# 1. INTRODUCTION

Turf Booking System, designed to make reserving your preferred sports facility seamless and efficient. Our system allows users to easily book and manage their turf rentals online, eliminating the hassle of phone calls and in-person visits. With our intuitive interface, you can browse available facilities, select your preferred dates and times, and complete your booking in just a few clicks.

Our Turf Booking System is perfect for sports enthusiasts, coaches, and facility managers seeking to optimize their scheduling processes. Key features include real-time availability, secure payment processing, and instant confirmation notifications. Additionally, our system allows for easy cancellations and modifications, ensuring flexibility and convenience. Whether you're booking a single session or managing multiple teams, our Turf Booking System streamlines the process, saving you time and effort.

**1.1 OBJECTIVE OF THE PROJECT**

The objective of the online turf booking system, "Turfzy," is to create a streamlined and user-friendly platform that allows users to easily book nearby sports fields based on their location and preferences. By offering a centralized system, Turfzy enables users to browse available turfs by district or location, view time slots, and make real-time bookings, providing a seamless experience. This platform is designed to support informed decision-making with detailed turf information, including images, addresses, and ratings. Additionally, Turfzy equips turf owners with tools for efficient facility management, allowing them to handle bookings and monitor availability effortlessly. Through secure payment options and transparent user reviews, Turfzy aims to foster trust and accessibility, making it a reliable service for both sports enthusiasts and turf managers alike.

**2. SYSTEM ANALYSIS**

System analysis is a step-by-step process used to identify and develop or acquire the software need to control the processing of specific application. System analysis is a continuing activity the stages of the systems development. System analysis is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. The outputs from the organization are traced through the various processing that the input phases through in the organization. This involves gathering information and using structured tools for analysis. A detailed study of this process must be made by various techniques like interviews; questionnaires etc.

**2.1 EXISTING SYSTEM**

The current turf booking process is manual and phone-based, involving time-consuming steps. Users first search for turfs using online searches, local directories, or referrals. They then contact turf owners to inquire about availability and rates, which often requires multiple exchanges. Once a suitable time slot and rate are agreed upon, users confirm the booking. Payment is typically made in person after using the turf, usually in cash or other accepted methods.

**DRAWBACKS OF THE EXISTING SYSTEM**

This manual process poses several challenges, including inefficiency, inconvenience, and the potential for scheduling conflicts. It necessitates significant time and effort from users, leading to a suboptimal booking experience.

1. **Time-Intensive:** Users spend significant time searching for turfs, contacting owners, and inquiring about availability and rates.

2. **User Inconvenience:** Manual searches for turfs and contact information are cumbersome, with no centralized platform for viewing availability and rates.

**3. Limited Real-Time Availability:** Users cannot view real-time availability, relying on turf owners for updates, causing delays and potential miscommunication.

4. **Inefficient Rate Comparison:** Comparing rates across turfs requires numerous phone calls, hindering quick identification of the best rates.

5. **Miscommunication Risk:** Verbal communication increases the likelihood of misunderstandings about booking details, availability, and rates.

6. **Lack of Instant Confirmation:** Users do not receive immediate booking confirmations, leading to uncertainty and potential double bookings or conflicts.

7. **Inconvenient Payments:** In-person payments post-play add an extra step, especially if the turf lacks flexible payment options.

8. **No Centralized Record-Keeping:** Without a digital system, tracking reservations, payments, and user preferences is difficult.

9. **Limited Reviews and Ratings Access:** Users cannot easily access reviews or ratings of turfs, essential for informed booking decisions.

10. **Scalability Issues:** As demand grows, the manual process becomes harder to manage, leading to errors and user dissatisfaction.

**2.2 PROPOSED SYSTEM**

The proposed system is an online website designed for football enthusiasts to seamlessly book nearby turfs and playgrounds based on their location. It features a user-friendly interface for easy navigation and allows users to search for the nearest turfs using their current location. Real-time availability ensures users have up-to-date information on free time slots. Comprehensive turf details, including rates and features, are provided. Users can select their preferred time slot, book it, and make payments online. This system aims to streamline the turf booking process, reducing time and effort compared to manual methods, and offers a centralized platform for managing reservations efficiently.

**ADVANTAGES OF PROPOSED SYSTEM**

1. **Enhanced Convenience:** Book turfs from any location, eliminating time-consuming calls and searches.

2. **Real-Time Availability:** Access up-to-date time slot availability for accurate bookings.

3. **Comprehensive Information:** Detailed turf information, including rates and features, for informed decisions.

4. **Efficient Booking:** Quick and easy online reservations reduce effort and time.

5. **Seamless Payments:** Integrated payment system for smooth transactions.

6. **Centralized Management:** Efficient admin and turf owner management of user accounts, listings, and bookings.

7. **Improved User Experience:** User-friendly interface and intuitive navigation.

8. **Reduced Miscommunication:** Digital platform ensures clear and accurate communication.

9. **Instant Confirmation:** Immediate booking confirmations prevent double bookings.

10. **Reviews and Ratings:** Access user reviews and ratings for informed choices.

11. **Scalability:** System scales to meet increasing demand.

12. **Enhanced Record-Keeping:** Centralized, accurate digital records of bookings, payments, and interactions.

**2.3 SYSTEM REQUIREMENT SPECIFICATION**

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform. An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-worked situations.

**Problem to be Solved**

By providing the students with relevant information at hand, we have made this portal more accessible and informative. This system helps user from usual server error problems and make result checking, revaluation, and other purposes faster and easier.

**Customer requirements**

* The system should be fast.
* User friendly
* Maintaining security of data
* Efficiency in data retrieval and management

**What the developer needs to know?**

* Must know the existing system and its drawbacks.
* Must know what will be needed in the proposed system.

**Business Requirements**

The system should be feasible both to developer and client. It should effective and should be able to complete in time. Developer should be responsible for developing the system, install the software and update the software whenever necessary, conducting any user training that may be needed for using the system.

**User Requirements**

The user requirement(s) specification is a document usually that specifies the requirements the user expects from software to be constructed in a software project.

Administrator has overall control in the system.

* Admin has overall control in the system.
* Provide test result.
* Faster processing.

**Functional Requirements**

Functional requirements define what a system is supposed to do. The system should perform the following functionalities.

* Login – Login of users and turf owners.
* Registration – Users\Turf owner can register with the system.
* Edit profile– Users\ Turf owner can update their profile.
* View status – Users\ Turf owner can view status of their orders and bookings.
* Verify – Admin can verify the users and either accept or reject them.
* Logout – System users can logout from panel.

**2.3.1 Hardware Specifications**

|  |  |
| --- | --- |
| Processor | : Intel core i5 or higher Processor |
| Speed | : 3.0GHz or higher |
| System bus | : 64bits |
| Memory | : 16GB RAM or Higher |
| Hard disk | : 40GB or Higher |
| Monitor | : 14" LCD Monitor |
| Keyboard | : 104 keys |
| Pointing Device | : Two or Three Button Mouse |

**2.3.2 Software Specifications**

Operating System : Windows

Front End : HTML, CSS

Scripting Language : Java script

Back End : MySQL Server Web Server XAMP v3.3.0 , PHP,

Browser : Google Chrome

**2.3.3 Front end**

HTML

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produce Web pages that include text, graphics and pointer to other Web pages (Hyperlinks). HTML is not a programming language, but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext

and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point.

We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

**CSS**

Cascading Style Sheets (CSs) is a style sheet language used for describing the presentation of a document written in a mark-up language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML. The language can be applied to any XML document, including plain XML. SVG and XUL and is applicable to rendering in speech, or on other media.

Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification or presentation characteristics, enable multiple HTML pages to share formatting by specifying on all platforms except Windows. MySQL ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools or install MySQL Workbench via a separate download. Many third party GUl tools are also available.

**JAVASCRIPT**

JavaScript is a script-based programming language that was developed by Netscape

Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then update the browser’s display accordingly. Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Clientside programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags,

<SCRIPT>…</SCRIPT>.

<SCRIPT LANGUAGE = “JavaScript”>

JavaScript statements

</SCRIPT>

**2.3.4 Back end**

**PHP**

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994.the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page. but it now stands for the recursive backronym PHP: Hypertext Preprocessors.

PHP code may be embedded into HTML code. or it can be used in combination with various web template systems. Web content management system and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images. with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

**MYSQL**

MySQL server is powerful database, and it requires limited programs and used has back end. It supports GUI and more application is developed by help this server. Collection of tables which holds the data is called database. A beginner can create their own database by click home page. ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools or install MySQL Workbench via a separate download. Many third-party GUI tools are also available.

**2.4 FEASIBILITY ANALYSIS**

A feasibility study is an evaluation and analysis of the potential of the proposed project which is based on extensive investigation and research to give full comfort to the decision makers. Feasibility studies aim to objectively and rationally uncover the strength and weakness of existing business of proposed venture, opportunities and threads as presented by the environment, the resources required to carry through, and ultimately the process for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to attain.As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations.

The four aspects in the feasibility study are:

* Technical feasibility
* Economic feasibility
* Operational feasibility
* Behavioural feasibility

**Technical Feasibility**

The technical feasibility centres on the existing system and what extend it can support the proposed addition. The technical feasibility assessment is focused on gaining and understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. The minimum requirements of the system are met by average user. The developer system has á modest technical requirement as only minimal or null changes are required for implementing system.

Normally associated with the technical feasibility includes:

* Development risk
* Resource availability
* Technology

The proposed system can work without any additional hardware or software support other than the computer system and networks. So, I analysed that the proposed system is much more technically feasible than other systems when comparing with the benefits of the new system.

**Economic Feasibility**

Economic feasibility analysis is also known as cost/benefit analysis. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. The proposed system reduces the operating cost in terms of time by automating the process. This system is economically feasible.

**Operational Feasibility**

Operational feasibility is a measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development

**Behavioural Feasibility**

People are inherently resistant to changes and computer is known for facilitating the chances. An estimate should be made to how strongly the users react towards the e development of the system. The proposed system consumes less time. Thus, the people are made to engage in some other important work.

**2.5 DATA FLOW DIAGRAM (DFD)**

**2.5.1 Introduction to data flow diagram**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing (structured design).

Data flow diagrams were invented by Larry Constantine, the original developer of structured design based on Martin and Estrin's "data flow graph" model of computation.

Data flow diagrams (DFDs) are one of the three essential perspectives of Structured System Analysis and Design Method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users can visualize how the system will operate, what the system will accomplish. and how the system will be implemented. The old system's data flow diagrams can be drawn up and compared with the new system's data flow diagrams to drawn comparisons to implement a more efficient system. Data flow diagrams can be used to provide the end user with physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data flow diagram.

Developing a data flow diagram helps in identifying the transaction data in the data model. There are different notations to draw data flow diagrams, defining different visual representation for process, data stores, data flow, and external entities. The first step is to draw a data flow diagram (DFD). A DFD also known as "bubble chart" has the purpose of clarifying system requirements and identifying major transformation that will become program in system design. So, it is starting point of the design phase that function that functionally decompose the

requirements specification down to the lowest level of details DFD consists of series of bubbles joined by lines. The bubbles represent data transformation and the Iines represent data flow in the system.

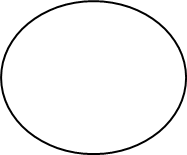
DFD Symbols: -

* Square- Defines source or destination of system.

A white background with black border

Description automatically generated

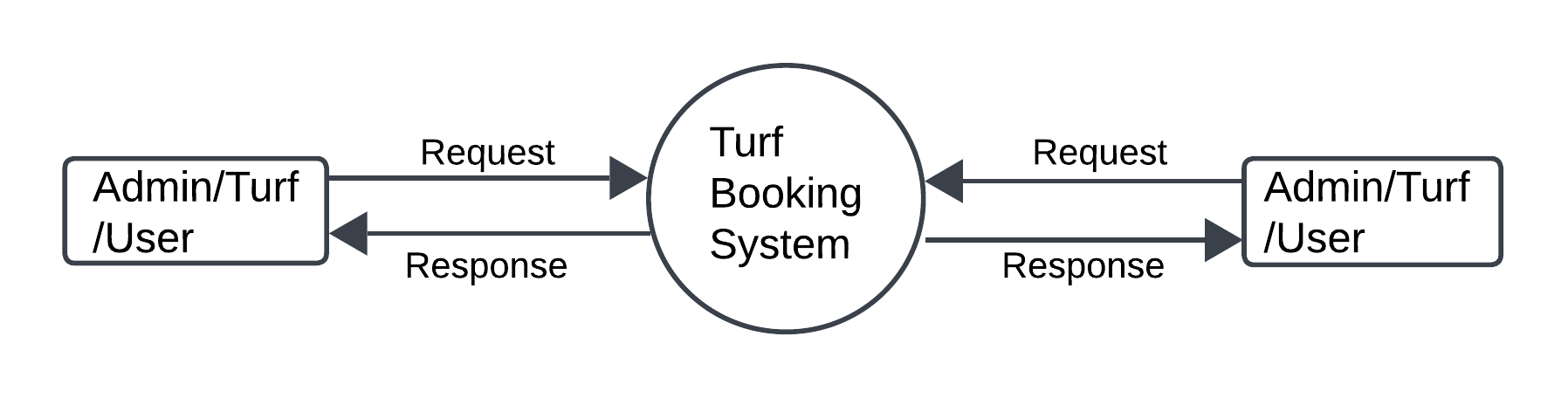
* Data flow - Identifies data flow Circle.
* Bubble - Represents a process that transforms incoming data to outgoing data.



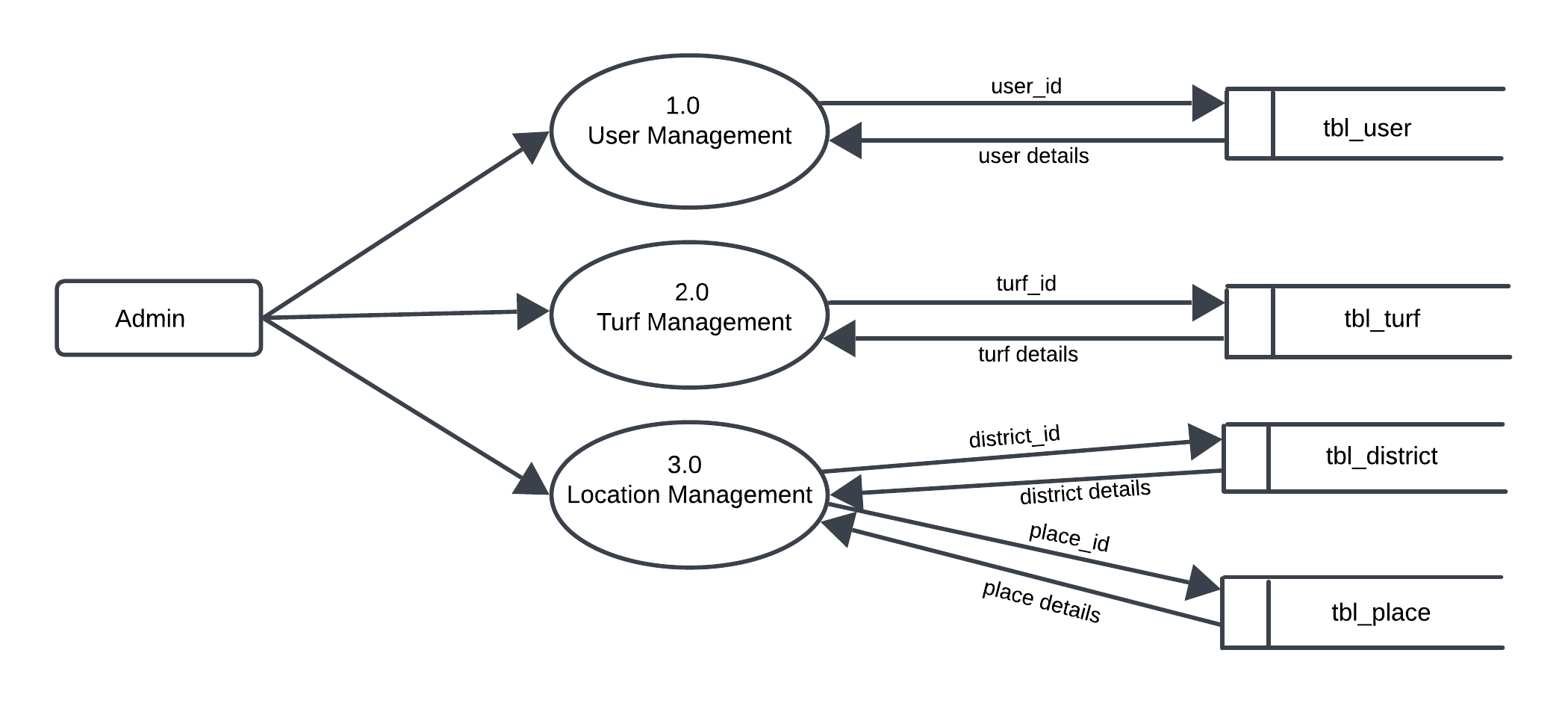
* Open rectangle- Data store

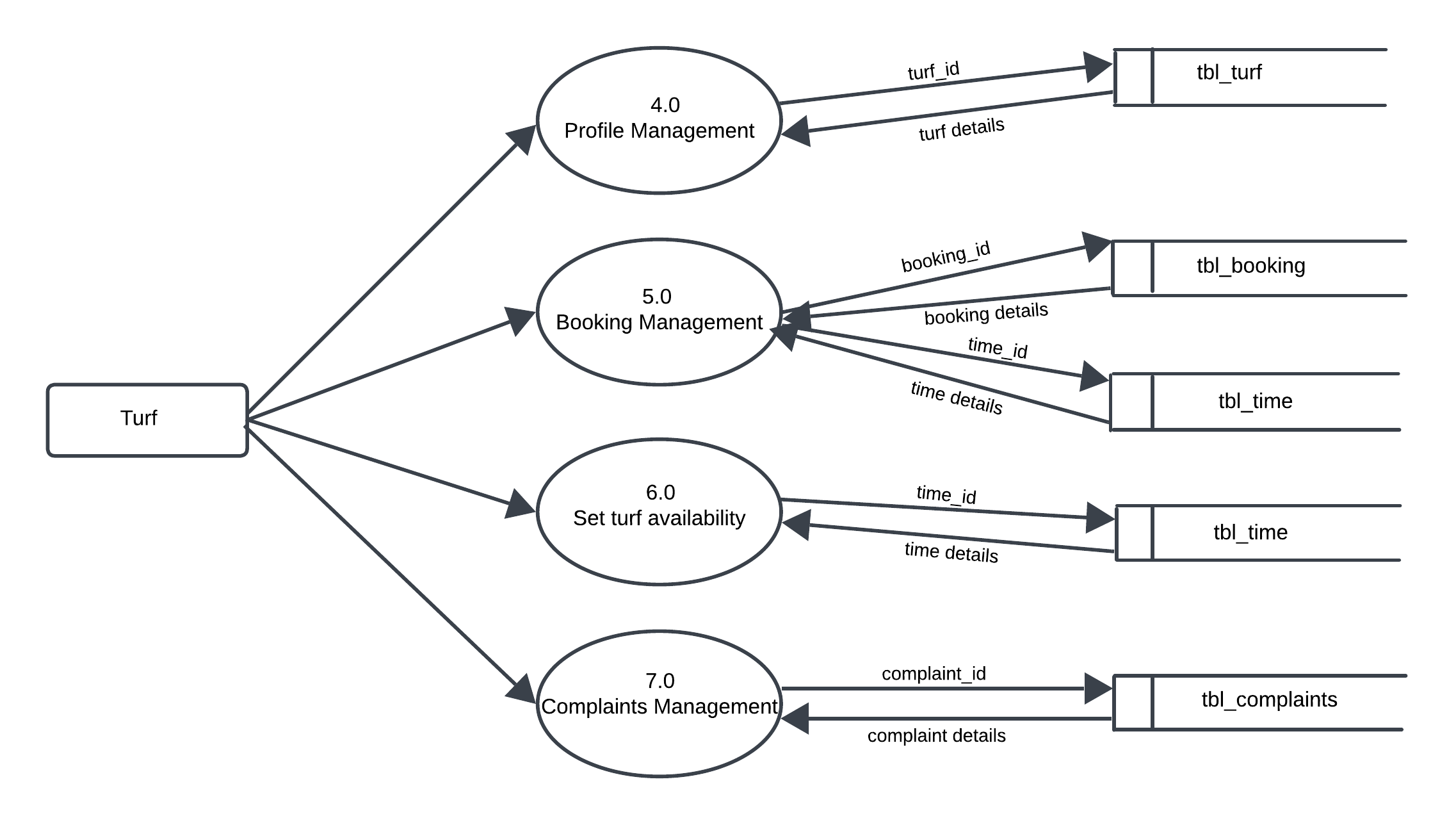
|  |  |
| --- | --- |
|  |  |

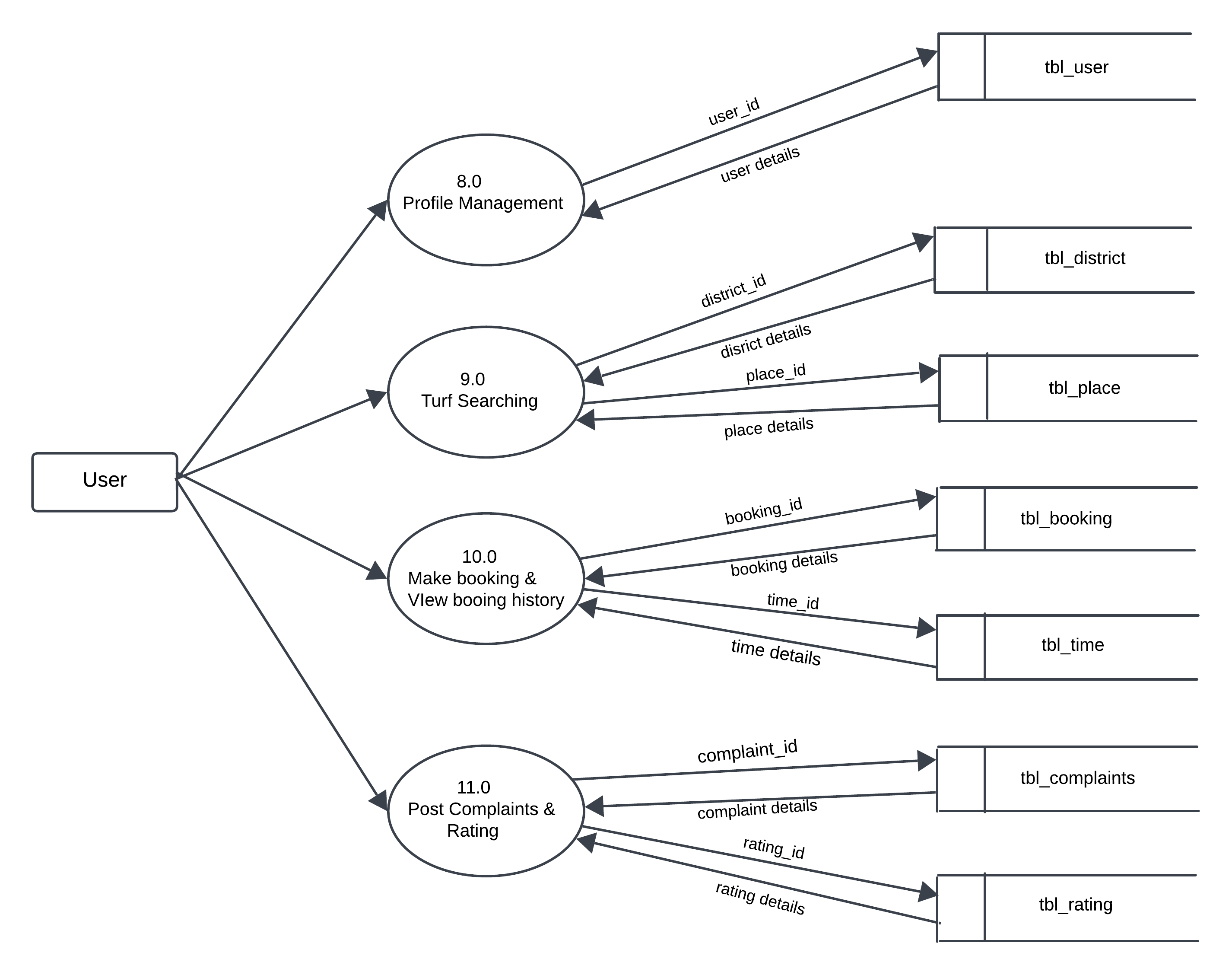
**LEVEL 0**



**LEVEL 1**

****

****

****

**3. SYSTEM DESIGN**

**3.1 INPUT DESIGN**

The quality of the system input determines the quality of the system output. Input specification describes the way data enter the system for processing. Input design features can ensure the reliability of the system and produce result from accurate data, or they can result in the production or erroneous information. The input design also determines whether the user can interact efficiently with the system.

In our system almost all inputs are being taken from the databases. To provide adequate inputs we have to select necessary values from the databases and arrange it to the appropriate controls.

**Admin**

Admin is the one who controls the system. The admin can access the page using username and password. Admin can add districts, place and type.

**User**

The user of the system is the customer. Users can create their profile and upload their details.

User can search and book turfs. They can also post if any complaints and rate the turfs.

**Turfs**

Turfs can register their profiles and time slots and amount. They can also reply to the user complaints.

**3.2 OUTPUT DESIGN**

One of the important features of an information system for users is the output produces. Output is the information delivered to users through the information system. Without quality of the output, the entire system appears to be unnecessary that users will avoid using it. Users generally merit the system solely by its output. In order to create the most useful output possible. One works closely with the user though an interactive process.

**Admin**

Admin can verify the turf and accept it base on the proof and can also view the users.

**User**

User can view booking history and status. User can post complaints and rating.

**3.3 TABLE DESIGN**

The data design transforms the information domain model created during analysis into the data structures that will be required to implement the software. The data objects and relationships defined in the entity relationship diagram and the detailed data content depicted in the data dictionary provide the basis for the data design activity.

The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. Database Management System allows data to be protected and organized separately from other resources. Database is a integrated collection of data. This is the difference between logical and physical data.

The organization of data in the database aims to achieve three major objectives:

* Data integration
* Data integrity
* Data independence

The databases are implemented using a DBMS package. Each DBMS has unique characteristics and general techniques for database design. There are 6 major steps in design process. The first 5 steps are usually done on paper and finally the design is implemented.

* Identify the table and relationships.
* Identify the data that is needed for each table and relationship.
* Resolve the relationship.
* Verify the design.
* Implement the design.

The database uses tables for storage. A table also contains records, which is a set of fields. All records, in a table have the same set of fields with different information. Uses 3 tables.

Each table contains key fields that establish relationships in the database and how the records are stored. There are primary key fields that uniquely identify a record in a table. There are also fields that contain the primary key from another table called foreign keys.

The various database tables that are used in this project are the following:

1. tbl\_admin

Primary Key: admin\_id

Description: This table shows the information about the details of the admin

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD NAME | DATA TYPE | SIZE | DESCRIPTION |
| admin\_id | INT |  | Unique ID of Admin |
| admin\_email | VARCHAR | 50 | Email of Admin |
| admin\_password | VARCHAR | 50 | Password of Admin |
| admin\_name | VARCHAR | 50 | Name of Admin |

2. tbl\_district

Primary Key: district\_id

Description: This table shows the information about the details of district.

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD NAME | DATA TYPE | SIZE | DESCRIPTION |
| district\_id | INT |  | Unique ID of District |
| district\_name | VARCHAR | 50 | Name of District |

3. tbl\_place

Primary Key: place\_id

Foreign Key: district\_id

Description: This table shows the information about the details of place.

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD NAME | DATA TYPE | SIZE | DESCRIPTION |
| place\_id | INT |  | Unique ID of Place |
| district\_id | INT |  | Unique ID of District |
| place\_name | VARCHAR | 50 | Name of Place |
| place\_pincode | INT |  | Pincode of place |

4. tbl\_turf

Primary Key: turf\_id

Foreign key: place\_id

Description: This table shows the information about the details of turf.

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD NAME | DATA TYPE | SIZE | DESCRIPTION |
| turf\_id | INT |  | Unique ID of Turf |
| turf\_name | VARCHAR | 50 | Name of Turf |
| turf\_email | VARCHAR | 50 | Email of Turf |
| turf\_password | VARCHAR | 50 | Password of Turf |
| turf\_address | VARCHAR | 100 | Address of Turf |
| place\_id | INT |  | Unique ID of Place |
| turf\_proof | VARCHAR | 300 | Proof of Turf |
| turf\_photo | VARCHAR | 300 | Photo of Turf |
| turf\_vstatus | VARCHAR | 50 | Turf status |
| type\_id | INT |  | Unique ID of turf type |

5. tbl\_user

Primary Key: user\_id

Description: This table shows the information about the details of user.

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD NAME | DATA TYPE | SIZE | DESCRIPTION |
| user\_id | INT |  | Unique ID of user |
| user\_name | VARCHAR | 30 | Name of user |
| user\_email | VARCHAR | 30 | Email of user |
| user\_password | VARCHAR | 30 | Password of user |
| User\_contact | VARCHAR | 30 | Contact number of user |

6. tbl\_booking

Primary key: booking\_id

Foreign Key: time\_id, user\_id, turf\_id

Description: This table shows the information about the details of booking.

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD NAME | DATA TYPE | SIZE | DESCRIPTION |
| booking\_id | INT |  | Unique ID of Booking |
| booking\_date | VARCHAR | 30 | Date of Booking |
| time\_id | INT |  | Unique ID of Time |
| user\_id | INT |  | Unique ID of User |
| turf\_id | INT |  | Unique ID of Turf |
| booking\_vstatus | int |  | Status of Booking |
| Booking\_amount | VARCHAR | 50 | The Amount of Booking |

7. tbl\_complaint

Primary Key: complaint\_id

Foreign Key: user\_id, turf\_id

Description: This table shows the information about the details of complaint.

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD NAME | DATA TYPE | SIZE | DESCRIPTION |
| complaint\_id | INT |  | Unique ID of Complaint |
| complaint\_title | VARCHAR | 30 | Title of Complaint |
| complaint\_content | VARCHAR | 200 | Content of Complaint |
| complaint\_reply | VARCHAR | 200 | Reply of Complaint |
| user\_id | INT |  | Unique ID of User |
| turf\_id | INT |  | Unique ID of Turf |

8. tbl\_rating

Primary Key: rating\_id

Foreign Key: turf\_id, user\_id

Description: This table shows the information about the details of rating.

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD NAME | DATA TYPE | SIZE | DESCRIPTION |
| rating\_id | INT |  | Unique ID of rating |
| rating\_value | VARCHAR | 50 | Control of rating |
| rating\_content | VARCHAR | 200 | Content of rating |
| rating\_datetime | VARCHAR | 30 | Date of rating |
| turf\_id | INT |  | Unique ID of turf |
| user\_id | INT |  | Unique ID of user |

9. tbl\_time

Primary Key: time\_id

Foreign Key: turf\_id

Description: This table shows the information about the details of time.

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD NAME | DATATYPE | SIZE | DESCRIPTION |
| time\_id | INT |  | Unique ID of Time |
| turf\_id | INT |  | Unique ID of turf |
| time\_start | VARCHAR | 30 | Starting time |
| time\_end | VARCHAR | 30 | Ending time |
| time\_amount | INT |  | Amount for time |

10. tbl\_type

Primary Key: type\_id

Description: This table shows the information about the type of the turf.

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD NAME | DATATYPE | SIZE | DESCRIPTION |
| type\_id | INT |  | Unique ID of type |
| type\_name | VARCHAR | 30 | Name of the type |

**4. SYSTEM IMPLEMENTATION AND TESTING**

**4.1 SYSTEM TESTING**

Testing is the process of examining the software to compare the actual behaviour with that of the excepted behaviour. The major goal of software testing is to demonstrate that faults are not present. In order to achieve this goal, the tester executes the program with the intent of finding errors. Though testing cannot show absence of errors but by not showing their presence it is considered that these are not present.

System testing is defined as the process by which one detects the defects in the software. Any software development organization or team has to perform several processes. Software testing is one among them. It is the final opportunity of any programmer to detect and rectify any defects that may have appeared during the software development stage. Testing is a process of testing a program with the explicit intention of finding errors that makes the program fail. In short system testing and quality assurance is a review in software products and related documentation for completion, correctness, reliability and maintainability.

System testing is the first stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct and the goal will be successfully achieved. A series of testing are performed for the proposed system before the proposed system is ready for user acceptance testing. The testing steps are,

* Unit testing
* Integration testing
* Acceptance Testing
* Validation

• Output testing

System Testing provides the file assurance that software once validated mast combined with all other elements. System testing verifies whether all elements have been combined

properly and that overall system function and performance is achieved. FA the integration of modules, the validation test was carried out over the system. It was that all the modules work well together and meet the overall system function and performance.

**1. Unit Testing**

Unit testing is caried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors. This has enabled the detection of errors in coding and logic.

Various test cases are prepared. For each module these test cases are implemented, and it is checked whether the module is executed as per the requirements and outputs the desired result.

In this test each service input and output parameters are checked.

In unit testing

* Module interface was tested to ensure that information properly flows into and out of the program under test.
* Boundary condition was tested to ensure that module operates properly at boundaries established to limit or restrict processing.
* All independent paths through the control structures were executed to ensure that all statements in the modules have been executed at least once.
* Error handling paths were also tested.

**2. Integration Testing**

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing.

Unit tested module were taken and a single program structure was built that has been dictated by the design. Incremental integration has been adopted here.

The modules are tested separately lor accuracy and modules are integrated too.th tn. using bottom-up integration i.e., by integrating from moving from bottom to the top of the system is

checked and errors found during integration are rectified. In this testing individual modules were combined and he m0duie wise Shifting was verified to be alright.

The entire software was developed and tested in small segments, where errors were easy to locate and rectify. Program builds (group of modules) were constructed corresponding to the successful testing of user interaction, data manipulation analysis, and display processing and database management.

**3.Validation Testing**

Validation testing is done to ensure complete assembly of the error-free software. Validation can be termed successful only if it functions in manner. Reasonably expected by the student under validation is alpha and beta testing. The student-side validation is done in this testing phase. It is checked whether the data passed to each student is valid or not. Entering incorrect values does the validation testing and it is checked whether the errors are being considered. Incorrect values are to be discarded. The errors are rectified.

In “Turf Booking System" verifications are done correctly. So, there is no chance for users to enter incorrect values. It will give error messages by using different validations. The validation testing is done very clearly and found it is error free.

**4. Output design**

After performing the validation testing the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format.

The output format on the screen was found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, the output comes out as specified requirement by the user. Hence output testing does not result in any Correction in the system.

Output This project is developed based on the user choice. It is user friendly. The output format is very clear to user. Output testing is done on Turf Booking System correctly.

**5. Acceptance Testing**

Acceptance involves running a suite of tests on the completed system. Each individual test, known as a Case, exercise particular operating condition of the operating condition of the user's environment or feature of the system, and will result in a pass fail, or Boolean outcome.

**4.2 SYSTEM IMPLEMENTATION**

The implementation is the final state, and it is an important phase. It involves the invalid programming system testing. user training and the operational running of developed proposed system that constitutes the application subsystems. A major task of preparing for implementation is education of users. which should really have been taken place much carrier in the project when they were belong involved in the investigation and design work. During the implementation phase system take physical shape. In order to develop a system implemented planning is very essential.

The implementation phase of the software development is concerned with translating design specification into source code. The user tests the developed system and changes are made according to their needs. Our system has been successfully implemented.

Before implementation several tests have been conducted to ensure that no errors are encountered during the operation. The implementation phase ends with an evaluation of the system after placing into the operation for a period of time.

The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from old system to new system. The system can be implemented only after testing is done and is found to be working to specifications. The implementation stage is a systems project in its own right.

The implementation stage involves following tasks:

* Careful planning.
* Design of method to achieve change over.
* Evaluation of the changeover method.
* Investigation of system and constraints

In the case of this project all the screens are designed first. For making it to be executable, codes are written on each screen and performs the implementation by creating the database and connecting to the server. After that the system, is Checked, whether it performs all the transactions correctly. Then databases are cleared and made it to be usable to the technicians.

**Implementation Plans**

The following are the step involved in the implementation plan of " Turf Booking System":

* Test system with sample data.
* Detection and correction of errors
* Make the necessary changes in the system.
* Check the existing system.
* Installation of hardware and software utilities.
* Training and involvement of user personals.

# 5.SECURITY TECHNOLOGIES& POLICIES

The protection of computer-based resources that includes hardware, software, data procedures and people against unauthorized use or natural disaster is known as System Security.

System Security can be divided into four related issues:

* Security
* Integrity
* Privacy
* Confidentiality

**SYSTEM SECURITY** refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

**DATA SECURITY** is the protection of data from loss, disclosure, modification and destruction.

**SYSTEM INTEGRITY** refers to the power functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

**PRIVACY** defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

**CONFIDENTIALITY** is a special status given to sensitive information in a database to minimize the possible invasion of privacy. lt is an attribute of information that characterizes its need for protection.

**SECURITY IN SOFTWARE** System security refers to various validations on data in form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered, and only valid operations are performed on the system.

The system employees two types check and controls:

**CLIENT-SIDE VALIDATION** Various client-side validations are used to ensure on the client side that only valid data is entered. Client-side validation saves server time and load to handle invalid data. Some checks imposed are:

* Forms cannot be submitted without filling up the mandatory data so that manual

mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.

* Tab-indexes are set according to the need and taking into account the ease of user while working with the system.

**SERVER-SIDE VALIDATION** Some checks cannot be applied at client side. Server-side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server-side checks imposed is:

* Server-side constraint has been imposed to check for the validity of primary key and

foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results into a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.

* User is intimating through appropriate messages about the successful operations or exceptions occurring at server side.
* Various Access Control Mechanisms have been built so that one user may not agitate

upon another. Access permissions to various types of users are controlled according to the organizational structure. Only permitted users can log on to the system and can have access according to their category. User- name, passwords and permissions are controlled on the server side.

# 6. MAINTENANCE

Software maintenance is the modification of a software product and delivery to correct faults, to improve performance or other attributes. Maintenance is the ease with which a program can be corrected if any error is encountered, adapted if its environment changes or enhanced if the customer desires a change in requirement. Maintenance follows conversation to extend that changes are necessary to maintain satisfactory operations relative to changes in the user's environment.

Maintenance often includes minor enhancements or corrections to problems that surface in the system's operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

**CATEGORIES OF MAINTENANCE**

Corrective Maintenance

Corrective maintenance is the most used maintenance approach, but it is easy to see its limitations. When equipment fails, it often leads to downtime in production, and sometimes damages other parts. In most cases, this is expensive. Also, if the equipment needs to be replaced, the cost of replacing it alone can be substantial. Reliability of systems maintained by this type of maintenance is unknown and cannot be measured. Corrective maintenance is possible since the consequences of failure or wearing out are not significant and the cost of this maintenance is not great.

Perfective Maintenance

Modification of a software product alter delivery to improve performance or maintainability. This term is used to describe changes undertaken to expand the existing requirements of the system. A successful piece or software lends to be subjected to the Succession of changes resulting in an increase in us requirements. This is based on premise that as the software becomes useful, the user experiment with new cases beyond the of Scope for which it was initially developed. Vxpansi01 n requirements can take the form enhancement of existing system functionality and improvement in computational efficiency.

Adaptive Maintenance

Modification of a software product performed after delivery to keep a product usable, changed or changing environment. Adaptive maintenance includes any work initiated because of moving the software to a different hardware or software platform. It is a change driven by the need to accommodate modifications in the environment of software system.

The environment in this context refers to the totality of all conditions and influences which act from outside upon the system. A change to the whole or part of this environment will Warrant a corresponding modification of the software.

Preventive Maintenance

Preventive maintenance is a schedule of planned maintenance actions aimed at the prevention of breakdowns and failures. The primary goal of preventive maintenance is to prevent the failure of equipment before it occurs. It is designed to preserve and enhance equipment reliability by replacing worn components before they fail. Preventive maintenance activities include equipment checks, partial or complete overhauls at specified periods.

Long-term benefits of preventive maintenance include:

* Improved system reliability.
* Decreased cost of replacement.
* Decreased system downtime.

**7. SCOPE FOR FUTURE ENHANCEMENT**

The system has been designed in such a way that it can be modified with very little effort when such needs arise in the future. New features can be added with slight modifications of software which make it easy to expand the scope of this project. Though the system is working on various assumptions, it can be modified easily to any kind of requirements. The system is also expected to be improvised by adding various features. One of the future enhancements is that the user can raise their complaints and queries through online and the admin and turf owner can respond to their queries

The future enhancements for the turf booking website aim to significantly improve user experience, streamline booking processes, and foster community engagement. Key developments will include a modernized user interface with mobile optimization for seamless access, real-time availability tracking, and diverse payment options to cater to a broader audience. Enhanced user profiles will allow for personalized experiences, while a robust feedback and rating system will encourage transparency and trust between users and turf owners. Additionally, integrating social media features and advanced search functionalities will make it easier for users to find and book their preferred turfs. We also plan to implement event management tools, enabling users to organize tournaments and register teams directly through the platform. By prioritizing these enhancements, we aim to create a dynamic and user-friendly environment that meets the evolving needs of our community.

**8.CONCLUSION**

In conclusion, our turf booking website is designed to provide a seamless and enjoyable experience for all users, whether you're a seasoned turf enthusiast or a newcomer. With an intuitive interface, real-time updates, and a variety of booking options, we make it easy to secure your spot on the turf. Our commitment to quality service and user satisfaction ensures that you can focus on what matters most—enjoying your time on the field. Join our community today and experience the thrill of turf booking like never before!

**9. BIBLIOGRAPHY**

1. Jalotte, Pankaj. " **SOFTWARE ENGINEERING",** Third Edition, Spring Publishers, Narosa Publishing House.
2. Ramakrishnan, Raghu. and\_ Gehrke, Johannes. **DATABASE MANAGEMENT** **SYSTEM**", Third Edition, McGraw-Hill Professional Publication.
3. Ramus Leodorf and Kevin Tatore. “**PROGRAMMING PHP** ”
4. Bayross, Ivan. "**SQL/MySQL**" Second Revised Edition, BPB Publications
5. https://www.w3schools.com
6. Template https://themewagon.com
7. https://www.geeksforgeeks.org

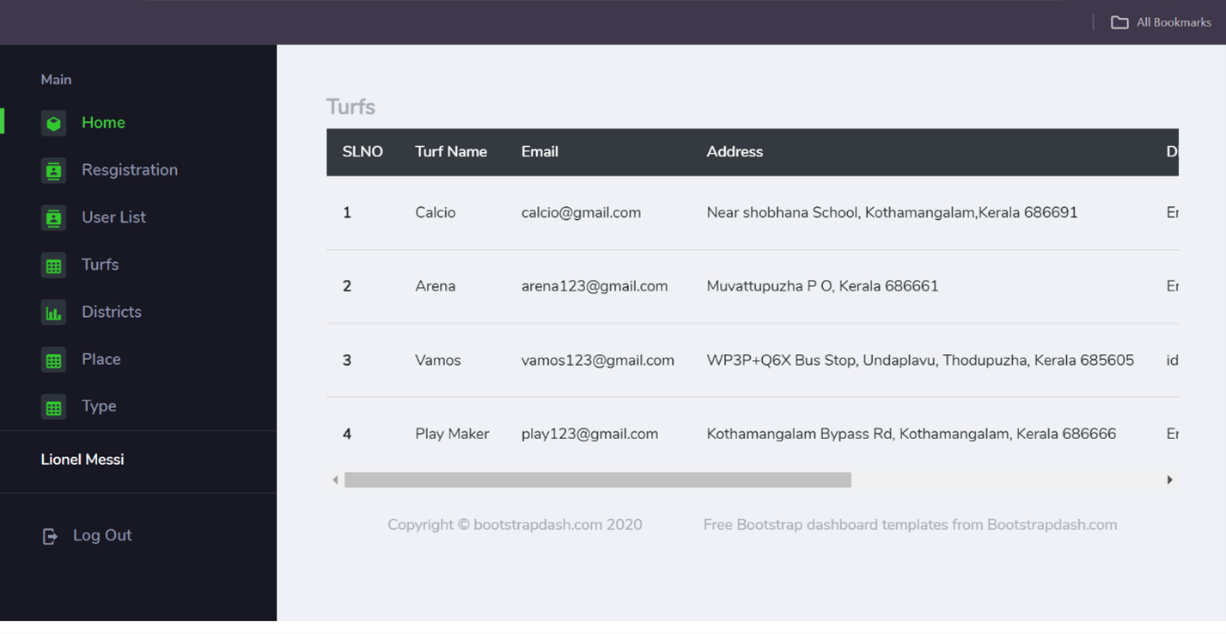
**10. APPENDIX**

**10.1 SCREEN SHOTS**

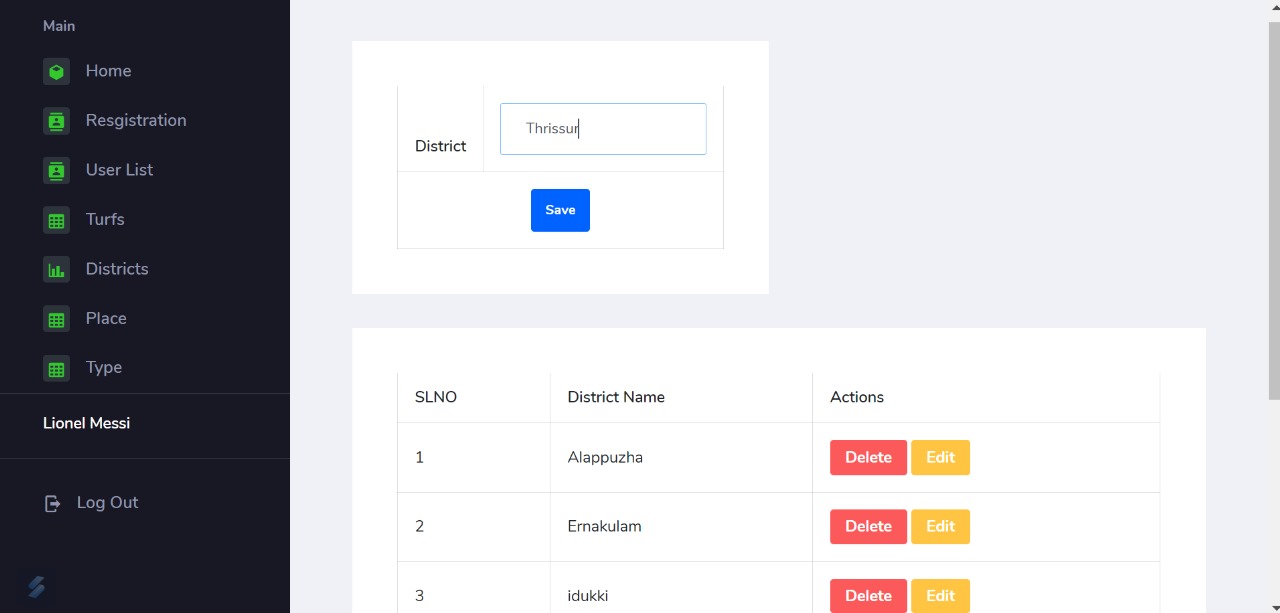
**INDEX PAGE**



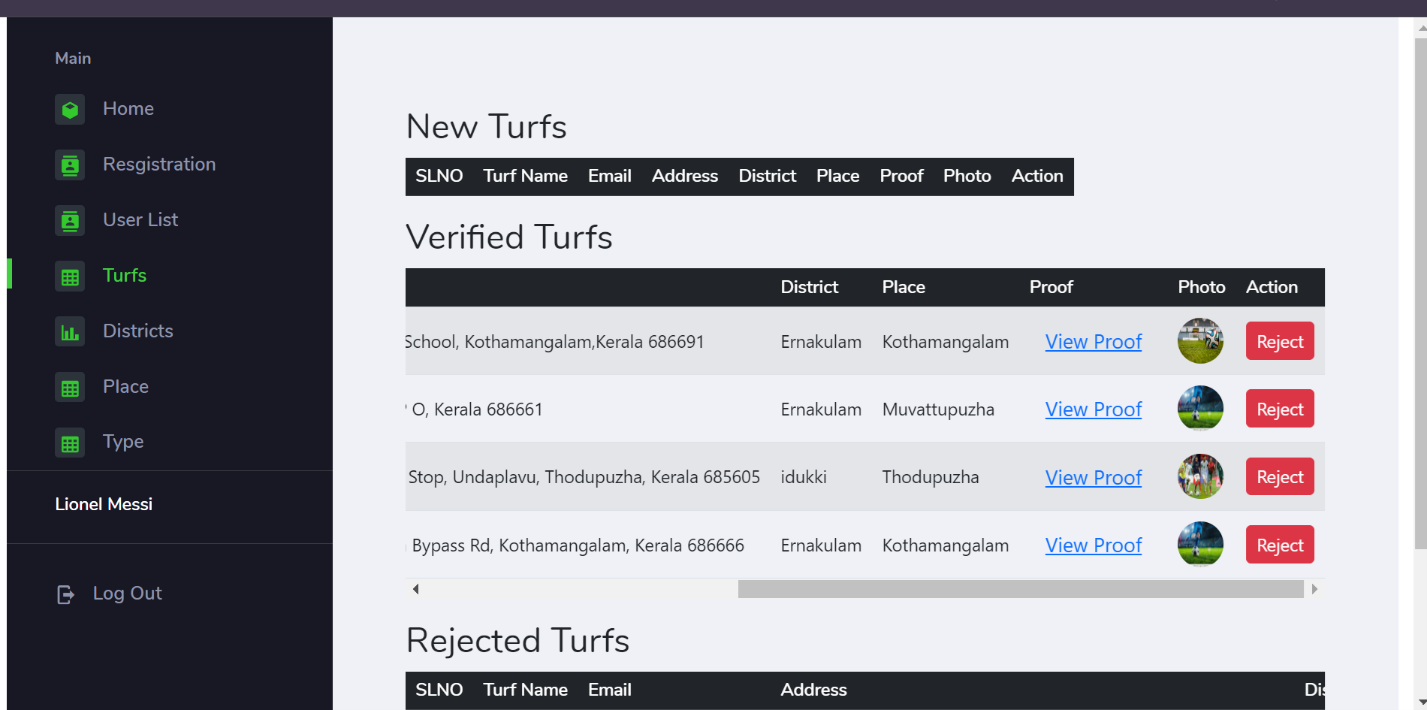
**ADMIN DASHBOARD**



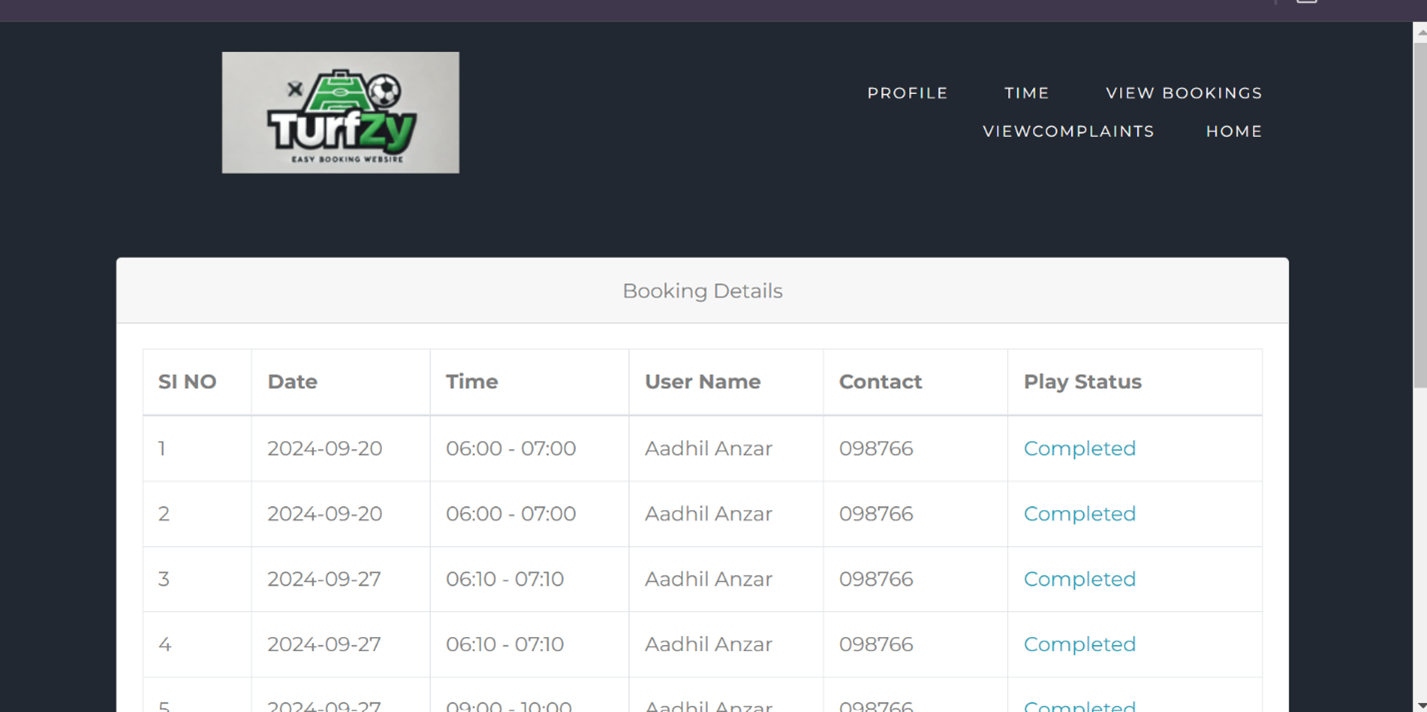
**ADMIN ADD DISTRICT**



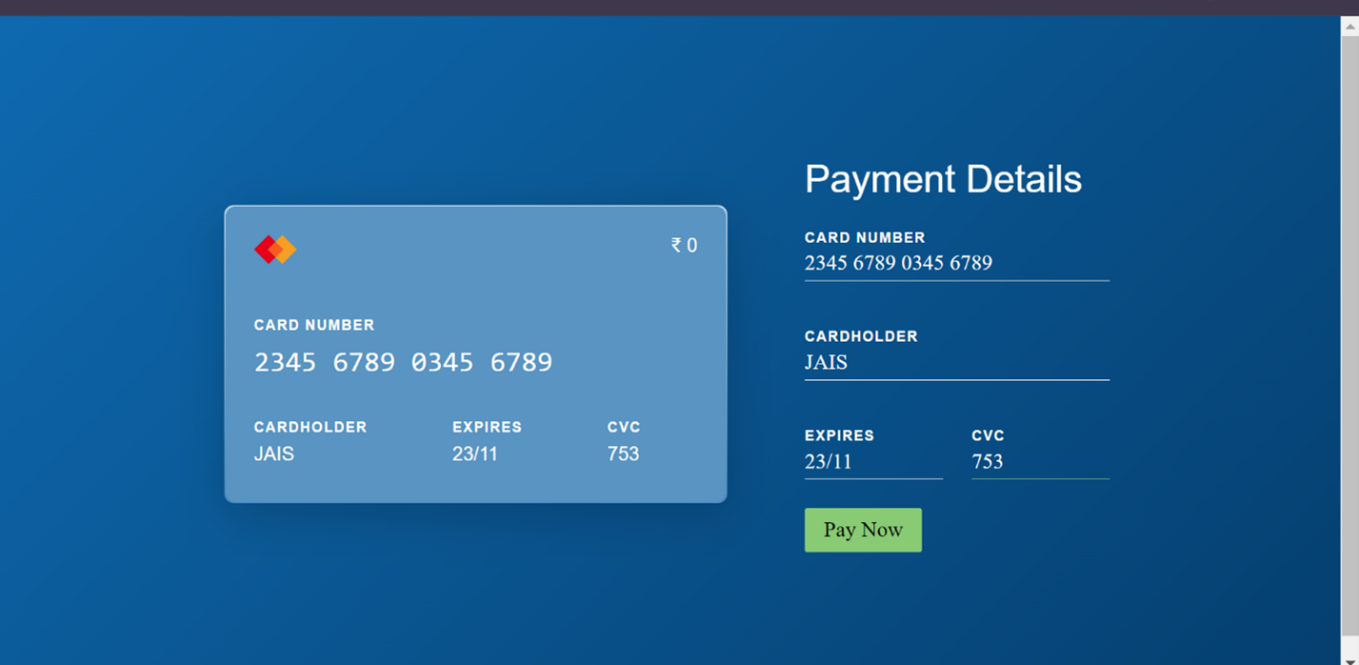
**ADMIN VERIFY TURF**



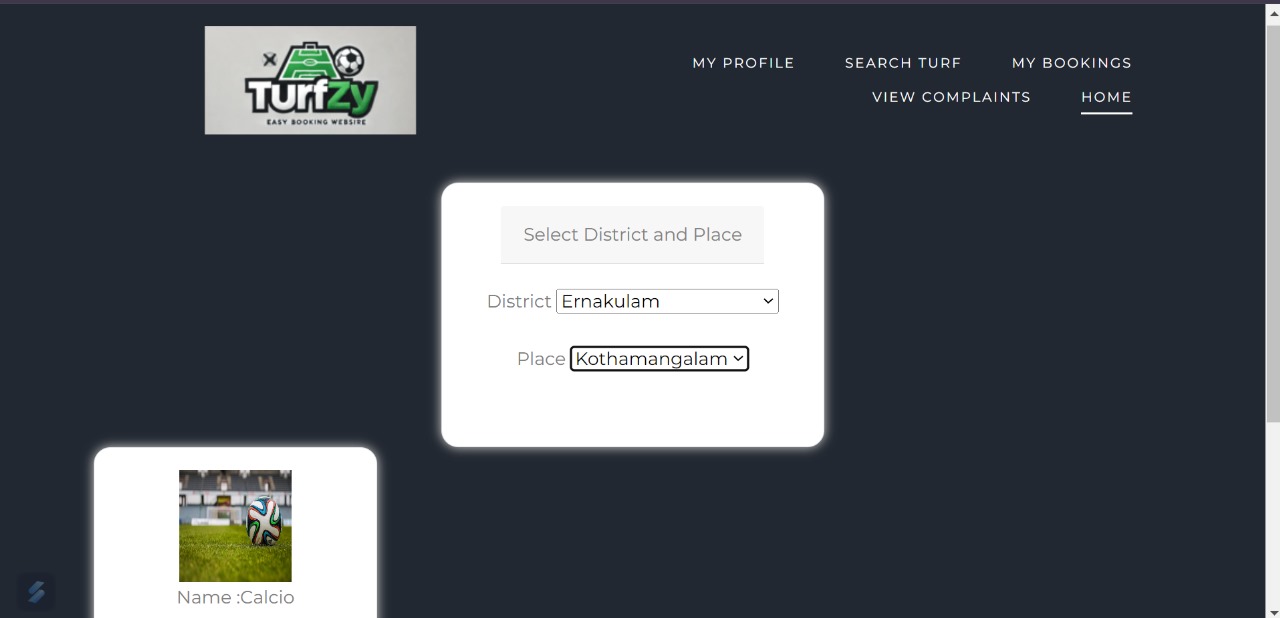
**USER VIEW BOOKINGS**



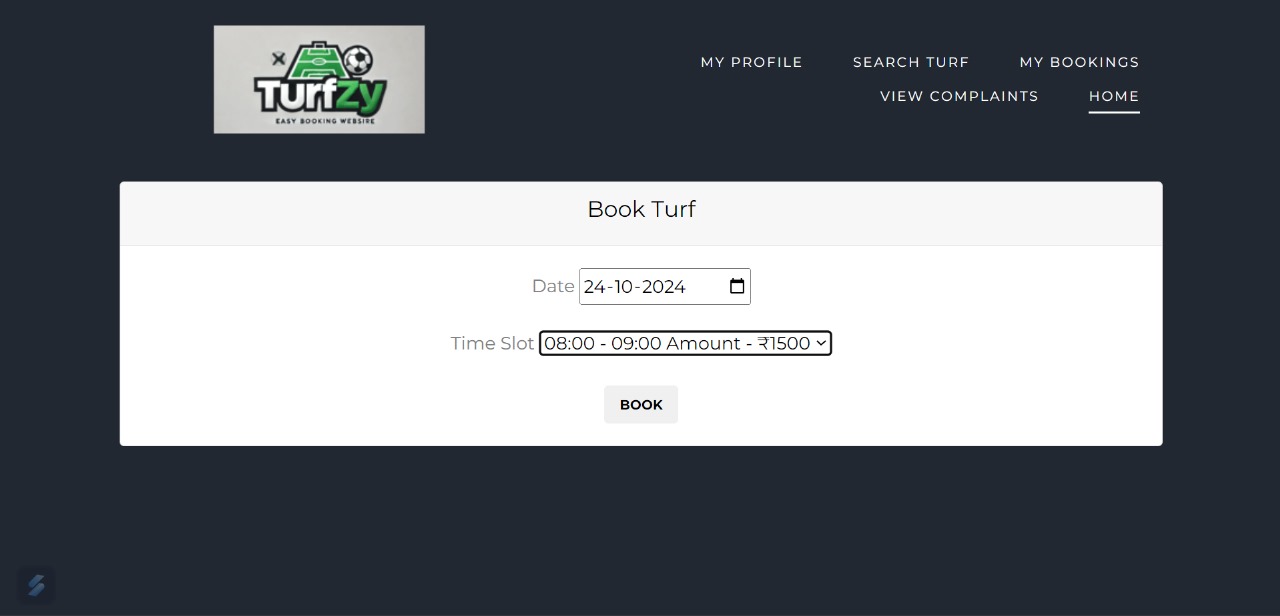
**PAYMENT**



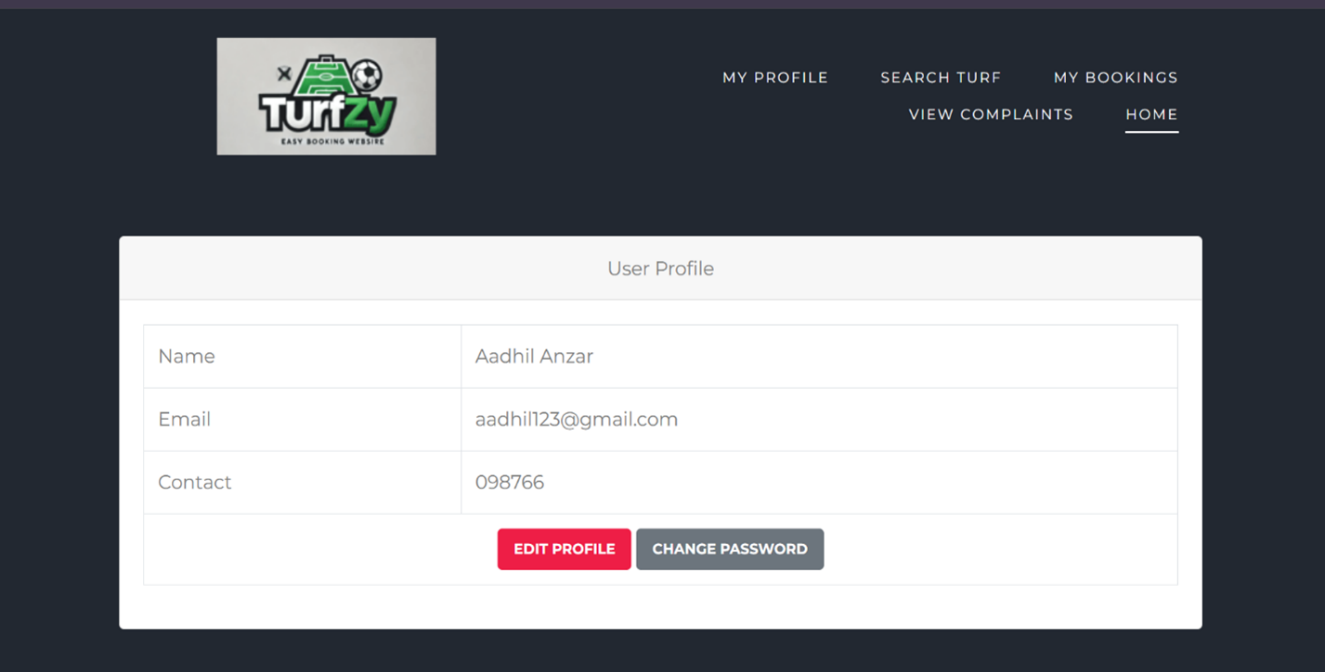
**SEARCH TURF**



**BOOK TURF**



**USER PROFILE**



**USER VIEW BOOKINGS**

