

# PRESENTATION REQUIREMENTS

Your presentation time is a **minimum of 15, maximum of 20 minutes, with 5 minutes for questions** . Your presentation has to include the following:

- The nature of the problem and the user requirements.
- The software methodology that you have used; how well/poorly it worked for your project and why.
- The team roles and the people assigned to each role. Any changes to roles should be clearly identified and explained.
- Progress/completion of the requirements, as well as testing, for each requirement. Scope changes and the reasons for these changes are highlighted and explained.
- Live demo that shows that product has been developed as per the documentation and meets the acceptance criteria.
- Lessons learned from the project, both on the team level and individual team member level.
- Discussion of when client was contacted for feedback during the project, the feedback received, as well as how the feedback was incorporated into the product.

**\*important\*** each slide should have dot points/essential info, and then whoever's talking goes into detail **without a script**

## **3821ICT – WIL Single Project**

**Improving the efficiency and effectiveness of large-scale extraction of electricity demand data published by AEMO as part of the 2022 Integrated System Plan (ISP).**

**Industry Partner:** The Centre for Applied Energy Economics and Policy Research (CAEPR)

**Client:** Nancy Spencer, Phillip Wild

# Team Roles

Naman - Project Manager and Quality Manager

- Organised the team and made sure everyone was working well together.
- Assured that the quality of the project was to a high degree.

Akshay - Business Analyst and Client Liaison

- Was the main form of communication between the client and the rest of the team.
- Organised meetings with the client.

Nathan - Developer and Tester

- Lead developer who worked on combining different aspects of the project
- Responsible for testing individual segments and the program as a whole.

Josh - Developer and Tester

- Developer who programmed multiple different segments of the project
- Responsible for testing individual segments and providing feedback.

# The Project

The project required the development of a system that could load data from the AEMO into a database and then be processed to show helpful information about the data.

- The project specified that the data should be loaded and displayed efficiently.
- Should minimise the hands-on activity required by the CAEEPR staff.
- Automate the process that is already in place by CAEEPR.

# The project (cont'd)

## Project Requirements:

### Backend database

- that can be accessed remotely

### Program GUI

- Simple but provides advanced functionality.

### Command Line support

- Able to automate the process.

### Custom Queries

- Able to give the user even more in-depth control over the program

# Methodology

## **AGILE methodology was used throughout development**

- Proved able to handle scope and feature changes as well as delays mid-development.
- Iterative approach was particularly useful for adapting to an project where scope and implementations were uncertain
  - Features did not always fit together well in their original implementation

# Sprint 1

Backend for loading .csv and excel files:

- Can handle incorrect separator on first pass.

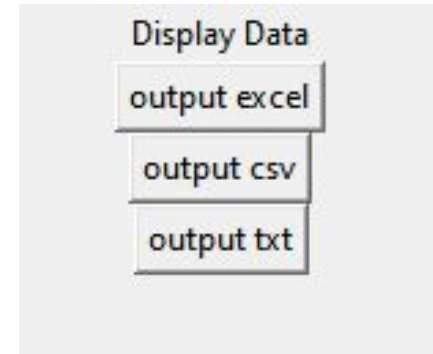
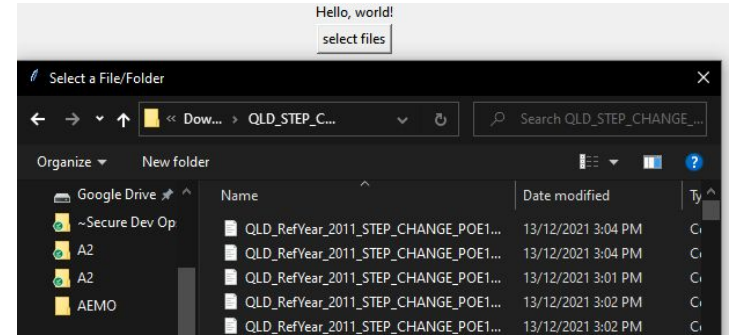
- Can change separator and string classifier.

- Unexpected data types or blank entries are safely ignored.

Displaying the data and printing it to files:

- Can print to csv, excel and space delimited text.

- All files are formatted the correct way.



# Sprint 2

Entering flawed data does not crash the program

Gives error messages that are informative

User has a chance to correct the error on the first attempt

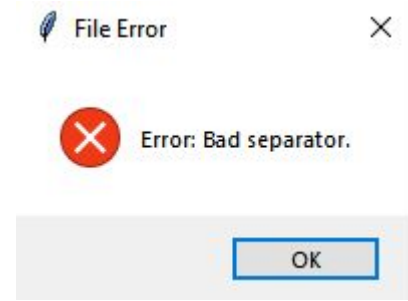
Program supports custom SQL queries

The program can be used to run custom SQL queries

Program can display the data

User can scroll through the data.

Consistent no matter file input.



Custom Query											
date	min	p90	median	average	p10	max	2011	2012	2013		
01/07/2021 00:30	5466.869453	6065.444784	5890.216268	5807.470516	5501.188875	6089.575699	5902.055255	6065.444784	6045.893041		
01/07/2021 01:00	5340.874351	5887.318839	5749.855673	5651.767907	5343.570715	5909.133919	5759.627788	5854.553391	5887.318839		
01/07/2021 01:30	5224.282647	5690.406804	5593.147672	5513.128102	5243.835305	5729.368651	5604.878624	5690.406804	5687.261961		
01/07/2021 02:00	5195.798572	5559.952158	5451.759717	5401.077184	5163.638902	5599.27729	5424.197938	5559.952158	5556.953271		
01/07/2021 02:30	5111.590055	5452.471065	5381.491786	5323.468866	5115.419368	5510.019194	5387.666595	5447.702089	5452.471065		
01/07/2021 03:00	5087.988267	5390.717014	5317.693229	5281.569979	5095.851781	5479.436053	5321.036064	5390.717014	5396.463909		
01/07/2021 03:30	5056.81263	5354.958425	5274.691249	5259.396078	5089.626953	5471.521918	5289.740112	5354.020605	5354.958425		
01/07/2021 04:00	5075.182008	5359.733888	5254.565677	5258.912171	5110.983719	5487.689023	5294.985473	5340.115538	5359.733888		
01/07/2021 04:30	5125.013182	5401.047059	5324.126498	5315.544293	5157.02602	5555.702889	5364.54883	5370.076317	5401.047059		
01/07/2021 05:00	5221.598504	5483.303196	5427.324319	5406.352373	5242.328605	5665.136521	5409.078287	5457.517084	5483.303196		
01/07/2021 05:30	5404.561723	5687.378008	5622.029918	5602.932481	5432.564526	5953.571011	5593.018563	5635.274127	5687.378008		
01/07/2021 06:00	5626.280096	5928.538004	5863.867881	5833.59907	5650.334562	6188.753806	5822.219004	5863.867881	5962.823785		
01/07/2021 06:30	5912.222003	6309.434419	6199.991716	6167.209914	5942.513552	6500.823993	6199.991716	6192.243605	6225.570644		
01/07/2021 07:00	6150.551921	6653.074966	6539.869123	6457.576581	6188.26334	6750.140647	6543.331142	6481.992265	6539.869123		
01/07/2021 07:30	6259.883107	6794.021051	6668.835363	6571.804533	6282.470952	6898.595496	6725.233869	6674.89667	6668.835363		
01/07/2021 08:00	6028.336172	6796.850265	6442.348724	6478.937444	6191.087674	6913.370283	6796.850265	6754.668844	6596.784237		
01/07/2021 08:30	5775.311422	6742.598401	6221.06563	6277.410838	5936.738688	6752.630651	6742.598401	6752.630651	6449.274769		
01/07/2021 09:00	5513.047382	6554.391405	5864.27285	6007.152186	5604.307895	6636.208038	6554.391405	6636.208038	6241.524517		
01/07/2021 09:30	5287.728254	6371.906497	5554.436407	5751.7934	5343.717706	5953.700463	6371.906497	6503.700463	6036.207974		
01/07/2021 10:00	4985.482776	6151.60847	5336.726844	5402.25366	5024.511643	6322.302716	6151.60847	6322.302716	5900.665795		
01/07/2021 10:30	4732.96701	6104.81417	5093.282473	5305.56157	4740.596079	6177.131048	6104.81417	6177.131048	5704.690996		
01/07/2021 11:00	4570.251082	6056.412938	5012.101748	5199.497954	4614.927659	6065.758233	6056.412938	6065.758233	5622.292558		



# Sprint 3

## First implementation of query interface

- Supports general and specific requests

Implemented a system for getting a date and time range from the user

- Uses a calendar widget to allow for an intuitive method of selecting the date.
- Allows the user to select specific times down to half hour intervals.

The screenshot shows a window titled "Display data" with a close button. It contains two identical calendar widgets side-by-side. Each calendar is for the month of July 2021. The first calendar has the 1st of July selected (highlighted in blue). The second calendar has the 1st of July selected (highlighted in blue). Below each calendar are two spinners for "Hour" and "Minute", both set to 0. A "Submit" button is located at the bottom right of the window.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	28	29	30	1	2	3	4
27	5	6	7	8	9	10	11
28	12	13	14	15	16	17	18
29	19	20	21	22	23	24	25
30	26	27	28	29	30	31	1
31	2	3	4	5	6	7	8

Hour: 0 Minute: 0

Hour: 0 Minute: 0

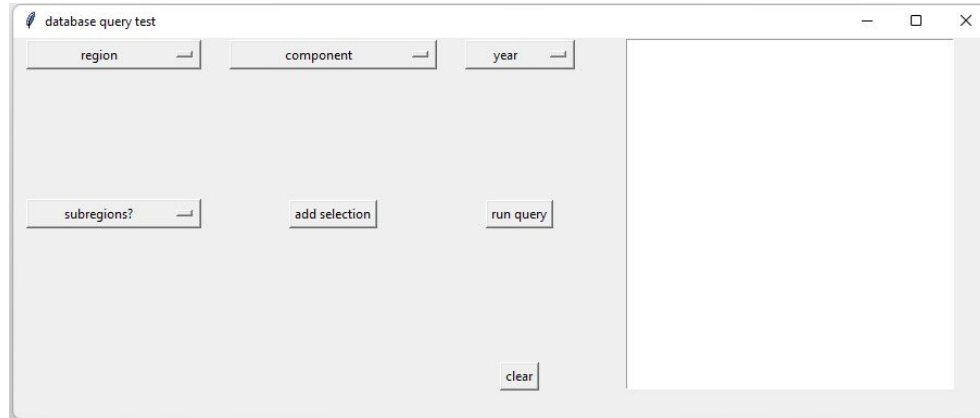
Submit

# Sprint 4

Began integration of all prior features into a deliverable version

Functional change according to client preference: components are now separated into scenario and component to make the number of options in a single dropdown reasonable

-dozens of similar entries in one dropdown now reduced to just 11



# Sprint 5

Added GUI functionality for loading, and deleting files.

Added GUI functionality for showing current date range, and separator settings.

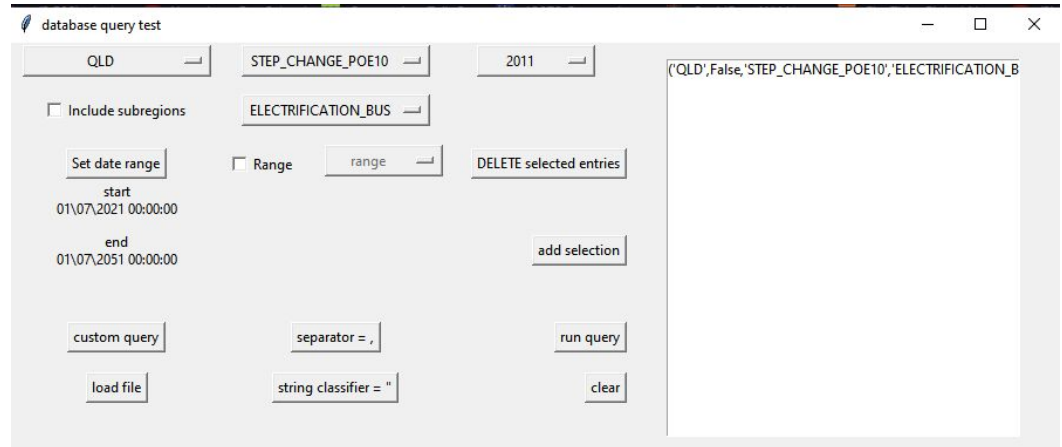
Additional backend for custom query: output printed to input box and .txt file.

Various QoL changes such as:

Error windows

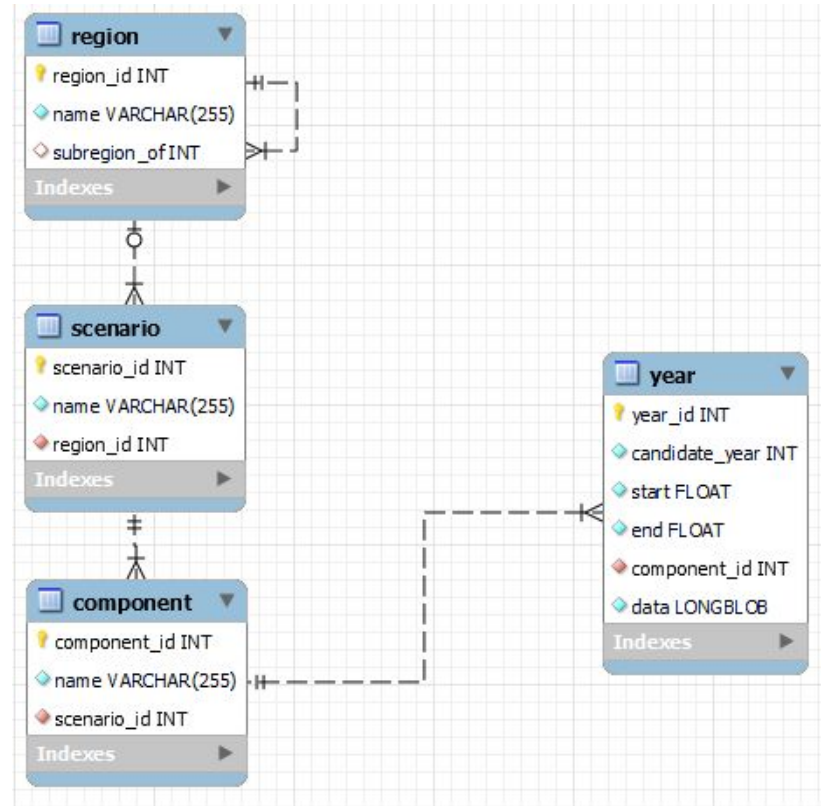
Confirmation boxes

Output and feedback



# Overall Progress

- GUI and associated functions are currently feature-complete
- QoL changes are still ongoing
- Command line arguments are still to be implemented



# Client Feedback

- Client is satisfied with the format in which the data was extracted from row to column and displayed in the GUI.
- Client is satisfied with the functionality within the GUI to search for one or multiple number of .csv files using year, component, region, scenario and display them in the GUI.
- Client is satisfied with the optional feature, where various output formats .csv, excel and space delimited text files can be viewed within the GUI.

# Lessons Learned

Nathan (Dev & Tester):

- Co-operative development
- Time-cost of tech-debt

Joshua (Dev & Tester):

- Working in a team
- Time management skills
- Further knowledge of python

Akshay (CL & BA):

- Facilitating effective communication.
- Analysing client needs and turning them into project requirements.
- Maintaining collaboration throughout project.

Naman (PM & QA):

- Working in Team.
- Communicate clearly.
- Turn feedback into insights.

# User Manual

- The user manual is easy to follow and provides all the information necessary for the operation of the program
- Split into 6 sections:
  1. Summary
  2. Installation
  3. Functions
  4. Operation
  5. Licences
  6. Additional Information

(Live demo)



(QnA)