

A Project Report
on
Adventure Nation

by
Sanchita Sonar(T120338554)
Shivani Suryawanshi (T120338564.)
Priyanka Tikone(T120338569)

Under the guidance of

Ms.Anagha Chaudhari

Department of Information Technology

Savitribai Phule Pune University

2016-2017





PIMPRI CHINCHWAD COLLEGE of ENGINEERING

Date:

CERTIFICATE

This is to certify that,

Sanchita Sonar (T120338554)
Shivani Suryawanshi(T120338564)
Priyanka Tikone (T120338569)

of class T.E IT, have successfully completed their project work on “**Adventure Nation**” at Pimpri Chinchwad College Of Engineering in the partial fulfillment of the Graduate Degree course in T.E at the Department of **Information Technology**, in the academic Year 2016-2017 Semester – I as prescribed by the Savitribai Phule Pune University.

Ms. Anagha Chaudhari
Guide

Dr.S.D.Thepade
Head of the Department

Acknowledgements

Right from the very beginning of this project, we have been getting help from many people in numerous ways. Through this acknowledgement, we are trying to express our gratitude to all those who have helped us in one way or the other in the successful completion of this project.

Although a single sentence hardly suffices, we would like to thank almighty god for blessing us with his grace and taking our endeavor to a successful culmination. We express our sincere gratitude to our project guides Prof Sandhya.S.Waghere , Prof.V.D.Katkar and Prof.Anagha Chaudhari for their constant support , valuable suggestions and providing us with adequate facilities , ways and means without which the successful completion of this project would not have been possible . We also thank our HOD Dr.S.D.Thepade for their timely support and suggestions. Last but not the least we would like to acknowledge our family members and friends for their unending support.

Sanchita Sonar(T120338554)
Shivani Suryawanshi(T120338564)
Priyanka Tikone(T120338569)

Contents

Sr. No.	Topic	Page No.
	Acknowledgement	i
	Contents	ii
	List of Tables	iii
	List of Forms	iv
	Abstract	v
Chapter-1	Introduction	1
	1.1 Motivation	1
	1.2 About the project	1
	1.3 Problem statement	2
	1.4 Framework of the proposed work	3
Chapter-2	Features	4
Chapter-3	Literature Review	5
	2.1 Introduction	5
	2.2 Existing methodologies	6
	2.3 Proposed methodologies	7
Chapter-4	Software Requirement Specification	8
	3.1 Hardware Requirements	8
	3.2 Software Requirements	8
Chapter-5	Assumptions	9
Chapter-6	Entity-Relationship Diagram	10
Chapter-7	Tables	14
Chapter-8	Forms & Reports	18
Chapter-9	Conclusion	32
	References	33

List of Tables

Table No.	Title	Page No.
7.1	Groups	14
7.2	Payment_detail	14
7.3	Customer_detail	15
7.4	Feedback	15
7.5	Employee	16
7.6	Trainer	16
7.7	Manager	16
7.8	Activity	17
7.9	Equipment	17

List of Forms

Figure No.	Title	Page No
8.1	Home	18
8.2	customer_registration	18
8.3	new_registration	19
8.4	add_grp_member	19
8.5	Existing_customer	20
8.6	ec_login	20
8.7	customer_details	21
8.8	Payment	21
8.9	Feedback	22
8.10	management_staff	22
8.11	t_login	23
8.12	em_login	23
8.13	Manager	24
8.14	customer_info	24
8.15	employee_info	25
8.16	new_emp_reg	25
8.17	activity_info	26
8.18	equipment_info	26
8.19	Report	27

Abstract

Adventure Nation is an adventure park management system which serves the purpose of easy computerized registration of customers. The system also provides the facility of easy governance of all the employees and activities that are taking place in the park to the park manager.

This E-governance system serves all the facilities at one place for customers and employees without much human intervention. The system consist of two modules. The customer's module includes customer registration, payment and feedback. The management module includes facilities like keeping track of all customers, employees as well as activity equipment. The system dynamically calculates profit and loss that the park has faced in its earnings.

The system is very easy to handle as well as understand, making it user friendly. Thus customers will be much more comfortable with this system than the traditional ways. This system is very helpful since it results in reduction of manual work and instead provide e-governance which is much better and easier. It also helps in managing the adventure park more efficiently thus increasing the performance of the business which ultimately results in increased profit.

Introduction

1.1 Motivation:-

Tourism is one of the ever progressing and upcoming field in India. Indian government is trying its utmost to promote tourism. One of attractions of India are its hill stations. We have hill stations like Manali, Shimla, Lonavla etc. One of the attractions of such places is the training of extreme sports. Extreme sports is a popular term for certain activities perceived as having a high level of inherent danger. Today's youth is crazy about extreme sports. The rush of adrenaline and thrill is something they crave very much for. This makes extreme sports business one of the emerging and successful business prospect.

Management of an extreme sports business is a subject which is not much thought about. We still use traditional methods for managing an extreme sports business. There are not many systems developed for such a problem. Since not much solutions have been developed for managing this business we decided to design a system which will make management of this business.

1.2 About The Project :-

We have designed a management system for an adventure park which provides training and joyrides for three sports which mainly includes Paragliding, Bungee jumping and Scuba diving. The joyride is a one day fare while training is for a duration of specified period.

The project contains two modules which mainly consist of customer module and management module. The customer module keeps track on the customers registering into the system .When customers arrive in the park first they have to register themselves. If the customer is old one then his account will already be there in the system. So if he is existing customer then he can view his profile. After registering they can go and enjoy whichever activity they are registered for. When they come back they have to do the payment and give feedback.

On the other hand , management module keeps track on employees and equipment used for the sports. The system has one equipment manager who keeps track on all the details of equipment. The trainer provides training for customers. Trainer is assigned to customers at the time of registration according to the status of availability. Trainer is given salary on the basis of number of customers he has provided service.

Manager has the sole right of adding a new employee profile or deleting profile of an employee when he leaves his job. Manager can also view all the employee profiles and equipment details. Customer can register in groups or he can register as a single participant. But even if he registers as a single participant he will be assigned a group name.

1.3 Problem Statement:-

To develop a management system which will keep track on the customers registering for activities in the adventure park. The system should be able to store the registration details of customers and display their profile if the customer record already exist in the system database. The manager should be able to insert or delete an employee record. He has the sole authority to make changes in employee profile. He can also view equipment details.

The equipment manager should be able to view equipment details so that he can keep track on the equipments used for the activities. The system should automatically assign trainer to the customers at the time of registration on the basis of availability status of the trainer. The system should be able to calculate the salary of trainer based on the number of customers he has serviced.

1.4 Framework of The Proposed Work :-

The system is designed considering two views –

1. Customer View
2. Management View

The customer module consists of :-

- Registration if you are new customer.
- Viewing your account information if you are old customer.
- Payment
- Giving feedback

The management module consist of

- Log in according to your designation
- Viewing your profile if you are a trainer
- Viewing equipment details if you are equipment manager
- Adding a new employee profile or deleting an employee profile when he leaves his job if you are manager
- Viewing customer details if you are manager
- Viewing report

Features

Adventure Nation is a very easy and simple management system. The graphical user interface which is designed using Java Swings is very user friendly.

Features are as follows:

1. The integration and sharing of the data files minimizes the duplication and redundancy of data to a great extent.
2. Easy and smooth interfaced designed with the help of java.
3. Can run on any type of operating system.
4. Fewer application programs are developed for obtaining various reports due to independence of programs and data.
5. Data Updating is easier due to integration of data files. With multi-update functionality.
6. Large database can be maintained.

Literature Review

2.1 Introduction:-

Extreme sports fall into adventure tourism category. Muller and Cleaver(2000)[5] defined adventure tourism as the capability to provide tourists with relatively high degrees of sensory stimulation. It is usually contained some physical challenging elements with the tourist's experience. While Buckley(2006) mentioned that the term adventure tourism is used to mean as guided commercial tours where these major attraction is an outdoor activity that has natural environment features and normally needs specialized sporting or equipment. It must be exiting for the tourists also. Therefore, adventure tourism can refer to activities done by the tourists, which have high level of perceived or real risk like diving, hiking, mountaineering, mountain biking, caving, sky-diving, skiing, snowboarding, white water rafting, kayaking, sailing, and sea kayaking. It also refer to a specific location of place with high degree of risk such as visiting desserts, jungles or mountaintops, polar regions and safaris. All these places have strong elements of adventure.

Adventure tourism can be divided into soft adventure and hard adventure. These terms are developed by researchers who devised a scale to explain the diversity of behavior, beginning with mild adventure also known as soft adventure at one end of the scale and progress to hard adventure at the other extreme. This continuum, involves different degrees of challenge, uncertainty, setting familiarity, personal abilities, intensity, duration and perceptions of control. A simpler way to describe soft and hard adventure is that soft adventure does not necessarily require past experience whereas hard adventure requires some experiences and proficiency in the activity prior to the tourism experience(Millington and Locke, 2001).

2.1.1 Motivation Behind Seeking Extreme Sports:-

Crandall (cited in Hall & Page 1999) who did a study on the motivations of the leisure travelers, outlined some motivational factors which derived from a synthesis of previous studies in this field. Below are the motivational factors listed by Crandall.

1. Enjoying nature, escaping from civilization:-

To get away from civilization for a while to be close to nature.

2. Escape from routine and responsibility:-

Change from daily routine. To get away from the responsibilities of daily life.

3. Physical exercise:-

For the exercise

4. Stimulus seeking:-

For the excitement. Because of the risks involved.

2.2 Existing Methodologies:-

The extreme sports business is more popular in hill station areas. This areas though well-developed due to tourism are still lacking in use of technology. They still rely on traditional methods. In case of extreme sports business, traditional methodology includes keeping customer records, payment details etc. in files or registers. Every detail be it regarding to customers, employees or equipment is kept written down on paper.

This makes it difficult to search for records, bookings, etc. when needed in emergency situations since they have to go through large number of records which takes time, plus it may not always be efficient. And since all logs are maintained in books or files, it is difficult to search for records after a long time (like after 5-10 years).

The performance of the business decreases when it comes to analyzing huge amount of data, summarizing data, or generating reports of all records which needs to be done every financial year by a particular service or workshop.

2.3 Proposed Methodology:-

The Adventure Nation is an e-governance system. To solve the issues faced by traditional methods, in this system everything is computerized. The customer when registers for some activity instead of filling a form, he has to register on adventure nation software system. He can check out his account if he is old customer. He can pay online instead of giving cash or cheque. He can also give feedback in this software, instead of writing it on paper.

The manager can keep records of all employees in the software, instead keeping it in files or registers. In files or registers erasing an employee record when he leaves job is more difficult and messy. This becomes easier due to use of software. Manager can view all equipment details just on one click. Searching of a record becomes faster and easier. Updating records in case of changes becomes easier if we use this system.

Hence use of this management system will reduce manual work and generate faster results for any query. Thus the performance of business will increase.

Software Requirement Specification

3.1 Hardware Requirements:-

- Intel i3 700MHz processor or higher.
- 512MB of RAM or more.
- 100% Windows® 7/8/10 or Ubuntu
- 20GB Hard disk space
- Standard color monitor
- Standard keyboard
- Standard mouse

3.2 Software Requirements :-

- Windows 7 version and onwards.
- NetBeans Ide 8.1
- MongoDB
- MS Word for documentation
- JDK
- Java
- MongoDB jar file

Assumptions

This management system is designed from only two point of views

1. Customer point of view
2. Management point of view

For designing of system we have assumed that, only customer or adventure park employee will log in to the system. Any other person like equipment dealer etc. are not taken into consideration. Only this two primary users will be able to use the system.

Manager is sole responsible for keeping track on employees and equipment. Manager is the only primary user who can directly access the database. No other user can directly view the underlying database. Other users than manager can only view information fetched from database and displayed on the screen.

We have assumed that the salary of manager and equipment manager is static while the salary of trainer is dynamic. The salary of trainer will be calculated based on the number of customers he has serviced. We have assumed that the trainer will not be assigned after registration. Instead we are assigning trainer dynamically based on his availability status.

We have assumed that after registration customer will go and enjoy whatever activity he has registered for. After that he will come back and do the payment.

Entity – Relationship Diagrams

Data Models

Before you look at specific symbols, it is important to understand the various levels of ERDs. There are several ways to model entity-relationship diagrams. The most high-level type is a conceptual data model; the next highest is the logical data model, and the lowest-level (and therefore most detailed) type is the physical data model.

Conceptual Data Model

- The ER model establishes a broad view of what should be included in the model set. Conceptual data models:
- Include important entities and the relationship between them.
- Do not specify primary keys.
- Conceptual ERDs can be used as the foundation for logical data models. They may also be used to form commonality relationship between ER models as a basis for data model integration.

Logical Data Model

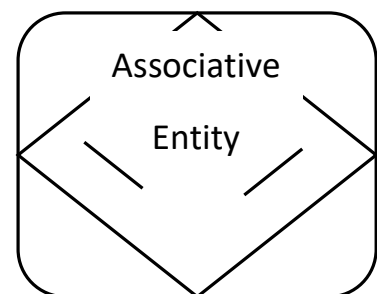
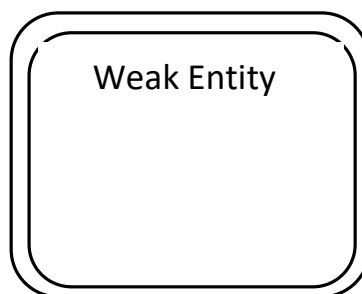
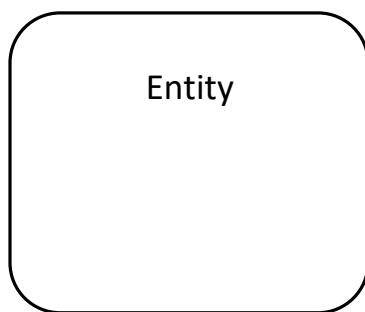
- This model contains more detail than the conceptual ER model, without regard to how information will be physically implemented in the database. Logical data models:
- Include all entities and relationships between them.
- Specify attributes for each entity.
- Specify primary key for each entity.
- Specify foreign keys, which identify the relationship between different entities.
- Involve normalization, which is the process of removing redundancy in a table so that the table is easier to modify. Normalization typically occurs by dividing a table into two or more tables and defining relationships between them.

Physical Data Model

- The physical data model represents the process of adding information to the database. This model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables. Physical data models:
- Specify all tables and columns.
- Include foreign keys to identify relationships between tables.
- May include denormalization, depending on user requirements.
- May be significantly different from the logical data model.
- Will differ depending on which DBMS is used.

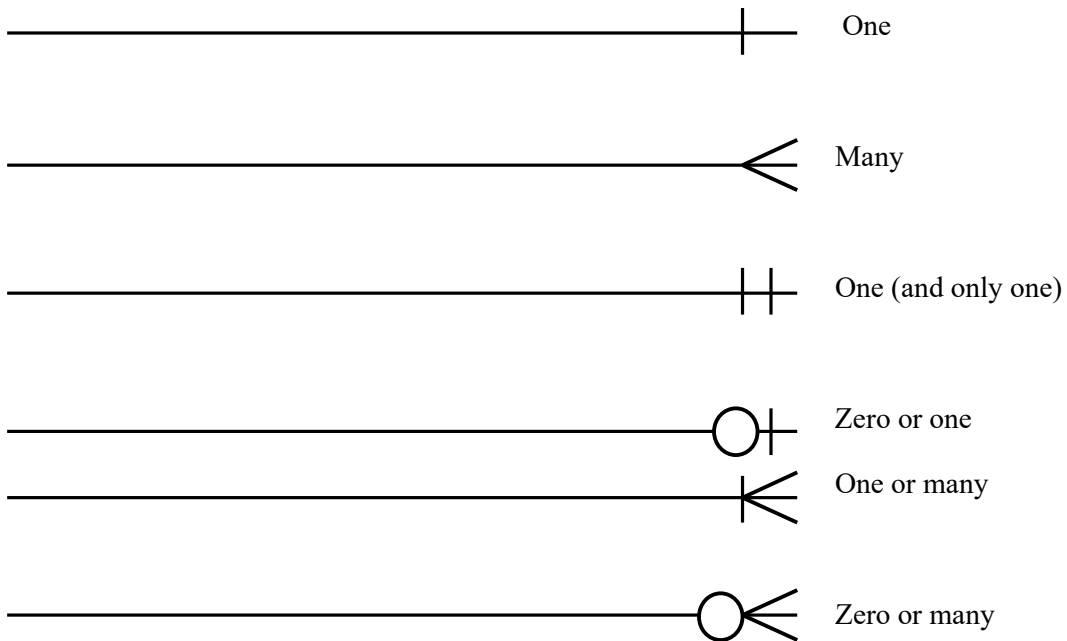
Entities

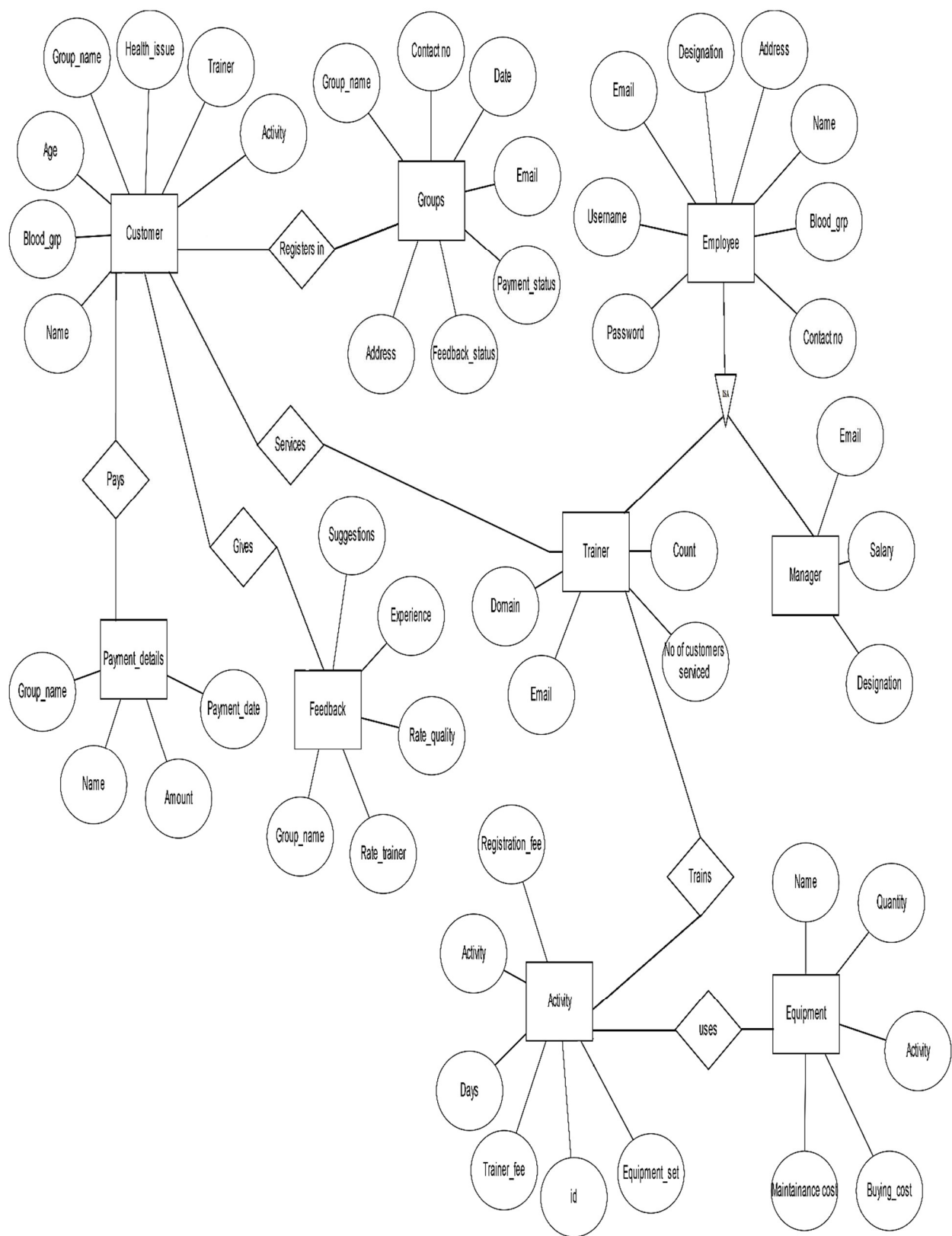
- Entities are objects or concepts that represent important data. They are typically nouns, e.g. patient, doctor or admin.
- Strong entities exist independently from other entity types. They always possess one or more attributes that uniquely distinguish each occurrence of the entity.
- Weak entities depend on some other entity type. They don't possess unique attributes (also known as primary key) and have no meaning in the diagram without depending on another entity.
- Associative entities are entities that associate the instance of one or more entity types. They also contain attributes that are unique to the relationship between those entity instances



Entity- Relationship Diagram Notation

- **Relationships** illustrate an association between two tables. In the physical data model, relationships are represented by stylized lines.
- **Cardinality** and **ordinarily**, respectively, refer to the maximum number of times an instance in one entity can be associated with instances in the related entity, and the minimum number of times an instance in one entity can be the styling of a line and its endpoint, as denoted by the chosen notation style.





Tables

7.1 Groups

Column Name	Data Type	Size
Grp_name	Varchar	20
Address	Varchar	20
Email	Varchar	20
Contact_no	Int	20
Payment_status	Boolean	-
Feedback_status	Boolean	-
Date	Date	20

7.2 Payment_detail

Column Name	Data Type	Size
Grp_name	Varchar	20
Card_holder	Char	20
Amount	Int	20
Payment_date	Date	20

7.3 Customer_detail

Column Name	Data Type	Size
Cust_name	Char	20
Cust_bloodgrp	Varchar	20
Cust_age	Int	20
Cust_activity	Varchar	20
Cust_group	Varchar	20
Cust_healthissue	Varchar	20
Trainer	Char	20

7.4 Feedback

Column Name	Data Type	Size
Grp_name	Varchar	20
Rate_trainer	Int	10
Rate_quality	Int	10
Experience	Varchar	20
Suggestions	Varchar	20

7.5 Employee

Column Name	Data Type	Size
Emp_name	Char	20
Emp_address	Varchar	20
Emp_email	Varchar	20
Emp_contact	Int	20
Emp_bloodgrp	Varchar	20
Username	Varchar	20
Password	Varchar	20
Designation	Varchar	20

7.6 Trainer

Column Name	Data Type	Size
Domain	Varchar	20
Cust_serviced	Int	20
T_email	Varchar	20
Count	Int	20

7.7 Manager

Column Name	Data Type	Size
Salary	Int	20
Designation	Varchar	20
M_email	Varchar	20

7.8 Activity

Column Name	Data Type	Size
Act_name	Varchar	20
Days	Varchar	20
Trainer_fee	Int	20
Registration_fee	Int	20
Equipment_set	Varchar	20

7.9 Equipment

Column Name	Data Type	Size
Name	Varchar	20
Buying_cost	Int	20
Maintainance_cost	Int	20
Quantity	Int	20
Activity	Varchar	20

Forms and Reports

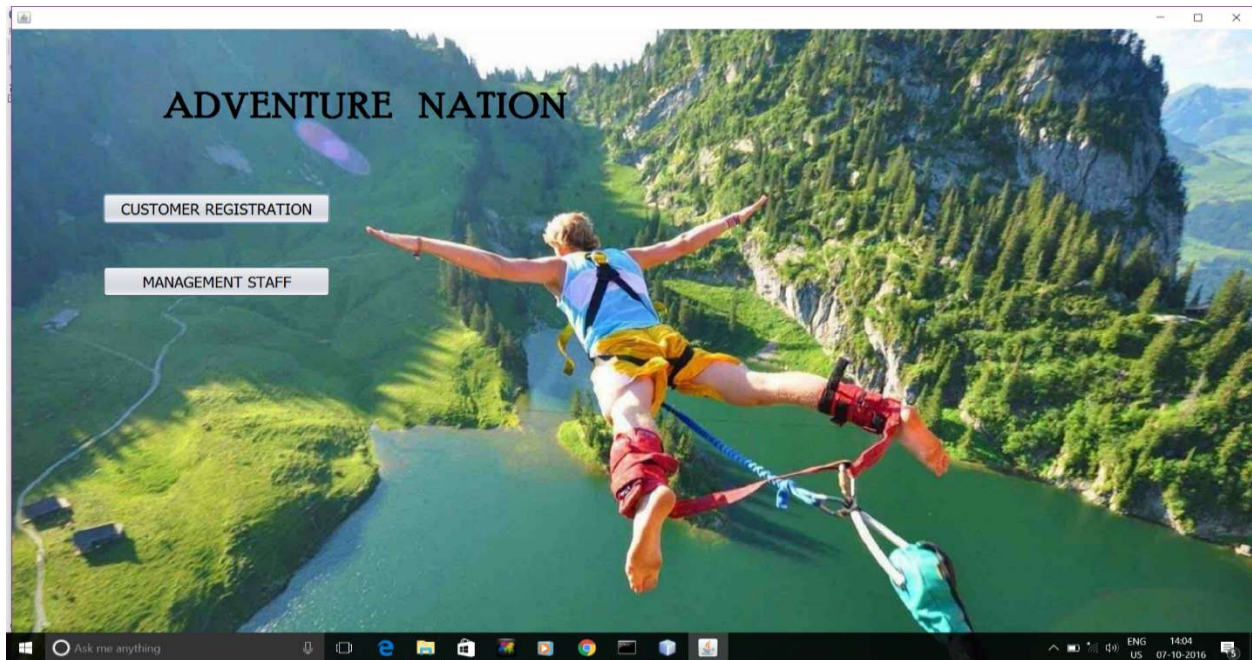


Fig 8.1 : Home page

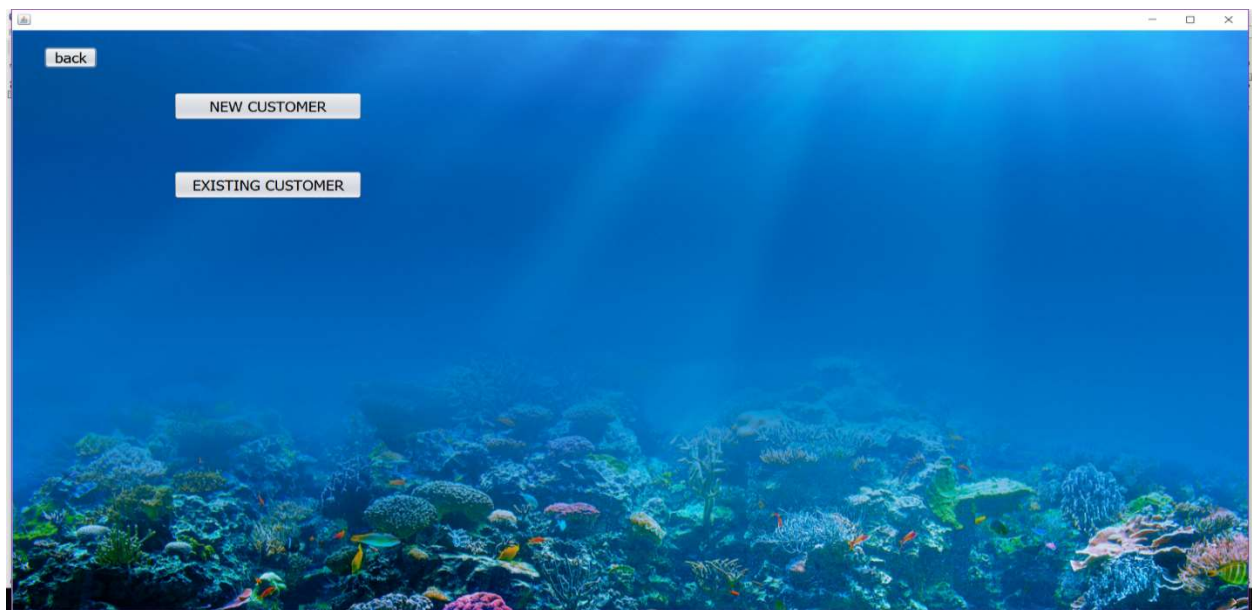


Fig 8.2 : Customer registration

back

Group Name

Address

Email ID

Mobile Number

ADD GROUP MEMBERS

Fig 8.3 : new_registration

Name

Blood Group

Age

Activity

Health Issues

SUBMIT ADD MEMBER

Fig 8.4 : add_grp_members

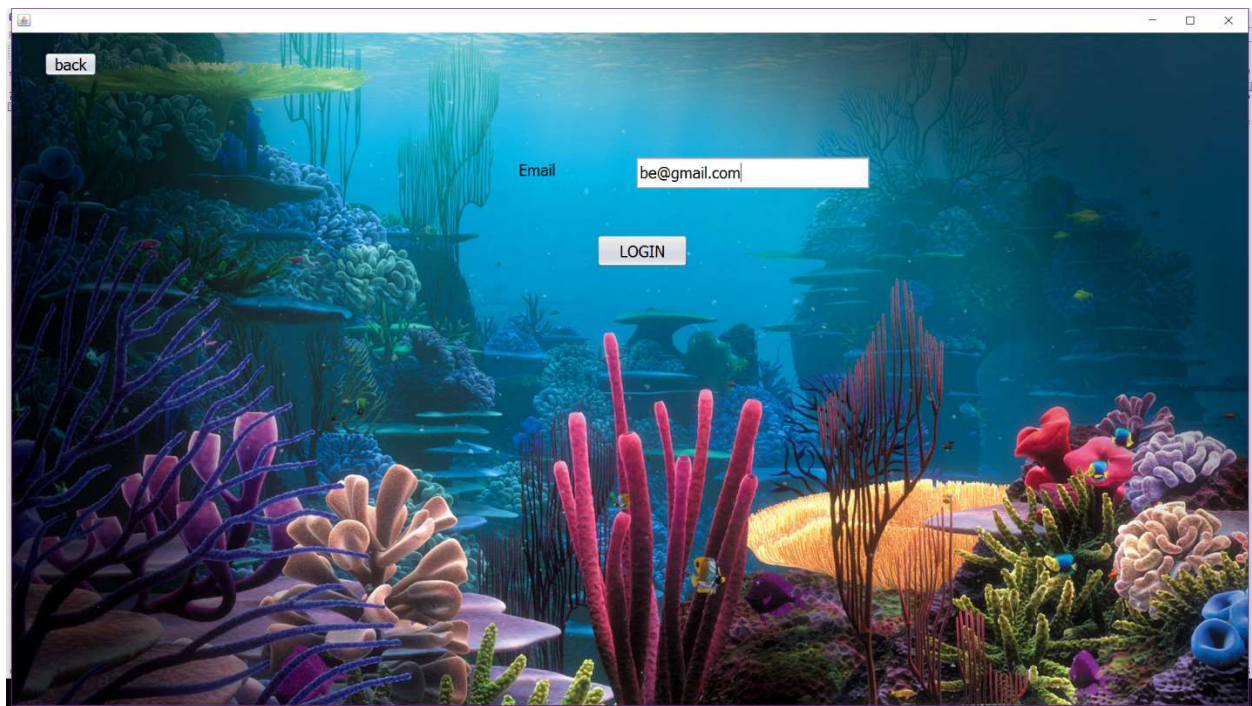


Fig 8.5 : existing_customer

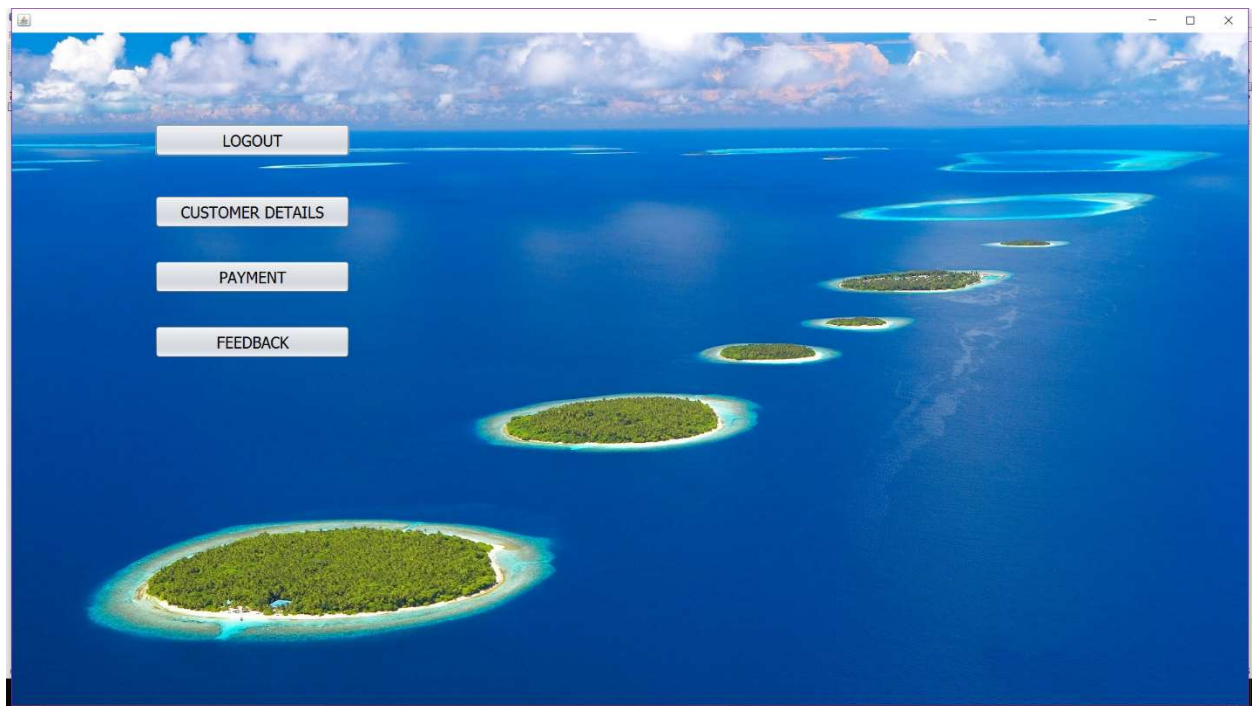


Fig 8.6 : ec_login

back

Group detail

Group Name: BE

Address: Aakurdi

Email ID: be@gmail.com

Contact no.: 9850100506

Group Members

Name	Blood group	Age	Activity	Health issue	Trainer
Anita	B+ve	25	Scubadiving training	NO	Abhishek Pali

Fig 8.7 : customer_detail

back

Card Number:

Card Holder Name:

CVV:

Expiry Date
Month: Year:

Payment details

Name	Activity	Cost
Anita	Scubadiving training	12000

Total: 12000

PAY NOW

Fig 8.8 : Payment

A feedback form interface displayed over a vibrant coral reef background. The form includes four input fields: 'Rate Trainer' with a dropdown menu showing '2', 'Rate Equipment Quality' with a dropdown menu showing '5', 'Share your experience' with a text area containing 'Very nice', and 'Suggestions (if any)' with a text area containing 'No'. A 'SUBMIT' button is located below the input fields.

Rate Trainer 2

Rate Equipment Quality 5

Share your experience Very nice

Suggestions (if any) No

SUBMIT

Fig 8.9 : Feedback

A login interface for management staff displayed over a vibrant coral reef background. The interface includes a 'back' button in the top left corner, three input fields: 'Username', 'Password', and 'Designation' (a dropdown menu showing 'Select'), and a 'Login' button. The background features a detailed coral reef scene with various colorful corals and small fish.

back

Username

Password

Designation Select

Login

Fig 8.10 : management_staff

Log out

PROFILE

Name: Priyanka Tikone

Address: Maulinagar,Dighi,pune 15

Email Id: ptikone2323@gmail.com

Contact No: 9767251678

Blood Group: A +ve

Username: priyanka

Password: priyanka

Domain: Bungee jumping

No. of Customers Served - **1**

Salary **1500**

Fig 8.11 : t_login

LOGOUT

Profile

Name: Sanchita Sonar

Address: Ganesh nagar,dighi,pune-15

Email ID: sanchitasonar@gmail.com

Username: sanchita

Contact no.: 9557459875

Password: sanchita

Blood group: A +ve

Salary: 25000

Equipment Detail

Sr no.	Equipment Name	Buying cost	Maintenance cost	Quantity	Activity
1	Paraglider	15000	4000	22	paragliding
2	Harness	3000	0	22	paragliding
3	Safety helmet	1000	0	22	paragliding
4	Radio and radio har...	1000	1000	22	paragliding
5	Variometer GPS	2000	0	22	paragliding
6	Hook knife	100	0	22	paragliding
7	Gear bungee cords	10000	3000	12	Bungee jumping
8	Gear bumper pads	5000	0	12	Bungee jumping
9	Harness	5000	0	12	Bungee jumping
10	Face shield	5000	0	12	Bungee jumping
11	Injection steel Cabin	5000	0	12	Bungee jumping

Fig 8.12 : em_login

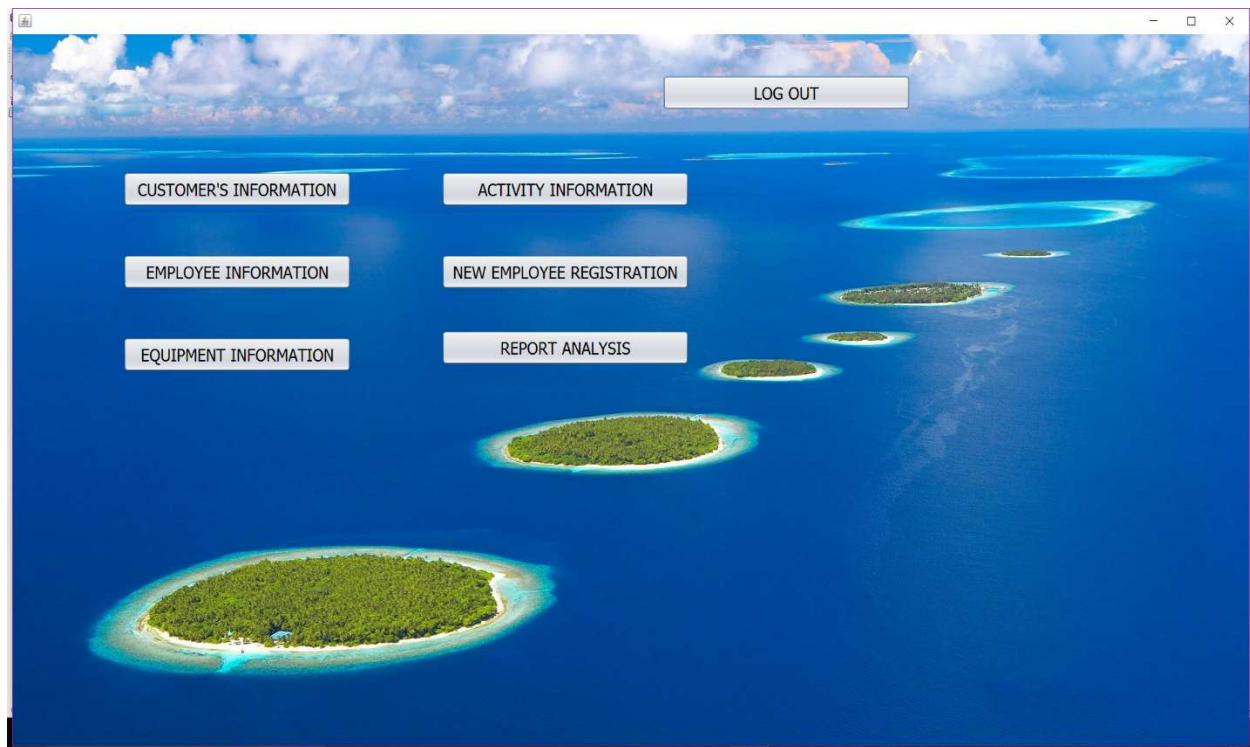


Fig 8.13 : Manager

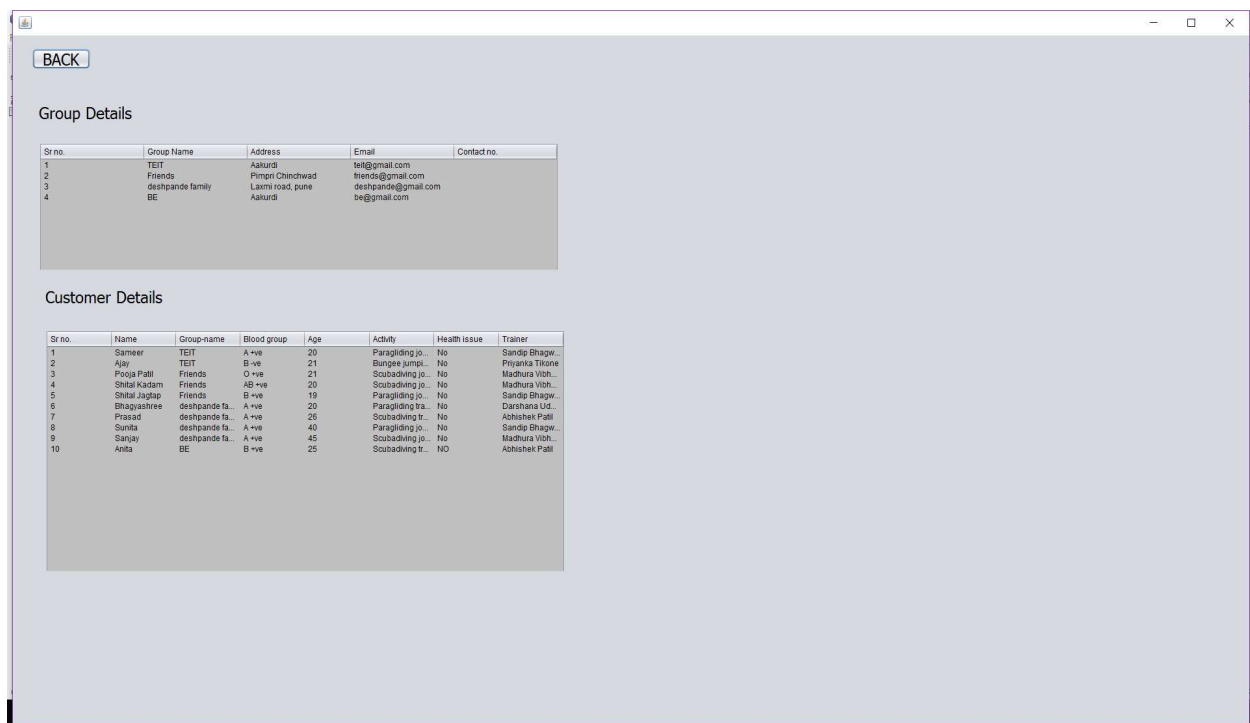


Fig 8.14 : customer_info

BACK

Employee Details

Sr no.	Employee Name	Address	Email	Contact	Blood grp	Designation
1	Shivani Suryawa.	Morevastt Chihl...	shivani1323@gg...	Morevastt Chihl...	b-ve	Manager
2	Sandhya Sonar	Ganesh nagar.d...	sandhyasonar@...	Ganesh nagar.d...	A-ve	Equipment Mana.
3	Madhura Vishude	shivajaya adarsh...	madhura15@gm...	shivajaya adarsh...	AB +ve	Trainer
4	Ashwasi Falli	vishranvewadi pun...	ashwasi@gma...	vishranvewadi pun...	b-ve	Trainer
5	Darshana Udhane	bijlunagar,nigdi	darshana29@gg...	bijlunagar,nigdi	A-ve	Trainer
6	Sandip Bhagwat	ganesh soc.tud...	sandip12@gmail...	ganesh soc.tud...	A-ve	Trainer
7	Priyanka Triore	Maulinagar,Dighi...	phone4232@gg...	Maulinagar,Dighi...	A-ve	Trainer

Fig 8.15 : employee_info

back

Name

Username

Password

Address

Email id

Blood group

Mobile Number

Designation

Domain *optional

SUBMIT

Fig 8.16 : new_emp_reg

Sr.no	Activity name	Days	Trainer fee	Registration fee	Equipment set
1	Paragliding joyride	1	2000	4000	10
2	Paragliding training	10	5000	10000	10
3	Scuba diving joyride	1	2500	5000	10
4	Scuba diving training	10	6000	12000	10
5	Bungee jumping	1	1500	3000	10

Fig 8.17 : activity_info

Sr.no	Equipment Name	Buying cost	Maintenance cost	Quantity	Activity
1	Paraglider	15000	4000	22	paragliding
2	Harness	3000	0	22	paragliding
3	Safety helmet	1000	0	22	paragliding
4	Radio and radio har...	1000	1000	22	paragliding
5	Varimeter GPS	2000	0	22	paragliding
6	Hook knife	100	0	22	paragliding
7	Gear bungee cords	10000	3000	12	Bungee jumping
8	Gear bumper pads	5000	0	12	Bungee jumping
9	Harness	5000	0	12	Bungee jumping
10	Face shield	5000	0	12	Bungee jumping
11	Lifeline chest Chalm	5000	0	12	Bungee jumping

Fig 8.18 : equipment_info

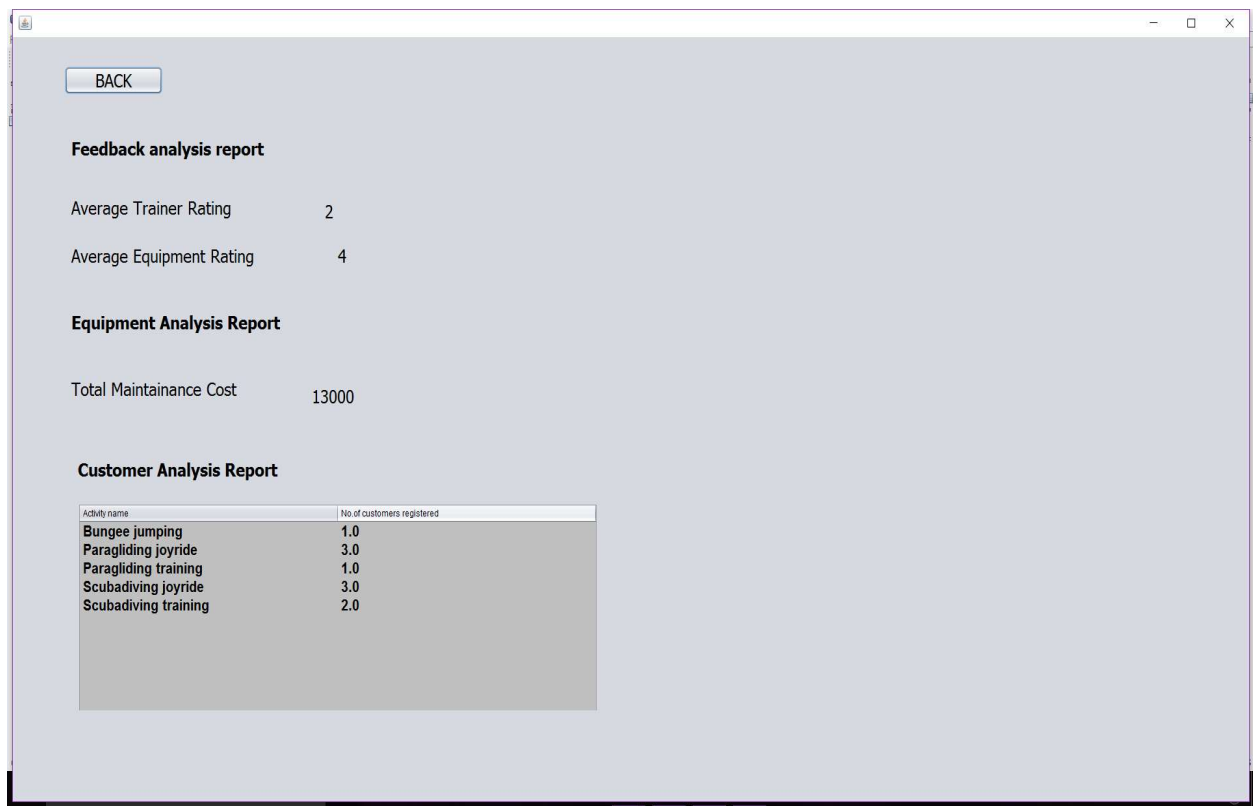


Fig 8.19 : report

Code:-

Insertion code:-

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt)
{
    if( (grp.getText().isEmpty()) || (addr.getText().isEmpty()) || (email.getText().isEmpty()) ||
        (mob.getText().isEmpty()) )
    {
        JOptionPane.showMessageDialog( grp, "Please fill all the details.");
        return;
    }

    String grp_name = (String)grp.getText();
    String address = (String)addr.getText();
    String email1 = (String)email.getText();
    String mobile = (String)mob.getText();
    Boolean chk=true;

    if(!email1.matches("^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-]+)*@[A-Za-z0-9-]+(\\.[A-Za-z0-9-9]+)*?(\\.[A-Za-z]{2,})$"))
    {
        JOptionPane.showMessageDialog(rootPane,"Invalid email id.");
        chk=false;
    }
    if(!mobile.matches("^[7-9][0-9]{9}$"))
    {
        JOptionPane.showMessageDialog(rootPane,"Please enter correct mobile number.");
        chk=false;
    }
    if(chk==true)
    {
        try{
            int flag=0;
            MongoClient mongo=new MongoClient("localhost",27017);
            DB db=mongo.getDB("Adventure_Nation");

            DBCollection groups = db.getCollection("Groups");
            DBCursor cursor = groups.find();
            while(cursor.hasNext()){
                BasicDBObject obj1 = (BasicDBObject)cursor.next();
                String grp1 = (String) obj1.get("Grp_name");
                if(grp1.contentEquals(grp.getText()))
                {
                    JOptionPane.showMessageDialog( grp, "Group name already exist.");
                    flag=1;
                    break;
                }
            }
        }
    }
}
```

```

        if(flag==0)
        {
            BasicDBObject obj = new BasicDBObject();
            obj.put("Grp_name",grp.getText());
            obj.put("Address",addr.getText());
            obj.put("Email",email.getText());
            obj.put("Contact-no",mob.getText());
            obj.put("payment_status",0);
            obj.put("feedback_status",0);
            obj.put("Date", new Date());
            groups.insert(obj);

            temp_grp = grp.getText();
            grp.setText("");
            addr.setText("");
            email.setText("");
            mob.setText("");
            add_grp_member.setVisible(true);
            new_registration.setVisible(false);
        }
    }
    catch(Exception e)
    {
        System.err.println(e.getMessage());
        e.printStackTrace();
    }
}

```

Updation Code :-

```

try{
    MongoClient mongo=new MongoClient("localhost",27017);
    DB db=mongo.getDB("Adventure_Nation");

    DBCollection groups = db.getCollection("Groups");
    DBCursor cursor = groups.find();
    while(cursor.hasNext()){
        BasicDBObject obj1 = (BasicDBObject)cursor.next();
        String email = (String) obj1.get("Email");
        if(email.contentEquals(email_login)){
            String grp = (String) obj1.get("Grp_name");
            BasicDBObject newdocument = new BasicDBObject().append("$set", new
BasicDBObject().append("feedback_status", 1));
            groups.update(new BasicDBObject().append("Email", email_login ), newdocument);
            DBCollection feedback1 = db.getCollection("Feedback");
            BasicDBObject obj = new BasicDBObject();
            //obj.put("_id",f);f++;
            obj.put("Grp_name",grp);

```

```

        obj.put("rate_trainer",rt.getSelectedItem());
        obj.put("rate_quality",quality.getSelectedItem());

        obj.put("experience",exp.getText());
        obj.put("suggestions",sugg.getText());
        feedback1.insert(obj);
        feedback.setVisible(false);
        ec_login.setVisible(true);
        return;
    }
}
}
catch(Exception e){
    System.err.println(e.getMessage());
    e.printStackTrace();
}
}

```

Report :-

```

private void jButton38ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JTable table;
    String[] columnNames = {"Activity name","No.of customers registered"};

    DefaultTableModel model = new DefaultTableModel();
    model.setColumnIdentifiers(columnNames);

    table = new JTable();
    table.setModel(model);
    table.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
    table.setBackground(Color.lightGray);
    table.setEnabled(true);
    table.setFillViewportHeight(true);
    table.setFont(new Font("", 1, 20));
    table.setRowHeight(25);

    JScrollPane scroll1 = new JScrollPane(table);
    scroll1.setBounds(0, 0, 800, 500);

    int atr=0;
    int aq=0;
    MongoClient mongo=new MongoClient("localhost",27017);
    DB db=mongo.getDB("Adventure_Nation");

    DBCollection feedback = db.getCollection("Feedback");
    DBCursor cursor1 = feedback.find();
}

```

```

        int i=0;
        while(cursor1.hasNext()){
            BasicDBObject obj1 = (BasicDBObject)cursor1.next();
            i=i+1;
            String trate = (String) obj1.get("rate_trainer");
            String qrte = (String) obj1.get("rate_quality");
            int trati = Integer.parseInt(trate);
            atr=atr+trati;
            int qrati = Integer.parseInt(qrte);
            aq=aq+qrati;
        }
        atr=atr/i;
        aq=aq/i;
        String atrs = Integer.toString(atr);
        datr.setText(atrs);
        String aqs = Integer.toString(aq);
        daer.setText(aqs);

        DBCollection equipment = db.getCollection("Equipment");
        DBCursor cursor2 = equipment.find();
        int tmc=0;
        while(cursor2.hasNext()){

            BasicDBObject obj2 = (BasicDBObject)cursor2.next();
            String mc = (String) obj2.get("maintainance cost");
            int mci = Integer.parseInt(mc);
            tmc=tmc+mci;
        }
        String tmcs = Integer.toString(tmc);
        dtm.setText(tmcs);

        DBCollection customer = db.getCollection("Customer_detail");
        String map = "function() { emit(this.cust_activity,1); }";
        String reduce = "function(key, values) {return Array.sum(values)}";
        MapReduceCommand cmd = new MapReduceCommand(customer, map, reduce,
            null, MapReduceCommand.OutputType.INLINE, null);
        MapReduceOutput out = customer.mapReduce(cmd);
        for (DBObject o : out.results()) {

            String x = (String)o.get("_id");
            double v = (double)o.get("value");
            model.addRow(new Object[] {x,v});
        }
        panel6.add(scroll1);
        report.setVisible(true);
        manager.setVisible(false);
    }

```

Conclusion

We have successfully implemented “Adventure Park Management System” using mongoDB as back end tool and java as front end development tool. We have provided a facility to manage and maintain all records related to the customers and employees in a database instead of files and registers, so that data can be accessed easily and efficiently. Also, generation of reports becomes easy with this system.

Thus the proposed Adventure Park Management System is a relatively easy and efficient way to manage as compared to existing methodologies. As for the conclusion, the objectives for this project were achieved and functioned well as the desired target. This system will help the adventure park database work systematically and will ease the user in order to manage all the data in the system. This system will help in increasing performance and profit of the business.

References

Books:-

1. Java the complete reference, 6th edition
2. “HEAD FIRST JAVA”, KRISTINA
3. Adventure tourism : the new frontier / John Swarbrooke ... [et al.].

Websites:

1. www.codeguru.com
2. www.dreamincode.com
3. www.github.com
4. www.oracle.com
5. www.ukessays.com
6. www.tutorialspoint.com
7. <https://docs.mongodb.com>

