (Front-End) - HTML, CSS and Javascript

Development tools (Back-End) - PHP

PHP is a server scripting language, it is widely used, free, and efficient tool.

Database Platform - MySQL and PHPmyadmin

MySQL is an open-source RDBMS, it stores data in tables and PHPmyadmin is a graphical tool intended to handle administration of MySQL over the web, while phpMyadmin provides a user interface for the mySQL RDBMS.

2.6 Expanded Description of the project

The software system will follow a Two-Tier Architecture due to its ease of use and maintainability as compared to a Three-Tier Architecture. However, the performance of a Two-Tier Architecture slows down with an increase in users [6], hence a Three-Tier Architecture will be implemented with an increase of the number of users. Figure 1 below shows the Two-Tier Architecture for the student marks/record system.

A Two-Tier Architecture is a software architecture where the interface runs on a client and the data layer is stored on a server[7].

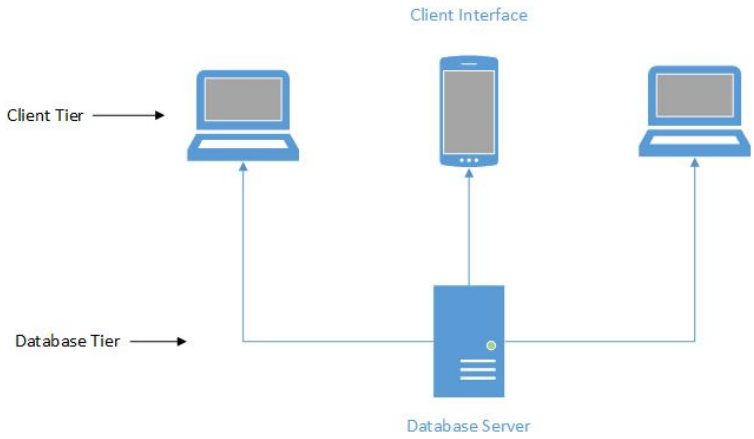


Figure 2 : Two-Tier Software Architecture

2.7 Constraints

- The User Interface language is English only.
- The program currently runs on a localhost.

2.8 ER Diagrams

The ER Diagrams below show how data is organized within the database.

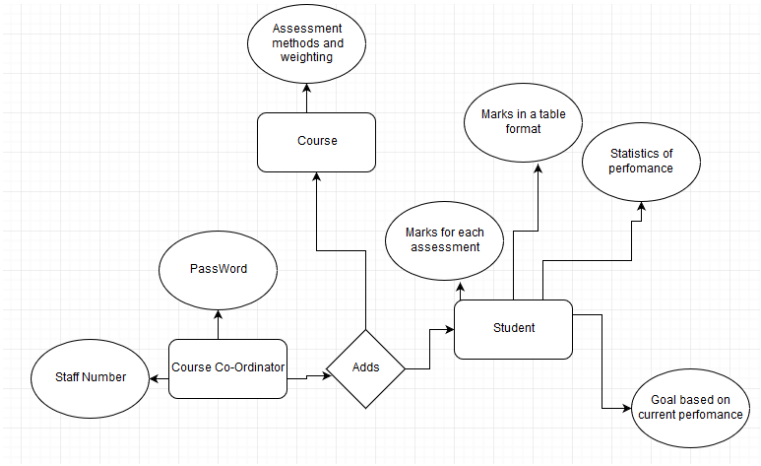


Figure 3 : ER Diagram for Course co-ordinator and student

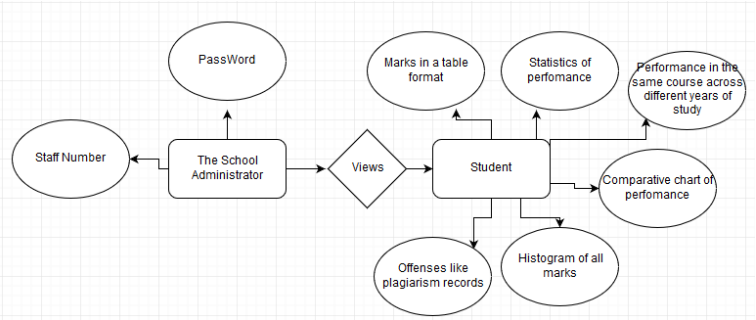


Figure 4 : ER Diagram for the administrator and student

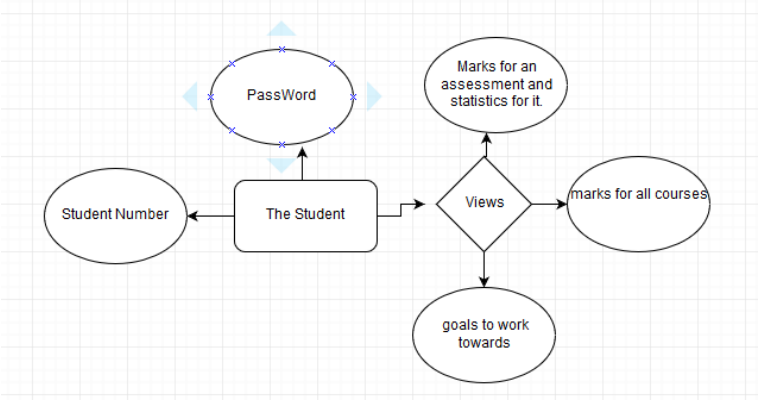


Figure 5 : ER Diagram for student

2.9 Hardware Specifications

Minimum Requirements

Client side

Windows	**Processor**	**RAM**	**Disk Space**
Internet Explorer	Intel Pentium III or AMD - 800 MHz	128 MB	100 MB
Google Chrome	Intel Pentium IV - SSE2 capable	128 MB	100 MB

Mac	**Processor**	**RAM**	**Disk Space**
Google Chrome	64 bit Intel processor	128 MB	100 MB

Linux	**Processor**	**RAM**	**Disk Space**
Google Chrome	Intel Pentium III processor	128 MB	100 MB

Server side

Linux	**Processor**	**RAM**	**Disk Space**
Apache 2	2 GHz processor or faster processor	1 GB (32 bit) or 2 GB (64 bit)	

Recommended Requirements

Client side

Windows	Processor	RAM	Disk Space
Internet Explorer	1 GHz or faster	1 GB (32 bit) or 2 GB (64 bit)	256 MB

Mac	Processor	RAM	Disk Space
Google Chrome	64 bit Intel processor	1 GB	256 MB

Linux	Processor	RAM	Disk Space
Google Chrome 1 GHz or faster	1 GB (32 bit) or 2 GB (64 bit)	256 MB	

Server side Server specifications will depend on the development of the project.

3. Part A : Front-end

This section presents the design of the user interface of the software. The user interface provides a visual platform for all the users. It provides a more user friendly environment to the client's database queries. The users are able to read and write from and into the database through the user interface. The user interface was implemented using HTML, JavaScript, CSS, and PHP. HTML was used for designing attributes or objects on user interface platforms. JavaScript was used for the animations on the welcome page. CSS gives application its custom look. PHP was used to help users query the database through the user interface.

The user interface consist of a login page. In the login page, all the users will be asked to input login details and that includes the user-name, password and domain. The login details will be queried into the database to see if they are valid. If the login details are not valid, an error message will be displayed to the user. If the login in details are valid, the user will be redirected to a specific page according to the specified domain. If the user is a course coordinator, the user will be redirected to a course coordinator page, if the user is an administrator, the user will be to an administrator page and same applies for the student. The page where a user will be redirected to depends on the domain specified. Each page where users will be redirected to has a logout button that takes the user back to the login home page.

3.1 Login page design

HTML: Different HTML tags used to describe HTML documents were used to describe the login page. The login page consists of the welcome text and wits pictures to create a more attractive but simple welcome page. A form with input tags for users to input the user-name and password was placed on the login page. The form also consist of a drop-down tag to select the domain. A submit button was also placed to submit the form with login details.

CSS: A CSS file was made for the home page. All the tags mentioned above were given an "id" which is used for reference in the CSS file. In the CSS file for the login page, each tag was given a position, colour, opacity, background colour and all the other styling necessary.

JavaScript: When the login page is opened, a welcome title slides in from the left of the page up until the middle of the page, when the welcome tittle is done sliding, a wits logo is displayed behind the welcome statement. All theses animations of objects on the login page were done using JavaScript.

PHP: PHP receives the posted form that has the login details of a specific user after when the user presses the submit button. The login details in the form are then sent to the database for verification. If the login details are valid, the user is then redirected to either a student page, course co-ordinator page or administrator page depending on the domain specified.

3.2 Student page design

HTML: The student page was designed to be more user friendly. The student page consists of a menu section where the student can select the options based on their level of access. All the options in the menu section are hypertext references that redirect the student to a specific page based on the selected option. Adjacent to the menu section is a 'more-information' section that explains in detail what each of the menu options is for.

The options in the menu section are:-

- Assessment marks for a course
- Statistics
- Assessment marks for all courses
- Performance goals

CSS: The styling of each attribute on the student page was defined on the CSS file.

PHP: When each of the hypertext references in the menu section is clicked, the student is redirected to a specific page depending on the option chosen. When a student selects assessment marks for all courses option, his/her student number will be used to query a table with final marks associated with that student number and the final marks for each course will be displayed to the student. When the student selects the statistics option, the student will be redirected to a page where pass rates and other statistics of each course will be displayed. When a student selects the assessment marks for a course option, the student will be redirected to a page where a list of courses which that particular student is doing will be displayed. When a student selects a particular course, marks for all forms of assessments (labs, assignments, tests and exams) for the course selected will be displayed. And lastly, when the student selects the performance goals option, the student will be redirected to a page that displays a message on how to improve based on their current marks.

3.3 Course Coordinator page design

HTML: The course coordinator page consist of a menu section with options limited to his/her privileges. Adjacent to the menu section is a 'more-information' section that further explains the options that can be selected by the course coordinator.

The options that are in the menu section are

- Assessment Methods
- Student Marks
- Table of Students
- Statistics
- Pass rate

CSS: All the styling for the attributes in the course coordinator page were defined in the CSS file.

PHP: PHP was used to redirect the course coordinator to appropriate pages when a selection on the menu is made. When the course coordinator selects the Assessment methods, the coordinator will be redirected to a page where he/she will be allowed to upload documents with the assessment methods and the weighting for each assessment. When the course coordinator selects the student marks option, the course coordinator will be redirected to a page where he will be allowed to print the a table of students and their marks. When the course coordinator selects the table of the students option, he/she will be redirected to a page to enter the marks of students into the database. The statistics option will redirect the course coordinator to a page where there is a summary of statistics for each course to be displayed. Lastly, the pass rate option will redirect the course coordinator to a page where the pass rate of the course will be displayed.

3.4 Head of School/Administrator page design

HTML: The administrator page consists of a menu section with options of things that an administrator can do. Adjacent to the menu section is the 'more-information' section where each option is further explained.

The menu section on the administrator page has the following options.

- Table of students
- Statistics
- Comparative chart
- Histogram
- Offences
- Performance comparison

CSS: A CSS file was created for the administrator page. All the styling of the attributes in the administrator page were designed in the CSS file.

PHP: The option: table of students, statistics and table of students work the same way as described in section

2.3 (Course Coordinator page design). The histogram option redirects the administrator to a page where a histogram of assessment marks of all courses taken by a specific student will be displayed. The Offences options will redirect the administrator to a page where a downloadable pdf file with all offences,like plagiarism, committed by students will be recorded. The performance comparison option redirects the administrator to a page where the administrator will be asked specify a course, after the course name has been specified, a graph showing pass rate percentages for the past ten year for the specified course will be displayed.

3.5 Use case diagrams

3.5.1 Student Use-Case Report

A use case report for a student is shown in Figure 2.

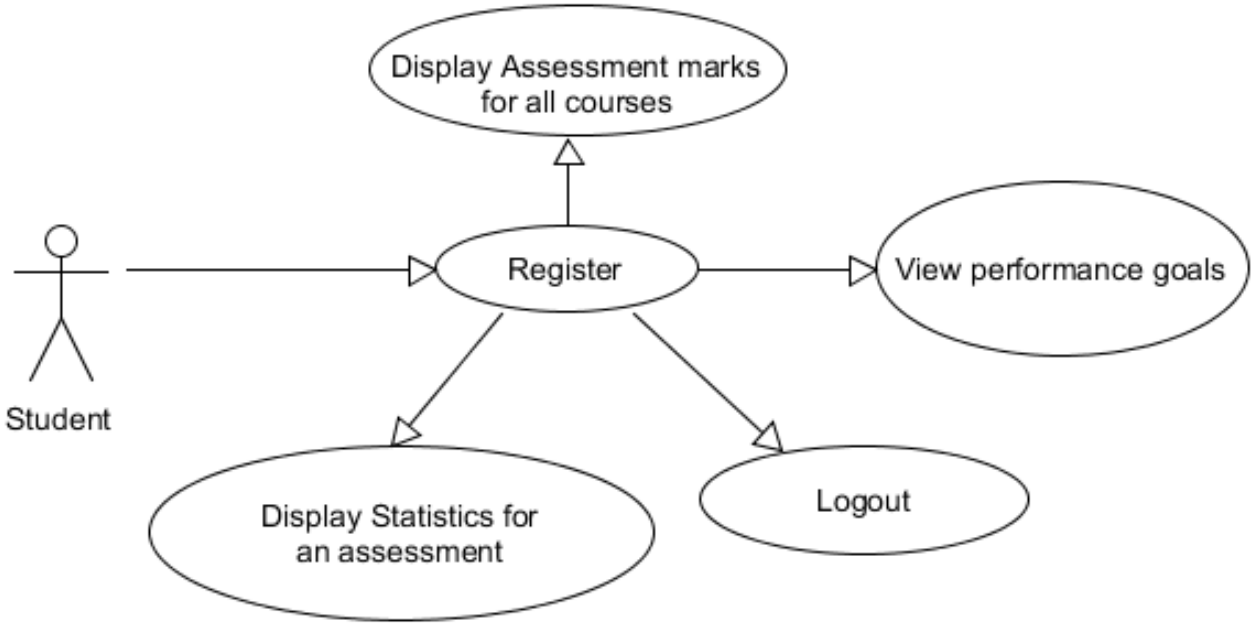


Figure 6 : Use Case Diagram For Student

Use case	**Description**
Register	The student need to sign in in order to view the results
Display assessment marks	A student must be to display assessment marks for all courses
Statistics for an assessment	A student must be able to display assessment marks for a course and the statistics for that assessment
View performance goals	A student must be able to view what perfomance goals are needed to pass pass the course based on current assessment
Logout	The student must be able to logout

3.5.2 Course Coordinator Use-Case Report

A use case report for Course Coordinator is shown in Figure 3.

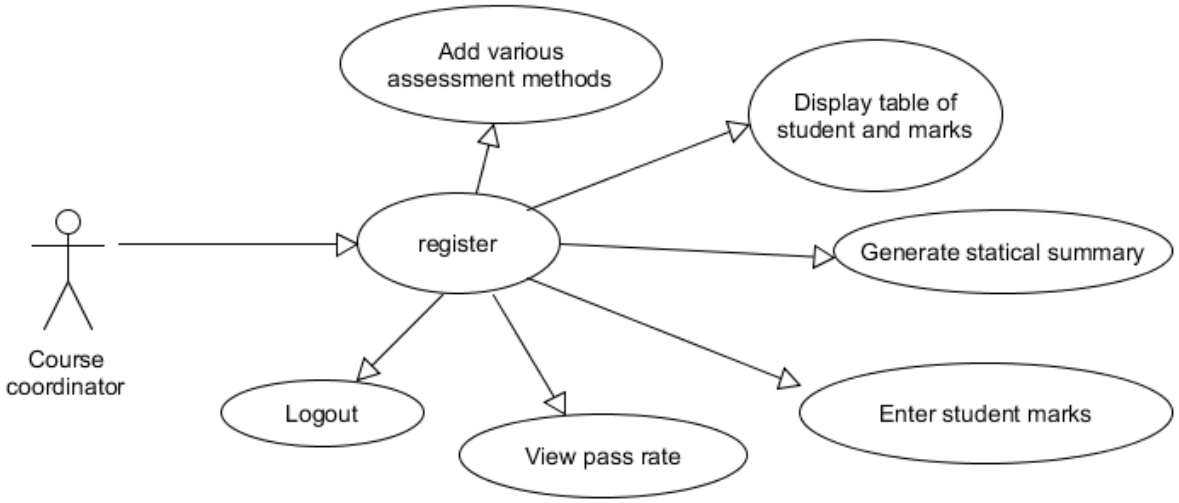


Figure 7 : Use Case Diagram For Staff

Use case	**Description**
Register	Staff need to sign in order to modify results
Add various assessment methods	Add various assessment method for the course and the weighting for each assessment
Display table of student and marks	Display or print out the table of studets and their marks
Generate statical summary	Generate A summary statistics of the perfomance of each student
Enter student marks	Enters the student's marks for each assessment in a user-friendly interface
View pass rate	View projected pass rate based on the assessment marks accumulated by the students in the class thus far
Logout	Coordinator member must be able to logout

3.5.3 Administrator Use-Case Report

A use case report for an Administrator is shown in Figure 4.

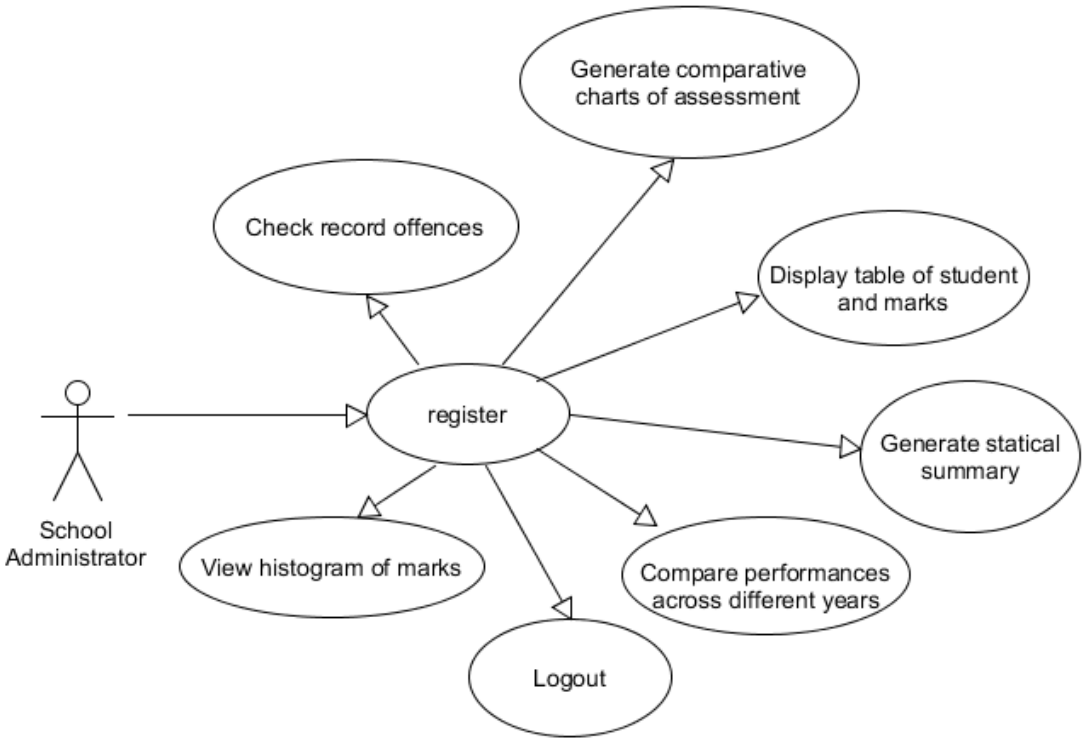


Figure 8 : Use Case Diagram For Administrator

Use case	**Description**
Register	Administrator need to sign in
Check record offences	The administrator must be able to check any record offences like plagiarism for a student
Generate comparative charts of assessment	Generate a comparative chart of the assessment marks of selected courses being taken by students of a particular group
Display table of student and marks	Display or print out the table of studets and their marks
Generate statical summary	Generate A summary statistics of the perfomance of each student
Compare performances across different years	compare performances in the same course across different years
View histogram of marks	View a histogram of assessment marks of all courses taken by a specific specific student
Logout	Administrator must be able to logout

4. Sequence Diagram

The sequence diagram in Figure 5 shows Sequence diagram of a student.

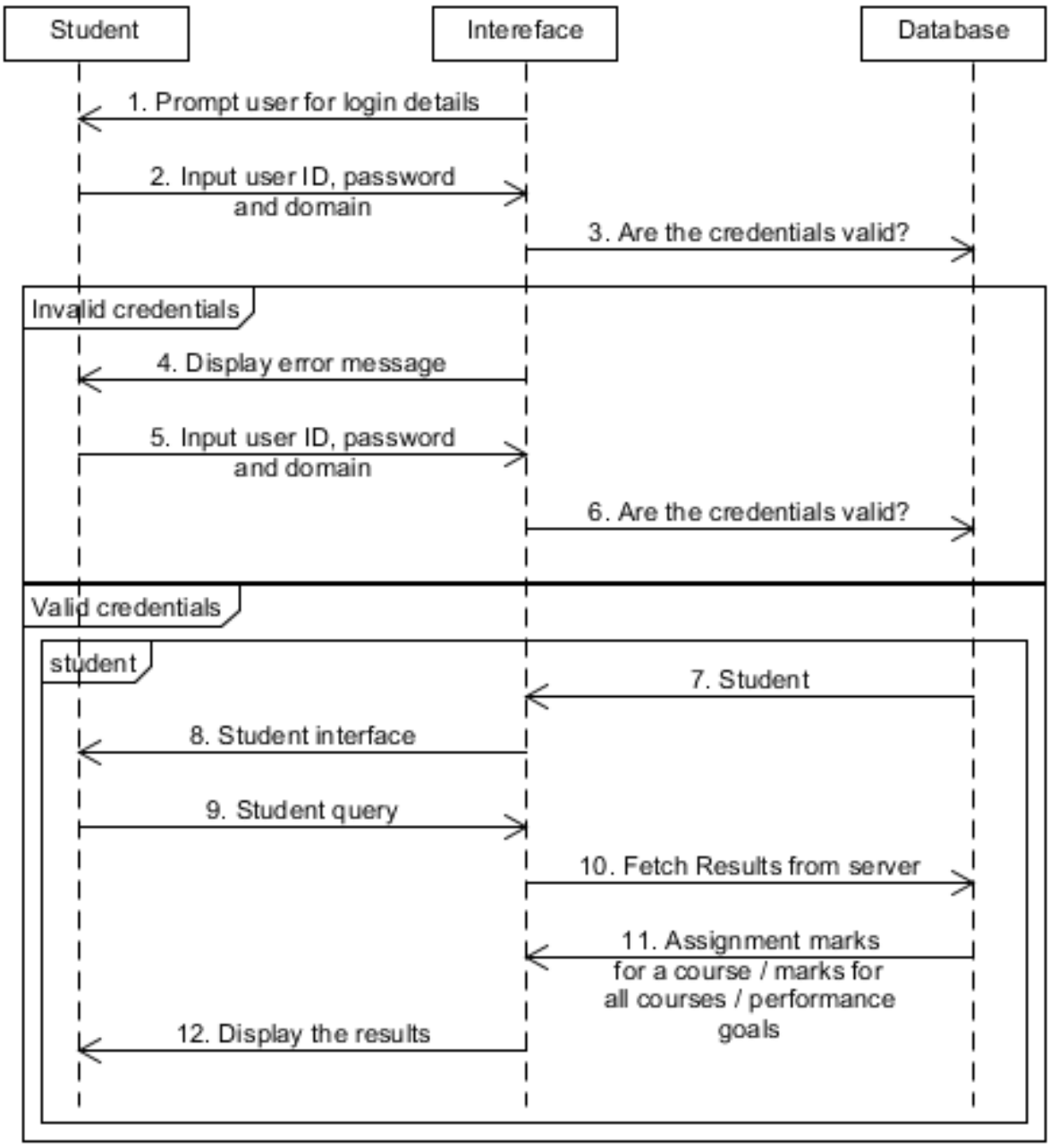


Figure 9 : Student Sequence Diagram

The sequence diagram in Figure 6 shows Sequence diagram of an administrator.

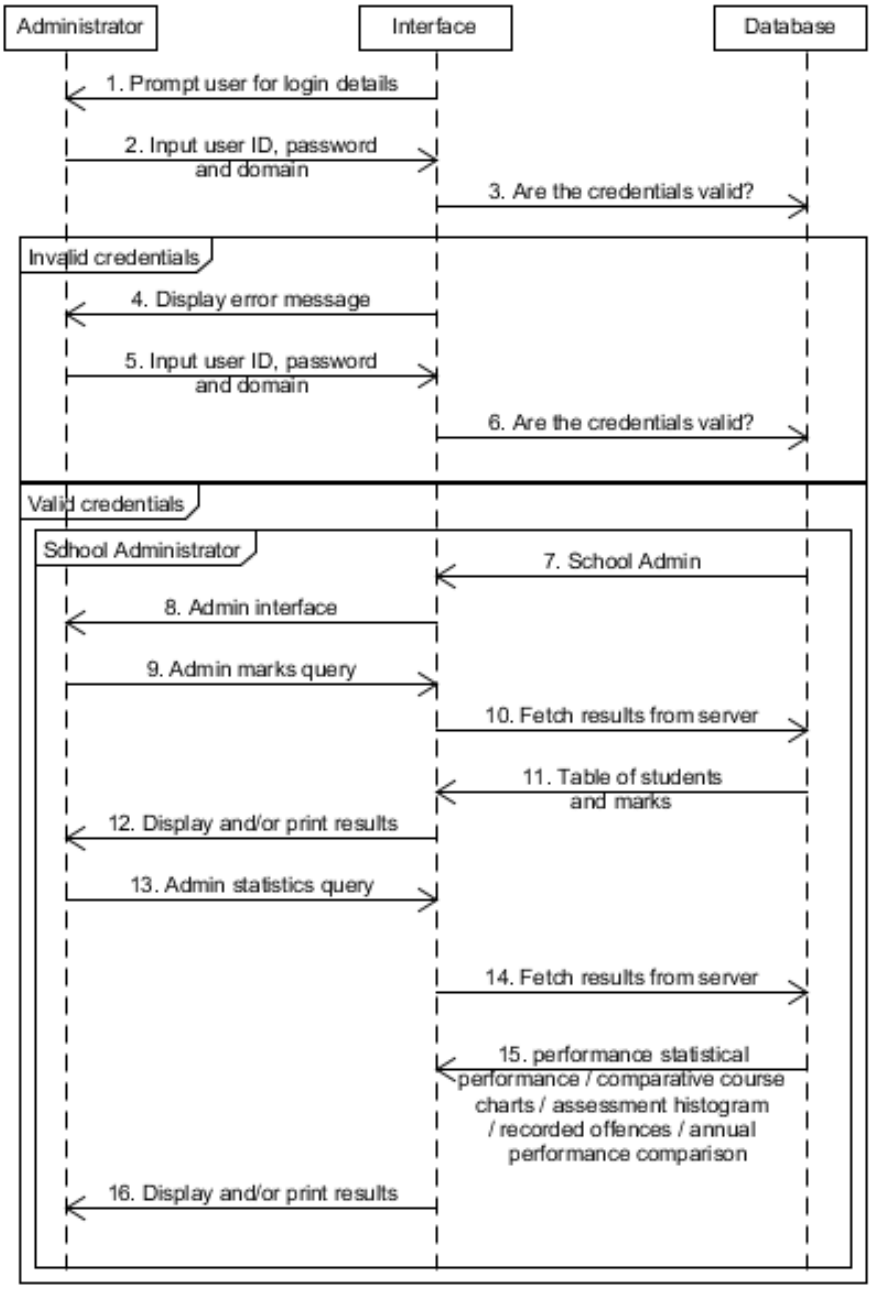


Figure 10 : Administrator Sequence Diagram

The sequence diagram in Figure 7 shows Sequence diagram of an course coordinator.

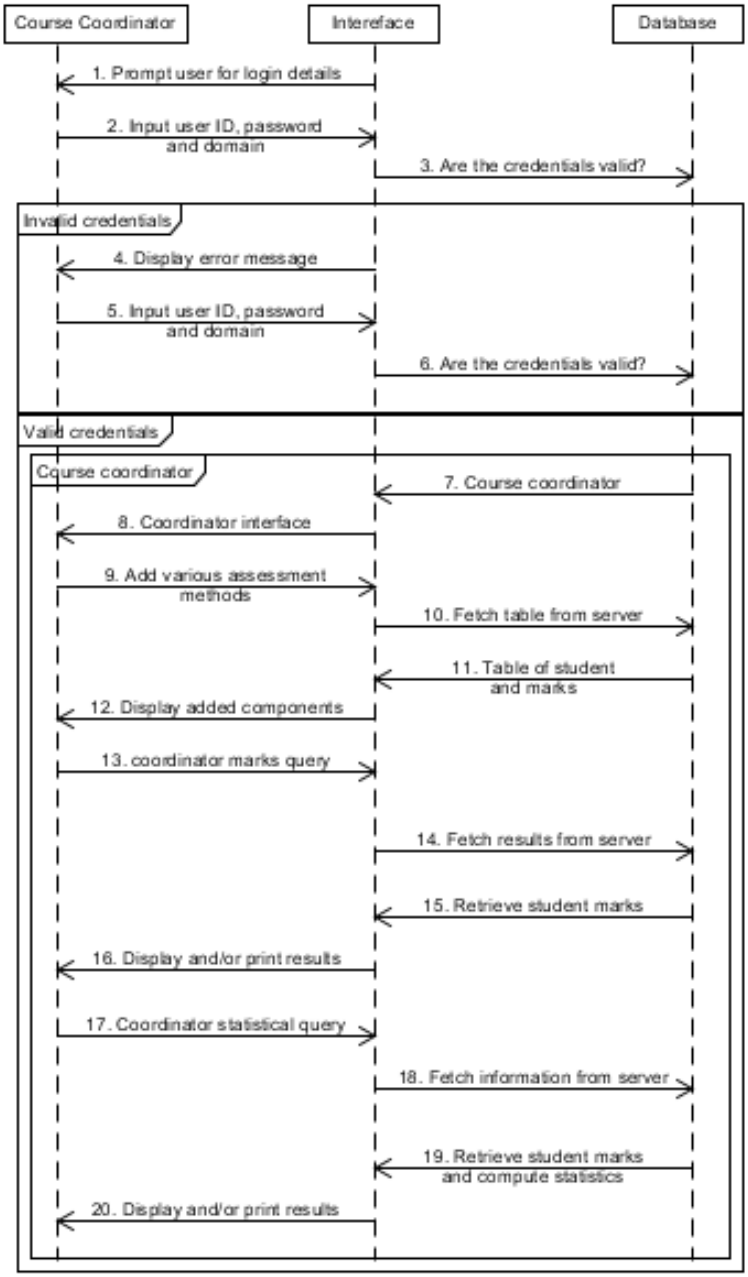


Figure 11 : Course Coordinator Sequence Diagram

5. Part B - Back-End

The Back-End constitute of three parts, these include the server,the database, and the application(Generally referred to as the business model), hence the student marks/record system is designed following a two tier architecture as described in the requirements gathering section. The developed Back-End follows a LAMP web development architectural design. That is, **L**inux is used as the operating system for the server to run on, **A**pache is deployed as the application server, **M**ySQL is used as the Relational SQL Database Management System (RDBMS) and **P**HP is used as a scripting server Language for realising the designed business model or application requirements as per user specification.

5.1 The Server

The Apache server is used to serve the web-pages that are designed and implemented by the front-end development team, this server is hosted within a Personal Computer (PC) and accessible using a local host of the machine. For the presented web application system, the Apache server is run on a Linux operating system. The Linux operating system is specifically chosen for the server to run on because of its Stability, Security and low Cost of operation [8]. As a result a PC running Linux Ubuntu operating system was chosen to host the server. The used Apache server is chosen because it is easy to install and operate, it also provides a secure, efficient HTTP services. Moreover, Apache is a thoroughly tested server partly because of its wide use, serving over a half of all the websites in the World Wide Web (WWW). The Server is open source and it provides free HTTP services.

5.2 The Database

The project requires that data is stored and later on read from or edited depending on the logged on user, to achieve this, there are multiple datasets that need to be considered and implemented. The RDBMS was chosen for implementation due to its advantages over typical Flat file database systems, flat file database systems are limited by the number of tables the database can hold, only one table per flat file can be implemented, it is thus prone to data duplication and data corruption [9, 10].

However RDBMS allow the implementation of multiple tables that are related, this ensures that storage of large amount of data is possible. RDBMS enables organising the large amount of data and defining the relationship between the tables based on the defined business model. MySQL RDBMS was thus chosen because it is an open source database that is well documented and widely used[11]. PHPMyadmin was used in order to provide an easy to use interface for the database design. Two databases were designed and implemented for the student mark/record system. In order to design these databases, it is important to analyse the data to be stored and how the data will be used.

5.2.1 Database Design

The multiple data sets to be stored in the databases are as a result of:

- Users that include the course coordinator, student and school administrator who need to be able to register themselves. This implies that the database should store login details for all the users, additionally there should be information to distinguish between these users. Due to this business model or requirement, a database for users was created, it was created as a separate database considering possible future login requirements for future applications.
  As shown in the figure below, the database has a user table that has the Username, Password and Domain as columns. The three columns in this table are necessary to identify the user logging in. The domain column is essential in defining the role of a user logged in, hence based on this column, a relevant page is redirected to.

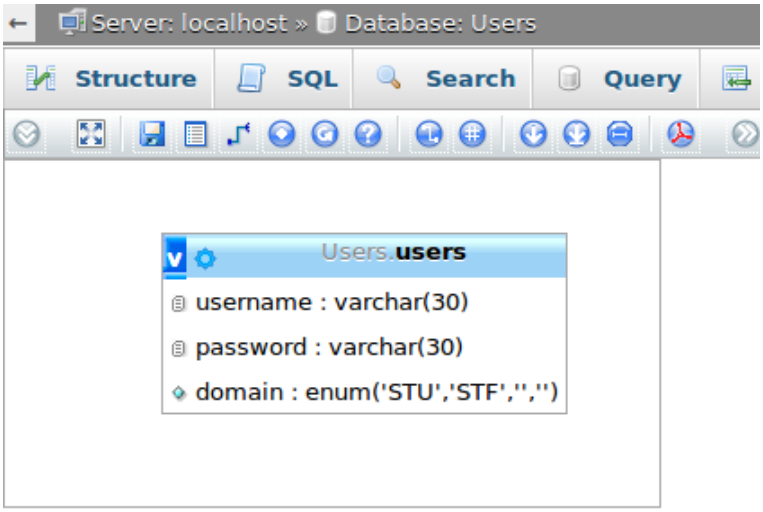


Figure 12 : Users database and table

The possible improvement in this user database design would be storing the password as an encrypted password rather than storing it as is, this would be done to improve system security considering the confidentiality of the information stored within the students record/marks system.

- The student information, individual course information, course components and their percentage contribution form part of a student record database system.

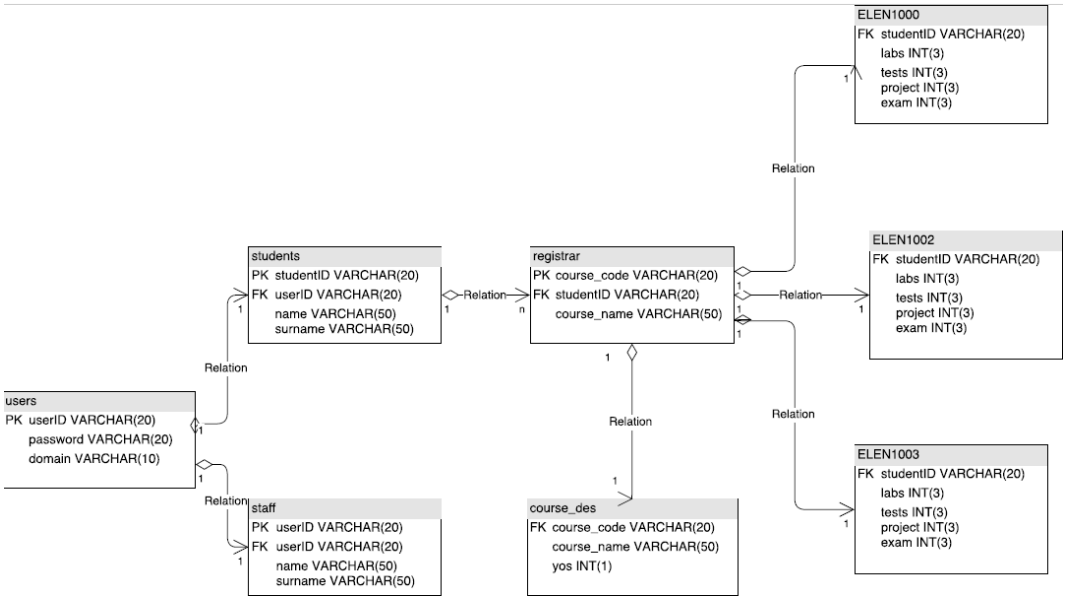


Figure 13 : Student record Database design

Students :
This table is meant to store each individual student's information that include the student's name, surname and studentID which is a unique key that can be used to identify a student. This table has an auto incremented id as a primary key.

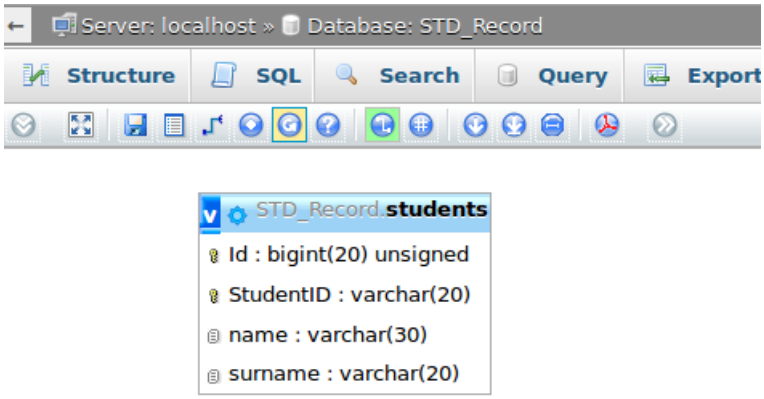


Figure 14 : Students

course_description : This table contains the course code and course name, as the name suggests, it provides a description of each individual course in details. The course code is related to the course name in this table, each course code is unique and thus defined as such in the database.
registrar : This table is a logic table which is populated based on student registration business logic. This is specifically for student registration to a particular course, it constitute of a studentID and course_code. On registration, the studentID is paired to a specific course_code registered and the table's primary key is the id which is an auto incremented value.

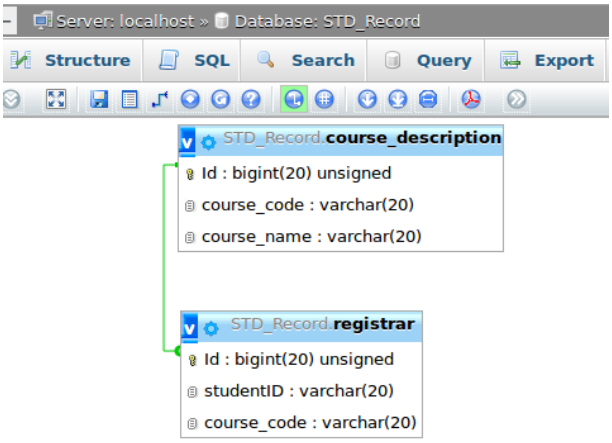


Figure 15 : Course description and Registration

staff : The staff table has a userID, this a unique variable, a staff name and surname as well as courses the lecture coordinates. An assumption that each individual lecture can not coordinate more than four courses was made to simplify the application due to implementation time constraint. However, an improvement in future versions would not make that assumption, instead lectures will be able to coordinate a non constant defined number of courses unless the business logic specifically require this to be the case.

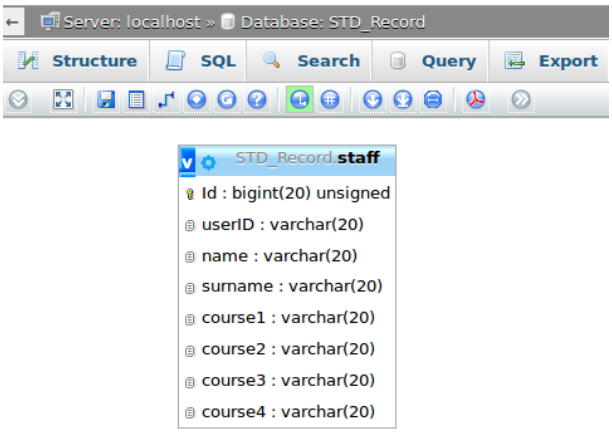


Figure 16 : Staff table

courses : In order to simplify the problem due to time constraints, each course is designed to have its own individual table. This implementation however repeats information and does not take advantage of RDBMS special features that involve linking tables. This table would be improved in future versions of the application ensuring that there is no such a repetition which is redundant and limiting to the use of data stored.
This design was opted for in order to get a working prototype that can be demonstrated with ease. As the system grows, the design would make queering data a difficulty, and it would take more unnecessary time to query the database, shown below is a table example for an individual course
This table contains all course information that include course components and the marks to each of the components. Figure below shows the design for this table.

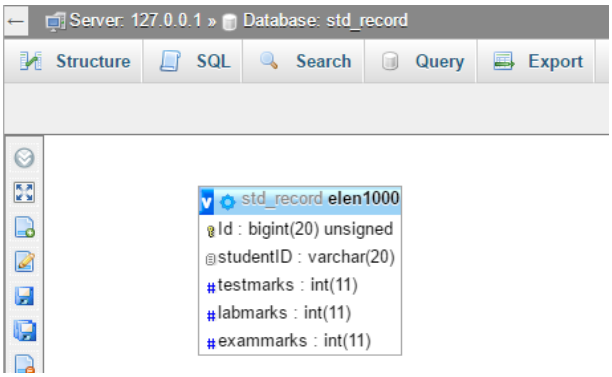


Figure 17 : course table

5.3 The application(Business model)

PHP is selected as a server scripting language. This language is chosen because of its simplicity, it is also a well documented programming language that is widely supported for web application development.

Interacting: the client tries logging in:

- Client enters credentials, these credentials get validated onsubmit.

- If the credentials are correct: a PHP script is used to determine if the user is a student or staff member.
- If the user is a student then the student will only have certain privileges such as reading from the database only.
- If the user is an administrator or a staff member then they may be allowed to have different privileges to edit the database: such as alter student results and add new students onto the database.

6. Scrum Agile Method

Scrum is a very quick and easy process which is widely used for managing and controlling software development projects in a rapidly changing environment. The aim of following the scrum methodology during a software development life cycle is to improve communication, maximise productivity, maximise cooperation amongst the team members and also controlling the chaos of conflicting interests. Scrum is a team based process and requires the co-operation of every member of the team [12].

Each member of scrum needs to be a part in one of the three scrum roles namely the product owner, team member and a scrum master [12].

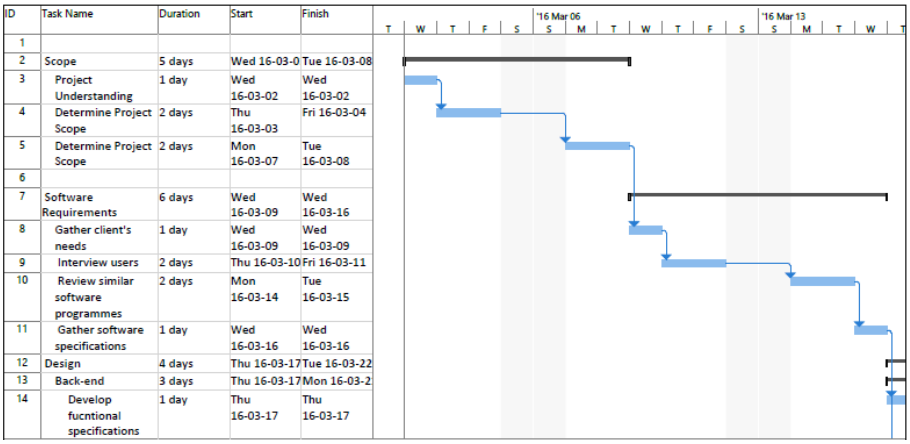
- **Product Owner**
  The product owner represents the stakeholders, customers and users. The product owner gives the goals of the project and the scrum team prioritises these as priorities of the project. The prioritized features will be selected and recorded in a product backlog. The Owner of the products needs to make sure that the project progresses and the project requirements are met. The project owner acts as a middle man between stakeholders and developers, he/she is also responsible of making plans and also announcing the release dates [12].
- **Team Members** The team members are the developers of the software. The team must be organised and cross functional. The work to be done will be divided amongst the team members and also encourages the team to improve [12].
- **Scrum Master**
  The scrum master solves all the problems and challenges that may be encountered during the course of the project [12].

In this project the Product owner is the lecturer, Etutu Otoo. He was responsible for issuing the projects requirements. The progress of the project was monitored by the product owner as submissions were made during the course of the project to monitor the success of the project.

The Team members are the developers mentioned in the stakeholder diagram in section 1. In this project, the scrum master for the project was unanimously chosen to be the scrum master.

6.1 Sprint Planning

6.1.1 Gantt chart



ID	Task Name	Duration	Start	Finish
1				
2	Scope	5 days	Wed 16-03-0	Tue 16-03-08
3	Project Understanding	1 day	Wed 16-03-02	Wed 16-03-02
4	Determine Project Scope	2 days	Thu 16-03-03	Fri 16-03-04
5	Determine Project Scope	2 days	Mon 16-03-07	Tue 16-03-08
6				
7	Software Requirements	6 days	Wed 16-03-09	Wed 16-03-16
8	Gather client's needs	1 day	Wed 16-03-09	Wed 16-03-09
9	Interview users	2 days	Thu 16-03-10	Fri 16-03-11
10	Review similar software programmes	2 days	Mon 16-03-14	Tue 16-03-15
11	Gather software specifications	1 day	Wed 16-03-16	Wed 16-03-16
12	Design	4 days	Thu 16-03-17	Tue 16-03-22
13	Back-end	3 days	Thu 16-03-17	Mon 16-03-2
14	Develop fucntional specifications	1 day	Thu 16-03-17	Thu 16-03-17

Figure 18 : Gantt chart part 1

ID	Task Name	Duration	Start	Finish
15	Review Functional Specification	1 day	Fri 16-03-18	Fri 16-03-18
16	Develop Unit Test using specifications	1 day	Mon 16-03-21	Mon 16-03-21
17	Front-end	4 days	Thu 16-03-17	Tue 16-03-22
18	Develop fucntional specifications	3 days	Thu 16-03-17	Mon 16-03-21
19	Review Functional Specification	1 day	Tue 16-03-22	Tue 16-03-22
20	Develop Unit Test using specifications	2 days	Mon 16-03-21	Tue 16-03-22
21	Intergration	15 days	Wed 16-03-2	Tue 16-04-12
22	Combine Back-end and Front-end	2 days	Wed 16-03-23	Thu 16-03-24
23	Perform tests to the overall product	2 days	Fri 16-03-25	Mon 16-03-28
24				
25	Documentation	11 days	Tue 16-03-29	Tue 16-04-12
26	Back-end	4 days	Tue 16-04-05	Fri 16-04-08

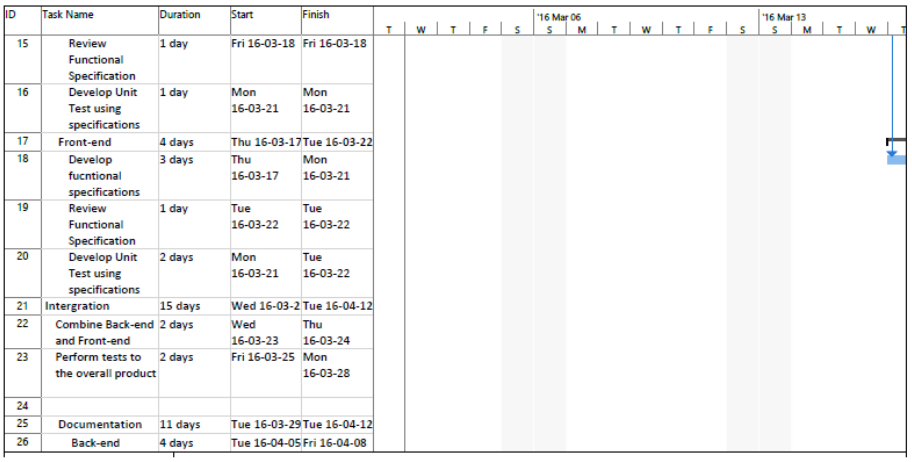


Figure 19 : Gantt chart part 2

ID	Task Name	Duration	Start	Finish
27	Front-end	4 days	Tue 16-04-05	Fri 16-04-08
28	main document	3 days	Thu 16-04-07	Sun 16-04-10
29				

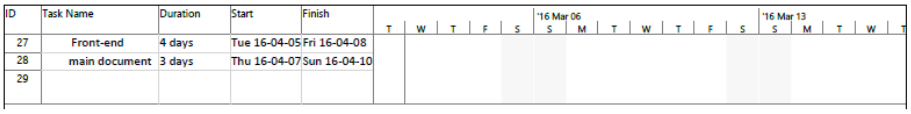


Figure 20 : Gantt chart part 3

Figure 21 : Gantt chart part 4

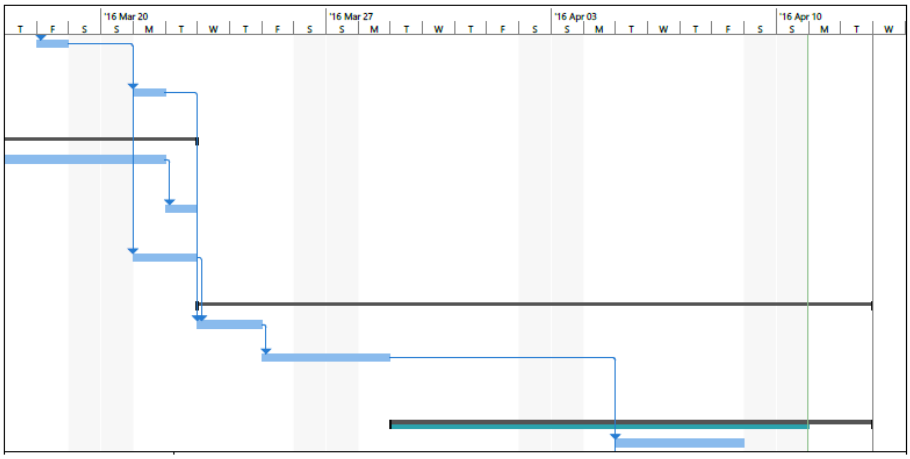


Figure 22 : Gantt chart part 5

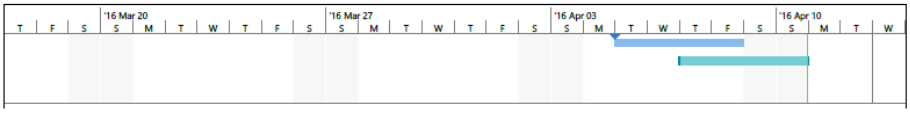


Figure 23 : Gantt chart part 6

6.1.2 Sprint Retrospective Sprint retrospective serves to reflect on the current progress of the project to identify areas that need improvement and to reinforce good habits.

Overall Sprint Retrospection

Well done	**future improvement**
Co-ordination	Scrum meetings were effective but took longer than planned quite often
Effective communication	Under-estimation of tasks
Punctuality	Uneven work distribution

The figure below shows that the estimated time was not followed precisely as highlighted in the improvement section above.

Burn down chart

Day	Burned down		Balance		Daily
	Planned	Actual	Planned	Actual	Completed
0			300	300	
1	4	5	296	295	5
2	5	5	291	290	5
3	5	6	286	284	6
4	4	4	282	280	4
5	5	6	277	274	6
6	6	6	271	268	6
7	5	5	266	263	5
8	8	8	258	255	8
9	6	6	252	249	6
10	5	5	247	244	5
11	6	7	241	237	7
12	8	8	233	229	8
13	7	7	226	222	7
14	8	9	218	213	9
15	8	8	210	205	8
16	8	6	202	199	6
17	4	4	198	195	4
18	5	5	193	190	5
19	6	6	187	184	6
20	4	5	183	179	5
21	5	5	178	174	5
22	6	6	172	168	6
23	5	5	167	163	5
24	7	8	160	155	8
25	5	6	155	149	6
26	6	6	149	143	6
27	8	8	141	135	8
28	6	6	135	129	6
29	6	6	129	123	6
30	5	7	124	116	7
31	8	8	116	108	8
32	6	6	110	102	6
33	5	7	105	95	7
34	6	6	99	89	6
35	6	6	93	67	6
36	8	8	85	59	8
37	6	8	79	53	8
38	7	8	72	45	8
39	6	6	66	35	6
40	8	8	58	27	8
41	10	10	48	17	10
42	8	8	40	13	8
43	10	10	30	6	10
44	7	4	23	-2	4
45	7	7	16	-10	7
46	8	8	8	#N/A	8
47	8	8	0	#N/A	8

Figure 24 : Burn down chart data

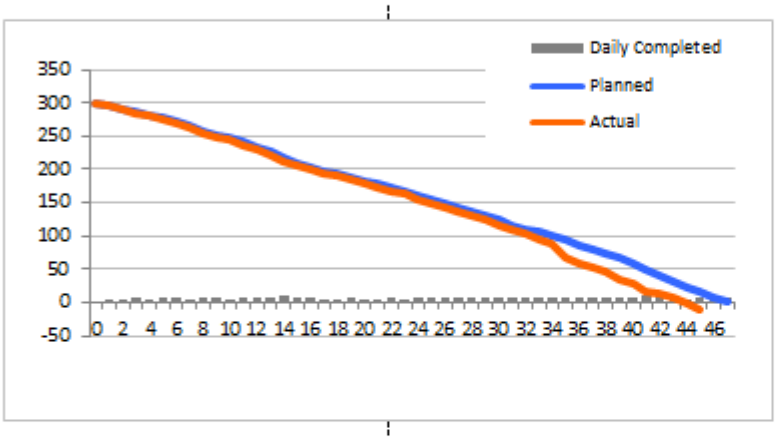


Figure 25 : Burn down chart

User	Feature	Priority	Status
Course Co-ordinator	Register /login	High	done
	Adds assessment methods	High	not done
	Adds assessment weights	High	not done
	Enters student marks for each assessment	High	not done
	Display table of students and their marks	High	done
	Statistics of the performance of the students	Medium	not done
	The projected pass rate based on the assessment marks	Low	not done
The School Admin	Register /login	High	done
	Display or print out the table of students and their marks	High	not done
	Generate a summary statistics of the performance of the students.	Low	not done
	Comparative chart of the assessment marks of selected courses	Low	not done
	Histogram of assessment marks of all courses taken by a specific student.	High	not done
	Register a student into database	High	done
	Any recorded offences (e.g., plagiarism) for a student.	High	not done
	The performances in the same course across different years may be compared.	Low	not done
The student	Register /login	High	done
	Display assessment mark for a course and the statistics for that assessment	High	done
	Display assessment marks for all courses	High	done
	Performance goals needed to pass the course	Low	not done

Figure 26 : Product backlog

7. Conclusion

The student record/marks system is presented in this document. With the aim of developing and implementing the system, project requirements are established and the design of the system is separated into the front-end and back-end subsystems. The scrum agile method is chosen as the (SDLC) methodology to quickly deliver a working prototype. As shown in the product backlog not all features were successfully implemented due to time constrains and their complexity.

REFERENCES

[1] R. H. Technology. "6 Basic SDLC Methodologies: Which One is Best?" https://www.roberthalf.com/technology/blog/6-basic-sdlc-methodologies-the-pros-and-cons, 2014. [Online; accessed 10-March-2016].

[2] VERSIONONE. "Understanding Scrum Project Management." https://www.versionone.com/scrum-project-management/, 2016. [Online; accessed 10-March-2016].

[3] E. Otoo. "Expanded Project Description." https://cle.wits.ac.za/access/content/group/ELEN4009_SoftwareEngineering2016/Project%20Descriptions/EnhancedProjDesc.pdf, 2016. [Online; accessed 22-March-2016].

[4] Apache. "Apache - HTTP Server Project." https://httpd.apache.org, 2016. [Online; accessed 9-March-2016].

[5] WhatIs.com. "Web server." http://whatis.techtarget.com/definition/Web-server, 2016. [Online; accessed 9-March-2016].

[6] G. Palaniswamy. "Two-Tier and Three-Tier Architecture with example." http://www.c-sharpcorner.com/UploadFile/gowth/two-tier-and-three-tier-architecture-with-example/, 2010. [Online; accessed 2-March-2016].

[7] techopedia. "Two-Tier Architecture." https://www.techopedia.com/definition/467/two-tier-architecture, 2016. [Online; accessed 2-March-2016].

[8] PCWorld. "Five Reasons Linux Beats Windows for Servers." http://www.pcworld.com/article/204423/why_linux_beats_windows_for_servers.html, 2015. [Online; accessed 24-February-2016].

[9] databasedev. "Flat File Database Design vs. Relational Database Design." http://www.databasedev.co.uk/flatfile-vs-rdbms.html, 2015. [Online; accessed 10-April-2016].

[10] databasedev. "Advantage & disadvantage of relational database." http://smhamiri.blogspot.co.za/2013/04/advantage-disadvantage-of-relational.html, 2013. [Online; accessed 10-April-2016].

[11] MySQL. "Why MySQL?" https://www.mysql.com/why-mysql/, 2015. [Online; accessed 24-February-2016].

[12] R. Stephens. *Software Engineering*, chap. 3, pp. 327–328. Crosspoint Boulevard: John Wiley and Sons Inc, first ed.