

Input/Output

<i>Input</i>	<i>Java Date Type</i>	<i>Example</i>
Name	String	"Samuel Bush"
Year	String	"2012/13"
Date of input	String	"20.02.2012"
Block	String	"A Block"
Grade	String	"10"
60 meter run	Date	"0:12.2"
200 meter run	Date	"0:59"
500 meter run	Date	"1:59"
800 meter run	Date	"2:22"
1,500 meter run	Date	"5:42"
Shot-put attempt #1	double	"30.1"
Shot-put attempt #2	double	"28.8"
Shot-put attempt #3	double	"31.2"
Javelin attempt #1	Double	"45.12"
Javelin attempt #2	Double	"42.1"
Javelin attempt #3	Double	"43.27"
Wall sit test	Date	"1:30"
Sit and reach test attempt #1	Double	"38.2"
Sit and reach test attempt #2	Double	"39.1"
Hurdles 60 meter run	Date	"0:17.3"

<i>Output</i>	<i>Java Data Type</i>	<i>Example</i>
Name	String	"Samuel Bush"
Year	String	"2012/13"
Block	String	"A Block"
Date of Input	String	"20.02.2012"
Grade	String	"10"
60 meter run	Date	"0:12.2"
200 meter run	Date	"0:59"
500 meter run	Date	"1:59"
800 meter run	Date	"2:22"
1,500 meter run	Date	"5:42"
Shot-put best attempt	Double	"31.2"
Javelin best attempt	Double	"45.12"
Wall sit test	Date	"1:30"
Sit and reach test best attempt	Double	"39.1"
Hurdles 60 meter run	Date	"0:17.3"
Average of Shot-put for Block 'x'	Double	"32.12"
Average of javelin for Block 'x'	Double	"43.84"
Average of wall sit test	Date	"1:21"

for Block 'x'		
Average of sit and reach test for Block 'x'	Double	"35.2"
Average of 60 meter run for Block 'x'	Date	"0:11.2"
Average of 200 meter run for Block 'x'	Date	"1:02.2"
Average of 500 meter run for Block 'x'	Date	"2:03"
Average of 800 meter run for Block 'x'	Date	"2:25"
Average of 1,500 meter run for Block 'x'	Date	"5:51"
Average of Shot-put for Grade '#'	Double	"31.12"
Average of javelin for Grade '#'	Double	"46.84"
Average of wall sit test for Grade '#'	Date	"1:11"
Average of sit and reach test for Grade '#'	Double	"33.2"
Average of 60 meter run for Grade '#'	Date	"0:12.2"
Average of 200 meter run for Grade '#'	Date	"1:00.2"
Average of 500 meter run for Grade '#'	Date	"2:05"
Average of 800 meter run for Grade '#'	Date	"2:15"
Average of 1,500 meter run for Grade '#'	Date	"5:49"

Chronological Development Plan

#1 - Finalize the GUI Class

Include all functions in visual
Variable names for all text-fields, buttons and other
Create methods for all buttons

#2 - Develop the Attributes Class

Learn the date class
Make gets and sets for all attributes
only 1 constructor

#3 - Create Input Method

Attributes to Display Table

```
pseudocode -
studentList.add(new Student(attributes by getText);
for loop(through rows of table){
for loop(through objects in ArrayList){
displayTable.setValueAt(studentArrayList.get(numOfObject).getAttributes());
    }
}
```

Set-up error messages

```
pseudocode -
JOptionPane.showMessageDialog("warning message",
JOptionPane.WARNING_MESSAGE);
```

```
Pseudocode for error messages
if(extBox empty){
JOptionPane.showMessageDialog("warning message",
JOptionPane.WARNING_MESSAGE);
}
```

#4 - Saving Method

Saving Method

```
pseudocode -
JFileChooser = new JFC();
JFileChooser.showOpenDialog();
try{
Filewriter = new FW(getSelected File);
BufferedWriter = new BF(FW);
for loop{
bw. write(studentList.get(i).getAttributes);
    }
}
catch
```

Opening Method

pseudocode -

```
JFileChooser = new JFC();
JFileChooser.showSaveDialog();
try{
  FileReader = new FR(getSelectedFile);
  BufferedReader = new BR(FR);
  readIn = "";
  readIn = br.readLine();
  StringTokenizer = new TK(readIn, ":");
  rowCounter = 0;
  while(st.hasMoreTokens){
    add row
    displayTable.setValueAt(st.nextToken, rowCounter, column);
  }
}
catch
```

#5 - Average Method

for class

for year

#6 - Finding best score

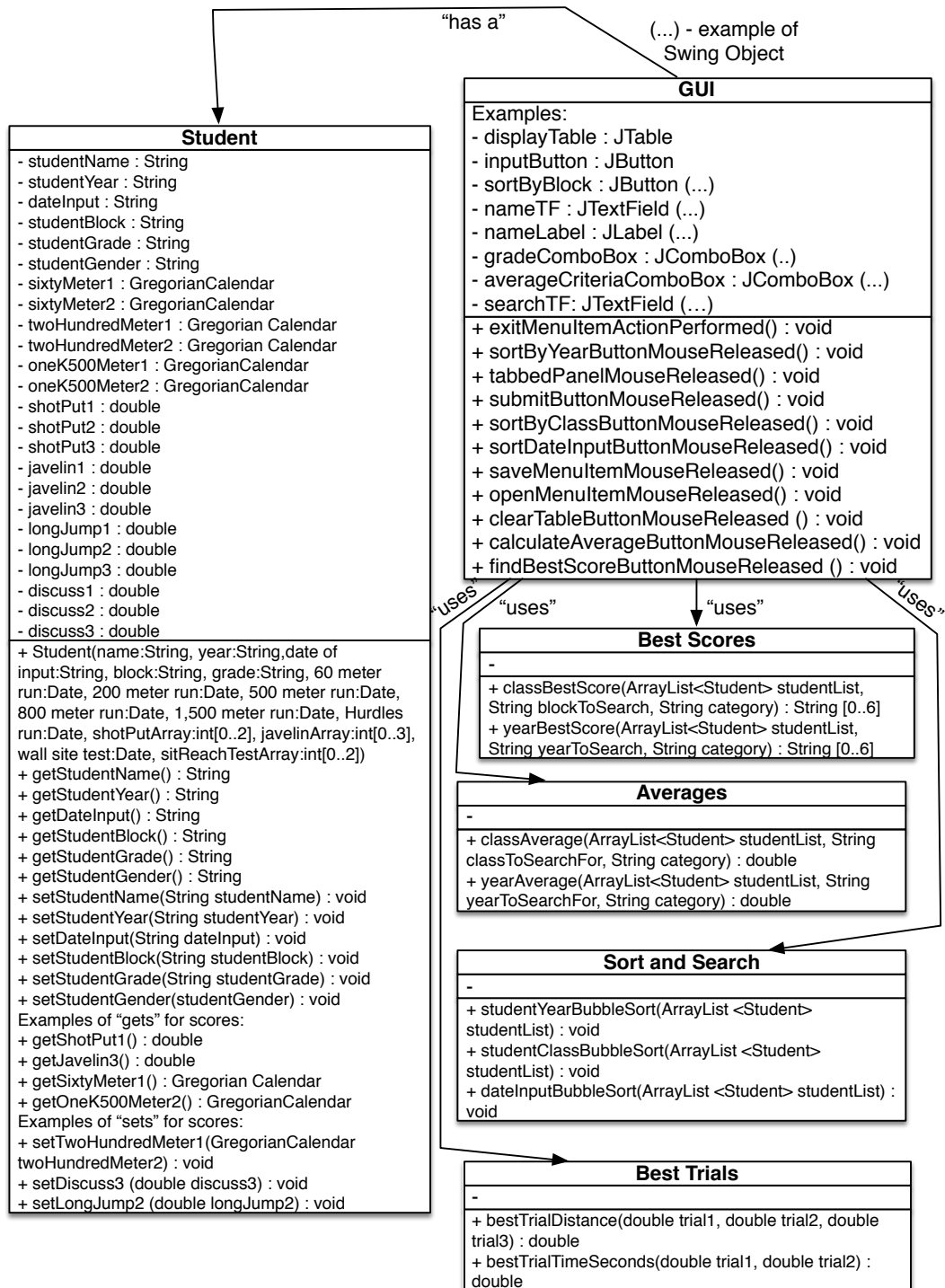
overall

in year

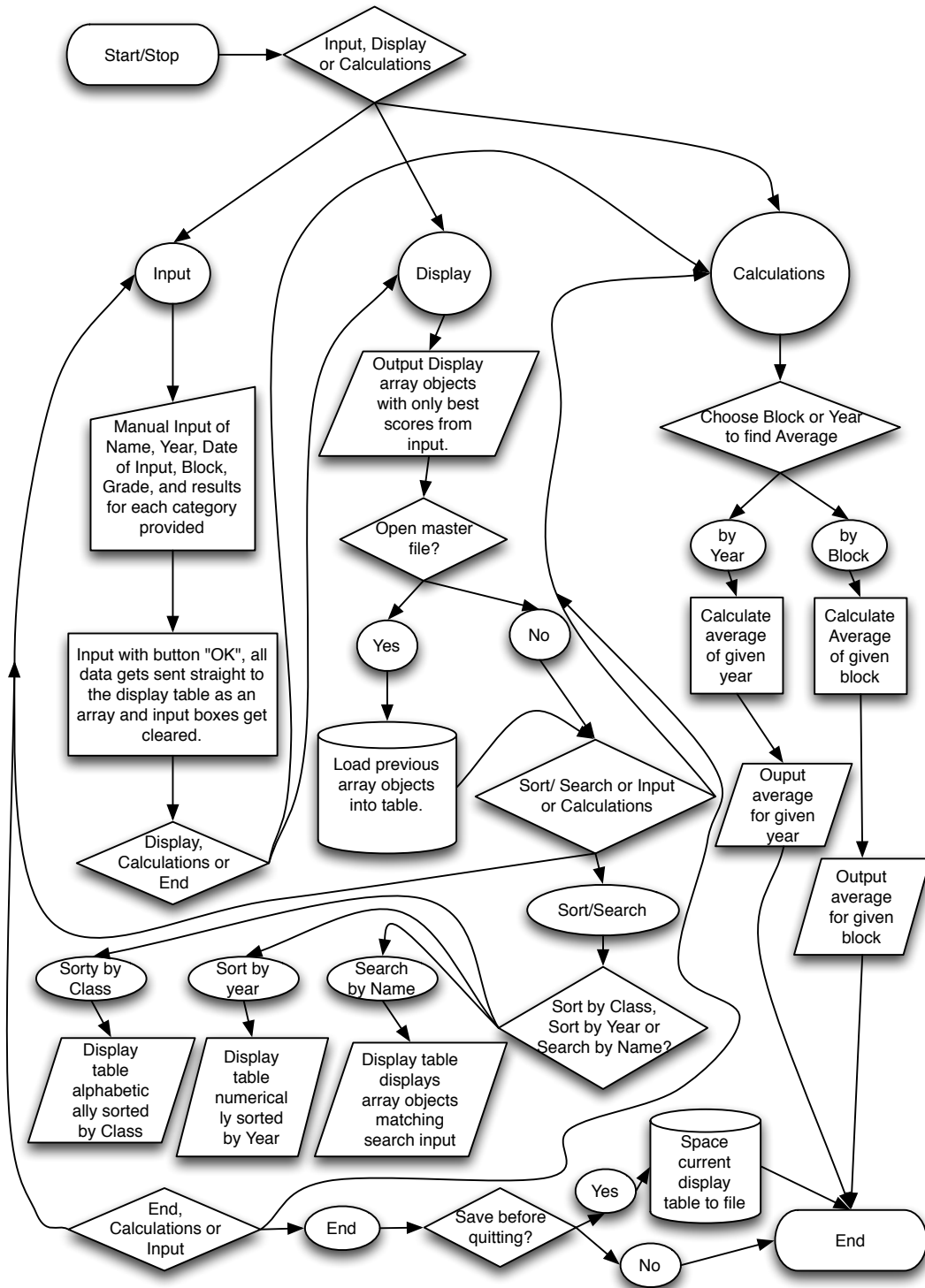
in grade level

in class

UML Diagram:



Flow Chart



Testing Plan:

Tasks performed by this Tab	Functioning Example to test	Border Case to test	Abnormal/Extreme Situations to test	Other Potential Problems to test
Input of Full Name	Robert Damon	N/A	<p>No spaces (RobertDamon) – program should not accept this, ask to re-submit</p> <p>Too many names (Robert Downey Alexander Maximilian Damon) – program should display a pop-up window asking to double check name</p> <p>Only one name (Robert) – program should not accept this, ask to re-submit</p> <p>Numbers input (Robert25 Damon) – program should not accept this, ask to re-submit</p>	<p>*Multiple Abnormal Situations – program should not accept input, pop-up window listing wrong messages</p> <p>*No information input – program should not accept input, pop-up window listing missing information</p>
Input of Grade Level	11	Less than 0 or higher than 12 (for ex. 0, 13) – program should not accept this, pop-up window that number must be between 0-12 and ask to re-submit	Any chars input (Grade 11) – should not be accepted, pop-up window with message to submit only a number and ask to re-submit	*See full name above
Input of School Year	2013/14	Wrong format (for ex. 2013/2014, 2013, 2013.14) program should not accept this, pop-up window that year should be in (yyyy/yy) format and ask to re-submit	Any chars input (year: 2013/14) – program should not accept this, pop-up window that year should be in (yyyy/yy) format, ask to re-submit	*See full name above
Input of Block	B	Only letter should be written (Block B) – program should	No char is written (2) – program should not accept this, pop-up window that block must contain a	*See full name above

		not accept this, pop-up window that only block letter should be written and ask to re-submit	letter and ask to re-submit	
Input of Throwing Distances	36.7	Unit is written into the input (36 meters) – program should not accept this, pop-up window that only the distance is required and ask to re-submit	Not written in double form (thirty-six) – program should not accept this, pop-up window that distance should be written in number and ask to re-submit	*See full name above
Input of (Running) Times	3:44.6	Recorded in wrong format (ie. 3 minutes, 3.44.6, 44.6 seconds) – program should not accept this, pop-up window that time has to be in format (mm:ss.millisecond) and ask to re-submit	Not written with letters (fifty five minutes) – program should not accept this, pop-up window that input must be in (mm:ss.millisecond) format and ask to re-submit	*See full name above

Display Tab

Tasks performed by this tab	Functioning Example to test	Border Case to test	Abnormal/Extreme Situations to test	Other Potential Problems to test
Search for a student	User inputs name of student to be searched for (Richard Damon) and presses the search button	Only one name is searched for (Damon, Richard) – program should not accept this, pop-up window that full name must be searched and ask to re-submit	Something other than a String is input (Richard123 Damon) – program should not accept this, pop-up window that name was incorrectly put in and ask to re-submit	No name is input – program should display pop-up window asking to fill in the search bar and re-submit
Sort the Students by Date of Input or by Year or by Class	User clicks on the button to sort by chosen category	N/A	The students are already sorted by chosen criteria – program should display pop-up window that the information is already sorted by the chosen criteria	There is not data in the display table – program should display pop-up window that information needs to be input first

Final Prototype:

File Edit Help

Input of Scores Results Analysis

Full Name: Grade: Gender: School Year: Date of Input: Block/Class:

60 Meter Run: 200 Meter Run: 1,500 Meter Run:

1: sec. 2: sec. 1: sec. 2: sec. 1: min. sec. 2: min. sec.

Javelin (in meters): Discuss (in meters): Shotput (in meters):

1: 2: 3: 1: 2: 3: 1: 2: 3:

Long Jump (in meters):

1: 2: 3:

Submit

File Edit Help

Input of Scores Results Analysis

Sort by Year
Sort by Class
Sort by Date Input

Name	Grade	Block	School Year	Gender	Date of
------	-------	-------	-------------	--------	---------

Clear Table

Averages:

Find of the Average for: a Year a Class

Find Average of

Class/Year:

Average Calculated:

Best Scores:

Find the best score for: a Year a Class

Find Average of

Class/Year:

Best Scores Found: