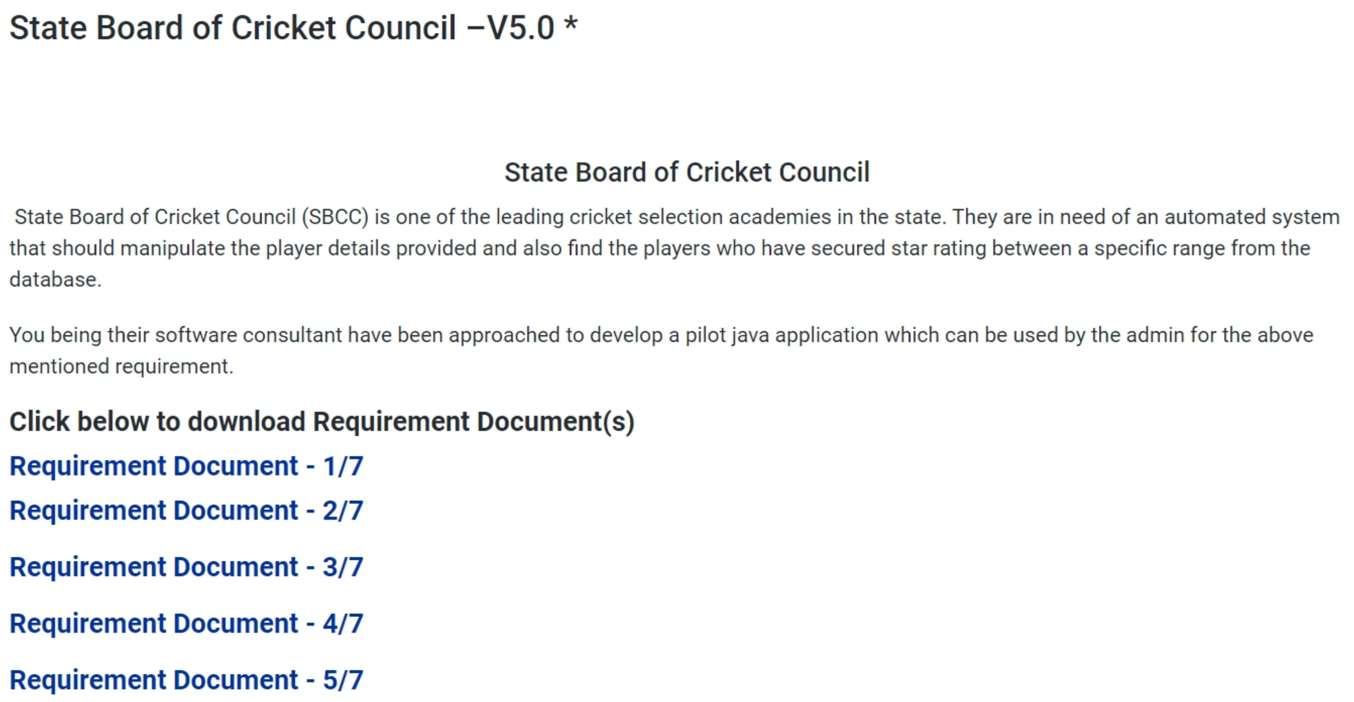

State Board of Cricket Council –V5.0 *

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

Requirement Document - 5/7

State Board of Cricket Council – Running Case study

Requirement 5: Validation with InvalidPlayerIdException

The State Board of Cricket Council (SBCC) wants the system to upgrade an old functionality with few features. Hence you being their software consultant have agreed to upgrade the existing validation functionality of the system and to integrate this functionality into the existing system.

Validation Rules:

The playerId has to be validated based on all the following constraints using the validatePlayerId method in the SBCCUtility class

1) The playerId should contain exactly 9 characters.
2) The first four characters in the playerId should be alphabets in uppercase.
3) The next four characters in the playerId should be digits.
4) The Last character in the playerId should be an alphabet in uppercase.

Examples of Valid playerId

AXCD6422D, JHDF9876J

Examples for Invalid playerId

AXC1234F – The playerId does not contain exactly 9 characters
AXC32234F – The Fourth character in the playerId is not an alphabet
ACDSH234F – The fifth character in the playerId is not a digit

ASDF87655 – The last character in the playerId is not an alphabet.
ASDF1234d – The last character in the playerId is not an alphabet in uppercase.

If the playerId is valid then this method has to return true else this method has to throw a user defined Exception “**InvalidPlayerIdException**” with a message **"Player with Id <<playerId>> is not valid".**

Component Specification: SBCCUtility Class

Component Name	**Type(Class)**	**Attributes**	**Methods**	**Responsibilities**
Validating the playerId And Exception Handling	SBCCUtility		boolean validatePlayerId(String playerId)	This method should validate the playerId, if valid return **true** else this method has to throw a user defined Exception “**InvalidPlayerIdException**” with a message **"Player with Id <<playerId>> is not valid".**

Component Specification: InvalidPlayerIdException (Exception Class)

Component Name	**Type(Class)**	**Attributes**	**Methods**	**Responsibilities**
Creating an Exception class	InvalidPlayerIdException		public InvalidPlayerIdException(String message)	

Component Specification: SBCCUtility Class

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

playerId is valid				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.

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 String playingZone	Include all necessary Getters and Setters for all the attributes Provide a no argument 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.

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.

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**" along with the exception message "**Player with Id <<playerId>> is not valid**" if invalid player details found.
When the user selects option **3 i.e., Validation with InvalidPlayerIdException**, 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**" along with the exception message "**Player with Id <<playerId>> is not valid**" if invalid player details found.
When the user selects option **4 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.service
- The UserInterface class should be inside the package com.sbcc.main
- **The InvalidPlayerIdException class should be inside the package com.sbcc.exception**
- 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 with InvalidPlayerIdException

2. Create Batsman or Bowler

3. Validation with InvalidPlayerIdException

4. Exit

Enter your choice

3

Enter the player details

HB1234D:Dhoni:5:120:50:55:198:100:North:Batsman:3:2

Player with Id HB1234D is not valid

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. Validation with InvalidPlayerIdException

4. Exit

Enter your choice

3

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. Validation with InvalidPlayerIdException
4. 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. Validation with InvalidPlayerIdException
4. Exit
Enter your choice
```

1

Enter the player details

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

Player with Id HXC234D is not valid

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

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. Validation with InvalidPlayerIdException

4. 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. Validation with InvalidPlayerIdException

4. Exit

Enter your choice

2

Enter the player details

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

Player with Id HXC234D is not valid

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. Validation with InvalidPlayerIdException

4. Exit

Enter your choice

4

Thank you for using SBCC application
```