**Design:**

Structure charts:

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

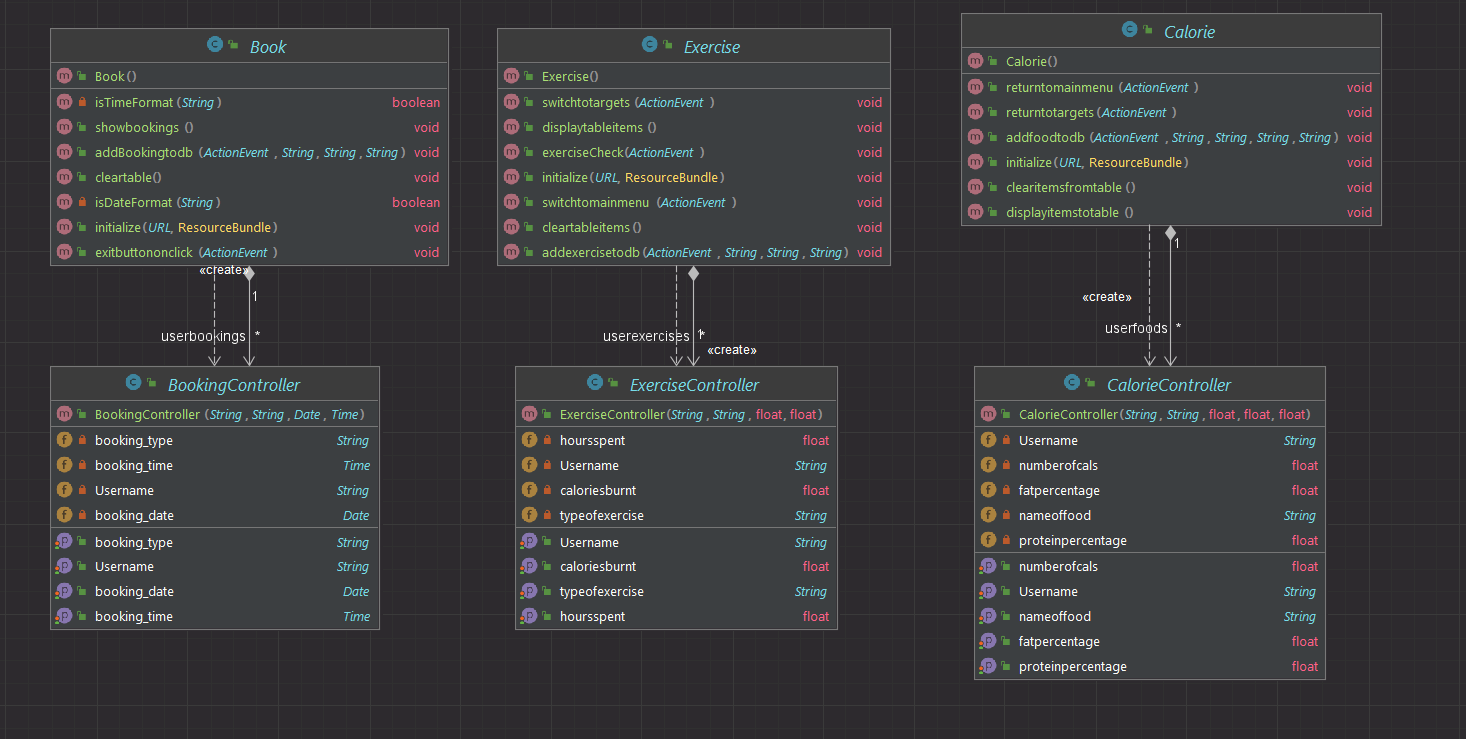
Flowchart remains the same since analysis.

Diagram

Description automatically generatedDFD (Data Flow Diagram):

|  |  |  |  |
| --- | --- | --- | --- |
| **Storage** | | **Processes** | |
| S1 | USERFOODS - table in the database holds info for calories, protein %, fat grams, and food name | P1 | The process of adding a food to not only the database but to the GUI and outputting the info the user enters |
| S2 | USEREXERCISE - table in the database holds info for exercise type, calories burnt and duration | P2 | The process of adding an exercise to not only the database but to the GUI and outputting the info the user enters |
| S3 | USERBOOKINGS - table in the database holds info for booking type, booking date and booking time | P3 | The process of adding a booking to not only the database but to the GUI and outputting the info the user enters |
| S4 | USERTARGETS - table in the database holds info for target weight, target active hours and target calorie intake | P4 | The process of changing and uploading new user targets and potentially overwriting old targets, also outputs them through GUI |
| S5 | USERACCOUNTS - table in the database holds info for username, password, current weight, height, current active hours and current calorie intake | P5 | The process of signing the user up with all required info being sent to the database |
|  |  | P6 | Logging in process checks Username and Password to whatever user submits and sends user to the home screen GUI |

Class Diagram (Where there are links):



Java Reference Diagram:

A screenshot of a video game

Description automatically generated

FULL JAVA CLASS DIAGRAM:

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

Entity Relationship Diagram:

Diagram

Description automatically generated

Data Dictionary:

useraccounts:

A screenshot of a computer

Description automatically generated with low confidence

userbookings:

A screenshot of a computer

Description automatically generated with low confidence

userexercise:

Table

Description automatically generated

userfood:

Table

Description automatically generated

usertargets:

Table

Description automatically generated

File Organisation and Processing:

Table

Description automatically generated

Total size:

Table

Description automatically generated

Some of my algorithms in pseudocode:

These include the ones I am most proud of and were the hardest to implement:

Merge Sort + Recursion:

function mergeSort(arr, L, R)

if L > R

M = (1 + R) / 2

mergeSort(arr, 1, M)

mergeSort(arr, M + 1, R)

merge(arr, L, M, R)

end if

end function

function merge(arr, L, M, R)

n1 = M - L + 1

n2 = R - M

left = new ArrayList(n1)

right = new ArrayList(n2)

for i = 0 to n1-1

left.add(arr[L + i])

end for

for j = 0 to n2-1

right.add(arr[M + 1 + j])

end for

i = 0

j = 0

k = L

while i < n1 AND j < n2

if left[i] <= right[j]

arr[k] = left[i]

i++

else

arr[k] = right[j]

j++

end if

k++

end while

while i < n1

arr[k] = left[i]

i++

k++

end while

while j < n2

arr[k] = right[j]

j++

k++

end while

end function

SQL Functions:

INSERT INTO userbookings (Username, booking\_type, booking\_date, booking\_time) VALUES (?,?,?,?)

SELECT \* FROM userbookings WHERE Username = ?

DELETE FROM userbookings WHERE Username = ?

INSERT INTO userfood (Username,nameoffood,numberofcals,fatpercent,proteinpercent) VALUES (?,?,?,?,?)

SELECT \* FROM userfood WHERE Username = ?

DELETE FROM userfood WHERE Username = ?

UPDATE usertargets +

"SET targetweight = '" + targetweight +

"', targetactivehours = '" + targetactivehours +

"', targetavcalintake = '" + targetavcalintake +

"' WHERE Username = '" + logincontroller.currentuser + "'"

INSERT INTO userexercise (Username,typeofexercise,caloriesburnt,hoursspent) VALUES (?,?,?,?)

SELECT \* FROM userexercise WHERE Username = ?

DELETE FROM userexercise WHERE Username = ?

“SELECT count(1) FROM useraccounts WHERE Username = '" + usernameTextField.getText() + "' AND Password = '" + passwordTextField.getText() + "'"

SELECT \* FROM useraccounts WHERE Username = ?

SELECT \* FROM usertargets WHERE Username = ?

Regular Expression:

function isDateFormat(input)

regex = "^\\d{4}-\\d{2}-\\d{2}$"

pattern = Pattern.compile(regex)

matcher = pattern.matcher(input)

return matcher.matches()

end function

function isTimeFormat(input)

regex = "^\\d{2}:\\d{2}:\\d{2}$"

pattern = Pattern.compile(regex)

matcher = pattern.matcher(input)

return matcher.matches()

end function

Design Overview:

* Through the use of a variety of algorithms and models shown, the solution will try to achieve the majority of the SMART targets mentioned in the analysis.