Module Interface Specification for REVITALIZE

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

April 5, 2023

1 Revision History

Date	Version	Notes
January 14th, 2023	Bill Nguyen	Added MIS for Main Menu and Calendar
January 14th, 2023	Bill Nguyen	Added Introduction, Notation, Acronyms
January 16th, 2023	Youssef Dahab	Added Sleep Container Module
January 17th, 2023	Youssef Dahab	Added Login, Label, & Circular Slider Modules
January 17th, 2023	Hasan Kibria	Added Diet and Recipe Modules
January 17th, 2023	Logan Brown	Added Workout and Sign-up Modules
January 17th, 2023	Bill Nguyen	Added Module Decomposition

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at https://github.com/BillNguyen1999/REVITALIZE/tree/main/docs/SRS

symbol	description
DS	Diet Section
HCI	Human-Computer Interface
LP	Login Page
MG	Module Guide
MIS	Module Interface Specification
MP	Main Page
REVITALIZE	Name of application
RS	Rest Section
SP	Sign-up Page
SRS	Software Requirements Specification
UAT	User Acceptance Testing
UI/UX	User Interface/User Experience
VnV	Verification and Validation
WS	Workout 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	$\mathbb{B}$	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 Module	
Behaviour-Hiding Module	Main Menu
	Calendar
	Login
	Container
	Label
	Circular Slider
	Diet Log
	Search or Add Food
	Custom Meal
	Search Recipe
	Recipe Results
	Recipe Details
	Workout Display
	Workout Edit
	Workout Log
	Signup
Software Decision Module	FoodT
	ExerciseT

Table 1: Module Hierarchy

6 MIS of Login

6.1 Login Module

6.2 Uses

react
react-native
globalStyles: CSS file to change designs of project
useRoute react file that is used to navigate between screens of project

6.3 Syntax

6.3.1 Exported Constants

N/A

6.3.2 Exported Types

N/A

6.3.3 Exported Access Programs

Name	In	Out	Exceptions
displayMainMenuScreen	~~NameOrEmail~~ string, ~~Password~~ string		
displayForgotPasswordScreen			
displaySignUpScreen			

6.4 Semantics

6.4.1 State Variables

NameOrEmail: string that stores user input of name or email

Password: string that stores user passwoord input

6.4.2 Environment Variables

LoginButton: button object that displays Main Menu screen when clicked

ForgotPasswordButton: button object that displays Forgot Password screen when clicked

SignUpButton: button object that displays Sign Up screen when clicked

6.4.3 Assumptions

N/A

6.4.4 Access Routine Semantics

displayMainMenuScreen(NameOrEmail, Password):

- transition: navigates to Main Menu screen when login button is pressed after successfully entering name or email and password
- exception: None

displayForgotPasswordScreen():

- transition: navigates to Forgot Password screen when forgot password link is clicked
- exception: None

displaySignUpScreen():

- transition: navigates to Sign Up screen when sign up link is clicked
- exception: None

6.4.5 Local Functions

N/A

7 MIS of Sign-up

7.1 Sign-up Module

7.2 Uses

react
react-native
globalStyles: CSS file to change designs of project
useRoute react file that is used to navigate between screens of project

7.3 Syntax

7.3.1 Exported Constants

N/A

7.3.2 Exported Types

N/A

7.3.3 Exported Access Programs

Name	In	Out	Exceptions
~~displayMainMenuScreen~~	~~Name, Email, Password, ConfirmPassword~~		
displayMainMenuScreen	string, string, string, string		
displayLoginScreen			

7.4 Semantics

7.4.1 State Variables

Name: string that stores user input of name

Email: string that stores user email

Password: string that stores user password input

ConfirmPassword: string that stores user password input

7.4.2 Environment Variables

signUpButton: Button object that creates account and navigates to Main Screen

loginButton: Button that navigates back to Login Screen

7.4.3 Assumptions

N/A

7.4.4 Access Routine Semantics

displayMainMenuScreen(Name, Email, Password, ConfirmPassword):

- transition: navigates to Main Menu screen when sign-up is validated and successful
- exception: None

displayLoginScreen():

- transition: navigates to login screen when login page link is clicked

- exception: None

7.4.5 Local Functions

N/A

8 MIS of Main Menu

8.1 Main Menu Module

8.2 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

8.3 Syntax

8.3.1 Exported Constants

N/A

8.3.2 Exported Types

MainScreen = this

8.3.3 Exported Access Programs

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

8.4 Semantics

8.4.1 State Variables

user: User

date: Date

8.4.2 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

8.4.3 Assumptions

N/A

8.4.4 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

8.4.5 Local Functions

forwardSetDate():

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

backwardSetDate():

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

9 MIS of Calendar

9.1 Calendar Module

9.2 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

9.3 Syntax

9.3.1 Exported Constants

N/A

9.3.2 Exported Types

CalendarScreen = this

9.3.3 Exported Access Programs

Name	In	Out	Exceptions
onDayPress			
onMonthChange			
onPressArrowLeft			
onPressArrowRight			

9.4 Semantics

9.4.1 State Variables

date: Date

9.4.2 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

9.4.3 Assumptions

N/A

9.4.4 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

9.4.5 Local Functions

N/A

10 MIS of Sleep

10.1 Container Module

10.2 Uses

react
react-native
react-native-reanimated
react-native-redash
Label: Module
Circular Slider: Module

10.3 Syntax

10.3.1 Exported Constants

PI := Math (object that provides mathematics functionality and constants)
TAU := 2 * PI

10.3.2 Exported Types

N/A

10.3.3 Exported Access Programs

Name	In	Out	Exceptions
DisplayContainer	~~start~~ $\mathbb{R}$, ~~end~~ $\mathbb{R}$		
onSlideCircularArc			

10.4 Semantics

10.4.1 State Variables

start: $\mathbb{R}$, the set bedtime

end: $\mathbb{R}$, the set wake up time

10.4.2 Environment Variables

BedTime: string object that displays the selected bedtime.

WakeUpTime: string object that displays the selected wake up time.

SleepTime: string object that displays the total sleep time.

ArcStartPos: polar coordinates object representing starting position of circular slider arc. Modifies BedTime and SleepTime when slid.

ArcEndPos: polar coordinates object representing ending position of circular slider arc. Modifies WakeUpTime and SleepTime when slid.

CircularSliderArc: string literal object representing an arc.

10.4.3 Assumptions

N/A

10.4.4 Access Routine Semantics

DisplayContainer():

- output: display bedtime, wake up time, sleep time, arc starting and ending positions, and circular slider arc
- exception: None

onSlideCircularArc():

- transition: recalculate total SleepTime based on the modified BedTime or WakeUpTime
- output: display new total SleepTime on screen

10.4.5 Local Functions

radToMinutes(rad):

- output: rad * 24 * 60 / TAU
- exception: None

absoluteDuration(start, end):

- output: start > end ? end + (TAU - start) : end - start

- exception: None

formatDuration2(duration):

- output: total sleep time formatted in hours followed by minutes.
- exception: None

11 MIS of Sleep

11.1 Label Module

11.2 Uses

react
react-native
react-native-reanimated
react-native-redash
@expo/vector-icons

11.3 Syntax

11.3.1 Exported Constants

PI := Math (object that provides mathematics functionality and constants)
TAU := 2 * PI

11.3.2 Exported Types

N/A

11.3.3 Exported Access Programs

Name	In	Out	Exceptions
DisplayImage			
DisplayLabel	~~start~~ $\mathbb{R}$, ~~end~~ $\mathbb{R}$		
onSlideCircularArc	~~start~~ $\mathbb{R}$, ~~end~~ $\mathbb{R}$		

11.4 Semantics

11.4.1 State Variables

start: $\mathbb{R}$, the set bedtime is passed from Container module

end: $\mathbb{R}$, the set wake up time is passed from Container module

11.4.2 Environment Variables

BedTime: string object that displays the selected bedtime.

WakeUpTime: string object that displays the selected wake up time.

11.4.3 Assumptions

N/A

11.4.4 Access Routine Semantics

DisplayImage():

- output: display bed icon, "BEDTIME" text, ring icon, and "WAKE UP" text
- exception: None

DisplayLabel(start, end):

- output: display user set BedTime and WakeUpTime
- exception: None

onSlideCircularArc(start, end):

- transition: modify BedTime and WakeUpTime values to new BedTime and WakeUpTime values respectively
- exception: None

11.4.5 Local Functions

radToMinutes(rad):

- output: rad * 24 * 60 / TAU
- exception: None

formatDuration(duration):

- output: set bed time and wake up time in the 24-hour clock format
- exception: None

12 MIS of Sleep

12.1 Circular Slider Module

12.2 Uses

react
react-native
react-native-reanimated
react-native-redash
react-native-svg

12.3 Syntax

12.3.1 Exported Constants

PI := Math (object that provides mathematics functionality and constants)
TAU := 2 * PI

12.3.2 Exported Types

N/A

12.3.3 Exported Access Programs

Name	In	Out	Exceptions
DisplayCircularSlider	ArcStartPos, ArcEndPos		
onSlideCircularArc	ArcStartPos, ArcEndPos		

12.4 Semantics

12.4.1 State Variables

ArcStartPos: polar coordinates object representing starting position of circular slider arc

ArcEndPos: polar coordinates object representing ending position of circular slider arc

12.4.2 Environment Variables

ArcStartPos: Modifies BedTime and SleepTime when slid.

ArcEndPos: Modifies WakeUpTime and SleepTime when slid.

CircularSliderArc: string literal object representing an arc.

12.4.3 Assumptions

N/A

12.4.4 Access Routine Semantics

DisplayCircularSlider(start, end):

- output: display arc starting position, ending position, and circular slider arc
- exception: None

onSlideCircularArc(ArcStartPos, ArcEndPos):

- transition: modify ArcStartPos and ArcEndPos coordinate values to new ArcStartPos and ArcEndPos coordinate values respectively when user slides circular arc
- exception: None

12.4.5 Local Functions

absoluteDuration(start, end):

- output: start > end ? end + (TAU - start) : end - start
- exception: None

ConvertArcStartPos(ArcStartPos):

- output: convert ArcStartPos from polar coordinates to canvas coordinates
- exception: None

ConvertArcEndPos(ArcEndPos):

- output: convert ArcEndPos from polar coordinates to canvas coordinates
- exception: None

13 FoodT Module

13.1 Module

IndicatorT

13.2 Uses

None

13.3 Syntax

13.3.1 Exported Constants

None

13.3.2 Exported Types

FoodT = {
calories, *#Calories in meal*
name, *#Name of meal*
carbs, *#Carbohydrates in meal*
protein, *#Protein in meal*
fat *#Fat in meal*
}

13.3.3 Exported Access Programs

None

13.4 Semantics

13.4.1 State Variables

None

13.4.2 State Invariant

None

13.4.3 Assumptions

None

13.4.4 Considerations

None

14 Diet Log Module

14.1 Uses

react
react-native
globalStyles: CSS file to change designs of project
Ionicons: Library for icons
useRoute react file that is used to navigate between screens of project

14.2 Syntax

14.2.1 Exported Constants

14.2.2 Exported Types

DietLogScreen = this

14.2.3 Exported Access Programs

Name	In	Out	Exceptions
calculateDailyNutrition	seq of FoodT	seq of <String, $\mathbb{R}$>	
removeFood	food: FoodT		

14.3 Semantics

14.3.1 State Variables

foodList: seq of FoodT
totalNutrition: seq of <String, $\mathbb{R}$>
date: Date

14.3.2 Environment Variables

addFoodButton: Button object that shifts user to Search or Add Food Module.

deleteFoodButton: Button object that deletes a food entry from this module when clicked

editFoodButton: Button object navigates to Log Meal Module

14.3.3 Assumptions

N/A

14.3.4 Access Routine Semantics

calculateDailyNutrition():

- transition: totalNutrition :=$\sum$ foodList
- exception: None

removeFood(food: FoodT):

- transition: foodList := {foodList} \ food. Sets the next date.month.value from the current date.month.value in monthText when clicked
- exception: None

14.3.5 Local Functions

N/A

15 Search or Add Food Module

15.1 Uses

react
react-native
globalStyles: CSS file to change designs of project
useRoute react file that is used to navigate between screens of project

15.2 Syntax

15.2.1 Exported Constants

15.2.2 Exported Types

DecisionScreen = this

15.2.3 Exported Access Programs

None

15.3 Semantics

15.3.1 State Variables

None

15.3.2 Environment Variables

searchRecipeButton: Button object that shifts user to Search Recipe Module.

addCustomMealButtom: Button object that shifts user to Log Meal Module

15.3.3 Assumptions

N/A

15.3.4 Access Routine Semantics

calculateDailyNutrition():

- transition: totalNutrition :=$\sum$ foodList
- exception: None

removeFood(food: FoodT):

- transition: foodList := {foodList} \ food. Sets the next date.month.value from the current date.month.value in monthText when clicked
- exception: None

15.3.5 Local Functions

N/A

16 Custom Meal Module

16.1 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

16.2 Syntax

16.2.1 Exported Constants

N/A

16.2.2 Exported Types

CustomMealScreen = this

16.2.3 Exported Access Programs

Name	In	Out	Exceptions
saveCustomMeal	foodInfo: seq of String	FoodT	

16.3 Semantics

16.3.1 State Variables

foodList: seq of String

16.3.2 Environment Variables

addButton: Save food information.

16.3.3 Assumptions

N/A

16.3.4 Access Routine Semantics

saveCustomMeal(foodInfo):

- transition: foodList := foodList ∪ FoodT(foodInfo)
- exception: None

16.3.5 Local Functions

N/A

17 Search Recipe Module

17.1 Uses

react
react-native
globalStyles: CSS file to change designs of project
Ionicons: Library for icons
useRoute react file that is used to navigate between screens of project

17.2 Syntax

17.2.1 Exported Constants

N/ A

17.2.2 Exported Types

SearchRecipeScreen = this

17.2.3 Exported Access Programs

Name	In	Out	Exceptions
searchRecipe	filterList: seq of <String, String>	Seq of <String, String>	
returnRecipeList		Seq of <String, String>	

17.3 Semantics

17.3.1 State Variables

filterList: seq of <String, String>
recipeList: Seq of <String, String>

17.3.2 Environment Variables

searchRecipeButton: Button object that shifts user to Search Results Module, calls searchRecipe().

17.3.3 Assumptions

N/A

17.3.4 Access Routine Semantics

searchRecipe(filterList):

- transition: Populates state variable recipeList using external API call.
- exception: None

17.3.5 Local Functions

N/A

18 Recipe Results Module

18.1 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

18.2 Syntax

18.2.1 Exported Constants

N/A

18.2.2 Exported Types

RecipeResultsScreen = this

18.2.3 Exported Access Programs

Name	In	Out	Exceptions
retrieveRecipeInfo	recipeLink: String	<String, FileObject>	

18.3 Semantics

18.3.1 State Variables

recipeList: Seq of <String, String>

18.3.2 Environment Variables

18.3.3 Assumptions

N/A

18.3.4 Access Routine Semantics

retrieveRecipeInfo(recipeLink):

- transition: Retrieve details and picture file of recipe found at recipeLink
- exception: None

18.3.5 Local Functions

N/A

19 Recipe Details Module

19.1 Uses

react
react-native
globalStyles: CSS file to change designs of project
Ionicons: Library for icons
useRoute react file that is used to navigate between screens of project

19.2 Syntax

19.2.1 Exported Constants

19.2.2 Exported Types

RecipeDetailsScreen = this

19.2.3 Exported Access Programs

N/A

19.3 Semantics

19.3.1 State Variables

recipeDetails: seq of <String, FileObject>

19.3.2 Environment Variables

addRecipeButton: Add recipe to Daily Food Log.

19.3.3 Assumptions

N/A

19.3.4 Access Routine Semantics

19.3.5 Local Functions

N/A

20 MIS of Workout

20.1 ExerciseT Module

20.2 Uses

N/A

20.3 Syntax

20.3.1 Exported Constants

N/A

20.3.2 Exported Types

ExerciseT = {
name: string
reps: $\mathbb{N}$
weight: $\mathbb{N}$
sets: $\mathbb{N}$
}

20.3.3 Exported Access Programs

N/A

20.4 Semantics

20.4.1 State Variables

N/A

20.4.2 Environment Variables

N/A

20.4.3 Assumptions

N/A

20.4.4 Access Routine Semantics

getWorkout(user, date):

- transition: Fetches the date's workout from Workout Log Module
- exception: None

20.4.5 Local Functions

N/A

20.5 Workout Display Module

20.6 Uses

react
react-native
globalStyles: CSS file to change designs of project
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
wger: Rest API providing exercise images and names
Workout Log: Module for storing workouts
ExerciseT: Module representing an exercise

20.7 Syntax

20.7.1 Exported Constants

N/A

20.7.2 Exported Types

WorkoutScreen = this

20.7.3 Exported Access Programs

Name	In	Out	Exceptions
getWorkout	User, Date	Workout: Seq of ExerciseT	

20.8 Semantics

20.8.1 State Variables

user: User

date: Date

20.8.2 Environment Variables

editButton: Button object that navigates to Workout Edit Module when clicked

workout: Text object that displays the current date's workout

20.8.3 Assumptions

N/A

20.8.4 Access Routine Semantics

getWorkout(user, date):

- transition: out := exerciseT from the date's workout in Workout Log
- exception: None

20.8.5 Local Functions

N/A

20.9 Workout Edit Module

20.10 Uses

react
react-native
globalStyles: CSS file to change designs of project

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
wger: Rest API providing exercise images and names
Workout Log: Module for storing workouts
ExerciseT: Module representing an exercise

20.11 Syntax

20.11.1 Exported Constants

N/A

20.11.2 Exported Types

N/A

20.11.3 Exported Access Programs

Name	In	Out	Exceptions
setExercise	exercise: string, exerciseT		
setReps	reps $\mathbb{N}$, exerciseT		
setWeight	weight: $\mathbb{N}$, exerciseT		
setSets	sets: $\mathbb{N}$, exerciseT		
removeExercise			
addPreviousWorkout	name: string		

20.12 Semantics

20.12.1 State Variables

user: User

date: Date

units: lbs or kgs

workoutName: string

exercises: Seq of ExerciseT

20.12.2 Environment Variables

addExerciseButton: Button object that adds blank exercise to Workout

saveButton: Saves changes to Workout Log

exitButton: Button object that leaves Workout Edit changeUnitsButton: Toggles the units from lbs to kgs and vice versa

removeExerciseButton: Deletes repective exercise from Workout Log

20.12.3 Assumptions

N/A

20.12.4 Access Routine Semantics

setExercise(name, exerciseT):

- transition: Adds the selected exercise name to Workout Log
- exception: None

setReps(reps, exerciseT):

- transition: Saves the selected number of reps to Workout Log
- exception: None

setWeight(weight, exerciseT):

- transition: Saves the selected weight in *units* to Workout Log
- exception: None

setSets(sets, exerciseT):

- transition: Saves the selected number of sets to Workout Log
- exception: None

removeExercise():

- transition: Deletes selected exercise from Workout Log
- exception: None

20.12.5 Local Functions

N/A

20.13 Workout Log Module

20.14 Uses

react
react-native
globalStyles: CSS file to change designs of project
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
wger: Rest API providing exercise images and names

20.15 Syntax

20.15.1 Exported Constants

N/A

20.15.2 Exported Types

WorkoutLog = this

20.15.3 Exported Access Programs

Name	In	Out	Exceptions
setName	name: string		

20.16 Semantics

20.16.1 State Variables

user: User

date: Date

workouts: <Seq of ExerciseT, name: string>

20.16.2 Environment Variables

N/A

20.16.3 Assumptions

N/A

20.16.4 Access Routine Semantics

setName(name):

- transition: Sets the name of workout (default value is date)
- exception: None

20.16.5 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.

21 Appendix

N/A