KKW MOKHETHI (2016213635)

**CSIS3724**

software engineering semester project

Innovative Systems

Table of Contents

[1 Requirements report 3](#_Toc22305396)

[Suggested modification(s): 3](#_Toc22305397)

[2 Specification Document 4](#_Toc22305398)

[2.1 Use cases and Scenarios 4](#_Toc22305399)

[2.1.1 **Login** use case 5](#_Toc22305400)

[2.1.2 **Present timetable** use case 6](#_Toc22305401)

[2.1.3 **Display lecturer’s contact details** use case 7](#_Toc22305402)

[2.1.4 **Set goal mark** use case 8](#_Toc22305403)

[2.1.5 **Set assessment date** use case 10](#_Toc22305404)

[2.1.6 **Set reminder of upcoming assessment** use case 11](#_Toc22305405)

[2.1.7 **Warn student about upcoming assessment** use case 12](#_Toc22305406)

[2.1.8 **Display assessment** use case 13](#_Toc22305407)

[2.1.9 **Display results recorded** use case 14](#_Toc22305408)

[2.1.10 **Provide report** use case 15](#_Toc22305409)

[2.1.11 **Send query or question** use case 16](#_Toc22305410)

[2.1.12 **Give feedback** use case 17](#_Toc22305411)

[**2.2 Class extraction** 18](#_Toc22305412)

[**2.2.1 Entity class extraction** 18](#_Toc22305413)

[(a) Noun-extraction 18](#_Toc22305414)

[(b) CRC cards 19](#_Toc22305415)

[(c) Class diagram 20](#_Toc22305416)

[**2.2.2 Boundary class extraction** 20](#_Toc22305417)

[**2.2.3 Control class extraction** 20](#_Toc22305418)

[2.2.4Class diagram (showing stereotypes) 21](#_Toc22305419)

[2.2.5Dynamic Modelling 22](#_Toc22305420)

[Statecharts 22](#_Toc22305421)

[Student class 22](#_Toc22305422)

[Lecturer class 23](#_Toc22305423)

[2.3 Sequence diagrams for: 24](#_Toc22305424)

[2.3.1 **Login** use case 24](#_Toc22305425)

[2.3.2 **Set goal mark** use case 24](#_Toc22305426)

[2.3.3 **Present timetable** use case 25](#_Toc22305427)

[2.3.4 **Display lecturer's contact details** use case 25](#_Toc22305428)

[2.3.5 **Set assessment date** use case and **Set reminder of upcoming assessment** use case 26](#_Toc22305429)

[2.3.6 **Display assessment** use case 26](#_Toc22305430)

[2.3.7 **Display results recorded** use case 27](#_Toc22305431)

[2.3.8 **Provide report** use case 27](#_Toc22305432)

[2.3.9 **Send query** use case and **Give feedback** use case 28](#_Toc22305433)

[3 Software Project Management Plan 29](#_Toc22305434)

[4 **Quotation for the system** 36](#_Toc22305435)

[**5 Design Documentation** 37](#_Toc22305436)

[Object Oriented Design 37](#_Toc22305437)

[1. Completion of the class diagram 37](#_Toc22305438)

[2. Performing a detailed design 38](#_Toc22305439)

[**Pseudo code** 38](#_Toc22305440)

[For login functionality: 38](#_Toc22305441)

[For set goal mark functionality: 39](#_Toc22305442)

[For set assessment date functionality: 39](#_Toc22305443)

[For performing calculations and predicting future results: 39](#_Toc22305444)

[6 Discussion of development 40](#_Toc22305445)

[7 Record of hours spent on the project 40](#_Toc22305446)

# 1 Requirements report

The requirements of the Timetable Application to be developed were clearly stated and I have gone through all of them in detail. It is understood that the system will be mainly designed and developed for students, to make it easy for them to access their weekly timetable and be kept up to date about any upcoming assessment, their performance and the mark needed to reach their set goal mark. The identified users of the Timetable Application will be both the students and the lecturer, with the lecturer providing all the information about upcoming assessments and the course work and the student receiving the information and interacting with the application.

The Timetable Application will be easy to use on both mobile devices and desktops as required. A student will easily login this application with their credentials and immediately view the timetable for the entire week unless they specify to view for the current day or view for the next three days. Each time an assessment date approaches (3 days before the due date) and reminder system will kick in and inform the student about the upcoming assessment.

The application will make it possible for the student to view marks of each course/subject without having to log in PeopleSoft marks system as the application will access the PeopleSoft system and calculate the current semester mark for each course and also calculate the required mark for each upcoming assessment in order to reach the goal mark that was set by the student on the first day of the semester.

# Suggested modification(s):

* The Timetable Application should not only display the subject and its lecture allocated time and show the venue and the lecturer’s name only after clicking the subject, instead, the venue and the lecturer responsible for that subject should be displayed as well. This will reduce the number of clicks in the application and makes the application to be user friendly, avoiding unnecessary clicks.
* Hovering on the lecturer’s name should quickly show the lecture’s contact details by the side without having to wait until clicking the lecture’s name and then be re-directed to the contacts page or section.

# 2 Specification Document

# 2.1 Use cases and Scenarios



## 2.1.1 **Login** use case



Brief description:

The login use case allows the student to gain access to the application when provided with correct credentials.

Step-by-Step description:

1. The student opens the timetable application.
2. A screen display is presented with a “login” section.
3. The “login” section has two text boxes, one for the username (student number) and the other for the password, and a login button to submit the credentials.
4. The student enters his/her student number and password in appropriate text boxes
5. Student clicks the login button.
6. Student credentials are validated.
7. The student is logged into the application.

Normal Scenario:

Katlego wants to navigate through the application and see how the application works.

1. Katlego opens the timetable application.
2. The first form is displayed invoking the student to get started
3. A login prompt is presented on the screen.
4. Katlego enters his student number and his password in respective text boxes.
5. Katlego submits his details by clicking the login button.
6. The system makes a validity check and validates the student’s details.
7. Katlego’s details are correct and validated.
8. Katlego successfully gained access to the application.
9. Katlego is presented with the main form and starts navigating through the application.

Exception Scenario:

1. Katlego clicks the login button straight away with attempt to access the application without providing any details.
   1. An error occurred.
   2. The application provides descriptive error messages indicating that the student number and a password are required.
2. Katlego enters his student number and password with CAPS on, resulting in typing in the wrong password.
   1. Access is denied.
   2. A descriptive error message informs Katlego that he has input incorrect details and he should try again.

# 2.1.2 **Present timetable** use case



Brief description:

The Present timetable use case allows the student to view the timetable for all courses that the student is enrolled for. This use case further extends the behaviour to view the lecturer’s contact details on hovering on the lecturer’s name.

Step-by-Step description:

1. As soon as the student is successfully logged in, by default, a weekly timetable is presented on the screen.
2. The Timetable displays the course code, the venue of the class and the lecture’s name under the course code.
3. On the side of the timetable there’s two additional options two view the timetable
4. The student is allowed to view the timetable for a day or for the next three days.
5. The view mode of the timetable changes dynamically.

Normal Scenario:

Michael wants to know which lectures will he be attending between Wednesday and Friday.

1. By default, a weekly timetable is presented.
2. Michael selects “Thee days” radio button option.
3. The timetable displays only lectures available from Wednesday until Friday.

Exception Scenario:

There’s no exception scenario for this use case. A week’s timetable is displayed by default and the use of radio buttons restricts the student to only three options, being the default one of a week, a day and of three days. Only one can be viewed at once. The student can never go wrong here.

## 2.1.3 **Display lecturer’s contact details** use case



Brief description:

The Display lecturer’s contact details use case enables the student to view the lecturer’s phone number and email address when the student hovers on the lecturer’s name. The Open required programme use case is part of the Display lecturer’s contact details and it is responsible for re-directing the student to the appropriate programme of communication depending on the contact detail that is clicked.

Step-by-Step description:

1. Student hovers on the lecturer’s name
2. The lecturer’s phone name and email address are displayed.
3. The student clicks one of the contact details:
4. If the student clicks the lecturer’s phone number:

* The Open required programme use case opens the phone call program if the student is using a mobile device and auto-fills the lecturer’s phone number in the call program, ready to make a call.

1. If the student clicks the lecturer’s email address:
2. The Open required programme use case opens Gmail in a new browser window (this applies both on the PC and mobile device) and auto-fills the lecturer’s email address on the recipient section.

OR

1. The Open required programme use case opens a custom Gmail application within the Timetable App and auto-fills the lecturer’s email address on the recipient section.

Normal Scenario:

Kabelo want send an email to Mr Nel and submit the doctor’s certificate as proof that he was not doing well and that is the reason he could not write the first semester test of CSIS2634.

1. Kabelo visits the timetable and locates to CSIS2634
2. Kabelo hovers on the Mr Nel’s name.
3. Mr Nel’s phone number and email address are displayed on the screen.
4. Kabelo clicks the email address of Mr Nel.
5. The Open required programme use case re-directs Kabelo to Gmail in a new browser window.
6. Mr Nel’s email address is auto-filled in the recipient section.
7. Kabelo continues writing and sending the email.

Exception Scenario:

There is no possible exception scenario for the Display the lecturer’s contact details since the only interaction of the student to the application is just hovering on the lecture’s name and the lecturer’s contact details are displayed.

## 2.1.4 **Set goal mark** use case



Brief description:

The Set goal mark use case enables the student to set a goal mark for each course, this is the mark that the student wishes to achieve by the end of the semester.

Step-by-Step description:

1. The “set goal mark” panel is activated and made visible by invoking the “Set Goal Mark” button.
2. For each semester, a list of computer science courses that the student is enrolled for is presented.
3. When the student selects a course, the course code is reflected in a label component above a numeric up-down component which records the mark.
4. Student selects the numeric up-down component.
5. Student sets a value ranging between 0 and 100 (inclusive)
6. Student clicks the “Set goal mark” button.
7. The goal mark for the selected course is recorded in a list component of goal marks.
8. Student clicks the “Save Goal Marks” button and all goal marks for each course are saved in the system.

Normal Scenario:

Michael is a third-year student, at the beginning of first semester he already has plans about his academic achievements and wishes to attain at least 72% for CSIS3714. He runs the application to set his personal goal mark.

1. Michael has logged in the application.
2. Michael navigates to the side menu.
3. Michael clicks the “Set Goal Mark” button.
4. The “Set Goal Mark” panel is displayed on the screen.
5. Michael selects CSIS3714 course from the first semester list of courses.
6. Michael sets the value to 72%
7. The value is accepted by the numeric up-down component.
8. Michael clicks the “Set Goal Mark” button.
9. The goal mark for CSIS3714 is recorded.
10. Michael repeats this process for all remaining courses setting a goal mark for each course.
11. The system displays a message confirming that Michael’s goal marks were successfully saved once he clicks the “Save Goal Marks” button.
12. The “Set Goal Mark” panel is deactivated and disappears from the screen.

Exception Scenario:

1. Michael clicks the “Set Goal Mark” button without specifying a course and a value.
   1. An error message is displayed indicating that a course needs to be specified before setting a goal mark and 0% cannot be accepted as a mark.
2. Michael sets the value to 45% and clicks the “Set goal mark” button. A warning message reads that any mark below 50% is a fail and will not be accepted as a goal mark.

# 2.1.5 **Set assessment date** use case



Brief description:

The Set assessment date use case enables the lecturer to set when an assessment is due.

Step-by-Step description:

1. The lecturer is provided with a drop-down list of all courses that he/she lectures.
2. The lecturer selects the desired course from the drop-down list.
3. The lecturer specifies the type of the assessment whether is it an assignment, a semester test or a practical test.
4. The lecturer sets the date of the assessment from a DateTimePicker component.
5. The lecturer clicks the Save button
6. A message is displayed indicating that the assessment date was successfully set and was saved.
7. Student information of all students who are enrolled for the selected course is updated.
8. Each student will now have a new assessment available for that particular course.

Normal Scenario:

Prof Stott has prepared the first semester test for CSIS3714 students a week before the test day. She logs in the application and sets the test date.

1. Prof Stott clicks the “Set Assessment Date” button.
2. Prof Stott is presented with a drop-down list of courses that she lectures.
3. Prof Stott scrolls through the list and selects CSIS3714.
4. Prof Stott selects the first semester test as the assessment type from a drop-down list.
5. Prof Stott specifies the date of the first semester test from the DateTimePicker component.
6. Prof Stott saves this activity by clicking the “Save” button.
7. The system sends a short feedback message indicating that the assessment date was set and saved.
8. Prof Stott is immediately redirected to the “Set Assessment Reminder” section to start setting a reminder for that assessment.

Exception Scenario:

1. Prof Stott quickly selects that the assessment is a first semester test
2. Prof Stott then sets the date from the DateTimePicker component
3. Prof Stott clicks the Save button.
4. An error occurs indicating that a course must be specified for an assessment.

2.1.6**Set reminder of upcoming assessment** use case



Brief description:

The Set reminder of upcoming assessment use case is part of the Set assessment date. This use case enables the lecturer to set a reminder in order to warn students about the assessment few days before the due date.

Step-by-Step description:

1. A list of assessments for each course is displayed on the screen.
2. The lecturer selects the assessment he/she wants to set a reminder for.
3. The lecturer picks the reminder date from the DateTimePicker component.
4. This activity is recorded as the “Set Reminder” clicked and a confirmation message is displayed on the screen.

Normal Scenario:

Prof Stott has to remind students about the first semester test two days before its due date.

1. Prof Stott selects the first semester test for CSIS3714 from the assessments list.
2. Prof Stott sets the reminder date from the relevant component.
3. Prof Stott clicks the “Set Reminder” button and the assessment reminder is stored in the system.

Exception Scenario:

1. Prof Stott specifies the reminder date as soon as she is redirected to the “Set Assessment Reminder” section then saves by clicking the “Set Reminder” button.
2. An error message is displayed indicating that an assessment should be specified before setting the date.

# 2.1.7 **Warn student about upcoming assessment** use case



Brief description:

This Warn student about upcoming assessment use case informs students about upcoming assessments and it is part of the Set reminder of upcoming assessment.

Step-by-Step description:

1. On the date set for the reminder, from the dashboard section the “Warning” button is activated and system indicates a red warning flag on that “Warning” button.
2. When the “Warning” button is clicked a message pops on the screen alerting the student about the upcoming assessment.

Normal Scenario:

Michael is running the application checking out his timetable for the week and as soon as he visits the dashboard, he notices a warning flag.

1. Michael clicks the “Warning” button invoking a warning message.
2. A descriptive warning message alerts Michael that he will be writing CSIS3714 in two days’ time.
3. Michael clicks the “Ok” button, the red flag disappears and the “Warning” button is disabled.

Exception Scenario:

None.

## 2.1.8 **Display assessment** use case



Brief description:

The Display assessment use case provides the student with available assessments for each course.

Step-by-Step description:

1. On the dashboard, when the “Assessments” button is clicked the system retrieve all recorded assessments of the logged in student.
2. All available assessments for each course are displayed on the screen.

Normal Scenario:

Prince wants to make preparations for his tests. He wants to know the test dates for each course in order to draft a plan for his preparations.

1. Prince navigates to the dashboard.
2. He clicks the “Assessments” button and all upcoming assessments that were set by lecturers for each course are presented on the dashboard
3. Prince notes down when will he writing his tests and logs out.

Exception Scenario:

When the student clicks the “Assessments” button and there’s no available upcoming assessments yet, nothing is displayed.

## 2.1.9 **Display results recorded** use case



Brief description:

The Display results recorded use case enables the student to check his/her current assessment marks making up the semester mark.

Step-by-Step description:

1. The student clicks the “View results” button on the dashboard
2. A request is sent to PeopleSoft marks system
3. PeopleSoft marks system compiles a full report of the student’s results for each course
4. PeopleSoft marks system sends the results back to the Timetable Application
5. The results recorded are displayed for each course

Normal Scenario:

Kabelo wants to know his performance so far this semester.

1. Kabelo navigates to the side menu
2. Kabelo clicks the “View results” button
3. A full report of results recorded is fetched from PeopleSoft marks system and presented to Kabelo on the screen.
4. Kabelo navigates to the dashboard from the side menu
5. He clicks the “Dashboard” button and the dashboard is presented on the screen
6. He invokes the “Overall Performance button” and views his performance for each course

Exception Scenario:

1. PeopleSoft marks system might be down (temporarily) at the moment the results recorded are being requested from it.
2. As a result, Kabelo may not be able to get his results.

## 2.1.10 **Provide report** use case



Brief description:

The Provide report use case makes a prediction of the student’s results available to both the student and the lecturer.

Step-by-Step description:

1. The systems compute the student’s predicted results based on their past performance and a trend of their current performance.
2. The predicted results are written in a report.
3. The report is sent to the lecturer in question.
4. The report is sent to the student as a blue information icon on the student’s timetable.

Normal Scenario:

1. Kabelo logs in the application and checks his timetable for the week.
2. On viewing the timetable, an icon pops out on the timetable.
3. Kabelo clicks the icon.
4. A report of Kabelo’s predicted results is presented on the screen.
5. Kabelo views his predicted results based on the trend of his current performance and past performance.
6. Kabelo closes the report window and continues to use the application.
7. Mr Nel logs in the application to check if he has set all assessments dates for CSIS2634, as he is busy with that, a question icon pops on his screen.
8. Mr Nel clicks the question icon
9. A short summary message is indicated on the screen about predicted results of all students for CSIS2634.
10. Mr Nel selects one student to view predicted results.
11. A full report of the selected student is presented on the screen.
12. Mr Nel views the report and closes the report as soon as he is done going through it.

Exception Scenario:

There is no exception scenario for the Provide report use case as the report is sent periodically when the system has predicted the student’s results. Both the student and the lecturer have an option to either view the report or ignore it. There only interaction with the report is to open and close it.

## 2.1.11 **Send query or question** use case



Brief description:

The Send query or question use case allows the student to send a query or a question to the lecturer whenever the student has any. The Receive notification use case is part of the Send query or question use case. This Receive notification use case will be used to inform the lecturer about the message sent by the student.

Step-by-Step description:

1. An “Online Chat” button is invoked from the side menu.
2. An Online Chat panel is presented on the screen with a query section.
3. The student should specify the lecturer that he/she wishes to send a query to from a drop-down list of lecturers.
4. Once the lecturer is specified and a query is typed in, the “Send Query” button is clicked then the Receive notification use case sends a notification message to the specified lecturer.
5. When the lecturer is online, he is notified about new queries.
6. The lecturer opens and reads the student’s query.

Normal Scenario:

Katlego wants to ask Mr Marais how wireless networks differs with ethernet networks on the data-link layer because he did not understand that during the lecture.

1. Katlego navigates to the side menu and open the Online Chat panel.
2. Katlego is presented with the query section selects Mr Marais from the drop-down list of recipients.
3. Katlego writes his query in the query Rich Text Box.
4. Katlego sends his query by clicking the “Send Query” button.
5. A notification message is then sent to Mr Marais.
6. When Mr Marais is online, he receives the notification message.
7. Mr Marais clicks to open the message and reads it.

Exception Scenario:

A:

1. Katlego quickly navigates to the query section and types in his query.
2. Soon after writing his query he clicks the “Send Query” button to send the query.
3. An error message shows on the screen.
4. A recipient must be indicated.
5. Katlego did not indicate who to send the query to.

## 2.1.12 **Give feedback** use case



Brief description:

The Give feedback use case enables the lecturer to send feedback to the student after reading the student’s query.

Step-by-Step description:

1. On the lecture’s side, upon reading the notification message, the lecture has a feedback section to respond to the student.
2. The lecturer writes feedback with regard to the student’s sent query.
3. A send button is clicked and feedback is sent to the student.
4. The student receives the lecturer’s feedback and reads it.

Normal Scenario:

Mr Marais sends feedback to Katlego.

1. Mr Marais explains, in writing, how the wireless networks and ethernet networks differ on the data-link layer.
2. He types this in the Rich Text Box
3. After writing the feedback he clicks the send button
4. A feedback message is sent to the student.
5. The student reads out the feedback.

Exception Scenario:

Mr Marais mistakenly clicks send feedback button soon after he read the message.

1. An error message is displayed.
2. There is no feedback to be sent.

# **2.2 Class extraction**

# **2.2.1 Entity class extraction**

(a) Noun-extraction**:**

1. Concise problem definition:

A student login the application and is presented with a personal timetable showing a Computer Science course code with the class venue and the lecturer’s name. When the student hover over the lecturer’s name contact details of the lecturer are displayed. Clicking on any contact detail invokes the required programme to be open. All assessment due dates are displayed on the dashboard with a reminder system set in place to warn the student about upcoming assessments. Current marks are acquired from PeopleSoft marks system, current semester mark for each course is calculated and the required mark per upcoming assessment to reach the set goal mark is calculated. A report of the student’s predicted marks is presented to both the lecturer and the student. The student sends queries to the lecturer when they have any and the lecturer receives a notification message and provides feedback regarding the student’s query.

1. Identifying nouns:

A student login the application and is presented with a personal timetable showing a Computer Science course code with the class venue and the lecturer’s name. When the student hover over the lecturer’s name contact details of the lecturer are displayed. Clicking on any contact detail invokes the required programme to be open. All assessments due dates are displayed on the dashboard with a reminder system set in place to warn the student about upcoming assessments. Current marks are acquired from PeopleSoft marks system, current semester mark for each course is calculated and the required mark per upcoming assessment to reach the set goal mark is calculated. A report of the student’s predicted marks is presented to both the lecturer and the student. The student sends queries to the lecturer when they have any and the lecturer receives a notification message and provides feedback regarding the student’s query.

Identified nouns: Student, application, timetable, course, class, venue, lecturer, programme, assessments, dashboard, reminder system, marks, semester mark, goal mark, report, queries, notification, message, feedback.

Nouns that lie outside the problem boundary:

Application, class, venue, programme, dashboard

Abstract nouns:

Marks, notification, queries, feedback

Nouns to be used as candidate classes:

Student, course, lecturer, assessment

(b) CRC cards**:**

|  |  |
| --- | --- |
| **Class:** Student | |
| **Responsibility:**  -Login the application  -View timetable  -Set Goal Mark  -View performance or results recorded  -View assessments  -Send queries to the lecturer | **Collaboration:**  -Course  -Lecturer  -PeopleSoftMarksSystem |

|  |  |
| --- | --- |
| **Class:** Lecturer | |
| **Responsibility:**  -Set assessment date  -Set reminder  -Provide feedback | **Collaboration:**  -Course  -Student  -ReminderSystem |

|  |  |
| --- | --- |
| **Class:** Course | |
| **Responsibility:**  -Be available for the students | **Collaboration:**  -Student  -Lecturer |

|  |  |
| --- | --- |
| **Class:** Assessment | |
| **Responsibility:**  -Provide due date to the student | **Collaboration:**  -Course  -Student  -Lecturer |

|  |  |
| --- | --- |
| **Class:** Enrol | |
| **Responsibility:**  -Registers student for a course | **Collaboration:**  -Course  -Student |

|  |  |
| --- | --- |
| **Class:** Class | |
| **Responsibility:**  -Accommodates students for a lecture for a particular course | **Collaboration:**  -Lecturer  -Student  -Course |

## (c) Class diagram



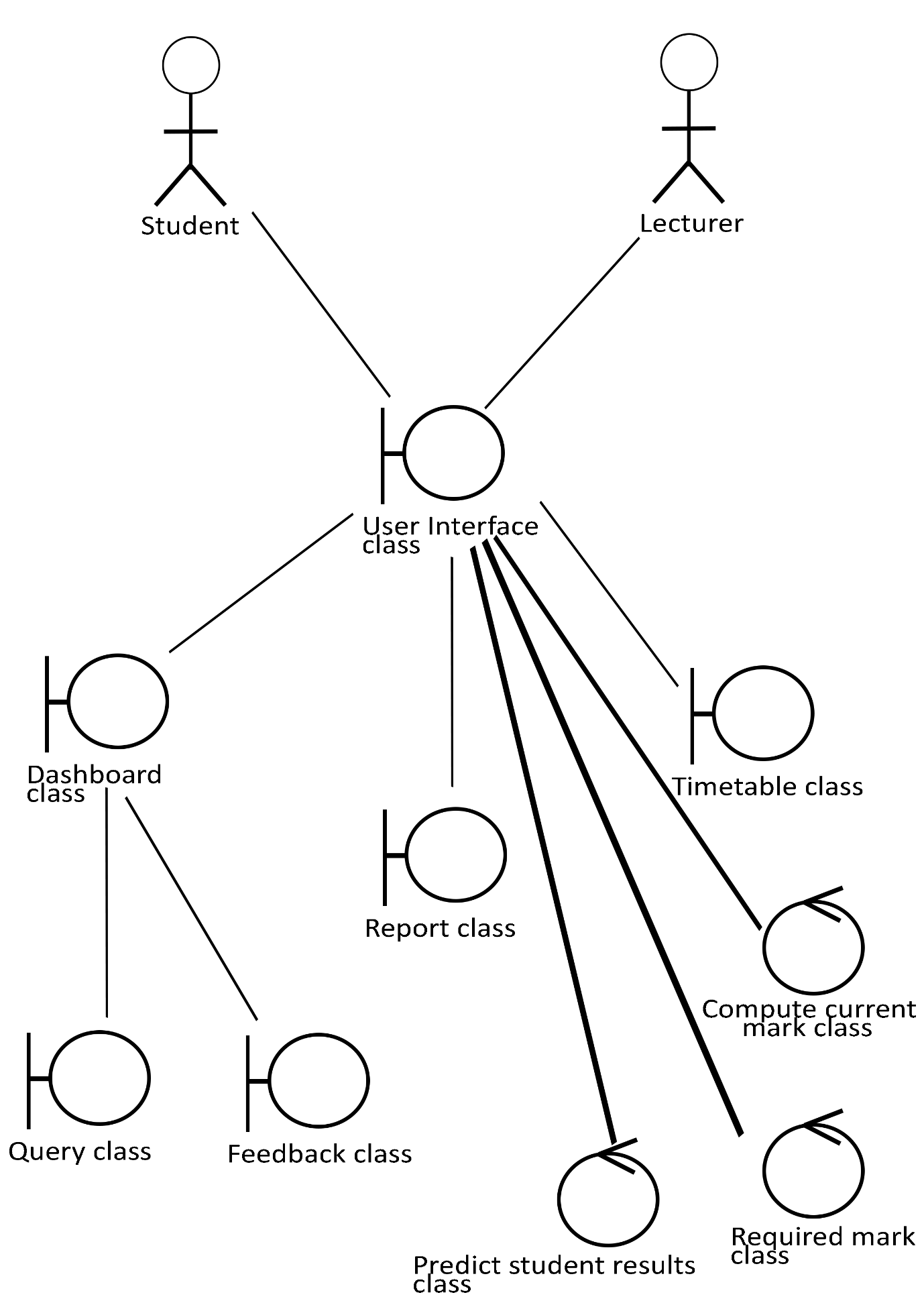
# **2.2.2 Boundary class extraction**

* User Interface class
* Timetable class
* Dashboard class
* Report class
* Query class
* Feedback class

# **2.2.3 Control class extraction**

* Compute current semester mark class
* Calculate required mark per upcoming assessment class
* Predict student results class

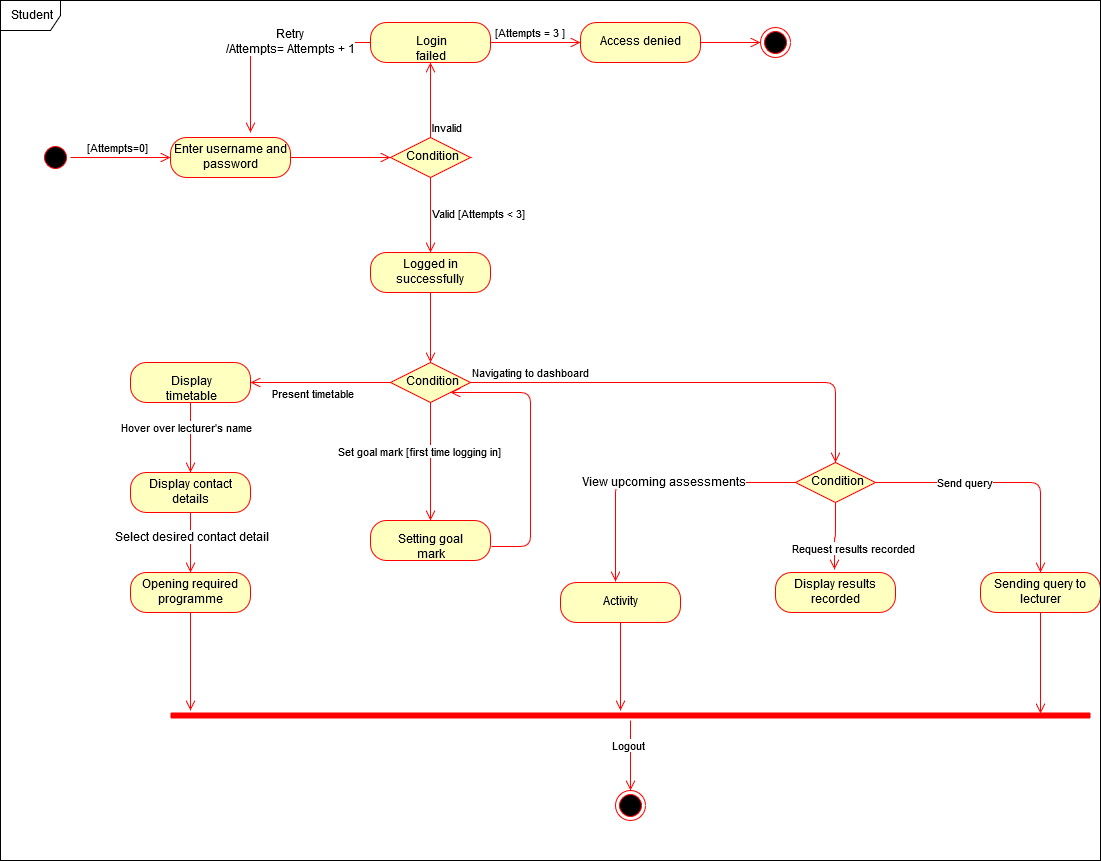
# 2.2.4Class diagram (showing stereotypes)



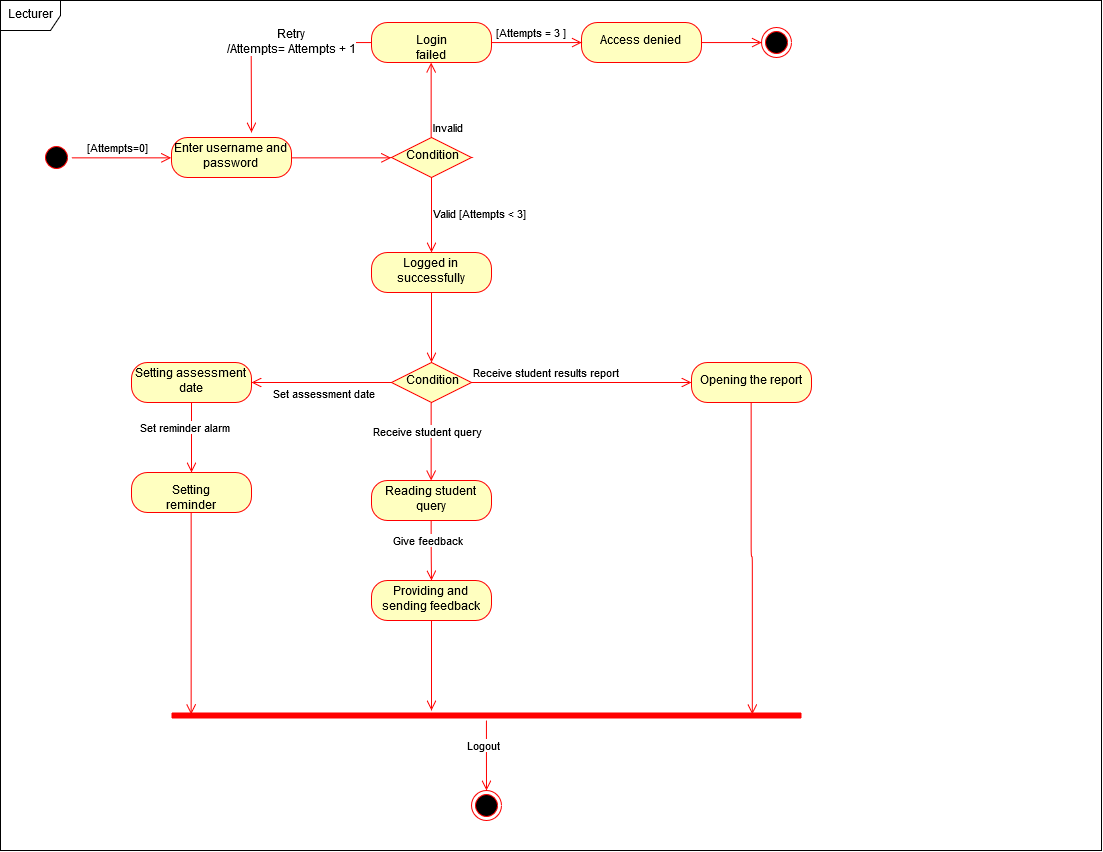
# 2.2.5Dynamic Modelling

# Statecharts

# Student class



# Lecturer class



# 2.3 Sequence diagrams for:

# 2.3.1 **Login** use case



# 2.3.2 **Set goal mark** use case



# 2.3.3 **Present timetable** use case



# 2.3.4 **Display lecturer's contact details** use case



# 2.3.5 **Set assessment date** use case and **Set reminder of upcoming assessment** use case



# 2.3.6 **Display assessment** use case



# 2.3.7 **Display results recorded** use case



# 2.3.8 **Provide report** use case



# 2.3.9 **Send query** use case and **Give feedback** use case



3 Software Project Management Plan

Table of Contents

**1 Overview**31

**1.1 Project Summary**31

**1.1.1 Purpose, Scope, and Objectives**31

**1.1.2 Assumptions and Constraints**31

**1.1.3 Project Deliverables**31

**1.1.4 Schedule and Budget Summary**31

**1.2 Evolution of the Project Management Plan**31

**2 Reference Materials**31

**3 Definitions and Acronyms**32

**4 Project Organization**32

**4.1 External Interfaces**32

**4.2 Internal Structure**32

**4.3 Roles and Responsibilities**32

**5 Managerial Process Plans**32

**5.1 Start-up plan**32

**5.1.1 Estimation Plan**32

**5.1.2 Staffing Plan**32

**5.1.3 Resource Acquisition Plan**32

**5.1.4 Project Staff Training Plan**32

**5.2 Work plan**33

**5.2.1-2 Work Activities and Schedule Allocation**33

**5.2.3 Resource Allocation**33

**5.2.4 Budget Allocation**33

**5.3 Control plan**33

**5.4 Risk Management Plan**33

**5.5 Project Close-out Plan**34

**6 Technical Process Plans**34

**6.1 Process Model**34

**6.2 Methods, Tools, and Techniques**34

**6.3 Infrastructure Plan**34

**6.4 Product Acceptance Plan**34

**7 Supporting Process Plan**34

**7.1 Configuration Management Plan**34

**7.2 Testing Plan**34

**7.3 Documentation Plan**34

**7.4-5 Quality Assurance Plan and Reviews and Audits Plan**35

**7.6 Problem Resolution Plan**35

**7.7 Subcontractor Management Plan**35

**7.8 Process Improvement Plan**35

**8 Additional Plans**35

**Security**35

**Training**35

**Maintenance**35

**1 Overview**

**1.1 Project Summary**

**1.1.1 Purpose, Scope, and Objectives**

The purpose of the project is developing a Timetable Application software product that will be used Computer Science department students in monitoring their weekly schedule. This software product will allow the student to login, view the weekly timetable and navigate to the dashboard checking all information regarding assessments including current marks of past assessments and dates of upcoming assessments. This software product will compute the current semester mark for each course and also calculate the required mark for each upcoming assessment in order to reach the goal mark that was set by the student on the first day of the semester. This software product will also be a platform of interaction between the student and the lecturer.

**1.1.2 Assumptions and Constraints**

Constraints include the following:

* The software product must be easy to use
* The software product must be usable on both mobile devices and desktops.
* Data should be easily accessible for the student.
* The software product must be reliable.
* The software product must be developed and delivered in such a way that further maintenance and additional functionality will be easy.
* The budget constraint must be met

**1.1.3 Project Deliverables**

A fully functional software product will be delivered in about 7-10 weeks

**1.1.4 Schedule and Budget Summary**

I will be working on each and every workflow. The duration and budget of each workflow are as follows:

* Requirements workflow (6 hour, R1 080)
* Analysis workflow (12 hours, R2 160)
* Design workflow (5 hours, R 900)
* Implementation workflow (rough estimate of 18 hours, R3 240)
* Testing workflow (3 hours, R540)

The total development time is 7-10 weeks R7 920

**1.2 Evolution of the Project Management Plan**

All changes in the project management plan must be agreed to by myself before they are implemented. All changes should be documented to keep the project management plan correct and up to date.

**2 Reference Materials**

None.

**3 Definitions and Acronyms**

UML – Unified Modeling Notation

**4 Project Organization**

**4.1 External Interfaces**

I will perform all the work in this project. I will make time to meet with the client at least once a week, informing him about progress and receive feedback about the current work and make necessary changes where needed.

**4.2 Internal Structure**

As mentioned in 4.1 that I will do all the work in this project, I represent the development team.

**4.3 Roles and Responsibilities**

I have performed the requirements workflow, so the analysis workflow follows. I will be responsible for performing the design workflow as well and implement all the classes and report artifacts, constructing all of them. In the process I will also ensure that the overall quality of the product meets the needs of the client.

**5 Managerial Process Plans**

**5.1 Start-up plan**

**5.1.1 Estimation Plan**

The total development time is estimated to be 7-10 weeks and the total internal cost is R7 740

**5.1.2 Staffing Plan**

I will be responsible as a system analyst, a designer, designing how the software product will perform all functions as per requirements of the system, a programmer to implement the design and develop the software product, a tester to test the code and check for any bugs that will be present and take on a managerial position as a manager managing all of the work being done during the course of this project.

**5.1.3 Resource Acquisition Plan**

The CASE tools needed for the development of this project are already acquired including the required software and hardware.

**5.1.4 Project Staff Training Plan**

Not needed.

**5.2 Work plan**

**5.2.1-2 Work Activities and Schedule Allocation**

First week – Determined all requirements artifacts of the software product.

Second week – Produced the business model indicating a use-case diagram along with all use-case scenarios pertaining the software product.

Third week –Have extracted all types of classes during the analysis workflow and submitted them to the client.

Fourth week – Producing the software project management plan and behavioural models, sequence diagram and state charts.

Fifth week – Elicit design artifacts and inspect them.

Sixth-Tenth week – All artifacts of the design workflow will be revisited and implemented, inspecting and integrating all classes, and conduct unit testing during the process. The software product will be developed during this period.

**5.2.3 Resource Allocation**

I will be focusing on each artifacts of the given workflow. I will be managing the work as well ensuring that I am doing everything correctly, that includes the overall quality of the software product. During the week every Tuesday I will sit and go through the work in progress and look into any problems present. Once a week, on Thursdays, I will arrange time to meet with the client to give a report and make the necessary modifications where needed. I will be responsible to ensure that the documentation is always updated throughout each workflow.

**5.2.4 Budget Allocation**

* Requirements workflow R1 080
* Analysis workflow R2 160
* Design workflow R 900
* Implementation workflow R3 240
* Testing workflow R 540
* Total R7 920

**5.3 Control plan**

I will constantly review the work ensuring that I am doing the work in accordance to the specification document and I will be held responsible for making sure that this project is done and delivered on time and within the specified budget. Every change to the software product will be well documented. I will conduct testing during each workflow and at the end.

**5.4 Risk Management Plan**

There are no (major) risks present for this software product. It is assumed that computer science students will be familiar with using the software product as it is easy to use, usable on both mobile devices and desktops and it is straight forward. Each workflow will be tested before commencing to the next. The software product will be tested extensively even before deployment.

**5.5 Project Close-out Plan**

Not applicable

**6 Technical Process Plans**

**6.1 Process Model**

The Iterative-and-Incremental Model

**6.2 Methods, Tools, and Techniques**

The software product will be developed with C# using the Iterative-and-Incremental Model.

**6.3 Infrastructure Plan**

The software product will be developed in a Windows Operating System on a laptop, using Microsoft Visio 2016 and Draw.io for UML diagrams and Visual Studio 2017 for developing the software product in C#.

**6.4 Product Acceptance Plan**

I will compile an acceptance criteria and communicate it to the client until we are on the same ground level. This acceptance criteria will be considered during development and be used to ensure that the software product accomplishes the acceptance tests. The acceptance of the software product will be achieved by following the steps of the Iterative-and-Incremental Model.

**7 Supporting Process Plan**

**7.1 Configuration Management Plan**

I will use a concurrent versioning system for storage and retrieval of all artifacts.

**7.2 Testing Plan**

Testing will be performed during the course of each workflow.

**7.3 Documentation Plan**

Documentation will be modified for every change that takes place in a given workflow and it will kept updated during each workflow.

**7.4-5 Quality Assurance Plan and Reviews and Audits Plan**

I will be testing the code of the software product and do integration testing as well and conducting extensive testing of the software product.

**7.6 Problem Resolution Plan**

I will monitor and attend problems that will arise during the development of the software product.

**7.7 Subcontractor Management Plan**

Not applicable.

**7.8 Process Improvement Plan**

Not applicable.

**8 Additional Plans**

**Security plan**

A user will be required to enter a password and be authenticated in order to use the software product.

**Training plan**

There will be no training needed for this software product. A short demonstration video of how to use the software product will be shot and made available.

**Maintenance plan**

For any maintenance needed for the software product I should be contacted and I will attend to it.

# 4 **Quotation for the system**

I will be working on each and every workflow. The duration and budget of each workflow are as follows:

* Requirements workflow (6 hour, R1 080)
* Analysis workflow (12 hours, R2 160)
* Design workflow (5 hours, R 900)
* Implementation workflow (rough estimate of 18 hours, R3 240)
* Testing workflow (3 hours, R540)

Total: R7 920

General expenses

* Water and electricity: 5% of the total amount of development time. This will be billed R396

Total amount: R 8 316

# **5 Design Documentation**

# Object Oriented Design

# Completion of the class diagram



# Performing a detailed design

# **Pseudo code**

## For login functionality:

Declare iAttempts variable and initialize it to 0

Declare sUsername variable

Declare sPassword variable

Read and store the value of username variable from the username textbox

Read and store the value of password variable from the password textbox

foreach studentnumber in a collection StudentNumbers

{

if sUsername matches the studentnumber

{

Call the LoadStudent(studentnumber) function and assign it to an instance of the student

if sPassword matches the password the an instance of the student

{

Create an instance of the main form

Hide the current form

Set the text of the form to the student’s student number

Clear all text boxes

Show/Display the main form

Close the login form

break;

}

}

}

Display a message "Incorrect username or password.", "Error" if the user could not login

Increment the iAttempts variable

Clear all text boxes

if iAttempts is equal to 3

{

Close the login form

Re-direct to the landing form

}

## For set goal mark functionality:

Declare Course variable

Declare GoalMark variable

Set Course to the selected item from the Courses dropdown list

Set GoalMark to the value of the spin edit

**If** GoalMark < 50 **then**

Display a warning message addressing that any mark below 50% is a fail and will not be accepted as a goal mark

**else**

{

Save the goal mark

Display a message that the goal mark was saved

Disable the GoalMark section

}

## For set assessment date functionality:

Declare Course variable

Declare AssessmentType variable

Declare Date variable

Set Course to the selected item from the Courses dropdown list

Set AssessmentType to the selected item from the Courses dropdown list

Set Date to the specified date in the calendar component

Save()

Display a message that the assessment was set and saved

## For performing calculations and predicting future results:

Get student current marks for completed assessments

**If** mark is between 0% and 39% **then**

Show a red icon symbolizing a fail

**else If** mark is between 40% and 50% **then**

Show a yellow icon symbolizing danger

**else If** mark is between 51% and 59% **then**

Show a blue icon symbolizing that the student is safe enough

**else**

Show a green icon symbolizing that the student is completely safe

# 6 Discussion of development

I will be developing this software product as a Desktop Windows based system with C# using Visual Studio 2017. Besides me being more familiar and comfortable with developing desktop applications than websites, a desktop system will be much easier to develop and maintain. A desktop application has better performance in terms of speed and memory. For a desktop application, security of the software will not be at a higher risk like there is too many security issues and reliability issues for a website application and it may be also disadvantaged by restrictions of running in a web browser.

brief description (half a page) of how the business objects “link” with the interface layer and the data access layer.

# 7 Record of hours spent on the project

* Requirements workflow (1 hour)
* Analysis workflow (14 hours)
* Use case descriptions and diagrams [5 hours]
* SPMP [3 hours]
* Statecharts [2 hours]
* Sequence diagrams [4 hours]
* Design workflow (2,5 hours)
* Detailed class diagram [2 hours]
* Pseudo code [00:25min]
* Discussion of development [00:5min]
* Implementation workflow (53 hours)
* Testing workflow (15 hours)