

## Module ICT3715

### INFORMATION AND COMMUNICATION TECHNOLOGY PROJECT

STUDENT NUMBER (Student completes)									
6	5	6	1	4	9	6	8	-	-

IDENTITY NUMBER (Student completes)													
9	3	0	2	2	4	5	0	5	3	0	8	8	

No handwritten assignments will be accepted.

#### **INSTRUCTIONS:**

Complete this Front Page (page 1)

Complete the Plagiarism Pledge (page 2)

Complete the header and footer with your own information

Add your practical system content to the document

Remove everything that is in brackets []

Make sure that your Table of Content is updated

Save the document as a **MS Word Document**, e.g., 12345678\_ICT3715\_Portfolio (replace 1234568 with your student number)

When you are done submit via myModules 2022 on the Module Site under **Project Portfolio Document**

Keep a copy of the original should there be problem with the upload

## PLAGIARISM PLEDGE BY THE STUDENT

1. I have read Unisa's plagiarism policy.
2. I understand Unisa's plagiarism policy.
3. I agree to abide by Unisa's plagiarism policy.
4. I have read the direct copying, plagiarism, and "patch-writing" document.
5. I understand what direct copying, plagiarism, and "patch-writing" is.
6. I undertake to avoid copying directly, plagiarism and patch writing.
7. All academic work, written or otherwise, that I submit is expected to be the result of my own skill and labour.
8. I understand that, if I am guilty of the infringement of breach of copyright/plagiarism or unethical practice, I will be subject to the applicable disciplinary code as determined by Unisa.
9. The marker has the right to refuse to assess the assignment and the system if plagiarism is detected.

Student name and Surname: Ralph Botes

Student number: 65614968



Student signature:

05-10-2022

Date:

System | Online examination file submission system

Due to the pandemic, the University of South Africa (UNISA) has decided to *conduct all examinations in an online environment.*

The *purpose of this project* is then to look at the digital environment and *design and develop a simulation of such a system, using the real-life scenario and environment.*

You thus must design and develop an online examination file submission system, that can be presented to the University to be used as an alternative to the current system.

**Note:** You are not allowed to develop any other system or use any other data that was not prescribed or provided to you.

## Table of Contents

<b>User Guide (Manual).....</b>	<b>6</b>
<b>Testing &amp; Implementation Plan / Installation Manual.....</b>	<b>7</b>
<b>Business Plan.....</b>	<b>8</b>
<b>Assignment 3.....</b>	<b>9</b>
<b>Section A = MIS Reports [20] .....</b>	<b>9</b>
Create the Daily MIS report (5) .....	9
Create the Weekly MIS report (5) .....	19
Create one other MIS report (5) .....	30
Create one other MIS report (5) .....	37
<b>Section B = Graphical User Interfaces [20].....</b>	<b>39</b>
Student login GUI (7) .....	39
Lecturer login GUI (7).....	42
Exam Department (Staff member) login GUI (6).....	45
User failed Login .....	47
<b>Assignment 2.....</b>	<b>48</b>
<b>Section A = Database [20] .....</b>	<b>48</b>
Create the database for your system and importing the data from your .csv file (20) .....	48
<b>Section B = Backup and Recovery for the Database and Programming code [10] .....</b>	<b>81</b>
Database (5).....	81
Programming code & Portfolio (assignments) (20) .....	87
<b>Assignment 1.....</b>	<b>93</b>
<b>Section A [4].....</b>	<b>93</b>
Programming Languages (2).....	93
Database (2).....	93
<b>Section B [10].....</b>	<b>93</b>

Cleaning the data .....	93
<b>Section C [10] .....</b>	<b>94</b>
Activity Diagram (7).....	94
ERD Diagram (3).....	97
<b>Section Backup and Recovery for the Database and Programming code [6] .....</b>	<b>97</b>
Backup and Recovery Software for the Database (3).....	97
Backup and Recovery process for the Programming code and your Portfolio (assignments) (3) .....	97
<b>Module Leader Approval.....</b>	<b>98</b>
<b>Request .....</b>	<b>98</b>
<b>Approval .....</b>	<b>99</b>

## User Guide (Manual)

[65614968 ICT3715 User Guide \(Manual\) Task4.docx](#)

## Testing & Implementation Plan | Installation Manual

[65614968 ICT3715 Implementation Plan \(Installation Manual\) Task 3.2.docx](#)

## Business Plan

[65614968 ICT3715 Business Plan and Testing Task 3.1.docx](#)

## Assignment 3

### Section A = MIS Reports [20]

Create the Daily MIS report (5)

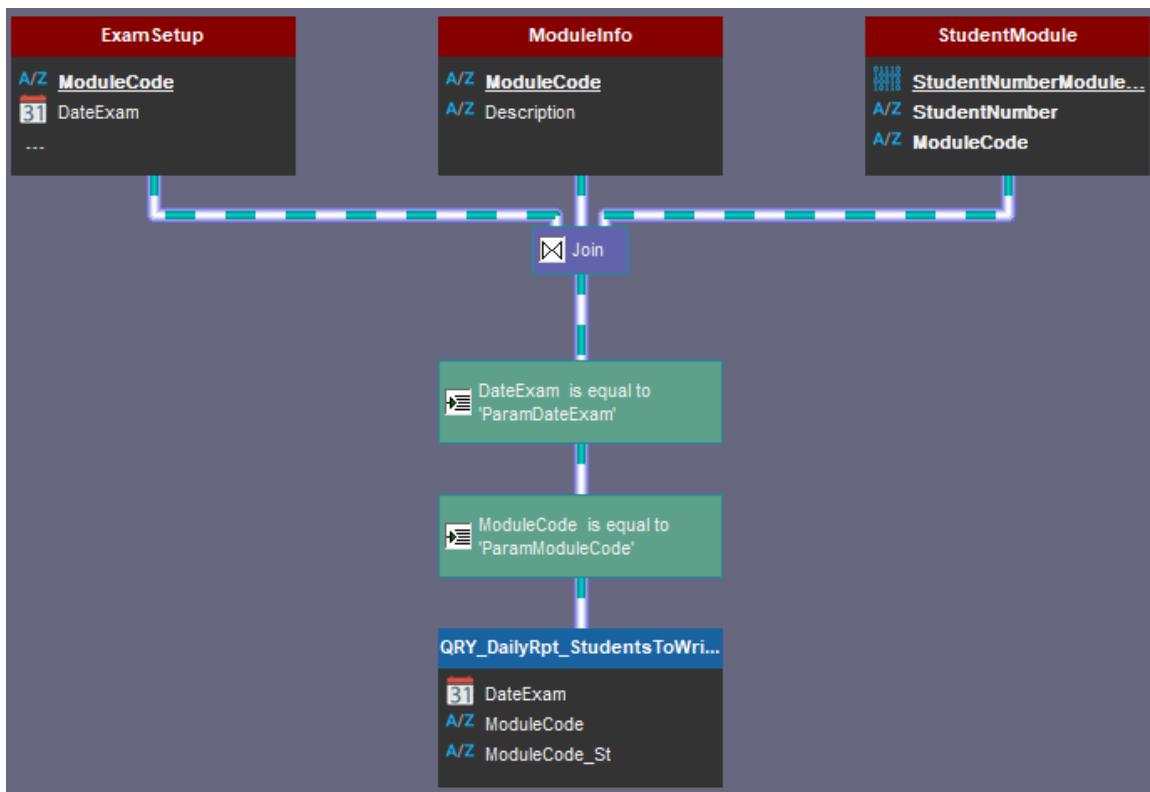
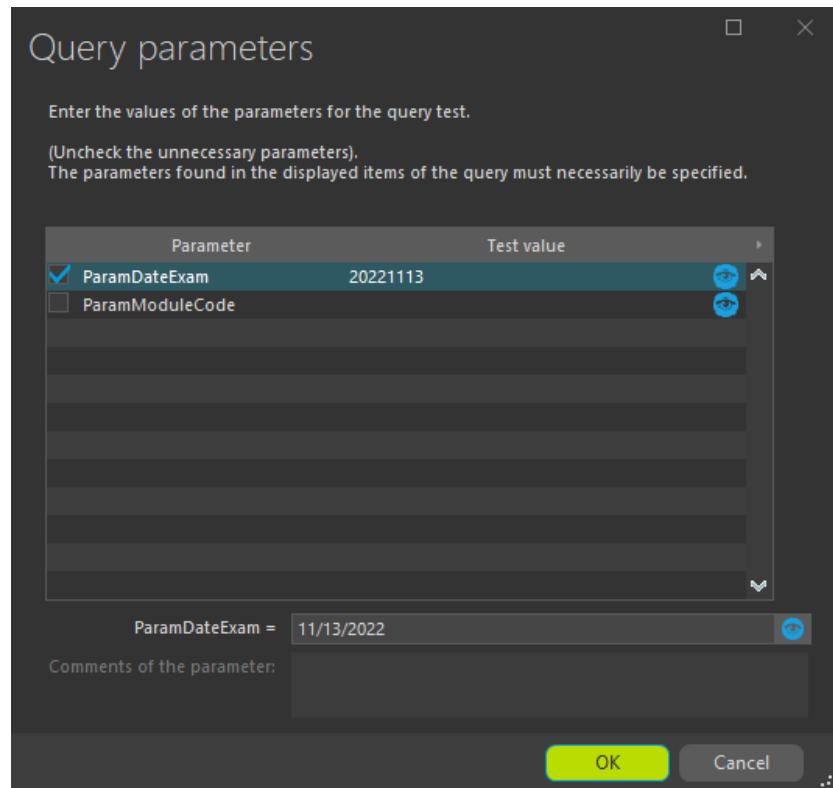
#### STUDENTS EXPECTED TO WRITE QUERY

##### Reason

This query is used to get each record of students who are expected to write on a specific day. The date is passed as a parameter and all the records found for that date is returned.

##### Query

```
SQL code of QRY_DailyRpt_StudentsToWrite * \ 
SELECT
    ExamSetup.DateExam AS DateExam,
    ExamSetup.ModuleCode AS ModuleCode,
    StudentModule.ModuleCode AS ModuleCode_St
FROM
    ExamSetup,
    ModuleInfo,
    StudentModule
WHERE
    ModuleInfo.ModuleCode = StudentModule.ModuleCode
    AND ExamSetup.ModuleCode = ModuleInfo.ModuleCode
    AND
    (
        ExamSetup.DateExam = {ParamDateExam}
        AND ExamSetup.ModuleCode = {ParamModuleCode}
    )
```

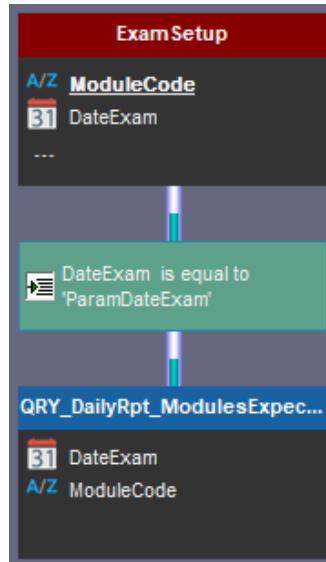
Screen dumps

## **MODULES EXPECTED TO BE WRITTEN QUERY**

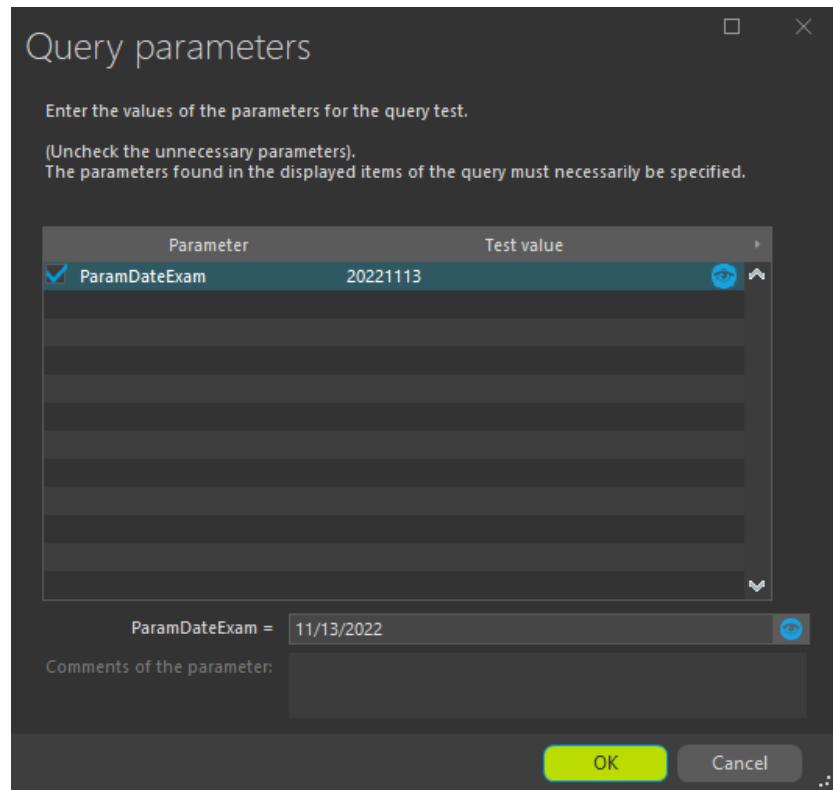
### Reason

This query is used to get each record of modules that are expected to be written on a specific day. The date is passed as a parameter and all the records found for that date is returned.

## Query



```
SQL code of QRY_DailyRpt_ModulesExpected
SELECT
    ExamSetup.DateExam AS DateExam,
    ExamSetup.ModuleCode AS ModuleCode
FROM
    ExamSetup
WHERE
    ExamSetup.DateExam = {ParamDateExam}
```

Screen dumps

DateExam	ModuleCode
13/11/2022	ICT1114
13/11/2022	CHE2611
13/11/2022	ENG1014

Right-click to print, export records, etc.

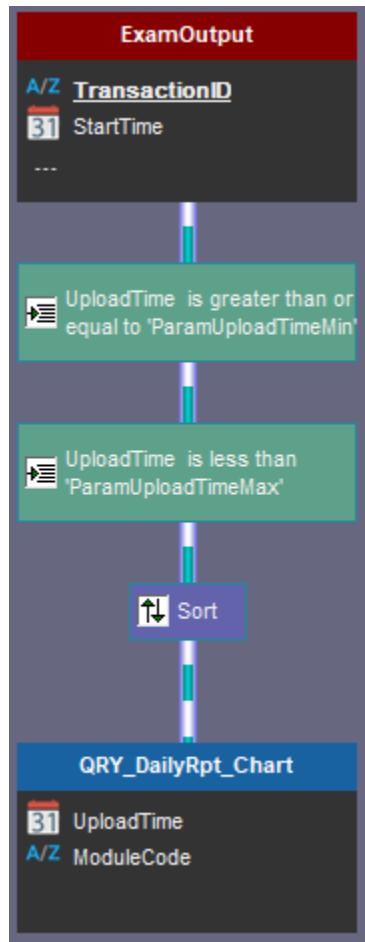
Export to an HFSQl file...

Close

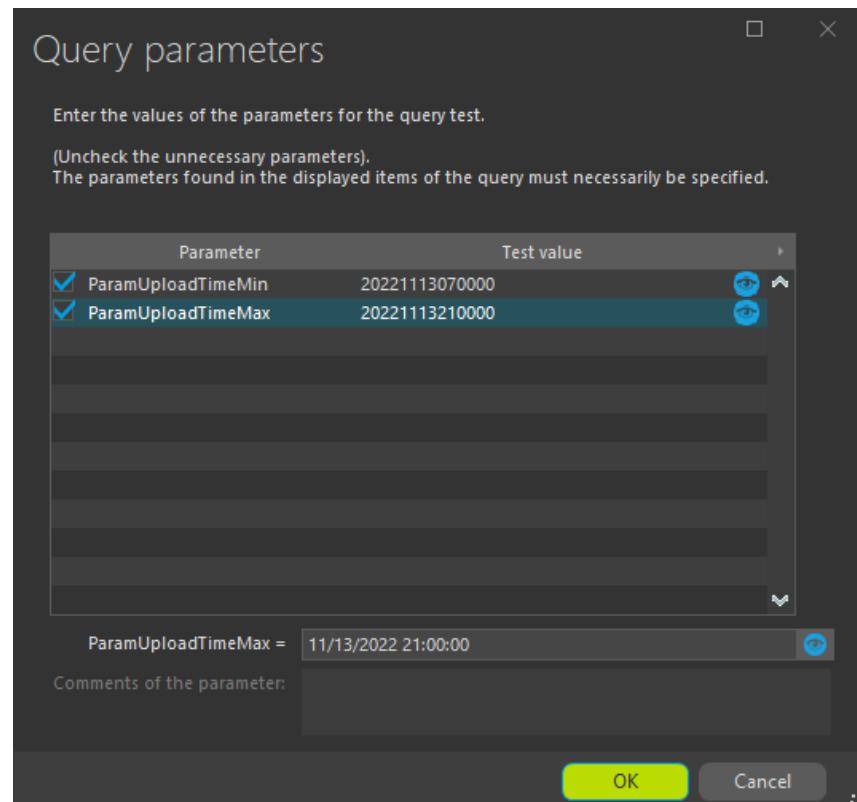
## POPULATE CHART QUERY

### Reason

A chart will be added that will view all the submissions for a specific date between 7:00 and 21:00. A function will be used to group the submissions by the hour. For example: Between 9:00 and 10:00 there were 352 submissions.

Query

```
SQL code of QRY_DailyRpt_Chart  *
SELECT
    ExamOutput.UploadTime AS UploadTime,
    ExamOutput.ModuleCode AS ModuleCode
FROM
    ExamOutput
WHERE
    ExamOutput.UploadTime >= {ParamUploadTimeMin}
    AND ExamOutput.UploadTime < {ParamUploadTimeMax}
ORDER BY
    UploadTime ASC
```

Screen dumps

The window title is "Preview of QRY\_DailyRpt\_Chart". It displays a message: "976 records correspond to your query." A table lists the data:

UploadTime	ModuleCode
13/11/2022 10:19	CHE2611
13/11/2022 10:19	ENG1014
13/11/2022 10:20	CHE2611
13/11/2022 10:20	ICT1114
13/11/2022 10:20	ENG1014
13/11/2022 10:20	ENG1014
13/11/2022 10:21	ICT1114
13/11/2022 10:21	CHE2611
13/11/2022 10:21	ICT1114
13/11/2022 10:22	ICT1114
13/11/2022 10:23	CHE2611
13/11/2022 10:24	ENG1014
13/11/2022 10:25	ICT1114
13/11/2022 10:25	CHE2611

Below the table, there is a message: "Right-click to print, export records, etc." and a "Export to an HFSQ file..." button. At the bottom are "Close" and "...:." buttons.

## STAFF MODULE INFO QUERY

### Reason

A module code is passed as a parameter to this query, then the module leader for the module's relevant information is returned. Basically, get the module leaders info for module code AST2652.

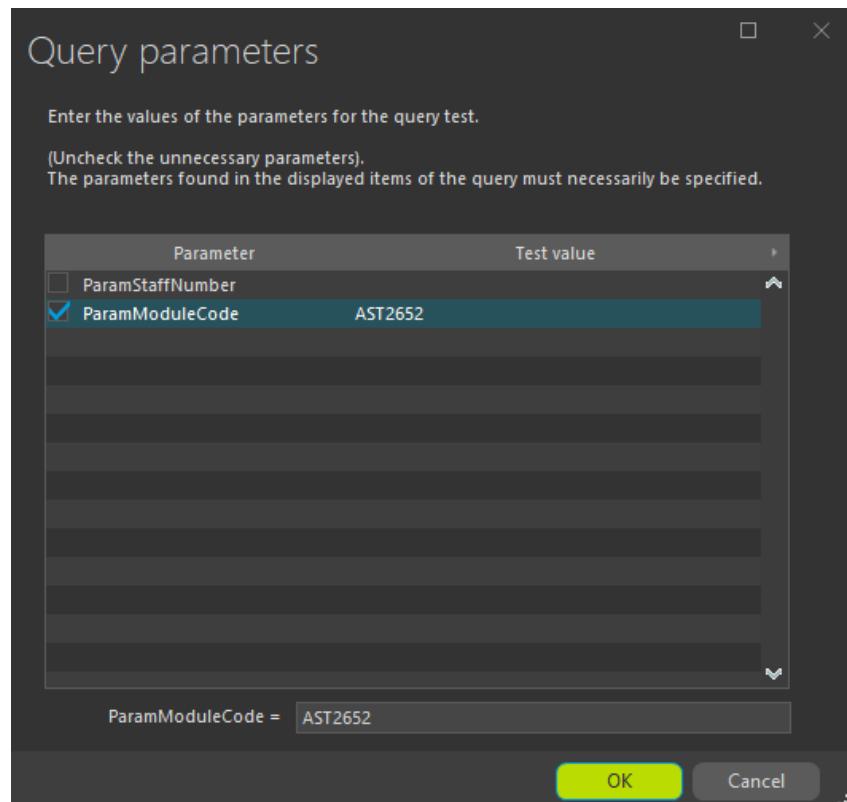
### Query

The screenshot shows a database interface with two main sections. The top section is titled "SQL code of QRY\_StaffModules" and contains the following SQL query:

```
SELECT
    ModuleLeader.ModuleCode AS ModuleCode,
    ExamSetup.DateExam AS DateExam,
    ModuleLeader.StaffNumber AS StaffNumber,
    ModuleInfo.Description AS Description,
    StaffInfo.Name AS LecturerName,
    StaffInfo.Email AS LecturerEmail
FROM
    Module_leader
```

The bottom section is titled "QRY\_StaffModules" and shows the resulting table structure with columns: ModuleCode (A/Z) and DateExam (B1). There is also an ellipsis (...).

```
SQL code of QRY_StaffModules
SELECT
    ModuleLeader.ModuleCode AS ModuleCode,
    ExamSetup.DateExam AS DateExam,
    ModuleLeader.StaffNumber AS StaffNumber,
    ModuleInfo.Description AS Description,
    StaffInfo.Name AS LecturerName,
    StaffInfo.Email AS LecturerEmail
FROM
    ModuleLeader,
    ExamSetup,
    ModuleInfo,
    StaffInfo
WHERE
    ModuleLeader.ModuleCode = ExamSetup.ModuleCode
    AND
    ModuleInfo.ModuleCode = ModuleLeader.ModuleCode
    AND
    StaffInfo.StaffNumber = ModuleLeader.StaffNumber
    AND
    (
        ModuleLeader.StaffNumber = {ParamStaffNumber} AND
        ModuleLeader.ModuleCode = {ParamModuleCode}
    )
```

Screen dumps

The window title is "Preview of QRY\_StaffModules". It displays a single record from a database table:

ModuleCode	DateExam	StaffNumber	Description	LecturerName	LecturerEmail
AST2652	10/11/2022	5017037	Accounting Sciences II	JH Nel	jh.nel@unisa.ac.za

Text at the bottom left: "Right-click to print, export records, etc." and "Export to an HFSQ file...". Bottom buttons: "Close" and a printer icon.

## Create the Weekly MIS report (5)

**TOTAL UNIQUE UPLOADS QUERY**Reason

This query will be used to get the total number of unique uploads for the past 7 days. A minimum and maximum date is passed as parameters.

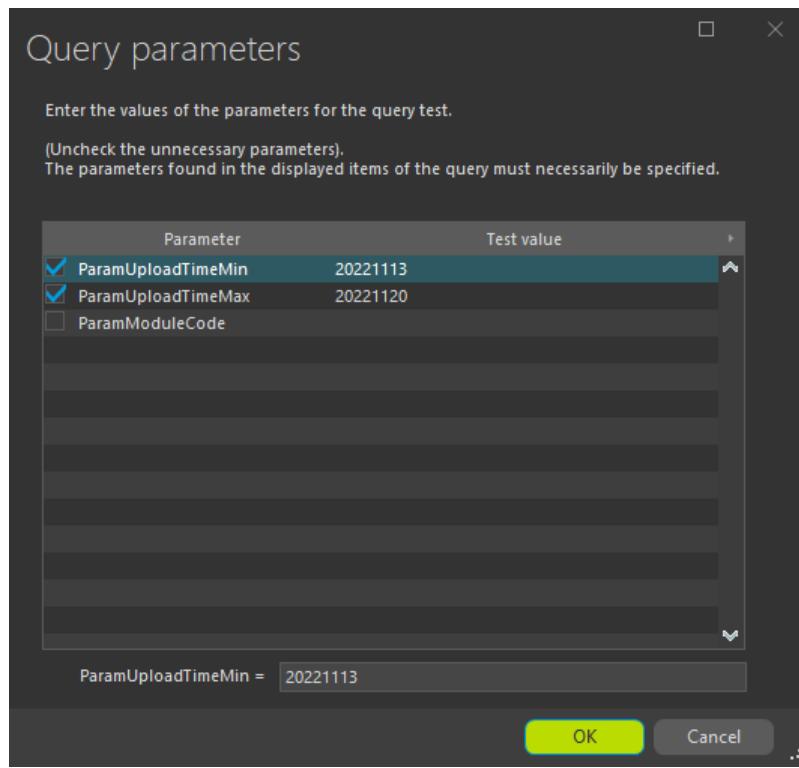
Query

**SQL code of QRY\_WeeklyRpt\_UnderUploads**

```
SELECT
    ExamOutput.ModuleCode AS ModuleCode
FROM
    ExamOutput
WHERE
(
    ExamOutput.UploadTime >=
    {ParamUploadTimeMin}
    AND ExamOutput.UploadTime <=
    {ParamUploadTimeMax}
)
ORDER BY
    ModuleCode ASC
```

**SQL code of QRY\_WeeklyRpt\_UnderUploads**

```
SELECT
    ExamOutput.ModuleCode AS ModuleCode
FROM
    ExamOutput
WHERE
(
    ExamOutput.UploadTime >= {ParamUploadTimeMin}
    AND ExamOutput.UploadTime <= {ParamUploadTimeMax}
    AND ExamOutput.ModuleCode = {ParamModuleCode}
)
ORDER BY
    ModuleCode ASC
```

Screen dumps

The dialog box is titled "Preview of QRY\_WeeklyRpt\_UniqueUploads" and displays the message "5829 records correspond to your query." A table view shows a single column of data, all entries being "CHE2611". At the bottom, there is a message "Right-click to print, export records, etc.", a "Export to an HFSQI file..." button, a printer icon, and "Close" and "...:." buttons.

## MODULES COMPLETED QUERY

### Reason

This query will be used to get the total number of modules uploaded for the past 7 days. A minimum and maximum date is passed as parameters.

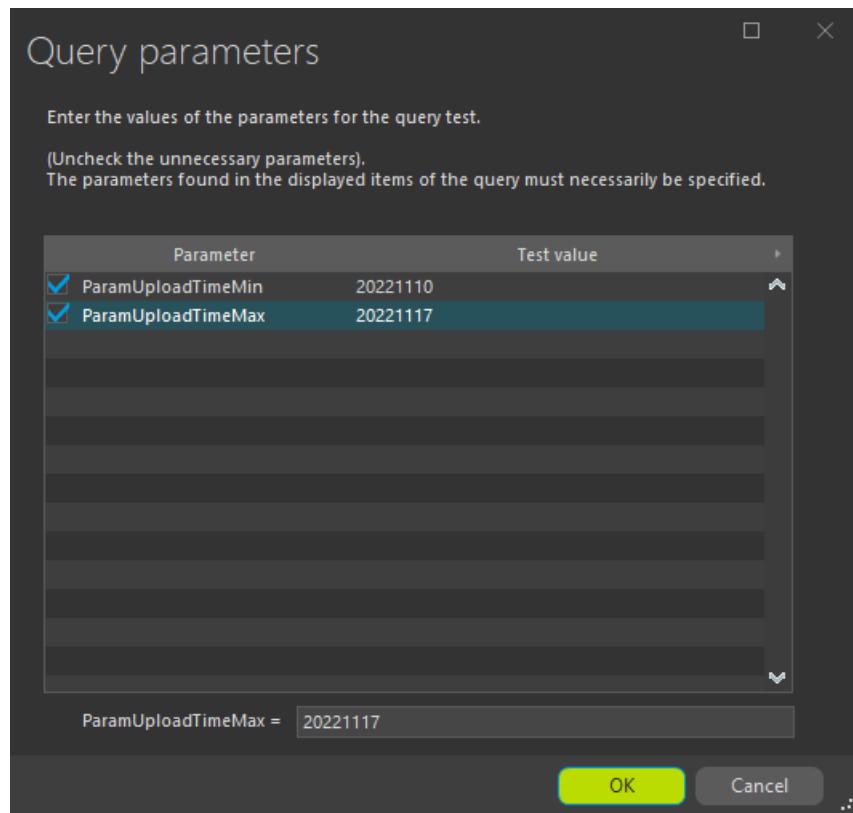
### Query

```
SQL code of QRY_WeeklyRpt_ModulesCompleted

SELECT
    ExamOutput.ModuleCode AS ModuleCode
FROM
    ExamOutput
WHERE
    ExamOutput.UploadTime >= {ParamUploadTimeMin}
    AND
    ExamOutput.UploadTime <= {ParamUploadTimeMax}
ORDER BY
    ModuleCode ASC
```

```
SQL code of QRY_WeeklyRpt_ModulesCompleted

SELECT
    ExamOutput.ModuleCode AS ModuleCode
FROM
    ExamOutput
WHERE
    ExamOutput.UploadTime >= {ParamUploadTimeMin} AND
    ExamOutput.UploadTime <= {ParamUploadTimeMax}
ORDER BY
    ModuleCode ASC
```

Screen dumps

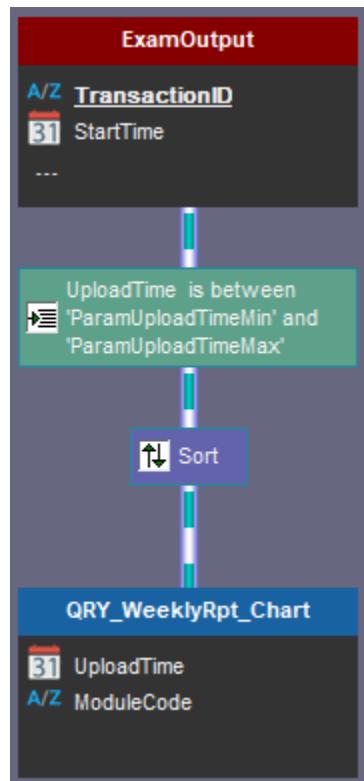
The screenshot shows a preview window titled 'Preview of QRY\_WeeklyRpt\_ModulesCompleted'. It displays a list of 8134 records corresponding to the query. The first few entries in the list are 'AST2652'. At the top, it says '8134 records correspond to your query.' Below the list, there is a note 'Right-click to print, export records, etc.' and a button 'Export to an HFSQ file...'. At the bottom, there is a printer icon and a 'Close' button.

## CHART DATES WRITTEN FOR WEEK QUERY

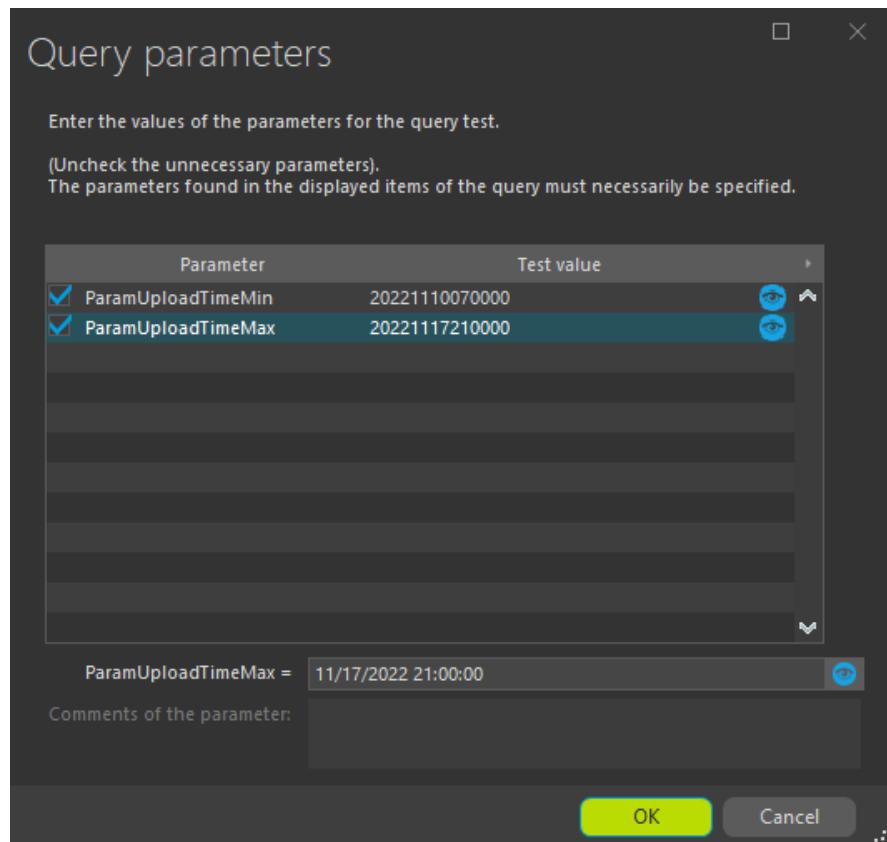
### Reason

A minimum and maximum date and time is passed to this query, the number of records for that week is returned. A function will be used to group the data by date to fill the chart.

### Query



```
SQL code of QRY_WeeklyRpt_Chart
SELECT
    ExamOutput.UploadTime AS UploadTime,
    ExamOutput.ModuleCode AS ModuleCode
FROM
    ExamOutput
WHERE
    ExamOutput.UploadTime BETWEEN {ParamUploadTimeMin} AND {ParamUploadTimeMax}
ORDER BY
    UploadTime ASC
```

Screen dumps

The screenshot shows a preview window titled 'Preview of QRY\_WeeklyRpt\_Chart'. It displays a list of 8767 records corresponding to the query. The table has columns for 'UploadTime' and 'ModuleCode'. The data shows various upload times (e.g., 10/11/2022 10:18, 10:20) and module codes (e.g., AST2652, ENG1011, ICT1111, MAT2612, ICT2612). At the bottom, there are buttons for 'Right-click to print, export records, etc.' and 'Export to an HFSQ file...', along with a 'Close' button.

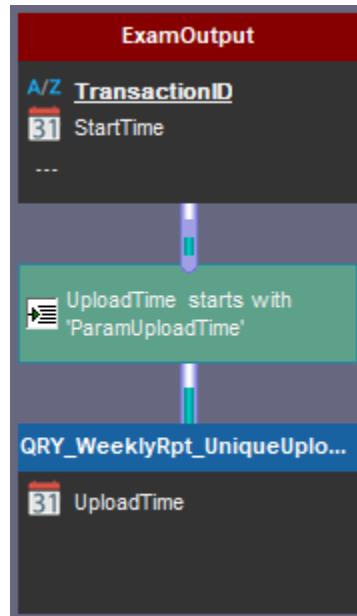
UploadTime	ModuleCode
10/11/2022 10:18	AST2652
10/11/2022 10:18	AST2652
10/11/2022 10:18	ENG1011
10/11/2022 10:18	ENG1011
10/11/2022 10:18	ICT1111
10/11/2022 10:18	MAT2612
10/11/2022 10:20	ICT2612
10/11/2022 10:20	AST2652
10/11/2022 10:20	ICT1111
10/11/2022 10:21	ENG1011
10/11/2022 10:21	ICT1111
10/11/2022 10:22	ICT1111
10/11/2022 10:22	ICT1111
10/11/2022 10:22	ENG1011
10/11/2022 10:22	ICT2612

## UNIQUE UPLOADS BY DATE QUERY

### Reason

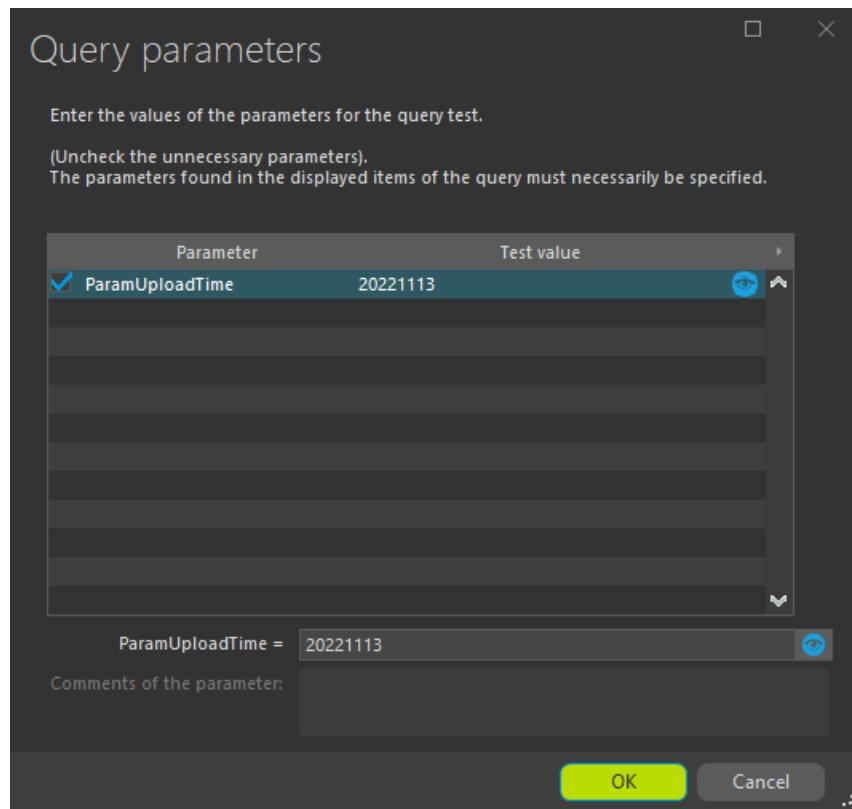
A date is passed to this query as a parameter. The query returns all the uploads for that date. A function will be used to group the data by the hour to fill the chart. Example: total uploads between 9:00 and 10:00 on the 13<sup>th</sup> of November 2022 is 324.

### Query



### SQL code of QRY\_WeeklyRpt\_UniqueUploadsByDate

```
SELECT
    ExamOutput.UploadTime AS UploadTime
FROM
    ExamOutput
WHERE
    ExamOutput.UploadTime LIKE {ParamUploadTime}%
```

Screen dumps

Preview of QRY_WeeklyRpt_UncqueUploadsByDate	
976 records correspond to your query.	
	UploadTime
	13/11/2022 11:01
	13/11/2022 11:13
	13/11/2022 11:19
	13/11/2022 11:53
	13/11/2022 10:50
	13/11/2022 11:01
	13/11/2022 11:58
	13/11/2022 11:27
	13/11/2022 11:41
	13/11/2022 11:08
	13/11/2022 11:45
	13/11/2022 10:54
	13/11/2022 11:59
	13/11/2022 11:10
	13/11/2022 11:04
Right-click to print, export records, etc.	
<a href="#">Export to an HFSQI file...</a>	

## STAFF MODULE INFO QUERY (RE-USSED)

### Reason

A module code is passed as a parameter to this query, then the module leader for the module's relevant information is returned. Basically, get the module leaders info for module code AST2652.

### Query

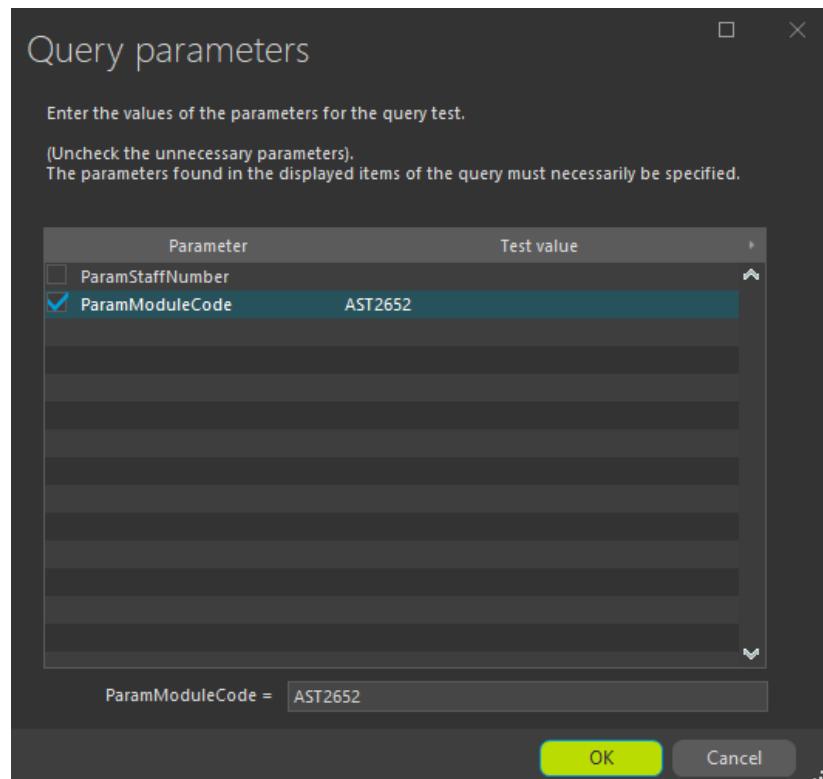
The screenshot shows a database interface with two main sections. The top section is titled "SQL code of QRY\_StaffModules" and contains the following SQL query:

```
SELECT
    ModuleLeader.ModuleCode AS ModuleCode,
    ExamSetup.DateExam AS DateExam,
    ModuleLeader.StaffNumber AS StaffNumber,
    ModuleInfo.Description AS Description,
    StaffInfo.Name AS LecturerName,
    StaffInfo.Email AS LecturerEmail
FROM
    ModuleLeader
```

The bottom section is titled "QRY\_StaffModules" and displays the results of the query as a table with three columns: "ModuleCode" (sorted A/Z), "DateExam" (with a count of 31), and an ellipsis (...). The "DateExam" column has a red icon next to it.

A/Z	ModuleCode	31	DateExam
			...

```
SQL code of QRY_StaffModules
SELECT
    ModuleLeader.ModuleCode AS ModuleCode,
    ExamSetup.DateExam AS DateExam,
    ModuleLeader.StaffNumber AS StaffNumber,
    ModuleInfo.Description AS Description,
    StaffInfo.Name AS LecturerName,
    StaffInfo.Email AS LecturerEmail
FROM
    ModuleLeader,
    ExamSetup,
    ModuleInfo,
    StaffInfo
WHERE
    ModuleLeader.ModuleCode = ExamSetup.ModuleCode
    AND
    ModuleInfo.ModuleCode = ModuleLeader.ModuleCode
    AND
    StaffInfo.StaffNumber = ModuleLeader.StaffNumber
    AND
    (
        ModuleLeader.StaffNumber = {ParamStaffNumber} AND
        ModuleLeader.ModuleCode = {ParamModuleCode}
    )
```

Screen dumps

Preview of QRY_StaffModules					
1 record corresponds to your query.					
ModuleCode	DateExam	StaffNumber	Description	LecturerName	LecturerEmail
AST2652	10/11/2022	5017037	Accounting Sciences II	JH Nel	jh.nel@unisa.ac.za

Right-click to print, export records, etc.

Export to an HFSQ file...

Close

Create one other MIS report (5)

### STAFF MODULE INFO QUERY (RE-USING)

#### Reason

A module code is passed as a parameter to this query, then the module leader for the module's relevant information is returned. Basically, get the module leaders info for module code AST2652. This data will be used to fill a chart.

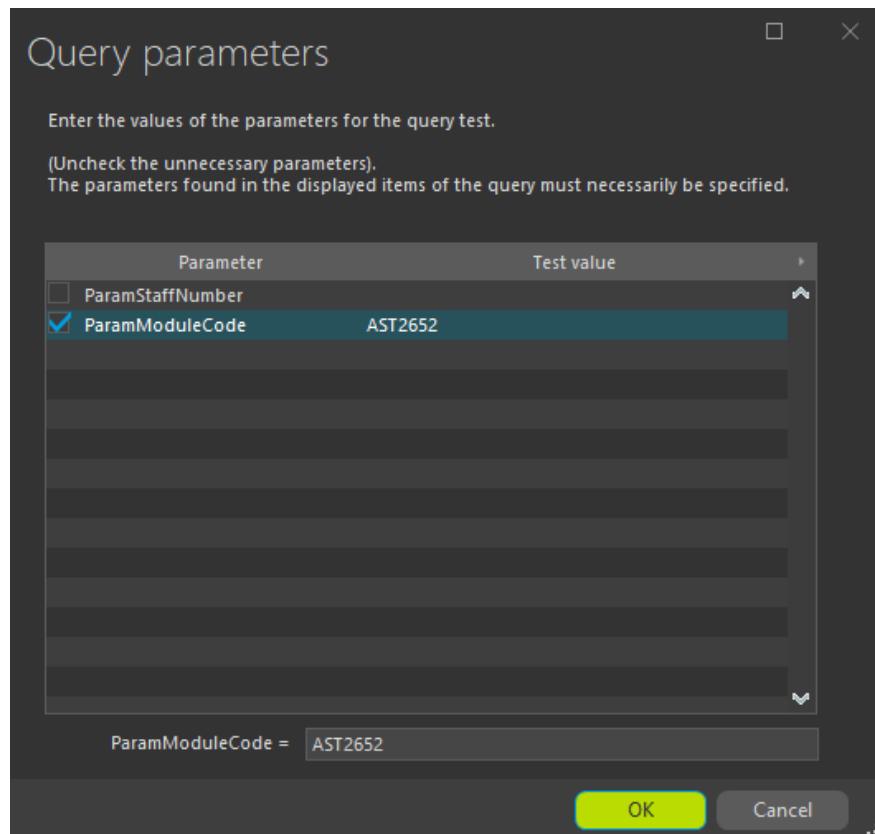
#### Query

The screenshot shows a database interface with two main sections. The top section is titled "SQL code of QRY\_StaffModules" and contains the following SQL query:

```
SELECT
    ModuleLeader.ModuleCode AS ModuleCode,
    ExamSetup.DateExam AS DateExam,
    ModuleLeader.StaffNumber AS StaffNumber,
    ModuleInfo.Description AS Description,
    StaffInfo.Name AS LecturerName,
    StaffInfo.Email AS LecturerEmail
FROM
    ModuleLeader
```

The bottom section is titled "QRY\_StaffModules" and shows the resulting table structure with columns: "ModuleCode" (A/Z), "DateExam" (B1), and an ellipsis (...). The "DateExam" column has a red warning icon.

```
SQL code of QRY_StaffModules
SELECT
    ModuleLeader.ModuleCode AS ModuleCode,
    ExamSetup.DateExam AS DateExam,
    ModuleLeader.StaffNumber AS StaffNumber,
    ModuleInfo.Description AS Description,
    StaffInfo.Name AS LecturerName,
    StaffInfo.Email AS LecturerEmail
FROM
    ModuleLeader,
    ExamSetup,
    ModuleInfo,
    StaffInfo
WHERE
    ModuleLeader.ModuleCode = ExamSetup.ModuleCode
    AND
    ModuleInfo.ModuleCode = ModuleLeader.ModuleCode
    AND
    StaffInfo.StaffNumber = ModuleLeader.StaffNumber
    AND
    (
        ModuleLeader.StaffNumber = {ParamStaffNumber} AND
        ModuleLeader.ModuleCode = {ParamModuleCode}
    )
```

Screen dumps

Preview of QRY_StaffModules					
1 record corresponds to your query.					
ModuleCode	DateExam	StaffNumber	Description	LecturerName	LecturerEmail
AST2652	10/11/2022	5017037	Accounting Sciences II	JH Nel	jh.nel@unisa.ac.za

Right-click to print, export records, etc.

Export to an HFSQI file...

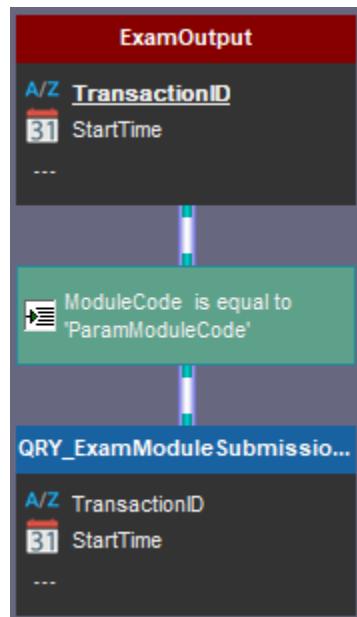
Close

## MODULE SUBMISSIONS MODULE CODE QUERY

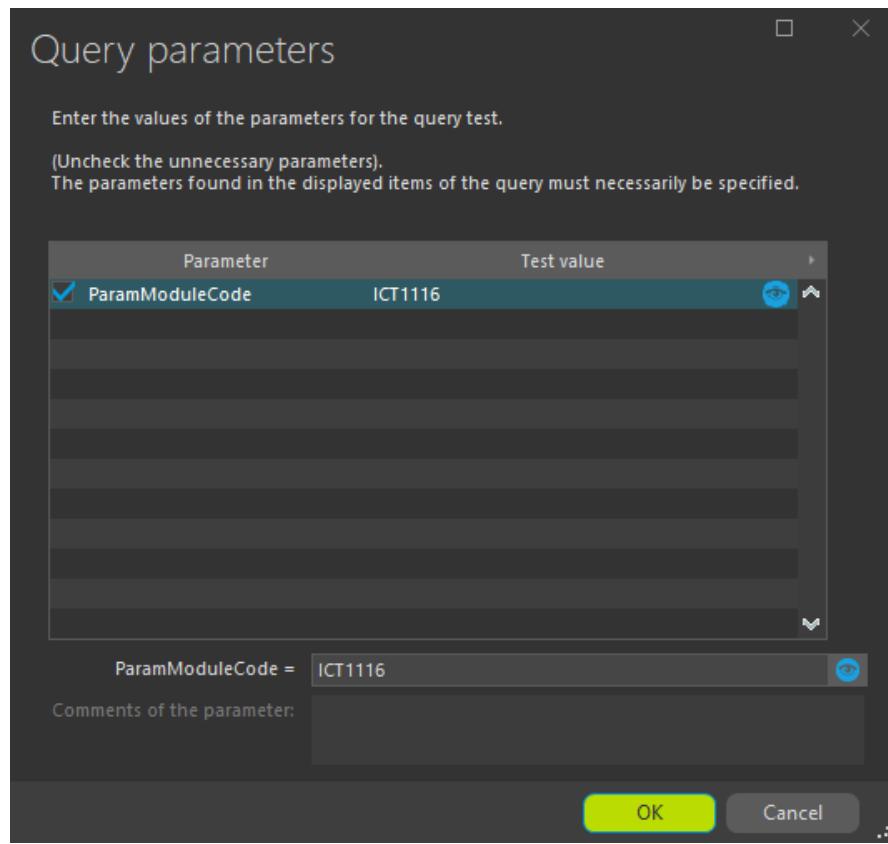
### Reason

After a module code is passed as parameter to this query, the query returns the records of all the submissions for that day. This data will be used to fill a chart.

### Query



```
SQL code of QRY_ExamModuleSubmissions
SELECT
    ExamOutput.TransactionID AS TransactionID,
    ExamOutput.StartTime AS StartTime,
    ExamOutput.UploadTime AS UploadTime,
    ExamOutput.AnswerPaperPDF AS AnswerPaperPDF,
    ExamOutput.StudentNumber AS StudentNumber,
    ExamOutput.ModuleCode AS ModuleCode
FROM
    ExamOutput
WHERE
    ExamOutput.ModuleCode = {ParamModuleCode}
```

Screen dumps

Preview of QRY_ExamModuleSubmissions						
316 records correspond to your query.						
TransactionID	StartTime	UploadTime	AnswerPaperPDF	StudentNumber		
ThnH1570-8411	15/11/2022 10:57	15/11/2022 11:42	15704367_ICT1116_EXAM_20221115-11:42:18.pdf	15704367		
wgaZ5427-7070	15/11/2022 11:23	15/11/2022 11:41	54271574_ICT1116_EXAM_20221115-11:41:04.pdf	54271574		
khXY1370-7976	15/11/2022 11:04	15/11/2022 11:22	13702516_ICT1116_EXAM_20221115-11:22:56.pdf	13702516		
ZkgI2208-8116	15/11/2022 10:23	15/11/2022 11:04	22087524_ICT1116_EXAM_20221115-11:04:39.pdf	22087524		
yucx6917-8455	15/11/2022 10:29	15/11/2022 10:48	69171951_ICT1116_EXAM_20221115-10:48:04.pdf	69171951		
Rahy5206-3942	15/11/2022 10:45	15/11/2022 11:33	52063360_ICT1116_EXAM_20221115-11:33:19.pdf	52063360		
sdCg3489-7394	15/11/2022 11:43	15/11/2022 11:55	34897858_ICT1116_EXAM_20221115-11:55:42.pdf	34897858		
qXHL1818-5423	15/11/2022 10:51	15/11/2022 11:41	18189010_ICT1116_EXAM_20221115-11:41:15.pdf	18189010		
qRO1120-2111	15/11/2022 10:26	15/11/2022 11:15	11207520_ICT1116_EXAM_20221115-11:15:02.pdf	11207520		
sVLr8770-7350	15/11/2022 10:38	15/11/2022 11:11	87701679_ICT1116_EXAM_20221115-11:11:30.pdf	87701679		
iQXu9591-8131	15/11/2022 10:44	15/11/2022 11:55	95915062_ICT1116_EXAM_20221115-11:55:35.pdf	95915062		
yCTG7122-3640	15/11/2022 10:50	15/11/2022 11:12	71225123_ICT1116_EXAM_20221115-11:12:30.pdf	71225123		
befQ5434-2170	15/11/2022 10:53	15/11/2022 11:22	54349440_ICT1116_EXAM_20221115-11:22:59.pdf	54349440		
ijDA3725-4653	15/11/2022 11:30	15/11/2022 11:51	37258908_ICT1116_EXAM_20221115-11:51:48.pdf	37258908		

Right-click to print, export records, etc.

**Export to an HFSQ file...**

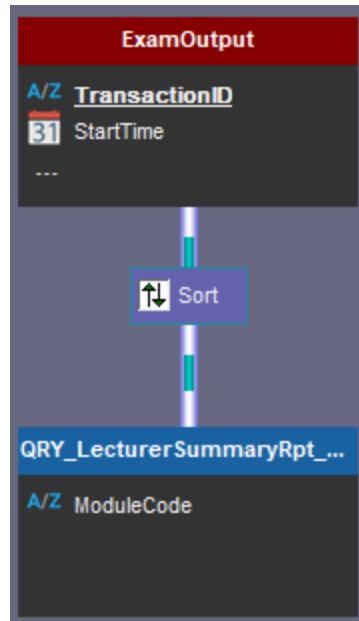
**Close**

## LECTURER SUMMARY CHART MODULES QUERY

### Reason

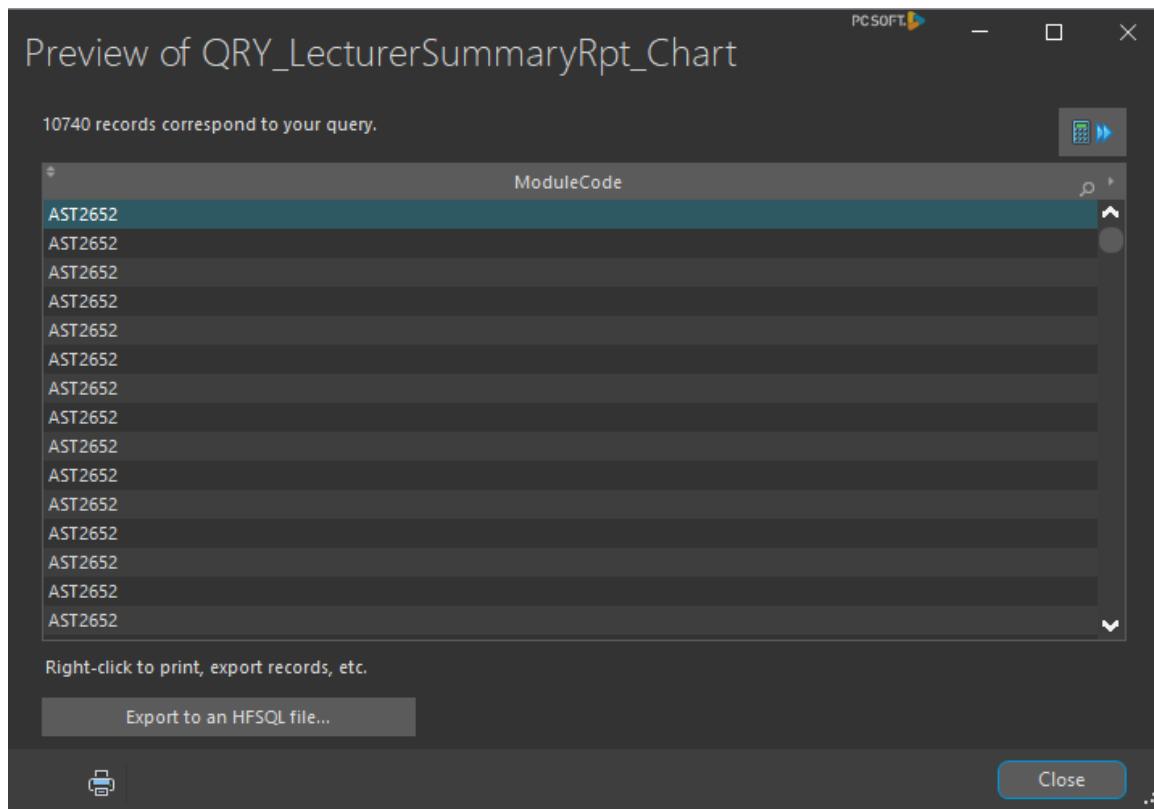
This query accepts no parameters. It simply gets all the records uploaded and returns the module codes. A special function will be used to group the data to fill the chart.

### Query



```
SQL code of QRY_LecturerSummaryRpt_Chart
SELECT
    ExamOutput.ModuleCode AS ModuleCode
FROM
    ExamOutput
ORDER BY
    ModuleCode ASC
```

## Screen dumps



Create one other MIS report (5)

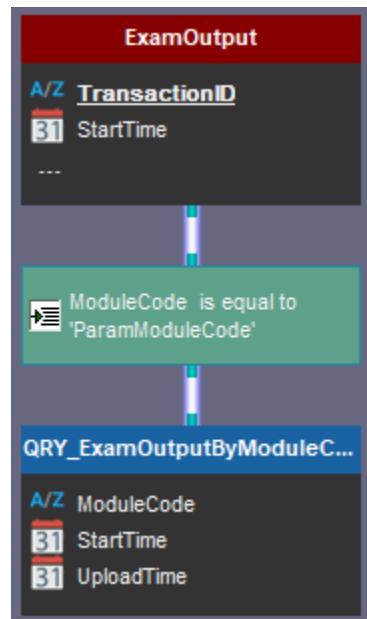
### MODULE STATISTICS QUERY

#### Reason

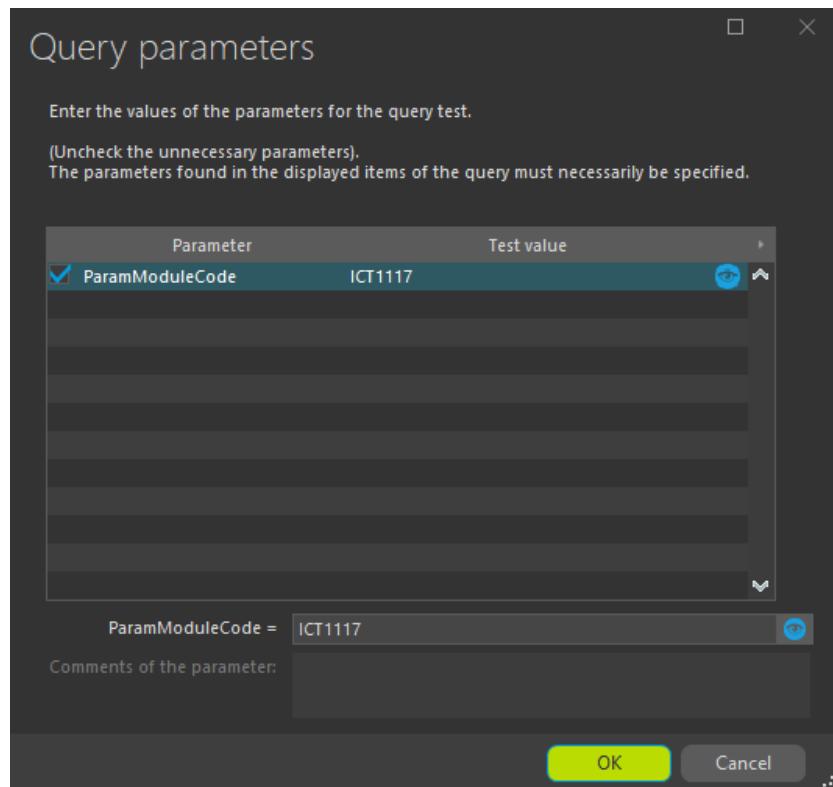
This query accepts a module code as parameter and returns all the records for that module submitted.

A function will be used to group the records by module code. Then more functions will be created to list each modules average, maximum and minimum upload times.

#### Query



```
SQL code of QRY_ExamOutputByModuleCode
SELECT
    ExamOutput.ModuleCode AS ModuleCode,
    ExamOutput.StartTime AS StartTime,
    ExamOutput.UploadTime AS UploadTime
FROM
    ExamOutput
WHERE
    ExamOutput.ModuleCode = {ParamModuleCode}
```

Screen dumps

Preview of QRY_ExamOutputByModuleCode			
316 records correspond to your query.			
ModuleCode	StartTime	UploadTime	
ICT1117	16/11/2022 11:17	16/11/2022 11:20	
ICT1117	16/11/2022 10:07	16/11/2022 10:23	
ICT1117	16/11/2022 10:36	16/11/2022 11:42	
ICT1117	16/11/2022 11:22	16/11/2022 11:41	
ICT1117	16/11/2022 10:25	16/11/2022 11:06	
ICT1117	16/11/2022 11:09	16/11/2022 11:58	
ICT1117	16/11/2022 10:48	16/11/2022 11:09	
ICT1117	16/11/2022 10:53	16/11/2022 10:56	
ICT1117	16/11/2022 10:27	16/11/2022 11:38	
ICT1117	16/11/2022 10:20	16/11/2022 10:55	
ICT1117	16/11/2022 11:39	16/11/2022 11:49	
ICT1117	16/11/2022 10:28	16/11/2022 11:07	
ICT1117	16/11/2022 10:57	16/11/2022 11:59	
ICT1117	16/11/2022 10:52	16/11/2022 11:45	
ICT1117	16/11/2022 11:17	16/11/2022 11:26	

Right-click to print, export records, etc.

Export to an HFSQ file...

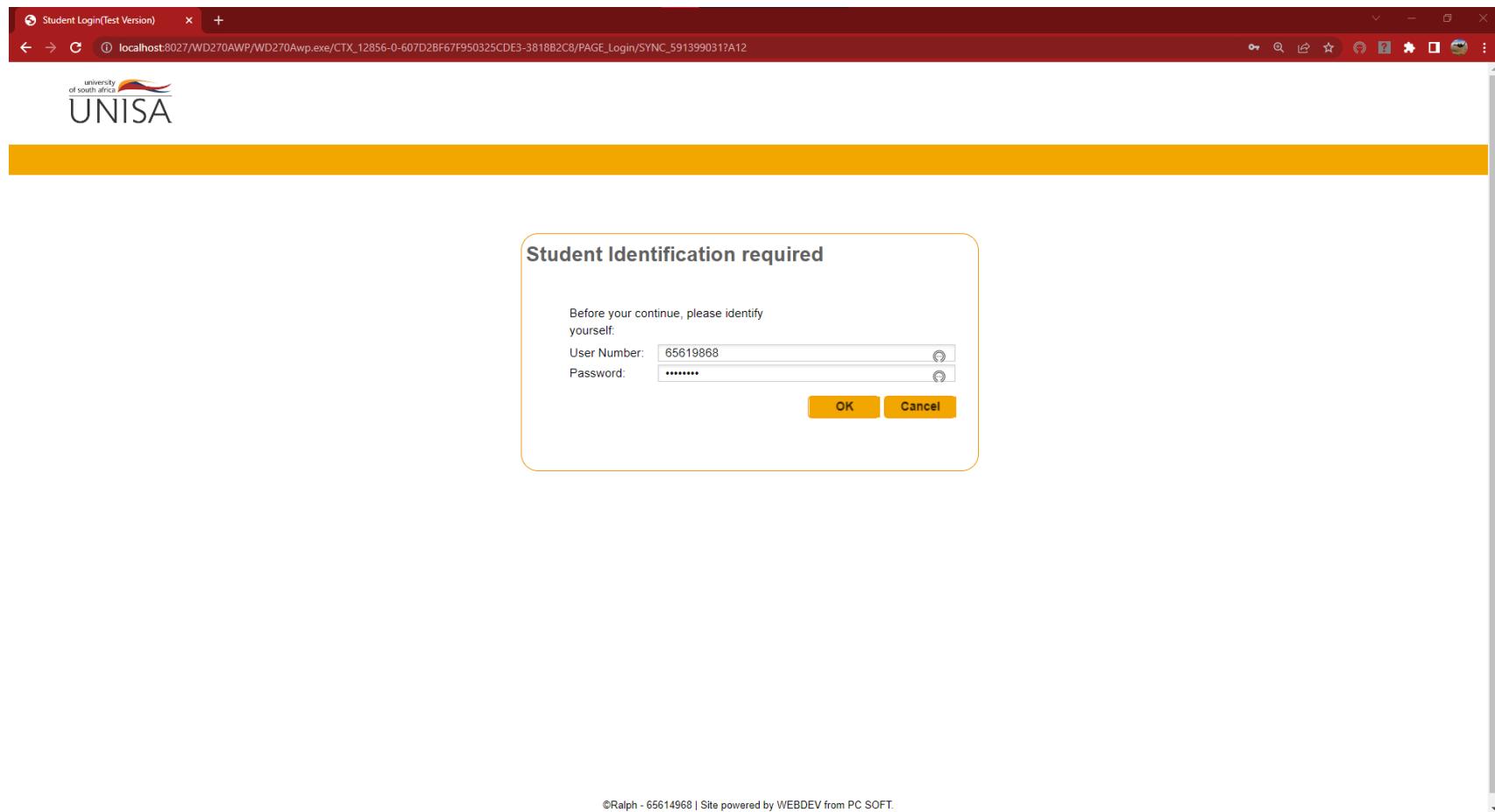
Close

## Section B = Graphical User Interfaces [20]

### Student login GUI (7)

#### Creating the Graphical User Interfaces for the Student Login

##### Screen dumps



Code**OK Button Click Code**

```

    [WL] Click on BTN_OK (onclick browser event) ( CELL_NoName1 )
1  IF EDT_USERNUMBER = "" THEN
2      ReturnToCapture(EDT_USERNUMBER)
3  END
4  IF EDT_PASSWORD = "" THEN
5      ReturnToCapture(EDT_PASSWORD)
6  END

    Click on BTN_OK ( CELL_NoName1 ) (server)      AJAX
1  bUserLogin is boolean
2  bUserLogin = Connection()

4  IF bUserLogin = False THEN
5      EDT_PASSWORD          = ""
6      STC_Error             = "Invalid user or password"
7      STC_Error..Visible     = True
8  ELSE
9      PageDisplay(PAGE_Student_Dashboard)
10 END

```

**Student Connection()**

PAGE\_Student\_Login

```

    S Local procedure Connection (server)      AJAX If Error:
1  PROCEDURE Connection()
2  bOnUser is boolean = False
3
4  // Finds the user
5  HReadSeekFirst(StudentInfo,StudentNumber,EDT_USERNUMBER,hIdentical)
6  IF HFound(StudentInfo) THEN
7
8      // Checks the password
9      IF StudentInfo.StudentPassword = EDT_PASSWORD THEN
10
11         // Generates a session
12         gsUserType      = "Student"
13         gnUserID        = StudentInfo.StudentNumber
14         gsUser          = StudentInfo.StudentName
15
16         GenerateSession()
17         bOnUser = True
18
19     END
20
21 END
22
23 RETURN bOnUser

```

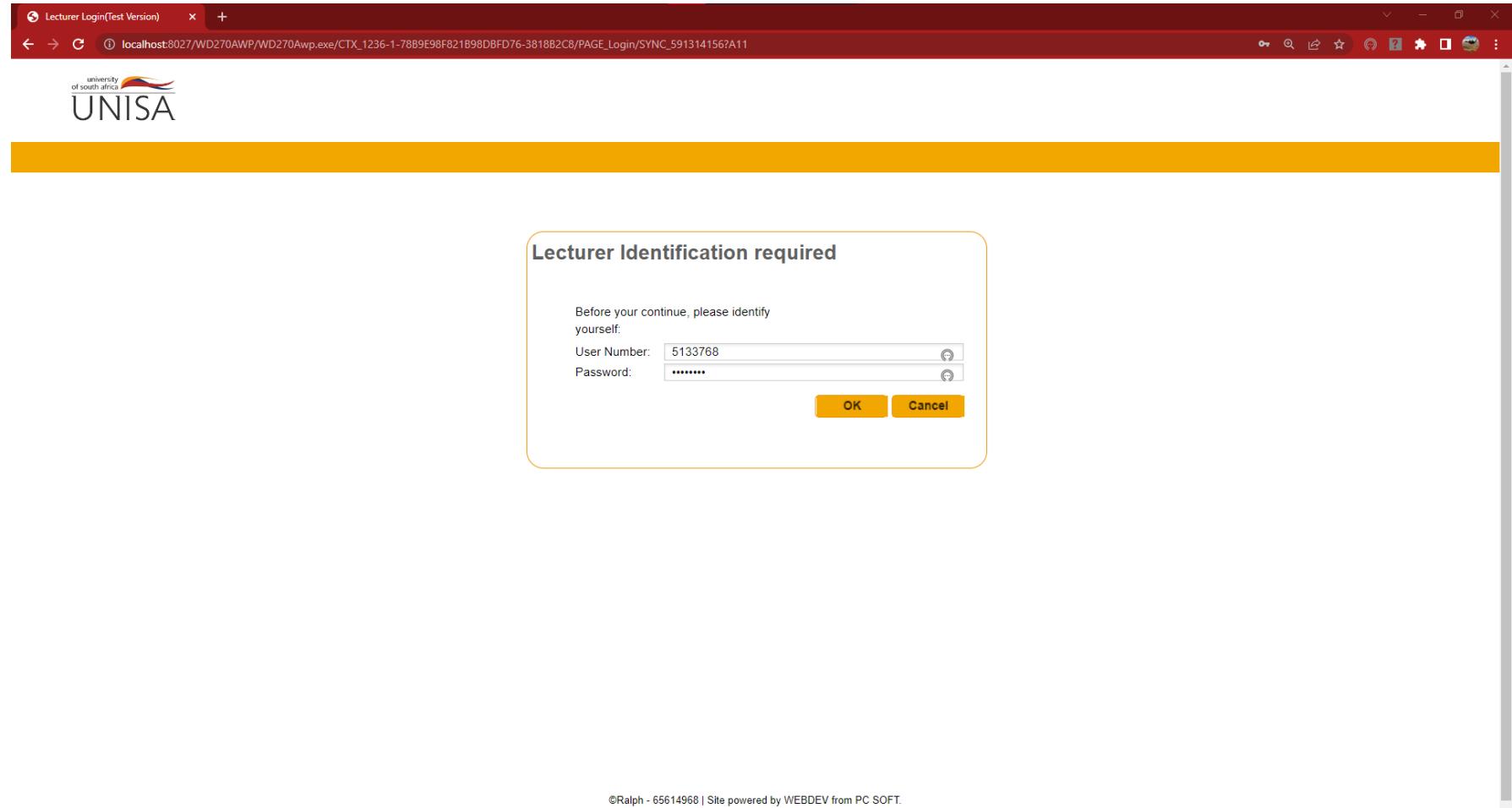
### Student GenerateSession()

```
P PAGE_Student_Login S Local procedure GenerateSession (server) AJAX If Error
1 PROCEDURE GenerateSession()
2 // Initialize new connection
3 New_connection is Connection
4 // Connection parameters
5 New_connection..Provider      = hAccessHFClientServer
6 New_connection..User          = gsUser
7 New_connection..Password      = EDT_PASSWORD
8 New_connection..Server        = "localhost"
9 New_connection..Database      = "ICT3715_DB"
10 New_connection..CryptMethod   = hEncryptionNO
11
12 // Generates a session identifier
13 dtdhDateTimeCurrent is DateTime = Today() + TimeSys()
14 gsSession = dtdhDateTimeCurrent + TAB + gnUserID + "-" + gsUser
15
16 HChangeConnection("ExamOutput,ExamSetup,StudentModule",New_connection)
17
18 gConnection = New_connection
19
20 // Backup
21 CookieWrite("session", gsSession, 1)
22
```

## Lecturer login GUI (7)

### Creating the Graphical User Interfaces for the Lecturer Login

#### Screen dumps



©Ralph - 65614968 | Site powered by WEBDEV from PC SOFT.

Code**OK Button Click Code**

```

    [WL] Click on BTN_OK (onclick browser event) ( CELL_NoName1 ) \
1 IF EDT_USERNUMBER = "" THEN
2   ReturnToCapture(EDT_USERNUMBER)
3 END
4 IF EDT_PASSWORD = "" THEN
5   ReturnToCapture(EDT_PASSWORD)
6 END

    Click on BTN_OK ( CELL_NoName1 ) (server)      AJAX
1 bUserLogin is boolean
2 bUserLogin = Connection()
3
4 IF bUserLogin = False THEN
5   EDT_PASSWORD           = ""
6   STC_Error              = "Invalid user or password"
7   STC_Error..Visible     = True
8 ELSE
9   PageDisplay(PAGE_Lecturer_Dashboard)
10 END

```

**Lecturer Connection()**

P PAGE\_Lecturer\_Login

```

[S] Local procedure Connection (server)      AJAX If Err
1 PROCEDURE Connection()
2   bOnUser is boolean = False
3
4   // Finds the user
5   HReadSeekFirst(StaffInfo,StaffNumber,EDT_USERNUMBER,hIdentical)
6 IF HFound(StaffInfo) THEN
7
8   // Checks the password
9   IF StaffInfo.Password = EDT_PASSWORD THEN
10
11     // Generates a session
12     gsUserType      = "Lecturer"
13     gnUserID        = StaffInfo.StaffNumber
14     gsUser          = StaffInfo.Name
15
16     GenerateSession()
17     bOnUser = True
18   END
19 END
20
21 RETURN bOnUser

```

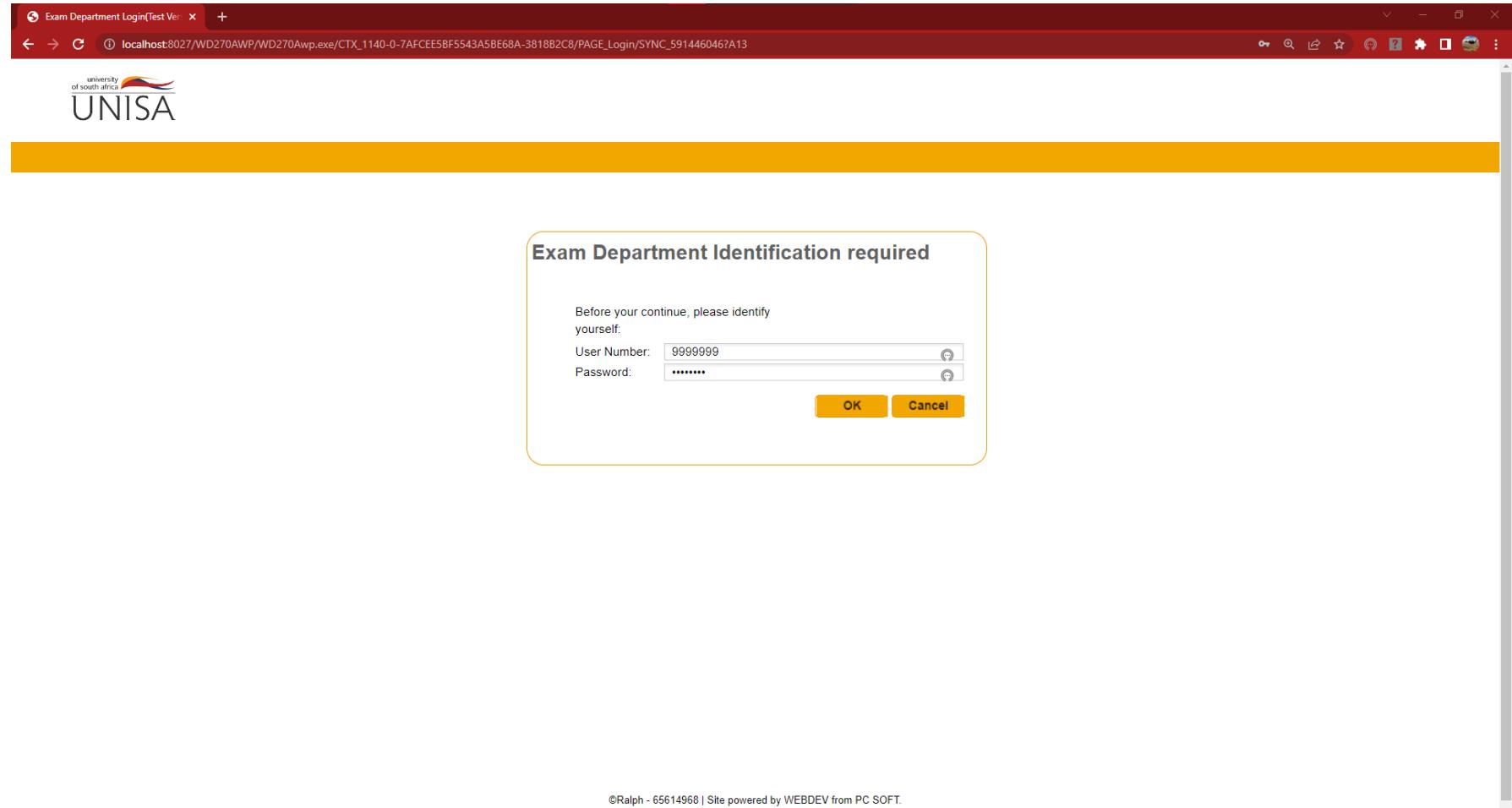
### Lecturer GenerateSession()

```
PAGE_Lecturer_Login | Local procedure GenerateSession (server) | AJAX | If Err
1 PROCEDURE GenerateSession()
2 // Initialize new connection
3 New_connection is Connection
4 // Connection parameters
5 New_connection..Provider      = hAccessHFClientServer
6 New_connection..User          = gsUser
7 New_connection..Password      = EDT_PASSWORD
8 New_connection..Server        = "localhost"
9 New_connection..Database      = "ICT3715_DB"
10 New_connection..CryptMethod   = hEncryptionNO
11
12 // Generates a session identifier
13 dtdhDateTimeCurrent is DateTime = Today() + TimeSys()
14 gsSession = dtdhDateTimeCurrent + TAB + gnUserID + "-" + gsUser
15
16 HChangeConnection("ExamSetup,ModuleLeader,StaffInfo",New_connection)
17
18 gConnection = New_connection
19
20 // Backup
21 CookieWrite("session", gsSession, 1)
```

## Exam Department (Staff member) login GUI (6)

### Creating the Graphical User Interfaces for the Exam Department (Staff member) Login

#### Screen dumps



Code**OK Button Click Code**

```

P PAGE_Exam_Department_Login | x
  Initializing of BTN_OK ( CELL_NoName1 ) (server)
1
  Click on BTN_OK (onclick browser event) ( CELL_NoName1 )
1 IF EDT_USERNUMBER = "" THEN
2   ReturnToCapture(EDT_USERNUMBER)
3 END
4 IF EDT_PASSWORD = "" THEN
5   ReturnToCapture(EDT_PASSWORD)
6 END

  Click on BTN_OK ( CELL_NoName1 ) (server)      AJAX
1 bUserLogin is boolean
2 bUserLogin = Connection()
3
4 IF bUserLogin = False THEN
5   EDT_PASSWORD           = ""
6   STC_Error              = "Invalid user or password"
7   STC_Error..Visible     = True
8 ELSE
9   PageDisplay(PAGE_Exam_Dashboard)
10 END

```

**Exam Department Connection()**

```

P PAGE_Exam_Department_Login | x
  Local procedure Connection (server)      AJAX If Error: by program
1 PROCEDURE Connection()
2   bOnUser is boolean = False
3
4   // Finds the user
5 IF EDT_USERNUMBER = ExamDeptNumber _AND_ EDT_PASSWORD = ExamDeptPassword THEN
6   // Generates a session
7   gsUserType      = ExamDeptUser
8   gnUserID        = ExamDeptNumber
9   gsUser          = ExamDeptUser
10
11   GenerateSession()
12   bOnUser = True
13 END
14
15 RETURN bOnUser

```

**Exam Department GenerateSession()**

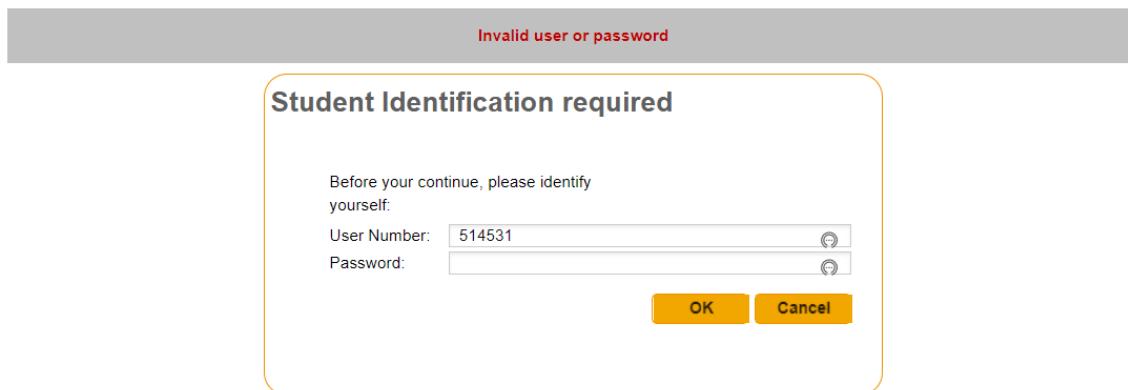
```

P PAGE_Exam_Department_Login S Local procedure GenerateSession (server) AJAX If Error: by program When Exe
1 PROCEDURE GenerateSession()
2 // Initialize new connection
3 New_connection is Connection
4 // Connection parameters
5 New_connection..Provider      = hAccessHFClientServer
6 New_connection..User          = gsUser
7 New_connection..Password      = EDT_PASSWORD
8 New_connection..Server        = "localhost"
9 New_connection..Database      = "ICT3715_DB"
10 New_connection..CryptMethod   = hEncryptionNO
11
12 // Generates a session identifier
13 dtdhDateTimeCurrent is DateTime = Today() + TimeSys()
14 gsSession = dtdhDateTimeCurrent + TAB + gnUserID + "-" + gsUser
15
16 HChangeConnection("ModuleLeader,ExamSetup,ModuleInfo,StaffInfo,StudentModule",New_connection)
17
18 gConnection = New_connection
19
20 // Backup
21 CookieWrite("session", gsSession, 1)

```

### User failed Login

If any of the login screens has a failed login due to wrong username or password or nonexistence, an error is given. Example:



## Assignment 2

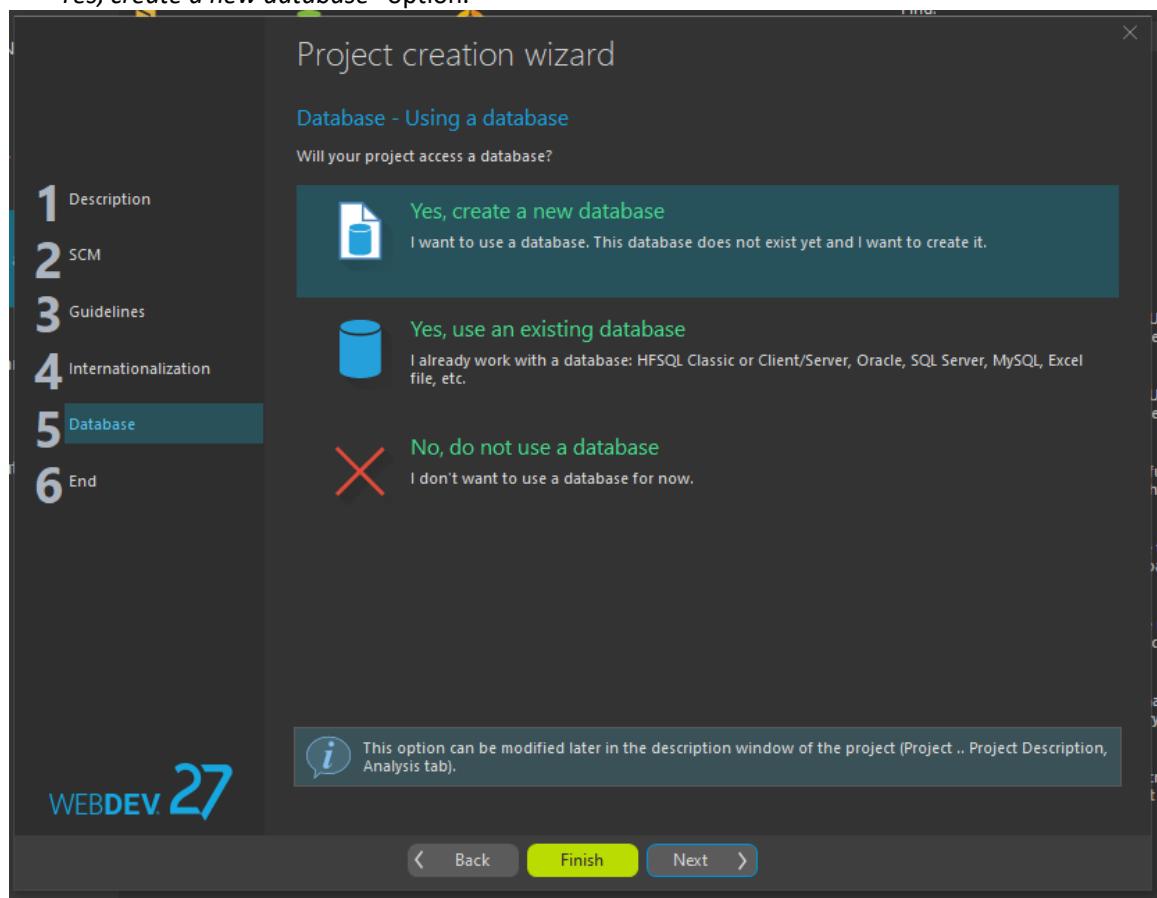
### Section A = Database [20]

Create the database for your system and importing the data from your .csv file (20)

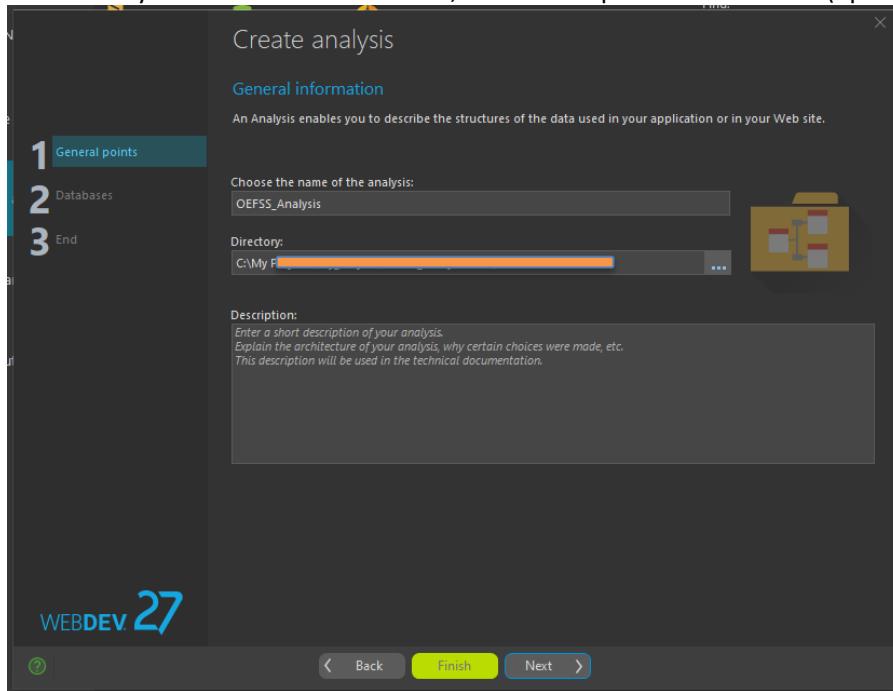
The following explains how I created everything for the database and how I populated it using WEBDEV and WLanguage from my cleaned data file Data.csv.

#### Creating the Database:

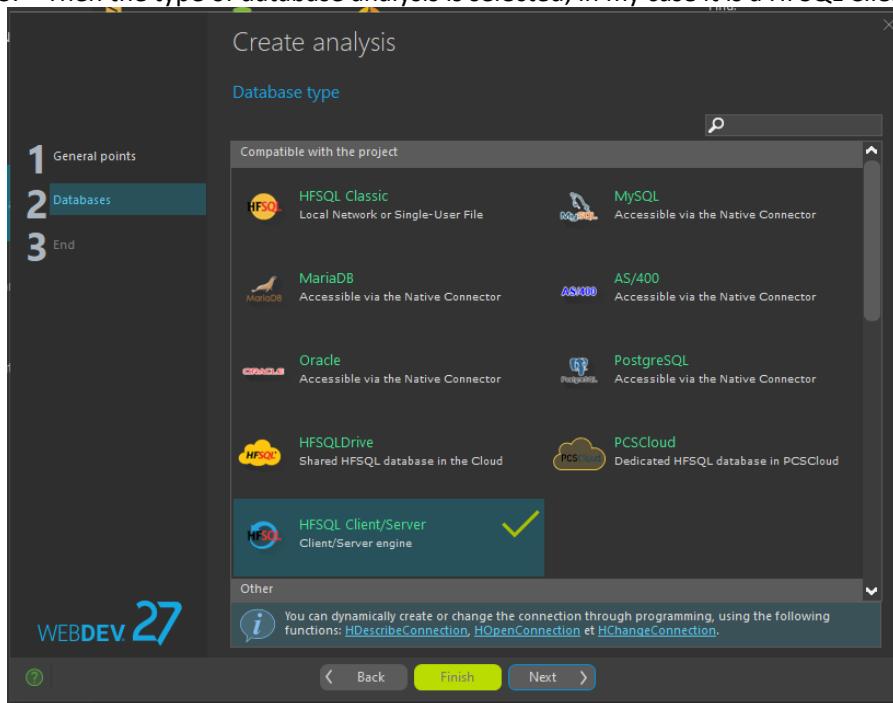
1. The creation of the database can be done in multiple ways, I created the database structure when I created the project. During the project creation process, the wizard gives you the option to create a new database, use an existing one or no database to use. I selected the "Yes, create a new database" option.



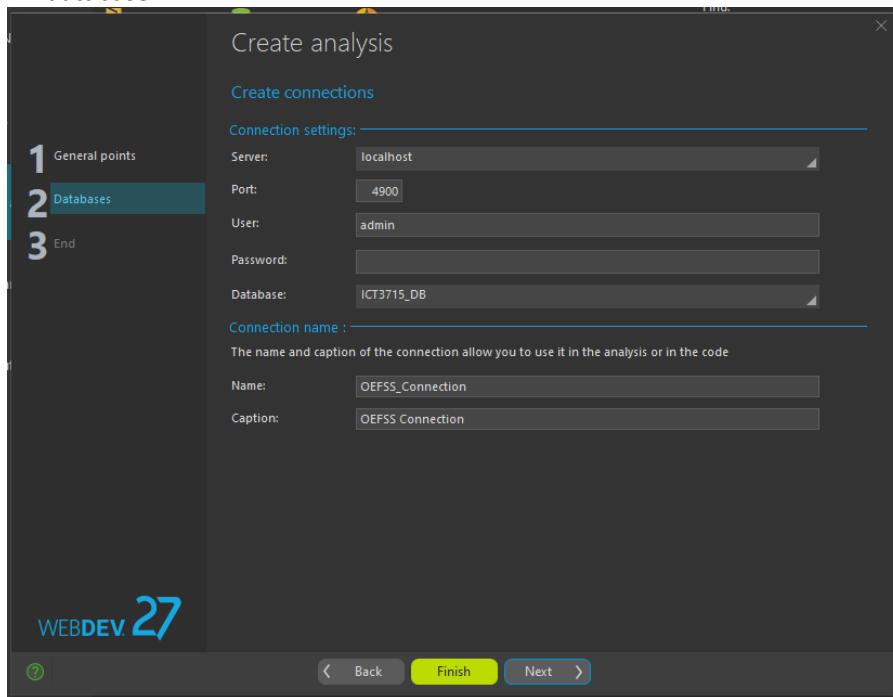
2. The following window is where you declare the name of the Analysis for the database, the directory to where it must be saved, and a description can be added (optional).



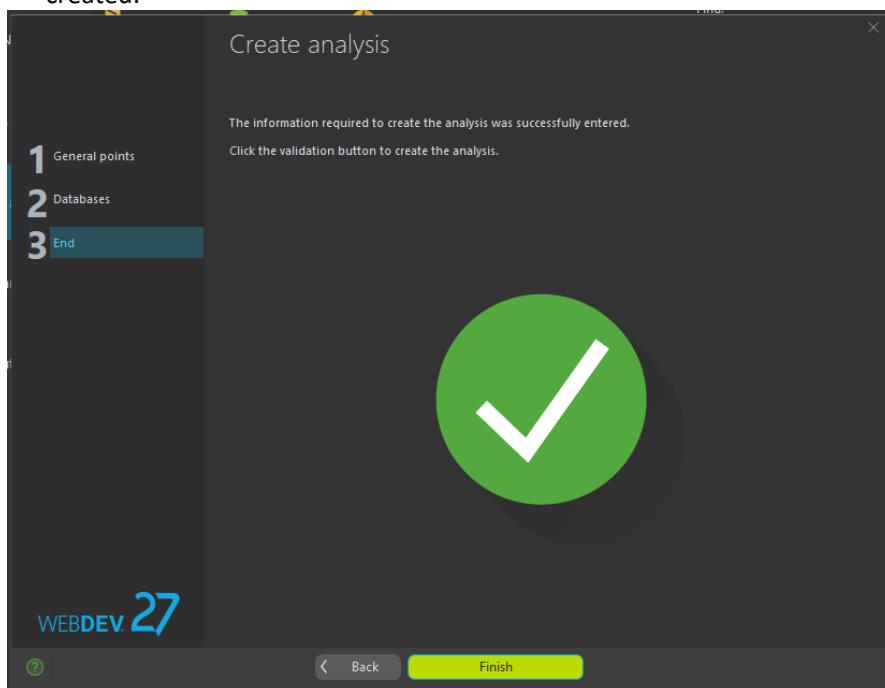
3. Then the type of database analysis is selected, in my case it is a HFSQL Client/Server.



4. The next window is where the connection to the server is created along with its settings. The connection will be used or a new connection can be created in code to access the data in the database.



5. The last window is used to finalize the creation of the database. On Finish select, it is created.

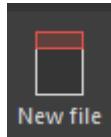


6. You are then directed to the Analysis page where the rest of the Database structure is designed.

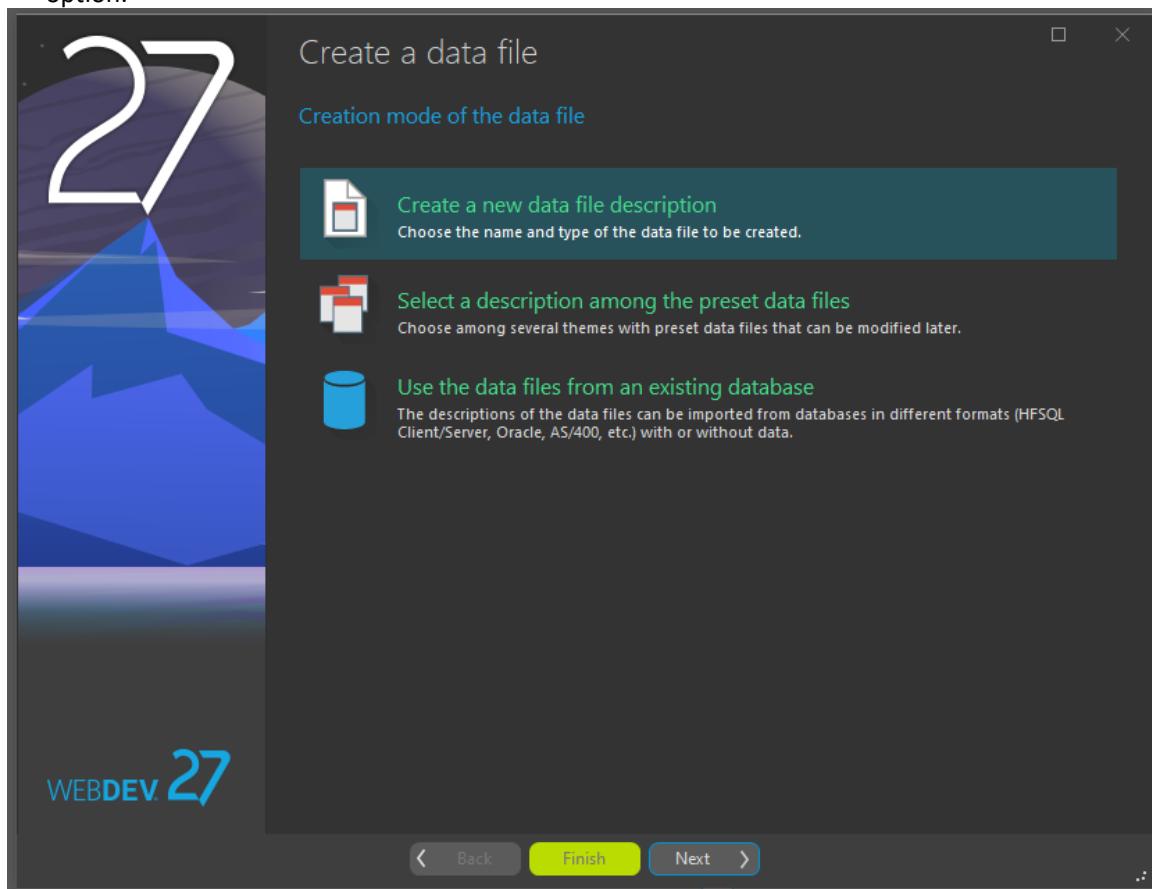
### Create a Data File (Table):

---

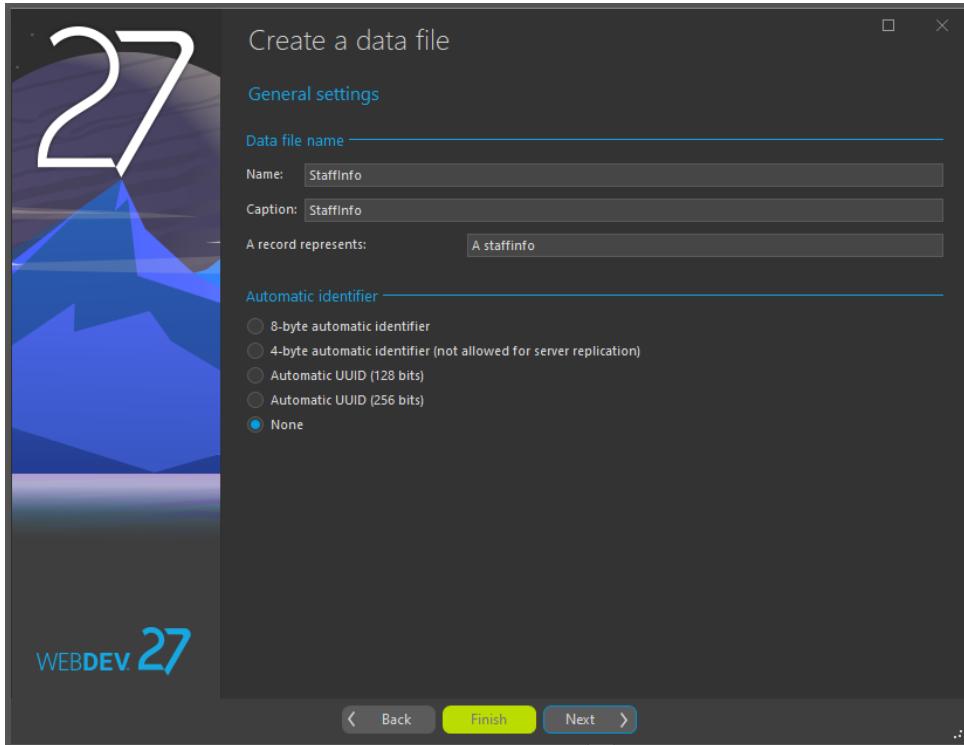
1. First select the New File icon in the Analysis tab to start the process.



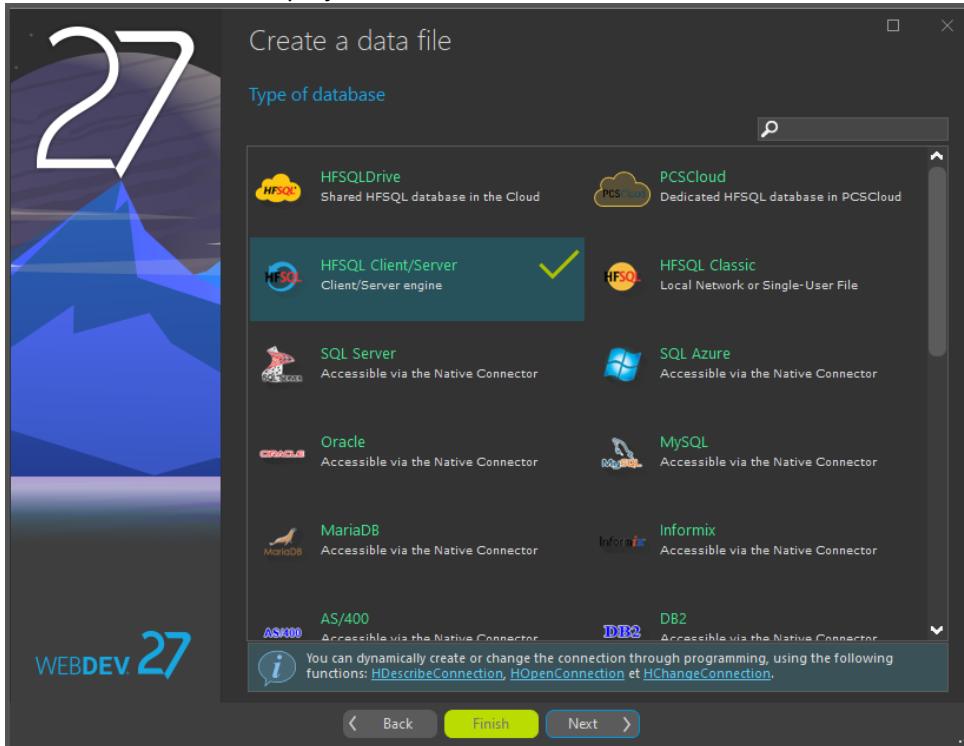
2. The creation process is started. There are a few options to choose from, like using the data file format of an existing database and selecting from preset data files given by WEBDEV. I wanted to create a new data file so I selected the "*Create a new data file description*" option.



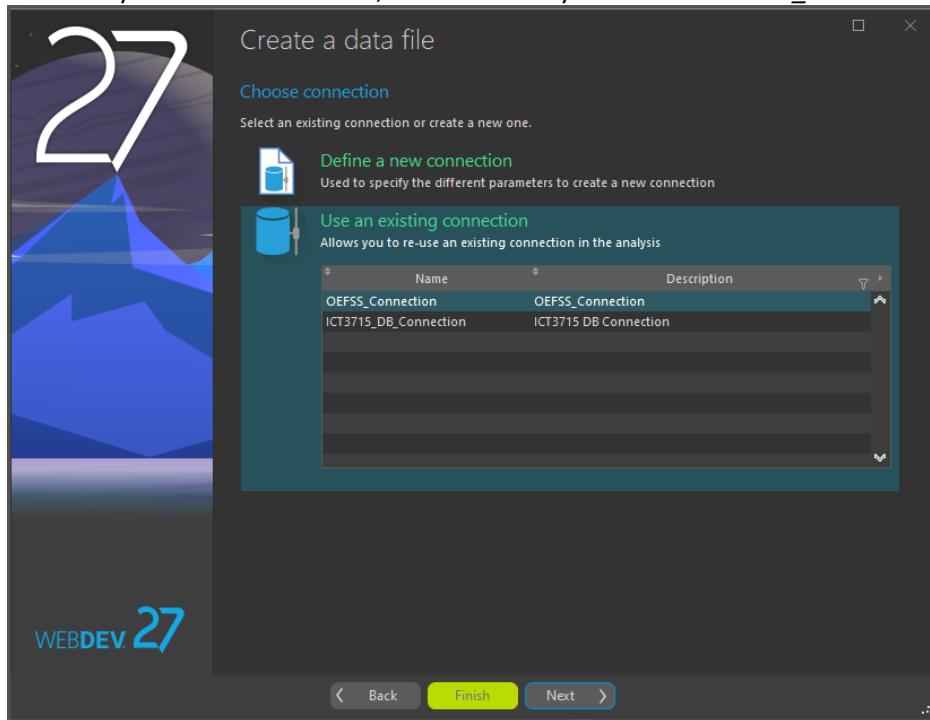
3. The next window allows me to select a name for my data file as well as how to handle the automatic identifier. As I will be creating my own Primary keys, no automatic identifiers are needed.



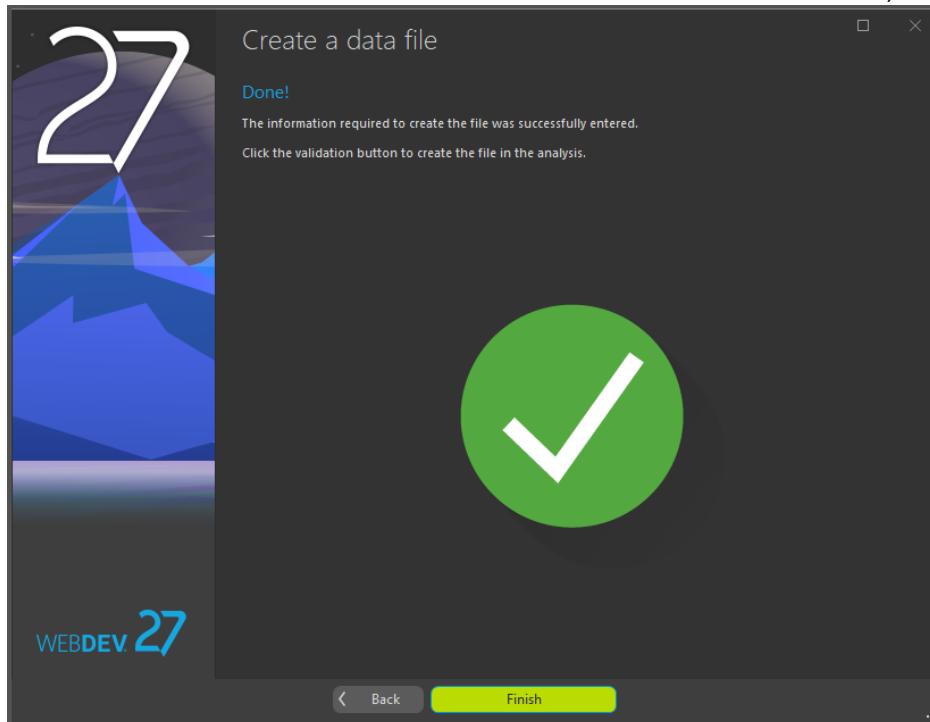
4. Here I get to select what type of data file I will be using. We will be using HFSQL Client/Server for this project.



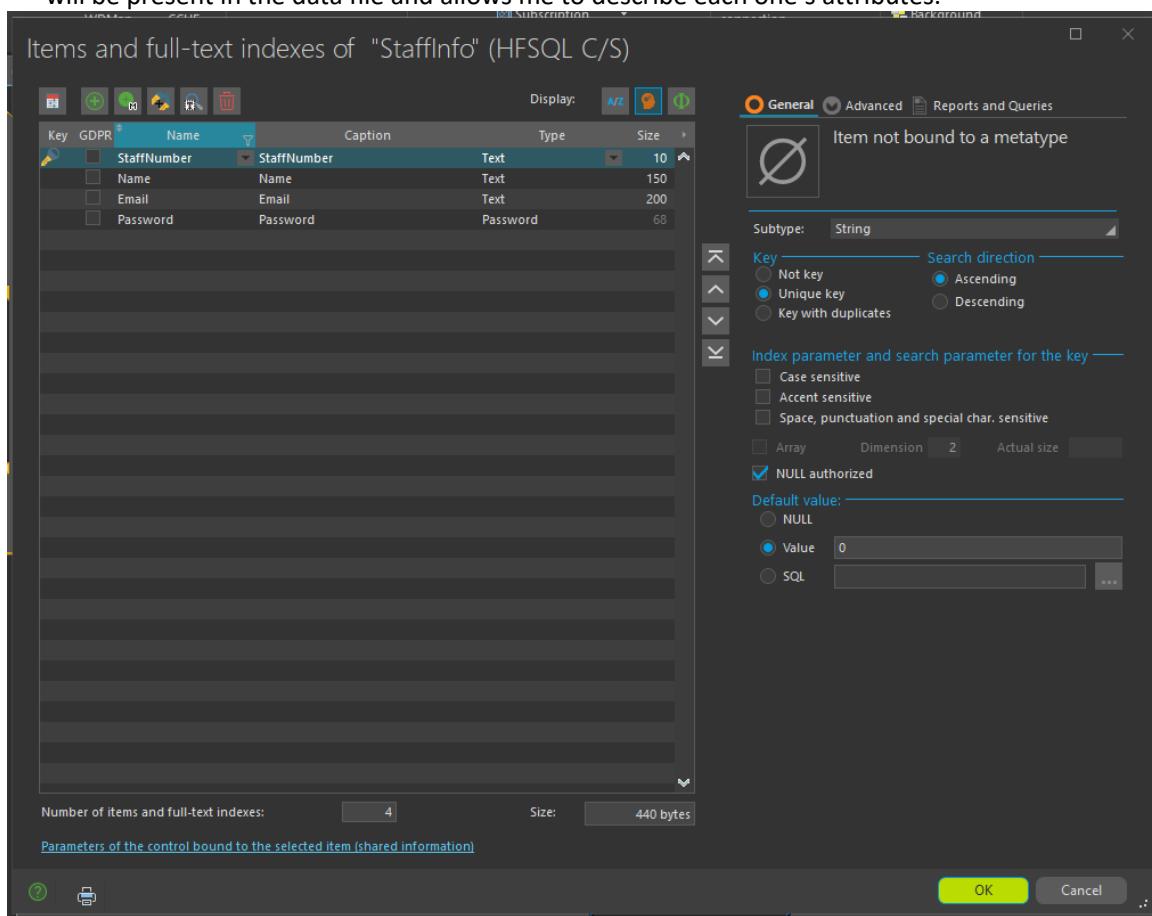
5. Because a Client/Server is used, a connection type is needed to allow access to the analysis. I already created a connection, so I selected my connection "OEFSS\_Connection".



6. The last window validates the creation of the data file and on Finish select, it is created.



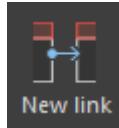
7. The item description window then opens and allows me to create each item (column) that will be present in the data file and allows me to describe each one's attributes.



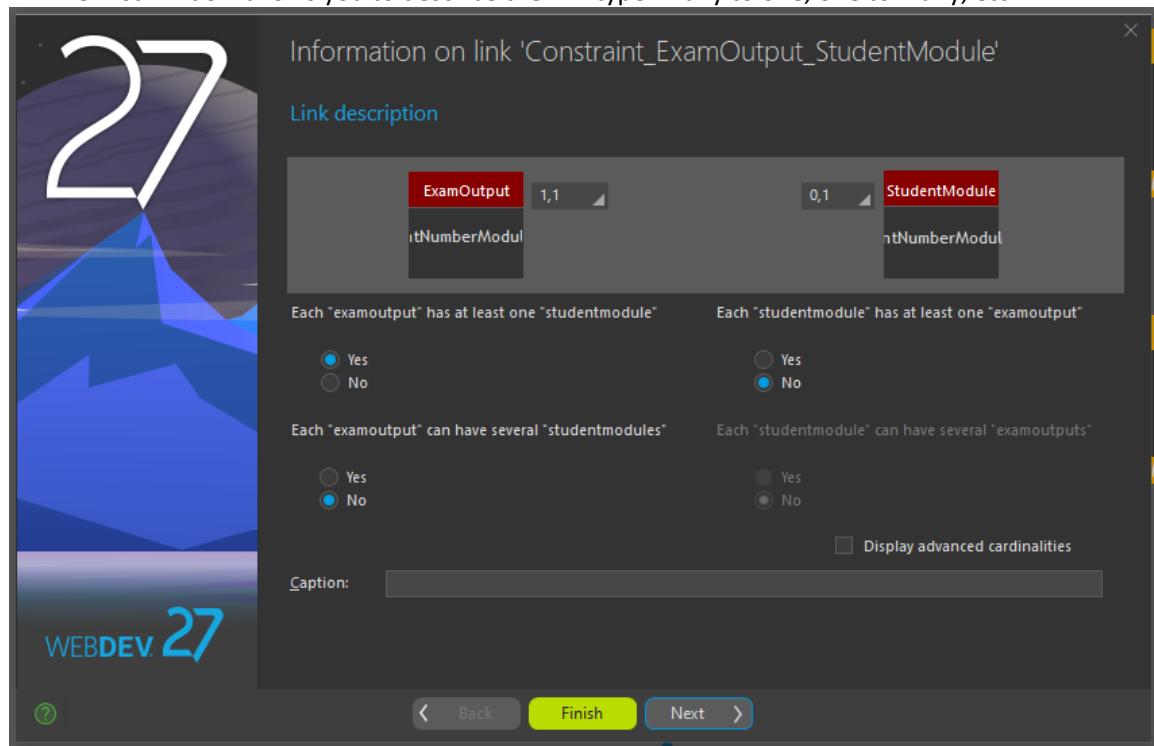
8. This process was followed for each data file created.

### Adding Links to Data Files

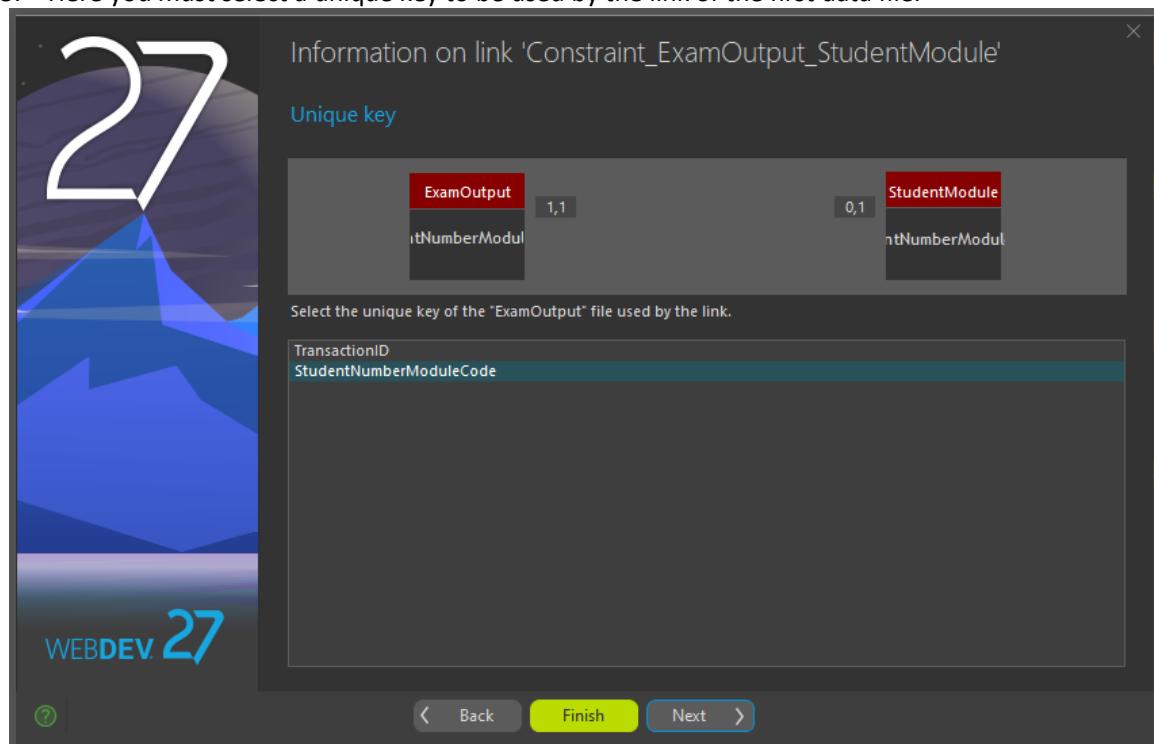
1. First select the New Link icon in the Analysis tab to start the process.



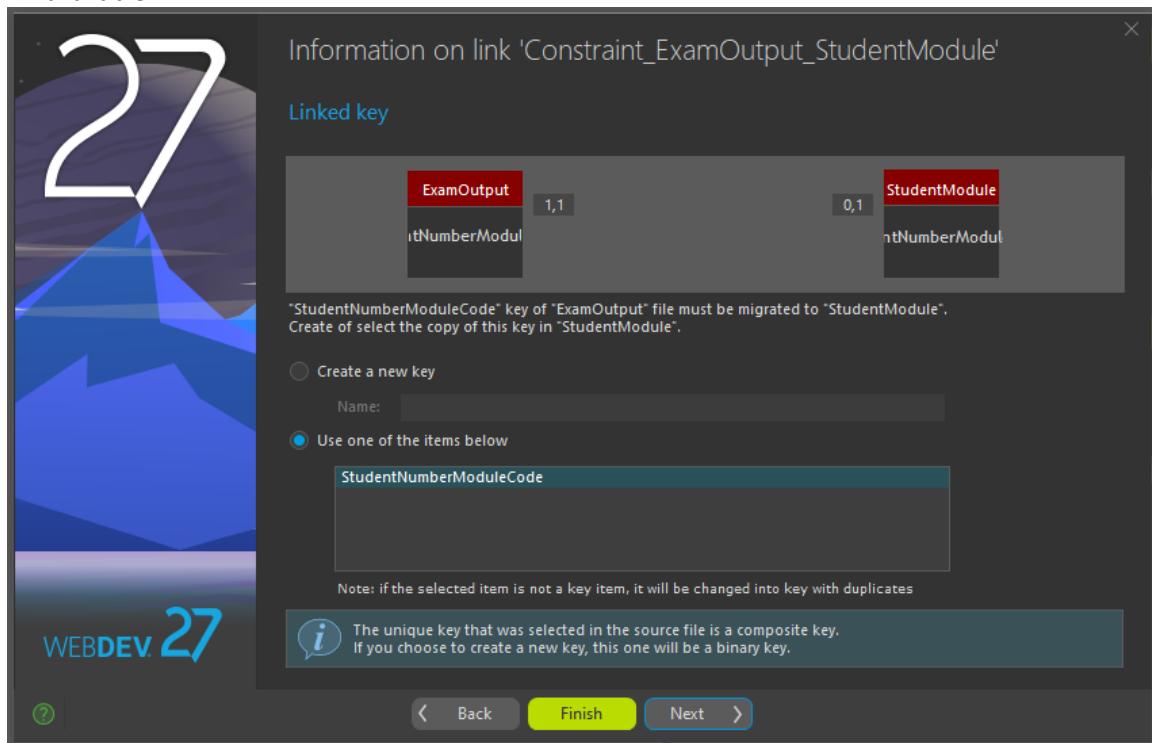
2. The first window allows you to describe the link type. Many to one, one to many, etc.



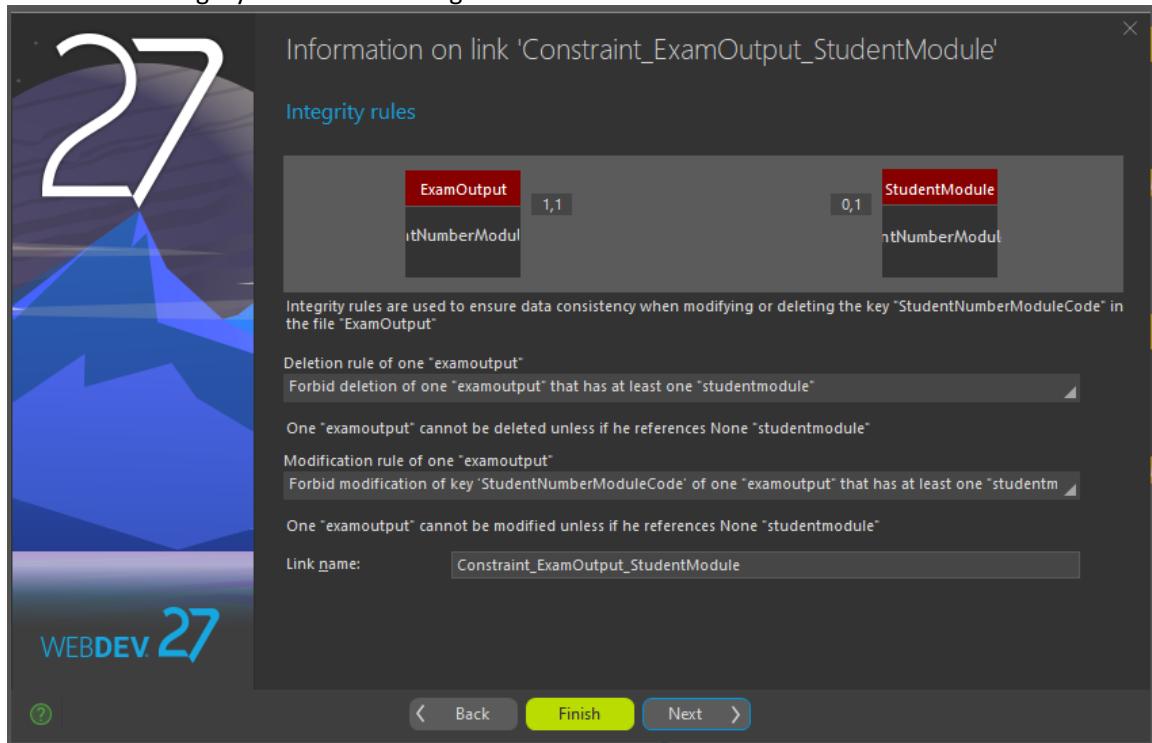
3. Here you must select a unique key to be used by the link of the first data file.



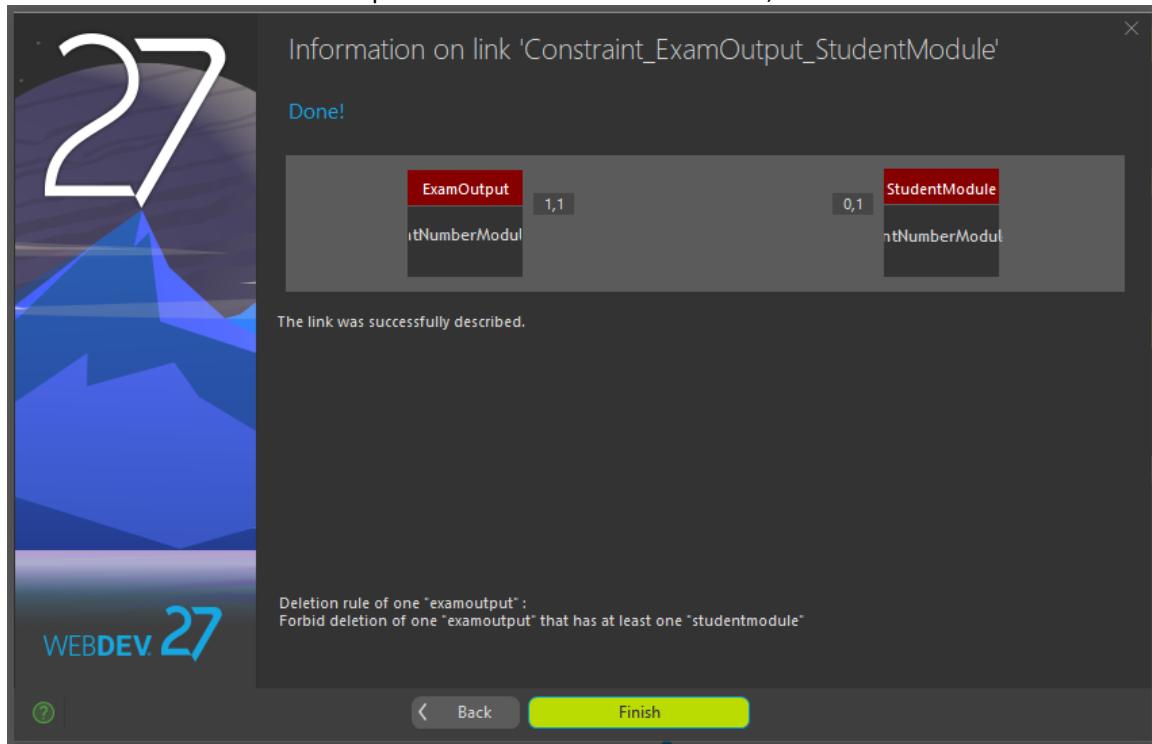
4. Then the link of the second data file must be selected or created in the data file if not available.



5. Here the integrity rules are set along with the link name.



6. A final view of the link description is shown and on Finish select, the link is created.

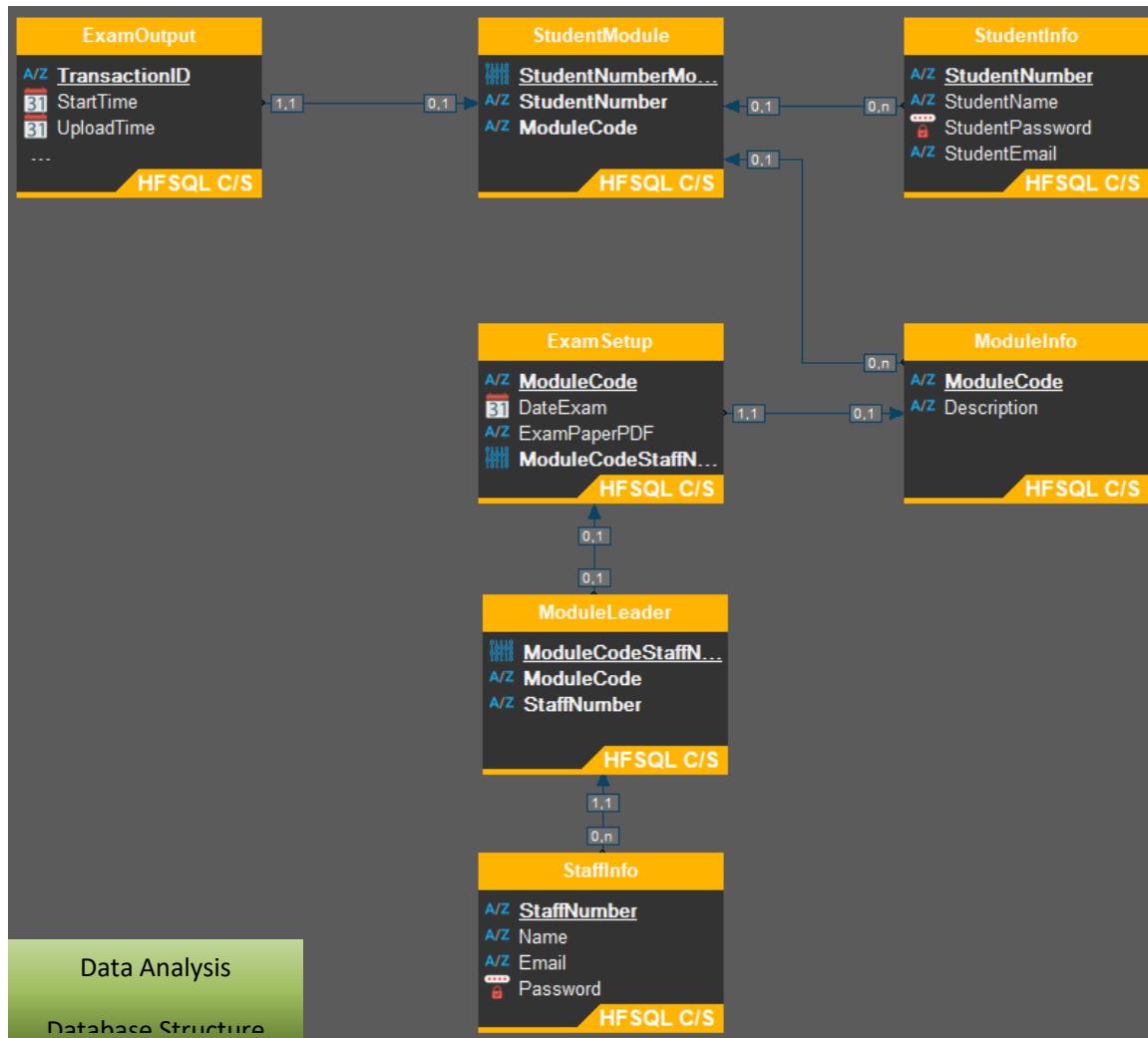


7. Here is the finished result.



8. This process was followed to create all the remaining links

In the Analysis, the data files (Tables) were created, and items (columns) were created and described. The complete finished result is as follows:



#### StaffInfo:

Key	GDPR	Name	Caption	Type	Size
		StaffNumber	StaffNumber	Text	10
		Name	Name	Text	150
		Email	Email	Text	200
		Password	Password	Password	68

#### ModuleLeader:

Key	GDPR	Name	Caption	Type	Size
		ModuleCodeStaffNu...	ModuleCode + StaffNumber	Composite key	17
		ModuleCode	Module Code	Text	7
		StaffNumber	StaffNumber	Text	10

**ModuleInfo:**

Key	GDPR	Name	Caption	Type	Size
	<input type="checkbox"/>	ModuleCode	Module Code	Text	7
	<input type="checkbox"/>	Description	Description	Text	100

**ExamSetup:**

Key	GDPR	Name	Caption	Type	Size
	<input type="checkbox"/>	ModuleCode	Module Code	Text	7
	<input type="checkbox"/>	DateExam	Date Exam	Date	8
	<input type="checkbox"/>	ExamPaperPDF	Exam Paper PDF	Text	150
	<input type="checkbox"/>	ModuleCodeStaffNumI	ModuleCodeStaffNumber	Sound, image, binary	17

**StudentInfo:**

Key	GDPR	Name	Caption	Type	Size
	<input type="checkbox"/>	StudentNumber	Student Number	Text	8
	<input type="checkbox"/>	StudentName	Student Name	Text	150
	<input type="checkbox"/>	StudentPassword	Student Password	Password	68
	<input type="checkbox"/>	StudentEmail	Student Email	Text	200

**StudentModule:**

Key	GDPR	Name	Caption	Type	Size
	<input type="checkbox"/>	StudentNumberMod	StudentNumber + ModuleCode	Composite key	15
	<input type="checkbox"/>	StudentNumber	Student Number	Text	8
	<input type="checkbox"/>	ModuleCode	Module Code	Text	7

**ExamOutput:**

Key	GDPR	Name	Caption	Type	Size
	<input type="checkbox"/>	TransactionID	Transaction ID	Text	13
	<input type="checkbox"/>	StartTime	Start Time	Date	8
	<input type="checkbox"/>	UploadTime	Upload Time	Date	8
	<input type="checkbox"/>	AnswerPaperPDF	Answer Paper PDF	Text	150
	<input type="checkbox"/>	StudentNumber	Student Number	Text	8
	<input type="checkbox"/>	ModuleCode	Module Code	Text	7
	<input type="checkbox"/>	StudentNumberModul	StudentNumber + ModuleCode	Composite key	15

### **Uploading Data:**

---

I created individual procedures for each Data file (Table) to run the uploading of the relevant data to the Database table.

In each procedure, the steps followed are relatively the same with minor differences as can be viewed in the code:

- Create the local variables and arrays that will be needed during the procedure execution.
- Open the file with the file path passed to the procedure.
- If the file has any lines, then continue.
  - Skip the first line in the file as it contains the column names.
  - Make sure it is not the end of the file then assign the first line to a string.
  - While its not the end of the file:
    - Split the string line read into an array of strings
    - Assign the Primary key to its variable and check if it already exists in the table, if it does, go to next line, else continue.
    - Assign the rest of the string values to their variables and do any formatting needed. For example, change a string date value to a date form.
    - Assign the final values to a new data file by adding each field, Datafile.item = value
    - Then add the item to the table.
    - Read the next line.
  - If it is the end of the data file, then close the file read.

### **Procedure UploadStaffInfo:**

```
PROCEDURE UploadStaffInfo(sPath)
// Import StaffInfo Data
// Variable declaration
nFileNum           is int
sLineRead          is string
arrLineArray       is array of strings

sStaffNumber       is string
sName              is string
sEmail             is string
sPassword          is string

nFileNum = fOpen (sPath, foRead)
IF nFileNum <> -1 THEN
  // Read the 1st line
  sLineRead = fReadLine(nFileNum)
```

```
// Check the end of file
IF sLineRead <> EOT THEN
    // Skip first line
    sLineRead = fReadLine(nFileNum)
END
WHILE sLineRead <> EOT
    // Process the line
    // Put line values in a string array
    arrLineArray      = StringSplit(sLineRead,";")

    // Retrieving the values of the array
    sStaffNumber          = arrLineArray[9]

    // Check if Record exists
    HReadSeekFirst(StaffInfo,StaffNumber,sStaffNumber)
    IF HFound() = True THEN      // Exists continue to new record
        sLineRead = fReadLine(nFileNum)
        CONTINUE
    END

    // Else get rest of data
    sName                = arrLineArray[10]
    sEmail               = arrLineArray[11]
    sPassword             = arrLineArray[12]

    // Add to StaffInfo
    StaffInfo.StaffNumber      = sStaffNumber
    StaffInfo.Name              = sName
    StaffInfo.Email             = sEmail
    StaffInfo.Password          = sPassword
    HAdd(StaffInfo)

    // Read the next line
    sLineRead = fReadLine(nFileNum)
END
fClose(nFileNum)
END
```

**Procedure UploadModuleLeader:**

```

PROCEDURE UploadModuleLeader(sPath)
// Import StaffInfo Data
// Variable declaration
nFileNum           is int
sLineRead          is string
arrLineArray       is array of strings

sStaffNumber       is string
sModuleCode        is string

nFileNum = fOpen (sPath, foRead)
IF nFileNum <> -1 THEN
    // Read the 1st line
    sLineRead = fReadLine(nFileNum)

    // Check the end of file
    IF sLineRead <> EOT THEN
        // Skip first line
        sLineRead = fReadLine(nFileNum)
    END
    WHILE sLineRead <> EOT
        // Process the line
        // Put line values in a string array
        arrLineArray = StringSplit(sLineRead,";")

        // Retrieving the values of the array
        sModuleCode      = arrLineArray[6]
        sStaffNumber     = arrLineArray[9]

        // Check if Record exists
        HReadSeekFirst(ModuleLeader,ModuleCodeStaffNumber,sModuleCode+sStaffNumber)
        IF HFound() = True THEN      // Exists continue to new record
            sLineRead = fReadLine(nFileNum)
            CONTINUE
        END

        // Add to ModuleLeader
        ModuleLeader.StaffNumber      = sStaffNumber
        ModuleLeader.ModuleCode        = sModuleCode
        HAdd(ModuleLeader)

        // Read the next line
        sLineRead = fReadLine(nFileNum)
    END
    fClose(nFileNum)
END

```

**Procedure UploadModuleInfo:**

```

PROCEDURE UploadModuleInfo(sPath)
// Import StaffInfo Data
// Variable declaration
nFileNum           is int
sLineRead          is string
arrLineArray       is array of strings

sModuleCode        is string
sDescription       is string

nFileNum = fOpen (sPath, foRead)
IF nFileNum <> -1 THEN
    // Read the 1st line
    sLineRead = fReadLine(nFileNum)

    // Check the end of file
    IF sLineRead <> EOT THEN
        // Skip first line
        sLineRead = fReadLine(nFileNum)
    END
    WHILE sLineRead <> EOT
        // Process the line
        // Put line values in a string array
        arrLineArray = StringSplit(sLineRead, ";")

        // Retrieving the values of the array
        sModuleCode = arrLineArray[6]

        // Check if Record exists
        HReadSeekFirst(ModuleInfo, ModuleCode, sModuleCode)
        IF HFound() = True THEN      // Exists continue to new record
            sLineRead = fReadLine(nFileNum)
            CONTINUE
        END

        // Else get rest of data
        sDescription = arrLineArray[7]

        // Add to ModuleInfo
        ModuleInfo.ModuleCode = sModuleCode
        ModuleInfo.Description = sDescription
        HAdd(ModuleInfo)

        // Read the next line
        sLineRead = fReadLine(nFileNum)
    END
    fClose(nFileNum)
END

```

**Procedure UploadExamSetup:**

```

PROCEDURE UploadExamSetup(sPath)
// Import StaffInfo Data
// Variable declaration
nFileNum           is int
sLineRead          is string
arrLineArray       is array of strings

sModuleCode        is string
sDate              is string
tDateExam          is Date
sExamPaperPDF      is string

nFileNum = fOpen (sPath, foRead)
IF nFileNum <> -1 THEN
    // Read the 1st line
    sLineRead = fReadLine(nFileNum)

    // Check the end of file
    IF sLineRead <> EOT THEN
        // Skip first line
        sLineRead = fReadLine(nFileNum)
    END
    WHILE sLineRead <> EOT
        // Process the line
        // Put line values in a string array
        arrLineArray = StringSplit(sLineRead, ";")

        // Retrieving the values of the array
        sModuleCode = arrLineArray[6]

        // Check if Record exists
        HReadSeekFirst(ExamSetup,ModuleCode,sModuleCode)
        IF HFound() = True THEN      // Exists continue to new record
            sLineRead = fReadLine(nFileNum)
            CONTINUE
        END

        // Get and set StartTime
        sDate = arrLineArray[13]      // Remove characters
        sDate = Replace(sDate,"-","")
        sDate = Replace(sDate,":","");
        tDateExam = sDate

        // Else get rest of data
        sExamPaperPDF = arrLineArray[15]

```

```

// Add to ExamSetup
ExamSetup.ModuleCode          = sModuleCode
ExamSetup.DateExam            = tDateExam
ExamSetup.ExamPaperPDF        = sExamPaperPDF
HAdd(ExamSetup)

// Read the next line
sLineRead = fReadLine(nFileNum)
END
fClose(nFileNum)
END

```

**Procedure UploadStudentInfo:**

```

PROCEDURE UploadStudentInfo(sPath)
// Import StaffInfo Data
// Variable declaration
nFileNum           is int
sLineRead          is string
arrLineArray       is array of strings

sStudentNumber     is string
sStudentName       is string
sStudentPassword   is string
sStudentEmail      is string

nFileNum = fOpen (sPath, foRead)
IF nFileNum <> -1 THEN
    // Read the 1st line
    sLineRead = fReadLine(nFileNum)

    // Check the end of file
    IF sLineRead <> EOT THEN
        // Skip first line
        sLineRead = fReadLine(nFileNum)
    END
    WHILE sLineRead <> EOT
        // Process the line
        // Put line values in a string array
        arrLineArray          = StringSplit(sLineRead,";")

        // Retrieving the values of the array
        sStudentNumber         = arrLineArray[2]

        // Check if Record exists
        HReadSeekFirst(StudentInfo,StudentNumber,sStudentNumber)
        IF HFound() = True THEN    // Exists continue to new record
            sLineRead = fReadLine(nFileNum)
            CONTINUE

```

```

// Else get rest of data
sStudentName          = arrLineArray[3]
sStudentPassword      = arrLineArray[5]
sStudentEmail         = arrLineArray[4]

// Add to StudentInfo
StudentInfo.StudentNumber    = sStudentNumber
StudentInfo.StudentName       = sStudentName
StudentInfo.StudentPassword   = sStudentPassword
StudentInfo.StudentEmail      = sStudentEmail
HAdd(StudentInfo)

// Read the next line
sLineRead = fReadLine(nFileNum)
END
fClose(nFileNum)
END

```

**Procedure UploadStudentModule:**

```

PROCEDURE UploadStudentModule(sPath)
// Import StaffInfo Data
// Variable declaration
nFileNum           is int
sLineRead          is string
arrLineArray       is array of strings

sModuleCode        is string
sStudentNumber     is string

nFileNum = fOpen (sPath, foRead)
IF nFileNum <> -1 THEN
    // Read the 1st line
    sLineRead = fReadLine(nFileNum)

    // Check the end of file
    IF sLineRead <> EOT THEN
        // Skip first line
        sLineRead = fReadLine(nFileNum)
    END
    WHILE sLineRead <> EOT
        // Process the line
        // Put line values in a string array
        arrLineArray          = StringSplit(sLineRead,";")

        // Retrieving the values of the array
        sStudentNumber        = arrLineArray[2]
        sModuleCode           = arrLineArray[6]

```

```

// Check if Record exists
HReadSeekFirst(StudentModule,StudentNumberModuleCode,sStudentNumber+sModuleCode)
IF HFound() = True THEN      // Exists continue to new record
    sLineRead = fReadLine(nFileNum)
    CONTINUE
END

// Add to StudentModule
StudentModule.StudentNumber          = sStudentNumber
StudentModule.ModuleCode             = sModuleCode
HAdd(StudentModule)

// Read the next line
sLineRead = fReadLine(nFileNum)
END
fClose(nFileNum)
END

```

### Procedure UploadExamOutput:

---

```

PROCEDURE UploadExamOutput(sPath)
// Import ExamOutput Data
// Variable declaration
nFileNum           is int
sLineRead          is string
arrLineArray       is array of strings

sTransactionID    is string
sDateAndTime      is string
tStartTime        is DateTime
tUploadTime       is DateTime
sAnswerPaperPDF   is string
sStudentNumber     is string
sModuleCode        is string

nFileNum = fOpen (sPath, foRead)
IF nFileNum <> -1 THEN
    // Read the 1st line
    sLineRead = fReadLine(nFileNum)

    // Check the end of file
    IF sLineRead <> EOT THEN
        // Skip first line
        sLineRead = fReadLine(nFileNum)
    END

```

```

WHILE sLineRead <> EOT
    // Process the line
    // Put line values in a string array
    arrLineArray = StringSplit(sLineRead,";")

    // Retrieving the values of the array
    sTransactionID           = arrLineArray[1]

    // Check if Record exists
    HReadSeekFirst(ExamOutput, TransactionID, sTransactionID)
    IF HFound() = True THEN      // Exists continue to new record
        sLineRead = fReadLine(nFileNum)
        CONTINUE
    END

    sAnswerPaperPDF          = arrLineArray[16]
    sStudentNumber            = arrLineArray[2]
    sModuleCode                = arrLineArray[6]

    // Get and set StartTime
    sDateAndTime               = arrLineArray[13]      // Remove characters
    sDateAndTime               = Replace(sDateAndTime,"-", "")
    sDateAndTime               = Replace(sDateAndTime,":", "")
    tStartTime                  = sDateAndTime

    // Get and set UploadTime
    sDateAndTime               = arrLineArray[14]
    // Remove characters : and -
    sDateAndTime               = Replace(sDateAndTime,"-", "")
    sDateAndTime               = Replace(sDateAndTime,":", "")
    tUploadTime                 = sDateAndTime

    // Add to ExamOutput
    ExamOutput.TransactionID     = sTransactionID
    ExamOutput.StartTime         = tStartTime
    ExamOutput.UploadTime        = tUploadTime
    ExamOutput.AnswerPaperPDF    = sAnswerPaperPDF
    ExamOutput.StudentNumber      = sStudentNumber
    ExamOutput.ModuleCode         = sModuleCode
    HAdd(ExamOutput)

    // Read the next line
    sLineRead = fReadLine(nFileNum)
END
fClose(nFileNum)
END

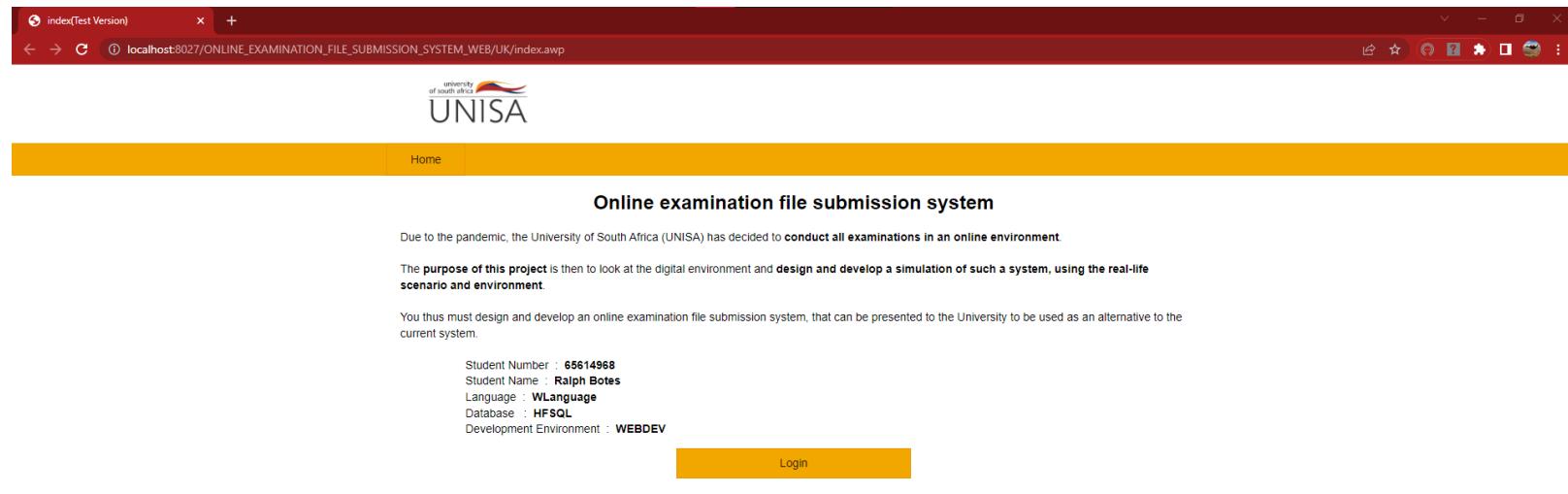
```

### Implementation of Procedures:

---

To implement these procedures and upload all the data, I started with my site creation.

I started with a simple Home page with some of the project details to start off with. A Login button is visible that redirects the user to the login page.



The Login page consists of a simple user number and password input fields. Added are details for different login users but at this stage we are going to concentrate on the UNISA Admin Login details. After the details are entered and validated, an admin session is created, and the user is directed to the Admin Dashboard page. (More on the login procedure later.)

A technical problem has occurred

Login(Test Version)

localhost:8027/WD270AWP/WD270Awp.exe/CONNECT/Online%20examination%20file%20submission%20system?\_WWREFERER\_=&\_WWNATION\_=3

university of south africa  
UNISA

Identification required

Before you continue, please identify yourself.

User Number:  Password:

UNISA Admin Login      OK      Cancel

User Number: 1000000  
Password:

ExamDept Login  
User Number: 9999999  
Password: acd25AFG

Staff Login  
User Number: 5133768  
Password: vly96SZM

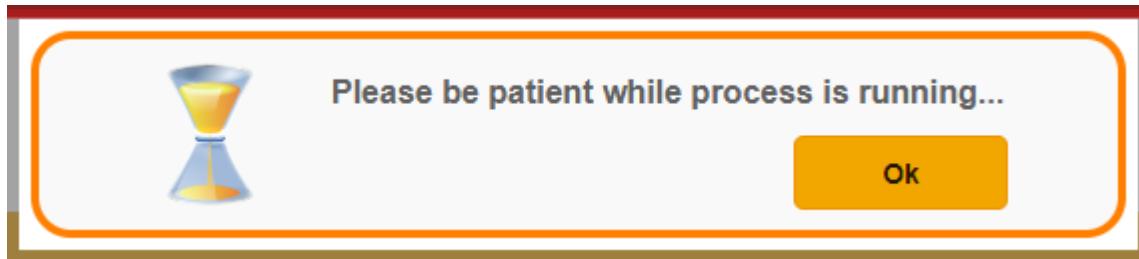
Student Login  
User Number: 29659162  
Password: gaq57UWR

Site powered by WEBDEV from PC SOFT.

Once the Admin user is logged in, a file path edit field and 7 buttons are visible. The edit field is used to input the cleaned data in the csv file's location on the admins drive. After this field is added and valid, the user can start uploading the data, one table at a time, by simply selecting the buttons for the relevant table.

The screenshot shows a web browser window for 'localhost:8027'. The title bar includes tabs for 'A technical problem has occurred' and 'Admin(Test Version)'. The main content area is titled 'Welcome UNISAAdmin' and features a yellow header bar with the text 'Admin Dashboard'. Below this is a central panel with a white background and a thin orange border. At the top of this panel is a text input field labeled 'File Location' containing the path 'F:\ICT3715\_Project\Online examination file submission system\Unisa Documents'. Below this are seven yellow rounded rectangular buttons, each labeled with a file type: 'Upload StaffInfo', 'Upload StudentInfo', 'Upload ModuleInfo', 'Upload ExamSetup', 'Upload ModuleLeader', 'Upload StudentModule', and 'Upload ExamOutput'. At the bottom of the page, a small note reads 'Site powered by WEBDEV from PC SOFT.'

On each button select, a popup message appears that informs the Admin user that the process may take some time. When the user selects the Ok button, the process starts and once it ends, the popup closes, allowing the Admin user to select the next data import button or logout.



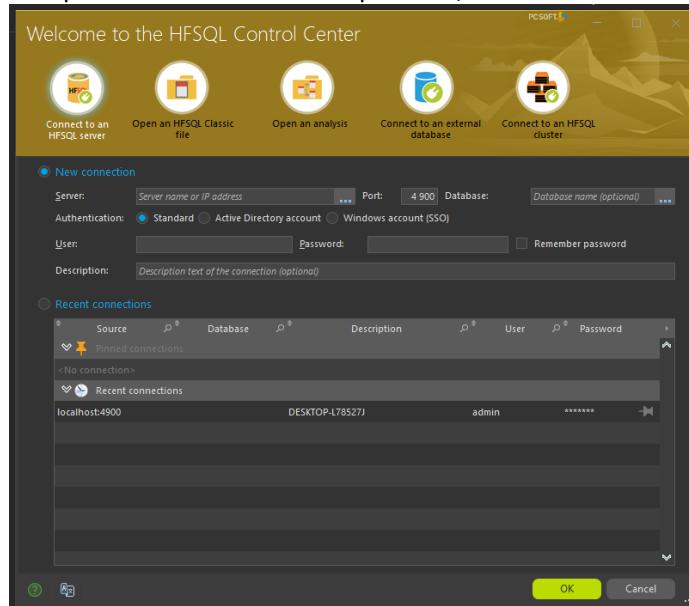
**The Data:**

To view the data after it has been successfully uploaded to the database:

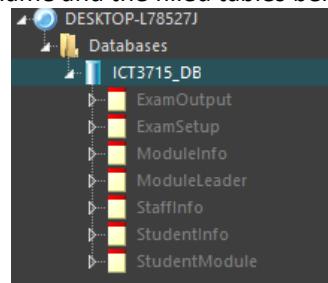
1. In WEBDEV, I select the HFSQL icon in the Tools tab.



2. This opens the HFSQL Control Center where I can connect to my HFSQL server and view the database. After I input the relevant data and press Ok, I enter the Command Center.



3. The window displays my database name and the filled tables below in a tree view form.



4. Below can be seen how the data has been filled into the ExamOutput table.

HFSQL Control Center

The screenshot shows the HFSQL Control Center interface with the 'ExamOutput' table selected. The table contains numerous rows of data, each representing an examination record. The columns include Rec #, TransactionID, StartTime, UploadTime, AnswerPaperPDF, StudentNumber, ModuleCode, and StudentNumberModuleCode. The data spans from 2022/11/10 to 2022/11/17, with various student numbers and module codes. A sidebar on the left provides navigation links for Databases, SQL query, File, Rights, Display, and Content.

Rec #	TransactionID	StartTime	UploadTime	AnswerPaperPDF	StudentNumber	ModuleCode	StudentNumberModuleCode
► 6 995	AAKU6977-3275	2022/11/17 10:37:14	2022/11/17 10:45:41	69775304.ICT1118_EXAM_20221117-10:45:41.pdf	69775304	ICT1118	69775304.ICT1118
► 6 147	AaNE6282-1455	2022/11/14 10:10:01	2022/11/14 11:37:01	62822532_CHE2612_EXAM_20221114-11:37:01.pdf	62822532	CHE2613	62822532.CHE2613
► 8 299	ABWV0809-1864	2022/11/21 10:44:47	2022/11/21 11:44:30	80090703_ENG1019_EXAM_20221121-11:44:30.pdf	80090703	ENG1019	80090703.ENG1019
► 10 363	AbyG9676-1188	2022/11/15 10:28:19	2022/11/15 11:45:23	96767679_ENG1016_EXAM_20221115-11:45:23.pdf	96767679	ENG1016	96767679.ENG1016
► 7 989	ACaIT764-1245	2022/11/19 10:58:13	2022/11/19 11:17:59	77645472 ICT1120_EXAM_20221119-11:17:59.pdf	77645472	ICT1120	77645472.ICT1120
► 5 441	ACvV5668-1142	2022/11/14 10:29:20	2022/11/14 11:00:53	56688994_ENG1015_EXAM_20221114-11:00:53.pdf	56688994	ENG1015	56688994.ENG1015
► 9 337	ACCV8881-1046	2022/11/13 10:15:52	2022/11/13 10:19:41	88817755_CHE2611_EXAM_20221113-10:19:41.pdf	88817755	CHE2611	88817755.CHE2611
► 8 527	aCQN8178-7302	2022/11/10 11:28:58	2022/11/10 11:38:13	81789402_AST2652_EXAM_20221110-11:38:13.pdf	81789402	AST2652	81789402.AST2652
► 8 169	ACVw7920-4174	2022/11/10 10:49:02	2022/11/10 11:21:03	79206071_ICT1111_EXAM_20221110-11:21:03.pdf	79206071	ICT1111	79206071.ICT1111
► 5 379	ACXO5615-5806	2022/11/20 11:05:13	2022/11/20 11:59:14	56154127_ENG1018_EXAM_20221120-11:59:14.pdf	56154127	ENG1018	56154127.ENG1018
► 9 969	ADAS9395-1061	2022/11/11 10:47:15	2022/11/11 11:57:37	93558660_CHE1504_EXAM_20221111-11:57:37.pdf	93558660	CHE1504	93558660.CHE1504
► 5 499	ADDL5720-1514	2022/11/11 10:39:49	2022/11/11 11:32:54	57207757_ENG1012_EXAM_20221111-11:32:54.pdf	57207757	ENG1012	57207757.ENG1012
► 8 047	adTf814-8724	2022/11/11 11:50:48	2022/11/11 11:57:37	78149989_ICT3611_EXAM_20221111-11:57:37.pdf	78149989	ICT3611	78149989.ICT3611
► 3 547	ADkw4074-6090	2022/11/15 10:24:42	2022/11/15 10:35:58	40745951_ICT1116_EXAM_20221115-10:35:58.pdf	40745951	ICT1116	40745951.ICT1116
► 8 959	AdTg3557-4082	2022/11/15 11:06:05	2022/11/15 11:21:52	85379184_ICT1116_EXAM_20221115-11:21:52.pdf	85379184	ICT1116	85379184.ICT1116
► 6 097	aEGG241-5574	2022/11/12 10:59:27	2022/11/12 11:49:21	62418217_ENG1013_EXAM_20221112-11:49:21.pdf	62418217	ENG1013	62418217.ENG1013
► 8 155	AERL7904-6200	2022/11/21 11:42:34	2022/11/21 12:01:41	79060382_ENG1019_EXAM_20221121-12:01:41.pdf	79060382	ENG1019	79060382.ENG1019
► 1 765	aEqW2604-3253	2022/11/14 10:39:07	2022/11/14 11:40:45	26045254_CHE2613_EXAM_20221114-11:40:45.pdf	26045254	CHE2613	26045254.CHE2613
► 8 201	AEZt7934-7895	2022/11/12 10:36:57	2022/11/12 11:52:35	79343861_CHE181T_EXAM_20221112-11:52:35.pdf	79343861	CHE181T	79343861.CHE181T
► 8 125	aFDW2665-8256	2022/11/12 11:13:41	2022/11/12 11:18:16	26658955_ICT1113_EXAM_20221112-11:18:19.pdf	26658955	ICT1113	26658955.ICT1113
► 213	AgfQt321-4815	2022/11/15 10:48:29	2022/11/15 10:53:43	13211140_ENG1016_EXAM_20221115-10:53:43.pdf	13211140	ENG1016	13211140.ENG1016
► 8 951	AGUH8524-6591	2022/11/11 10:34:41	2022/11/11 10:48:27	85246241_ICT1112_EXAM_20221111-10:48:27.pdf	85246241	ICT1112	85246241.ICT1112
► 9 581	Agb9062-1256	2022/11/15 11:41:52	2022/11/15 11:50:19	90625157_CHE3701_EXAM_20221115-11:50:19.pdf	90625157	CHE3701	90625157.CHE3701
► 1 643	aGFb2523-5530	2022/11/15 10:28:25	2022/11/15 11:07:48	25253693_ICT1116_EXAM_20221115-11:07:48.pdf	25253693	ICT1116	25253693.ICT1116
► 1 788	aGfG2633-5482	2022/11/13 11:38:03	2022/11/13 11:52:55	26333882_CHE2611_EXAM_20221113-11:52:55.pdf	26333882	CHE2611	26333882.CHE2611
► 3 063	aghU3703-6650	2022/11/16 10:09:14	2022/11/16 11:43:43	37032496_ICT1117_EXAM_20221116-11:43:43.pdf	37032496	ICT1117	37032496.ICT1117
► 7 025	agK6999-9936	2022/11/14 11:19:25	2022/11/14 11:41:49	69998350_ENG1015_EXAM_20221114-11:41:49.pdf	69998350	ENG1015	69998350.ENG1015
► 2 899	agL9356-4546	2022/11/19 10:49:25	2022/11/19 10:54:57	35680037_ELE2561_EXAM_20221119-10:54:57.pdf	35680037	ELE2561	35680037.ELE2561
► 5 533	agow1618-9122	2022/11/15 10:04:55	2022/11/15 10:23:57	16180306_ICT1116_EXAM_20221115-10:23:57.pdf	16180306	ICT1116	16180306.ICT1116
► 3 441	AgSd3997-2693	2022/11/16 10:41:57	2022/11/16 11:50:09	39979752_ENG1017_EXAM_20221116-11:50:09.pdf	39979752	ENG1017	39979752.ENG1017
► 1 679	aGSy2545-5482	2022/11/14 11:00:10	2022/11/14 11:52:49	25457526_ENG1015_EXAM_20221114-11:52:49.pdf	25457526	ENG1015	25457526.ENG1015
► 2 805	AGUV3466-1094	2022/11/19 10:20:53	2022/11/19 11:28:56	34661111ICT1120_EXAM_20221119-11:28:56.pdf	34661111	ICT1120	34661111.ICT1120
► 1 455	agVb2325-9763	2022/11/11 10:09:50	2022/11/11 10:22:05	23256795_ICT1112_EXAM_20221111-10:22:05.pdf	23256795	ICT1112	23256795.ICT1112
► 3 871	agYd4334-1002	2022/11/18 10:18:22	2022/11/18 10:52:54	43347882_ICT1119_EXAM_20221118-10:52:54.pdf	43347882	ICT1119	43347882.ICT1119
► 10 089	Agvn9475-1072	2022/11/19 11:00:49	2022/11/19 11:36:42	94753373_ICT1120_EXAM_20221119-11:36:42.pdf	94753373	ICT1120	94753373.ICT1120
► 3 291	AGyA3889-5121	2022/11/19 11:30:10	2022/11/19 11:52:24	38892383_ICT1120_EXAM_20221119-11:52:24.pdf	38892383	ICT1120	38892383.ICT1120
► 8 343	agzy0041-3286	2022/11/13 10:18:55	2022/11/13 10:54:29	80416107_ENG1014_EXAM_20221113-10:54:29.pdf	80416107	ENG1014	80416107.ENG1014
► 2 229	ahbo3015-2085	2022/11/14 10:03:07	2022/11/14 10:46:08	30159486_CHE2613_EXAM_20221114-10:46:08.pdf	30159486	CHE2613	30159486.CHE2613
► 45	ahext156-8866	2022/11/10 10:46:31	2022/11/10 11:20:44	11566745_ENG1011_EXAM_20221110-11:20:44.pdf	11566745	ENG1011	11566745.ENG1011
► 5 853	AHKa6027-1046	2022/11/13 11:26:48	2022/11/13 11:29:44	60270496_ICT1114_EXAM_20221113-11:29:44.pdf	60270496	ICT1114	60270496.ICT1114
► 5 463	AHNB56RL-7852	2022/11/17 16:41:44	2022/11/17 17:31:04	56834008_ICT1118_EXAM_20221117-17:31:04.pdf	56834008	ICT1118	56834008.ICT1118

Total number of records: 10 740

Close

5. Following are the rest of the filled tables.

The screenshot shows the HFSQL Control Center interface with the 'StudentModule' table selected. The table has columns: Rec #, Description, StudentNumber, ModuleCode. The data grid contains 740 rows of student-module assignments. A sidebar on the left lists various database objects like ExamOutput, ExamSetup, ModuleInfo, etc. The bottom status bar indicates a total of 10,740 records.

Rec #	Description	StudentNumber	ModuleCode
1	11141311,ICT1113	11141311	ICT1113
2	11141311,AST2652	11141311	AST2652
3	11153623,CHE1504	11153623	CHE1504
4	11153623,ICT1114	11153623	ICT1114
5	11162384,ICT1115	11162384	ICT1115
6	11162384,CHE181T	11162384	CHE181T
7	11207520,CHE2611	11207520	CHE2611
8	11207520,ICT1116	11207520	ICT1116
9	11260693,CHE2613	11260693	CHE2613
10	11260693,ICT1117	11260693	ICT1117
11	11261613,ICT1118	11261613	ICT1118
12	11261613,CHE3701	11261613	CHE3701
13	11276374,CHE3703	11276374	CHE3703
14	11276374,ICT1119	11276374	ICT1119
15	11286271,CHE3704	11286271	CHE3704
16	11286271,ICT1120	11286271	ICT1120
17	11296841,ICT2612	11296841	ICT2612
18	11296841,CHE4801	11296841	CHE4801
19	11348843,ELE2561	11348843	ELE2561
20	11348843,ICT3611	11348843	ICT3611
21	11350641,ICT3722	11350641	ICT3722
22	11350641,ENG1011	11350641	ENG1011
23	11357287,MAT2612	11357287	MAT2612
24	11357287,ENG1012	11357287	ENG1012
25	11372737,AST2652	11372737	AST2652
26	11372737,ENG1013	11372737	ENG1013
27	11384884,ENG1014	11384884	ENG1014
28	11384884,CHE1504	11384884	CHE1504
29	11447994,ENG1015	11447994	ENG1015
30	11447994,CHE181T	11447994	CHE181T
31	11477936,CHE2611	11477936	CHE2611
32	11477936,ENG1016	11477936	ENG1016
33	11483352,ENG1017	11483352	ENG1017
34	11483352,CHE2613	11483352	CHE2613
35	11490131,CHE3701	11490131	CHE3701
36	11490131,ENG1018	11490131	ENG1018
37	11492193,CHE3703	11492193	CHE3703
38	11492193,ENG1019	11492193	ENG1019
39	11523952,ICT1111	11523952	ICT1111
40	11523952,CHE3704	11523952	CHE3704
41	11533815,CHE4801	11533815	CHE4801

Total number of records: 10,740

HFSQL Control Center

The screenshot shows the HFSQL Control Center interface. On the left, there's a sidebar with navigation links: DESKTOP-L785271, Databases (with ICT3715\_DB selected), New query, Open a query, Optimize and repair, Manage rights, Refresh, Delete, Content, Description, Properties, Connections, Locks, Content, Search, Logs, Transactions, and Help. The main area displays a table titled 'StudentInfo' with columns: Rec #., StudentNumber, StudentName, StudentPassword, and StudentEmail. The table contains 5370 records. The first few records are:

Rec #.	StudentNumber	StudentName	StudentPassword	StudentEmail
111	11141311	OD MASEKO	*****	11141311@mylife.unisa.ac.za
573	11153623	KY MAKHAFOLA	*****	11153623@mylife.unisa.ac.za
643	11162384	QP KALENGA	*****	11162384@mylife.unisa.ac.za
1 133	11207520	UU MAPHUTHA	*****	11207520@mylife.unisa.ac.za
1 473	11260693	EU UBSI	*****	11260693@mylife.unisa.ac.za
1 560	11261613	PT NGQENGELELE	*****	11261613@mylife.unisa.ac.za
1 841	11276374	QD MACHETE	*****	11276374@mylife.unisa.ac.za
2 121	11286271	TN MMOLOTSI	*****	11286271@mylife.unisa.ac.za
2 408	11296841	EX MPHABELE	*****	11296841@mylife.unisa.ac.za
2 584	11348843	UX SEWNARAIN	*****	11348843@mylife.unisa.ac.za
2 932	11350641	CK GONDO	*****	11350641@mylife.unisa.ac.za
3 030	11357287	CV MANDIPIRA	*****	11357287@mylife.unisa.ac.za
233	11372737	XH THWALA	*****	11372737@mylife.unisa.ac.za
581	11384884	RQ SIBNDLANA	*****	11384884@mylife.unisa.ac.za
874	11447994	VJ MHLONGO	*****	11447994@mylife.unisa.ac.za
998	11477936	SR HANKEY	*****	11477936@mylife.unisa.ac.za
1 215	11483332	XO CHARUMBIRA	*****	11483332@mylife.unisa.ac.za
1 754	11490131	W MOKOENA	*****	11490131@mylife.unisa.ac.za
1 845	11492193	EP GILL	*****	11492193@mylife.unisa.ac.za
2 049	11523952	UD CHINYANGA	*****	11523952@mylife.unisa.ac.za
2 339	11533815	U MANGOUA MENDIJA	*****	11533815@mylife.unisa.ac.za
2 626	11536884	PB MTJWARA	*****	11536884@mylife.unisa.ac.za
2 767	11566745	VN MAKHUDU	*****	11566745@mylife.unisa.ac.za
3 173	11582146	KM MAPHWANZA	*****	11582146@mylife.unisa.ac.za
3 228	11627788	JD WILLIAMS	*****	11627788@mylife.unisa.ac.za
3 229	11628404	OA MAPETA	*****	11628404@mylife.unisa.ac.za
3 230	11634229	Z NKHUMLENI	*****	11634229@mylife.unisa.ac.za
3 231	11641994	FK MPHABELE	*****	11641994@mylife.unisa.ac.za
3 232	11642685	KW VAN DER MERWE	*****	11642685@mylife.unisa.ac.za
3 233	11648513	ZS THUPAVAKGOSI	*****	11648513@mylife.unisa.ac.za
3 234	11676666	SB STEVENSON	*****	11676666@mylife.unisa.ac.za
3 235	11702638	UP NAMWIHA	*****	11702638@mylife.unisa.ac.za
3 236	11704823	WI MINYAI	*****	11704823@mylife.unisa.ac.za
197	11713318	RJ MOTSHUMI	*****	11713318@mylife.unisa.ac.za
352	11747258	MV MASHA	*****	11747258@mylife.unisa.ac.za
727	11823496	A MTSHALI	*****	11823496@mylife.unisa.ac.za
1 017	11855287	VY MANSOOR	*****	11855287@mylife.unisa.ac.za
1 424	11885511	UP TSOTETSI	*****	11885511@mylife.unisa.ac.za
1 532	11963068	QP GWEBU	*****	11963068@mylife.unisa.ac.za
1 976	11978715	IQ BONKE	*****	11978715@mylife.unisa.ac.za
2 133	12010166	MF MANAMA	*****	12010166@mylife.unisa.ac.za

Total number of records: 5 370

HFSQL Control Center

Rec #.	StaffNumber	Name	Email	Password
11	1389884	D Nel	d.nel4@unisa.ac.za	*****
9	1550751	F Pretorius	f.pretorius@unisa.ac.za	*****
24	1722194	PG Smith	pg.smith@unisa.ac.za	*****
13	1927785	L Bussu	l.bussu@unisa.ac.za	*****
21	2067863	D Zulu	d.zulu@unisa.ac.za	*****
8	3858852	PK Singh	pk.singh@unisa.ac.za	*****
10	4102703	FCD Mhalatsi	fcd.mhalatsi@unisa.ac.za	*****
12	4511492	TK Debeer	tk.debeer@unisa.ac.za	*****
15	4556445	M Matzi	m.matzl@unisa.ac.za	*****
18	4760241	OD Bhojle	od.bhojle@unisa.ac.za	*****
20	4866926	VV Erasmus	vv.erasmus@unisa.ac.za	*****
19	4975413	MD Naidoo	md.naidoo@unisa.ac.za	*****
1	5017037	JH Nel	jh.nel@unisa.ac.za	*****
3	5133768	LM Nikosi	lm.nikosi@unisa.ac.za	*****
16	5489641	MWS Sharpwell	mws.sharpwell@unisa.ac.za	*****
2	5509541	B Baxter	b.baxter@unisa.ac.za	*****
14	6249547	PA Ramaposa	pa.ramaposa@unisa.ac.za	*****
23	6939679	SS Strydom	ss.strydom@unisa.ac.za	*****
4	7443352	NN Vanzyl	nn.vanzyl@unisa.ac.za	*****
17	7460174	MT Seopa	mt.seopa@unisa.ac.za	*****
6	7912653	SX Sithole	sx.sithole@unisa.ac.za	*****
22	8404212	W Timakwi	w.timakwi@unisa.ac.za	*****
7	9025988	K Tau	k.tau@unisa.ac.za	*****
5	9530053	PJ Sithebe	pj.sithebe@unisa.ac.za	*****

Total number of records: 24

The screenshot shows the HFSQL Control Center interface. On the left, there is a sidebar with various database-related options: DESKTOP-L78527, Databases (including ICT3715\_DB, ExamOutput, ExamSetup), ModuleInfo (selected), ModuleLeader, StaffInfo, StudentInfo, and StudentModule. The main area displays a table titled "ModuleInfo" with columns: Rec #., ModuleCode, and Description. The table lists 33 records. The first few rows show:

Rec #.	ModuleCode	Description
1	AST2652	Accounting Sciences II
2	CHE1504	Introduction to Chemistry I
3	CHE181T	Chemistry I (Theory)
4	CHE2611	Inorganic Chemistry II (Theory)
5	CHE2613	Organic Chemistry II (Theory)
6	CHE3701	Inorganic Chemistry III (Theory)
7	CHE3703	Organic Chemistry III (Theory)
8	CHE3704	Analytical Chemistry III
9	CHE4801	Inorganic Chemistry IV
10	ELE2561	Electrical Engineering II (Theory)
11	ENG1011	Engineering Methods
12	ENG1012	Engineering Design
13	ENG1013	Engineering Smart systems
14	ENG1014	Engineering Numerical analysis
15	ENG1015	Design and Manufacture I
16	ENG1016	Mechanical Design I
17	ENG1017	Engineering Materials
18	ENG1018	Engineering Science
19	ENG1019	Engineering Practice I
20	ICT1111	Introduction to Programming
21	ICT1112	Introduction to GUI
22	ICT1113	Introduction to Web Application
23	ICT1114	Introduction to Databases
24	ICT1115	Introduction to Information Systems
25	ICT1116	Introduction to Operating Systems
26	ICT1117	Introduction to Formal Logic
27	ICT1118	Language and Life Skills
28	ICT1119	End User Computing
29	ICT120	Introduction to Applications
30	ICT2612	Internet Programming II
31	ICT3611	Internet Programming III
32	ICT3722	Advanced Databases
33	MAT2612	Mathematics II

Total number of records: 33

HFSQL Control Center

The screenshot shows the HFSQL Control Center application window. On the left is a sidebar with the following items:

- DESKTOP-L78527J
- Databases
  - ICT3715\_DB
  - ExamOutput
  - ExamSetup**
  - ModuleInfo
  - ModuleLeader
  - StaffInfo
  - StudentInfo
  - StudentModule
- New query
- Open a query
- Optimize and repair
- Manage rights
- Refresh
- Delete
- SQL query
- Description
- Properties
- Connections
- Locks
- Content
- Search
- Logs
- Transactions

The main content area displays a table titled "Content" with the following columns:

Rec #.	Description	ModuleCode	DateExam	ExamPaperPDF	ModuleCodeStaffNumber
2	AST2652		2022/11/10	AST2652.pdf	
3	CHE1504		2022/11/11	CHE1504.pdf	
6	CHE181T		2022/11/12	CHE181T.pdf	
7	CHE2611		2022/11/13	CHE2611.pdf	
9	CHE2613		2022/11/14	CHE2613.pdf	
12	CHE3701		2022/11/15	CHE3701.pdf	
13	CHE3703		2022/11/16	CHE3703.pdf	
15	CHE3704		2022/11/17	CHE3704.pdf	
18	CHE4801		2022/11/18	CHE4801.pdf	
19	ELE2561		2022/11/19	ELE2561.pdf	
22	ENG1011		2022/11/10	ENG1011.pdf	
24	ENG1012		2022/11/11	ENG1012.pdf	
25	ENG1013		2022/11/12	ENG1013.pdf	
26	ENG1014		2022/11/13	ENG1014.pdf	
27	ENG1015		2022/11/14	ENG1015.pdf	
28	ENG1016		2022/11/15	ENG1016.pdf	
29	ENG1017		2022/11/16	ENG1017.pdf	
30	ENG1018		2022/11/20	ENG1018.pdf	
31	ENG1019		2022/11/21	ENG1019.pdf	
32	ICT1111		2022/11/10	ICT1111.pdf	
33	ICT1112		2022/11/11	ICT1112.pdf	
1	ICT1113		2022/11/12	ICT1113.pdf	
4	ICT1114		2022/11/13	ICT1114.pdf	
5	ICT1115		2022/11/14	ICT1115.pdf	
8	ICT1116		2022/11/15	ICT1116.pdf	
10	ICT1117		2022/11/16	ICT1117.pdf	
11	ICT1118		2022/11/17	ICT1118.pdf	
14	ICT1119		2022/11/18	ICT1119.pdf	
16	ICT1120		2022/11/19	ICT1120.pdf	
17	ICT2612		2022/11/10	ICT2612.pdf	
20	ICT3611		2022/11/11	ICT3611.pdf	
21	ICT3722		2022/11/12	ICT3722.pdf	
23	MAT2612		2022/11/10	MAT2612.pdf	

Total number of records: 33

**Problems I had**

1. When designing my procedures, I had some trouble getting to the right format for my dates and times to upload to my database. I used the WEBDEV's Debugger to keep track of how the data changed at every line. I created a few test variables and made single changes to each. WEBDEV's Debugger allows for live debugging that allowed me to view each line before and after it executed to determine what happened. I then used the tests I made and got the exact date and time formats I needed to upload the data to my database.
2. I struggled with the ERD / Data Analysis diagram as WEBDEV allows for foreign keys, but it is implemented in a different way, using relationships between data files. At the end of the day, as I created links, I created composite keys that directly allowed for unique keys to act as foreign keys in the data files that needed it.

At the end of the day, getting to where I am at the time of writing, everything went pretty flawless despite the two issues I had indicated above. As can be seen throughout this project, WEBDEV is quite a unique and simple Development Environment to use that allows for quick change between Back- and Front-end and easy debugging to fix any errors and issues that may arise.

## Section B = Backup and Recovery for the Database and Programming code [10]

### Database (5)

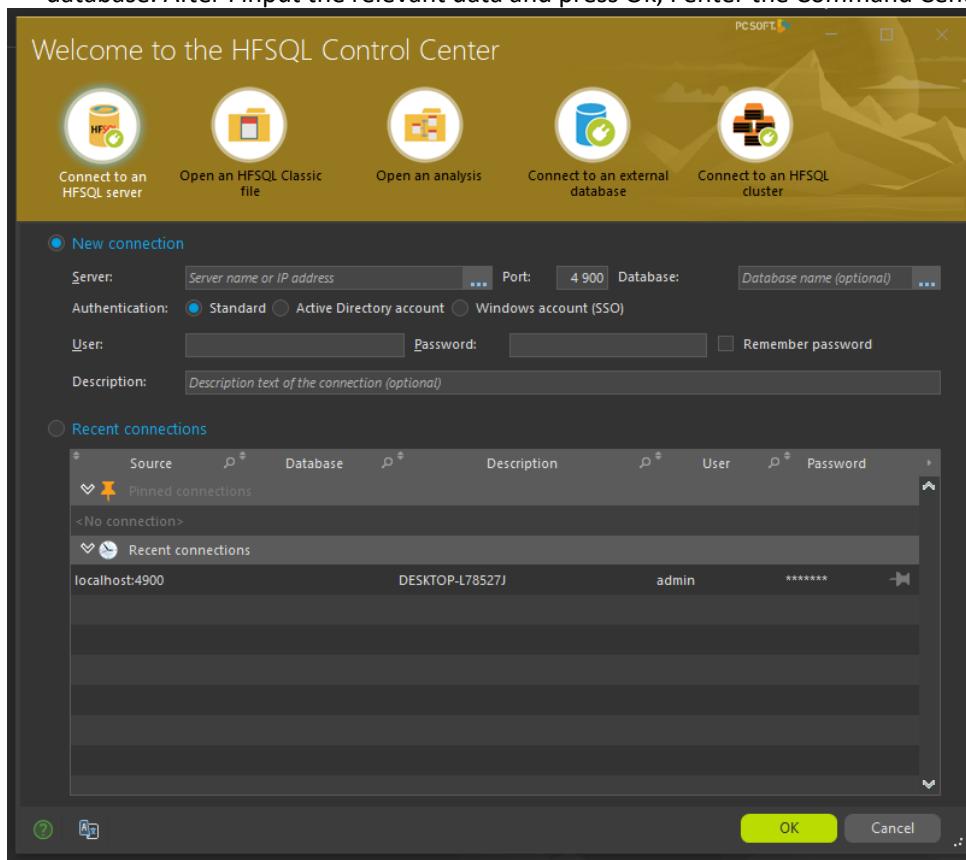
#### Database Backup

WEBDEV HFSQL Control Centre, allows for easy Database backup and recovery on the server in HFSQL Client/Server. To backup the database, I will follow the following steps:

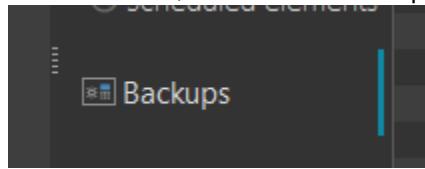
1. In WEBDEV, I select the HFSQL icon in the Tools tab.



2. This opens the HFSQL Control Center where I can connect to my HFSQL server and view the database. After I input the relevant data and press Ok, I enter the Command Center.



3. Once the HFSQL Control Centre is open, I select the Backups Tab.

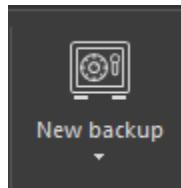


4. This opens the Backups internal window. Here, previous backups can be handled.

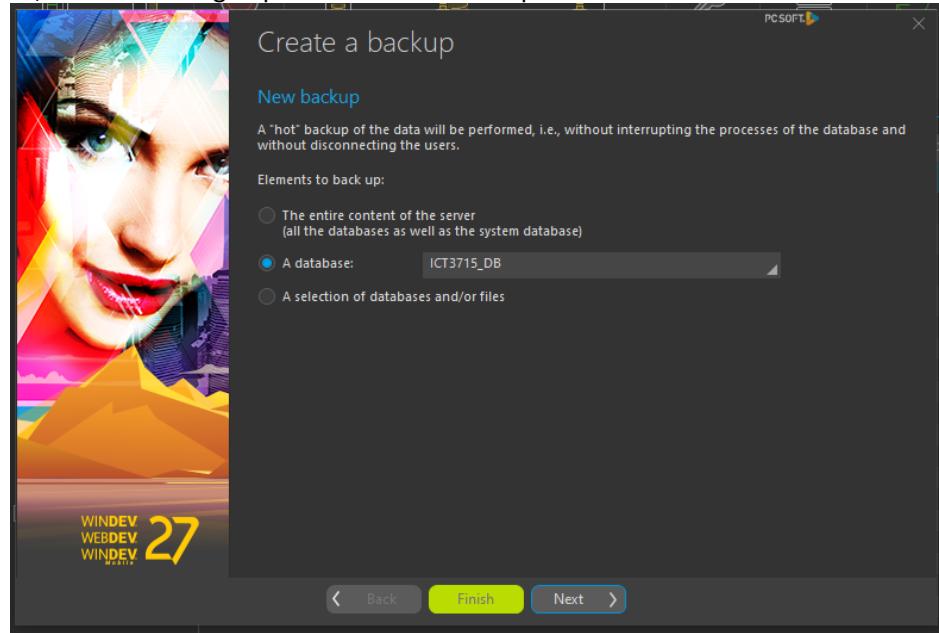
The screenshot shows the 'Backups completed' section of the HFSQL Control Centre. The table lists five backup entries:

ID	Status	Date	Description	Destination	Content
27	Backup completed	06/06/2022 at 14:26:42	Backup created by the HFSQL Control Center on 22/05/2022.	Database: ICT3715_DB	C:\PC\SOFT\HFSQL\DESKTOP-L7BS27\server-backup\Backup20220522134225.zip
25	Backup completed	06/06/2022 at 15:32:42	Backup created by the HFSQL Control Center on 22/05/2022.	Database: ICT3715_DB	C:\PC\SOFT\HFSQL\DESKTOP-L7BS27\server-backup\Backup20220522134225.zip
7	Backup completed	22/05/2022 at 14:28:22	Backup created by the HFSQL Control Center on 22/05/2022.	Database: ICT3715_DB	C:\PC\SOFT\HFSQL\DESKTOP-L7BS27\server-backup\Backup20220522134225.zip
6	Backup completed	22/05/2022 at 22:08:22	Backup created by the HFSQL Control Center on 22/05/2022.	Database: ICT3715_DB	C:\PC\SOFT\HFSQL\DESKTOP-L7BS27\server-backup\Backup20220522220822.zip
5	Backup completed	22/05/2022 at 15:54:27	Backup started by the HFSQL Control Center on 22/05/2022.	Database: ICT3715_DB	C:\PC\SOFT\HFSQL\DESKTOP-L7BS27\server-backup\Backup20220522135427.zip

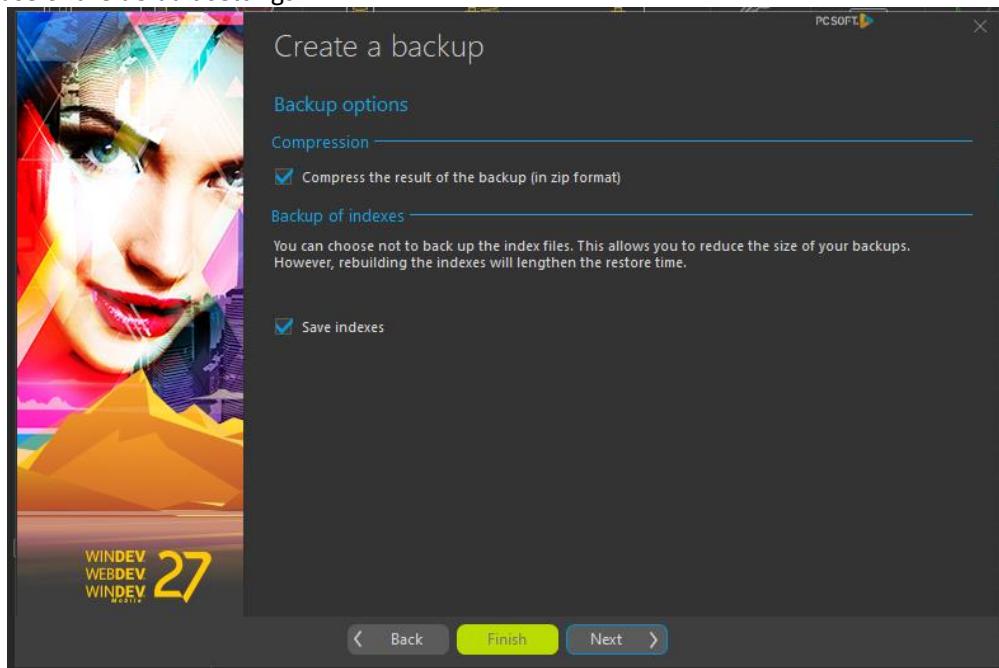
5. To create a backup, I select the New backup option.



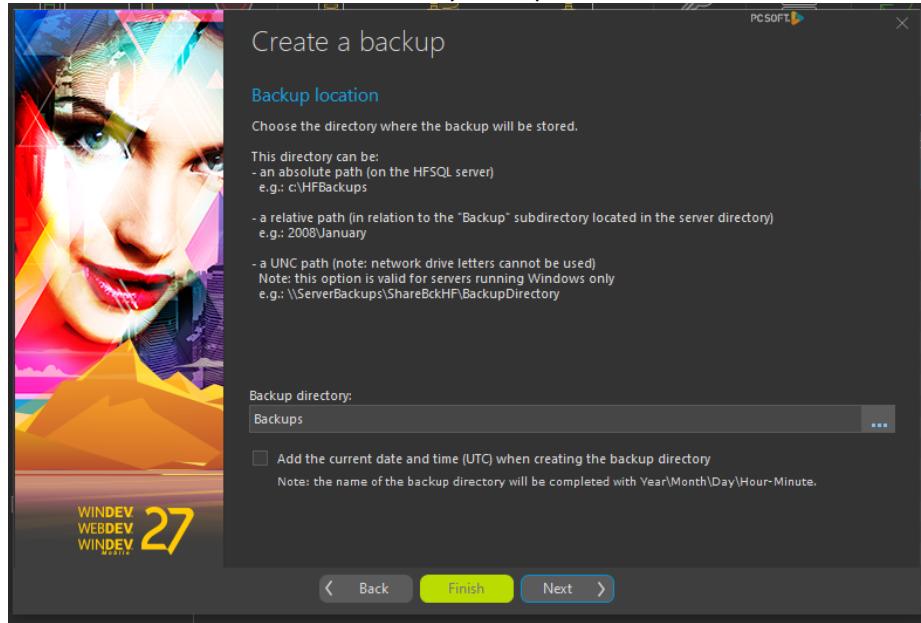
6. The Create Backup window is displayed. This is a hot backup, where a specific backup can be done. Scheduled backups are available, which I already implemented but this is a basic idea of how it is done. I use the hot backup as my main backup thus making sure it is performed especially before and after large changes. The hot backup window gives me different backup options, I will be making a specific database backup.



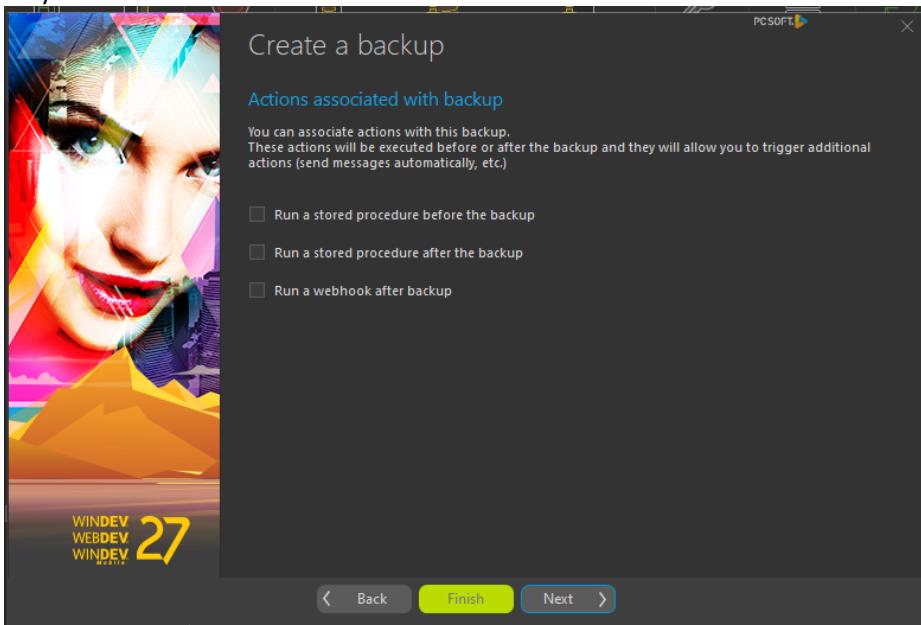
7. After selecting my database to back up, I can choose how the backup will be stored, I make use of the default settings.



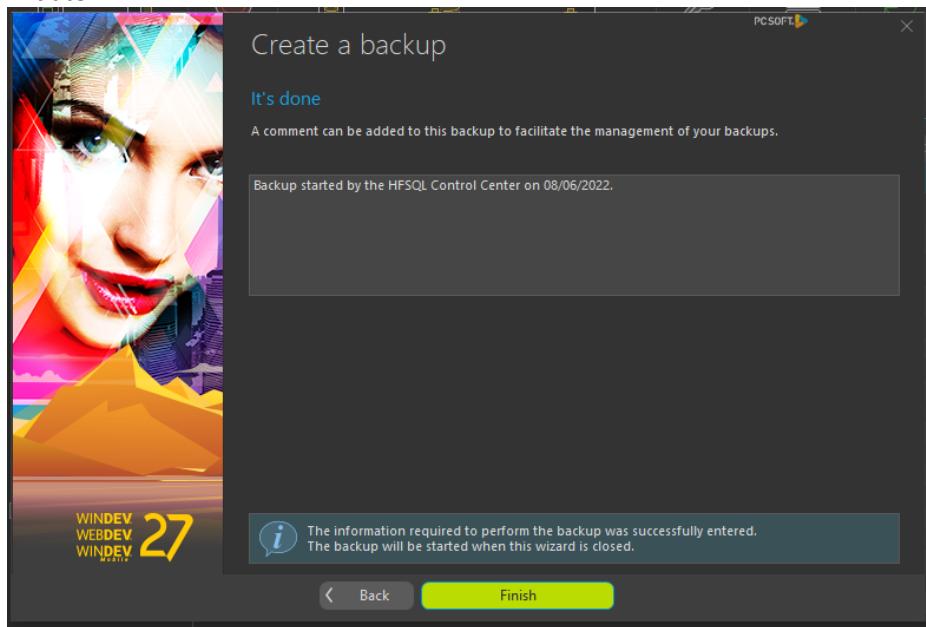
8. I select a directory where the backup will be saved. After the backup is done, I copy and paste the backup folder in different directories, one of which is my UNISA OneDrive file so there are multiple copies available to me, allowing me to be ready for any crashes that may occur. I also selected the date and time to be added to my backup file.



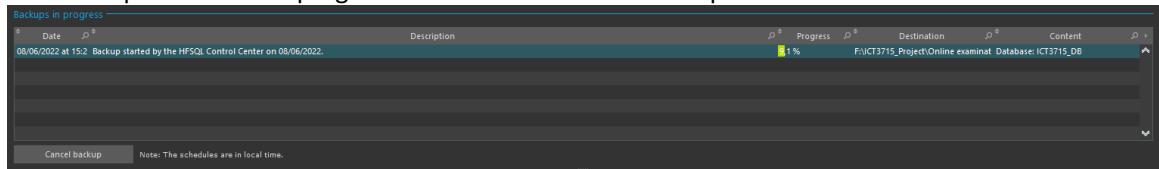
9. The following window allows for advanced additions to my backups. For my project, this is not necessary so I will move on.



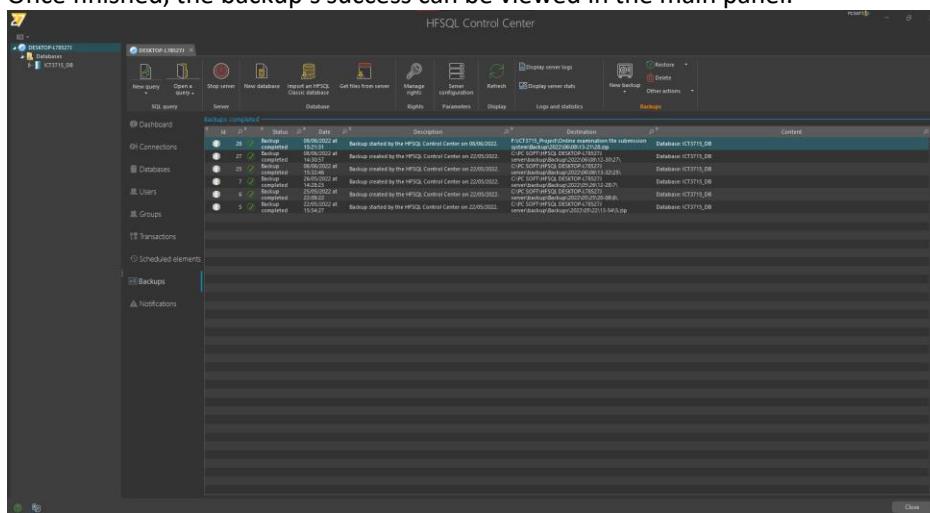
10. This window indicates the settings process is done and on Finish selected the backup will initiate.



11. The backup starts and its progress can be viewed in the Backups window.

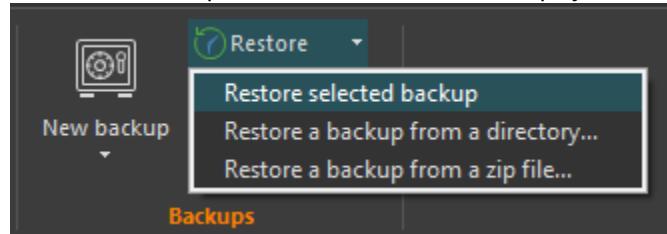


12. Once finished, the backup's success can be viewed in the main panel.

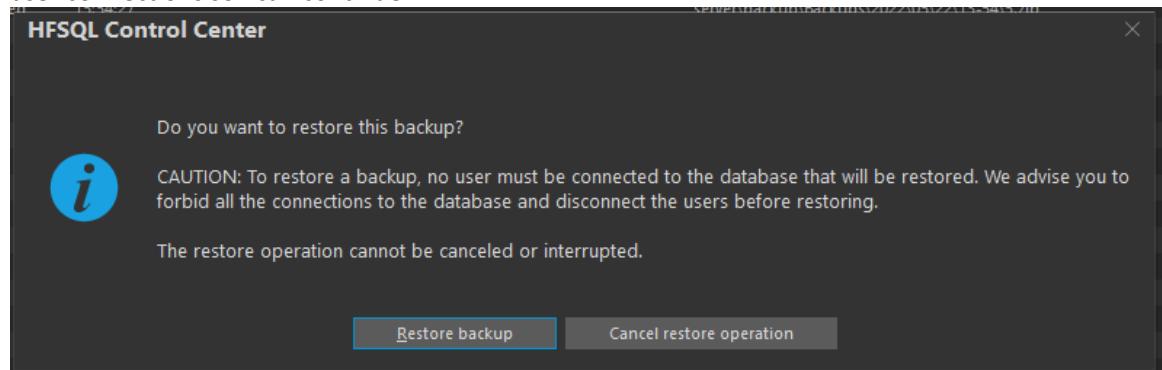


## Database Restore

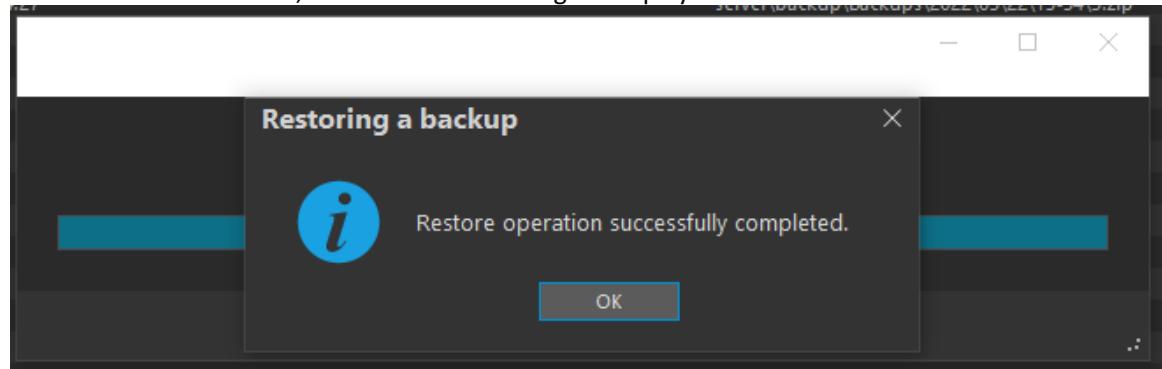
13. Restoring a backup can be done from the same window. Simply select the Restore Option and select a restore option, I will restore the backup I just made, It is already selected.



14. A popup window appears with some basic information regarding the restore. I have no active user connections so I can continue.



15. Once the Restore is done, a confirmation message is displayed.



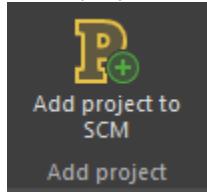
16. As I stated, I created a schedule backup for every day after I have worked on my database. The procedure described above is followed to do so with only the added option of setting the schedule.

## Programming code & Portfolio (assignments) (20)

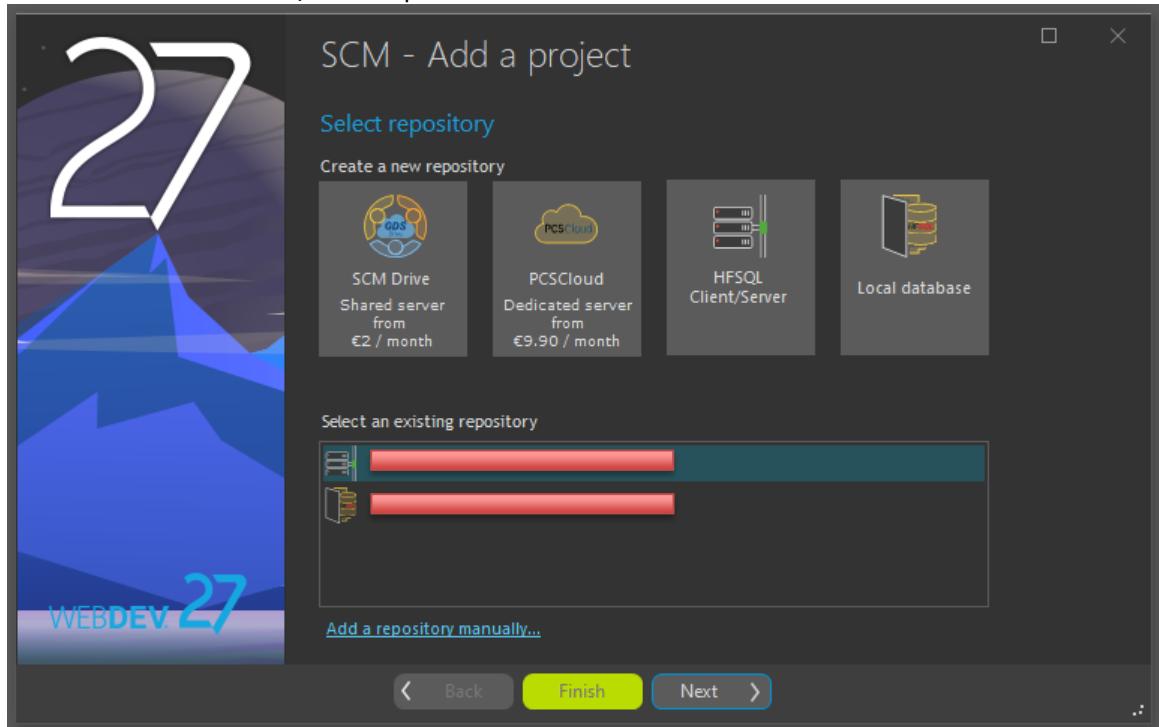
I will do this with WEBDEV's SCM (Source Code Manager). The SCM uses a repository that contains all the sources of my project: procedures, classes, windows, pages, reports, components, analyses, etc. The SCM allows me to manage the history of modifications and versions and allows me to work remotely as my SCM repository will be saved on a cloud server in HFSQL Client/Server mode. This will make backup and restore, simple and easy.

### Connecting to the SCM

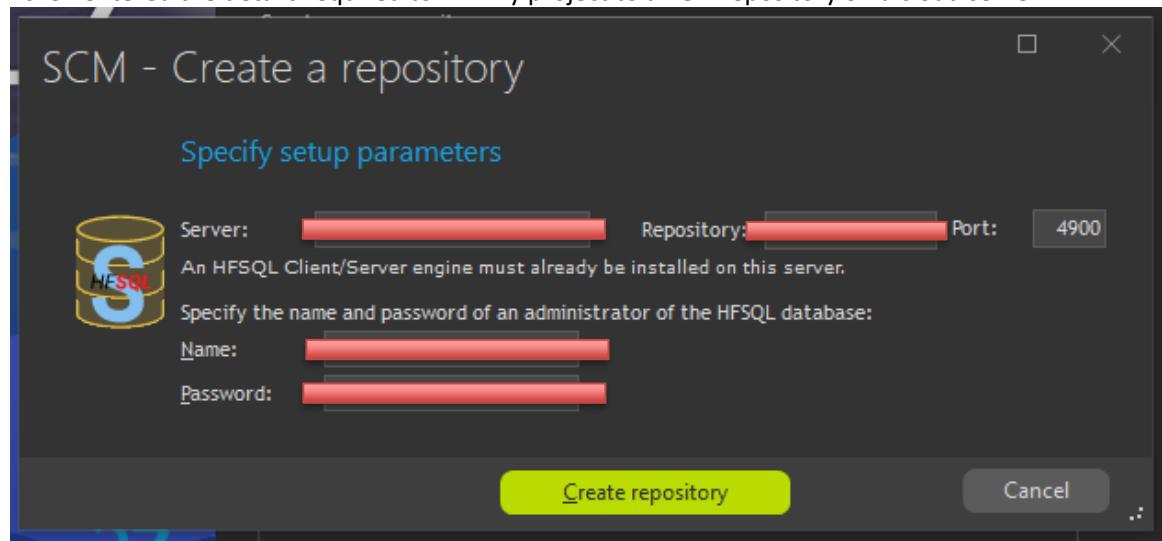
1. In the project SCM tab is the Add Project to SCM option.



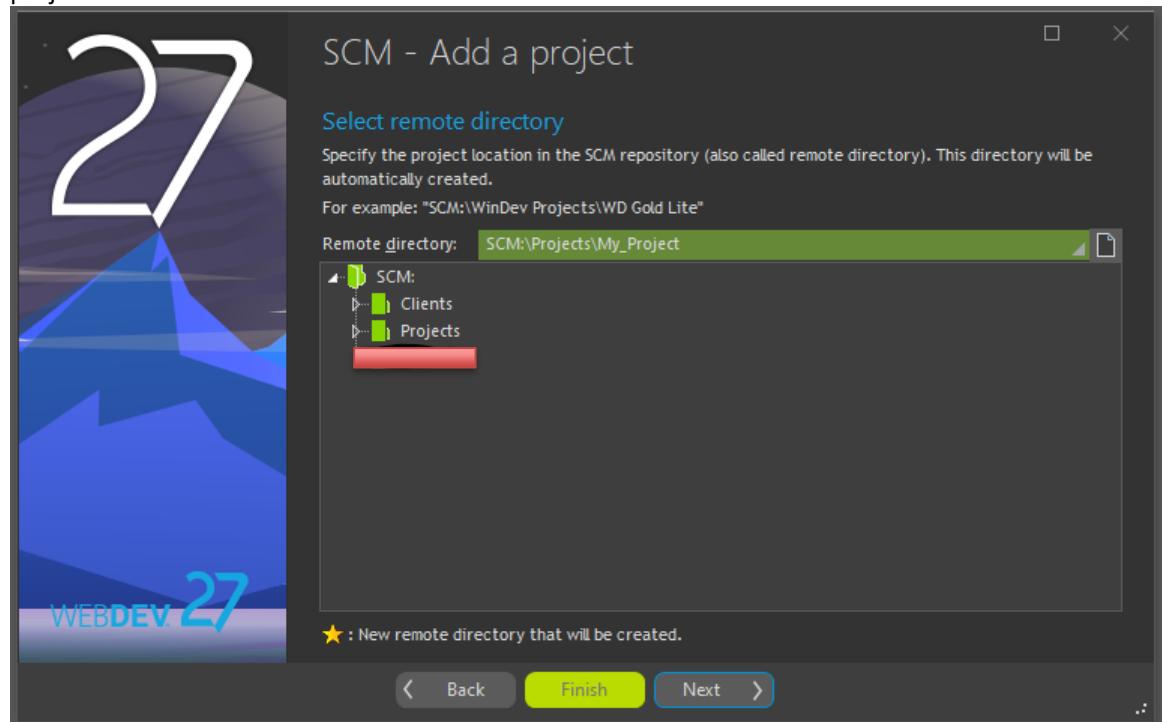
2. This option opens the repository window. To add my project to the remote repository, I selected the HFSQL Client/Server option.



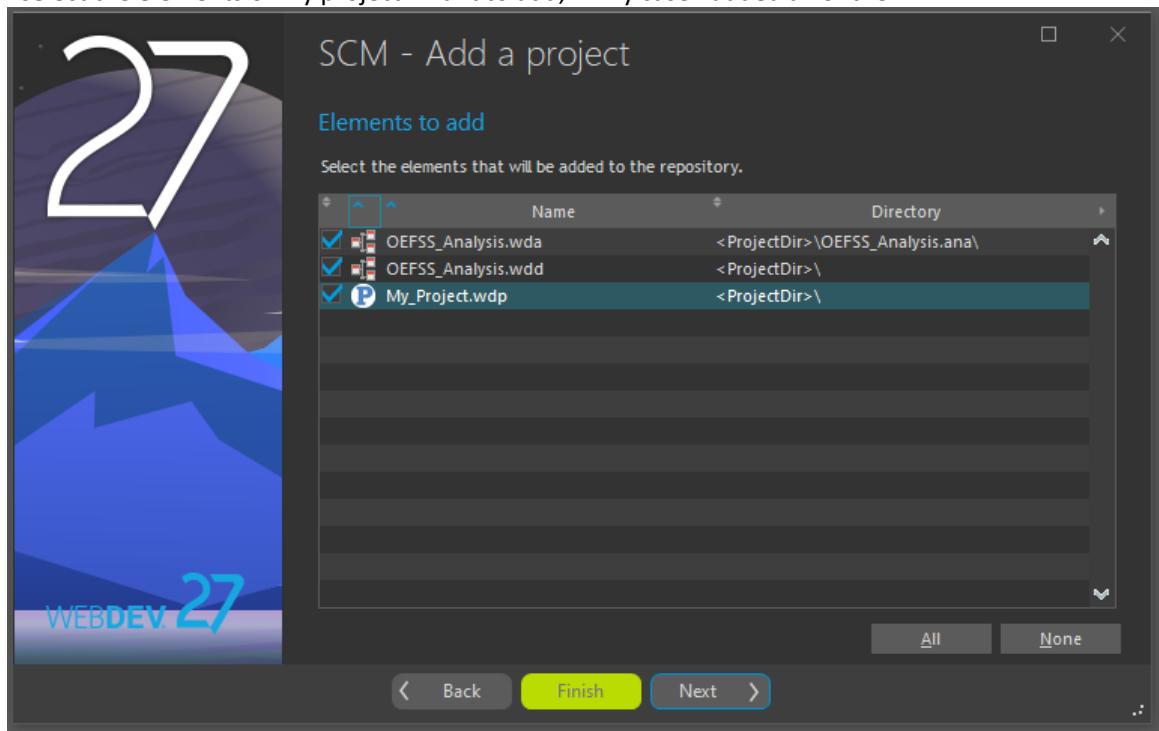
3. I then entered the details required to link my project to a new repository on a cloud server.



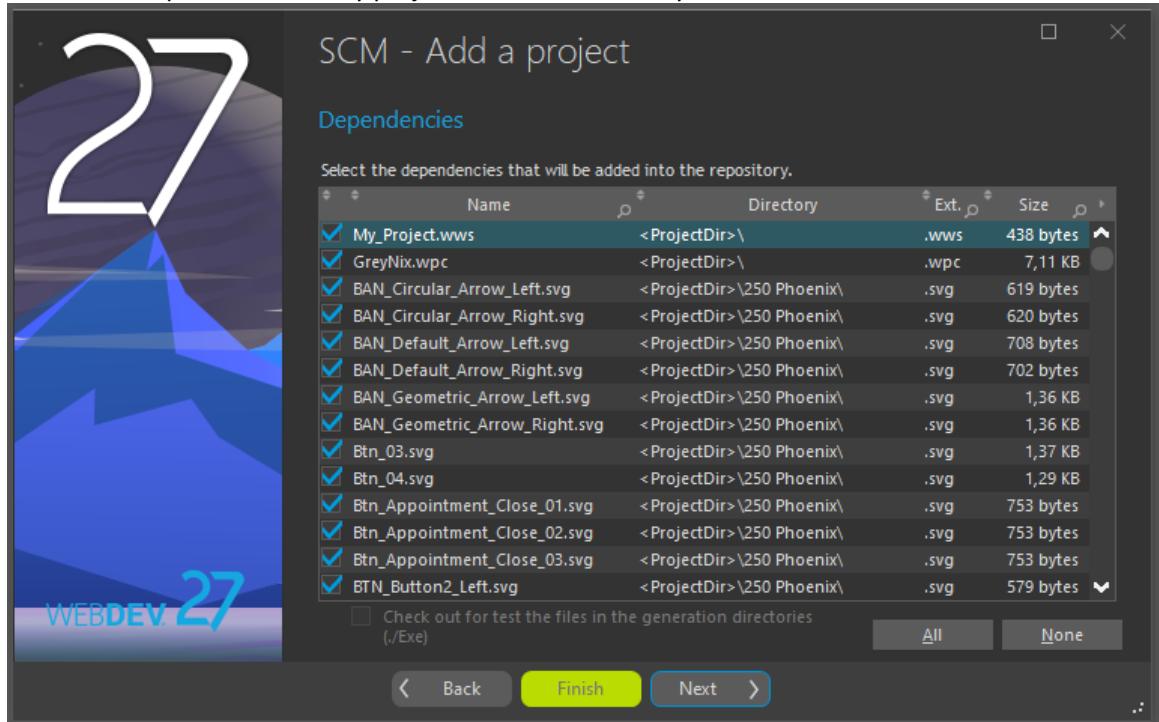
4. After connecting to the repository, I can add folders and subfolders where I can store multiple projects.



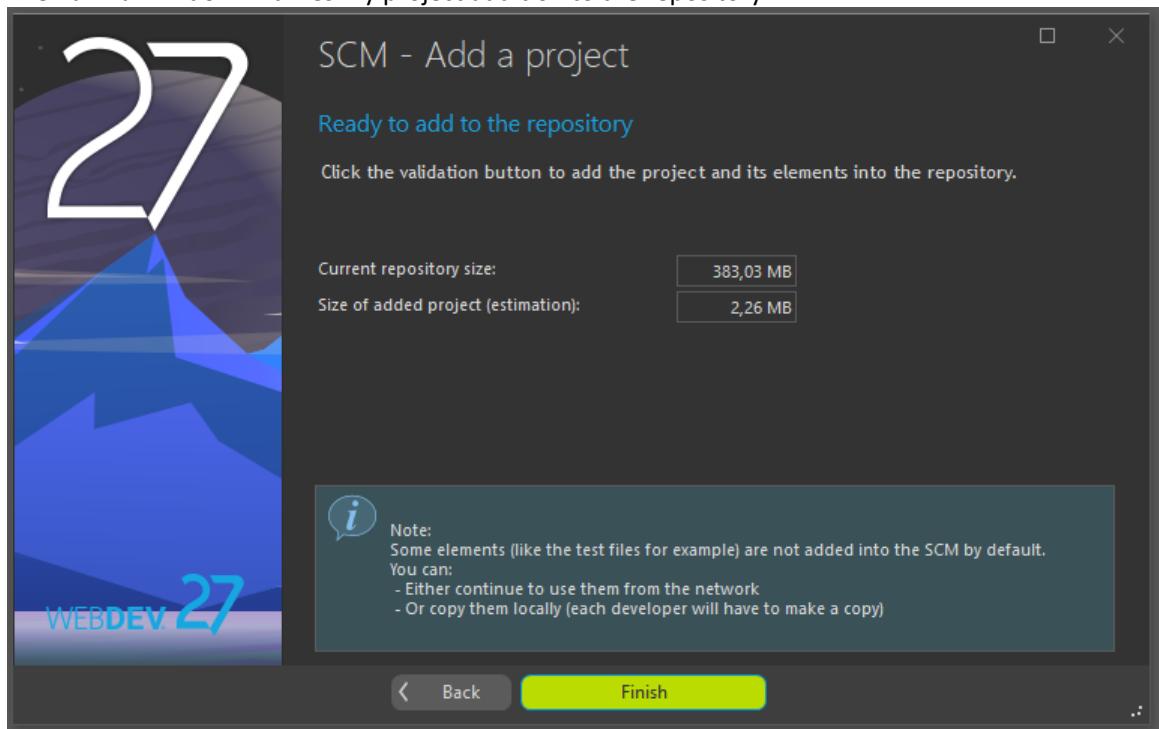
5. I select the elements of my project I want to add, in my case I added all of them.



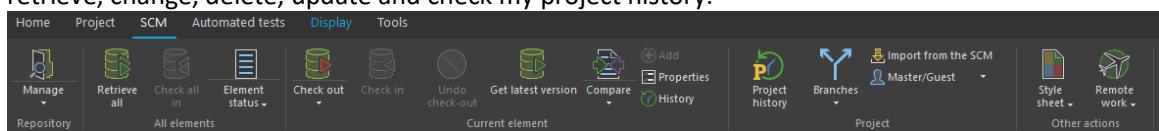
6. I select the dependencies of my project I want to add, in my case I added all of them.



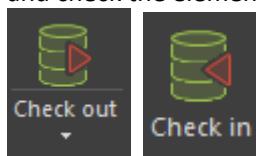
7. Then a final window finalizes my project addition to the repository.



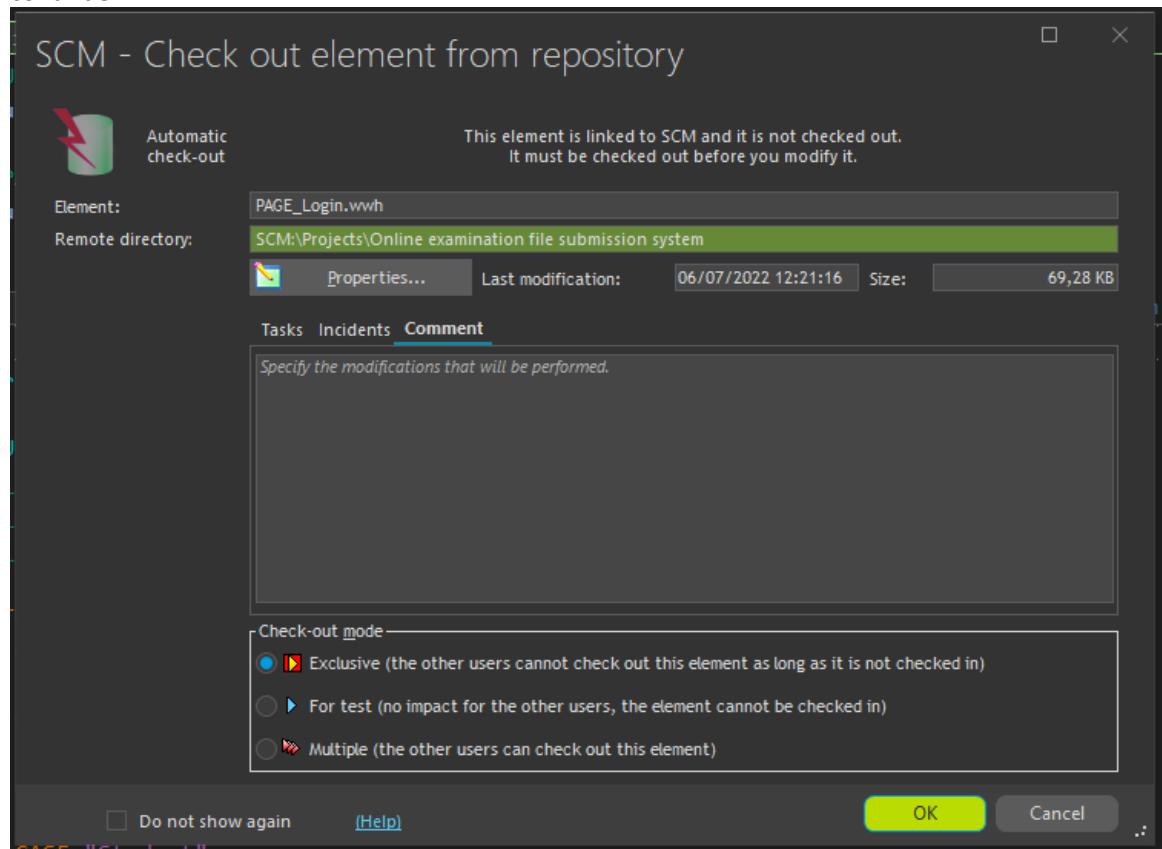
8. After my project was successfully added, the following toolbar appears that allows me to retrieve, change, delete, update and check my project history.



9. The Check out option allows me to retrieve, for example, the Login page, change some code and check the element back in to the repository with the Check in option.



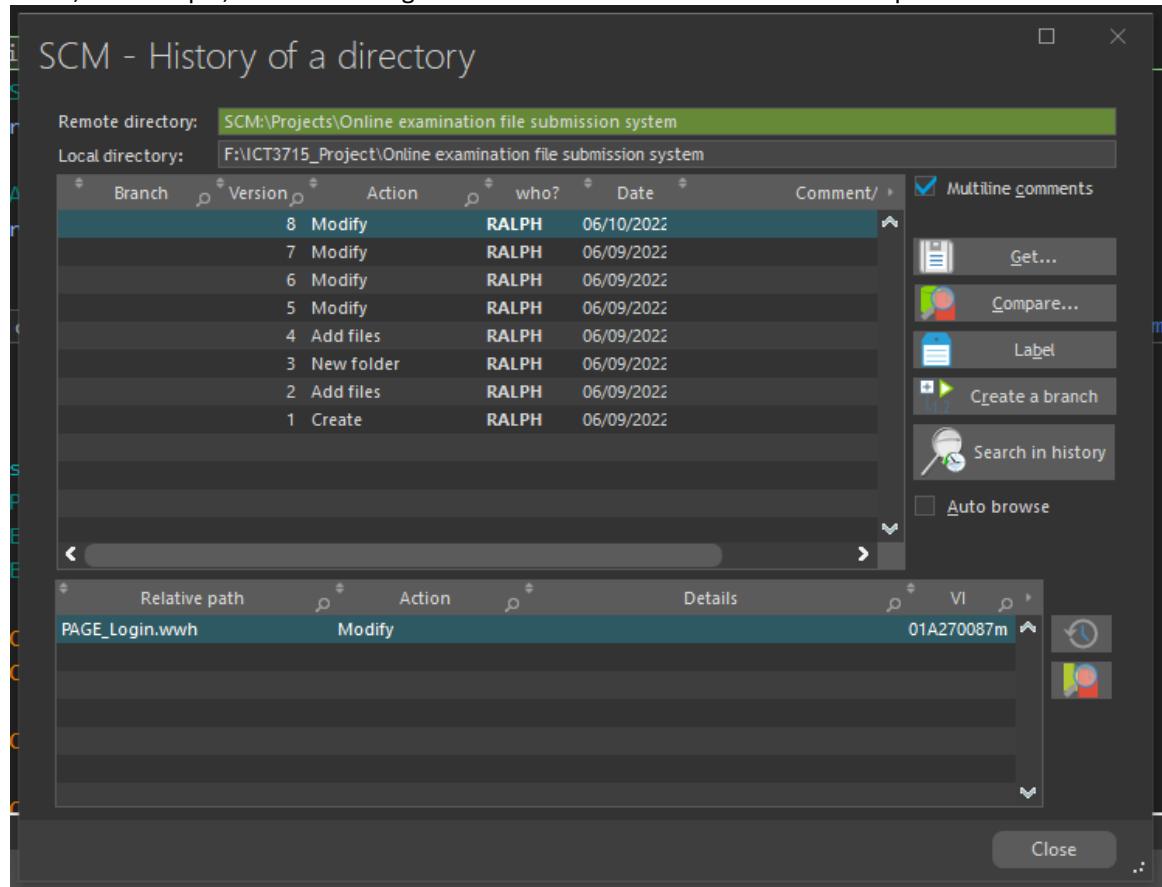
10. The Check out window allows for multiple options of check out, I keep the default value and continue.



11. To view my project history, I can select the Project History option.



12. This option allows me to view, compare, and retrieve previous versions and changes made for when, for example, I have made a grave mistake and want to revert back to a previous save.



The SCM allows me to handle my project remotely from anywhere and have access to previous versions and element modifications that allows for easy restore. It's basically an auto project backup system that also keeps my project safe and ready for me to restore anything, anytime.

## Assignment 1

### Section A [4]

#### Programming Languages (2)

##### **WLanguage**

A powerful 5th Generation language that will allow me to meet all the requirements stipulated by the project.

The project will be built using **WLanguage** in the **WEBDEV** development environment.

(Special permission was granted by Module Leader, Ina Scholtz. Please see email at end of document.)

#### Database (2)

##### **HFSQ**

WEBDEV's official database.

(Special permission was granted by Module Leader, Ina Scholtz. Please see email at end of document.)

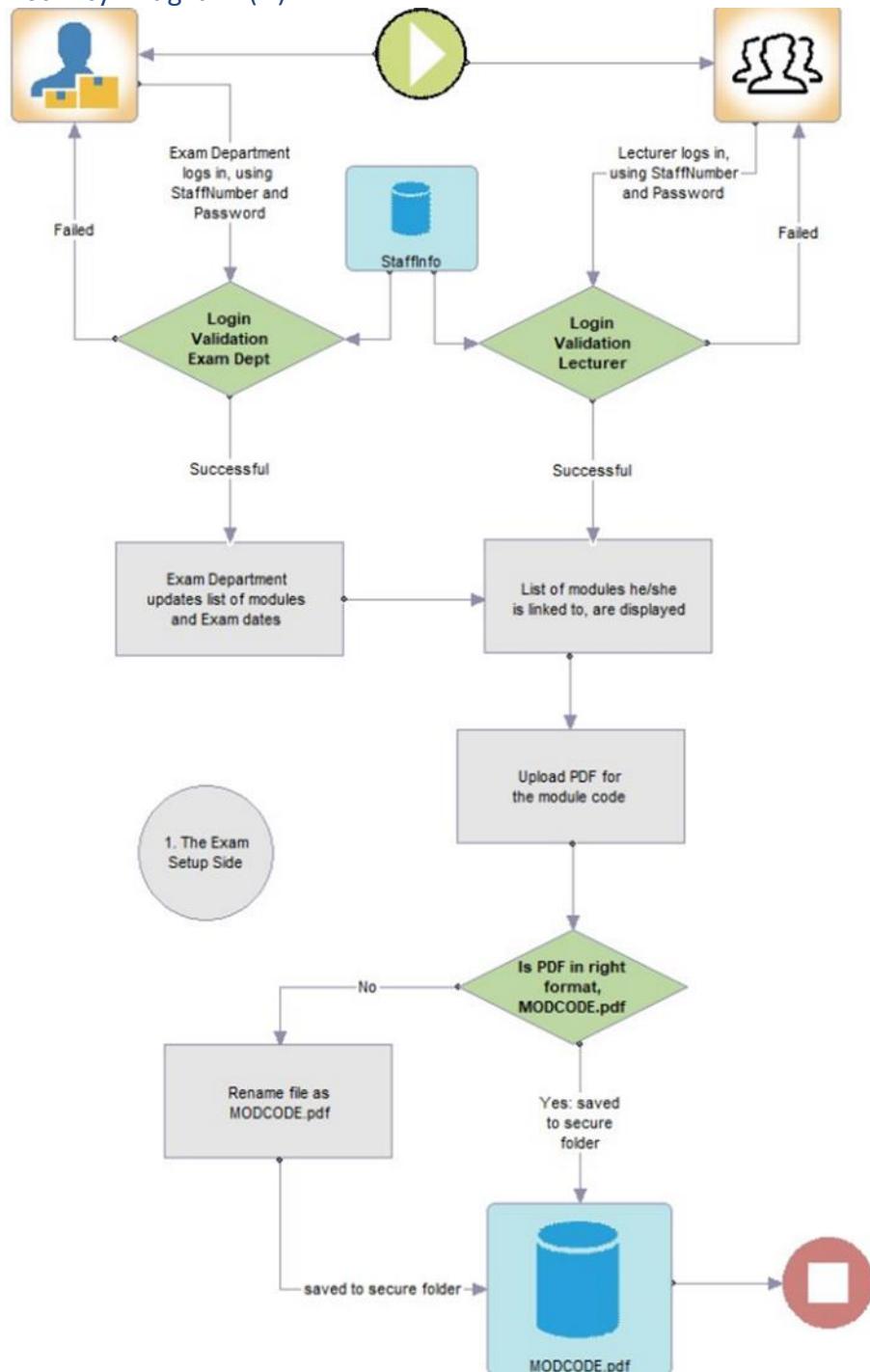
### Section B [10]

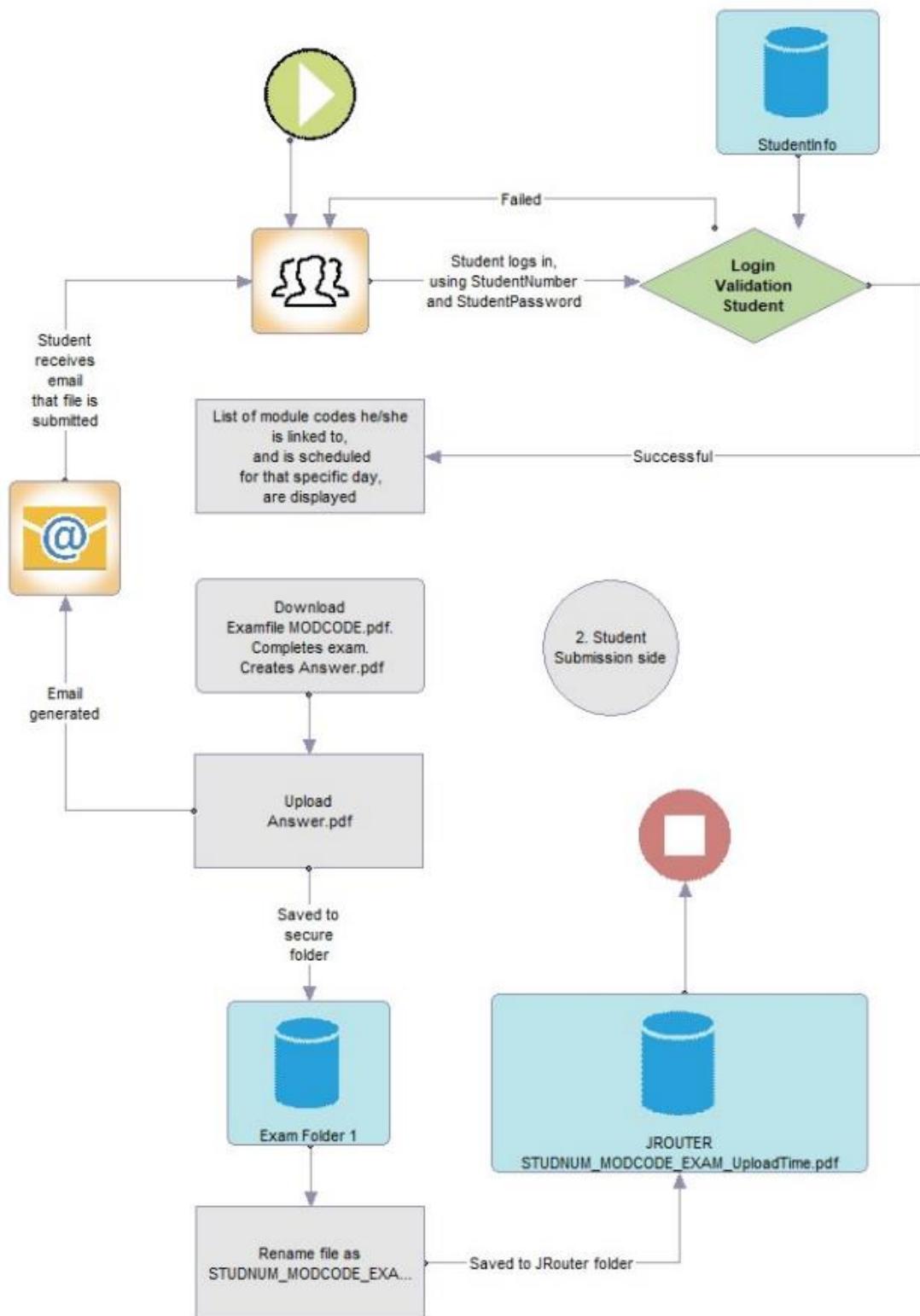
#### Cleaning the data

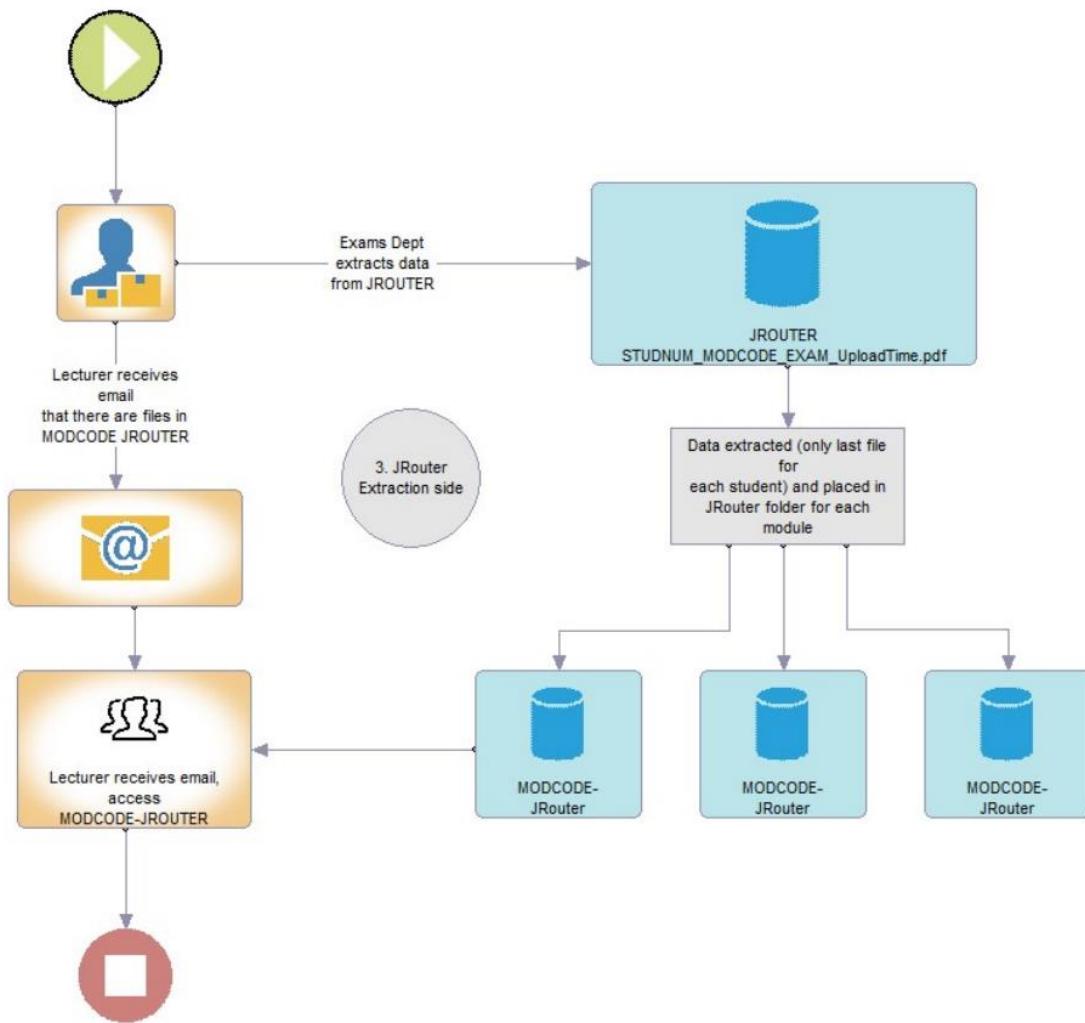
(Cleaned and attached .csv file to the end of my assignment as PDF)

## Section C [10]

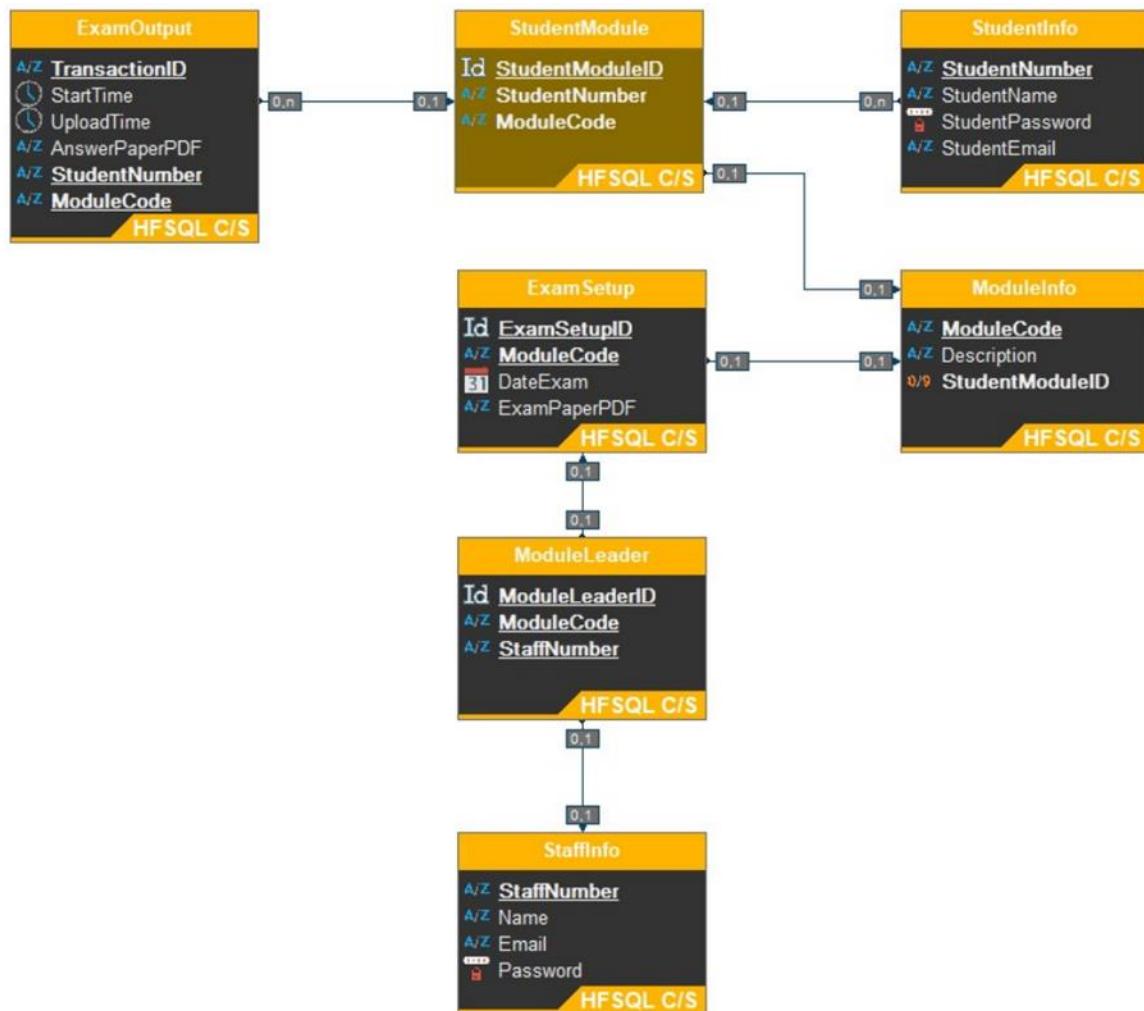
### Activity Diagram (7)







### ERD Diagram (3)



## Section Backup and Recovery for the Database and Programming code [6]

### Backup and Recovery Software for the Database (3)

- WEBDEV HFSQL Control Centre, allows for easy Database backup and recovery on the custom server in HFSQL Client/Server mode that I plan on using to also store my project which will also allow me to work on my project remotely.
- Make a separate backup of the Database and save it in OneDrive. WEBDEV makes it possible to restore an entire Database with ease this way.

### Backup and Recovery process for the Programming code and your Portfolio (assignments) (3)

- WEBDEV's SCM (Source Code Manager). The SCM uses a repository that contains all the sources of my project: procedures, classes, windows, pages, reports, components, analyses, etc. The SCM allows me to manage the history of modifications and versions and allows me to work remotely as my SCM

repository will be saved on a custom server in HFSQL Client/Server mode. This will make backup and restore, simple and easy.

- Make a separate backup of the Project and save it in OneDrive. WEBDEV makes it possible to restore an entire Project with ease this way.

## Module Leader Approval Request

**From:** Ralph Botes <[65614968@mylife.unisa.ac.za](mailto:65614968@mylife.unisa.ac.za)>

**Sent:** Thursday, April 28, 2022 1:18 PM

**To:** Scholtz, Ina <[scholid@unisa.ac.za](mailto:scholid@unisa.ac.za)>

**Subject:** 65614968 - Project, different language request

Good day,

I recently had a conversation with my employer, Ben Dell from Riebens Computers Pty Ltd, with regards to my task under ICT3715, who employed me as a Junior Software Developer 2 months ago.

Today he came back to me and asked me to apply at Unisa for using a different programming language and development environment to develop my year end project.

I would like to request to do so as I am currently employed and using this development environment 8 hours a day. This project is a perfect opportunity to utilize my work development environment and skills.

Software: PCSoft WEBDEV

Language: WLanguage (very similar in structure to C# and Java)

More: <https://windevsa.co.za/>

Brochure: <https://www.pcsoft-windev-webdev.com/webdev-us.pdf>

This is one of my development environments at my job, and I will also be teaching other developers this and I would love your permission to focus solely on this.

All requirements for the tasks as detailed in my curriculum can be met by using this alternative environment.

If you require any other info to aid your decision, please let me know.

Kind regards

Ralph Botes

## Approval

Re: 65614968 - Project, different language request



Scholtz, Ina  
To Ralph Botes

Good day Ralph,

Thank you for your e-mail.

Welcome to the Module ICT3715.

I had a look at the Brochure and it looks like a very interesting tool to use to design and develop your software system.

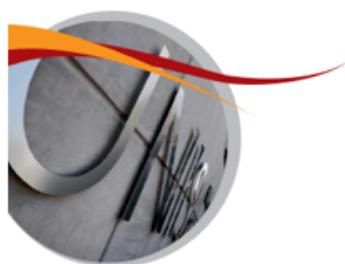
I thus approve that you may use the tool as long as you can accomplish all the requirement set out for the software system.

Please make sure that you include this e-mail in your Portfolio as proof that you are may use this tool.

Good luck with your system.

Have a wonderful day.

Greetings



**Mrs. ID Scholtz**  
*M Tech BIS*  
Senior Lecturer  
Computer Science Department  
Tel: 011 670 9179  
E-mail: scholid@unisa.ac.za



<http://www.unisa.ac.za>

Unisa Coronavirus (COVID-19) update: [www.unisa.ac.za/coronavirus](http://www.unisa.ac.za/coronavirus)