Summer Training Report on Fantasy Cricket Game with Python

A Project Report

Submitted by

Kunal Sharma (20BCS5686)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Chandigarh University

June-July 2022

BONAFIDE CERTIFICATE

I, student of Bachelor of Engineering in Computer Science & Engineering, 5th Semester, session: June-July 2022, Chandigarh University, hereby declare that the work presented in this Project Report entitled "Fantasy Cricket Game" is the outcome of my own work, is bona fide and correct to the best of my knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patent work and has not been submitted to any other university or anywhere else for the award of any degree or any professional diploma.

Training Certificate

INTERNSHALA TRAININGS

Certificate of Training

Kunal Sharma

from Chandigarh University has successfully completed a 6-week online training on **Programming with Python**.
The training consisted of Introduction to Python, Using Variables in Python, Basics of Programming in Python, Principles of Object-oriented Programming (OOP), Connecting to SQLite Database, Developing a GUI with PyQT, Application of Python in Various Disciplines, and The Final Project modules.
In the final assessment, Kunal scored 85% marks.
We wish Kunal all the best for future endeavours.

Sarvesh Agarwal
FOUNDER & CEO, INTERNSHALA

Date of certification: 2022-06-29 Certificate no. : 1C307D95-1AB5-DDE1-E35F-1BA6DA06FACF
For certificate authentication, please visit https://trainings.internshala.com/verify_certificate

ACKNOWLEDGEMENT

It is my pleasure to be indebted to various people, who directly or indirectly helped me to learn this course and who influenced my thinking, behaviour and acts during study. I am thankful to Internshala for this course and for providing me the platform to learn and apply the skills learned through a project. I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best viable way, and I will continue to work on their improvement, to attain desired career objectives. Hope to continue cooperation with all of you in the future.e

Kunal Sharma

List of Figures

TABLE OF CONTENTS

ABSTRACT

This project aims to make Create a Fantasy Cricket game in Python. The game should have all the features displayed in the mock-up screens in the scenario. To calculate the points for each player, we can use rules similar to the sample rules displayed below. It is trivial to install on a Windows PC allowing students to take their interest further. For many the hurdle of installing a Pascal or C compiler on a Windows machine is either too expensive or too complicated. It is a flexible tool that allows both the teaching of traditional procedural programming and modern OOP; It can be used to teach a large number of transferable skills. It is a free and open-source application that is available under MIT license which can be used by anyone to modify and change as per their preference.

CHAPTER-1

Introduction

1.1 Overview

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

1.2 Background Study

Fantasy cricket is a part of the fantasy sports genre. It is an online game in which a virtual team of real cricket players is created, and points are scored depending on how those players perform in real-life matches. To win a tournament, players must work towards attaining the maximum points and the highest rank on the leaderboard.

1.3 Project Planning

Project planning is part of project management. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. The logical dependencies are defined, and the necessary resources are estimated which will be allocated to each activity. At this stage, the project plan may be optimized to achieve the appropriate balance between resource usage and project duration to comply with the project objectives. Once established and agreed, the plan becomes what is known as the baseline. Progress will be measured against the baseline throughout the life of the project.

1.4 Problem Identification

Create a Fantasy Cricket game in Python. The game should have all the features displayed in the mock-up screens in the scenario. To calculate the points for each player, we can use rules similar to the sample rules displayed below.

1.5 Task Identification

After identifying the problems, it is now time to lay out the task that the development of web-app will follow. The app is made open source for anyone to share and contribute. No login/sign-up forms were attached. Simple and aesthetically pleasing UI is to be implemented so even a tech-illiterate person can navigate effortlessly.

1.6 Software and Hardware tools required for the Project:

Hardware Required:

- Processor: Pentium IV or Above
- RAM: 2GB or above
- Hard Disk: 50GB or above
- Input Devices: Keyboard, Mouse
- Output Devices: Monitor

Software Required:

- ✓ Operating System: Linux, Ubuntu, Mac, Windows XP, 7, 8, 8.1, 10
- ✓ Software: Python 3.10
- ✓ Libraries: PyQT, Sqlite

CHAPTER-2

Literature Survey

2.1 Problem Definition

Create a Fantasy Cricket game in Python. The game should have all the features displayed in the mock-up screens in the scenario. To calculate the points for each player, we can use rules similar to the sample rules displayed below.

2.2 Objectives

The functional requirements are those requirements that are necessary to the eye of the user and the client. Here we try to make the module possible to accomplish the need of the desired function.

The objectives are as follows:

- The project will be open source so anyone will be able to contribute to it. They will also be able to take this project as base and build upon it.

- The details user adds will be saved for a period of times so that the work will not be lost because of some unprecedented circumstances.

- More sophisticated UI will also be implemented which will help user navigate more easily and thoroughly.

- Making the application free to use and ad-free by using free to use software and tools.

CHAPTER-3

Design Flow/Process

3.1 Constraints Identification

1. *Time:* Time management is a crucial part in any project. Same goes for the resume building web-app, even after defining life cycle some external factors can delay the progress of the project. So, the developers have to keep time constraints in their minds.

2. *Scope:* The project will deliver a responsive web-app made using next.js which will help users make their resume easily and effortlessly.

3. *Quality*: The web-app is designed to deliver quality resumes and but as anything requires previous experiences this is no different. So, there may be some quality constraints which may make the web-app sub-par.

4. *Risk:* Every product involves some risk throughout its life cycle. So, there can be many risks involved with this project but the most crucial one would be the use of only one template as it can be off-putting for some users.

5. *Benefits:* The main benefit of the web-app is the simple and easy to use UI which may repulse user as they might consider it too simple looking and not something extravagant.

3.2 Methodology Followed

Creating a web-application require multiple steps which are as follows:

- Designing the GUI
- Coding the GUI.
- Creating a database
- Storing information in the database
- Connecting the database to the python app.
- Adding all the functionalities

3.3 Data Flow Diagram

Data Flow Diagrams show the flow of data from external entities into the system, and from one process to another within the system. There are four symbols for drawing a DFD:

I. Rectangles representing external entities, which are sources or destinations of data.
II. Ellipses representing processes, which take data as input, validate and process it and output it.
III. Arrows representing the data flows, which can either, be electronic data or physical items.
IV. Open-ended rectangles or a Disk symbol representing data stores, including electronic stores such as databases or XML files and physical stores such as filing cabinets or stacks of paper.

Figures below are the Data Flow Diagrams for the current system. Each process within the system is first shown as a Context Level DFD and later as a Detailed DFD. The Context Level DFD provides a conceptual view of the process and its surrounding input, output and data stores. The Detailed DFD provides a more detailed and comprehensive view of the interaction among the sub-processes within the system.

Context Level Diagram

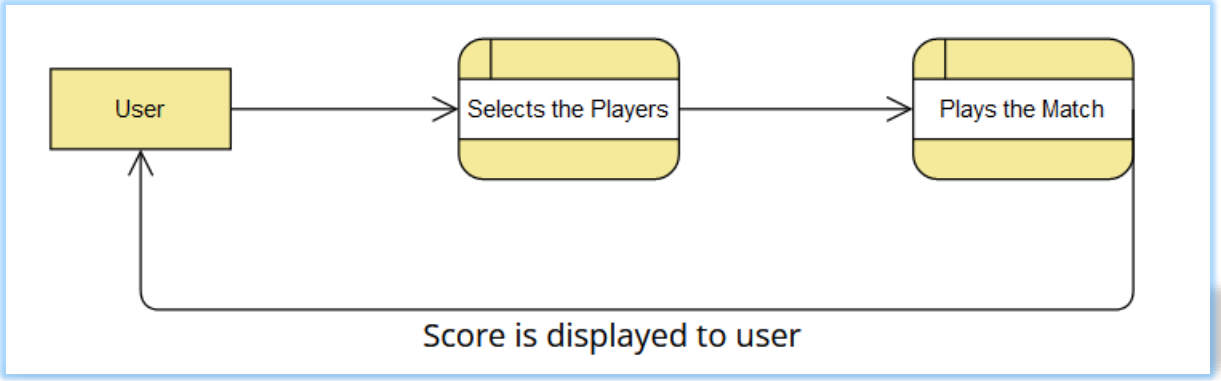

Fig.3.1

DFD Level - 1

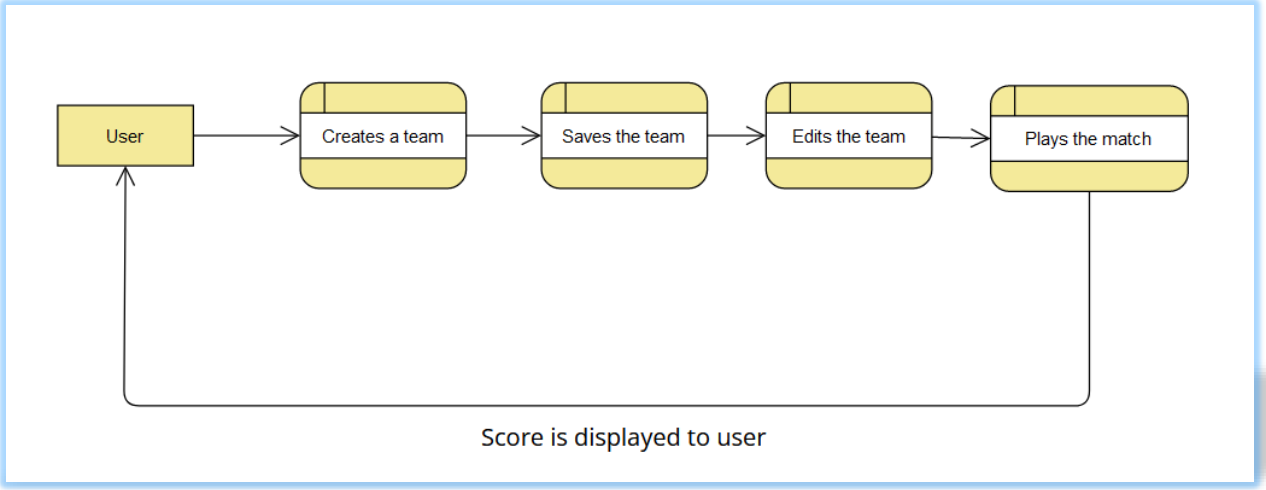

Fig.3.2

3.4 ER Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems.

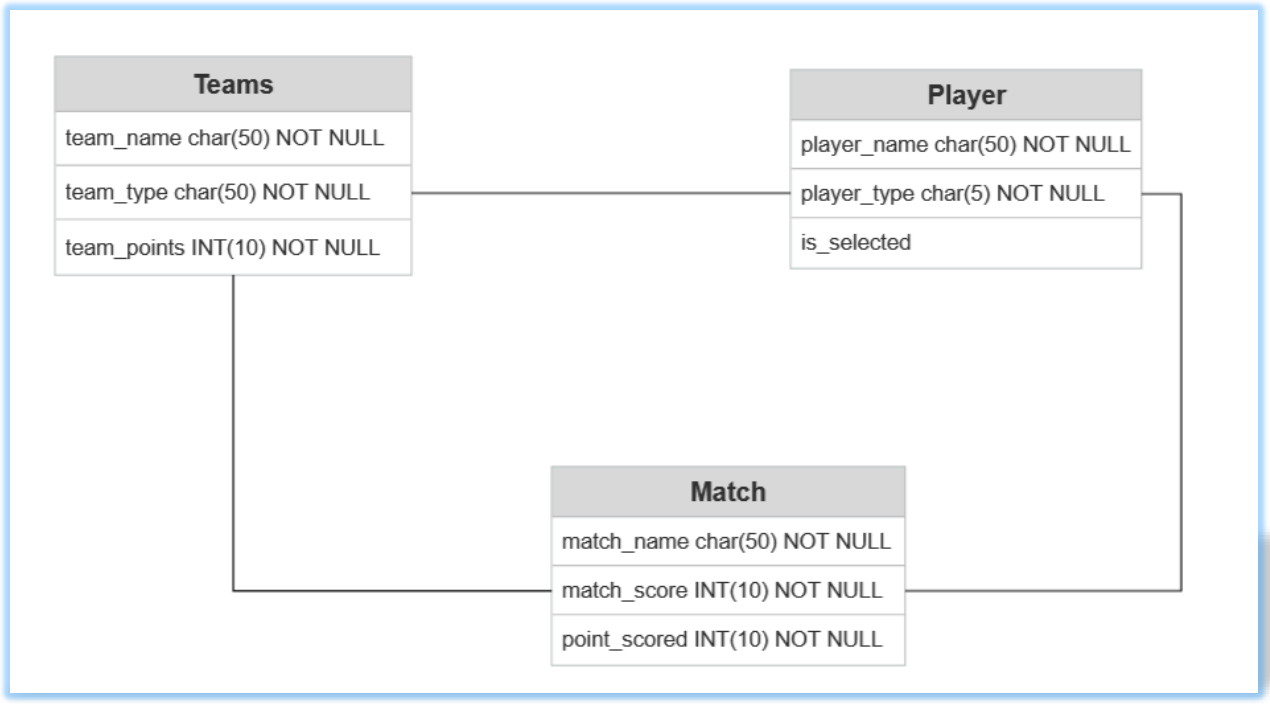
Teams
team_name char(50) NOT NULL
team_type char(50) NOT NULL
team_points INT(10) NOT NULL
Player
player_name char(50) NOT NULL
player_type char(5) NOT NULL
is_selected
Match
match_name char(50) NOT NULL
match_score INT(10) NOT NULL
point_scored INT(10) NOT NULL

Fig.3.3

3.5 Sequence Diagram

The Sequence Diagram models the collaboration of objects based on a time sequence. It shows how the objects interact with others in a particular scenario of a use case. With the advanced visual modelling capability, you can create complex sequence diagram in few clicks. Besides, Visual Paradigm can generate sequence diagram from the flow of events which you have defined in the use case description.

Actor

An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e., "role") of some entity that is relevant to the specification of its associated use cases. Thus, a single physical instance may play the role of several different actors and, conversely, a given actor may be played by multiple different instances. Since an actor is external to the subject, it is typically defined in the same classifier or package that incorporates the subject classifier.

Call Message

A message defines a particular communication between Lifelines of an Interaction. Call message is a kind of message that represents an invocation of operation of target lifeline.

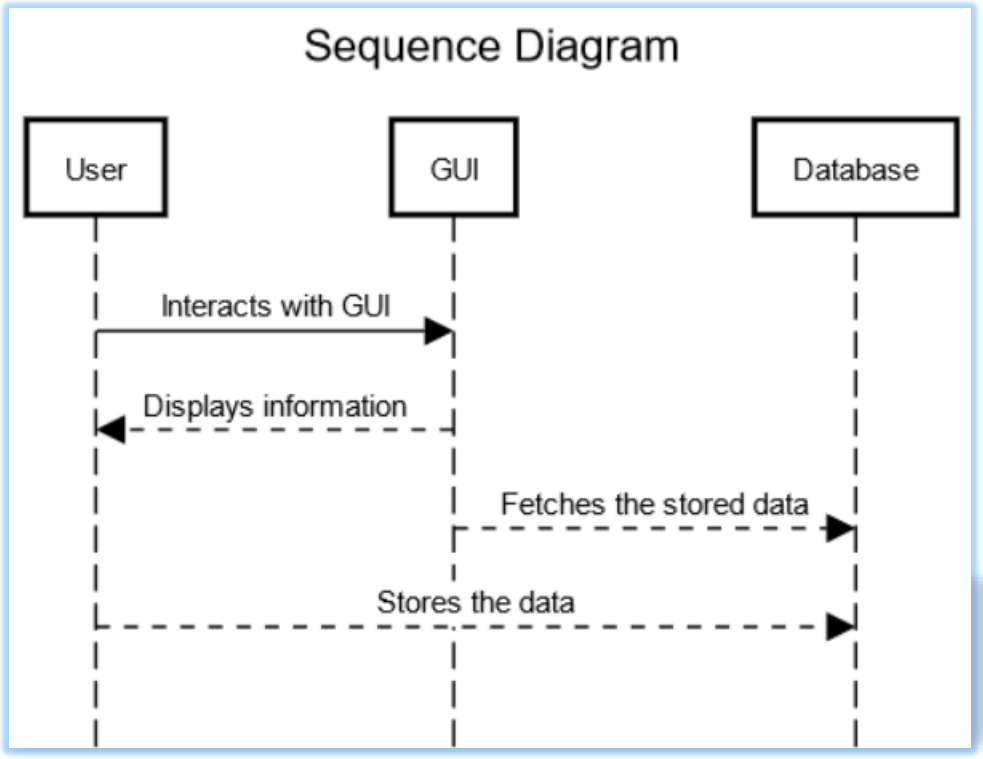

Fig.3.4

CHAPTER-4

Results and Discussion

4.1 Result:

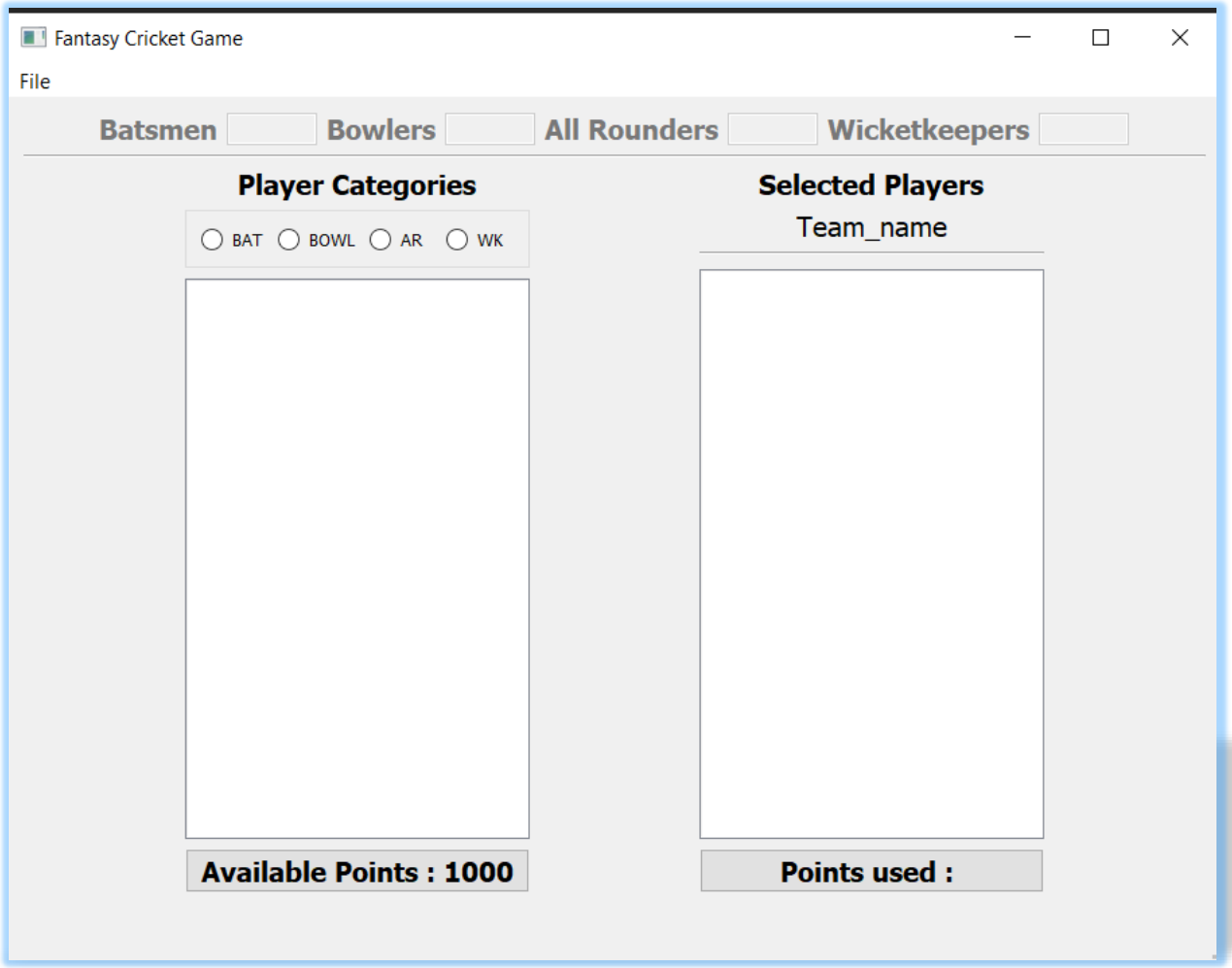

Fig.4.1

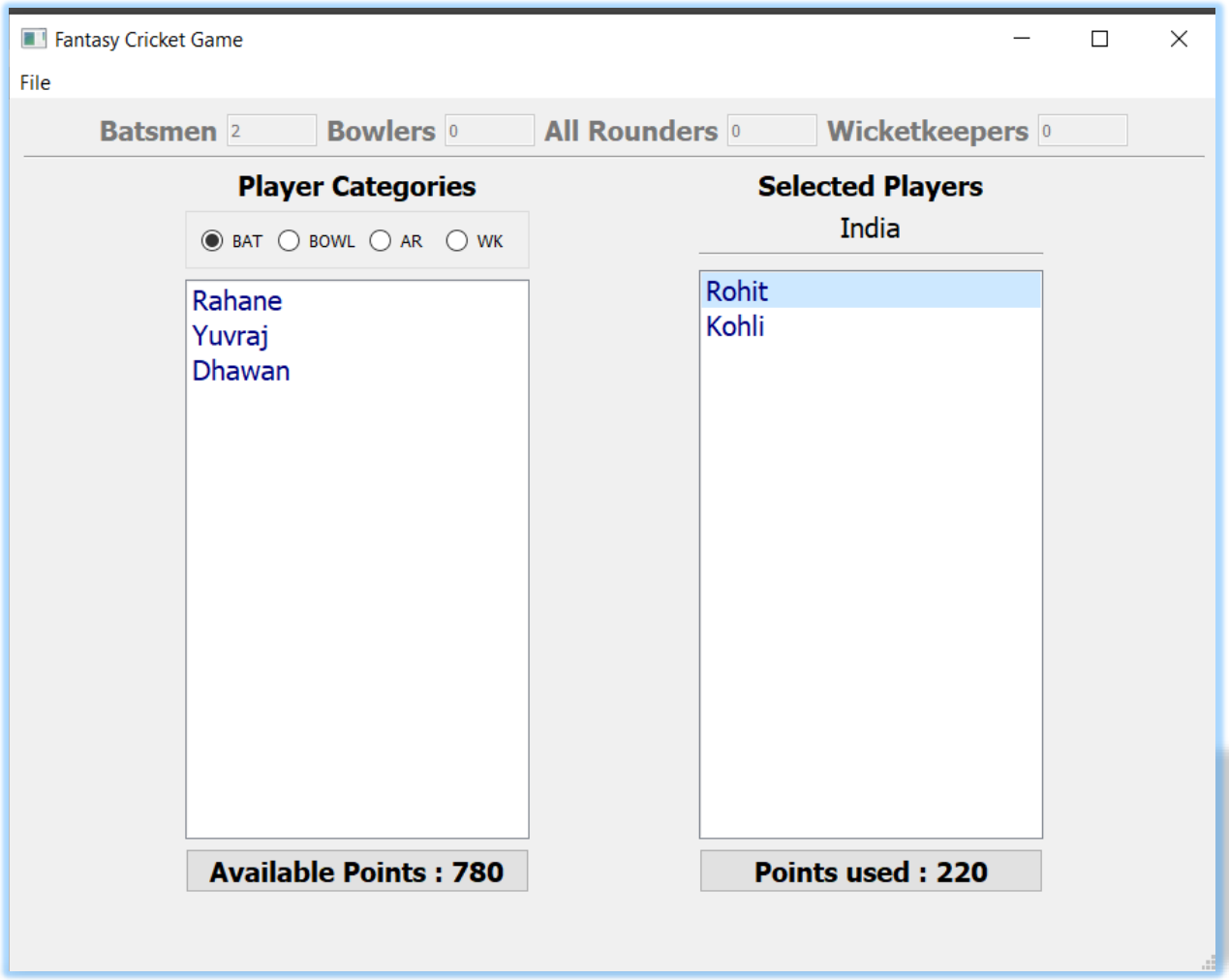
Fantasy Cricket Game
File
Batsmen 2
Bowlers 0
All Rounders 0
Wicketkeepers 0
Player Categories
BAT
BOWL
AR
WK
Rahane
Yuvraj
Dhawan
Available Points : 780
Selected Players
India
Rohit
Kohli
Points used : 220

Fig.4.2

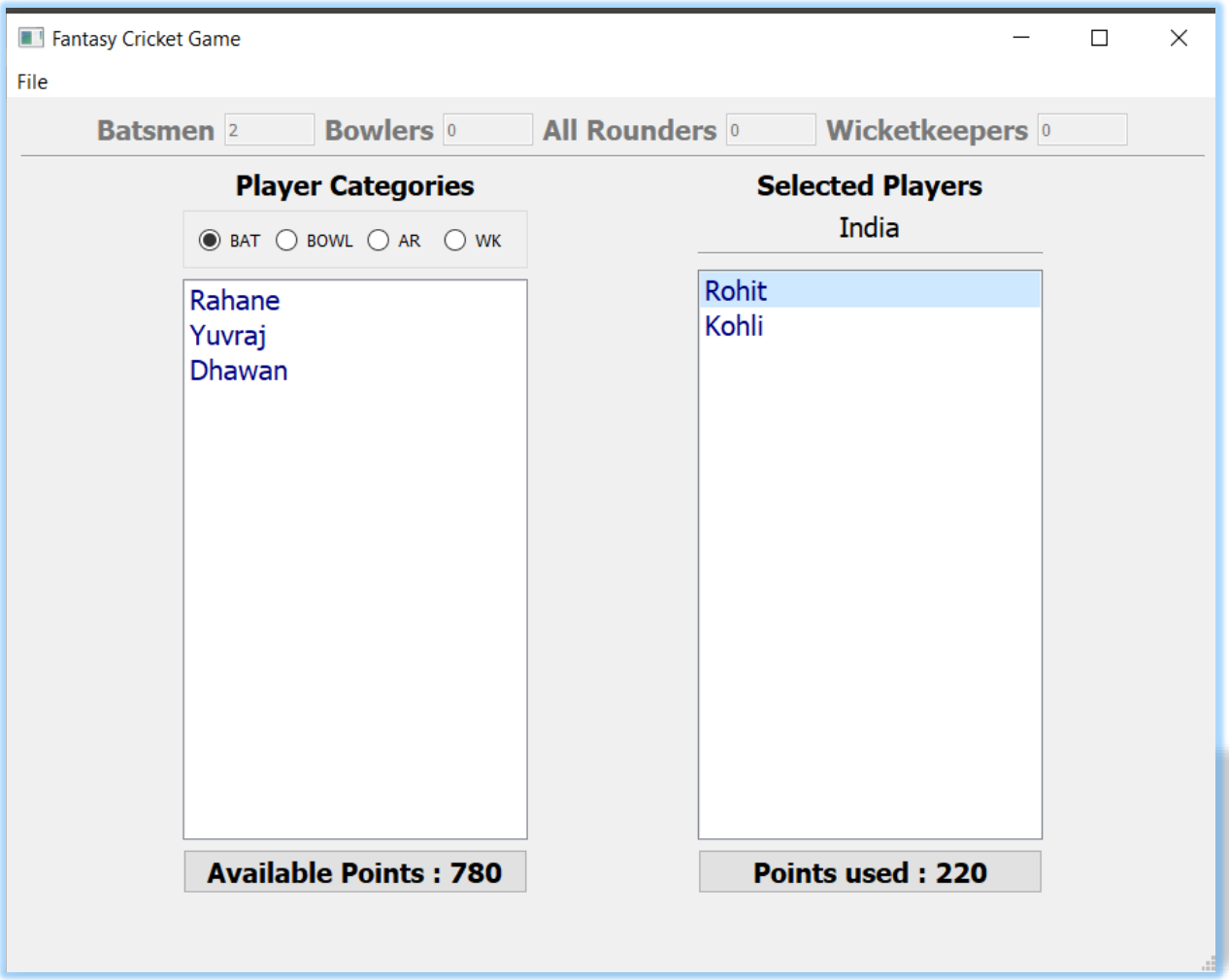
Fantasy Cricket Game
File
Batsmen 2
Bowlers 0
All Rounders 0
Wicketkeepers 0
Player Categories
BAT
BOWL
AR
WK
Rahane
Yuvraj
Dhawan
Available Points : 780
Selected Players
India
Rohit
Kohli
Points used : 220

Fig.4.3

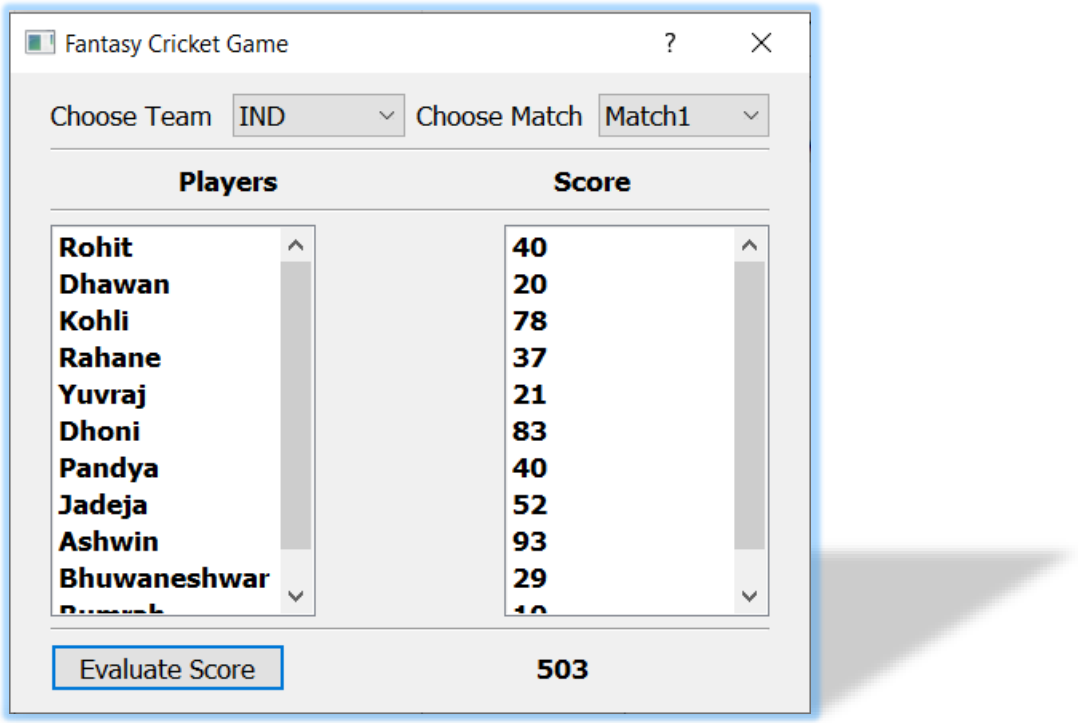
Fantasy Cricket Game
Choose Team
IND
Choose Match
Match1
Players
Score
Rohit
Dhawan
Kohli
Rahane
Yuvraj
Dhoni
Pandya
Jadeja
Ashwin
Bhuwaneshwar
40
20
78
37
21
83
40
52
93
29
Evaluate Score
503

Fig.4.4

4.2 Project Code:

Main.py:

```
from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(900, 675)
        MainWindow.move(100, 10)
        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Preferred, QtWidgets.QSizePolicy.Preferred)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(MainWindow.sizePolicy().hasHeightForWidth())
        MainWindow.setSizePolicy(sizePolicy)
        MainWindow.setContextMenuPolicy(QtCore.Qt.CustomContextMenu)
        MainWindow.setAutoFillBackground(False)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.verticalLayout = QtWidgets.QVBoxLayout(self.centralwidget)
        self.verticalLayout.setObjectName("verticalLayout")
        self.horizontalLayout = QtWidgets.QHBoxLayout()
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.horizontalLayout_5 = QtWidgets.QHBoxLayout()
        self.horizontalLayout_5.setObjectName("horizontalLayout_5")
        spacerItem = QtWidgets.QSpacerItem(250, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
        self.horizontalLayout_5.addItem(spacerItem)
        self.label_4 = QtWidgets.QLabel(self.centralwidget)
        self.label_4.setEnabled(False)
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
```

```
    self.label_4.setFont(font)
    self.label_4.setObjectName("label_4")
    self.horizontalLayout_5.addWidget(self.label_4)
    self.e1 = QtWidgets.QLineEdit(self.centralwidget)
    self.e1.setEnabled(False)
    self.e1.setObjectName("e1")
    self.horizontalLayout_5.addWidget(self.e1)
    self.label_6 = QtWidgets.QLabel(self.centralwidget)
    self.label_6.setEnabled(False)
    font = QtGui.QFont()
    font.setPointSize(12)
    font.setBold(True)
    font.setWeight(75)
    self.label_6.setFont(font)
    self.label_6.setObjectName("label_6")
    self.horizontalLayout_5.addWidget(self.label_6)
    self.e2 = QtWidgets.QLineEdit(self.centralwidget)
    self.e2.setEnabled(False)
    self.e2.setObjectName("e2")
    self.horizontalLayout_5.addWidget(self.e2)
    self.label_7 = QtWidgets.QLabel(self.centralwidget)
    self.label_7.setEnabled(False)
    font = QtGui.QFont()
    font.setPointSize(12)
    font.setBold(True)
    font.setWeight(75)
    self.label_7.setFont(font)
    self.label_7.setObjectName("label_7")
    self.horizontalLayout_5.addWidget(self.label_7)
```

```
self.e3 = QtWidgets.QLineEdit(self.centralwidget)
self.e3.setEnabled(False)
self.e3.setObjectName("e3")
self.horizontalLayout_5.addWidget(self.e3)
self.label_8 = QtWidgets.QLabel(self.centralwidget)
self.label_8.setEnabled(False)
font = QtGui.QFont()
font.setPointSize(12)
font.setBold(True)
font.setWeight(75)
self.label_8.setFont(font)
self.label_8.setObjectName("label_8")
self.horizontalLayout_5.addWidget(self.label_8)
self.e4 = QtWidgets.QLineEdit(self.centralwidget)
self.e4.setEnabled(False)
self.e4.setObjectName("e4")
self.horizontalLayout_5.addWidget(self.e4)
spacerItem1 = QtWidgets.QSpacerItem(250, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
self.horizontalLayout_5.addItem(spacerItem1)
self.horizontalLayout.addLayout(self.horizontalLayout_5)
self.verticalLayout.addLayout(self.horizontalLayout)
self.line = QtWidgets.QFrame(self.centralwidget)
self.line.setFrameShape(QtWidgets.QFrame.HLine)
self.line.setFrameShadow(QtWidgets.QFrame.Sunken)
self.line.setObjectName("line")
self.verticalLayout.addWidget(self.line)
self.horizontalLayout_3 = QtWidgets.QHBoxLayout()
self.horizontalLayout_3.setObjectName("horizontalLayout_3")
spacerItem2 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
```

```
if __name__ == "__main__":
    import sqlite3

    conn = sqlite3.connect('fantasy.db')
    import sys

    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
    conn.close()
```

evaluateTeam.py:

```
from PyQt5 import QtCore, QtGui, QtWidgets

class Mini_Ui(object):
    def setupUi(self, Dialog):
        Dialog.setObjectName("Fantasy Cricket Game")
        Dialog.resize(500, 400)
        self.verticalLayout = QtWidgets.QVBoxLayout(Dialog)
        self.verticalLayout.setContentsMargins(25, -1, 25, -1)
        self.verticalLayout.setObjectName("verticalLayout")
        self.horizontalLayout = QtWidgets.QHBoxLayout()
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.label_2 = QtWidgets.QLabel(Dialog)
        font = QtGui.QFont()
        font.setFamily("Tahoma")
        font.setPointSize(10)
        self.label_2.setFont(font)
        self.label_2.setObjectName("label_2")
        self.horizontalLayout.addWidget(self.label_2)
        self.cb0 = QtWidgets.QComboBox(Dialog)
        font = QtGui.QFont()
        font.setFamily("Tahoma")
        font.setPointSize(10)
        self.cb0.setFont(font)
        self.cb0.setObjectName("cb0")
        import sqlite3
        conn = sqlite3.connect('fantasy.db')
        self.horizontalLayout.addWidget(self.cb0)
        sql="select name from teams"
        cur=conn.execute(sql)
        teams=[]
```

```
def evaluate(self):
    import sqlite3
    conn = sqlite3.connect('fantasy.db')
    team=self.cb0.currentText()
    self.lw1.clear()
    sql1="select players, value from teams where name='"+team+"'"
    cur=conn.execute(sql1)
    row=cur.fetchone()
    selected=row[0].split(',')
    self.lw1.addItems(selected)
    teamttl=0
    self.lw2.clear()
    match=self.cb1.currentText()
    for i in range(self.lw1.count()):
        ttl, batscore, bowlscore, fieldscore=0,0,0,0
        nm=self.lw1.item(i).text()
        cursor=conn.execute("select * from "+match+" where player='"+nm+"'")
        row=cursor.fetchone()
        batscore=int(row[1]/2)
        if batscore>=50: batscore+=5
        if batscore>=100: batscore+=10
        if row[1]>0:
            sr=row[1]/row[2]
            if sr>=80 and sr<100: batscore+=2
            if sr>=100:batscore+=4
        batscore=batscore+row[3]
        batscore=batscore+2*row[4]
        bowlscore=row[8]*10
        if row[8]>=3: bowlscore=bowlscore+5
```

```
    def retranslateUi(self, Dialog):
        _translate = QtCore.QCoreApplication.translate
        Dialog.setWindowTitle(_translate("Dialog", "Fantasy Cricket Game"))
        self.label_2.setText(_translate("Dialog", "Choose Team"))
        self.label.setText(_translate("Dialog", "Choose Match"))
        self.cb1.setItemText(0, _translate("Dialog", "Match1"))
        self.label_5.setText(_translate("Dialog", "Players"))
        self.label_4.setText(_translate("Dialog", "Score"))
        self.pushButton.setText(_translate("Dialog", "Evaluate Score"))
        self.scorelabel.setText(_translate("Dialog", "---"))


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    Dialog = QtWidgets.QDialog()
    ui = Mini_Ui()
    ui.setupUi(Dialog)
    Dialog.show()
    sys.exit(app.exec_())
```

CHAPTER-5

Conclusion and Future Scope

5.1 Conclusion

The app made using Python3 is free to use and open source. This app helps user to make their own team, select their own players, play their own matches and score points. This app has a lot of potential as it can be used to play with friends as a fun past-time activity. The GUI developed using PyQT framework is very user friendly and easy to use. The database used is SqLite which is a SQL database that is easy to use and integrate in Python workflow.

5.2 Future Scope

The project has a very vast scope in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion.

The following are the future scope of the project:

- A better GUI could be implemented
- Lots of preset matched could be added.
- App could be made multilingual.
- System to play matches that are happening in real life.

References

[1] https://www.tutorialspoint.com/pyqt/

[2] https://www.tutorialspoint.com/sqlite/sqlite_quick_guide.html

[3] https://trainings.internshala.com/python-training

[4] https://www.w3schools.com/python/

[5] SequenceDiagram.org - UML Sequence Diagram Online Tool

[6] UML Class Diagram -- SmartDraw

[7] ER Diagram (ERD) - Definition & Overview | Lucidchart

[8] https://www.canva.com/