Module Interface Specification for REVITALIZE

Team 13, REVITALIZE
Bill Nguyen
Syed Bokhari
Hasan Kibria
Youssef Dahab
Logan Brown
Mahmoud Anklis

January 14, 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 https://github.com/BillNguyen1999/REVITALIZE/tree/main/docs/SRS

symbol	description
REVITALIZE	Name of application
UAT	User Acceptance Testing
UI/UX	User Interface/User Experience
HCI	Human-Computer Interface
MG	Module Guide
MIS	Module Interface Specification
SRS	Software Requirements Specification
VnV	Verification and Validation
LP	Login Page
SP	Sign-up Page
MP	Main Page
DS	Diet Section
WS	Workout Section
RS	Rest Section

Contents

3 Introduction

The following document details the Module Interface Specifications for the REVITALIZE app. The REVITALIZE app is an all-in-one health and wellness app, comprised of 1 main section and 3 major subsections. The main section is a calendar which organizes and documents the contents of the 3 subsections. The 3 subsections are the diet section, workout section, and sleep section.

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at https://github.com/BillNguyen1999/REVITALIZE/tree/main/docs.

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 $(c_1 \Rightarrow r_1|c_2 \Rightarrow r_2|...|c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by REVITALIZE.

Data Type	Notation	Description
character	char	a single symbol or digit
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
natural number	$\mathbb{N}$	a number without a fractional component in [1, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)
boolean	bool	value can be True (1) or False (0)
user	User	represents user object, for users of REVITALIZE
date	Date	represents date object, which is useful to add/set/manipulate dates

The specification of REVITALIZE 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, REVITALIZE 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 of Main Menu

6.1 Main Menu Module

6.2 Uses

react
react-native
globalStyles: CSS file to change designs of project
Ionicons: Library for icons
Moment Library is used for Dates (ex. setting date formats (YY/MM/DD))
useRoute react file that is used to navigate between screens of project

6.3 Syntax

6.3.1 Exported Constants

6.3.2 Exported Types

MainScreen = this

6.3.3 Exported Access Programs

Name	In	Out	Exceptions
displayDietScreen	User, Date		
displayExerciseScreen	User, Date		
displaySleepScreen	User, Date		
displayCalendarScreen			

6.4 Semantics

6.4.1 State Variables

user: User

date: Date

6.4.2 Environment Variables

dateText: Text object that displays the selected date.

dateButton: Button object that displays Calendar Screen when clicked.

forwardButton: Button object that displays the next day from current Date value in dateText when clicked

backwardButton: Button object that displays the previous day from current Date value in dateText when clicked

dietButton: Button object that displays Diet Screen when clicked

exerciseButton: Button object that displays Exercise Screen when clicked

sleepButton: Button object that displays Sleep Screen when clicked

6.4.3 Assumptions

N/A

6.4.4 Access Routine Semantics

displayDietScreen(user, date):

- transition: Navigates to Diet Screen when dietButton is pressed
- exception: None

displayExerciseScreen(user, date):

- transition: Navigates to Exercise Screen when exerciseButton is pressed
- exception: None

displaySleepScreen(user, date):

- transition: Navigates to Sleep Screen when sleepButton is pressed
- exception: None

displayCalendarScreen():

- transition: Navigates to Calendar Screen when dateButton is pressed
- exception: None

6.4.5 Local Functions

forwardSetDate():

- transition: date.value := date.value + 1. Sets the next day from the current Date value in dateText when clicked.
- exception: None

backwardSetDate():

- transition: date.value := date.value - 1. Sets the previous day from the current Date value in dateText when clicked.
- exception: None

7 MIS of Calendar

7.1 Calendar Module

7.2 Uses

react
react-native
globalStyles: CSS file to change designs of project
react-native-calendars: Library useful for implementing calendars in react-native
useRoute react file that is used to navigate between screens of project

7.3 Syntax

7.3.1 Exported Constants

7.3.2 Exported Types

CalendarScreen = this

7.3.3 Exported Access Programs

Name	In	Out	Exceptions
onDayPress			
onMonthChange			
onPressArrowLeft			
onPressArrowRight			

7.4 Semantics

7.4.1 State Variables

date: Date

7.4.2 Environment Variables

monthText: Text object that displays the selected month.

forwardMonthButton: Button object that displays the next month from current month value in monthText when clicked

backwardMonthButton: Button object that displays the previous month from current month value in monthText when clicked

7.4.3 Assumptions

N/A

7.4.4 Access Routine Semantics

onDayCalendar():

- transition: Changes date value to selected date value in CalendarScreen
- exception: None

onMonthChange():

- transition: Changes date.month.value to new date.month.value and monthText will be changed to string value of new date.month.value
- exception: None

onPressArrowRight():

- transition: date.month.value := date.month.value + 1. Sets the next date.month.value from the current date.month.value in monthText when clicked
- exception: None

onPressArrowLeft():

- transition: date.month.value := date.month.value - 1. Sets the previous date.month.value from the current date.month.value in monthText when clicked
- exception: None

7.4.5 Local Functions

N/A

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.

8 Appendix

[Extra information if required —SS]