THE WEEKEND PLATONIX

Rakhosi Lerato – 2016164382

Tau Tsephe – 2016410132

Leputla Mofehlo (Project Leader) – 2015304948

Ngobese Lungisa – 2015240092

Mosia Serame – 2014175117

Mokoena Katleho – 20163770975

Maine Gopolang – 2016372720

DOCUMENTATION

CSIS3724

Table of contents

Total time spend on the project

Names of Students	Hours
Lungisa	64 + 2
Katleho	67 + 2
Lerato	66 + 4
Serame	63.5 + 2
Tau	61 + 3
Gopolang	62 + 2
Mofehlo	68.5 + 2

The added hours simply specify the time that the individuals spent on the project after the meetings.

Lungisa

Workflows	Hours
Requirements	15
Analysis	19 + 2 extra hours
Design	5
Implementation	20
Testing	5

Katleho

Workflows	Hours
Requirements	15
Analysis	21
Design	5
Implementation	21 + 2 extra hours
Testing	5

Tau

Workflows	**Hours**
Requirements	15
Analysis	18
Design	5 + 3 extra hours
Implementation	19
Testing	5

Lerato

Workflows	**Hours**
Requirements	15 + 2
Analysis	21 + 2 extra hours
Design	5
Implementation	20
Testing	5

Serame

Workflows	**Hours**
Requirements	15
Analysis	20
Design	5 + 2 extra hours
Implementation	19
Testing	5.5

Mofehlo

Workflows	**Hours**
Requirements	16
Analysis	21
Design	5.5
Implementation	21
Testing	5

Spent additional Hours on complying documentation ➔ 2 Hours

Gopolang

Workflows	Hours
Requirements	13
Analysis	20
Design	4
Implementation	20 + 2 extra hours
Testing	5

Glossary

Student: Refers to a person who is studying at a university or other place of higher education can either be an undergraduate or post graduate.

Lecturer: Refers to a person who gives lectures, especially as an occupation at a university or college of higher education.

Dashboard: Refers to the user interface of an application where users can interact with the items in the menu of the dashboard to meet their needs.

Course: Refers to a unit of teaching that typically lasts one academic term, is led by one or more instructors, and has a fixed roster of students, a course is usually an individual subject.

Timetable: Refers to a table for coordinating these three elements: Students, Lecture Rooms and Time slots other factors include the Module name, and the type of Lecture rooms available

Assessments: Refers to the systematic process of documenting and using empirical data on the knowledge, skill, attitudes, and beliefs to refine programs and improve student learning. Examples of Assessments include Tests, Projects assignments and home works

PeopleSoft: Refers to a database system where lectures can lodge the marks of students after grading the assessments these marks will be used later to calculate the semester mark.

Semester Mark/ Predicate: Refers to the percentage out of a hundred that a student got after all his assignments are combined together. This mark is used to determine whether the student qualifies to write the exam or not.

Goal Mark: Refers to the mark that a student sets at the beginning of each semester for each module the aim is for the student to reach that mark at the end of the semester.

REQUIREMENTS

External Interface requirements

1.1.1. User Interface

Hardware Interface

The application would not require any direct hardware interfaces.

Software Interface

The application will require a software like DBMS to be incorporated.

Communications Interfaces

The application will make use of algorithms which will be inserted in related child classes so it can derive data from the DBMS.

Functional Requirements

Describes what the system should perform.

1.1.1.1. Functional requirement 1

Title 1: Mobile and Desktop application

Description: a student should be able to download the application either through their computer or mobiles. The application has to be able to retain the same layout on different operating systems (Cross platform).

1.1.1.2. Functional requirement 2

Title 2: User login

Description: The student must be able to login to the timetable application. Once access is granted: An interface showing all the modules registered as hyperlinks will appear.

1.1.1.3. Functional requirement 3

Title 3: Viewing the timetable

Description: The student's personal timetable will show (Monday - Monday) subjects, remain fixed throughout the semester. Student will be able to view the table anytime they wish (discard the options of when to view the timetable).

Administrative function

Feature 1: Due dates

The administrator shall be able to set due dates which should reflect on the application. The application will have a link to Assessments, the link shall lead to the Window of Assessments Deadlines: The lecturer will be able to update the dates of the Tests and assignments, which will trigger the reminder system from the student's side to warn them about the deadline they have to meet 3 days prior, an icon with the number of reminders will be featured in the window.

Feature 2: Set reminder

The administrator shall be able to set a reminder to warn students of upcoming assessments. The Application will have an algorithm that derive information from an excel sheet provided by the university examination board, use that to update the dates of the examinations (to both students and lecturers).

The Application

Title 1: Access PeopleSoft

Description: The application will access the existing PeopleSoft marks system and display the results recorded for a student directly to them. The application will use an algorithm that will derive information about the Computer Science Modules of a student from a database management system of PeopleSoft.

1.1.1.4. Functional requirement 2

Title 2: Semester mark computation and goal setting feature

Description: The application will compute the current semester mark and what mark will be needed for every upcoming assessment to reach a goal mark. Each student will be assigned an opportunity to set all the goal marks the first time after going through a module guide and seeing an assessment breakdown per each module taken. The feature will ultimately calculate the semester mark of each student and indicate if they've qualified or not.

Title 3: Accessing the lecturer details

Description: When the student clicks on a module, a timetable, which features venues and times, will appear, followed by the name of the lecturer (Teaching Assistant), office number, contact details, which feature email address and the office phone number. Furthermore, the application needs to make adjustments when it is opened on a mobile device - When the office number is clicked it should lead straight to the call program of the mobile device, an algorithm will be used for this.

Additional Functionalities

Title: Online submissions and Plagiarism Check

Description: There will be online submissions feature for all assignments. The application will further use an algorithm to check for plagiarism among the students ‵assignments. Students will click on the link that will lead them to a window where they will be able to insert the caption of the assessment, dropdown to load a file to be submitted and clicking submitted after the clicked on the submit button, an alert message will be sent to report the submission success.

Reasons: It is reliable to cater for inconveniences such as being able to work and submit the work from remote areas. It also helps to reduce the plagiarism levels among students.

Title: Feedback Utility

Description: The lecturer will be able to give feedback on assessments submitted both physically and electronically this will give student fruitful feedback on which areas of the module they need to improve. There will be a feedback icon, when clicked on, a softcopy of the assessment with areas of corrections and compliments will be downloaded if the assessment was submitted online, otherwise a pdf document referring to the hardcopy written will have jotted down points reflecting on corrections or compliments found on the hardcopy submitted.

Reasons: It helps with measuring the progress of a student, help them to focus on the areas in which they have problems on time and start working on improving them. It clarifies how the marks were allocated, which reduces student complaints and remark levels.

Discarded Functionalities

Title: Prediction of marks

Description: The functionality that entails the prediction of marks and the ultimate outcome (Pass or Fail) if a student continues with a current performance trend. There is really not a need for the utility as it will serve to demotivate Student's final outcome. It makes predictions from an immaterial source (past performance is immaterial).

Added Requirements After the Client's Feedback

Title 1: Online chat with lecture

Description: this is a function that works like the online communication between two people but in this case it will be a communication between lecture and the students, the students will be able to communicate with a lecture at any time. The lecture will be able the see the message only if he/she is online, then lecture will give the student a feedback on the message.

Title 2: Machine Learning

Description: The machine learning is the algorithm that computes the final semester mark of the students based on the test marks and the assignments, and it shows if the students was able to qualify to write the exam.

Example 1: if the students marks or semester mark is below 39% then the system will show red icon with message "Fail".

Example 2: if the students marks or semester mark is (40% to 50%) then the system will show yellow icon with message "Danger".

Example 3: if the students marks or semester mark is (51% to 59%) then the system will show blue icon with message "Safe enough".

Example 4: if the students marks or semester mark is (60% to 100%) then the system will show green icon with message "Completely Safe".

Design/Specification

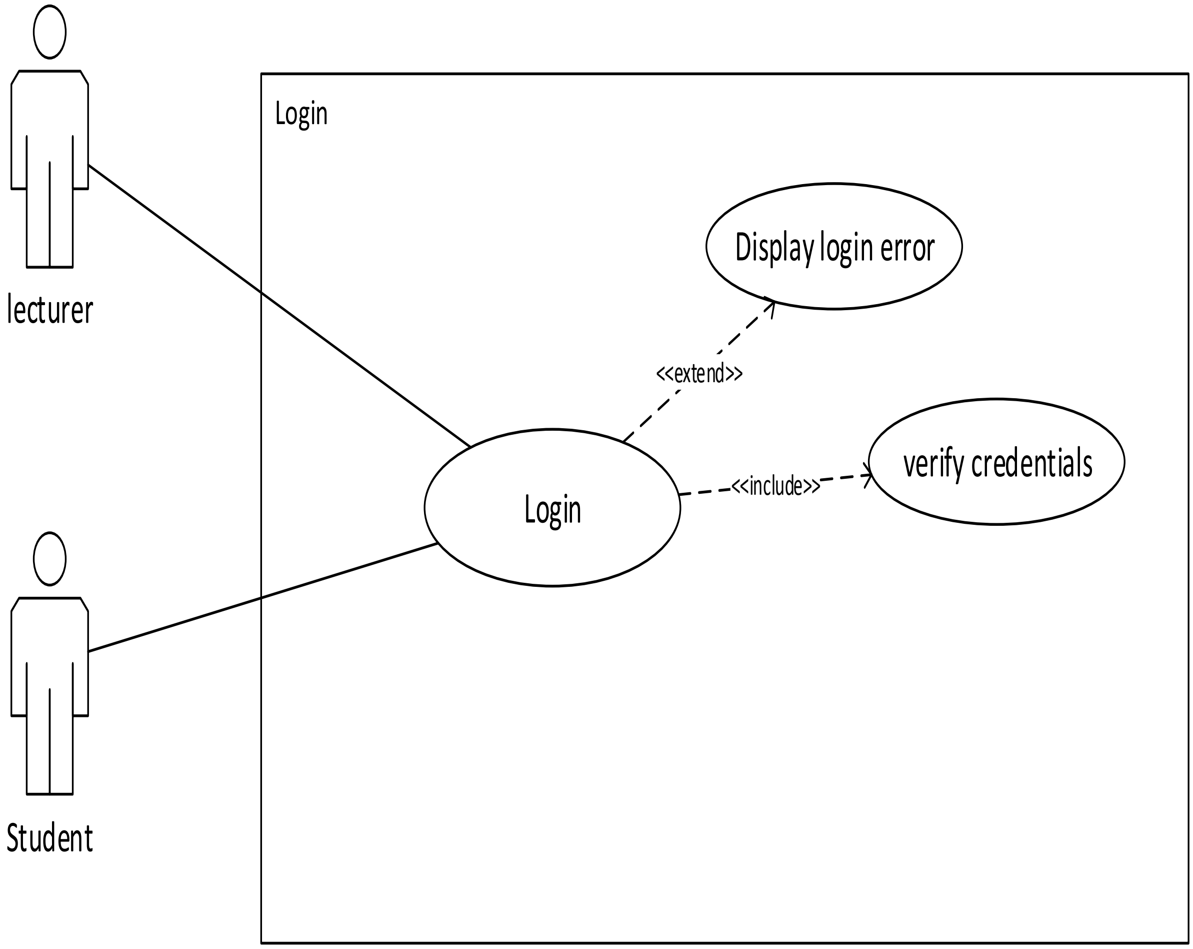
Noun extraction paragraph

A *student* is able to *login* to the application using his/her own credentials and a *dashboard* containing the student's Computer science *modules* and class *timetable* appears. The *application* is usable on both *mobile and desktop* devices. Each module is a link to its *homepage* where all *lecturer's details* and *assessments (tests, assignments and examination)* along with their due dates will be presented. A reminder system is set in place to remind students of upcoming due dates. All assessments information is administered by the *lecturer* with the exception of the examination dates which will be made available by the examination department via an *excel sheet* on the *UFS website*. Student's *results* will be reported directly to them by accessing people soft. A Student is able to set a *goal mark* for each module and the application computes the current semester mark and what mark will be needed to reach the set goal and the goal mark is tracked by a *goal tracker*. A *report* is presented to the lecturer on whether the student will pass or fail based on past performance and current performance as an icon to the student in question. *Student dashboard* will be made available displaying timeline of past and upcoming assessment and student's overall performance. An *online platform* is available where the student and the lecture can communicate, the student can send *queries and questions* and lecturer can receive a notification and once he/she is online can give *feedback* to the student.

Noun extraction

Login, Student, Lecturer, Module, Device, Assessment, Timetable, Test, Mark, Class time, Reminder, Dashboard, forum

UseCases



Login:

Brief Description: Enter your username and password to login into the application.

Step-by-step description:

- Step 1: The user opens dashboard, two textboxes are displayed, one for username and the other for password.
- Step 2: The user inserts username in the designated area.
- Step 3: The user inserts password in the designated area.
- Step 4: Presses the submit button to validate the login credentials.

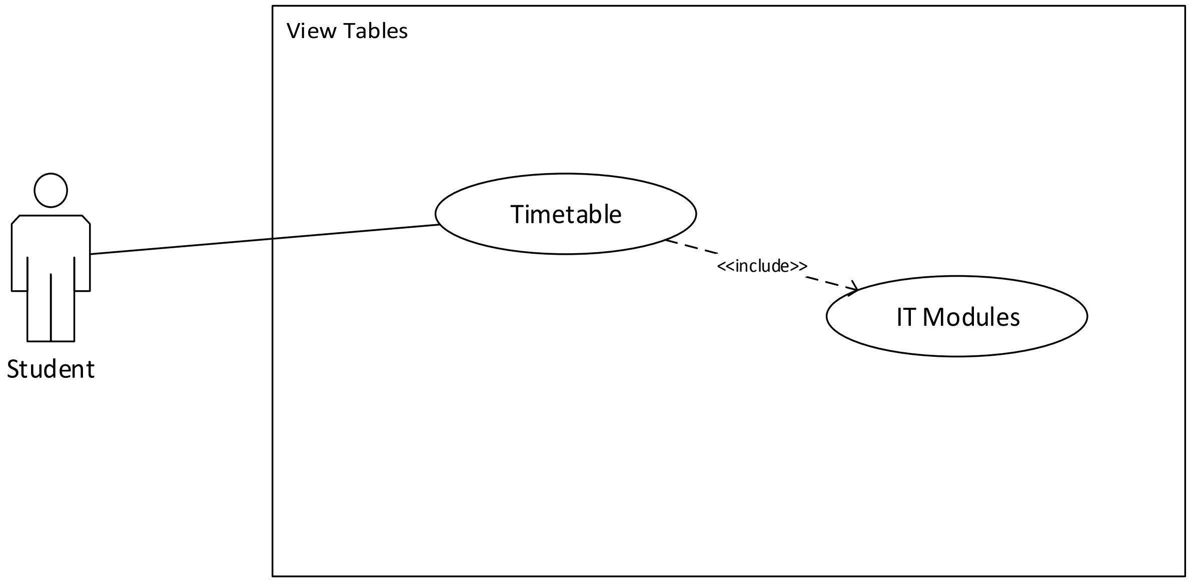
Normal Scenario:

1. Thabo opens dashboard, two textboxes are displayed, one for username and the other for password.

2. Thabo inserts the credentials in the designated areas displayed in the dashboard.
3. He clicks the submit button after entering them, to validate the login request.
4. Access is granted, the home page of the application with various links is displayed.

Worst Scenario:

1. Thabo opens dashboard, two textboxes are displayed, one for username and the other for password.
2. Thabo enters the invalid credentials (expired password or inserting wrong credentials).
3. The error alert message is displayed denoting the denied access.



View Table:

Brief Description: Click the Timetable link to view the Computer Science Modules for the week.

Step-by-step description:

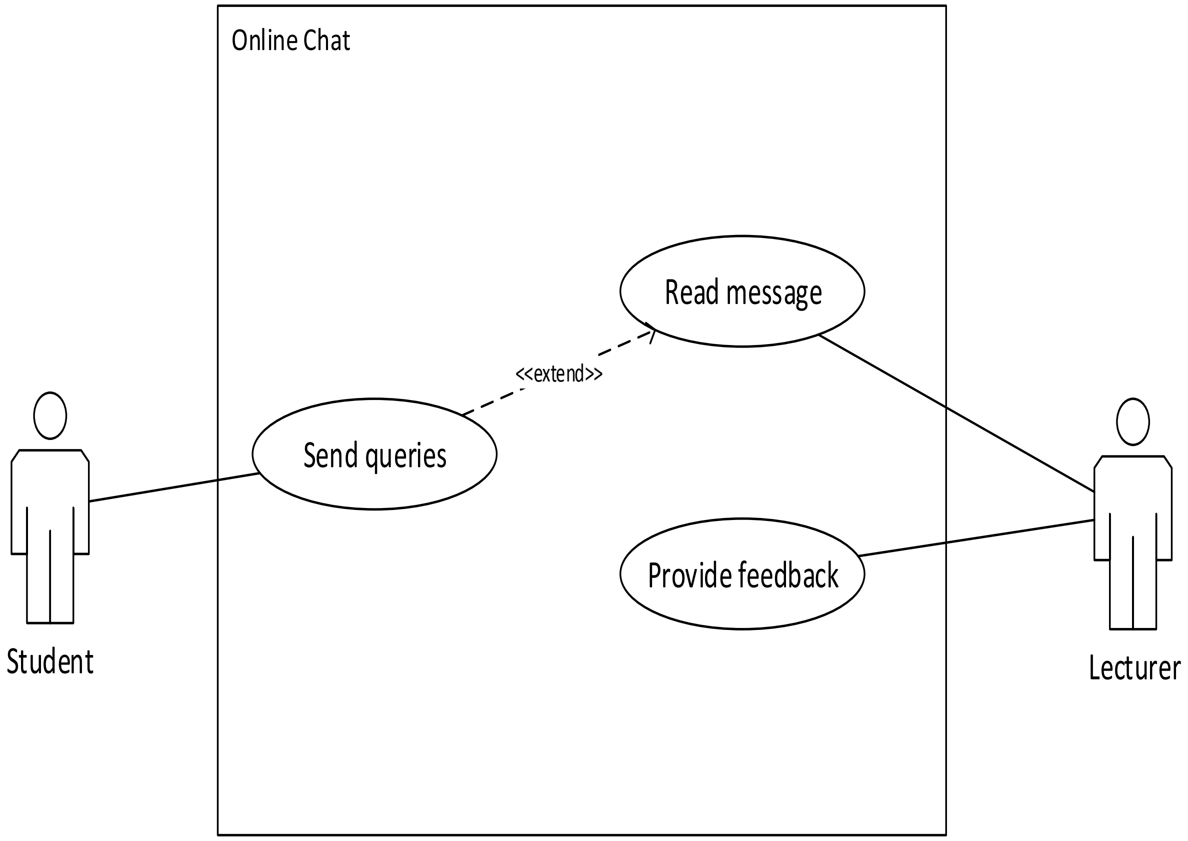
- Step 1: Student clicks on the timetable link.
- Step 2: The system will have a pop up page displaying the timetable.

Normal Scenario:

1. Thabo clicks on the timetable link.
2. The system will have a pop up page displaying the timetable.

Worst Scenario:

1. Thabo clicks on the timetable link.
2. The system does show timetable, due to the pop-ups being blocked depending on the specific browser settings.
3. Thabo must click on the system settings to allow the pop up messages to be displayed.
4. The system will show the timetable.



Online Chat with Lecturer:

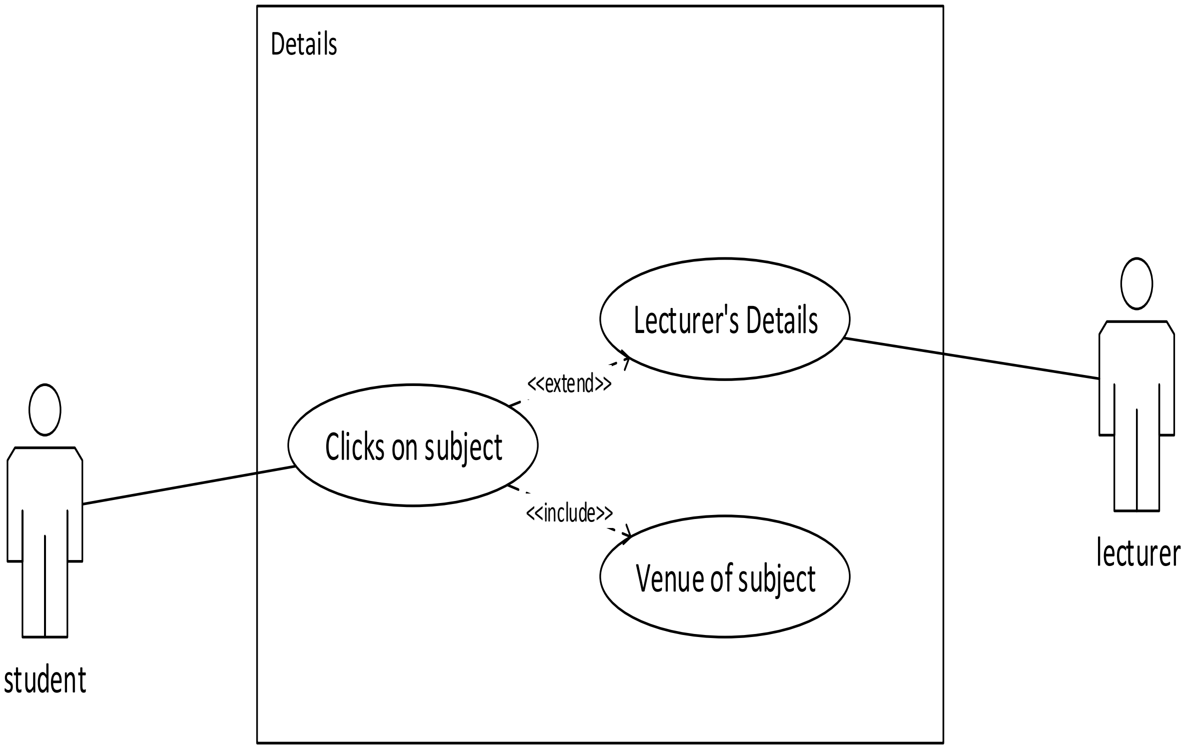
Brief Description: Communication forum between student and lecturer will be interfaced.

Step-by-step description:

- Step 1: Student clicks on a specific module link.
- Step 2: The system will display the homepage of the module, with the communication forum link listed under resources.
- Step 3: The student clicks on the communication forum link
- Step 4: The interface of the communication forum will be displayed.

Normal Scenario:

1. Thabo clicks on a specific module link.
2. The system will display the homepage of the module, with the communication forum link listed under resources.
3. Thabo clicks on the communication forum link to start a discussion with the lecturer
4. The interface of the communication forum will be displayed.



Lecturer Details:

Brief Description: Display lecturer details

Step-by-step description:

- Step 1: Student clicks on a specific module link.
- Step 2: The system will display the homepage of the module, with the link course information listed under resources.
- Step 3: The student clicks on the course information link
- Step 4: The page that displays lecturer details and the module guide will be displayed.

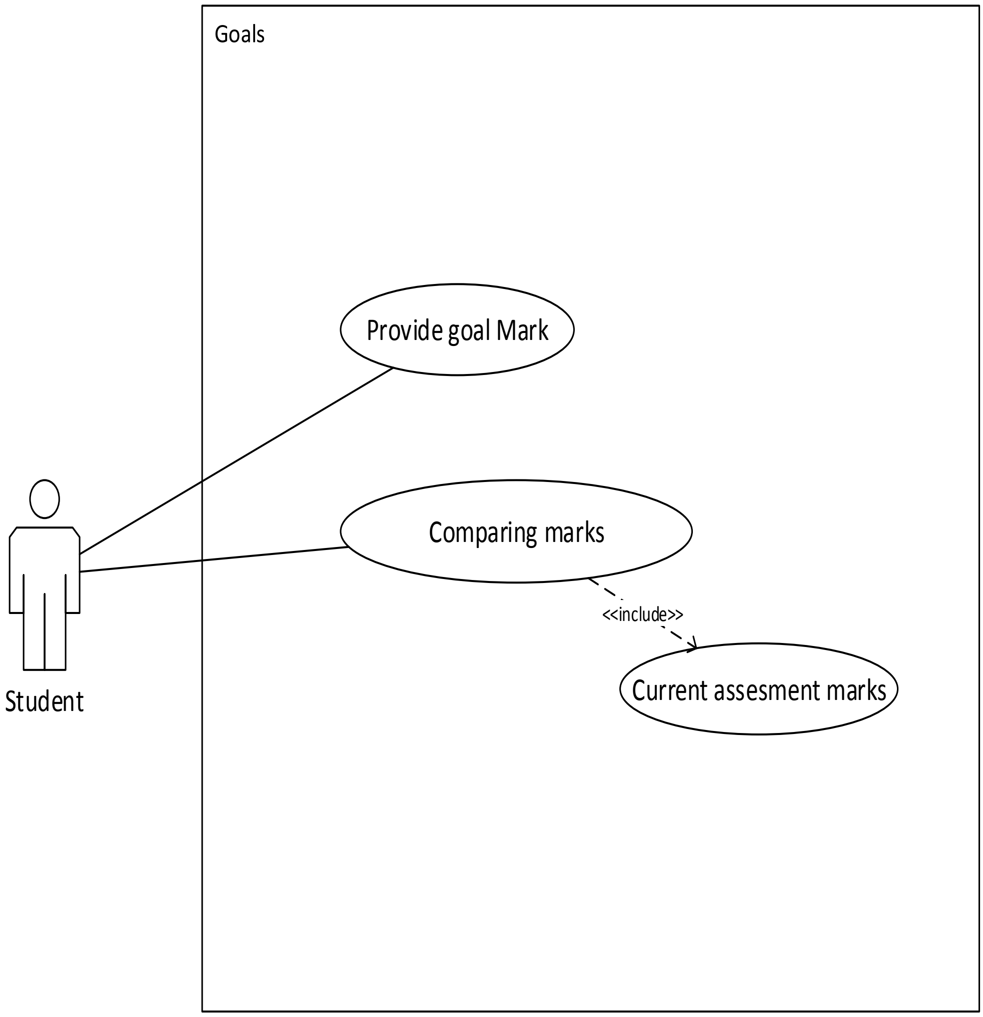
Normal Scenario:

1. Thabo clicks on a specific module link.

2. The system will display the homepage of the module, with the link course information listed under resources.
3. Thabo clicks on the course information link
4. The system displays lecturer details page.

Worst case Scenario:

1. Thabo clicks on a specific module link.
2. The system will display the homepage of the module, with the link course information listed under resources.
3. Thabo clicks on the course information link
4. The system displays the lecturer details page but the lecturer did not upload his/her details or the systems displays the previous lecturer's details.



Set goal

Brief description

Student is presented with a dashboard that allows them to set goal for computer science modules.

Step by step description

1. A student clicks on the module link
2. They are a presented with a link "set goal".
3. The user is presented with a textbox where he/she enters their goal mark
4. The system saves the goal mark on the computer science server to use it later in semester.

5. The systems than returns to the homepage.

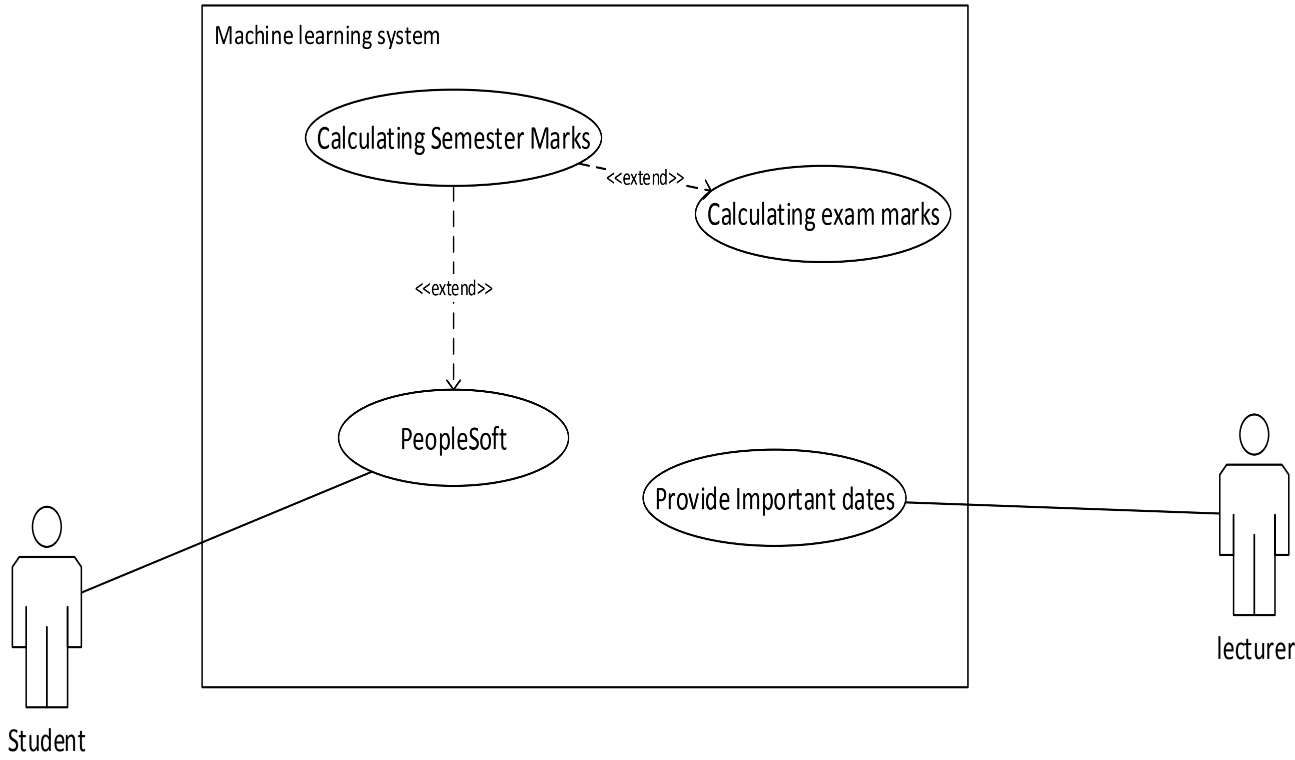
Scenarios

Best Case scenario

1. Thabo clicks on the module link.
2. Thabo is presented with the set goal link and Thabo clicks on it.
3. Thabo is presented with a textbox where 0he enters a goal mark.
4. Thabo clicks on save.
5. The system allows Thabo to save the goal mark on the computer science server

Worst case scenario

1. Thabo clicks on the module link.
2. Thabo is presented with the set goal link and Thabo clicks on it.
3. Thabo is presented with a textbox and he enters a string instead of a double
4. Thabo clicks on save and the system identifies that error and it returns a red text stating that Thabo should write a number instead of a text.
5. The systems clear the textbox and returns the cursor to the textbox.
6. Systems allows Thabo 3 tries to enter the wrong data type after it returns him to the homepage dashboard.



Machine learning

Brief Description

The system checks based on assessment calculation whether the student will make predicate or not.

Step by step Description

1. On Predicate day the system is allowed access to the student's marks so it can perform calculations
2. The system uses the machine learning algorithm to perform calculations
3. Based on the marks that system has access to from student dashboard
4. The System calculates the marks and shows the following cases base on final mark of the student

   4.1 If the student marks are between 0-39 a red icon indicating a fail will appear next to module link

4.2 If the student marks are between 40-50 a yellow icon indicating danger to the student will appear next to module link

4.3 If the student marks are between 51-59 a blue icon indicating the student is safe enough will appear next to module link

4.4 If the student marks are between 60-100 green icon indicating the student is completely safe enough will appear next to module link

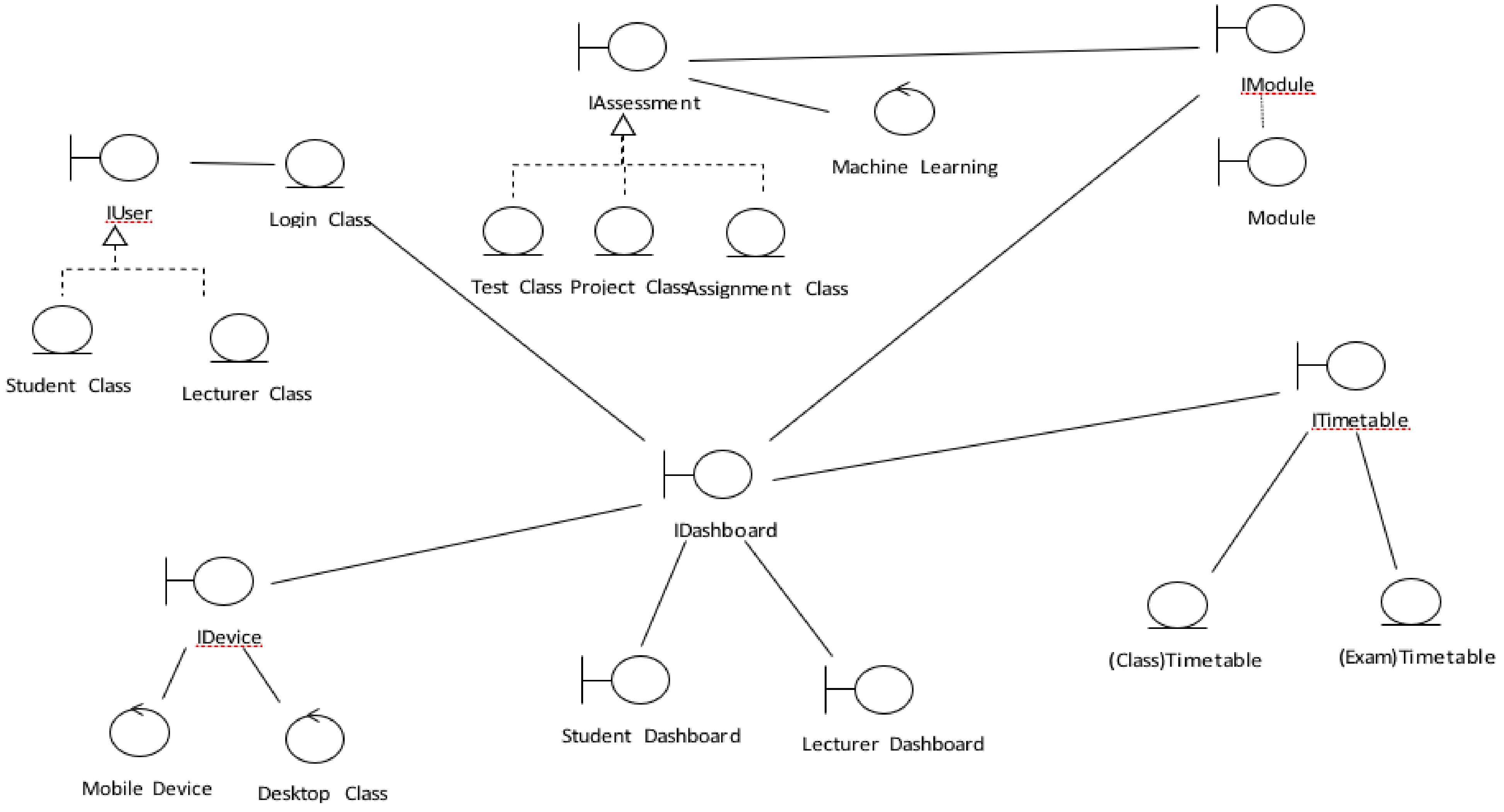
Best case scenario

1. The system gets Thabo’s marks between (0 – 100) from the Thabo’s dashboard
2. The system gets Thabo’s System engineering marks
3. The system uses machine learning to calculate Thabo’s final semester mark
4. After the calculation Thabo’s mark is 75% so he doesn’t make predicate and the system shows a green next to the system engineering module

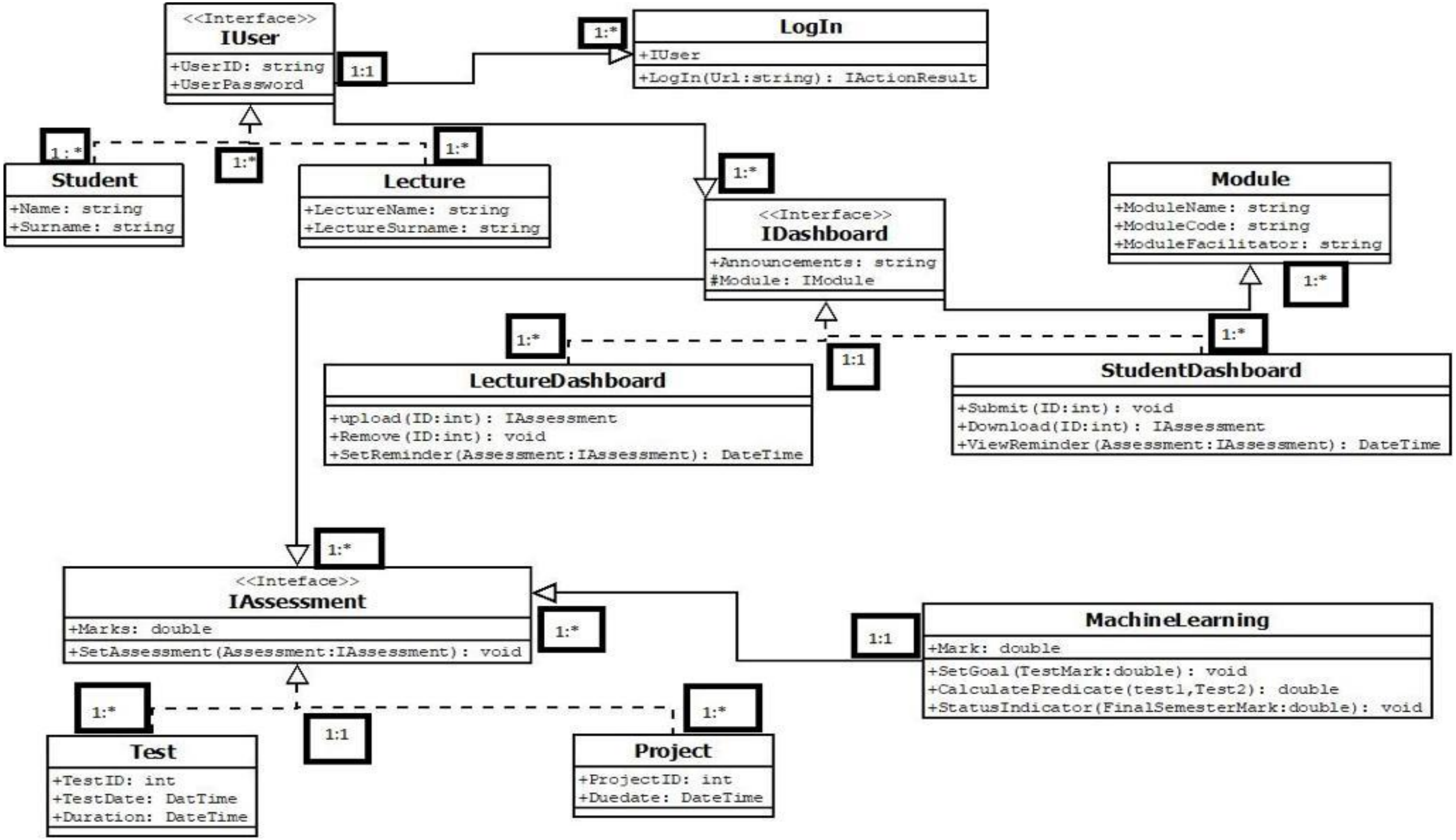
Worst Case scenario

1. The system gets Thabo’s System engineering marks
2. The system uses machine learning to calculate Thabo’s final semester mark
3. After the system does calculations Thabo’s marks are above 100
4. The marks bounce back and Thabo ends up not getting a predicate mark.

Class Diagram

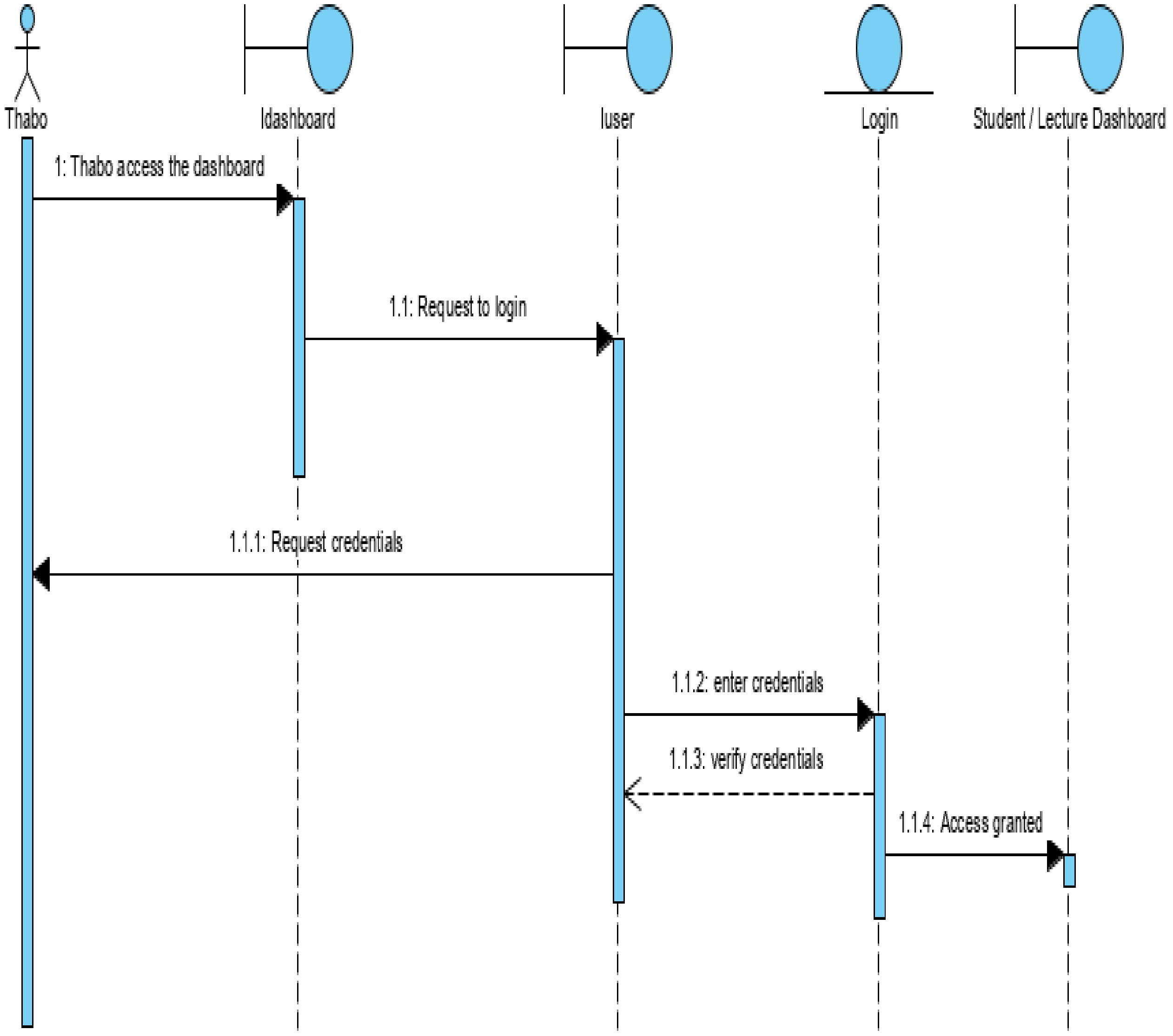


UML diagram

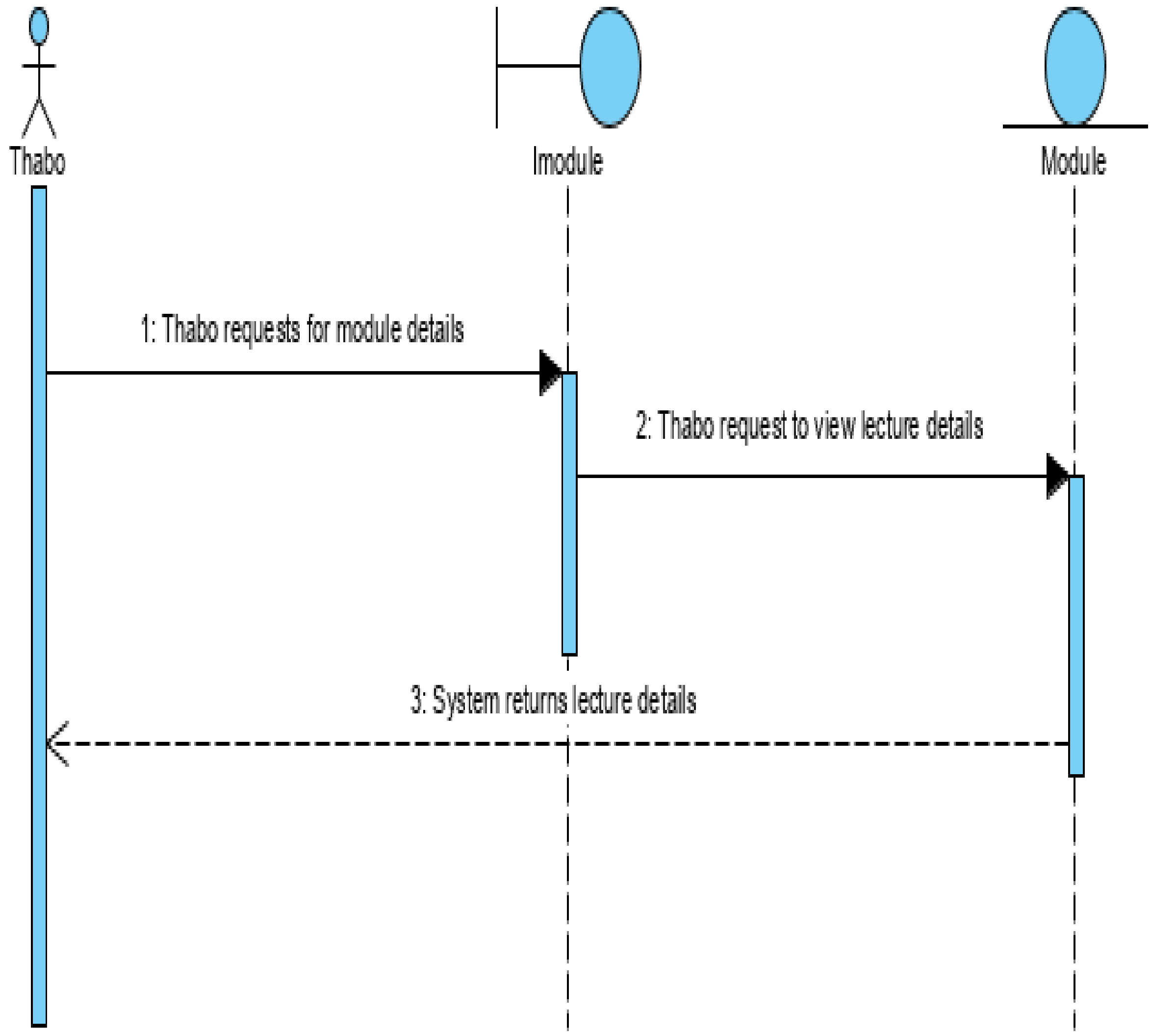

<<Interface>>
IUser
+UserID: string
+UserPassword
1:1
1:*
LogIn
+IUser
+LogIn(Url:string): IActionResult
1: *
1:*
1:*
Student
+Name: string
+Surname: string
Lecture
+LectureName: string
+LectureSurname: string
1:*
<<Interface>>
IDashboard
+Announcements: string
#Module: IModule
Module
+ModuleName: string
+ModuleCode: string
+ModuleFacilitator: string
1:*
1:*
1:*
1:1
LectureDashboard
+upload(ID:int): IAssessment
+Remove(ID:int): void
+SetReminder(Assessment:IAssessment): DateTime
StudentDashboard
+Submit(ID:int): void
+Download(ID:int): IAssessment
+ViewReminder(Assessment:IAssessment): DateTime
1:*
<<Inteface>>
IAssessment
+Marks: double
+SetAssessment(Assessment:IAssessment): void
1:*
1:1
MachineLearning
+Mark: double
+SetGoal(TestMark:double): void
+CalculatePredicate(test1,Test2): double
+StatusIndicator(FinalSemesterMark:double): void
1:*
1:1
1:*
Test
+TestID: int
+TestDate: DatTime
+Duration: DateTime
Project
+ProjectID: int
+Duedate: DateTime

SEQUENCE DIAGRAMS

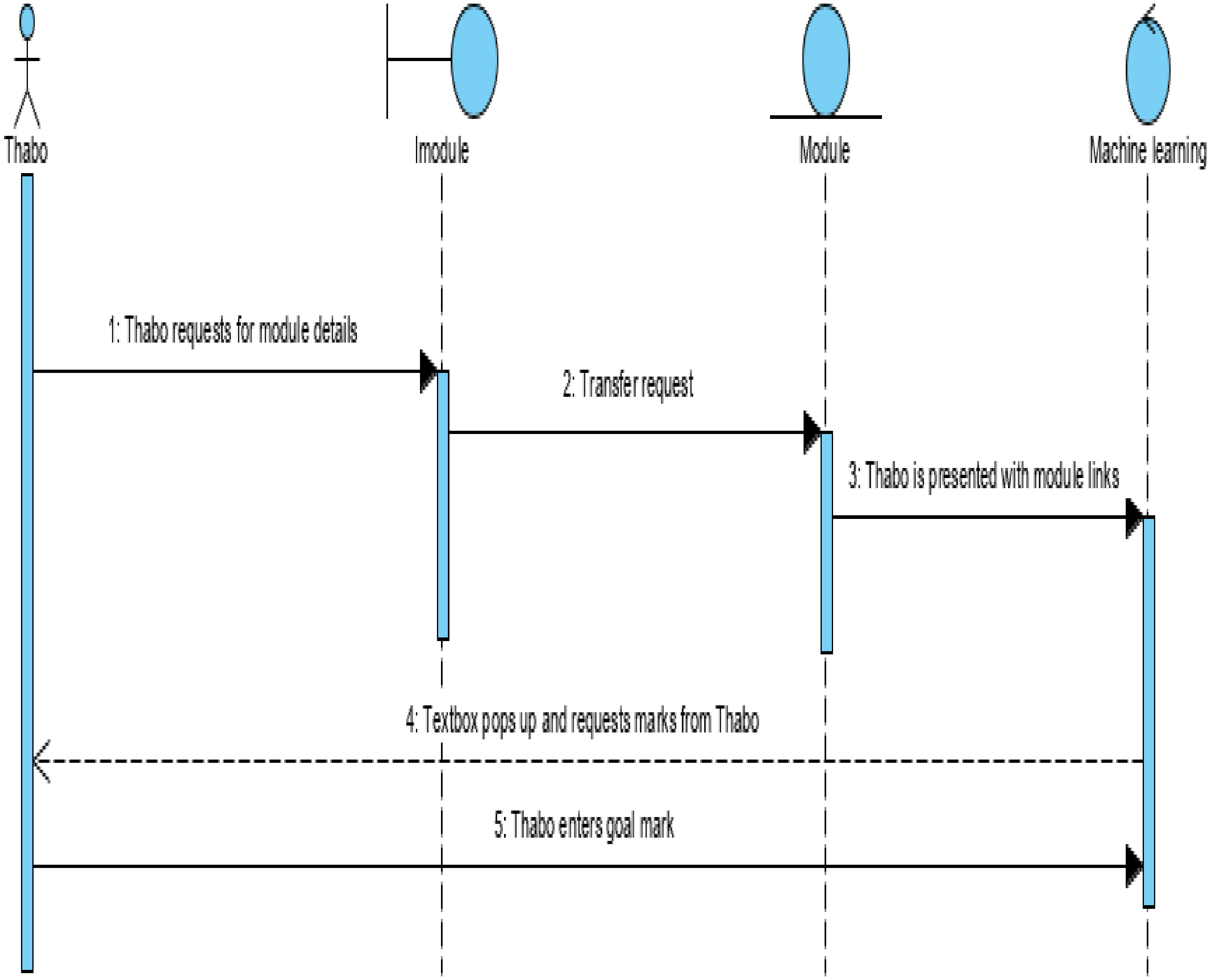
Login


Thabo
Idashboard
Iuser
Login
Student / Lecture Dashboard
1: Thabo access the dashboard
1.1: Request to login
1.1.1: Request credentials
1.1.2: enter credentials
1.1.3: verify credentials
1.1.4: Access granted

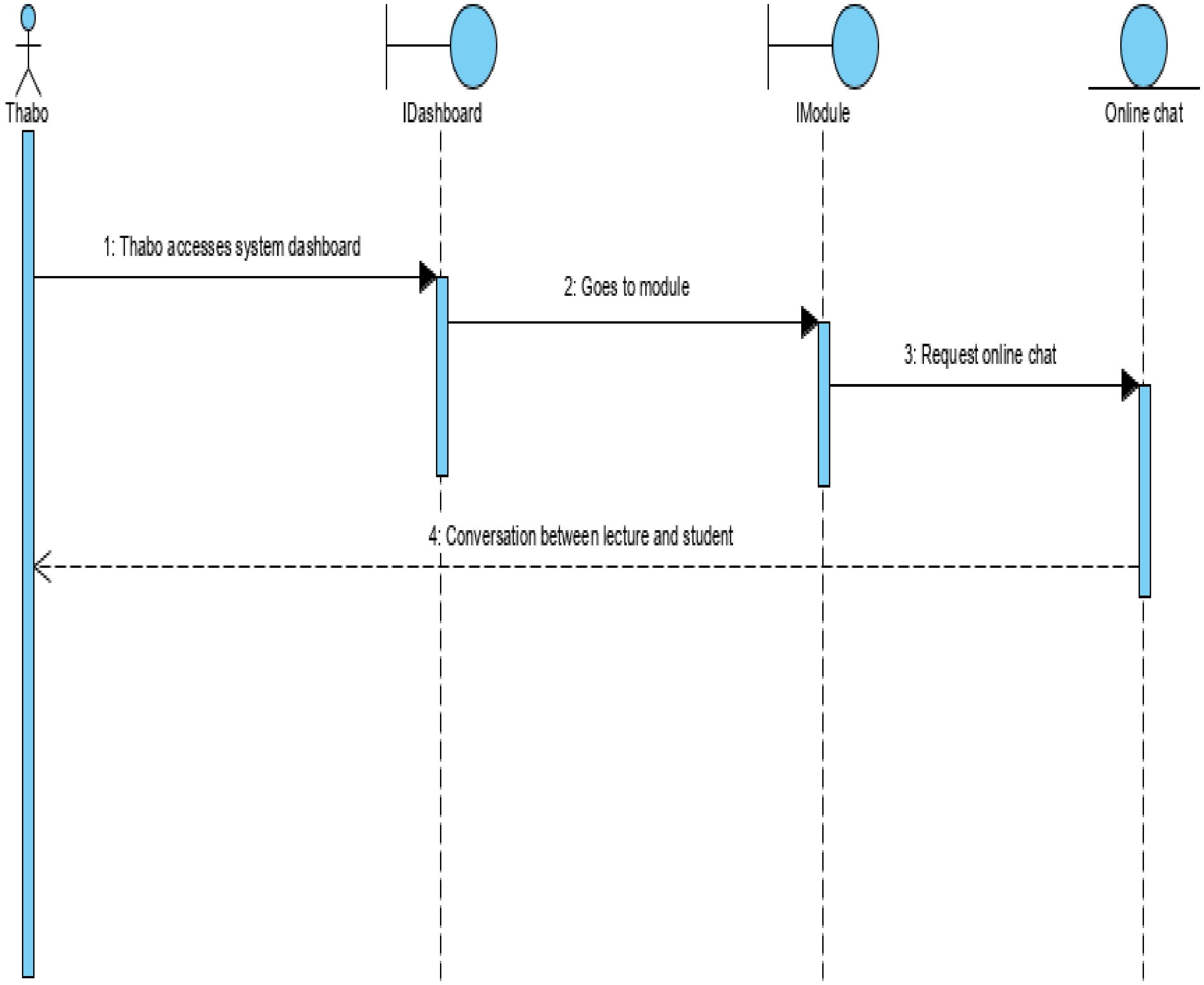
Lecture Details


Thabo
Imodule
Module
1: Thabo requests for module details
2: Thabo request to view lecture details
3: System returns lecture details

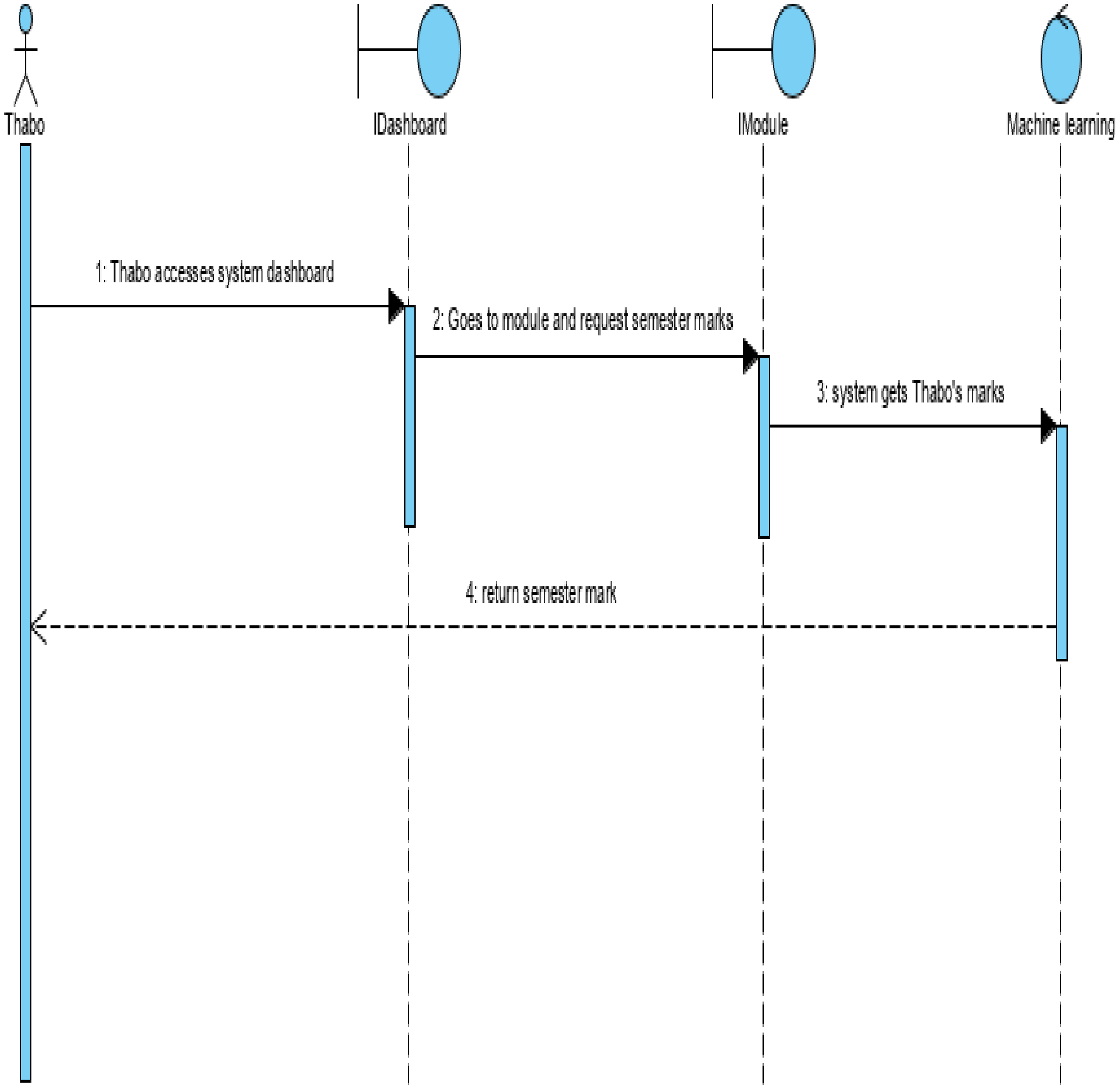
Set goal


Thabo
Imodule
Module
Machine learning
1: Thabo requests for module details
2: Transfer request
3: Thabo is presented with module links
4: Textbox pops up and requests marks from Thabo
5: Thabo enters goal mark

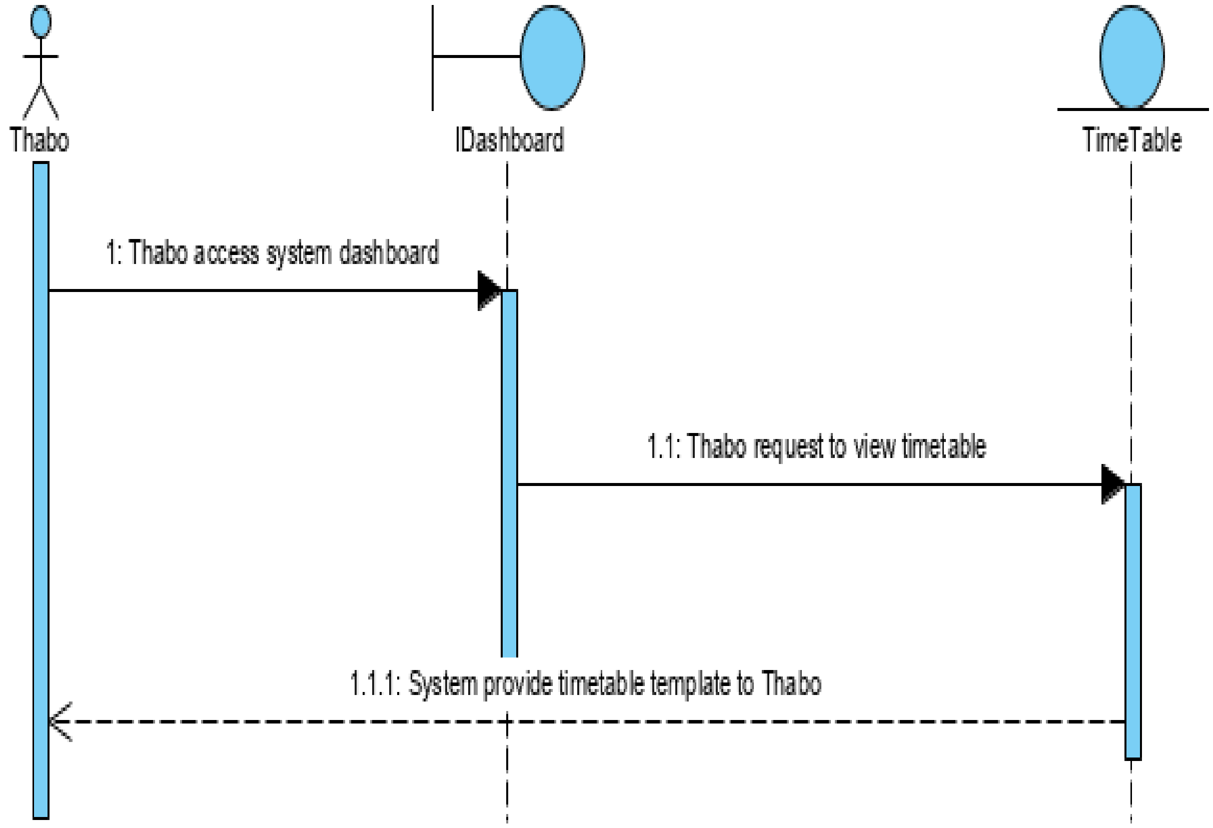
Online Chat


Thabo
IDashboard
IModule
Online chat
1: Thabo accesses system dashboard
2: Goes to module
3: Request online chat
4: Conversation between lecture and student

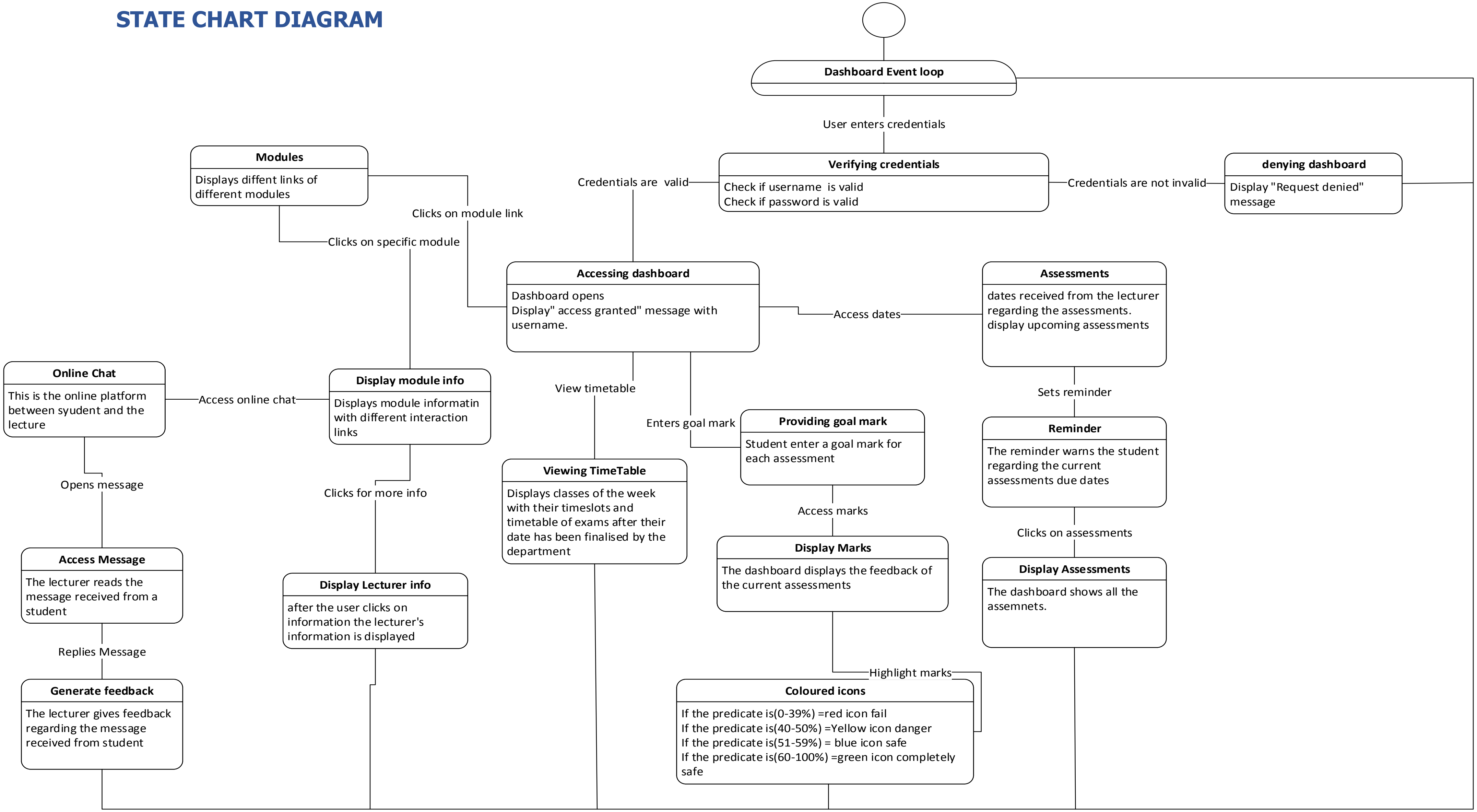
Machine learning


Thabo
IDashboard
IModule
Machine learning
1: Thabo accesses system dashboard
2: Goes to module and request semester marks
3: system gets Thabo's marks
4: return semester mark

View Timetable


Thabo
IDashboard
TimeTable
1: Thabo access system dashboard
1.1: Thabo request to view timetable
1.1.1: System provide timetable template to Thabo

STATE CHART DIAGRAM


Dashboard Event loop
User enters credentials
Verifying credentials
Check if username is valid
Check if password is valid
Credentials are valid
Credentials are not invalid
denying dashboard
Display "Request denied" message
Modules
Displays diffent links of different modules
Clicks on module link
Clicks on specific module
Accessing dashboard
Dashboard opens
Display" access granted" message with username.
Access dates
Assessments
dates received from the lecturer regarding the assessments.
display upcoming assessments
Online Chat
This is the online platform between syudent and the lecture
Access online chat
Display module info
Displays module informatin with different interaction links
View timetable
Enters goal mark
Providing goal mark
Student enter a goal mark for each assessment
Sets reminder
Reminder
The reminder warns the student regarding the current assessments due dates
Opens message
Clicks for more info
Viewing TimeTable
Displays classes of the week with their timeslots and timetable of exams after their date has been finalised by the department
Access marks
Clicks on assessments
Access Message
The lecturer reads the message received from a student
Display Lecturer info
after the user clicks on information the lecturer's information is displayed
Display Marks
The dashboard displays the feedback of the current assessments
Display Assessments
The dashboard shows all the assemnets.
Replies Message
Highlight marks
Generate feedback
The lecturer gives feedback regarding the message received from student
Coloured icons
If the predicate is(0-39%) =red icon fail
If the predicate is(40-50%) =Yellow icon danger
If the predicate is(51-59%) = blue icon safe
If the predicate is(60-100%) =green icon completely safe

Pseudocode

Machine Learning

```
/* Here were are changing the module link that leads a user to the module homepage
        the colour of the link is based on the student's final semester mark e.g. red for marks
        between 0 and 39 */
      public void Machinelearning()
      {
         public double dMarkFinal;

      if(dMarkFinal <= 39) /* if semester mark is bellow 39 or equal 39 then the colour link colur changes to red */
         {
            a : link {colour: red;}
         }
         else if (dMarkFinal <= 40  || dMarkFinal == 50)/* if semester mark is above 40 or equal 50 then the colour link colur changes to red */
         {
            a : link {colour : yellow;}
         }
         else if (dMarkFinal <= 51 || dMarkFinal == 59)/* if semester mark is above 51 or equal 59 then the colour link colur changes to red */
         {
            a : link{colour : blue;}
         }
```

```
            else if(dMarkFinal >= 60)/* if semester mark is above 60 or equal 60 then
the colour link colur changes to red */
            {
                a : link {colour : green}
            }
```

Goal Setting Algorithm:

```
Input: n, n = {N} N = 1; n is the goal mark
Initialize: f(v)max = 0; iteration = 0;
//Call Pro-PAM algorithm to find K medoids of S
Call .........algorithm incorporated in the ML to store the goal mark

foreach N comforming to n do

        Compute f(v):
        if f(v) < f(v)max then
        iteration = iteration + 1;
        //Goal mark should be promising and attainable (Cannot be less than 50)

        else
        f(v)max = f(v)
        BestGoalSet = n (CurrentMark)
  End
```

Return Output: GoalMark = BestGoalSet

Lecturer Details

private IRepositoryWrapper _repository;

public IActionResult Details(int id) which is the method that will show the lecturer details

```
{
        Var lecturer = _repository.Lectuer.GetLecturerDetails(id);

        If (lecturer == null)

          Return NotFound();

}

  else

{

        Return View(lecturer),

}
```

Check Timetable

<embed src="files/Brochure.pdf" type="application/pdf" width="100%" height="600px" />

Login

Private read-only User Manager<Identity User> _user Manager; Provides the application-programming interface for user sign in.

private readonly SignInManager<IdentityUser> _signInManager; Provides the application programming interface for user sign in.

public AccountController(UserManager<IdentityUser> userManager, account controller is the architecture handles any incoming URL

```
SignInManager<IdentityUser> signInManager)
{
    _userManager = userManager;
    _signInManager = signInManager;
}
[AllowAnonymous]
```

AccountController you will notice the generous use of the AllowAnonymous Attribute on various login and register controller actions.

```
public IActionResult Login(string returnUrl)
```

Action method return models to views, file streams, redirect to other controllers

```
{
    return View (new LoginModel
    {
        ReturnUrl = returnUrl
    });
}
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Login(LoginModel loginModel)
```

Just like the existing Forms Authentication module, the turn of the protected resource that the user attempted to access is sent as a request urn query string parameter. We need to ensure this is passed to us when the user submits the login form so it stored in the Login Model and rendered as a hidden input on the page.

```
{
```

if (ModelState.IsValid) **ModelState**.**IsValid** tells you if any **model** errors have been added to **ModelState** . The default **model** binder will add some errors for basic type conversion issues (for example, passing a non-number for something which is an "int").

```
    {
        IdentityUser user =
        await _userManager.FindByNameAsync(loginModel.Name);
        if (user != null)
        {
            var result = await _signInManager.PasswordSignInAsync(user,
            loginModel.Password, false, false);
            if (result.Succeeded)
            {
                return Redirect (loginModel?.ReturnUrl ?? "/Home/Index");
            }
        }
    }
    ModelState.AddModelError("", "Invalid Username or password");
    return View(loginModel);
}
[HttpPost]
public async Task<IActionResult> Logout ()
{
    await _signInManager.SignOutAsync();
```

```
            return RedirectToAction ("Index", "Home"); the redirect action return to the controller class

        }

        [AllowAnonymous]

        public IActionResult AccessDenied ()

        {

            return View ();

        }
```

SOFTWARE PROJECT MANAGEMENT PLAN

1.1Project Summary

1.1.1. Purpose, Scope and Objectives **–** The product will allow the student to login and direct them to the modules they take in the Computer Science department. A student must be able to click on the appropriate module and view to see their distinctive timetable. Other functionalities include, student being able to see all the assessments due dates and a reminder system which notifies them about the deadlines when it is less than three (3) days left. The product will also load the marks of the students derived from the DBMS PeopleSoft directly. Furthermore, the product will incorporate machine learning and use this to make calculations of the mark needed to pass, comparing the goal mark inserted by the student during the beginning of the semester and the actual mark obtained and indicate the current state. When the student clicks on each module, the product needs to display the details of the lecturer; the name, office number and email address (and phone number in mobile device).

The other user that needs to be authorized is the lecturer, the product should allow them to also login and direct them to the modules they give out. The product should provide the lecturer with the functionality to administer the dates and marks of the assignments and tests. The product will also cater for a collaboration tool which allow the interactive environment between lecturer and students to talk about the progress and any subject related to the module. With all of this being said, it is clear that the product needs to provide the appropriate dashboard based on the logged in user.

Constraints and Assumptions –

Assumptions include:

- Assuming that the application will be operated by users who are computer literate to a certain degree, those being the student, lecturers and other relevant staff members.

Constraints

- The quality and the standard of the product has to be appealing.
- We have a deadline to meet, therefore time of essence here.
- The construction of the architecture has to cater for future modifications of functionalities which will be added.
- The product has to be user-friendly.
- The product has to be well adjusted to operate on both mobile devices and desktop.

$$Hours\ of\ project = Hours\ per\ week_{(assuming\ programmer\ works\ 8\ hour\ per\ dsy)} \times 12weeks$$

$$= 1.5 \times 5 \times 12$$

$$= 90\ Hours$$

- The resources include stationary, electricity, and internet all total up to: R 6600.00

- The Budget for the project

$$Budget = Bill\ of\ software + (hours\ per\ week \times hourly\ rate)$$

$$= R8000.00 + R6600.00 + (90 \times R180.00)$$

$$= R30\ 800.00$$

- Research based on similar projects that have been completed before will be re-used in this project together with all the managerial structure

1.1.2. Project Deliverables – The final product with everything required by the user completed as well as better suggestions by expert incorporated, as well as the user manual. All will be handed after 10 weeks of the project commendation. The client will also be delivered will

Schedule and Budget

- Week 1-4: Gather and Complete the requirements testing of requirement is required
- Week 5-6: Analysis of requirement is done and iterative and incremental method is used and the classes are extracted
- Week 7-8 Class Diagrams, Sequence Diagrams, State Charts Diagrams are done and the Software Project Management Plan are done and Completed then they are tested.
- Week 8-10 Programmers get their Modules and coding begins
- Week 8-12 Testing of the project is done now using user case scenarios and the SQA puts the software into intensive tests and faults are corrected and it is delivered to the client with a binding contract.

1.2. Evolution of the project Management plan

- All changes in the project management plan must be agreed to by each and every team member before they are implemented
- To ensure that the project management plan is up to date all changes must be documented

2. Reference Materials:

All the artifacts will conform to Unified-Process's documentation. Programming, modelling and testing standards

3. Definitions and acronyms:

Not applicable

4. Project Organization

4.1 External Interfaces: **Not applicable**

4.2 Internal Interfaces:

All the work of this project will be performed by (Maine Gopolang, Leputla Mofehlo, Mokoena Katleho, Mosia Serame, Ngobese Lungisa, Rakhosi Lerato, Tsepe Tau).

The team members will submit the work on a weekly basis to the client. The client will go through the work and the client will somehow specify where changes and modifications could be made by the team on the software product.

4.3 Roles and Responsibilities

Since the team makes use of the democratic team approach for this project, most of the roles and responsibilities are shared amongst the team members.

For each workflow, every group member will do their own part and the artifacts are combined during the weekly meetings. Each group member is responsible for the quality of the artifacts he produces. The group will oversee the integration and the overall quality of the software product. Lungisa Ngobese will then communicate with the client.

5. Managerial process plans

5.1. Start-up plan

5.1.1. Estimation plan –

Documentation of the project is estimated to take 9 weeks and implementation of the prototype to take 3 weeks. The total development time is estimated to be 12 weeks with an estimated budget of R30 800.00

5.1.2. Staffing plan -

As stated before, the project is developed using the democratic team approach for the first 2 weeks. Each member of the team studies the client requirements and the team meets twice a week to compile client requirements and iterate requirements until all members are satisfied. Week 3 from the requirements, each member of the team puts together a list of use cases and the team meets twice a week to go through all use cases and iterate both use cases and requirements. Week 4 the team meets twice to compile class diagrams, week 5 and 6 delegate each use case to a member of the team to design a state-charts and sequence diagrams.

5.2 Work Plan

5.2.1. Work activities and Schedule Allocation

Week 1: 3rd & 5th August – (Completed) we discussed the client requirements and gained the understanding of the application domain and how is going to fit in the environment where is going to be used. Some of the client's requirements where not practical and we had to remove some of the requirements and some of the requirements that will make the software product more effective.

Week 2: 26th, 27th & 29th August - The report that was sent to the client came back with some other additional requirements the client wants. We iterated the requirements, with that requirements we created the use case diagrams that will show the interaction between the software product and the users of the software product. Every use case we designed have the normal scenario and the worst case scenario. This scenario will help us to know the errors that the users of the software product might make during the interaction.

Week 3: 2nd & 5th September - We were on the analysis phase so we extracted the noun from the paragraph we created to describe the software product, with the nouns that were extracted we indicated the classes and the subclasses. UML and class diagrams were designed with the classes that were collected from the noun extraction.

Week 4: 9th September - we iterate the use case diagrams and after the iteration we designed the sequence diagrams for each of the use case diagram.

Week 5: 16th & 19th September - this period is allocated to iterating and incrementing all the artifacts from the analysis workflow. SPMP was created with all the necessary information provided by the team members.

Week 6-10: Iterating and incrementing all the completed artifacts from the requirements workflow to the design workflow. Therefore, all the artifacts will be checked and refined where needed, whilst all members of the group are still ensuring quality in all of the artifacts. : Further testing will be performed and all the artifacts from different workflows are put together and the entire project is finalized. All the deliverables are inspected.

Resource Allocation:

The team follows the democratic set up – All the artifacts and their deliverables will be allocated among all 7 members.

- Host of regular meetings, twice per week for integration of all the artifacts and iterations for ensuring there is quality and everything blends well.
- Otherwise, make use of the group chat to interacting with each other, sending the material through, engaging each other and clarifying any uncertaintie

5.2.2 Budget Allocation:

Person-hour and payments

Names	Requirements Hours	Analysis Hours	Design Hours	Implementation Hours	Testing Hours	Rate 180/H
Gopolang	13	20	4	20	5	11 160
Lerato	15	21	5	20	5	11 880
Tau	15	18	5	21	5	11 160
Mofehlo	16	21	4.5	21	5	12 330
Katleho	15	21	5	21	5	12 060
Lungisa	15	19	5	20	5	11 520
Serame	15	20	5	19	5.5	11 430
Total	**119/h**	**140/h**	**33.5/h**	**142/h**	**35.5/h**	**R 81 540**

5.3. Control Plan

If there are any changes that affect the budget we have placed and changes the artifacts and need to be introduced. We all sit down as a group, work around the idea and reach a common ground, if this imposes changes we note it down and it will be documented.

We do not require any outside quality assurance as we have made it a mandate that it is every group member's responsibility to assure there is quality in anything we produce in this software product. We also hold each other accountable by running tests and ensuring there is quality in any of the inputs.

We set deadlines dates for any of the duties that we need to serve, every member is held accountable upon the delivery of the duty they needed to perform. Any of the problems we encounter, we sit down as a team and evaluate the situation without taking any sides.

6. Technical process

6.1 Process Model

The unified process will be used for this project.

6.2 Methods, Tools, and Techniques

All workflows will be performed in accordance with the unified process. The prototype will be implemented in HTML and CSS, and the final product will be implement in c#.

6.3 Infrastructure Plan

The prototype will be developed on wix.com running on Firefox browser on a windows operating system personal computer. The final product will be implemented on visual studio running on a windows operating system personal computer.

6.4 Project Acceptance Plan

Acceptance of our product by the client will be achieved by following all the steps of the unified process. Another method which we will use to achieve Acceptance of our final product is that a prototype will be built for the client to interact and familiarize himself/herself with the product to be developed

7. Supporting process plans

7.1 Configuration management plan

- CVS will be used throughout for all artifacts.

7.2 Testing plan

The testing workflow of the Unified Process will be used. It will be carried out in parallel with the other workflows.

- The programmers will guide the other members through their code.
- SQA will perform unit testing, integration testing and product testing.
- Finally, the client will perform acceptance testing.

7.3 Documentation plan

The documentation will be produced as specified in the Unified Process

7.4 Quality assurance plan, reviews and audits plan

No phase is complete until the documentation is finalised by the approval of the Software Quality Assurance group.

If the products of the earlier phase must be modified due to a feedback loop, then that phase is only complete if the documentation for that phase have been checked and approved by the Software Quality Assurance group.

7.5 Problem resolution plan

If a problem arises, the team will identify the causes of the problem and brainstorm the possible solutions. The team will select the best solutions and implement the plan. A follow-up plan must be developed in case the one implemented plan wasn't effective enough.

7.6 Subcontractor management plan

Not applicable here

7.7 Process improvement plan

For improved software quality ISO 9000 and CMM will be used.6

8. Additional plans:

Security: A password and username will be needed to use the product

Training: Not Applicable

Maintenance: Corrective maintenance will be performed by the team for a period of 6 months at no cost. A contract will be drawn for Perfective maintenance and adaptive maintenance.

Incremental and Iterative processes

Requirements

The specifications of the required functionality were received from the client, they were analyzed, and team managed identify applicable functionalities. The functionalities that were not applicable were discarded. After the second meeting with the client, new additional functionality requirements were introduced. Therefore, the team refined the initial requirements through iterating with each increment needed. The final specification was compiled and sent back to the client for approval

Analysis

Given the approval from the client, the specifications were broken into series of components. Each of the components that were identified the team managed to derive the classes and place them according to their categories, for example, control classes grouped together with border classes and entity classes, and how they are compatible with one another. Interfaces are built to enforce compatibility and flexibility as well to improve the robustness of the product's architecture. Through each identity Planning and testing were are performed and every changed introduced to the system is accounted for, the same occurred for composition of attributes and methods which provide the inter relationships between the classes.

Design

Using the classes and their attributes from analysis the UML diagram was constructed. The relationships and related multiplicities were outlined and displayed on the UML diagram. Use cases were developed from the UML diagrams, with various case scenarios. Iterations were performed to come up the usable use cases displaying needed scenarios.

Sequence diagrams became derived from the scenarios embodied in the use cases. Increments were enforced by developing statechart diagrams to show the functinality of the product as a whole system. Iterations were perfomed with each increment brought forth.

Code selection and compilation:

- Login – Mofehlo
- Machine Learning – Serame and Lungisa
- Check timetable – Gopolang
- Set goal – Lerato
- Lecturer details – Katleho and Tau
- Online chat – The whole team

Integration

At the end of the day all the modules are tested, and checked to see if they all fit together, and the every activity is frozen at the end of the day. The following day the team resumes from where we started. This was done so that the modules can get used to working together early and see if the system works perfectly.

QUOTATION

ELEMENTS REQUESTED	DESCRIPTION	ESTIMATED HOURS	PRICE
Analysis workflow	Analyse and refine the requirements	21	R4 000
Design workflow	Refine the artifacts of the analysis workflow	8	R4 000
Implementation workflow			
Login page	Includes: • Google authentication	1	R480
Timetable page	Includes: • A list of modules • Class venues and time • Printable page	1.50	R480
Online chat with the lecturer	Send queries	4	R1 120
	Receive feedback	4	R1 200
	A notification when you have a message.	1	R240
Lecture details	This page will display all the lecturer's details i.e. contact details, office number, consultation hours.	1	R240
	Editable page, in case the lecturer changes contact details, or office.	3	R2160

Goal setting	Students provide a goal mark at the beginning of the semester.	2	R960
Machine learning	Compute semester mark.	2	R960
	Compute mark needed to reach a certain goal.	2.5	R1200
	Predicts if the student will "pass" or "fail" based on their current performance.	6	R2000
	Generates prediction's report for the lecture.	1	R480
	Includes icons to indicate exam promise based on the student's achieved predicated	1	R480
Testing workflow			
Non-Execution-Based testing	Test the software without using test-cases	3	R4 000
Execution-Based testing	Test the application against the use-cases	4	R6 000
SUBTOTAL			R30 000
GENERAL EXPENSES			
Water			R800
Rent			R8 000
Electricity			R9 000
Internet			R4 000
Insurance			R7 000
SUBTOTAL			R28 800
SALES TAX			14%
TOTAL			R58 800