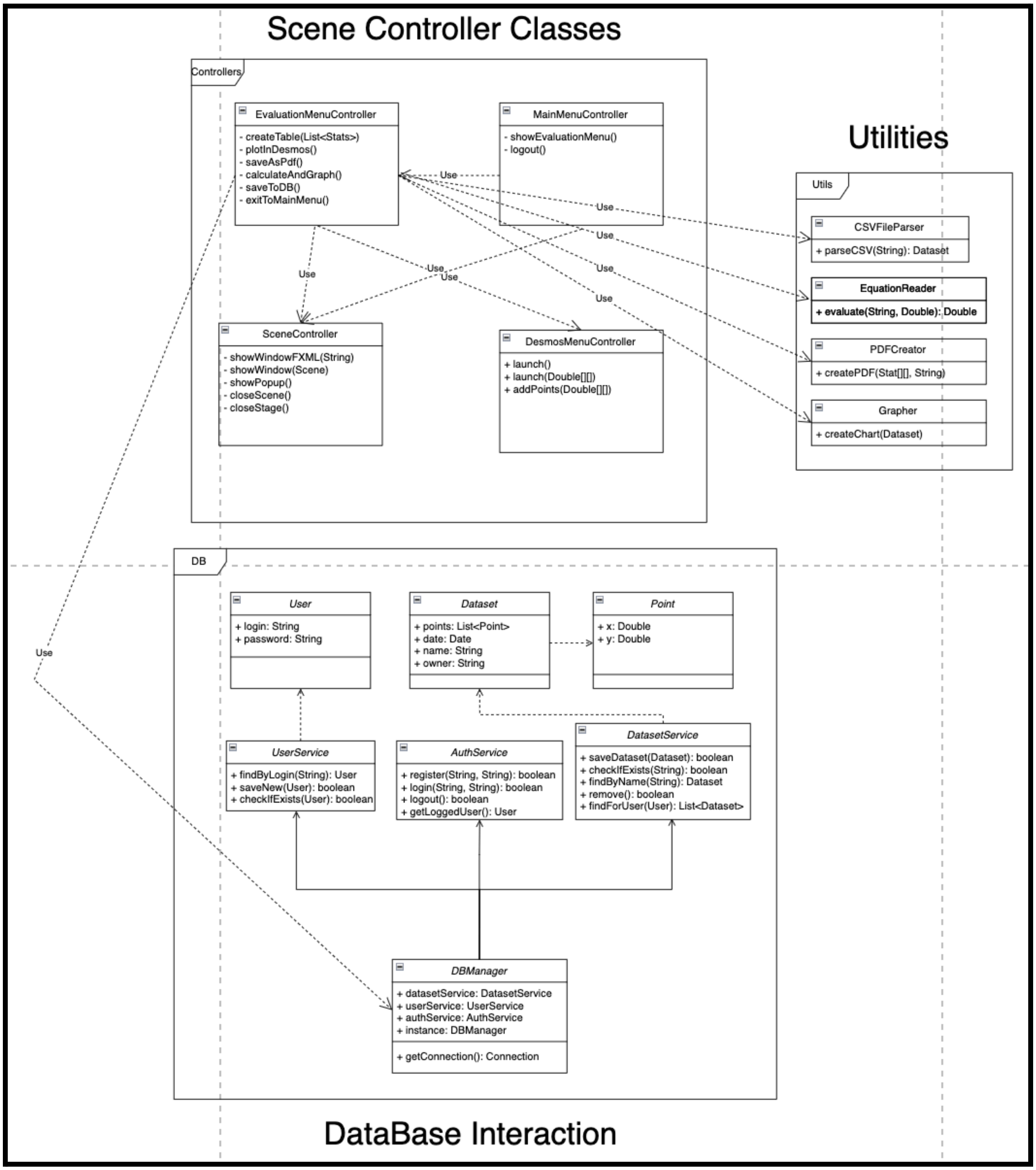
StatLab Desktop Application

Table of Contents

Product overview

The product is a desktop application and consists of several components: a class of **static methods representing statistical functions for statistical evaluation** of user-uploaded data sets, a **formula ‘parser’** which is used to evaluate user-inputted theoretical argument dependency expressions (inputted by the user), **an interactive graphing tool** for user data, which includes a Desmos Graphing Calculator API, an API for the user to automatically generate and save PDF files with the plotted charts and calculated statistical values, as well as a **Lab Data Base** portal to interact with a connected to a database and log / edit of all the user-uploaded experimental data sets for convenience.

UML Diagram


Scene Controller Classes
Controllers
EvaluationMenuController
- createTable(List<Stats>)
- plotInDesmos()
- saveAsPdf()
- calculateAndGraph()
- saveToDB()
- exitToMainMenu()
MainMenuController
- showEvaluationMenu()
- logout()
Use
Use
Use
Use
Use
Use
Use
Use
SceneController
- showWindowFXML(String)
- showWindow(Scene)
- showPopup()
- closeScene()
- closeStage()
DesmosMenuController
+ launch()
+ launch(Double[][])
+ addPoints(Double[][])
Utilities
Utils
CSVFileParser
+ parseCSV(String): Dataset
EquationReader
+ evaluate(String, Double): Double
PDFCreator
+ createPDF(Stat[][], String)
Grapher
+ createChart(Dataset)
DB
User
+ login: String
+ password: String
Dataset
+ points: List<Point>
+ date: Date
+ name: String
+ owner: String
Point
+ x: Double
+ y: Double
Use
UserService
+ findByLogin(String): User
+ saveNew(User): boolean
+ checkIfExists(User): boolean
AuthService
+ register(String, String): boolean
+ login(String, String): boolean
+ logout(): boolean
+ getLoggedUser(): User
DatasetService
+ saveDataset(Dataset): boolean
+ checkIfExists(String): boolean
+ findByName(String): Dataset
+ remove(): boolean
+ findForUser(User): List<Dataset>
DBManager
+ datasetService: DatasetService
+ userService: UserService
+ authService: AuthService
+ instance: DBManager
+ getConnection(): Connection
DataBase Interaction

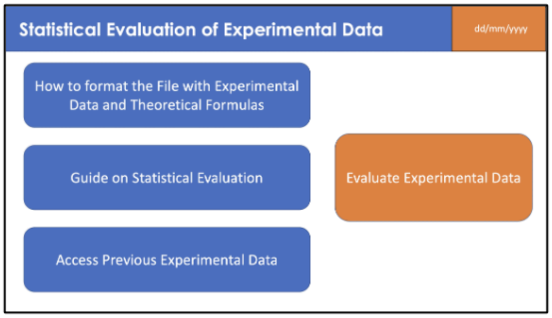
GUI Prototype - refined after second interview with client.

The prototype was refined after the 2nd interview in order to include a brighter colour palette, as well as a separate window with a web view of Desmos Graphing Calculator.

1) Login / Sign-Up Pop-Up window

- Standard Login and Register window

2) Main Menu window



Statistical Evaluation of Experimental Data
dd/mm/yyyy
How to format the File with Experimental Data and Theoretical Formulas
Guide on Statistical Evaluation
Access Previous Experimental Data
Evaluate Experimental Data

Will guide the user to windows with additional text information

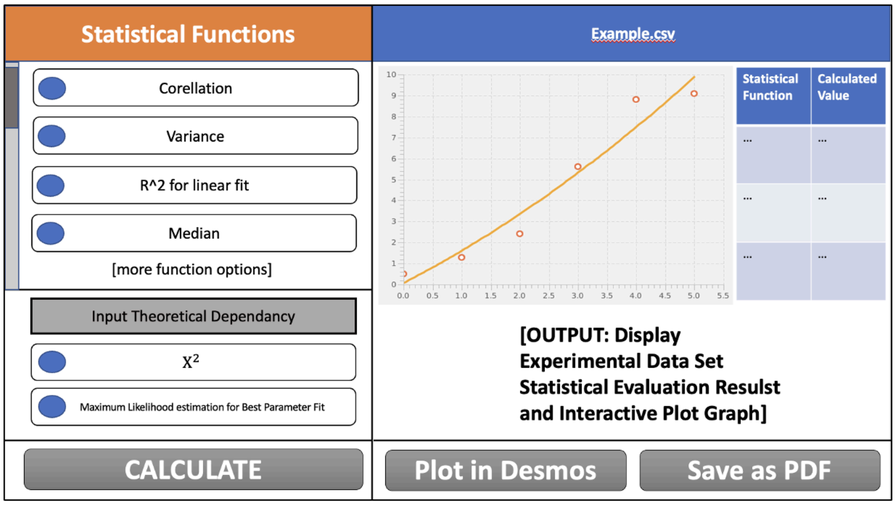
This button will take the user to the "Upload Exp. Data" file drop-box and, consequently, to the "Evaluate Exp. Data" window

This button will guide the user to a window for interacting with the Action Log Data Base via the application

Experimental Dataset files are uploaded as 2-dimensional arrays in csv formats.

3) "Evaluate Experimental Data" window:

Display File Name



Statistical Function	Calculated Value
...	...
...	...
...	...

[OUTPUT: Display Experimental Data Set Statistical Evaluation Resulst and Interactive Plot Graph]

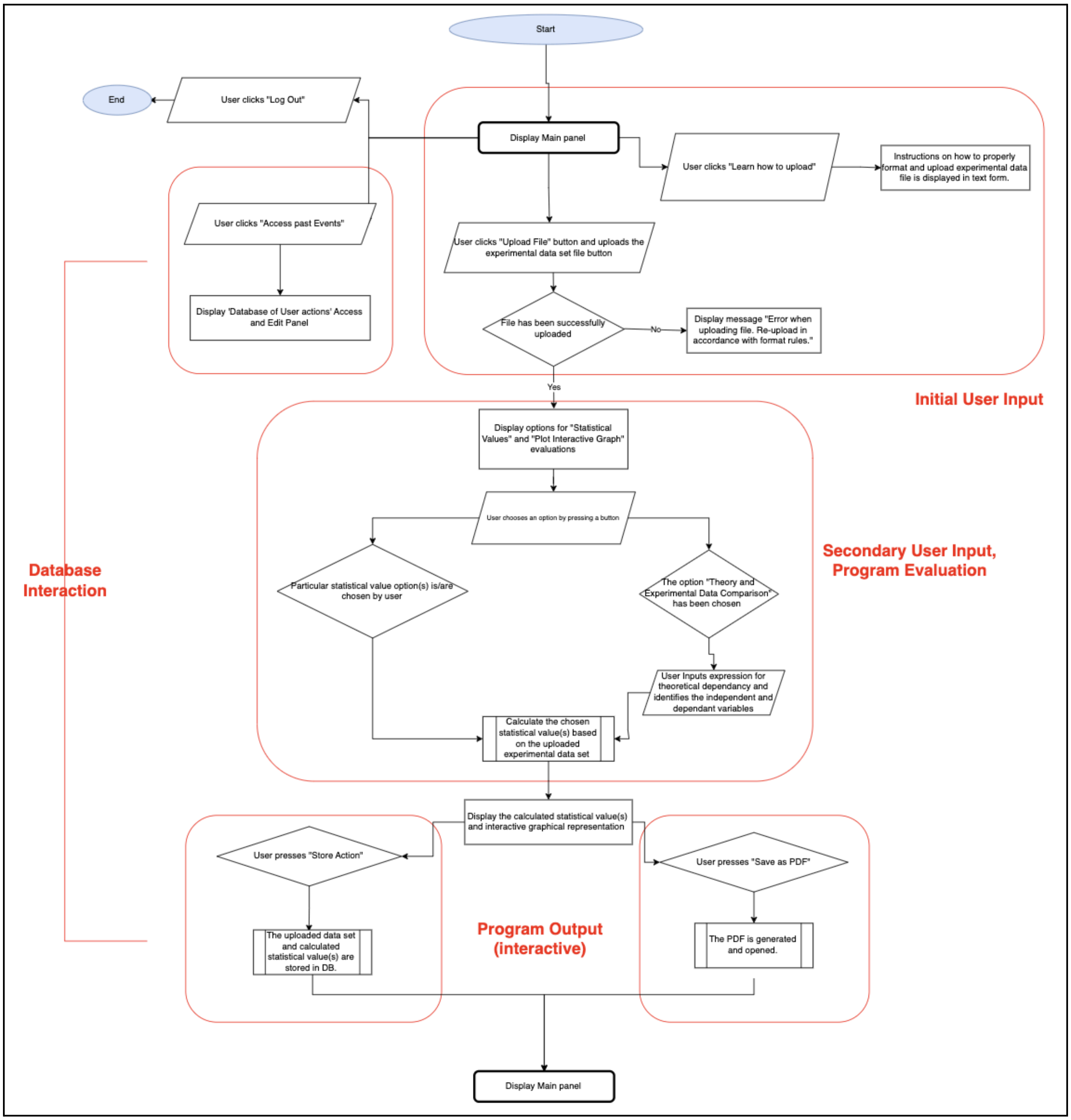
The user will check all the needed boxes for statistical evaluation options

Input theoretical dependency expression by typing or via scanning a hand-written expression via **OCR**

The User will press this button upon ticking all the needed boxes

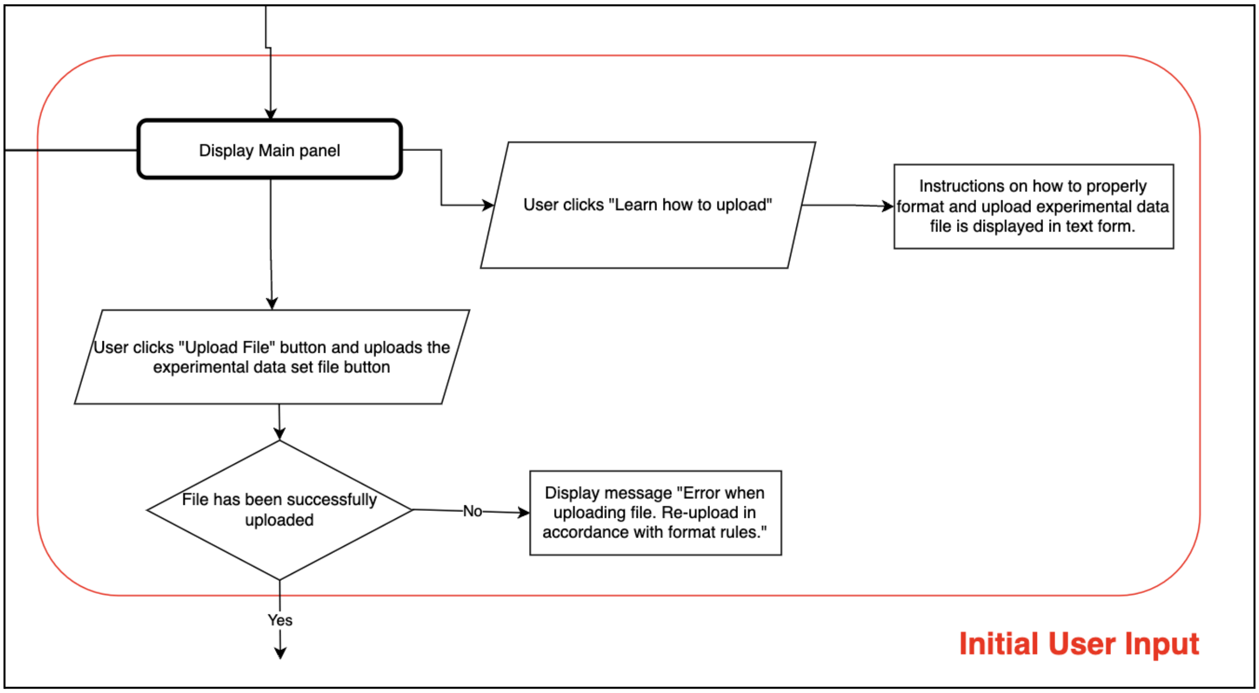
The User can press these buttons only after the desktop calculated the statistical functions and and plotted the chart; this will take the user to the next window.

User Interaction Flowchart

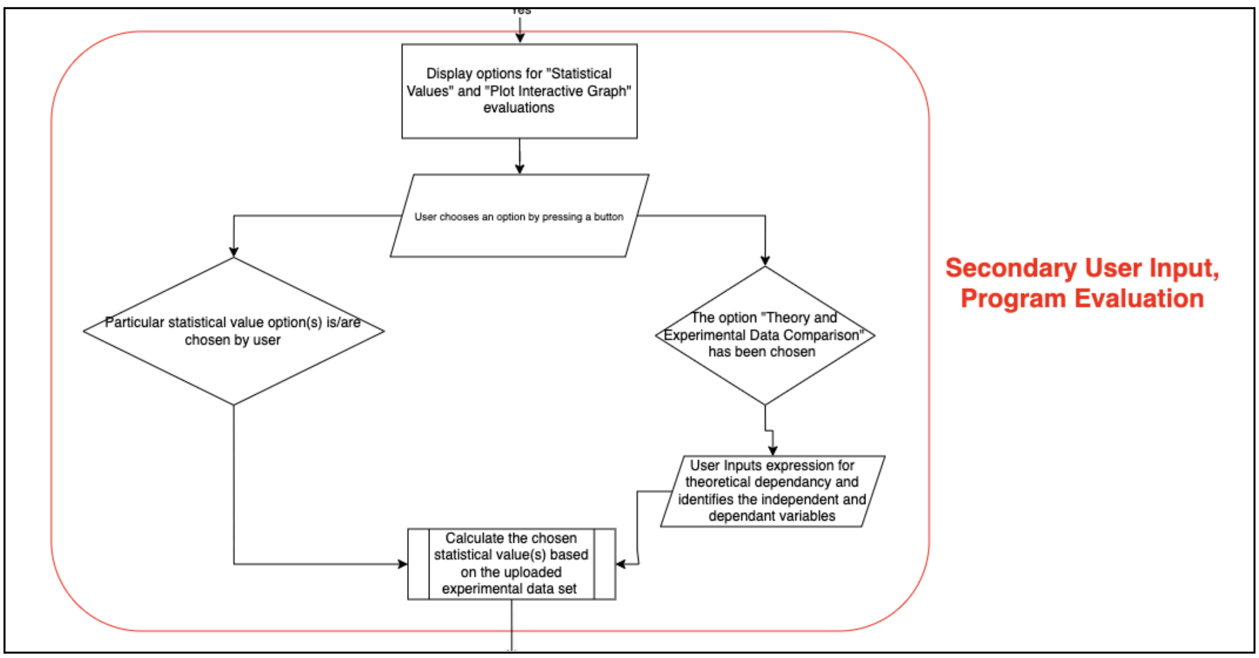


Start
End
User clicks "Log Out"
Display Main panel
User clicks "Learn how to upload"
Instructions on how to properly format and upload experimental data file is displayed in text form.
User clicks "Access past Events"
Display 'Database of User actions' Access and Edit Panel
User clicks "Upload File" button and uploads the experimental data set file button
File has been successfully uploaded
No
Display message "Error when uploading file. Re-upload in accordance with format rules."
Yes
Initial User Input
Display options for "Statistical Values" and "Plot Interactive Graph" evaluations
User chooses an option by pressing a button
Particular statistical value option(s) is/are chosen by user
The option "Theory and Experimental Data Comparison" has been chosen
User Inputs expression for theoretical dependancy and identifies the independent and dependant variables
Calculate the chosen statistical value(s) based on the uploaded experimental data set
Secondary User Input, Program Evaluation
Database Interaction
Display the calculated statistical value(s) and interactive graphical representation
User presses "Store Action"
The uploaded data set and calculated statistical value(s) are stored in DB.
User presses "Save as PDF"
The PDF is generated and opened.
Program Output (interactive)
Display Main panel

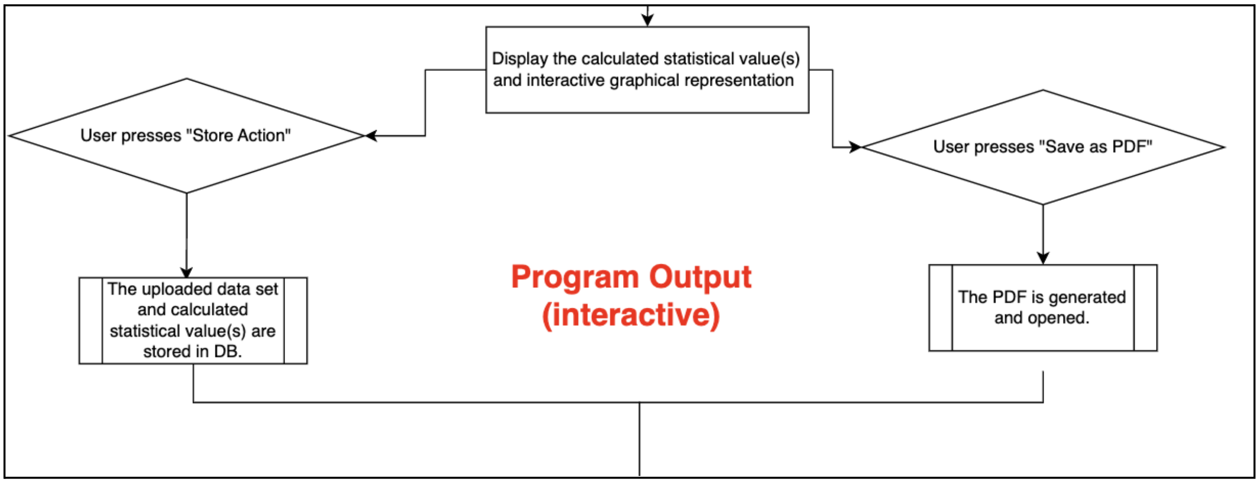
1) Initial User Input


Display Main panel
User clicks "Learn how to upload"
Instructions on how to properly format and upload experimental data file is displayed in text form.
User clicks "Upload File" button and uploads the experimental data set file button
File has been successfully uploaded
No
Display message "Error when uploading file. Re-upload in accordance with format rules."
Yes
Initial User Input

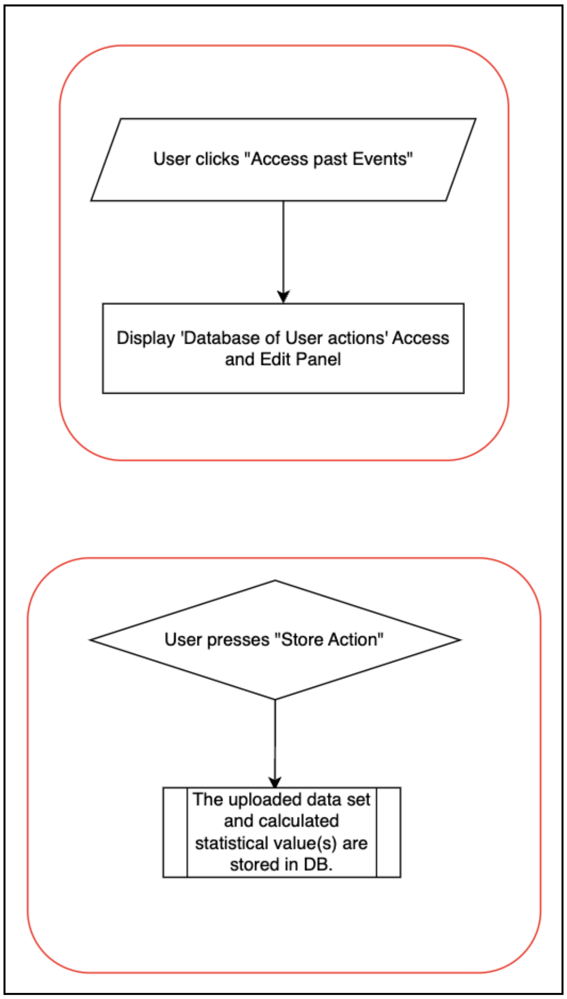
2) Secondary User Input


Yes
Display options for "Statistical Values" and "Plot Interactive Graph" evaluations
User chooses an option by pressing a button
Particular statistical value option(s) is/are chosen by user
The option "Theory and Experimental Data Comparison" has been chosen
User Inputs expression for theoretical dependancy and identifies the independent and dependant variables
Calculate the chosen statistical value(s) based on the uploaded experimental data set
Secondary User Input, Program Evaluation

3) Program Output


Display the calculated statistical value(s) and interactive graphical representation
User presses "Store Action"
User presses "Save as PDF"
The uploaded data set and calculated statistical value(s) are stored in DB.
Program Output (interactive)
The PDF is generated and opened.

4) Interaction with Database


User clicks "Access past Events"
Display 'Database of User actions' Access and Edit Panel
User presses "Store Action"
The uploaded data set and calculated statistical value(s) are stored in DB.

Equation Parser

User inputted expressions must be evaluated to plot them on the graph; I chose Reverse Polish Notation using Binary Calculation tree for this purpose[1] as it guarantees optimisation for time of computation.

1) **Converting Infix Expression to RPN:**

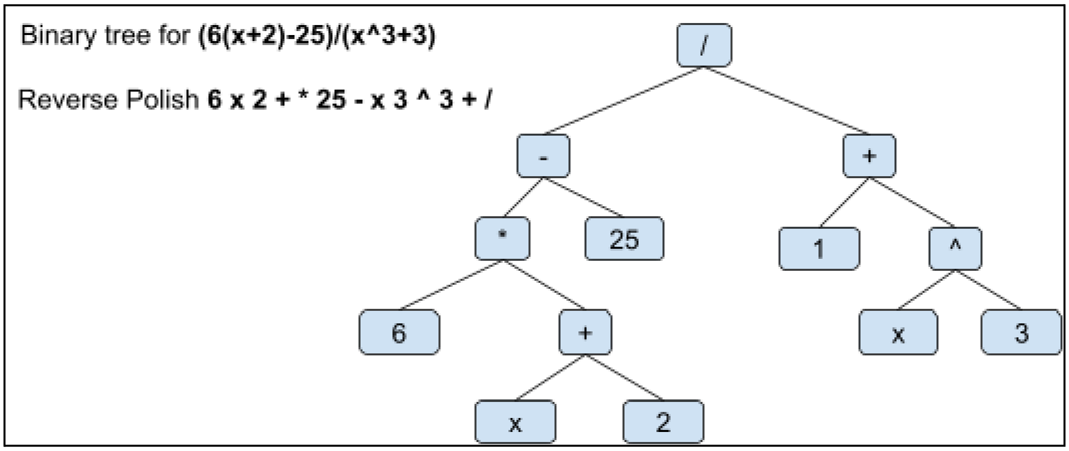
- Use the Shunting Yard Algorithm to convert the infix arithmetic expression to Reverse Polish Notation (RPN).
- This step ensures that the operators follow their respective operands in the RPN format.

2) **Iterate through the RPN expression.**

- If an operand is encountered, create a leaf node with its value.
- If an operator is encountered, create an internal node with the operator as its value.
- Pop the last two elements from the stack and set them as children of the new internal node.
- Push the new node back onto the stack.
- Repeat until the entire expression is processed, leaving the root of the binary tree on the stack.
- Evaluating the Binary Tree:

3) **Recursively evaluate nodes:**

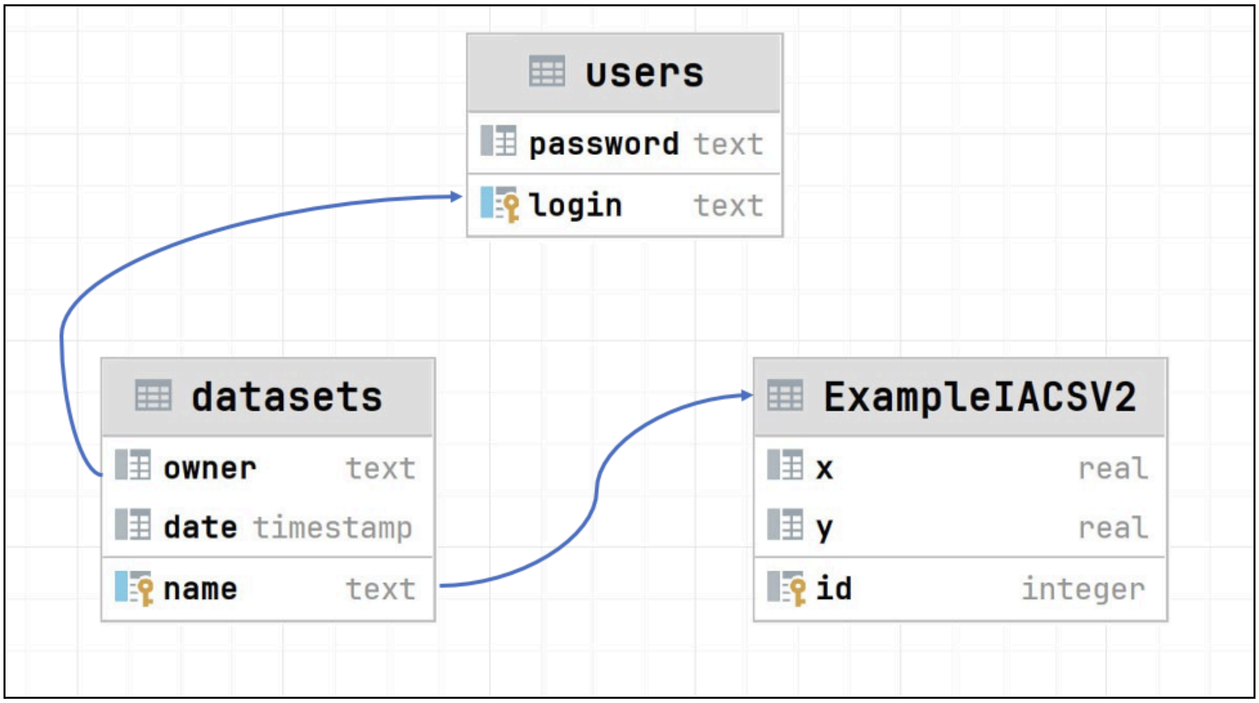
- If the node is an operand, return its value.
- If the node is an operator, evaluate its left and right children recursively.
- Perform the operation indicated by the operator on the results of the left and right subtrees.
- Return the result.


Binary tree for (6(x+2)-25)/(x^3+3)
Reverse Polish 6 x 2 + * 25 - x 3 ^ 3 + /
/
-
+
*
25
1
^
6
+
x
3
x
2

[1] Sooki. “What on Earth Is an Expression Tree?” Medium, February 17, 2023.

DataBase Diagram

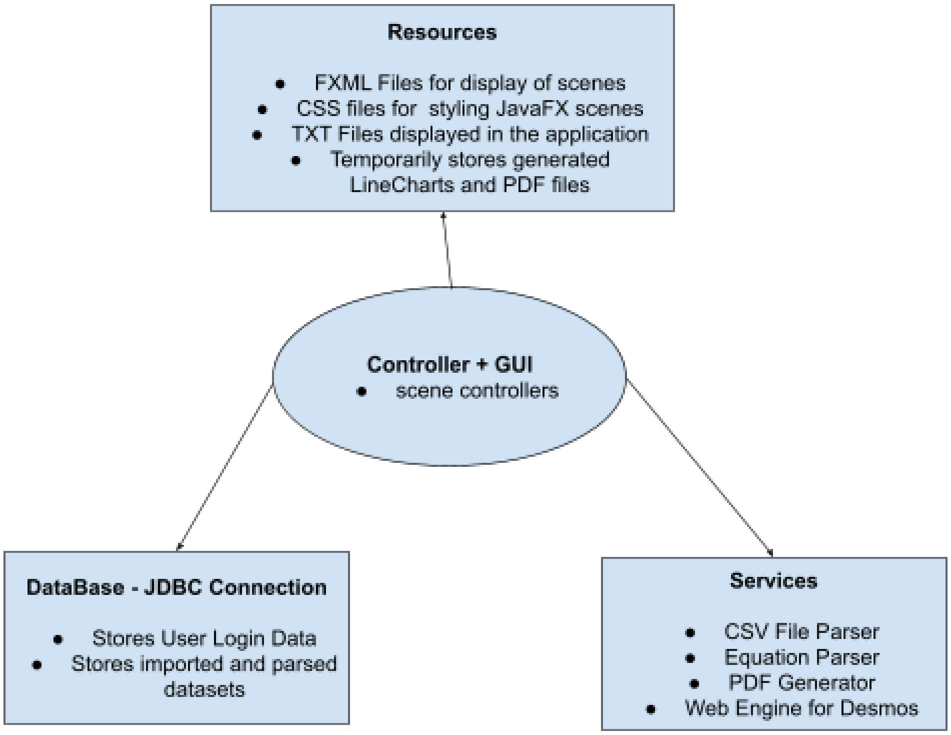
Normalised database: each user in the “users” table has their own “datasets” table



Creating “datasets” table for user using scripts in Java (connection by JDBC)

```
CREATE TABLE IF NOT EXISTS datasets
                (
                     name TEXT PRIMARY KEY,
                     owner TEXT,
                     date TIMESTAMP
                 );
);
```

Structure of interaction between Program Modules

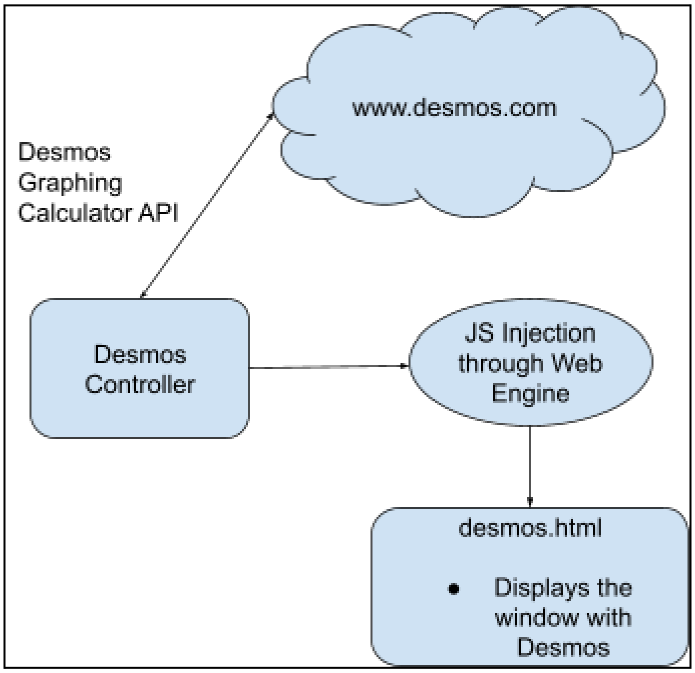

Resources
FXML Files for display of scenes
CSS files for styling JavaFX scenes
TXT Files displayed in the application
Temporarily stores generated LineCharts and PDF files
Controller + GUI
scene controllers
DataBase - JDBC Connection
Stores User Login Data
Stores imported and parsed datasets
Services
CSV File Parser
Equation Parser
PDF Generator
Web Engine for Desmos

External Libraries and APIs Used

They will be imported in the project via Maven

Name	Purpose
JavaFX	Creating GUI; plotting data with LineChart
iTextPDF	Generating PDF for user; optimal and easy-to-implement library
Maven Apache Commons	Statistical Functions to be applied to the parched data set
Desmos Graphing Calculator API and Web Engine	Displaying user-imported data in Desmos in the application through Web View
SQLite JDBC	Interacting with the SQLite DataBase in Java via scripts

Scheme for Desmos Graphing Calculator API interaction with Desktop App


www.desmos.com
Desmos Graphing Calculator API
Desmos Controller
JS Injection through Web Engine
desmos.html
• Displays the window with Desmos

Chronological development plan

1) Initial Client Interview (1 day)
    - Understand the needs of my client, Mr. B

2) Primary research (1 week)

- Brainstorm potential computer program solutions to the problem
- Analyzing the resources available for the execution of a client's problem solution.
    + What information is available or needs to be acquired?

- Research existing graphing and statistical data analysis computational solutions/tools, such as:
    + Excel
    + R (R Studio)
    + Python (via Jupyter)
- Research existing Interactive Graphing tools, such as:

    - Excel
    - Geogebra

- Research manifold statistical evaluations for experimental data employed in Physics, appropriate for high-school level

- **Project concept development**

3) Prototyping the Design (1.5 weeks)

- Interface prototype with SceneBuilder (main menu, statistical evaluation option selection, DB access panel)
- Software architecture

4) Second Client Interview (1 day)

- Presenting and discussing project concept and prototype
- Discuss usability, security, compatibility, and maintainability constraints of client

5) Software development (6 weeks)

Backend, class Main:

- Implementing statistical evaluations into code in class StatFunctions
- Algorithms for “reading” various forms of user input (csv of experimental data; theoretical dependency expressions, typed in or scanned).
- Connect application to database, algorithms for access, editing and deleting logs (class DBConnect)
- Understand how to connect API for automatic generation of PDFs and display of user-uploaded data sets in Desmos.

Frontend: GUI programming

- Check with client if GUI can be considered accessible and user-friendly
- Testing according with testing plan, corrections implemented if needed
- Compare with success criteria before presenting to client

6) Client presentation

- End-user testing
- Product presentation to client (final client interview)
- Final feedback implementation before project completion
- Discuss future product support with client

Testing Plan

Success criterion	Expected result/action	Details of the test	Error Handling
1	1) A new username and password for login are set – display “Success” message. ‘Login’ window pops up 2) Display warning pop-up window (either “password is too short” or “duplicate login”) 3) User logged out and returned to login screen	1) Registration: New login details (username and password) are entered, “Create account” button pressed 2) The entered password is less than 6 characters or the entered username login is duplicate 3) The user presses the logout button	Verify that the account/login creation for application is correct with ‘if’ statements. Establish database with login credentials
1	1) ‘Main Menu’ window is opened 2) “Incorrect Login” PopUp message is displayed	1) Inputting username and password for an account registered in the Database 2) Inputting username and password for an account not registered in the Database	Password hashing, database access via SQLite
2	The application opens a new concurrent window with readable text guide on the usage of the application	Reader clicks the “Read Guide” button while in the main menu.	BufferReader of the txt guide file in the Resources folder of the application.
3	1) A new experimental data set is uploaded and the user is transferred to the ‘Statistical Evaluation and Interactive Graphs’	1) The user uploads a 2-dimensional CSV data set in the ‘File Upload’ window (the delimiters are either “;” or “,”) 2) The user uploads	IF statements, FileParser and FileChooser classes; BufferReader

	window 2) “Incorrect Data Format. Read instructions in main menu” pop up message is displayed; the user is returned to the Main Menu window. 3) Display “FIle not Chosen” and return to the main menu.	data with an incorrect format or non-numerical values 3) The user does not choose a CSV file	
4,5	The chosen statistical values for the data set and the interactive graph are displayed in the window	The user chooses the statistical evaluation options and presses the “Evaluate” button	JavaFX Checkboxes and LineChart node elements; realised through controller class; null-pointer exceptions.
5	1) Display the plot of the expression on the interactive graph alongside the plotted imported experimental data set. **The Chart is clear, the points are discernable.** 2) Display the popup window with the error message “Unable to read expression.”; returns the reader to the evaluation menu	1) The user types in an explicit-single-argument expression as a function and presses the “Evaluate” button 2) The expression is of incorrect formatting (i.e. has an unrecognized operator or variable)	IF Statements to check if the user inputted the expression (optional); Check that the expression has a single argument and all parameters are defined; check that syntax of expression is algebraically correct - this is realised through the Reverse Polish Notation Expression Binary Tree.
6	A new concurrent window opens in the application that displayed the imported data points in Desmos, displayed through Web View ● Display	The user presses the “Plot in Desmos” button in the Evaluation Menu	Web Engine connected to the Desmos Graphing Calculator API has its own error handling algorithms

	“Connection Error” popup message in accordance with Desmos API’s error handling in case of connectivity issues		
7	1) A window opens to allow the reader to choose the file destination for the generated file; after this, the template-outlined PDF file is displayed with the date of generation, the plotted graph and calculated statistical values. 2) Display “Cannot create empty PDF” popup window and return to Evaluation Menu. 3) Display popup with “No destination chosen”; return user to the evaluation menu ● If the PDF with the name already exists, the FileChooser will display the “File with this name already exists” window	1) The user presses the “Generate PDF” button in the Evaluation Menu **after plotting the graph.** 2) The user presses the “Generate PDF” button in the Evaluation Menu **before plotting the graph.** 3) The user presses “Cancel” on the file destination chooser window	iTextPDF null pointer exceptions; FileChooser error handling; IF Statements.
8	Display the message “Data Set successfully saved”. In the “Access Previous Logs” window, the saved data set is displayed	The user presses the “Save Data Set” button to save the imported dataset in the application and store it in a database securely.	JDBC SQLite scripts to Database: therefore, error handling of issues with Database
9	In the window with the	Users can access and	JDBC SQLite scripts

	saved data, names of files with their date of upload are displayed in a dynamic table. 1) The dataset is removed from the table and the database 2) The Evaluation Menu is opened with the chosen experimental data set for evaluation ● In case of DataBase error, display pop-up window with “DataBase error”	delete their stored datasets and evaluations, with changes accurately reflected in the database 1) “Delete” button is pressed for a data set 2) “Evaluate” button is pressed for a data set	to Database: therefore, error handling of issues with Database

Bibliography:

- Sooki. “What on Earth Is an Expression Tree?” Medium, February 17, 2023. https://sooki.medium.com/what-on-earth-is-an-expression-tree-c4ffe0ca25ac.