

# **BACHELOR OF INFORMATION TECHNOLOGY**

## **GRADUATE DIPLOMA IN INFORMATION SYSTEMS**

**SDV602 SOFTWARE DEVELOPMENT,  
2ND LANGUAGE**

**SEMESTER 2, 2022**

### **COURSE OUTLINE AND ASSESSMENT DETAILS**

Tutor: Todd Cochrane Phone +64 (3) 5463645

Office: G303

E-mail: Todd.Cochrane@nmit.ac.nz

## **DIGITAL TECHNOLOGIES**

**G-Block, Nile Street, Nelson**

*Whāia te iti kahurangi ki te tuohu koe me he maunga teitei*  
(Seek the treasure you value most dearly: if you bow your head, let it be to a lofty mountain)

## Class Times

Day	Time	Room
Monday	09:00 – 11:00	M307
Friday	14:00 – 16:00	A215

Please note class times or rooms may change due to circumstances outside of Digital Technologies control. Please regularly refer to your timetable on your MyNMIT and Moodle posts from the Digital Technologies Team.

## WELCOME

*Nau mai, haere mai ki te akoranga hangarau  
matihiko (Welcome to this Digital Technology paper)*

## COURSE PURPOSE

To broaden the students' software development horizon by experiencing a new programming language and environment. By using a language, possibly from a different vendor and/or is aimed at a different hardware platform or environment the students will gain valuable and marketable expertise. Building on the prerequisite course(s), students will apply the learnt analysis and design methodologies to the new programming environment, and if necessary adapt them to suit the characteristics of the chosen programming language.

## LEARNING OUTCOMES

Outcome	Guide
1 Examine and show understanding of a new programming language and identify its purpose and characteristics.	a The heritage and philosophy of the programming language is explored b The platform(s) for developing and running software applications are analysed and understood c The characteristics, strengths and weaknesses of the new programming environment are understood.
2. Demonstrate independence in the investigation and effective application of language syntax features.	a The syntax of the programming language is studied and applied to problems of intermediate difficulty. b The syntax differences between this language and an earlier one studied are understood, pitfalls and dangers of careless use are recognised. c The accompanying (class) libraries are explored and applied to a variety of smaller projects.
3. Effectively design and implement a software project in response to the requirements of a project brief. The software produced will be of an intermediate to advanced level.	a Apply the software design and development skills acquired in this and prerequisite courses to design, code and test an intermediate to advanced level project.

*Ka mate koe i te paoa; kāore, he kāuta*

(A job that offers a good return or promotion will make up for some difficulties or inconvenience)

### Prescribed Pre-Requisites Courses:

SDV501 or SDV503 Introduction to Software Development or equivalent skills and knowledge.

## STUDENT RESOURCES

### NMIT Ecampus

Ecampus is the main platform to deliver course resources and information from the Digital Technologies Team, please check this regularly.

You can access: Lecture notes, Digital Technologies announcements, Tutor forum posts, activities, articles, websites links, current assessment outline and marking rubrics, assessment submission dropbox, Turnitin, <http://ecampus.nmit.ac.nz/moodle/>

### Prescribed Textbook:

There is no prescribed textbook for this course. However, you are expected to use appropriate online documentation sources for example

### Software

You need a BYDO Laptop attached to the NMIT network. All other software is discussed in class. For example, we will also download and install other software development tools, for example Python and Visual Studio Code.

*Ehara taku toa i te toa takitahi, engari he toa takitini*  
(My strength is not that of a single warrior but that of many)

## PROJECT WORK (ASSESSMENT AND LEARNING - Approach)

This course takes a project centred approach to work. The course learning outcomes and aims with their associated outcomes are integrated into the activities and assessments you undertake during completion of your project work. You will meet **Learning Outcome 3**. "Effectively design and implement a software project in response to the requirements of a project brief. The software produced will be of an "intermediate to advanced level" by undertaking a project as specified in the project brief.

**Learning Outcome 1**. "Examine and show understanding of a new programming language and identify its purpose and characteristics."

Throughout the course as you develop your project your will gain knowledge and insight as follows:

- a) The heritage and philosophy of the programming language is explored
- b) The platform(s) for developing and running software applications are analysed and understood
- c) The characteristics, strength and weaknesses of the new programming environment are understood.

**Learning Outcome 2.** "Demonstrate independence in the investigation and effective application of language syntax features...". In your practices as a software developer you are expected to develop and demonstrate how you developed your capacity and understanding as follows:

- a The syntax of the programming language is studied and applied to problems of intermediate difficulty.
- b The syntax differences between this language and an earlier one studied are understood, pitfalls and dangers of careless use are recognised.
- c The accompanying (class) libraries are explored and applied to a variety of smaller projects.

Throughout the course you will be working on a single project. That is divided into three milestones. At each milestone you hand in and present all the work you undertook towards that milestone. For the first two milestones you receive feedback on the work you submit at the milestone. The final milestone introduces new material; however, you are also required to submit all work you have undertaken in the project. At milestone three, additional marks will be given as specified for the third and final milestone, however no further feedback will be given.

Marks given at milestones one and two will be final (summative) marks that contribute to your overall mark for the project.

The project brief with milestone marking schedules can be found at the end of this course outline.

In addition to working on your project you are expected to participate and attend the course. You are to collect evidence of your participation and practise as your work on course, the best way to do this is to set up an online storage location (GitHUB repo) that can be shared with your tutor to store material you produce as you study and work on SDV602.

Assessment Schedule (due dates may change)

Assessment	What	Due Date		Weighting
1. Milestone Reviews	One for each milestone	During the course, assessed at the end of the course		10%
2. Project	Project in three milestones as follows	See course schedule		90%
	Project Milestone 1		20%	
	Project Milestone 2		25%	
	Project Milestone 3		45%	
		<b>Total</b>		100%

Assignments will only be accepted if handed in on or before the due date unless there are special circumstances that are discussed with the course coordinator prior to the due date.

### Requirements for Successful Course Completion:

- Mark of 40% or more in every summative assessment
- Gain a course result of C (50%) or higher

We will discuss together our rules for how the class will operate but there are three things that are vital if you wish to succeed on this course:

- Attendance
- Participation
- Working on your course throughout the semester, not just at the end.

*E kore te matua e rawe ki te moana takai ai, engari anō a uta*  
(Don't wait until the last minute to do something, be prepared)

## COURSE CREDITS, LEVEL AND TIME BUDGET

NMIT Credits	15
Level	6
EFTS	0.125
Teaching hours	60
Workplace learning hours	0
Total learner managed hours	90
Total hours of student learning	150

	Weeks	Hours	Total
<b>Teaching Hours</b> This includes scheduled classes or tutor support time, and/or guided online time. Teaching hours include scheduled times when tutor support is available. Teaching hours include assessment and assessment feedback time, time scheduled for lab assessments, directed time in the library and on projects with tutor support, participation in online activities, workshops, laboratories, supervised field trips, and other tutor-directed study activities.	15	4	60
<b>Self-directed Learning</b> Self-directed learning hours are hours outside of classroom time or the direct supervision of the tutor that the average student is expected to spend in order to complete the course successfully. This may include distance learning, reading of course materials, study groups, preparation, homework, self-paced learning in computer labs, research etc.	15	6	90
<b>TOTAL HOURS OF STUDENT LEARNING</b>			<b>150</b>

*E kore nei e taea i te rā ki te waru*  
(It cannot be achieved in a hurry)

## PROGRAMME REGULATIONS AND POLICIES

All Programme Regulations and NMIT Policies are accessible from a link on the Information Technology Home Moodle page under the tab [Programme Regulations, Course Descriptors and Policies](#).

### THE IMPORTANCE OF UNDERSTANDING PROGRAMME REGULATIONS

Please familiarise yourself with your Programme Regulations (qualification you are studying), these regulations govern all aspects of the programme and the actions Digital Technologies can take to support your learner journey. Please note that Programme Regulations supersede course outlines where applicable.

Pay special attention to:

- Graduate Outcomes
- Core transferrable Skills
- Completion Requirements for your programme
- Course Results
- Attendance Requirements
- Special Assessment Circumstances

### COURSE RESULTS

The overall percentage mark for the course is calculated by adding the weighted results for all summative assessments. To derive the course result the overall percentage mark is converted into a grade using Course Result Key AC-NMIT-06. Refer to the Course Descriptor for details of the requirements for the successful completion of the course.

#### AC-NMIT-06:

RESULT	MARK RANGE (%)	DESCRIPTION
A+	85 - 100	Pass with Distinction
A	80 - 84	
A-	75 - 79	
B+	70 - 74	Pass with Merit
B	65 - 69	
B-	60 - 64	
C+	55 - 59	Pass
C	50 - 54	
D	40 - 49	Fail grades
E	0 - 39	

*Ahakoā he aha te kai hei rite mō te huahua, e kore e rite*

(An effort may be made to copy or approximate the taste but it is never the same as the original)

## COURSE SCHEDULE

This is indicative only. Topics WILL change depending on technical availability, new developments and student interest. Note this is a LIVE document, check it once a week.

WEEK	Beginning	TOPIC	PRACTICE AND TUTORIAL
<b>Week 1</b>	18 July	Course outline with projects How to compare programming languages Typing : strong vs weak, dynamic vs static.	Describe your App as in Project Brief Work on Linked in tutorial (1) <a href="https://www.linkedin.com/learning/python-for-non-programmers/python-from-zero?autoplay=true&amp;resume=false&amp;u=76059146">https://www.linkedin.com/learning/python-for-non-programmers/python-from-zero?autoplay=true&amp;resume=false&amp;u=76059146</a> <a href="https://replit.com/languages/python3">https://replit.com/languages/python3</a> Alternative <a href="https://www.linkedin.com/learning/python-essential-training-14898805/getting-started-with-python?autoplay=true&amp;u=76059146">https://www.linkedin.com/learning/python-essential-training-14898805/getting-started-with-python?autoplay=true&amp;u=76059146</a>
<b>Week 2</b>	25 July	Input Flow control	What is good documentation? Guido's guides App structure ... Work on Linked in tutorial (2) <a href="https://www.linkedin.com/learning/advanced-python/welcome?autoplay=true&amp;u=76059146">https://www.linkedin.com/learning/advanced-python/welcome?autoplay=true&amp;u=76059146</a>
<b>Week 3</b>	1 August	Command arguments Procedures and functions GLOBAL Input, output	GUI libraries – recommend/choose for course. <a href="https://www.linkedin.com/learning/rapid-application-development-with-python/rapid-development-of-python-gui-apps-with-wxglade?autoplay=true&amp;u=76059146">https://www.linkedin.com/learning/rapid-application-development-with-python/rapid-development-of-python-gui-apps-with-wxglade?autoplay=true&amp;u=76059146</a>
<b>Week 4</b>	8 August	GUI, and Graphics: Charts Modules	About writing Milestone reviews
<b>Week 5</b>	15 August	Milestone One review and work - Hand in Milestone 1	
<b>Week 6</b>	22 August	Milestone Two, Setting the goals	
<b>Week 7</b>	29 August	List Comprehensions	<a href="https://docs.python.org/3/tutorial/datastructures.html">https://docs.python.org/3/tutorial/datastructures.html</a>
<b>Week 8</b>	5 September	Map, Reduce Filter	
<b>Week 9</b>	12 September	Local Storage, database	to device file storage, To local database
<b>Week 10</b>	19 September	Milestone Two review and work - Hand in Milestone 2	
	26 September	Term Break	
	3 October	Term Break	
<b>Week 11</b>	10 October	Milestone Three, setting the goals	Web Services / httpRequest
<b>Week 12</b>	17 October	Network services sockets and pipes.	Connecting to a networked db service
<b>Week 13</b>	24 October	Chat systems	No class on Monday (Labour day)
<b>Week 14</b>	31 October	Project refinement	Project refinement
<b>Week 15</b>	7 November	Assessment week #1	
<b>Week 16</b>	14 November	Assessment week #2	Hand in Milestone three Essay, Journal

**No classes on the following date:**

- Labour Day



*He iti hau marangai e tū te pāhokahoka*  
(A little storm and then a rainbow appears)

**ASSESSMENT SHEETS – Are provided in the following pages.**

# SDV602 Assessment

## Project Milestone Reflective Reviews (3)

<b>Assessment:</b>	Assignment 1 Reviews.
<b>Course:</b>	SDV602 SOFTWARE DEVELOPMENT, 2ND LANGUAGE
<b>Given:</b>	At time specified in the Course Outline
<b>Due</b>	At time specified in the Course Outline
<b>Weighting:</b>	10% of the course grade
<b>Tutor:</b>	Todd Cochrane
<b>Moderator:</b>	_____

### LEARNING OUTCOMES

This assignment focuses on the following course learning outcomes

1	Examine and show understanding of a new programming language and identify its purpose and characteristics.
2	Demonstrate independence in the investigation and effective application of language syntax features.
3	Effectively design and implement a software project in response to the requirements of a project brief. The software produced will be of an intermediate to advanced level.

### CONDITIONS

- For this assessment you are to write your own writing in a journal. Hand in material as specified on your class website, in the course dropbox.
- APA 7TH in-text citations and references are required for any sources.
- The assessment may be submitted to Turnitin.
- This assessment is to be done individually.
- All work submitted for assessment must be entirely your own.

### REFLECTIVE REVIEW BRIEF

Work on the course is based on a single project through which you demonstrate your understanding and capacity to undertake web development as covered in the course. As reflective review is a record of your reflection on your practice as you learn about another programming system. Hand in **One review** with each milestone.

### Marking of the Reviews

They must reflect what you have learnt and developed as you became familiar with the new programming system. An overall grade will be awarded according to the following rubric. The marker will ask "have all of the sections been covered? Are the entries reflecting on the practices learnt and covered?". **A score out of 10** will be awarded as the grade for this assessment. Marks are deducted for not following APA7th citation system.

The best approach in this course is to use your reviews to prepare for writing a short 1500-word essay that covers part of learning outcome one," *Examine and show understanding of a new programming language and identify its purpose and characteristics.* ", by writing reviews that compare the new

programming language with the one you have already studied. For each programming system, the new and the one you have experienced, write to cover the following.

- a) The heritage and philosophy of the programming language is explored
- b) The platform(s) for developing and running software applications are analysed and understood
- c) The characteristics, strength and weaknesses of the new programming environment are understood.

### Marking rubric

0	1-3	4-5	6-7	8-10
No submission	Submission is a direct statement of practice undertaken or covered in the milestone	Submission is a direct statement of practice undertaken AND covered in the milestone	Submission is a direct statement of practice undertaken and covered in the milestone, with insights on your practice that show your thoughts on how you do the work.	Submission is a direct statement of practice undertaken and covered in the milestone, with insights on your practice that show your thoughts on how you do the work. and reflection on how that work can be applied in future, your "rationale".
Total				

# SDV602 Portfolio - project

**Assessment:** Portfolio - project.

**Course:** SDV602 SOFTWARE DEVELOPMENT, 2ND LANGUAGE

**Given:** At time specified in the Course Outline

**Due** At time specified in the Course Outline

**Weighting:** 90% of the course grade

**Tutor:** Todd Cochrane

**Moderator:** \_\_\_\_\_

## LEARNING OUTCOMES

This assignment focuses on the following course learning outcomes:

1	Examine and show understanding of a new programming language and identify its purpose and characteristics.
2	Demonstrate independence in the investigation and effective application of language syntax features.
3	Effectively design and implement a software project in response to the requirements of a project brief. The software produced will be of an intermediate to advanced level.

## CONDITIONS

- For this assessment you are to write your own website solutions, and reports. Hand in material as specified for each milestone as described below and on your class website, in the course drop box.
- APA 7TH in-text citations and references are required for any sources.
- The assessment may be submitted to Turnitin.
- This assessment is to be done individually.
- All work submitted for assessment must be entirely your own.

## PROTFOLIO PROJECT BRIEF

Work on the course is based on a single project through which you demonstrate your understanding of the programming language and capacity to undertake analysis and development in that programming system, as covered in the course.

You are to develop an application that provides live and interactive graphical displays, that include charts, to enable presentation of different graphical views of data sources for a business (or scientific) analyst. Each live interactive graphical display allows exploration of data from a data source – these are Data Explorer Screens (DES), see figure 1. At each DES the app accepts text input and mouse clicks, and displays a brief description of the situation that relates to the information being displayed in an adjustable chart (you are to use the code libraries discussed in class). The application provides data explorer screens. Each screen includes a chart and accepts text-based input, as well as pointer (i.e. mouse based) input. Each DES presents views of data in a chart displayed as well as a brief summary of data fields.

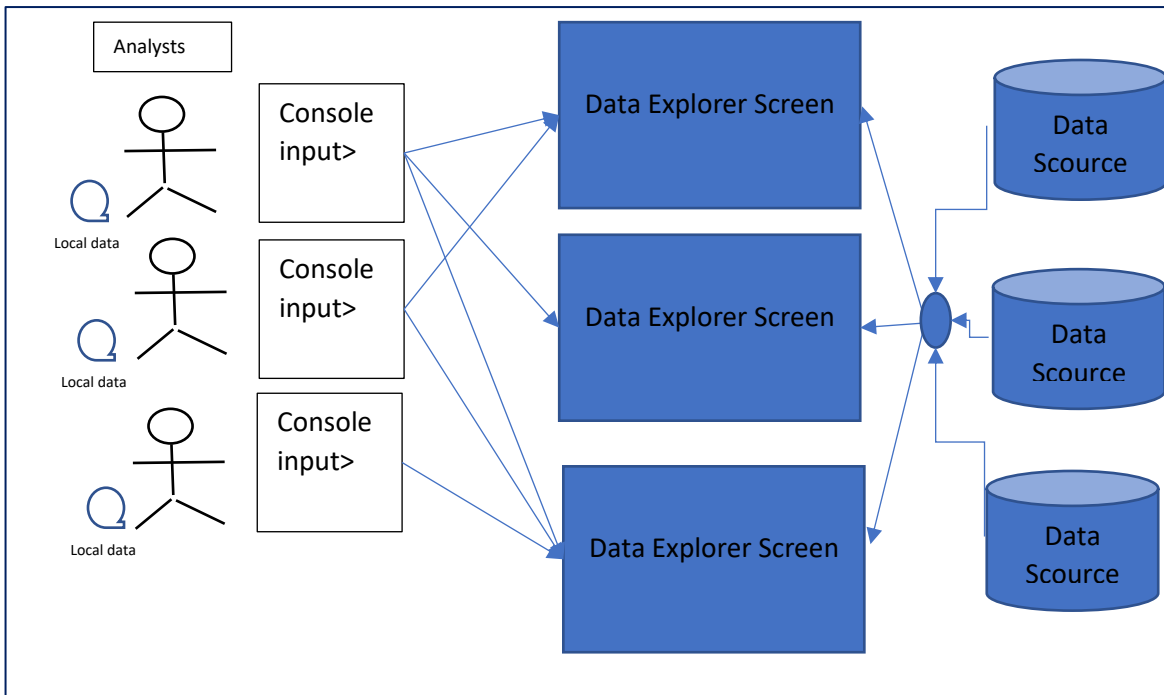


Figure 1 Application sketch, see figure 2 for DES details. Each analyst starts with a console input.

The application is to provide console text-based input to control and display at least three different DESs, see figure 2. Each DES presents charts (each based on their individual data source) and a summary or comparative analytical information. The final system is to provide a chat system for analysts viewing the current DES, that is accessed using a remote http-based API through a remote data service. The final system is to retrieve data from a shared remote data source. The screens are to include a number of options for views of the data at each display.

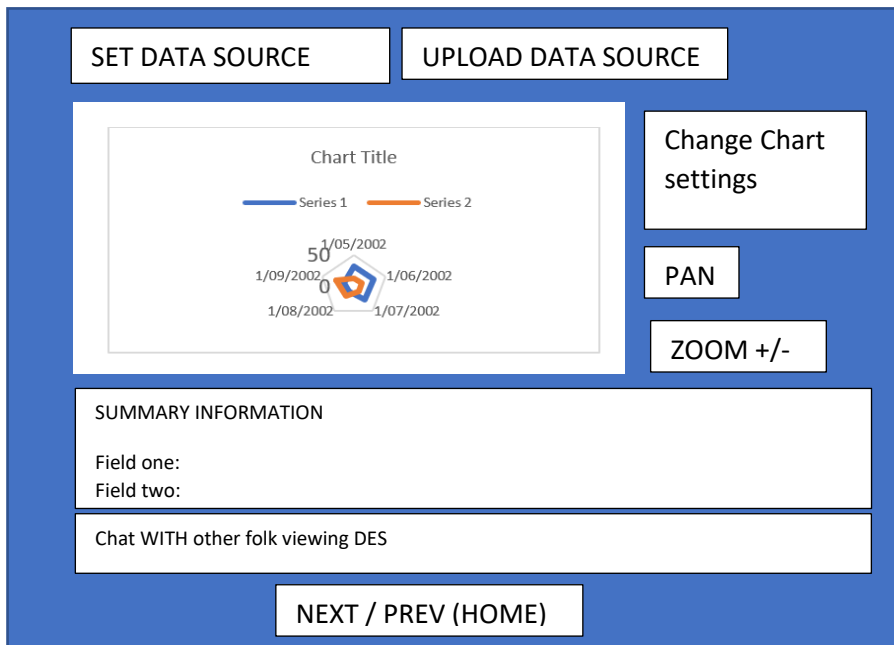


Figure 2 Data Explorer Screen - indicative sketch.

The FINAL application is to allow more than one analyst to view the same DES at the same time and upload and retrieve data on a shared remote server. Two or more analysts can communicate with each other using a text chat system (that is to be included as part of each DES) when they are viewing the same screen. The final application is to include user login and logout.

**Technical overview**

You are required to create a set of storyboards that make up your implementation of the supporting software. The system you develop is to be an implementation of your design. A set of test scripts should be developed as part of your system developments. You will be influenced by the architecture determined by the software development platform, however a suitably identified architecture beyond this is preferred in the development of your application.

**Required from you**

You are to write a one-page description of your application, its purpose, and a brief overview of the parts of the application. Send that description to your tutor by email, the day before the first class of the second week.

# SDV602 PROJECT MILESTONE ONE DETAILS: MARKING SHEET

DUE DATE: as specified in the course outline.

TOTAL MARKS: 60 marks (worth 20% of the course)

## Brief - In Summary

Create a storyboard depicting ALL the screens and interaction in the application.  
Include a description of your app and narrative descriptions of the interactions depicted on each storyboard.

Create a first prototype of your app that implements prototypes for three interactive DESs for the application (without user login and logout).

## How to submit your project work.

Hand in all source code and any documentation in one folder that you have uploaded to a GIT repository. Using your course website put the URL of your GIT repository into the dropbox provided for this milestone. Make sure you name your repository OR a folder in your repository SDV602 Milestone 1 <your student name>.

## Work detail and Marking schedule

The application needs to have three DESs and navigation between all.

### 1. Design and Documentation (18 marks)

You are to hand in:

A brief description of the purpose of your application. (6 marks)

0	1-2	3-4	5-6
No description	Brief description without details for screens	Screens detailed	Motivation for the application (business or scientific)

A set of storyboards depicting the working of the WHOLE application. Each storyboard is to include a table that details ALL interactions, inputs and outputs. (12 marks)

Marks are deducted for missing aspects.	Storyboard (1) The storyboard depicts a screen	Details (2) All actions and displays are labelled and described, must include how to move from one screen to another	
Login			/3
One			/3
Two			/3
Three			/3
		Total	/12

### 2. Implement **tests scripts** that runs three DESs (36 marks)

Twelve marks per screen. Each DES:

- accepts correct input and displays correct information (12 marks)
- displays a place holder for the chart (12 marks)
- provides navigation to other DESs, must include a top command interface to display them all (12 marks)

The screens must run without presenting errors.

For each DES additive marks	Correct input, (2) Display (2)	Placeholder (2) produces a Chart (2)	Navigation has a command console interface to run DESs , (2) Moves to a different screen (2)	Total
One				✓12
Two				✓12
Three				✓12
Total				✓36

3. *Coding practices (up to **6 marks** deducted)*

- Comments using the convention of the programming system (2 marks)  
Reason for mark:
- Code formatting according to established programming system conventions, (2 marks)  
Reason for mark:
- Naming (constants, variables, procedures, classes, modules) according to established programming system conventions (2 marks)  
Reason for mark:



## SDV602 PROJECT MILESTONE TWO DETAILS: MARKING SHEET

DUE DATE: as specified in the course outline.

TOTAL MARKS: 75 marks (worth 25% of the course)

### Brief - In Summary

Starting with the previous three DES implementations, complete local data reading, uploading and merging, that is immediately displayed for each DES. Build application architecture using suitable abstractions. (In Python, use Modules, and possibly Classes, think about "domain specific language style" architecture). Implement "local" upload.

### How to submit your project work.

Hand in all source code and any documentation in one folder that you have uploaded to a GIT repository. Using your course website put the URL of your GIT repository into the dropbox provided for this milestone. Make sure you name your repository OR a folder in your repository SDV602 Milestone 2 <your student name>.

### Work details and Marking schedule

*Implements the Application architecture* **72 marks**

- The application uses a Python module that supports code reuse for the three DESs screens. Each DES reuses the module (36 marks)

For each DES additive marks	Module is used by the DES (2)	The DES makes call to all domain methods or procedures in the Module (4)	The call to each domain method on the module produces a change in the DES (6)	Total
One				/12
Two				/12
Three				/12
Total				/36

- Data for DES is read from the local disk (Use **local data services**) (36 marks)

For each DES additive marks	DES displays a different graph based on the data source (1)	DES chart display can be adjusted, for example "Zoom in", "Zoom out", "Pan". (2)	DES includes an upload function, that uploads a suitable dataset for its graph display (4)	DES displays new data merged with existing data after upload (5)	Total
One					/12
Two					/12
Three					/12
Total					/36

*Is operating without errors* **3 marks**

**One mark deducted for each error.**

# SDV602 PROJECT MILESTONE THREE DETAILS: MARKING SHEET

DUE DATE: as specified in the course outline.

TOTAL MARKS: 135 marks (worth 45 % of the course)

We will take care to make sure you provide the expected outcomes. Please make sure you start this earlier enough to get feed in from your tutor/s.

## Brief - In Summary

Implement, multi-user aspects of your App. Compare Python with another programming language.

## Work details and Marking schedule

*The marking will look carefully at the way the system is implemented and check that the system handed in is operating without errors.*

### 1. Implement multi-user aspects of your app (30 marks)

Include:

- a. A log in facility. (12 marks) - divided into
    - i. Validate the login using a remote database system. (6 marks)
    - ii. Maintain and manage session status in an encapsulated manner. (6 marks)
  - b. Create a chat system, that displays the analysts viewing the same screen. (18 marks) divided into:
    - i. Works and is tested to completion (6 marks)
    - ii. Clear use of architecture and separation of concerns (6 marks)
    - iii. Clear App structure (6 marks)
2. Implement remote storage and retrieval of data sources.

Data for DES is read from a remote data store (Use **remote data services**) (36 marks)

For each DES additive marks	DES displays a different graph based on the remote data source (1)	DES includes an upload function, that uploads a suitable dataset to the remote data store for its graph display (4)	DES displays new data merged with existing data after upload, from the remote data store (5)	Total
				/12
One				/12
Two				/12
Three				/12
Total				/36

3. Write a short 1500-word **essay** that includes a tabular comparison between Python and a different programming language or system you have used previously (C# or JavaScript). **(69 marks)**

BEFORE YOU START ON THIS ESSAY CONFIRM THAT THE TUTOR ACCEPTS THE PROGRAMMING LANGUAGE AND SYSTEM YOU ARE PROPOSING TO COMPARE WITH PYTHON.

The essay is to present for EACH programming language or system:

- The heritage and philosophy of the programming language is explored (20 marks)  
Ten marks for each programming language
- The platform(s) for developing and running software applications for that language are analysed and described. (code libraries, IDEs) (20 marks)  
Ten marks for each programming language
- The characteristics, strengths and weaknesses of the programming environment are described. (20 marks)  
Ten marks for each programming language as follows

	0	1-4	5-7	8-10
a) The heritage and philosophy of the programming language is explored	Not covered	Heritage: the story of when, how and why the programming language was established. Without details or citations. Philosophy: Presents anecdotal perspective of the culture, and how that is maintained. Missing details	Heritage: the story of when, how and why the programming language was established. Without citations. Philosophy: Presents anecdotal perspective of the culture, and how that is maintained.	Heritage: the story of when, how and why the programming language was established with a brief biography of key players and their contribution – this may include quality committees. Philosophy: describes the paradigms applied in its development, and how that becomes a culture and continues to

				be maintained. Includes APA7 citations
b) The platform(s) for developing and running software applications for that language are analysed and described. (code libraries, IDEs)	Not covered	Misses parts of the following. Presents one platform and describes how it works. Describes how the code libraries are used.	Presents one platform, and describes how it works. Describes how the code libraries are used.	Presents a scope for the platforms, and describes them. Describes how the code libraries are used, with APA7 citations.
c) The characteristics, strengths and weaknesses of the programming environment are described.		Misses parts of the following. The programming language type system is described with examples. Relates advantages and disadvantages of the system overall.	The programming language type system is described with examples. Relates advantages and disadvantages of the system overall. Misses type system details and or advantages or disadvantages	The programming language type system is described with examples. Relates advantages and disadvantages of the system overall. Include APA7 citations

A comparison of each. 9 marks

0	3-4	5-6	7-9
No submission	Aspects are missing from these: Each programming system is compared in terms of a) Their heritage and philosophy b) Platforms and code libraries c) strength and weaknesses	Some minor aspects are missing but includes all of these: Each programming system is compared in terms of a) Their heritage and philosophy	Each programming system is compared in terms of a) Their heritage and philosophy b) Platforms and code libraries c) strength and weaknesses APA7 citations

		b) Platforms and code libraries c) strength and weaknesses	
--	--	---	--

Note APA 7TH in-text citations and references are required for any sources.

**How to submit your project work.**

Hand in all source code and any documentation in one folder that you have uploaded to a GIT repository. Using your course website put the URL of your GIT repository into the dropbox provided for this milestone. Make sure you name your repository OR a folder in your repository SDV602 Milestone3 <your student name>.