



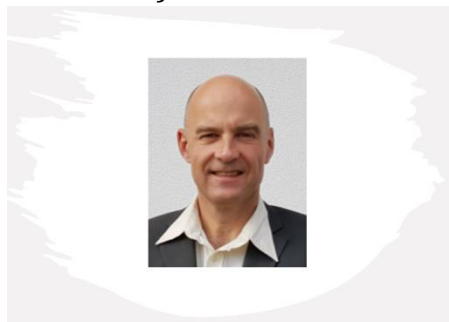
Master Project Report

MATLAB interface to a MySQL database

Mr. Veeresh Nashimath

Matriculation No: **973425**

Under the Professional Guidance of



Prof. Dr.-Ing. Peter Zentgraf, M.Sc.

E-mail: peter.zentgraf@th-rosenheim.de

P

Preface

This report has been written in July 2021. The project was completed at Technische Hochschule Rosenheim in the period of **15.03.2021 – 25.08.2021**. Project was carried out under the expert guidance and immense support of ***Prof.Dr.Peter Zentgraf***. I would like to thank him for his consistent help. I would also like to thank **Technische Hochschule Rosenheim** for providing me this opportunity to execute the project and use its utilities.

This report is subjected to **Matlab App Designer**. The report is intended to enable better understanding of developing an application using Matlab App Designer which allows to read database entries, process and analyze data by different plot options.

Contents

1.	Overview	4
2.	MATLAB App Designer	5
2.1	Introduction to App Designer:	5
2.2	Windows of App Designer:	5
2.3	Example	7
3.	Model-View-Controller	9
4.	Object Oriented Programming	10
4.1	The Components of a MATLAB Class:	10
4.2	Defining Properties and Methods:	11
4.3	Working with objects:	11
5.	List of Features in MySQL2Matlab	13
6.	Connection and Interaction with Database using MATLAB	14
7.	Overview of MYSQL2MATLAB App:	18
8.	Save and Load Sessions	24
8.1	How to Save Session	24
8.2	How to Load already saved session	25
9.	Export and Import	26
10.	Command-Manager:	27
11.	App Testing Framework	29
12.	Web Based Application	30
13.	Project Summary and Future scope:	34
14.	References	35
15.	Appendix 1: User Manual version 1	36

1. Overview

The MATLAB App 'MySQL2Matlab' is used to connect and read out any MySQL Database. Furthermore, it allows to read database entries by some logical operations, it is possible to analyze filtered data by different plotting options. The connection to the MySQL server runs with use of the MATLAB Database Explorer Toolbox.

In this report, how the MATLAB program was implemented, what are the features involved in application and how app testing is carried out are explained in detail in following sections.

2. MATLAB App Designer

2.1 Introduction to App Designer:

App designer is a part of MATLAB Software which is used for developing professional apps. The interface of App Designer is user friendly. The main advantage of App Designer is that the programmer just needs to drag and drop components in the layout and the coding part of the component structure is automatically done in the back end. The programmer then just needs to program the call-back function for each component. The components are limited but it almost covers all the necessary functions.

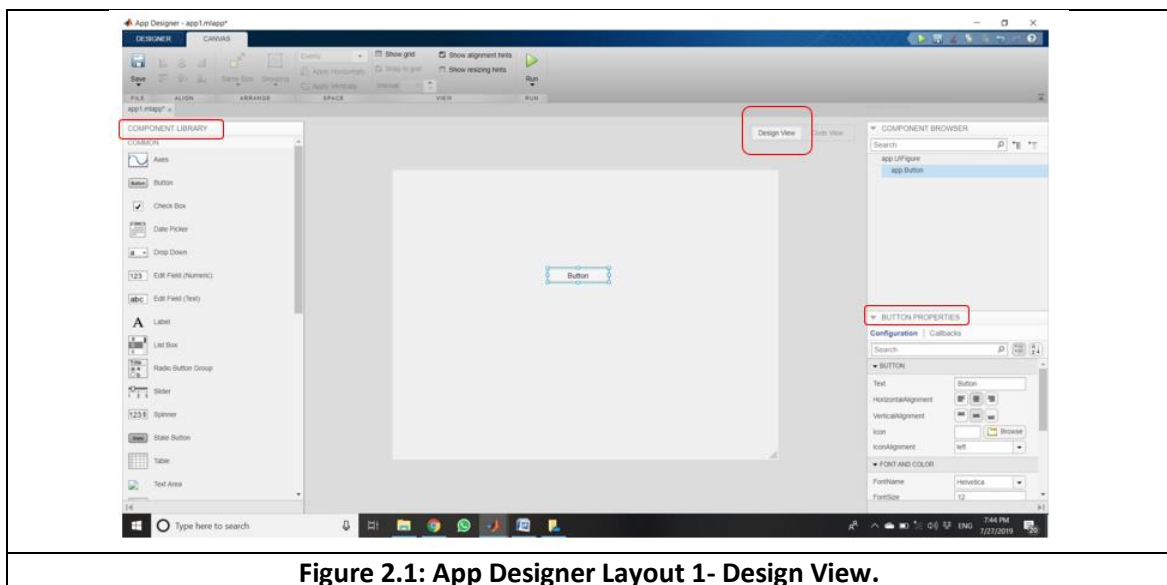


Figure 2.1: App Designer Layout 1- Design View.

In the above figure one can see the layout of App Designer. The left section consists of components library in which one can choose the components to be added to the application.

2.2 Windows of App Designer:

The App Designer has two main windows – Design View and Code View.

2.2.1 Design View: In this window the application must be designed. Designing means to add components from the library to the window and configure it. User can change the name, size, text, orientation, style, etc. of the components. In the above figure, for example, a button is added to the layout. On the right section one can find the Button Properties tab. In this section all the parameters of the component can be changed very easily. Available components in the library are:

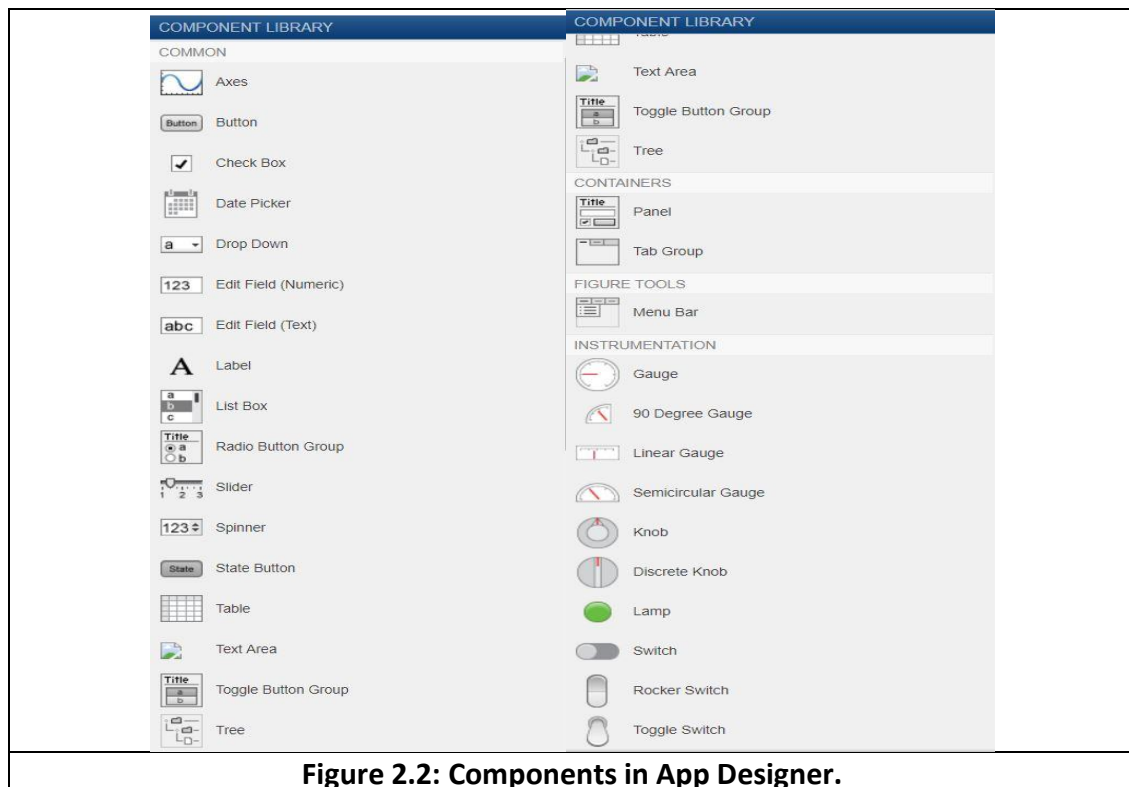


Figure 2.2: Components in App Designer.

2.2.2 Code View: After the designing of an application is done the next step is to program the components. This is done in the code view section.

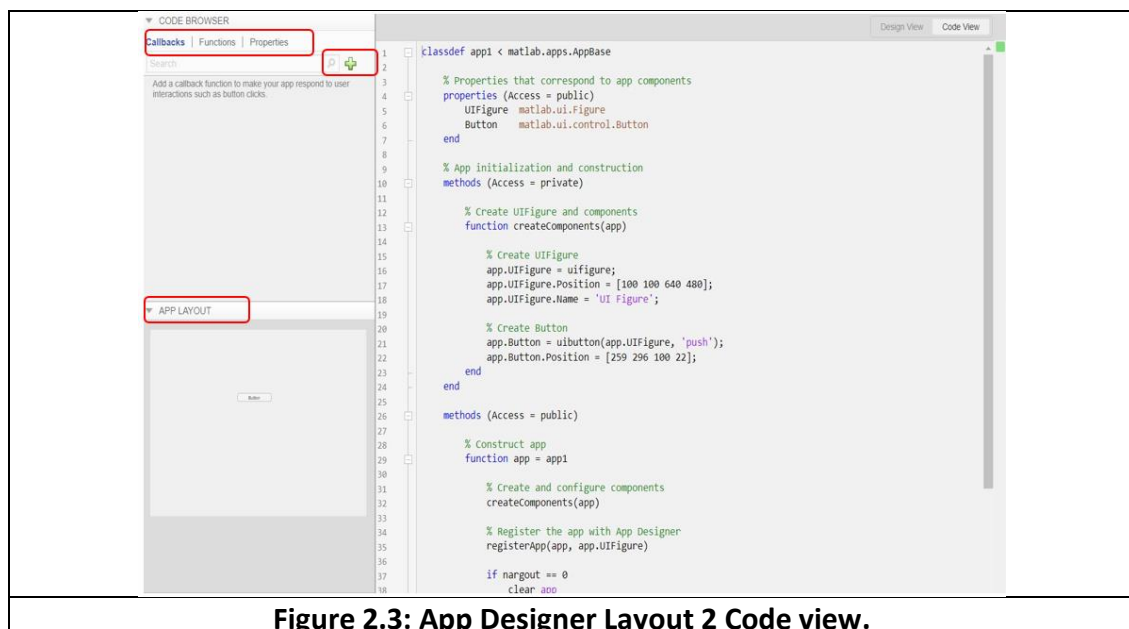
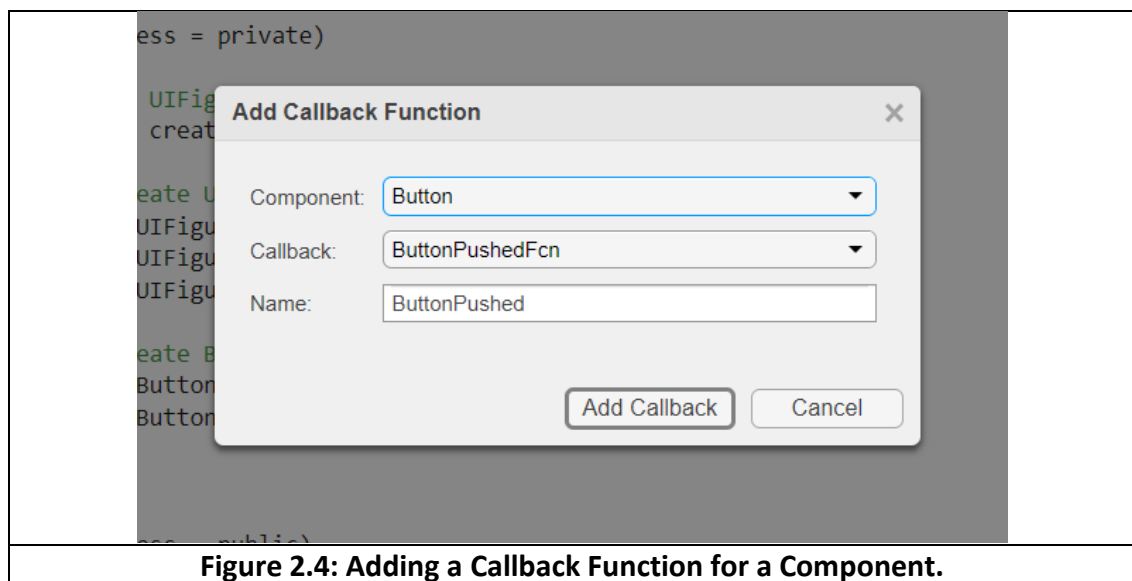


Figure 2.3: App Designer Layout 2 Code view.

The center of this section is the region where the code is written. Initially one can observe that there are many different codes already present. This is the main advantage of using App

Designer. The user is not concerned about the core programming which must be done for designing the component itself. Once the App Designer is started the programming for the layout and for all the components is done automatically.

The left section consists of Code Browser and App Layout. The code browser is used for adding the Callback, Functions and Properties. Once the add sign (in green) is clicked the following dialogue box is displayed. In this the user must define the parameters for the Callback function such as for which component the function has to be called, name of the callback function and the Callback function. Similarly, the user can add a function or a property.



The above figure is for a component. The user can also add start up function. This is applicable when no component is selected so by default the UI figure is selected. Startup functions are the functions which are executed when the App is executed. These functions can be helpful for displaying initial values and for calling functions which must be initially called.

2.3 Example

Here is an example of a callback function for adding two values.

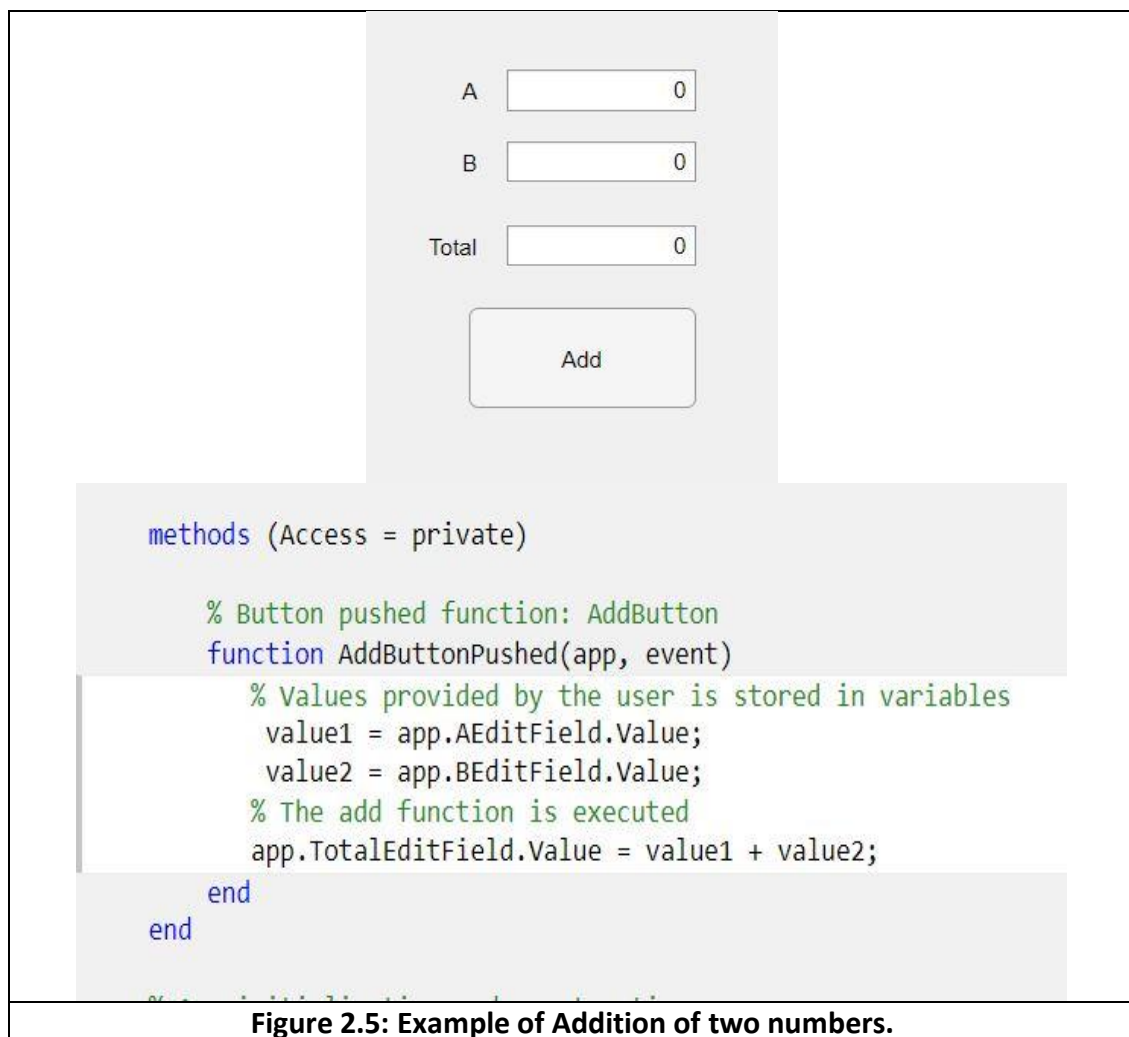


Figure 2.5: Example of Addition of two numbers.

Explanation of the Example: Here three edit field numeric component and one button is taken. In the code view a callback function is assigned as shown in **Figure 2.5** and the above code is written. Once the button is pushed first the values which user has entered is stored in variables value1 and value2 and later the total field box displays the addition of value1 and value2.

3. Model-View-Controller

Model–View–Controller (usually known as MVC) is a Software design pattern commonly used for developing Applications that divides the related program logic into three interconnected elements [2]. The big idea behind MVC is that each section of the code has a purpose, and those purposes are different. Part of code holds the data of app, some of code is responsible for visualization, and some other part of code controls how app functions.

Traditionally used for desktop graphical user interfaces (GUIs), this pattern has become popular for designing web applications. Popular programming languages have MVC frameworks that facilitate implementation of the pattern.

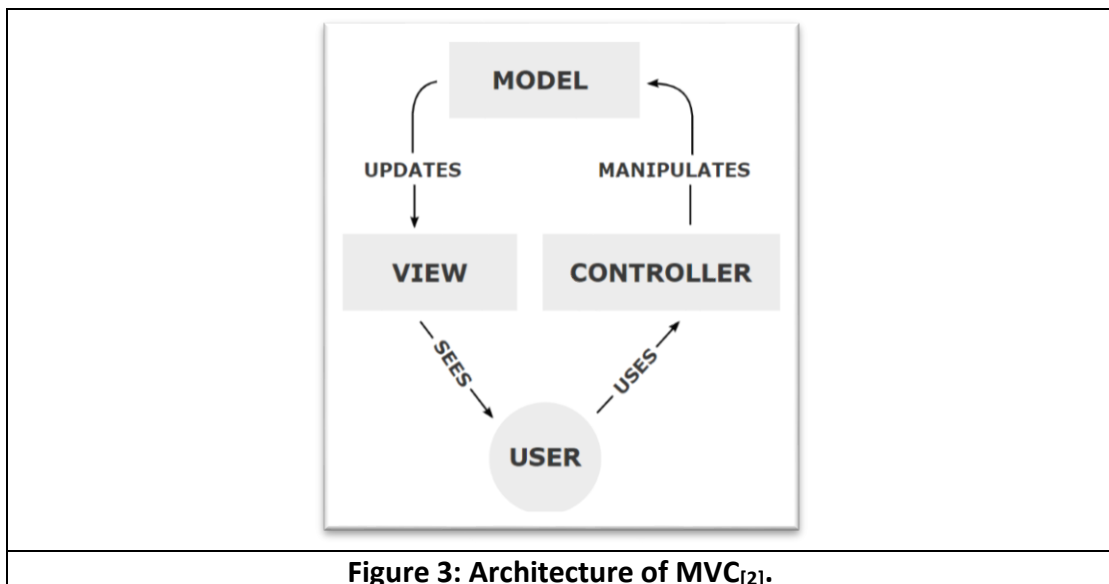


Figure 3: Architecture of MVC_[2].

3.1 Parts of MVC:

Model: Model code typically reflects real-world things. This code can hold raw data, or it will define the essential components of your app. For instance, if you were building a To-do app, the model code would define what a “task” is and what a “list” is – since those are the main components of a to do app [6].

View: View code is made up of all the functions that directly interact with the user. This is the code that makes your app look nice, and otherwise defines how your user sees and interacts with it [7].

Controller: Controller code acts as liaison between the Model and the View, Receiving user input and deciding what to do with it. It is the “brain” of the application, and ties together the model and the view [7].

4. Object Oriented Programming

Definition:

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which can contain data and code: data in the form of fields (often known as attributes or properties), and code in the form of procedures (often known as methods)^[1].

Using object-oriented programming in MATLAB, it is possible to manage software complexity by organizing your code into logical components that are easier to maintain and extend. Code duplication can be avoided by creating reusable objects with well-defined interfaces that hide the complexity of the underlying code. Furthermore, your objects can evolve and change over time without introducing incompatibilities in client code.

4.1 The Components of a MATLAB Class:

A MATLAB class contains a blueprint or set of instructions used to build a specific type of object. Class definitions start with the `classdef` keyword and have three major components:

- Properties blocks define the properties that store data for each of the objects of the class
- Methods blocks contain a set of functions that define the operations that can be performed on each object of the class
- Events blocks define messages that an object will send to other parts of an application when something changes in that object.

```
classdef sads % Sensor Array Data Set Class

    properties
        Data % Sampled sensor data
        SampleRate % Sample rate (Hz)
    end

    properties (Access = private)
        Wavelength % Wavelength of sources (m)
    end

    methods
        function obj = sads(Data, Wavelength, SampleRate)
            % SADS Create sensor array data set
            obj.Data = Data;
            obj.SampleRate = SampleRate;
            obj.Wavelength = Wavelength;
        end
        function angles = doa(obj)
            % DOA Estimate direction of arrival of sources in the data
            [mags, fflip] = magfft(obj, 256);
            maxtab = peakdet(mags, .1);
            angles = sort(fflip(maxtab(:,1))*180);
        end
    end

    methods (Access = private)
        function [mags, fflip] = magfft(obj, zeroPadTo) ...
    end

end
```

Figure 4.1: A sample MATLAB class definition

4.2 Defining Properties and Methods:

When defining a class, you can specify attributes to control how your class properties and methods behave and how they are accessed from outside the object. For example, properties and methods can be public, private, or protected. When specifying properties, you can use validation syntax to avoid writing code for error checking.

<u>Property Attributes</u>	<u>Method Attributes</u>
Abstract	Abstract
Access (public, private, or protected)	Access (public, private, or protected)
Constant	Hidden
Dependent	Sealed
GetAccess	Static
GetObservable	
Hidden	
SetAccess	
SetObservable	
Transient	

<u>Example Property Validation Syntax</u>	
<code>properties (Access = public)</code>	
Data	<code>{:,} {mustBePositive, mustBeFinite}</code>
Spacing	<code>{mustBeInteger}</code>
SampleRate	<code>{mustBePositive}</code>
<code>end</code>	
<code>properties (Access = private)</code>	
Wavelength	<code>{mustBeGreaterThan(Wavelength,0.001)}</code>
<code>end</code>	

Figure 4.2: A sample MATLAB property / method attributes and validation syntax to enforce property types, sizes, and values.

4.3 Working with objects:

You can create objects using a special method called the class constructor. Calling the constructor is like calling any MATLAB function and can be used to create a single object or an array of objects. You can access object properties just like you would access the fields of a struct. Object methods are called just like ordinary MATLAB functions.

MATLAB objects have unique features relative to other languages. For example, you can modify a class at any time and objects of that class will update immediately. In addition, MATLAB manages the lifecycle of objects without requiring any explicit memory allocation or deallocation and without the type of non-deterministic garbage collection used in some other languages.

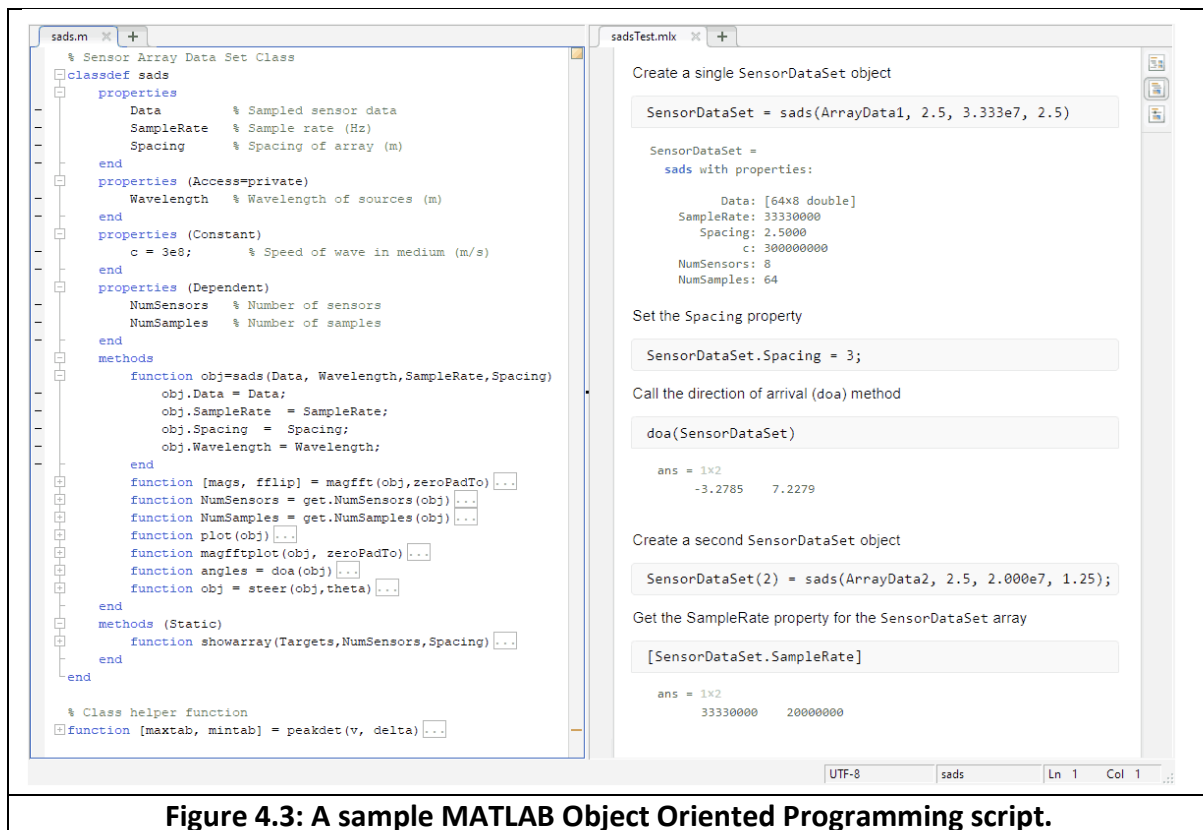


Figure 4.3: A sample MATLAB Object Oriented Programming script.

5. List of Features in MySQL2Matlab

Below table contains all the features and its corresponding status of MySQL2Matlab application.

Feature	Feature Description	Status of Feature
Custom Login UI	As a User I want to have a custom login dialog which demands a database url, username and password. So that I can login on an arbitrary database. Acceptance Criteria: A UI dialog on startup is provided to log into an arbitrary database. Backend is capable to login on an arbitrary MySql database.	Implemented
DiffPlot	As a User I am able to create a diffplot between on one column (succeeding-preceding element) or two columns. So that I can display the difference between two columns Acceptance Criteria: UI offers diffplot field and backend is capable of it.	Implemented
Google Earth Simulation	As a User I am able to do a Google Earth Simulation with my longitude and latitude data. Acceptance Criteria: I can do a Google Earth Simulation like in old MySQL2Matlab.	Need to be Implemented
Visualization	As a User I am able to select columns of a database table and plot them in 3d and 2d Acceptance Criteria: I can plot all columns in an arbitrary manner in 2d and 3d	Implemented
Databasefilter	As a User I am able to filter data according to a given column and request the data from the database. Acceptance Criteria: UI has possibility to filter whole table according to a filter parameter. Filter Parameters which must work: GP_ID, Times, Counter	Implemented
Export	As a User I want to be able to export the current table as .csv data Acceptance Criteria: UI gives the possibility to export data to .csv Backend is capable of doing this	Implemented
Coordinate Transformation	As a user I want to be able to convert my latitude and longitude information according to a given geodesic transformation Acceptance Criteria: Latitude and Longitude data gets converted according to a specified transformation Data gets plotted	Need to be Implemented
Visualize DWD Cells	As a User I want to be able to plot the DWD cells for a given timestamp Acceptance Criteria: I can plot the DWD cells for a given timestamp	Need to be Implemented
Animate Haildefence flight	As a User I want to be able to run an animation which shows the flight path and the DWD Cells for a given time frame Acceptance Criteria: I can see the flight path and development of the DWD Cells for a given timeframe	Need to be Implemented
Creation of KML data	As a User I am able to create a KML file for google earth like in old MySQL2Matlab Acceptance Criteria: KML data can be created	Need to be Implemented
Tests are provided	Framework from Mathworks to ensure ongoing quality of the software and make sure further development can go on	Implemented
Evaluate own functions on plot data	As a user I want to be able to plot columns with a given matlab own function. For example user inputs "flip(xData)" This gets evaluated with matlab eval() and executed Acceptance Criteria: Input textfield in UI and capable backend	Implemented
SumPlot	As a User I am able to create the cumulated sum plot (function cumtrapz)	Implemented
Logical Combination	As a User I can filter one column with AND, OR, XOR with another column, like column x AND column y with values 2 < values < 4	Implemented
Plot range selection by date and time	As a User I can select a plot range by any DateTime column the database provides. At Plotrange select dialog the user selects the desired DateTime column, then the start point of time and the stop point of time of the plot.	Implemented
Plot the Number of consecutive equal data	Plot the frequency of each element of column	Implemented
Web Based App	Application must run in web browser	Implemented
Save and Load sessions	User can save and load session at any state of application	Implemented

6. Connection and Interaction with Database using MATLAB

The Database explorer is one of the apps in MATLAB which connects quickly to a database, explore the database data, and import data into the MATLAB workspace in a visual way.

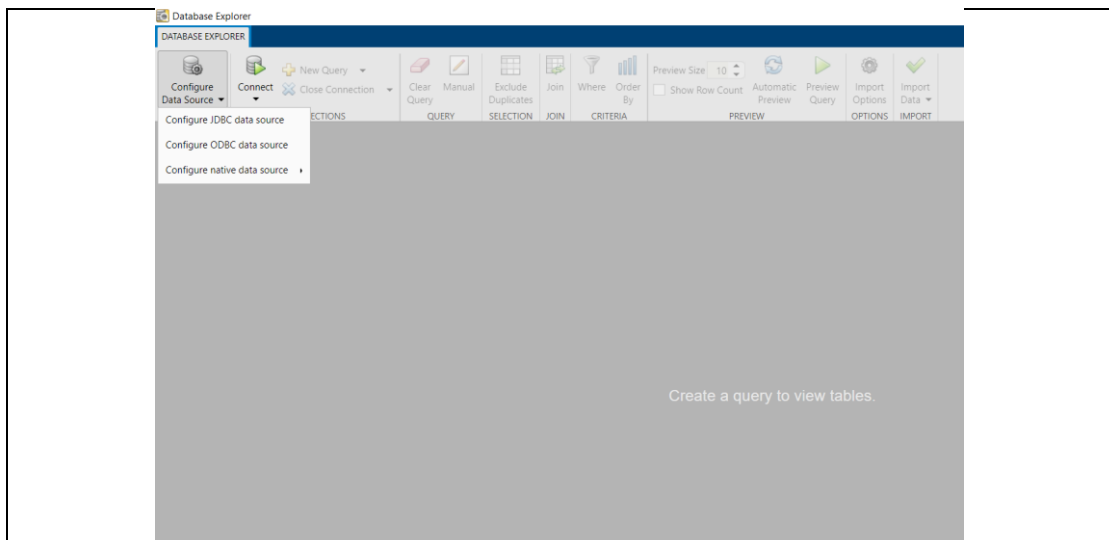
If user wants to browse the data in required database quickly, this app suitable to interact with database^[8].

Using the Database Explorer app,

- Create and configure ODBC and JDBC data sources.
- Establish multiple connections to the same or different databases.
- Select tables and columns of interest.
- Fine-tune selections using SQL query criteria.
- Preview selected data.
- Customize import options.
- Import selected data into the MATLAB workspace for analysis.
- Save generated SQL queries.
- Generate MATLAB code.

Following steps explains general steps involved in extracting data from Database using database explorer.

Step1: Configure Data Source: Click on Configure Data Source icon, select relevant data source, enter connection parameters, and click on Test.



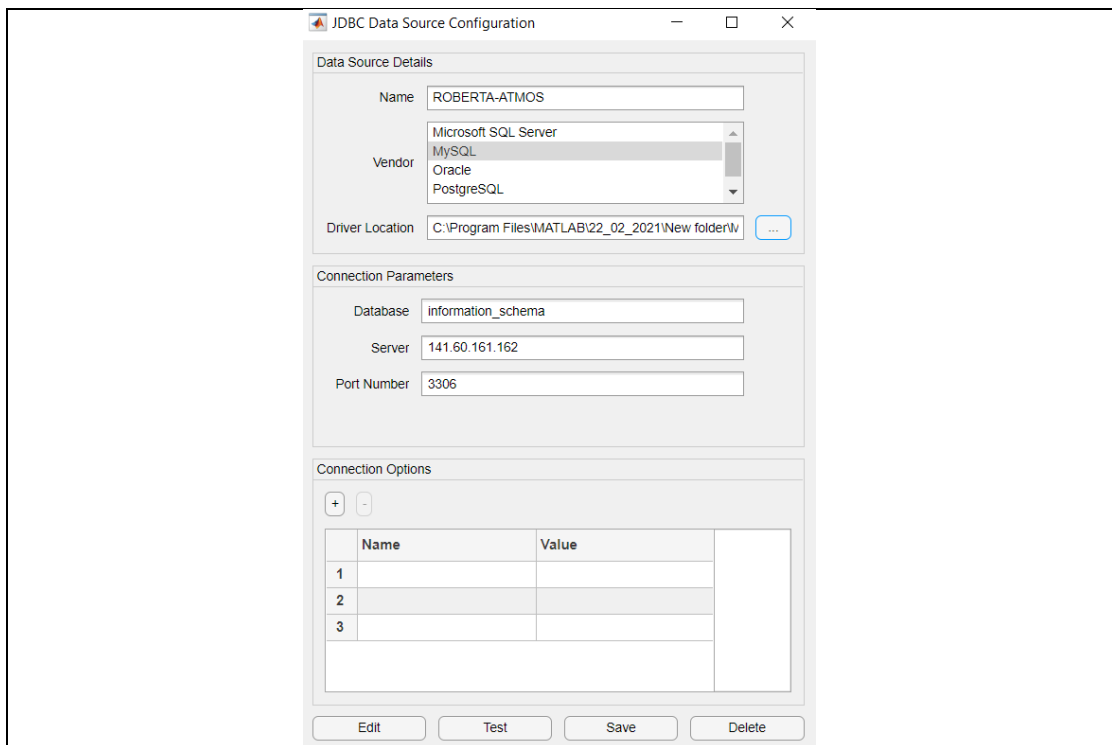


Figure 6.1: Configure Data Source.

Step 2: After successful test, connect to the database by entering Log-In credentials.

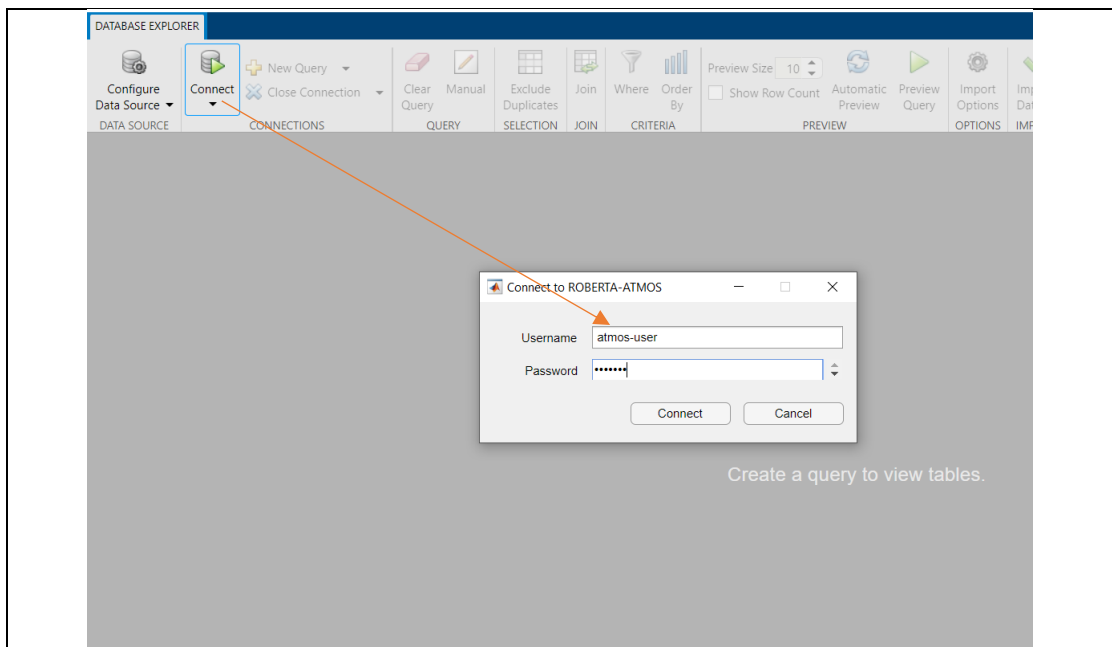


Figure 6.2: Connect to database.

Step 3: After successful Logged in user can select required table and corresponding columns of the database as shown below. As highlighted by circle MATLAB generates automatic SQL Query based on user selected of table and columns. It is also possible to write a SQL in manual mode.

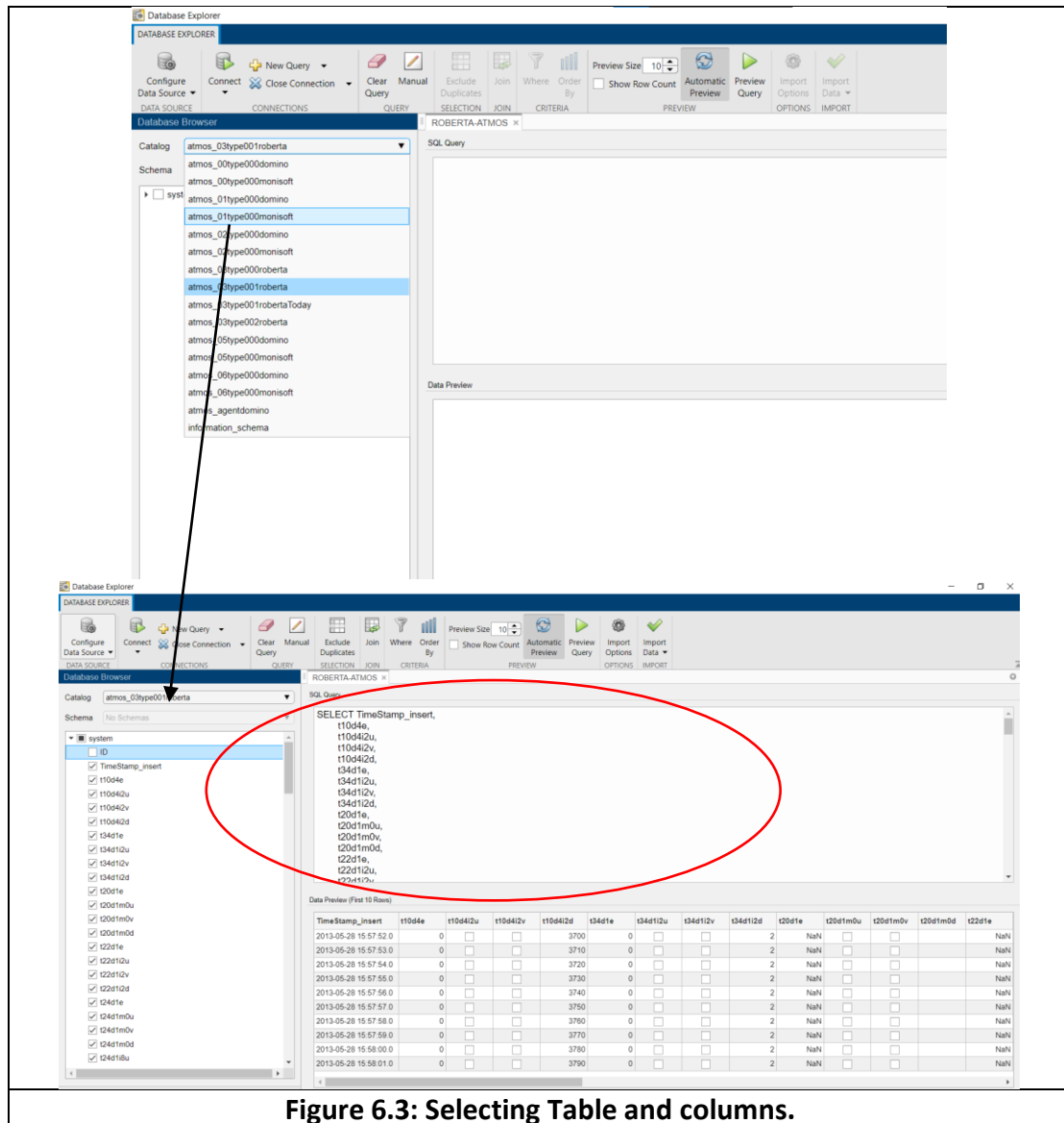
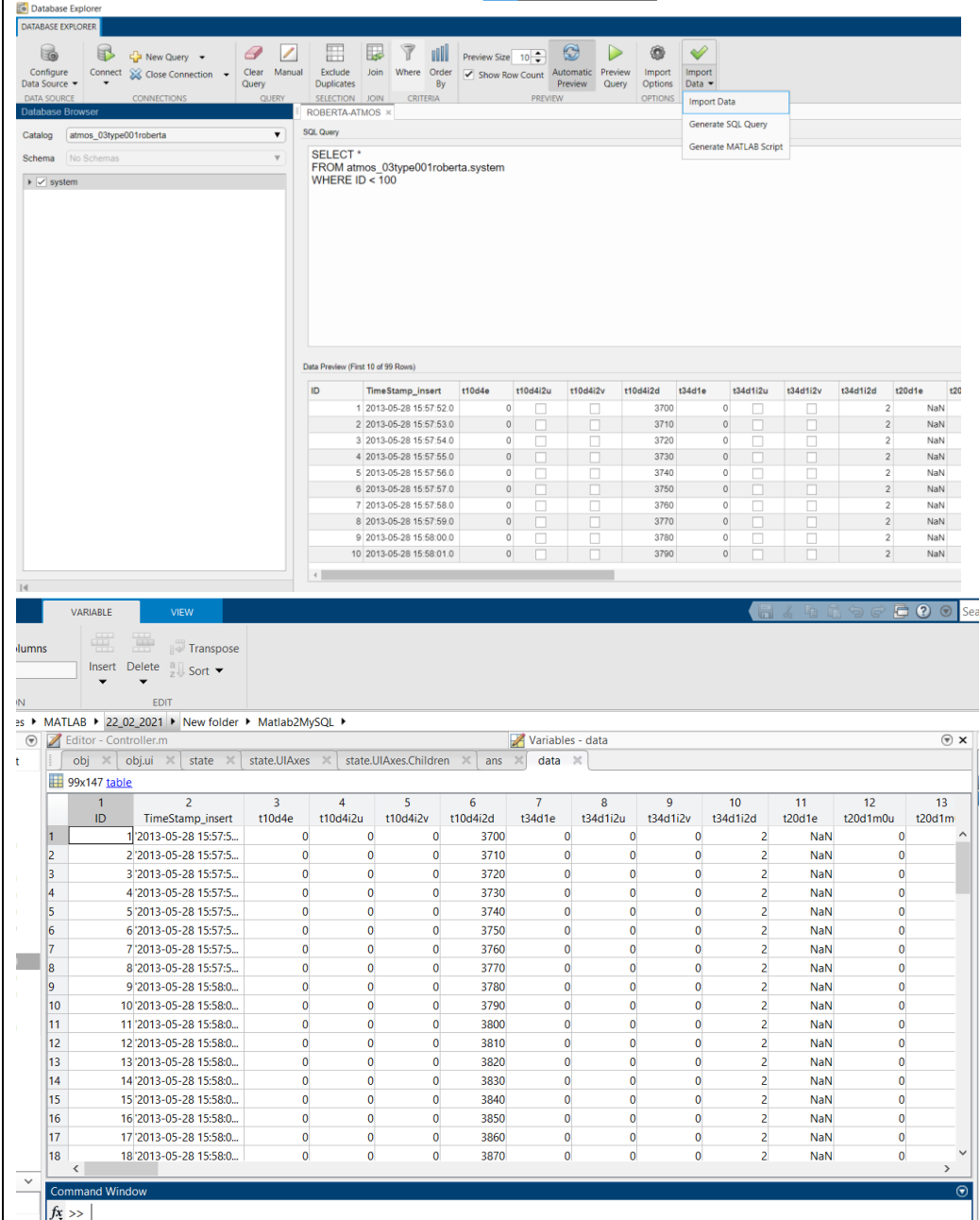


Figure 6.3: Selecting Table and columns.

Step 4: After writing required SQL syntax, user can also extract the data into MATLAB workspace by clicking on Import Data.



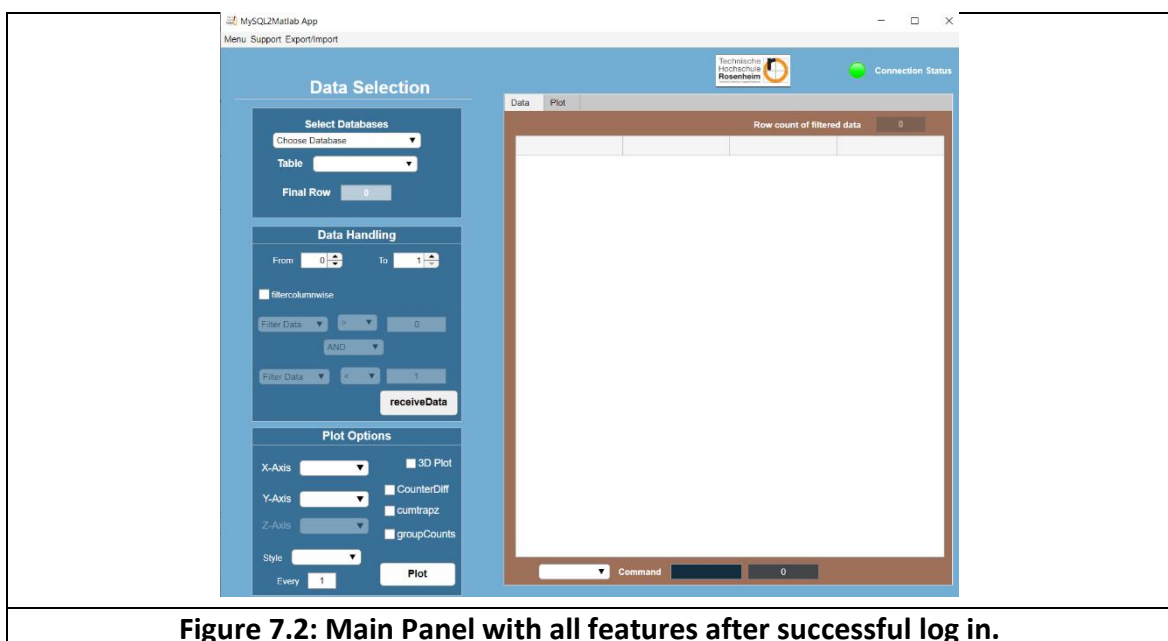
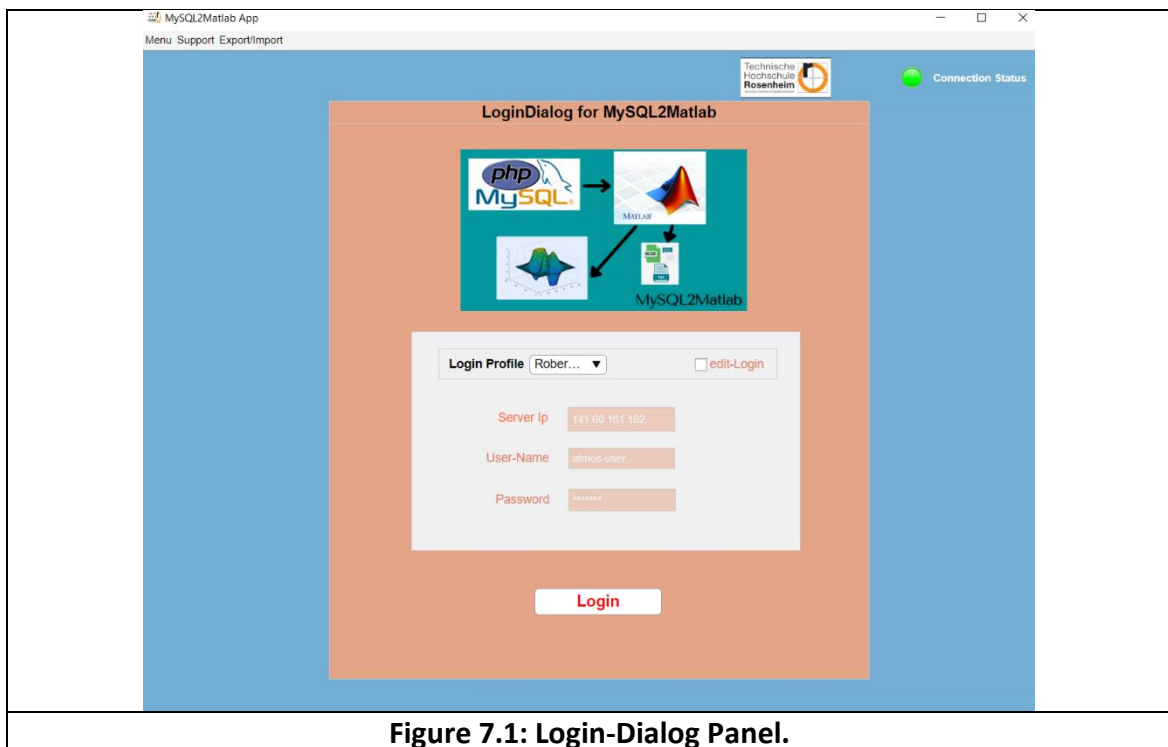
The screenshot shows the Database Explorer interface with a SQL query: `SELECT * FROM atmos_03type001roberta.system WHERE ID < 100`. The 'Import Data' menu is open, showing options to 'Generate SQL Query' and 'Generate MATLAB Script'. Below the query, a data preview table is shown with columns: ID, TimeStamp_insert, t10d4e, t10d4i2u, t10d4i2v, t10d4i2d, t34d1e, t34d1i2u, t34d1i2v, t34d1i2d, t20d1e, t20d1m0u, t20d1m. The table contains 10 rows of data. Below the preview, the MATLAB interface is shown with the 'Variables - data' window displaying the imported data as a table with 13 columns and 18 rows.

ID	TimeStamp_insert	t10d4e	t10d4i2u	t10d4i2v	t10d4i2d	t34d1e	t34d1i2u	t34d1i2v	t34d1i2d	t20d1e	t20d1m0u	t20d1m
1	2013-05-28 15:57:52.0	0			3700	0	0	0	2	NaN	0	
2	2013-05-28 15:57:53.0	0			3710	0	0	0	2	NaN	0	
3	2013-05-28 15:57:54.0	0			3720	0	0	0	2	NaN	0	
4	2013-05-28 15:57:55.0	0			3730	0	0	0	2	NaN	0	
5	2013-05-28 15:57:56.0	0			3740	0	0	0	2	NaN	0	
6	2013-05-28 15:57:57.0	0			3750	0	0	0	2	NaN	0	
7	2013-05-28 15:57:58.0	0			3760	0	0	0	2	NaN	0	
8	2013-05-28 15:57:59.0	0			3770	0	0	0	2	NaN	0	
9	2013-05-28 15:58:00.0	0			3780	0	0	0	2	NaN	0	
10	2013-05-28 15:58:01.0	0			3790	0	0	0	2	NaN	0	

Figure 6.4: Selecting Table and Columns.

7. Overview of MYSQL2MATLAB App:

In this section, all the features of Graphical user interface developed for this application is shown and explained below. There are two panels designed in the UI, one is for log-in Panel and other one is MySQL2Matlab application main window. After successful logged in, log-in panel will be deleted, and main panel will appear. Below **Figure 7.1** and **Figure 7.2** shows log in panel and main panel respectively.



7.1 Data Selection:

Data selection section has 2 dropdowns and one numeric field. One of the dropdowns is used to Choose database and another one is used to select table of corresponding database. Numeric field is used to showcase total number of Rows present is selected Table. **Figure 7.1.1** shows that selected Database and Table are “atmos_03type001roberta” and “system” and final row count “658609” indicates that total number of rows present in table “system”.

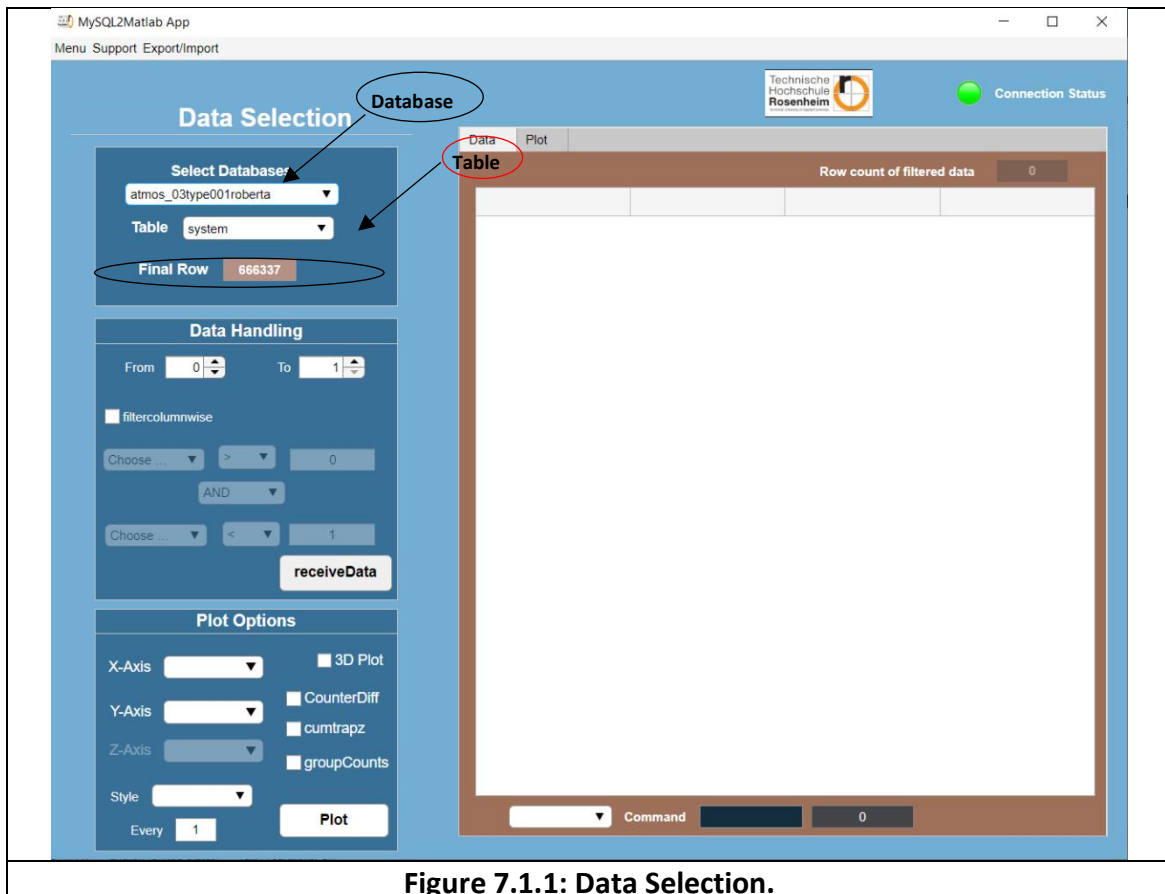


Figure 7.1.1: Data Selection.

7.2 Data Handling

Data Handling features are used to apply filter on selected Table of Database. Many features are implemented in this section which are mainly divided in to two types and those are explained in section 7.2.1 and 7.2.2.

7.2.1 Applying filters using Limits

User can filter the Data of selected table by entering values of limits in From and To spinner features. For example, if user wants to offset first 100rows of table and needs next 50rows, that means user needs only data between 100th row and 150th row of the table, in this case user should enter from spinner value as 100 and 50 as To spinner value. Refer **Figure 7.2.**

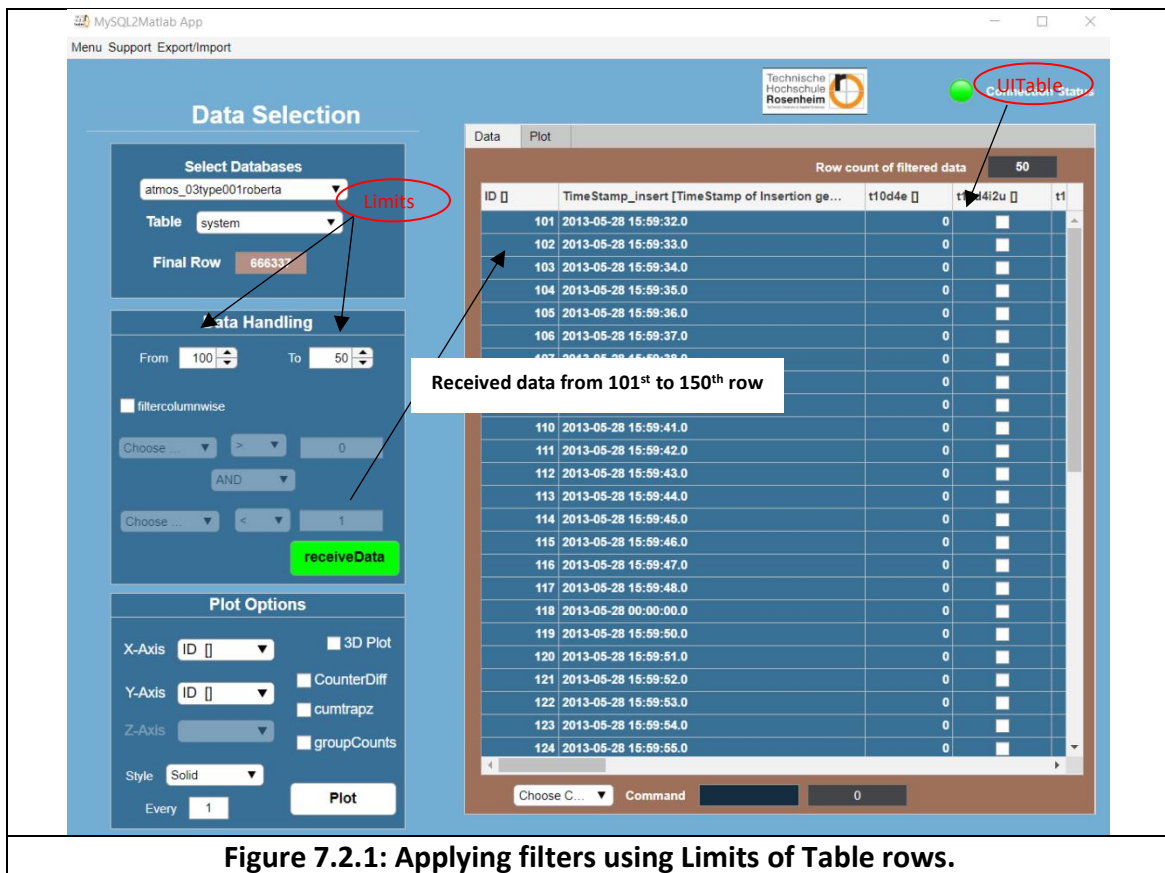


Figure 7.2.1: Applying filters using Limits of Table rows.

7.2.2 Applying filters by columns:

Other option to filter the data is by applying filters between the columns by clicking on checkbox 'filtercolumnwise' which allows to activate the dropdowns to select columns and logical operations. In Fig 7.2.2 filtering is done between single column using logical AND operations.

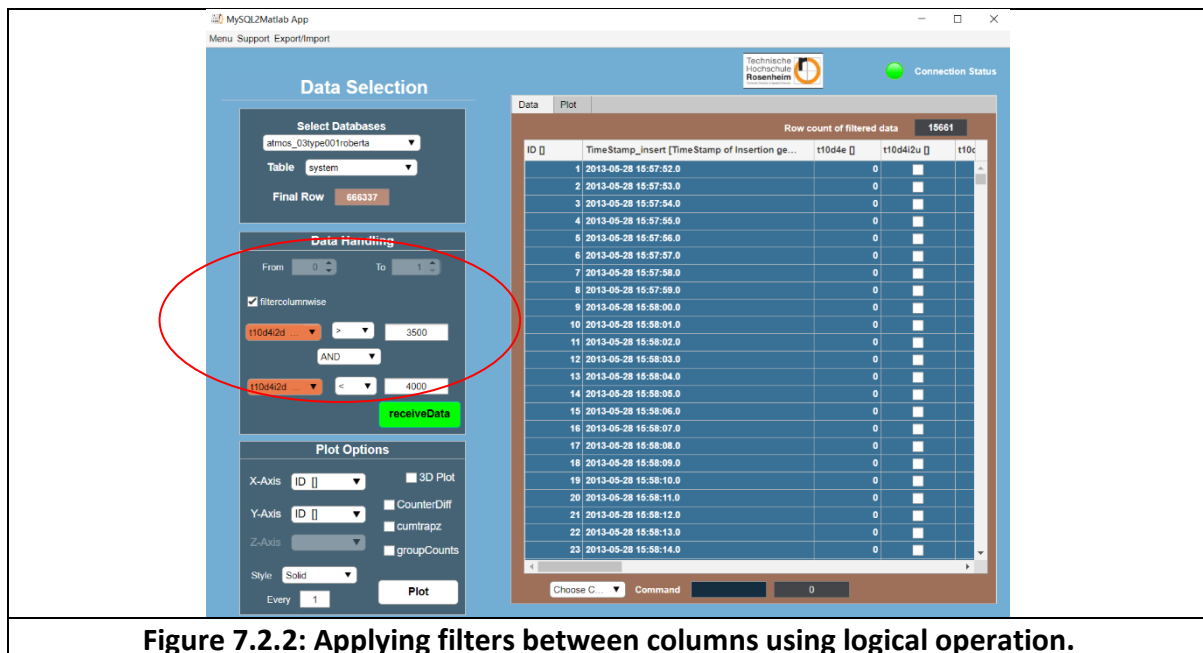


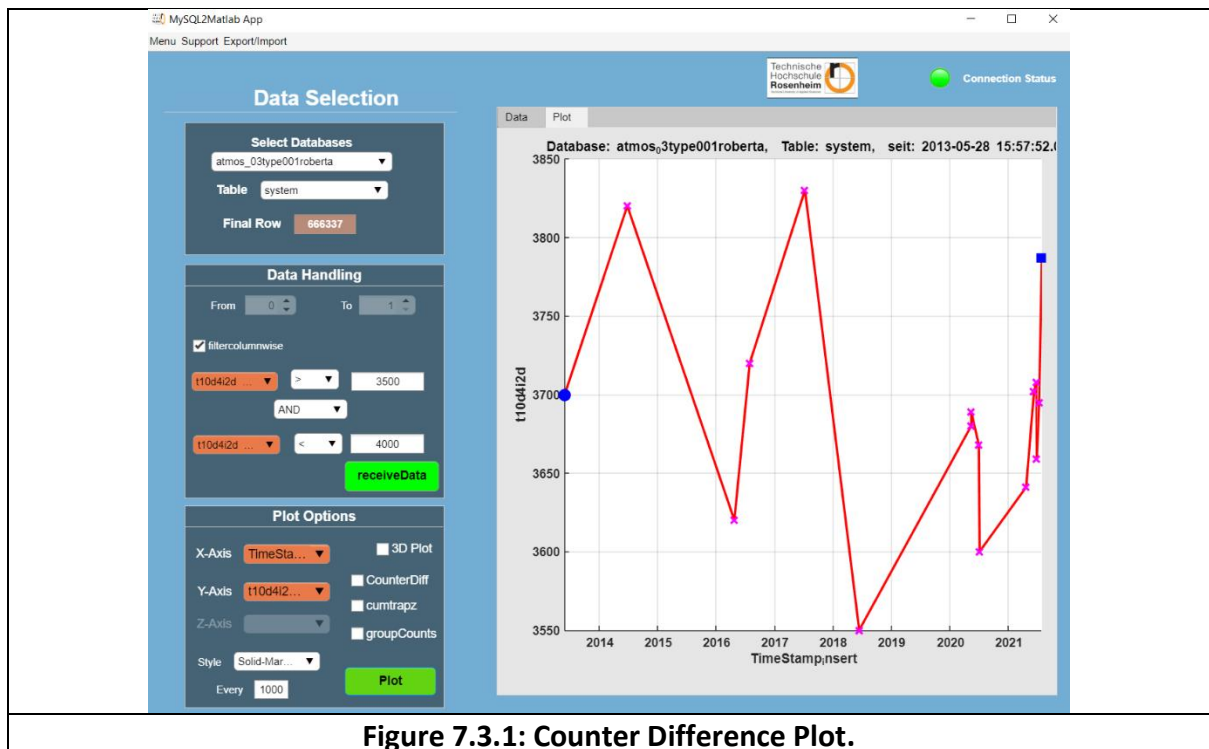
Figure 7.2.2: Applying filters between columns using logical operation.

7.3 Plot Options:

After extracting the filtered data in the UITable, user can analyze the columns of UITable by using features available in plot options. There are 3 types of plot options are available which are explained below in detail. User can also plot with different kinds of marker options by selecting required marker from “**style**” dropdown. It is also possible set the ratio of plotting values by entering value in “**Every**” editfield.

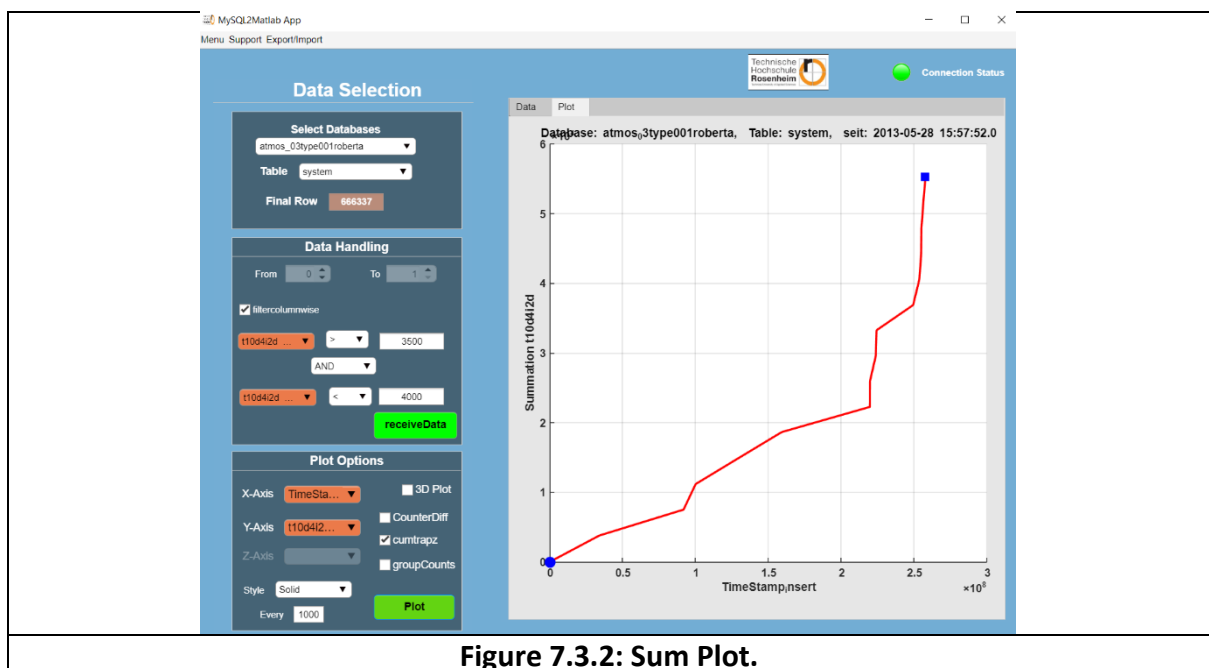
7.3.1 Counter Difference Plot:

User can convert selected Y-Axis column values to counter difference values and plot those against selected X-Axis column values by selecting checkbox “**CounterDiff**” refer below **Figure 7.3.1** to get more clarification.



7.3.2 Sum Plot:

User can convert selected X-Axis column values to Cumulative sum values and plot those against selected X-Axis column values by selecting checkbox “**cumtrapz**” refer below **Figure 7.3.2** to get more details.



7.3.3 Plot the Number of consecutive equal data: It is also possible to plot frequency of each value of X-Axis against X-Axis values by selecting '**groupCounts**' checkbox. Refer Figure 7.3.3.

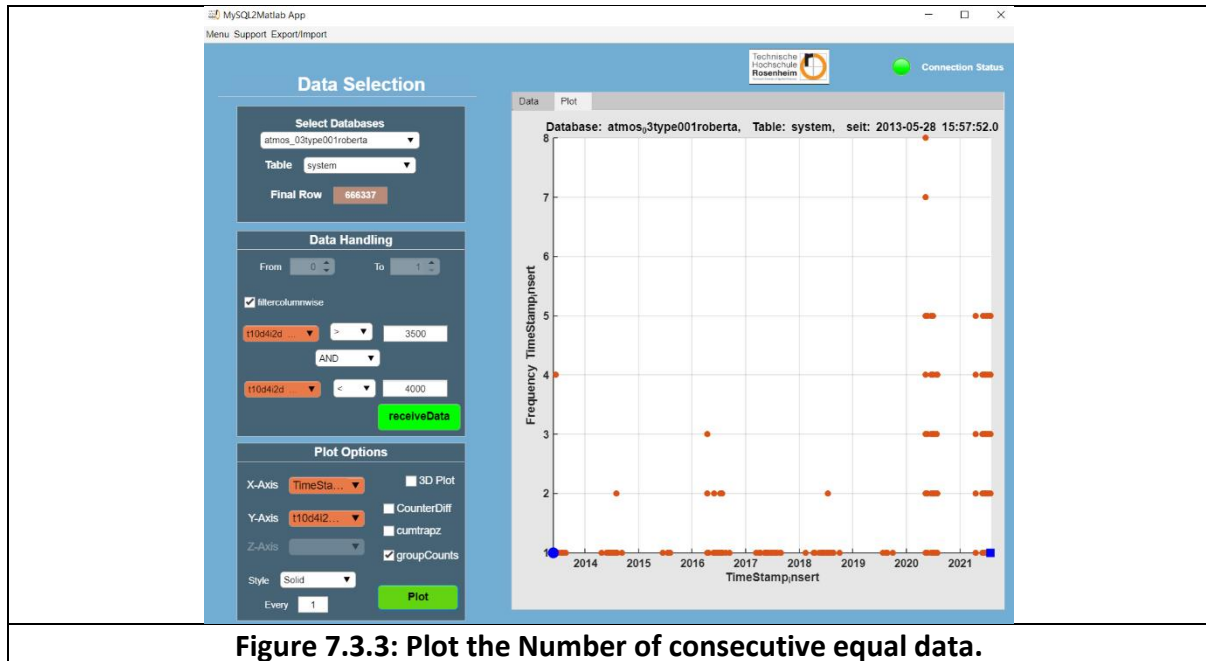


Figure 7.3.3: Plot the Number of consecutive equal data.

7.3.4 3D plot

3D plot can be generated by using 3D checkbox and selecting Z-Axis dropdown value. Even it is also possible to use CounterDiff and cumtrapz options to generate in 3D environment. Refer **Figure 7.3.4**.

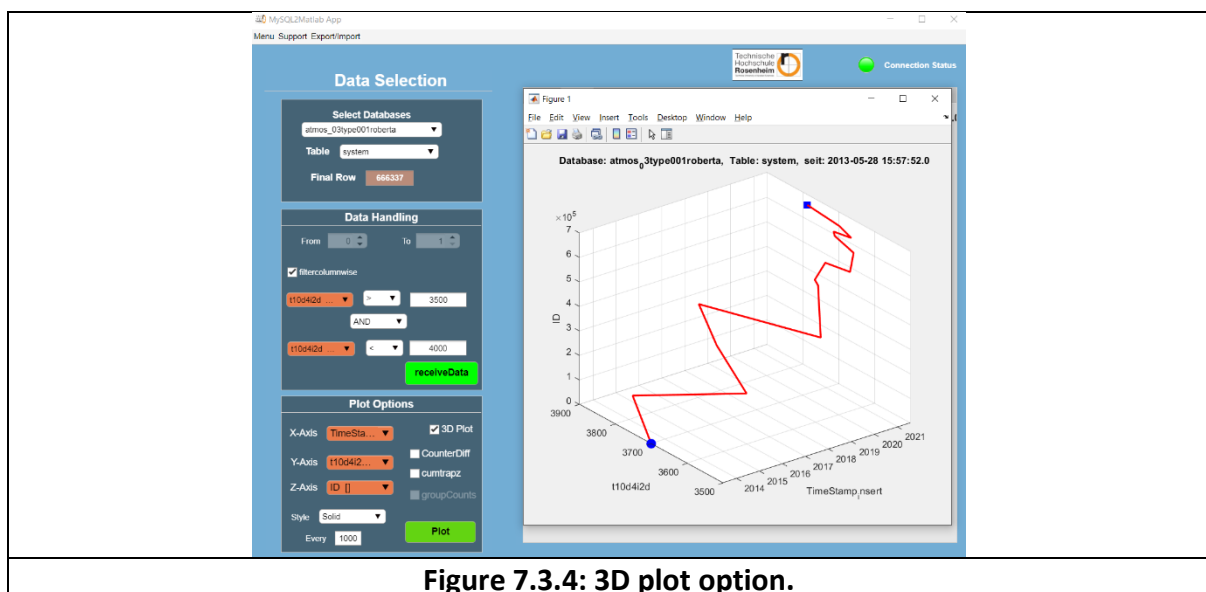


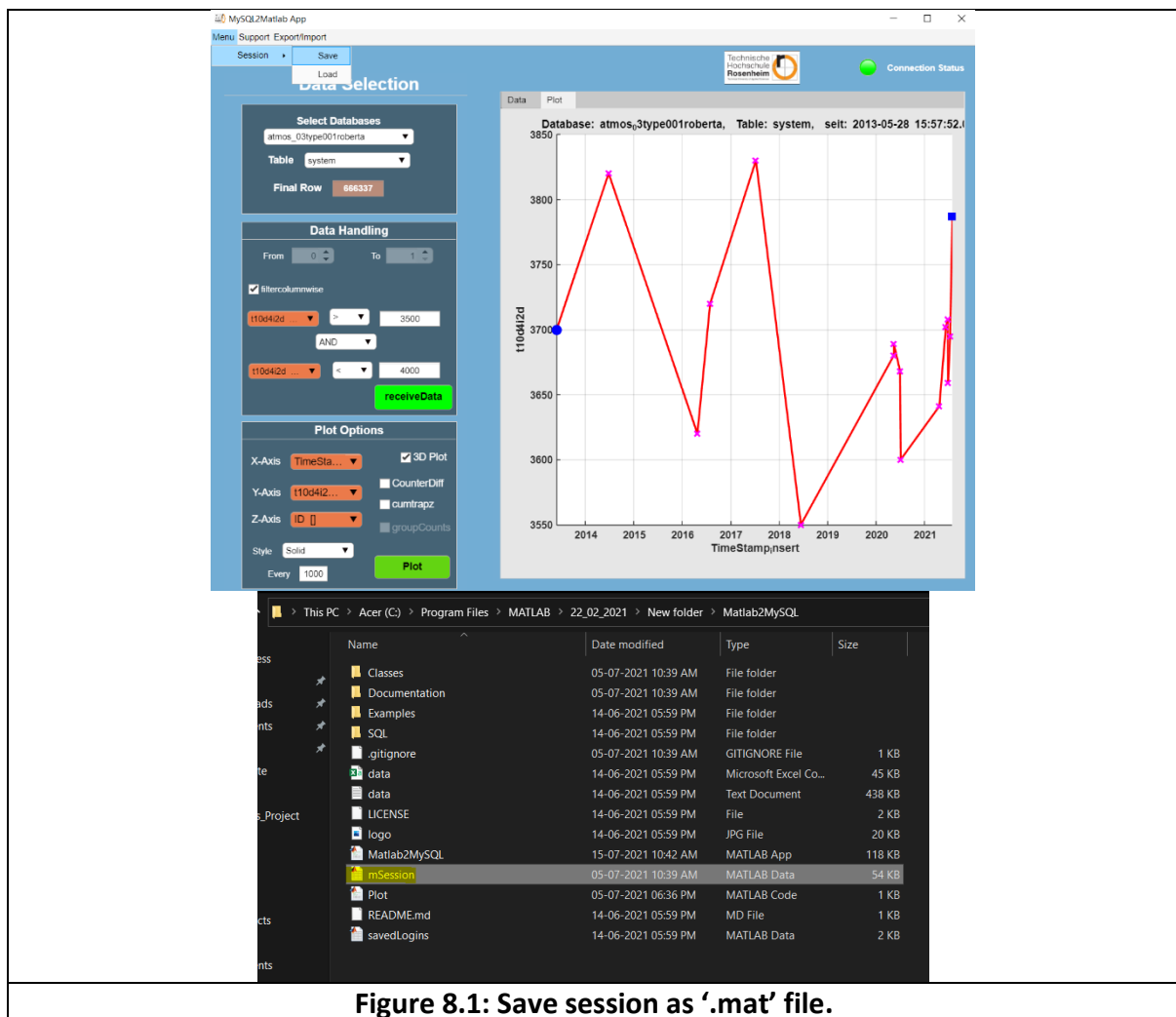
Figure 7.3.4: 3D plot option.

8. Save and Load Sessions

It is always hard to repetitive tasks, to avoid those repetitive work, save and load session features are implemented in MySQL2Matlab. If user wants to save his/her work and load it at later time, it is possible by using 'save' and 'load' features available in application.

8.1 How to Save Session

In order to save the state of work before closing the application, user needs to navigate to **menu** option then select **save** option which creates the state of work as '.mat' file format at selected path. Refer below **Figure 8.1**.



8.2 How to Load already saved session

In order to load the previously saved session as **‘.mat’** file, navigate to **menu** option and select **Load** and load the corresponding **‘.mat’** file to get previous state of application. Refer below **Figure 8.2**.

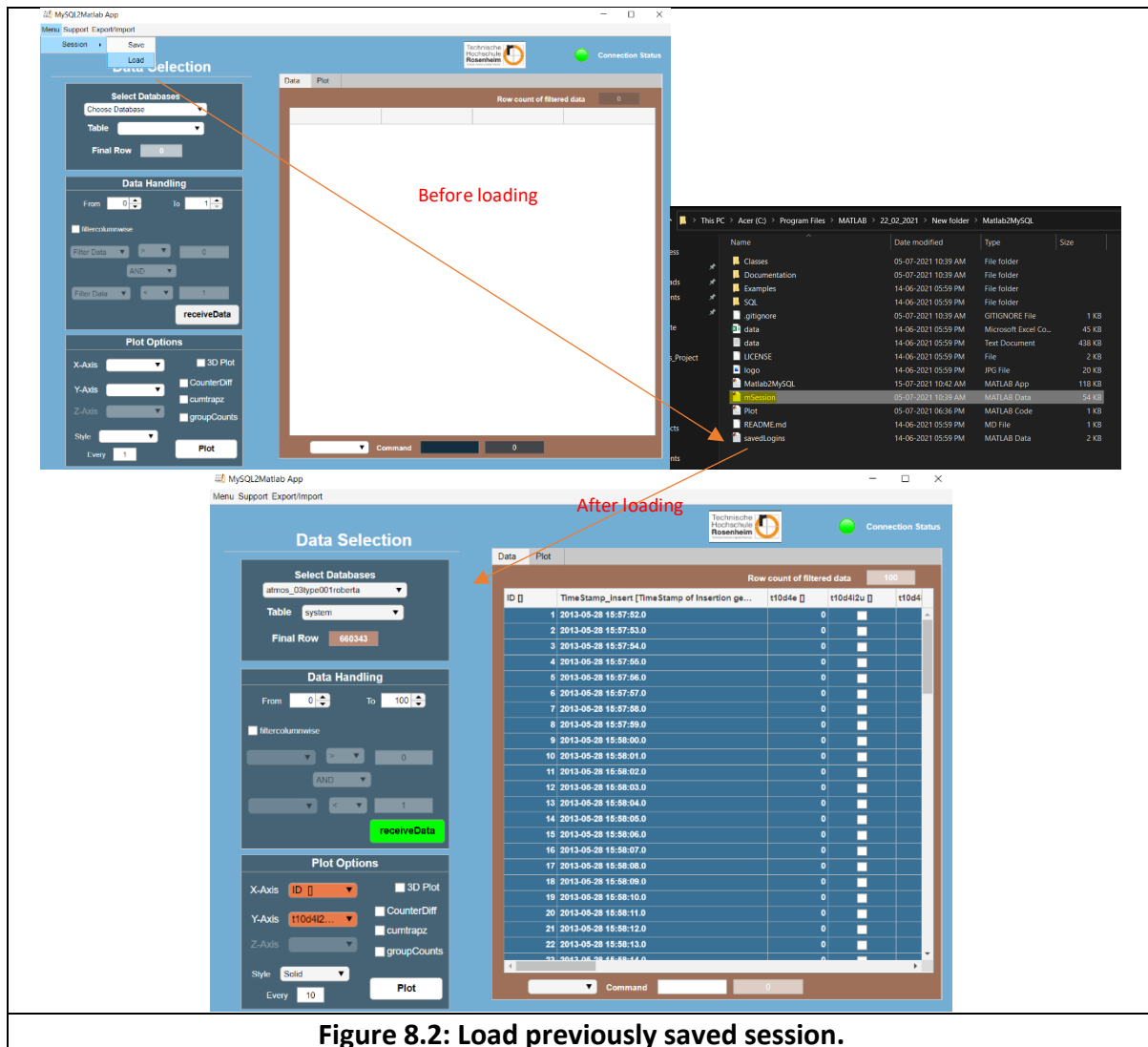


Figure 8.2: Load previously saved session.

9. Export and Import

- It is possible to export UITable data as different file formats like '.csv', '.txt' and '.mat'. In order to do this user must navigate to Export/Import option and select required format in which he/she wants to export. Refer **Figure 9.1**.
- User can import any '.xls' and load it into UITable. to do this user must navigate to **Export/Import** option and select **Import**.

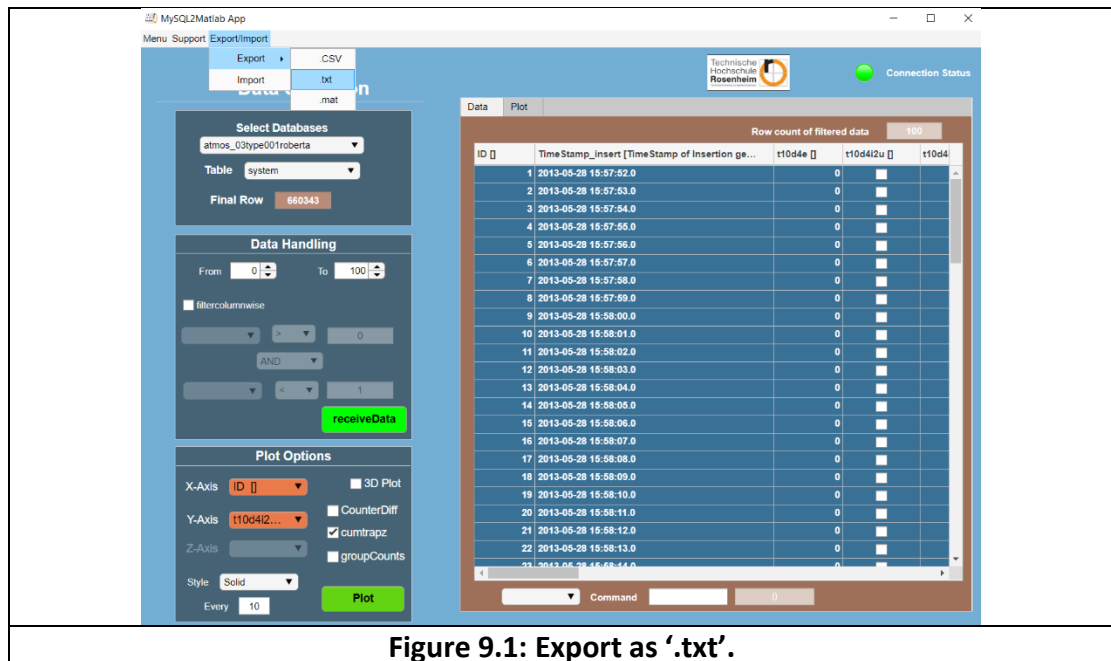
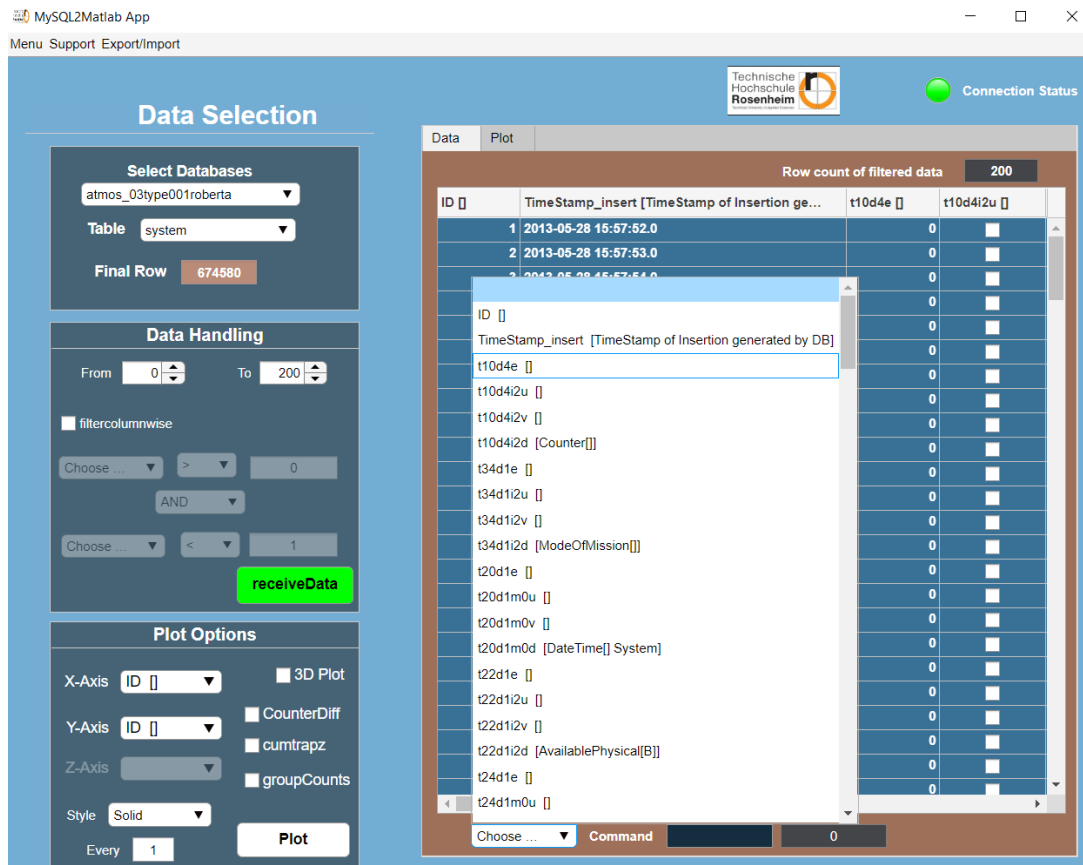


Figure 9.1: Export as '.txt'.

10. Command-Manager:

Command manager has ability to apply MATLAB own functions which comes from the user input as string and manipulate them. User can select the any one of the columns of already retrieved data into UITable which must be manipulated. Below example shows how to select a column and where to enter the command to get required result.

Step 1: Select anyone of the columns from command drop-down as shown in figure.



Step 2: After selecting required column, enter the command or the MATLAB own Function in the Command text edit field as shown below and click enter. Result will be displayed adjacent to the command.

In this example Maximum value of Column “**t10d4i2d**” is displayed by using the function called ‘**max**’ which is **5690**.

MySQL2Matlab App

Menu Support Export/Import

Technische Hochschule Rosenheim

Connection Status

Data Selection

Select Databases
atmos_03type001roberta

Table system

Final Row 674580

Data Handling

From 0 To 200

☐ filtercolumnwise

Choose ... > 0

AND

Choose ... < 1

receiveData

Plot Options

X-Axis ID ☐ 3D Plot

Y-Axis ID ☐ CounterDiff

Z-Axis ☐ cumtrapz

Style Solid ☐ groupCounts

Every 1

Plot

ID	TimeStamp_insert [TimeStamp of Insertion ge...	t10d4e	t10d4i2u
1	2013-05-28 15:57:52.0	0	
2	2013-05-28 15:57:53.0	0	
3	2013-05-28 15:57:54.0	0	
4	2013-05-28 15:57:55.0	0	
5	2013-05-28 15:57:56.0	0	
6	2013-05-28 15:57:57.0	0	
7	2013-05-28 15:57:58.0	0	
8	2013-05-28 15:57:59.0	0	
9	2013-05-28 15:58:00.0	0	
10	2013-05-28 15:58:01.0	0	
11	2013-05-28 15:58:02.0	0	
12	2013-05-28 15:58:03.0	0	
13	2013-05-28 15:58:04.0	0	
14	2013-05-28 15:58:05.0	0	
15	2013-05-28 15:58:06.0	0	
16	2013-05-28 15:58:07.0	0	
17	2013-05-28 15:58:08.0	0	
18	2013-05-28 15:58:09.0	0	
19	2013-05-28 15:58:10.0	0	
20	2013-05-28 15:58:11.0	0	
21	2013-05-28 15:58:12.0	0	
22	2013-05-28 15:58:13.0	0	
23	2013-05-28 15:58:14.0	0	
24	2013-05-28 15:58:15.0	0	

Row count of filtered data 200

t10d4i2d ... Command max 5690

11. App Testing Framework

To ensure ongoing quality of the software and make sure further development can go on, User Interface tests are created using App Testing Framework from Mathworks^[4]. Tests are provided and evaluated for each feature of the UI with 4 different Matlab code files. Below **Figure 10.1** shows m-Code file written for testing the DatabaseSelection features of the application.

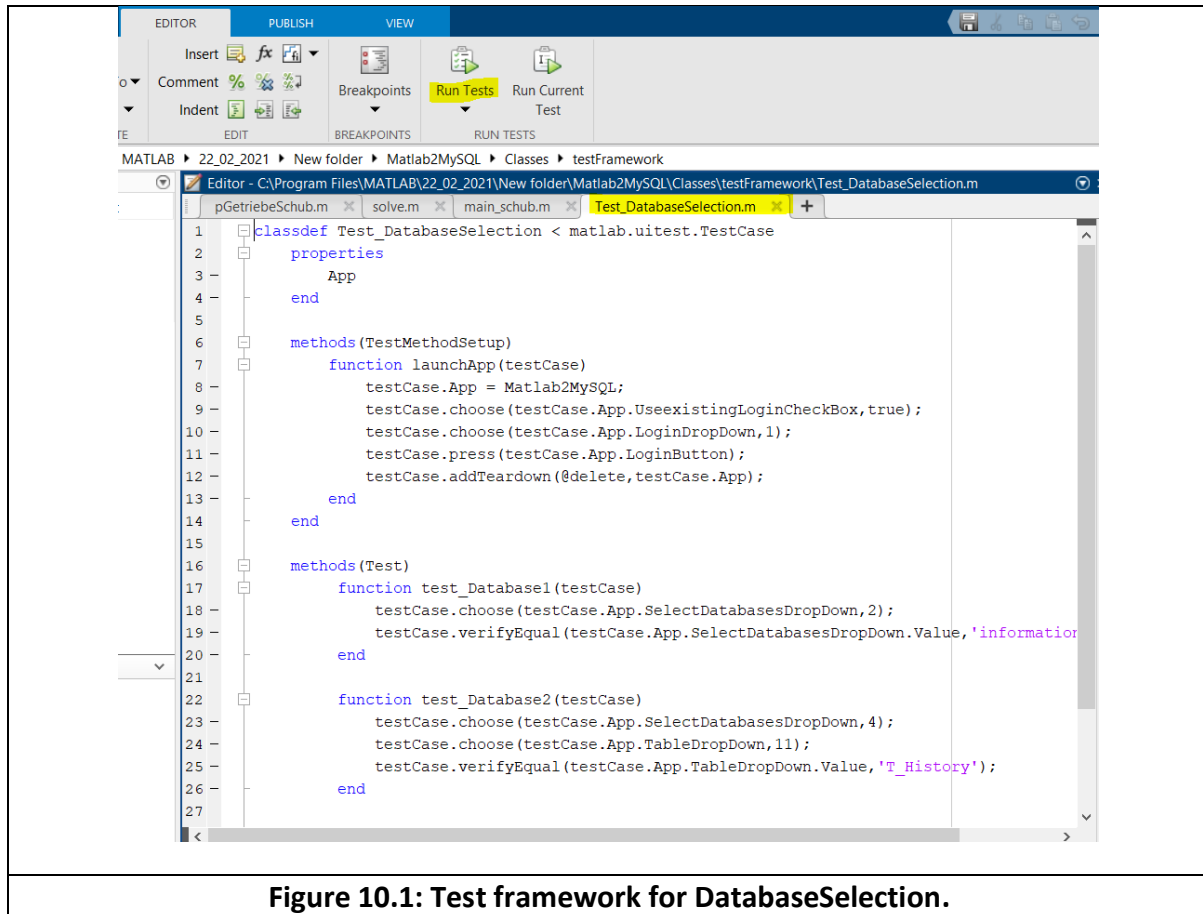


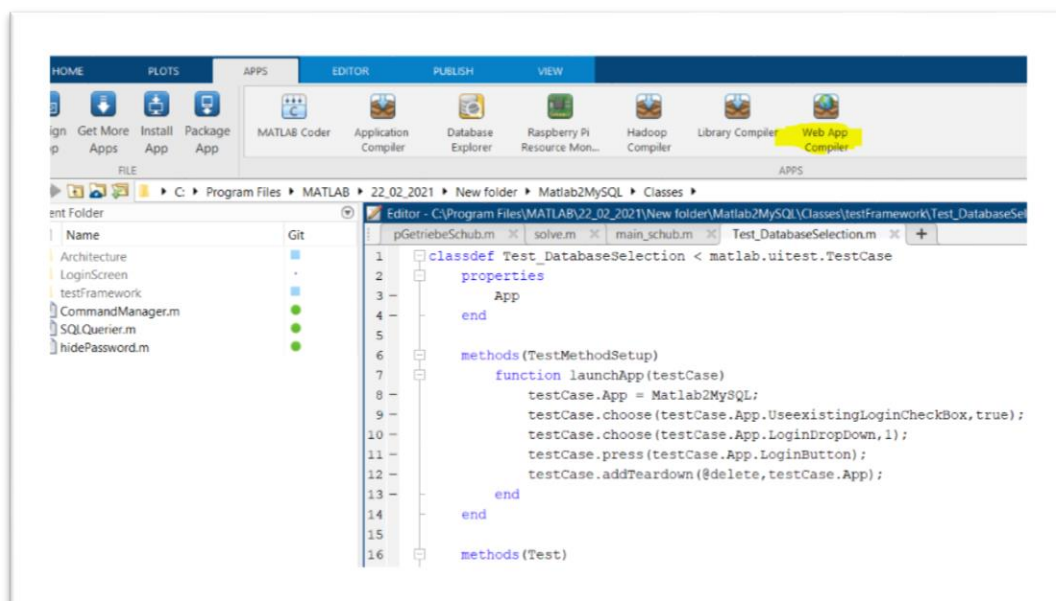
Figure 10.1: Test framework for DatabaseSelection.

12. Web Based Application

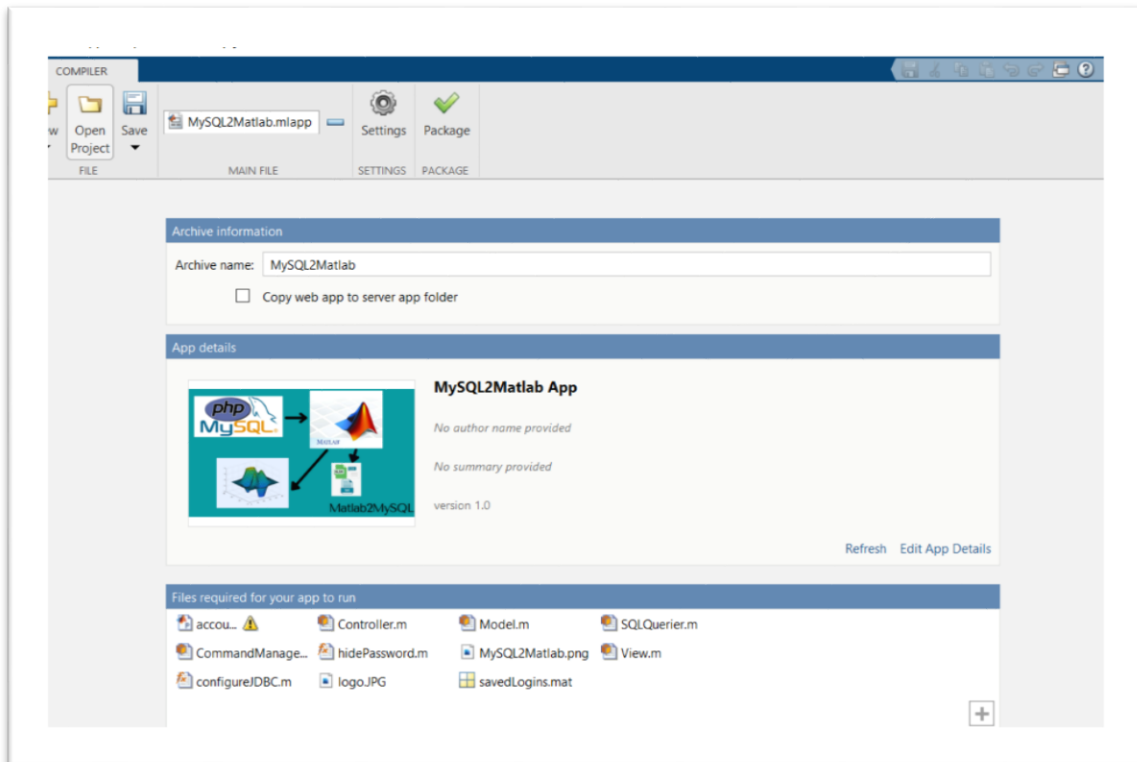
Web apps are MATLAB apps that can run in a web browser. You create an interactive MATLAB app using App Designer, package it using the Web App Compiler, and host it using either the development version of MATLAB Web App Server or the MATLAB Web App Server™ product. Each web app has a unique URL and can be accessed from a web browser using HTTP or HTTPS protocols. The server has a home page listing all available hosted web apps. You share web apps by sharing the unique URL to a web app or the URL to the home page of the server^[5]. Suitable web browsers to run web based applications are **Google chrome** and **Microsoft Edge**.

MYSQL2MATLAB application also made it to run in a web browser by generating '.ctf' format and deployed it to MATLAB web app server. Below figures explains the steps involved in creation of web-based application.

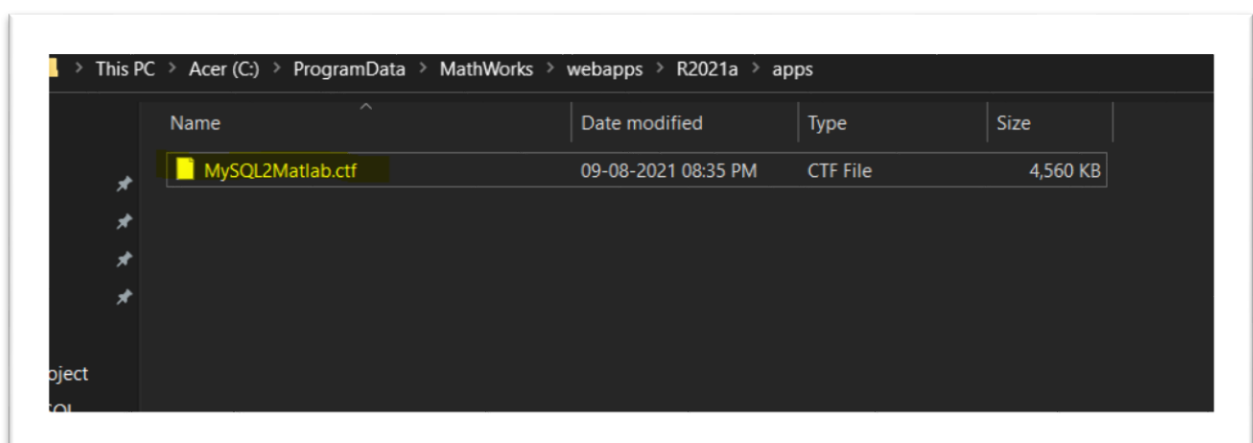
Step 1: Click on Web app Compiler to package the App.



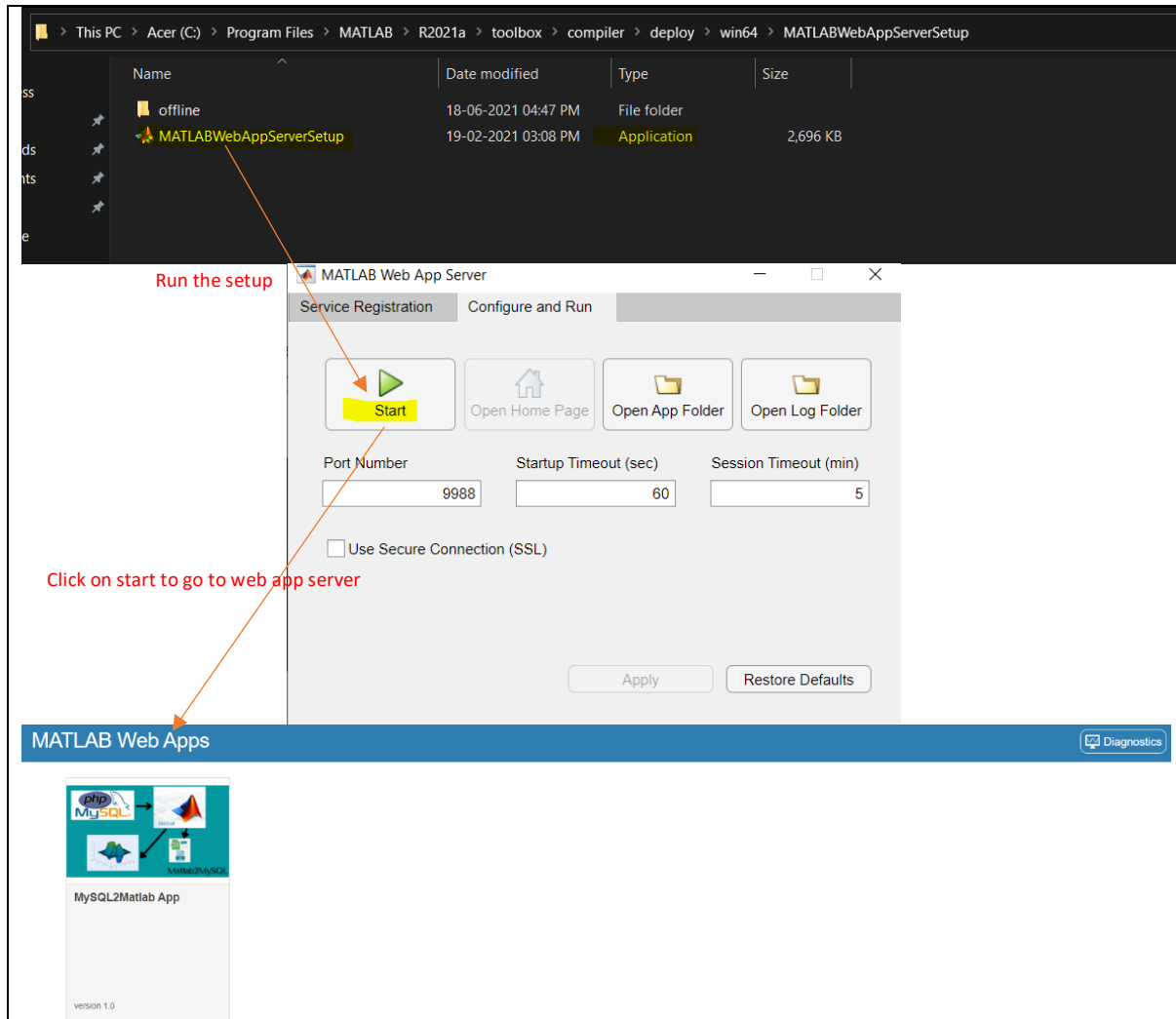
Step 2: Click on Add app file to load MATLAB app file, after loading app file, matlab automatically detects supporting files which are required to run the app as shown in below figure. Click on Package to generate ‘.ctf’ file.



Step 3: Generated ‘.ctf’ has to be deployed in the following path:
“C:\ProgramData\MathWorks\webapps\R2021a\apps”

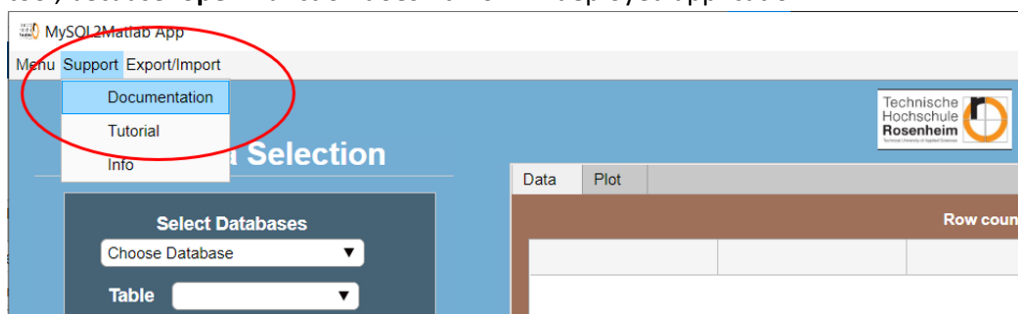


Step 4: After deployment of **'.ctf'** in webapps folder, user needs to set up and run MATLAB WebAppServer application. After successful run user can run the app in Web based environment. Refer following figures.

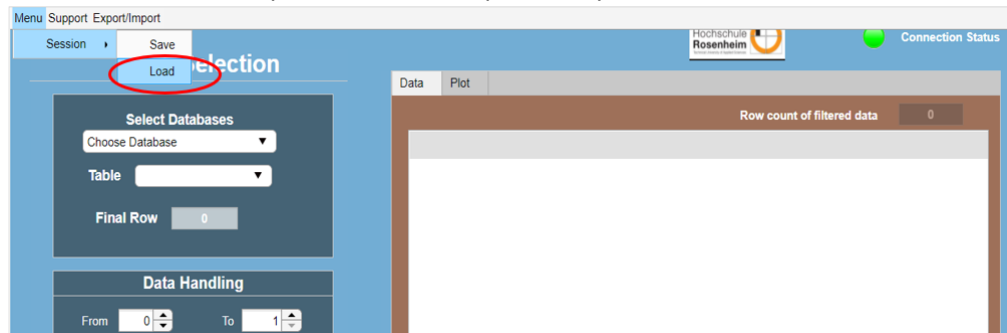


Features which are not available in Web based application:

- **Documentation:** It is not possible to open documentation which is related to MySQL2Matlab tool, because **'open'** function doesn't work in deployed application.



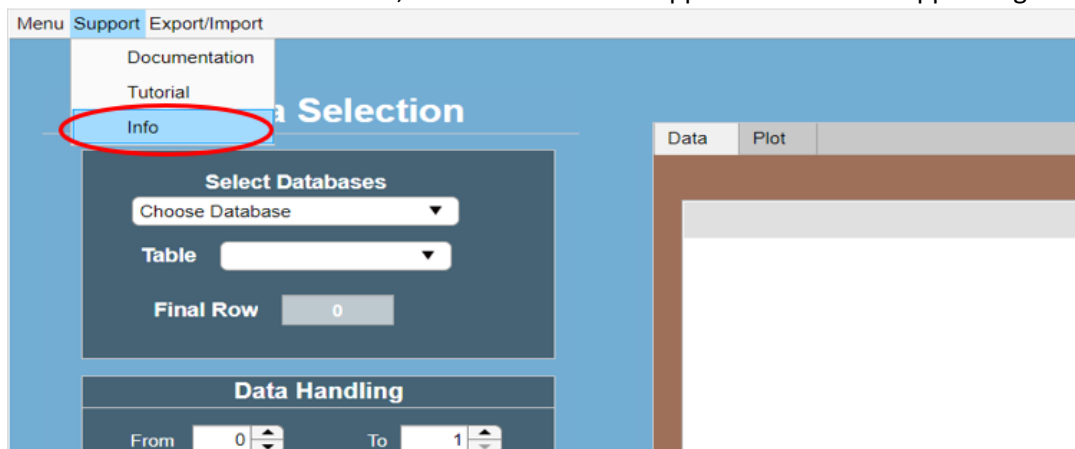
- **Load Session:** It is not possible to load a previously saved session.



- **3D Plot:** Web-based application doesn't support multiwindow app, hence not able to use 3Dplot option.

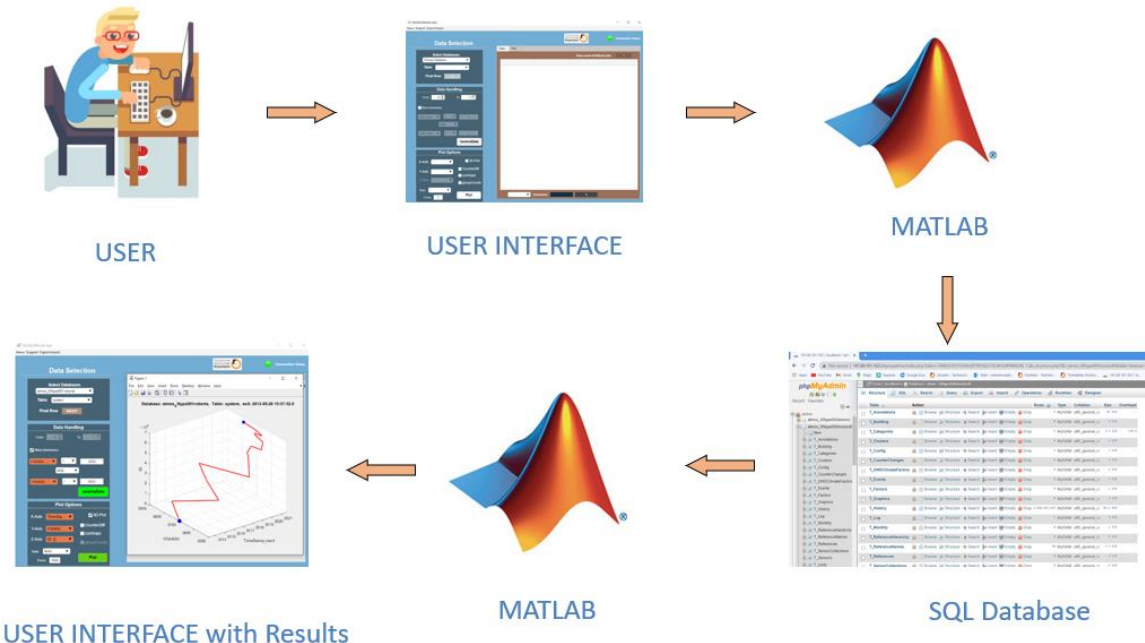


- **Info:** Info feature is not available, because Web based application doesn't support mgsbox.



13. Project Summary and Future scope:

The project was successfully completed with all the tasks running properly. The interface was made user friendly as much as possible. Following depiction shows the working of the model.



The program can be modified further in many aspects.

Future scope:

- **Google Earth Simulation:** Google earth simulation can be implemented with longitude and latitude data.
- **Coordinate Transformation:** User can be able to convert my latitude and longitude information according to a given geodesic transformation.
- **Visualize DWD Cells:** User can be able to plot DWD (Detection and forecast of convective) cells for a given time.
- **Animate Haildefence flight:** As a User I want to be able to run a animation which shows the flight path and the DWD Cells for a given time frame.
- **Creation of KML data:** Implementation of KML file generation feature for Google earth simulation.
- Colors and fonts of the components in the GUI can be changed.

14. References

- [1] <https://de.mathworks.com/products/matlab/object-oriented-programming.html>

- [2] <https://en.wikipedia.org/wiki/Model-view-controller>

- [3] <https://de.mathworks.com/products/matlab/app-designer.html>

- [4] <https://de.mathworks.com/help/matlab/ref/matlab.uitest.testcase-class.html>

- [5] <https://de.mathworks.com/help/compiler/web-apps.html>

- [6] <https://github-wiki-see.page/m/sundaynwoye/todo-list/wiki/How-it-works-%28technically%29>

- [7] <https://www.codecademy.com/articles/mvc>

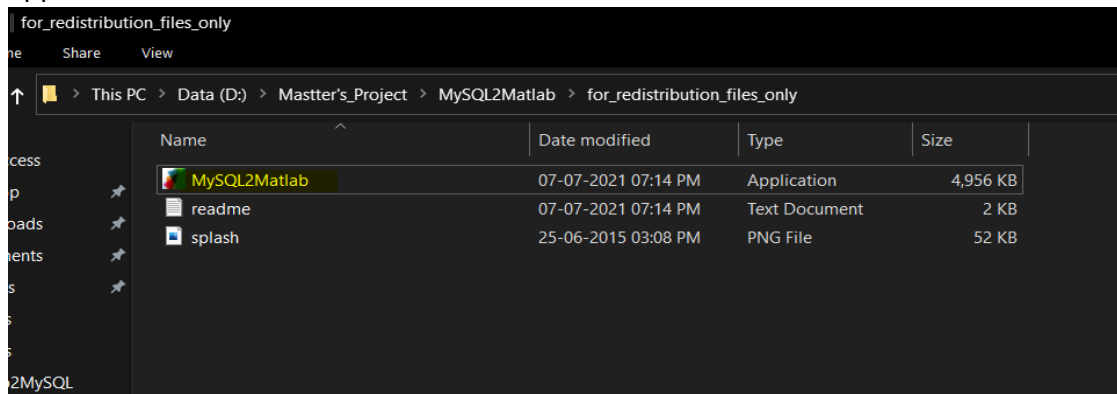
- [8] <https://de.mathworks.com/help/database/ug/databaseexplorer-app.html>

15. Appendix 1: User Manual version 1

For Installation:

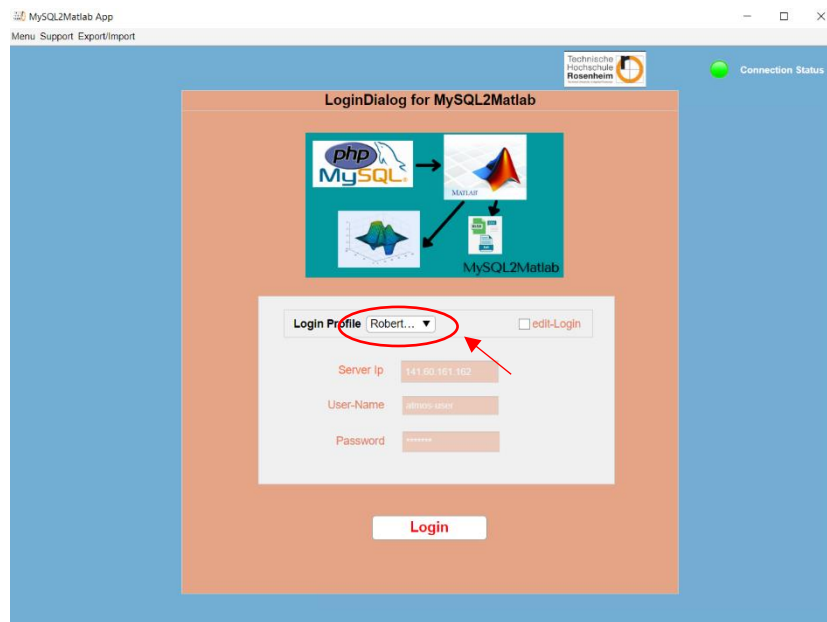
You should carry out the following steps in-order to be able to use the program as a standalone application:

- Please Unzip the zip file in a subdirectory of your choice, for example "MySQL2Matlab"
- The program is only functional if you have also downloaded and installed the MATLAB Compiler Runtime (MCR) library version 9.10 (R2021a). To download the same use below link.
<http://www.mathworks.com/products/compiler/mcr/index.html>
- Check whether you have a directory C: \ temp. If not, put it on. Otherwise, the program will not be executed.
- After successful unzipping of MySQL2Matlab_01, please select folder "for_redistribution_files_only" and Select MySQL2Matlab (as shown below) application file and **Run-as administrator**.

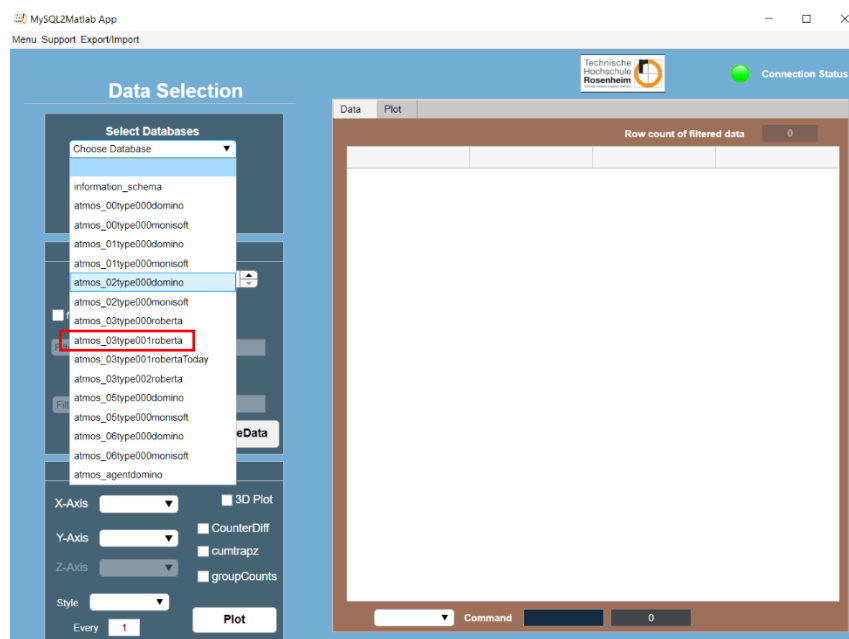


For smooth running of application:

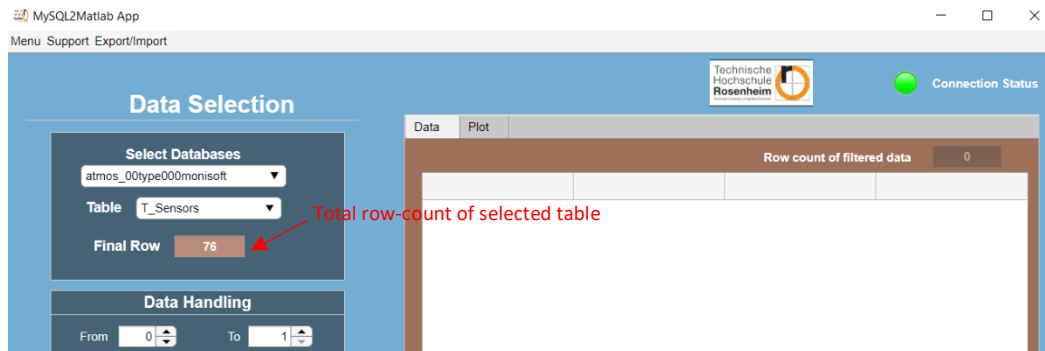
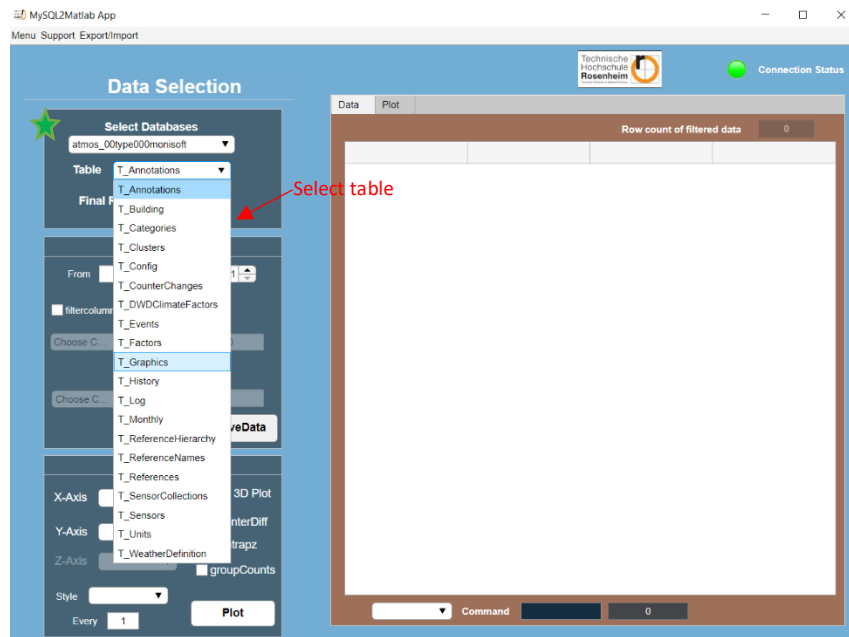
As soon as a user runs the application or .exe file, it asks for log-in option in order to connect to MySQL database. User can select "Use existing Log-in" option and click on Log-in button as shown below.



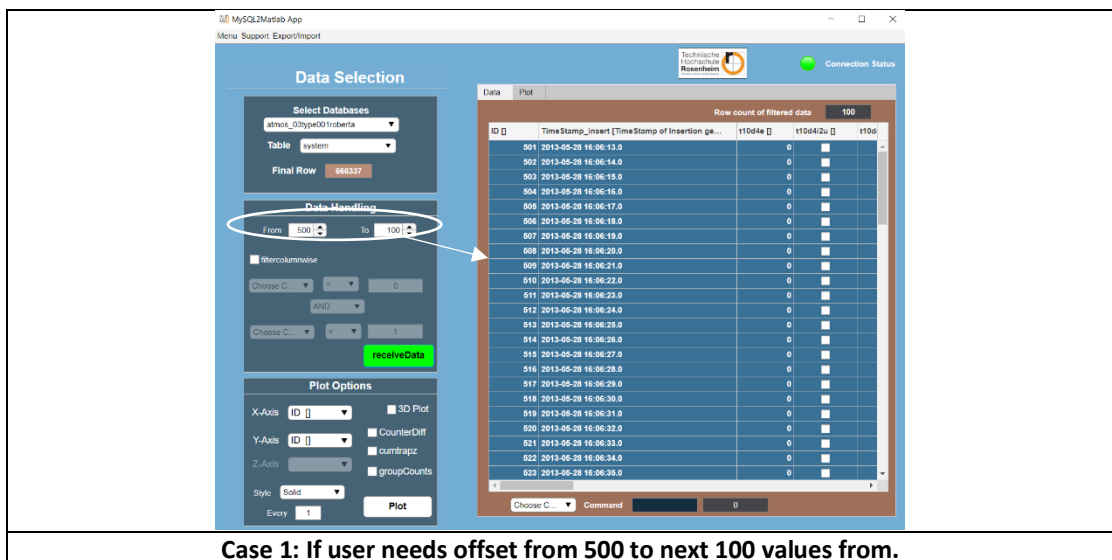
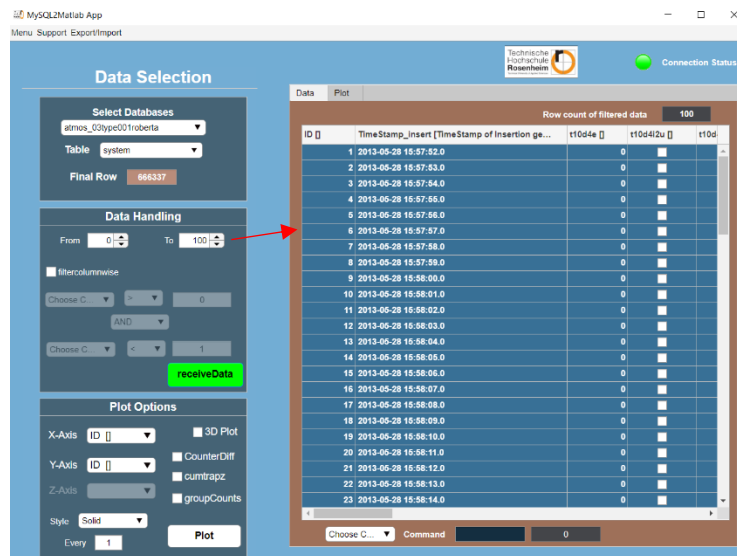
- After the successful Log-in, User can select required Database by using dropdown “Available Databases” dropdown as shown in below figure.



- After selecting database, user can select required Table available in selected databases by using “Table” dropdown. As soon as user selects table, total rows of selected table will be read as “Final Row” as shown in below.

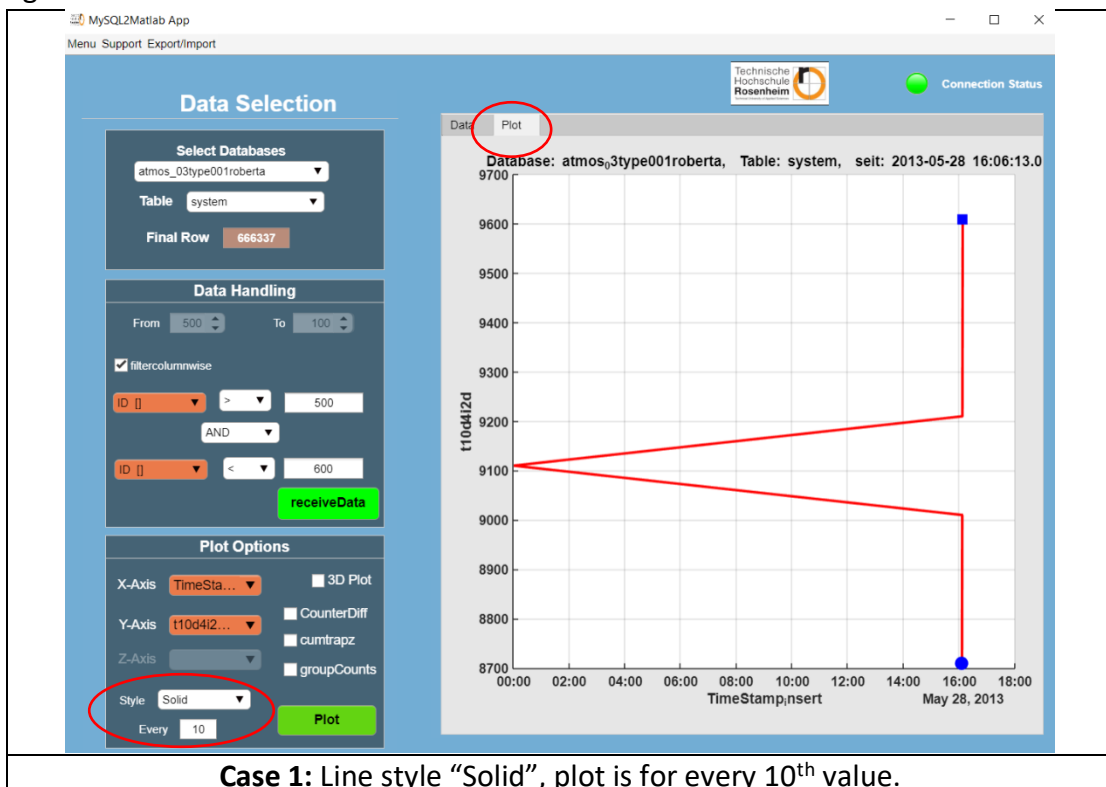


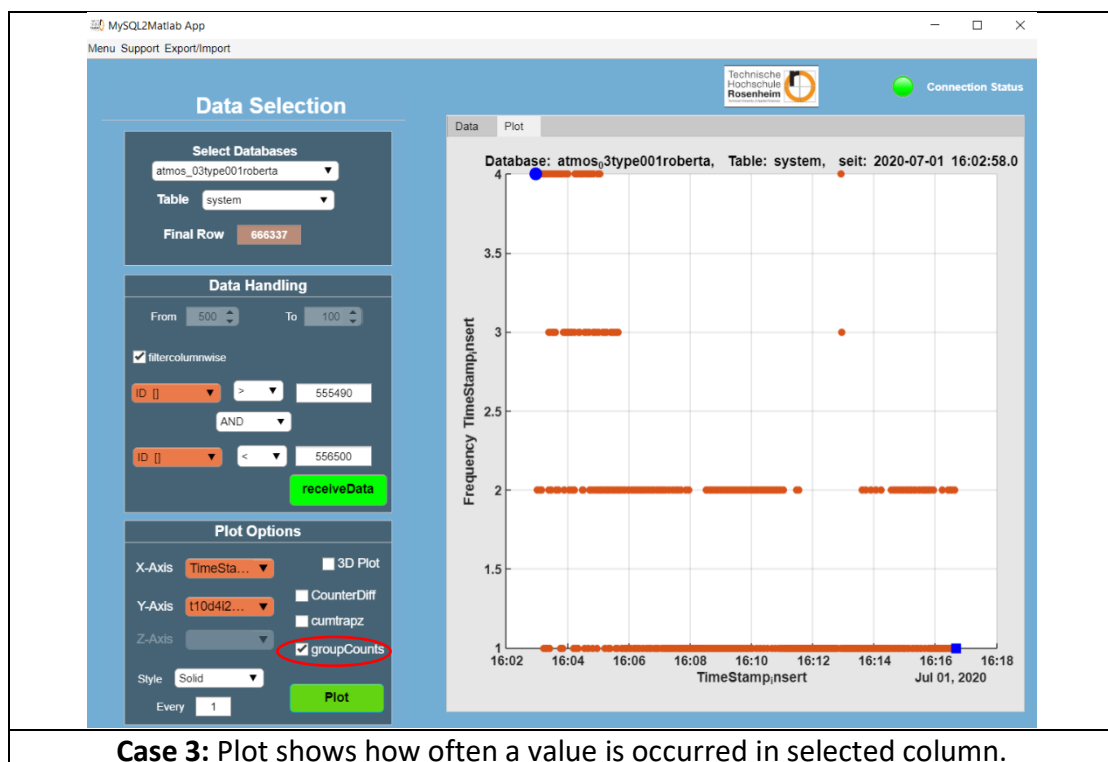
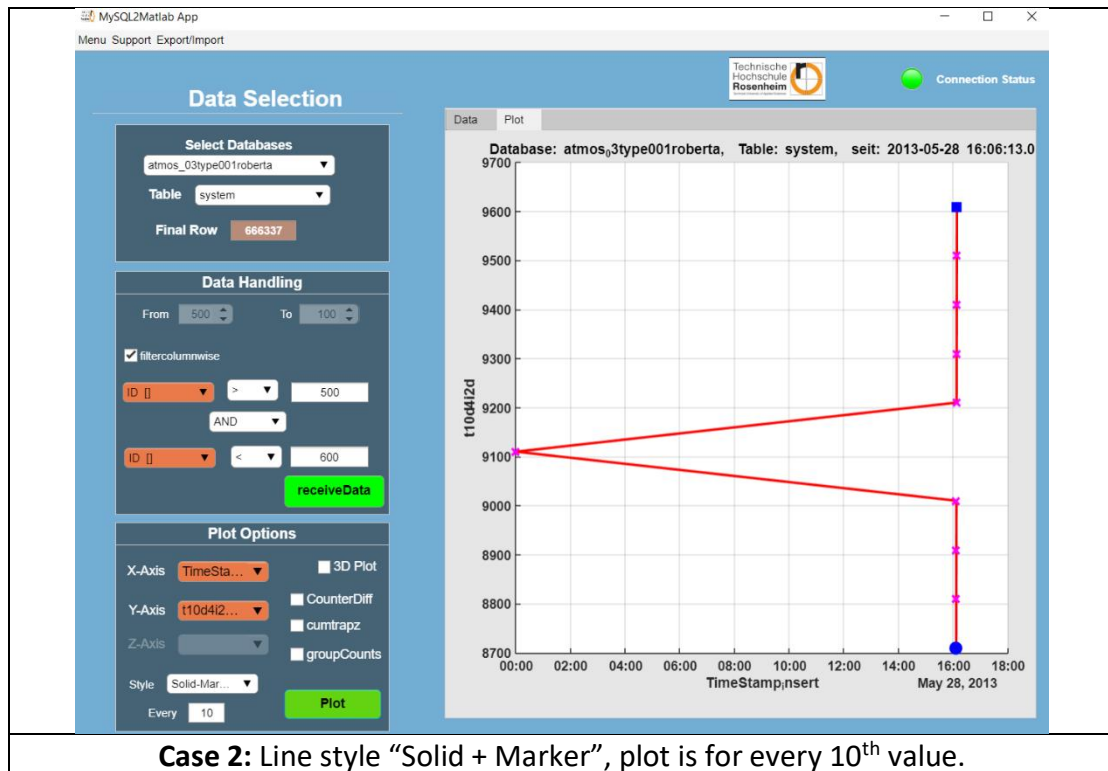
- **Data Loading:** For extracting the selected table, user can specify the limit of table by entering “From” and “To” values. Examples are shown below.



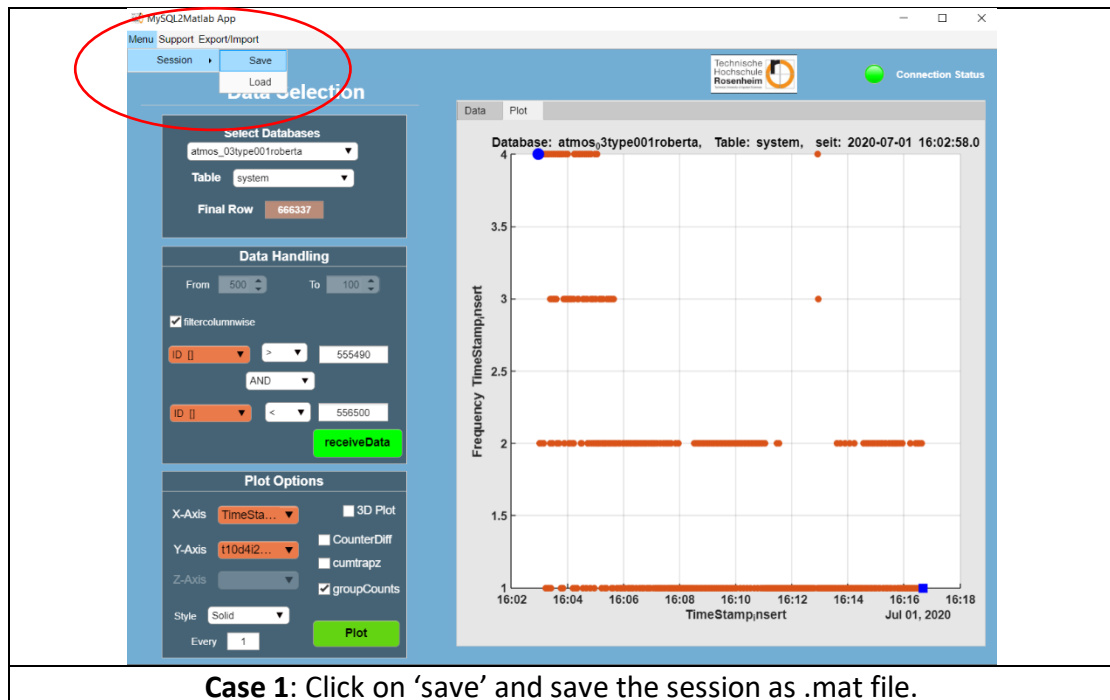


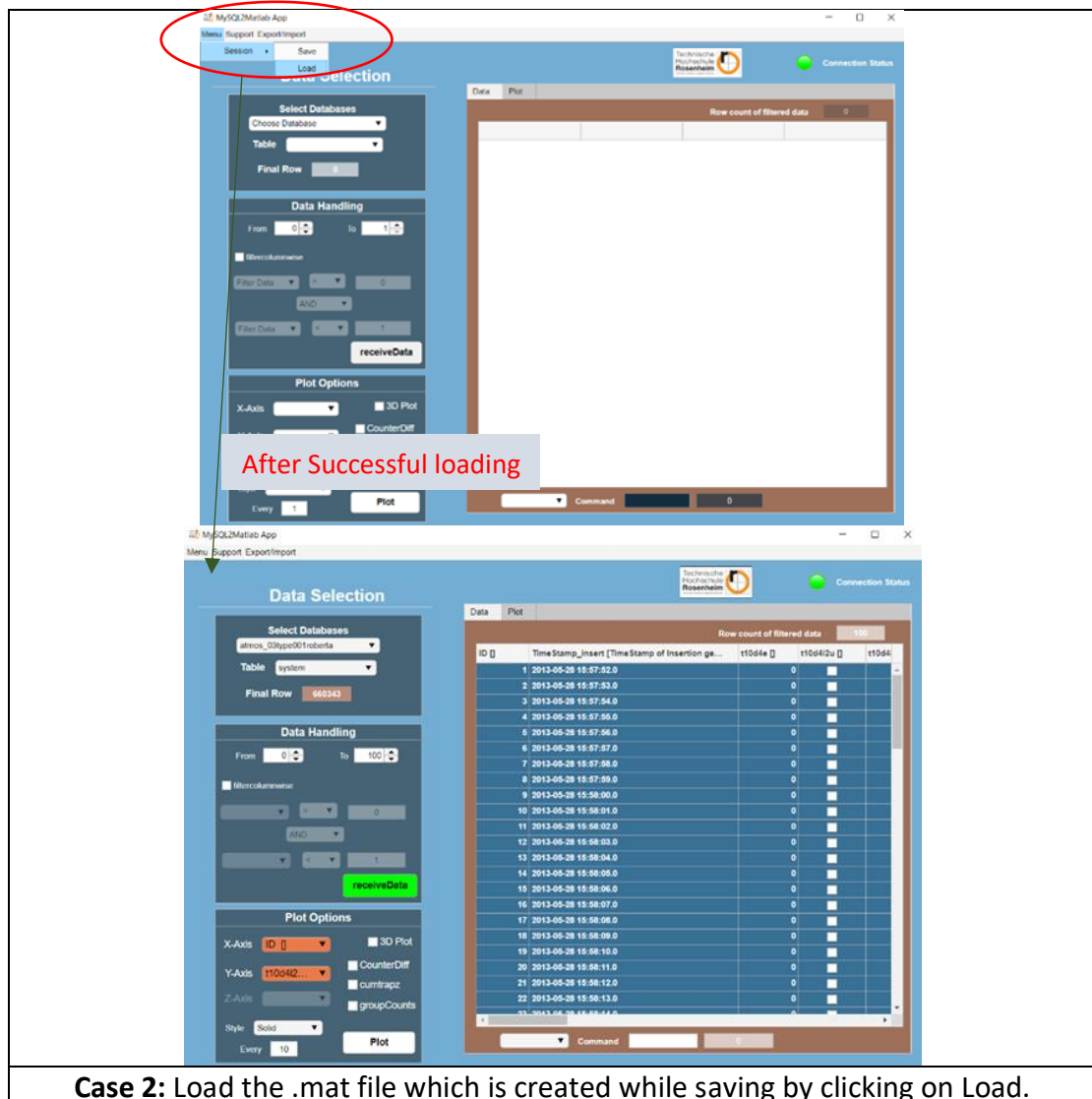
- **Plot options:** User can select X-axis and Y-axis using available dropdown. It is also possible to use some CounterDiff or cumtrapz by selecting boxes. Please see below figures for more details.





- User can save and load the session at any point of time by clicking save and load options available in “Menu”. Please refer below figures.





Note: After loading Background colour of “Plot” button is not turned to green, therefore user needs to click on “Plot” button to get corresponding graph.