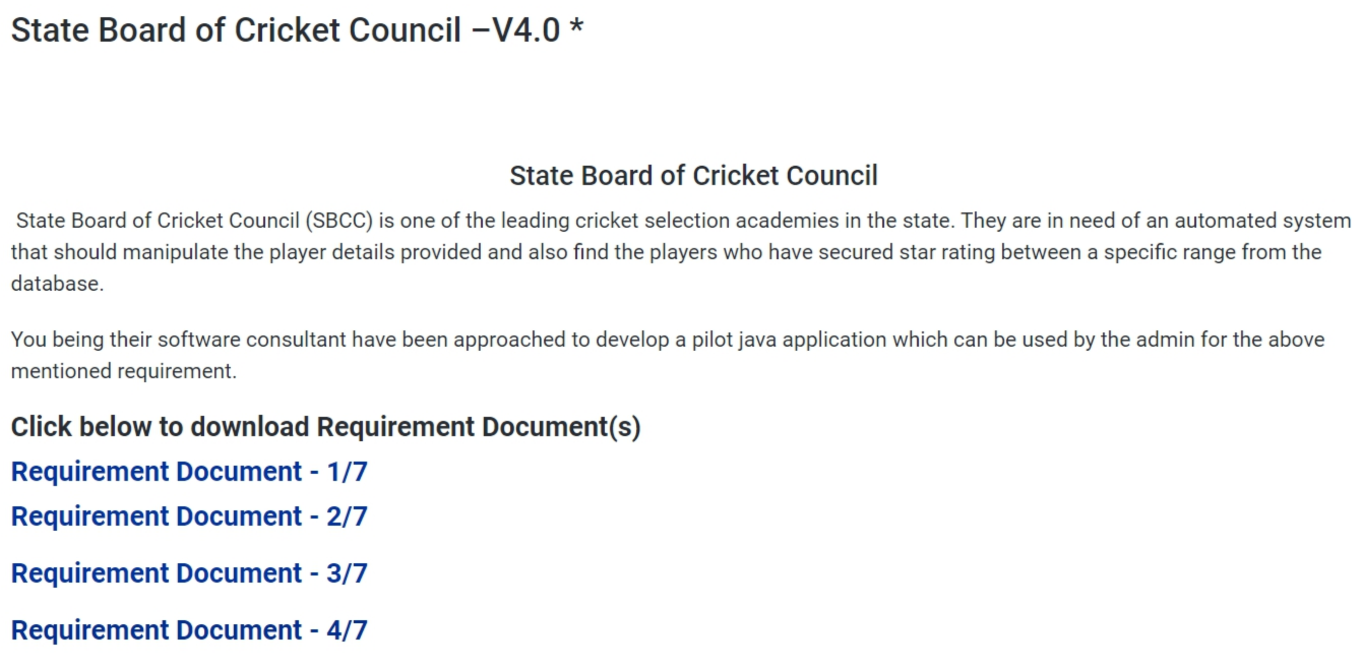

State Board of Cricket Council –V4.0 *

State Board of Cricket Council

State Board of Cricket Council (SBCC) is one of the leading cricket selection academies in the state. They are in need of an automated system that should manipulate the player details provided and also find the players who have secured star rating between a specific range from the database.

You being their software consultant have been approached to develop a pilot java application which can be used by the admin for the above mentioned requirement.

Click below to download Requirement Document(s)

Requirement Document - 1/7
Requirement Document - 2/7

Requirement Document - 3/7

Requirement Document - 4/7

State Board of Cricket Council – Running Case study

Requirement 4: Create Batsman/Bowler and find star rating

The State Board of Cricket Council (SBCC) wants the system to segregate the Players based on the player type. The player type should be either Batsman or Bowler. If the player type is “Batsman” then create a player of type Batsman. Else if the player type is “Bowler” create a player of type Bowler. You being their software consultant have been approached to integrate this functionality into the existing system.

The parsePlayerDetails method has to invoke the validatePlayerId method in the SBCCUtility class. If the playerId is valid create the player based on the player type. If the playerType is “**Batsman**” then, create a player of type Batsman and return the same. Else if the playerType is “**Bowler**” then create a player of type Bowler and return the same. If the playerId is invalid then this method has to return null.

Component Specification: SBCCUtility Class

Component Name	**Type(Class)**	**Attributes**	**Methods**	**Responsibilities**
Parse data, and Construct	SBCCUtility		public Player parsePlayerDetails(String	This method takes the String which holds all the player details as an argument. This

Player Object if the playerId is valid			playerDetails)	method has to invoke the validatePlayerId method in the SBCCUtility class by passing the playerId as a parameter, If the playerId is valid create the player based on the player type. If the playerType is "**Batsman**" then, create a player of type Batsman and return the same. Else if the playerType is "**Bowler**" then create a player of type Bowler and return the same. If the playerId is invalid then this method has to return null.
Validating the playerId	SBCCUtility		public boolean validatePlayerId(String playerId)	This method should validate the playerId, if valid return **true** else return **false**.

Include a **public abstract void findStarRating()** method in the Player class.

Component Specification: Player (Model class)

Component Name	**Type (Class)**	**Attributes**	**Methods**	**Responsibilities**
Create an abstract method	**Player**		**public abstract void findStarRating()**	
	Player	String playerId String playerName int matchesPlayed int runScored	Include all necessary Getters and Setters for all the attributes Provide a no argument	

		String playingZone	and a five argument constructor in the given order playerId, playerName, matchesPlayed, runScored and playingZone.	
Calculate total runs scored by the Player	Player		public int calculateTotalRuns(String[] securedRuns)	This method takes a String array as an argument which contains the runs scored by the player in each match. It has to calculate the total runs scored by the player by summing the runs scored by the player in each match and return the sum.

Create a concrete sub class named Batsman that inherits the Player class. The findStarRating method in the Batsman class has to calculate the rating of the player based on the number of hundreds and number of fifties scored by the batsman and set this rating value to starRating attribute in the Batsman class.

The formula for calculating the rating is as follows

Rating = ((number of hundreds * 10.0) + (number of fifties * 5.0)) * matchesPlayed / 100

For eg:

HXCB1124D:Dhoni:5:50:130:55:102:100:North:Batsman:3:2

Rating = ((3 *10.0) + (2 *5.0)) * 5 / 100 ==> 2.0

Component Specification: Batsman (Model Class)

Component Name	**Type (Class)**	**Attributes**	**Methods**	**Responsibilities**
	Batsman	int noOfHundreds int noOfFifties double starRating	Include all necessary Getters and Setters for all the attributes Provide a no argument and a seven argument constructor in the given order playerId, playerName, matchesPlayed, runScored, playingZone, noOfHundreds and noOfFifties	
concrete sub class	Batsman		public void findStarRating ()	This method has to calculate the rating of the player based on the number of hundreds and number of fifties scored by the batsman and set this rating value to the starRating attribute in the Batsman class.

Create a concrete sub class named Bowler that inherits the Player class. The findStarRating method in the Bowler class has to calculate the rating of the player based on the number of maiden overs and number of hat-trick wickets taken by the bowler and set this rating value to starRating attribute in the Bowler class.

The formula for calculating the rating is as follows

Rating = ((number of Maidens * 5.0) + (number of hattrick * 10.0)) * matchesPlayed / 100

For eg:

SAFG1243P:Mahee:3:20:30:55:South:Bowler:4:0

Rating = ((4 * 5.0) + (0 *10.0)) * 3 / 100 ==> 0.6

Component Specification: Bowler (Model Class)

Component Name	**Type (Class)**	**Attributes**	**Methods**	**Responsibilities**
	Bowler	int noOfMaiden int noOfHattrick double starRating	Include all necessary Getters and Setters for all the attributes Provide a no argument and a seven argument constructor in the given order playerId, playerName, matchesPlayed, runScored, playingZone, noOfMaiden and noOfHattrick	
concrete sub class	Bowler		public void findStarRating ()	This method has to calculate the rating of the player based on the number of maiden overs and number of hat-trick wickets taken by the bowler and set this rating value to starRating attribute in the Bowler

				class.

In the UserInterface class, **in the main method provided, fill the code to produce the output as shown in the Sample input and Output.**

When the user selects option **1 i.e., Validate player details**, it should get the player details from the user, and invoke the method to parse the player details. If valid player is returned then display the player details such as playerId, playerName, matchesPlayed, runScored and playingZone, else display "**Please provide a valid record**".

When the user selects option **2 i.e., Create Batsman or Bowler**, it should get the player details from the user, and invoke the method to parse the player details. If valid player is returned then display the player details based on the player type (Batsman or Bowler), else display "**Please provide a valid record**".

When the user selects option **3 i.e., Exit**, display the message "**Thank you for using SBCC application**" and end the program.

OVERALL DESIGN CONSTRAINTS:

  - The Player class should be inside the package com.sbcc.model
  - **The Bowler class should be inside the package com.sbcc.model**
  - **The Batsman class should be inside the package com.sbcc.model**
  - The SBCCUtility class should be inside the package com.sbcc.utility
  - The UserInterface class should be inside the package com.sbcc.main
  - Adhere to the design specifications mentioned in the case study.
  - **The classes and methods should be declared as public and all the attributes should be declared as private.**
  - **Do not change or delete the class/method/attributes, names or return types which are provided to you as a part of the base code skeleton.**
- Please make sure that your code does not have any compilation errors while submitting.

Sample Input and Output 1 [Values given in bold represents the input]:

```
1. Validate player details

2. Create Batsman or Bowler

3. Exit

Enter your choice
```

```
2
Enter the player details
HXCB1234D:Dhoni:5:50:130:55:102:100:North:Batsman:3:2
Player Id: HXCB1234D
Player Name: Dhoni
No. of matches played: 5
Total runs scored: 437
Playing zone: North
Number of Hundreds: 3
Number of Fifties: 2
Star Rating: 2.0
1. Validate player details
2. Create Batsman or Bowler
3. Exit
Enter your choice
2
Enter the player details
SAFG1243P:Mahee:3:20:30:55:South:Bowler:4:0
Player Id: SAFG1243P
Player Name: Mahee
No. of matches played: 3
Total runs scored: 105
Playing zone: South
Number of Maidens: 4
Number of Hattricks: 0
```

Star Rating: 0.6

1. Validate player details

2. Create Batsman or Bowler

3. Exit

Enter your choice

2

Enter the player details

HXC234D:Dhoni:5:20:130:55:102:100:North:Batsman:3:1

Please provide a valid record

// Note: Display the message << Please provide a valid record>> if the parsePlayerDetails method returns null

1. Validate player details

2. Create Batsman or Bowler

3. Exit

Enter your choice

1

Enter the player details

HXCB1234D:Dhoni:5:20:130:55:102:100:North:Batsman:3:1

Player id: HXCB1234D

Player name: Dhoni

No. of matches played: 5

Total runs scored: 407

Playing zone: North

1. Validate player details

2. Create Batsman or Bowler

3. Exit

Enter your choice

1

Enter the player details

HXC234D:Dhoni:5:20:130:55:102:100:North:Batsman:3:1

Please provide a valid record

// Note: Display the message << Please provide a valid record>> if the parsePlayerDetails method returns null

1. Validate player details

2. Create Batsman or Bowler

3. Exit

Enter your choice

3

Thank you for using SBCC application