

Module Interface Specification for MECHTRON 4TB6

Team 25, Formulate

Ahmed Nazir, nazira1

Stephen Oh, ohs9

Muhanad Sada, sadam

Toluwalayomi Babayeju, babayejt

January 18, 2023

1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Introduction	1
4	Notation	1
5	Module Decomposition	1
6	MIS	3
6.1	Module - ui_main.py	3
6.1.1	Description	3
6.1.2	Classes	3
6.2	Module - ui_functions.py	3
6.2.1	Description	3
6.2.2	Classes	3
6.2.3	Functions	4
6.2.4	Exception Handling	4
6.3	Module - main.py	4
6.3.1	Description	4
6.3.2	Classes	4
6.4	Module - resource_rc.py	5
6.4.1	Description	5
6.4.2	Classes	5
6.4.3	Functions	5
7	Appendix	7

3 Introduction

The following document details the Module Interface Specifications for

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at <https://github.com/ahmed-nazir/Capstone>.

4 Notation

The structure of the MIS for modules comes from Hoffman and Strooper (1995), with the addition that template modules have been adapted from Ghezzi et al. (2003). The mathematical notation comes from Chapter 3 of Hoffman and Strooper (1995). For instance, the symbol $:=$ is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | \dots | c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by MECHTRON 4TB6.

Data Type	Notation	Description
character	char	a single symbol or digit
integer	\mathbb{Z}	a number without a fractional component in $(-\infty, \infty)$
natural number	\mathbb{N}	a number without a fractional component in $[1, \infty)$
real	\mathbb{R}	any number in $(-\infty, \infty)$

The specification of MECHTRON 4TB6 uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, MECHTRON 4TB6 uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	Level 2
Hardware-Hiding	
Behaviour-Hiding	Input Parameters Output Format Output Verification Temperature ODEs Energy Equations Control Module Specification Parameters Module
Software Decision	Sequence Data Structure ODE Solver Plotting

Table 1: Module Hierarchy

6 MIS

6.1 Module - ui_main.py

6.1.1 Description

Python file generated by PyQt designer which sets up the application's window and its design

6.1.2 Classes

Class: Ui_MainWindow() - Contains all methods for setting up the application's window and its static front end design

Methods	Parameters	Return
setupUi() - Takes a PyQt MainWindow object and sets up it's layout according to the ui file created in designer	Self, MainWindow [QMainWindow]	None
retranslateUi() - Sets the static text of the GUI's buttons and labels	Self, MainWindow [QMainWindow]	None

6.2 Module - ui_functions.py

6.2.1 Description

Imports all necessary libraries for backend functions, creates connection to database, and contains class for UI functions

6.2.2 Classes

Class: UIFunctions() - Contains the functions that are connected to buttons in the application's UI

Methods	Parameters	Return
toggleMenu() - Handles the animation for toggling the side menu	Self, maxWidth [integer], enable [boolean]	None
login_into_app() - Checks if the enter username/password are valid and correct and signs user into their account	Self	None
continue_signup() - Checks if all the sign up fields are valid and stores account/login details in database	Self	None

6.2.3 Functions

Function	Parameters	Return
hash_new_password() - Generates a hashed password based on the user's inputted password	password [string]	salt [string], hashed_pass [string]
is_correct_password() - Checks if inputted password matches stored password in database	salt_hex [string], stored_hash [string], pass_to_check [string]	Boolean

6.2.4 Exception Handling

Input validation of the user information is the main form of exception handling. User fields for signing up are checked to ensure that they are not empty and that the password follows the rules of having 8 minimum characters and includes an alphabet, number, and a non-alphanumeric character. When logging in, inputted passwords are checked to ensure that they match the passwords stored in the database. Users will see error messages in the GUI according to what they inputted incorrectly.

6.3 Module - main.py

6.3.1 Description

Imports backend functions and frontend setup of GUI. This is also used to start and run the desktop application

6.3.2 Classes

Class: MainWindow() - Initializes a PyQt main window that is defined in ui_main.py and connects the buttons in the desktop application's UI to backend functions defined in ui_functions.py

Methods	Parameters	Return
__init__() - Initializes the application and connects UI buttons to backend functions	Self	None
changeText() - Add text to menu buttons when toggling full side menu and vice-versa	Self	None

6.4 Module - resource_rc.py

6.4.1 Description

Python file generated by PyQt resource compiler and sets up all the PyQt resources (local images) to be displayed during runtime of application

6.4.2 Classes

6.4.3 Functions

Function	Parameters	Return
qInitResources() - Registers the raw byte data of each image to the Qt resource system	None	None
qCleanupResources() - Unregisters the raw byte data of each image to the Qt resource system	None	None

References

- Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.
- Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL <http://citeseer.ist.psu.edu/428727.html>.

7 Appendix