Sri Lanka Institute of Information Technology

Project report

MLB_10.02_02

Wild-life Safari Trip Management System

Information Systems and Data Modeling – IT1090

B.Sc. (Hons) in Information Technology

PROJECT ID	MLB_10.02_02
CASE STUDY NAME	Wild-life Safari Trip Management System
CAMPUS/CENTER	Malabe Campus

Group Details:

	Student Registration Number	Student Name
1	IT21189944	Madusanka G.K. I
2	IT21190216	Thisera W.N.M
3	IT21379956	Hettiarachchi V. E
4	IT21377280	Rajapaksha C. S
5	IT21189630	Hewavitharana D. L

Contents

1 Introduction

When investigating wildlife safari systems, we looked at how the internet connects customers and businesses, as well as how a website may make a service more accessible to clients. As a result, we came up with a solution for the wildlife safari industry by creating a website where people can browse and buy safari packages online. Customers can also utilize our website to ask questions about our services or goods, receive customer service, and suggest improvements by simply registering. We provide several services for scheduling wild safari tours in Sri Lanka's several national parks via the internet.

A variety of technologies must be explored and understood to develop a wild safari management website. Programming languages (such as HTML, CSS, and JavaScript) and relational databases were among them (Such as My SQL).

A database must be able to manage a wide range of data. A database is required because managing a computer-based stored data system is easier than managing written or typed documents. Customers and administrators save time and effort because of this. Furthermore, since the data is stored in a database, there is no risk of losing it. The system's users are the only ones who can control it. It is vital to collect requirements and do requirement analysis while building a database. The functional and nonfunctional specifications, as well as the data requirements, can then be determined.

2 Hypothetical Scenario

"Wild Quest SL" is an online wildlife safari system that provides adventurers with a wide range of services. Our customers can choose from several safari packages that we have presented for them, all of which include amazing offers and discounts. As a guest, the user (unregistered user) can check promotions, view feedback, check packages, check availability, and check frequently asked questions. Also, the customer can create a user account using the signup button and then the user can log in to the system using the login field. Once the registered user logs in to the system, the website's database will validate the user and give access to the website. Besides the guest user's facilities, after logging in to the system, a registered user can make or cancel reservations, share experiences, and give feedback, and edit his own account details. Also, he can contact website owners using the "contact us" field. In that field, the user can see the website's contact information. If you want, users can send a direct message using the "send message" option. Website give chance to the user for donation to save wildlife. Both unregistered and registered users can donate. They can donate with their details or, if they want, they can donate as an anonymous person. Also, users can donate monthly.

Tour guides also have access to the system. They are the ones who check the users' messages and reply to them. Users require the authority of the system admin or manager to share experience and make or cancel reservations. An admin can manage staff accounts, manage user accounts, and approve cancel or reservation requests. System admin also can refund for the approved cancel requests. the system admin is the one who updates safari details by adding maps, images, videos, and package details. Also, he can update the social media accounts.

A manager has access to all these things, and besides that, he can generate reports, manage funds, and manage salaries. Both the system administrator and the manager have administrative accounts, with the manager having access to the higher level.

3 Requirement Analysis

3.1 Main Requirements

3.1.1 Functional Requirements

Main Functions of the website and Events that take place between the users and the system is described by the Functional requirements. Five users are using this Wild Quest Online safari system. Namely: Guest, Registered User, Administrator, and the Developer. They access this system in different ways where it is related to them.

1. **<u>Guest and Registered user (They can access the front-end of the system).</u>**

User requirements-

- Guest can Donate money to the wildlife maintenance by entering relevant details without login to the system.
- Guest can check promotions available in packages.
- Guest can check FAQ.
- Guest views the available Feedbacks.
- Guest can get register to the system by providing required details for the registration.
- Guest can check availabilities for reservations.
- Guest can Check packages of the website.
- Registered User login to the system using by providing required user login credentials.
- Registered User can do Reservations.

- Registered User can Share experiences and give feedbacks on the Website.

- Registered User can cancel Reserved Reservations.
- Registered User can Edit own Account details.
- Registered User can Contact Wildlife Quest in the Contact us page of the website.
- Registered User can coordinate by a friendly social media facility available in website or else can send a message mentioning the requirement.

System requirements-

- System should validate the credential entered by the Guest.
- System Should approve and Record the Donation details.
- System should display available promotions about packages and Shared experiences and Feedbacks Uploaded by the customers in the website.
- System should approve registration Details and Create a user Account.
- System should display availabilities of Reservations on the website.
- System Should display Packages with facilities on the website so as Customers can check Packages.
- System should validate the login Credentials entered by the Registered User.
- System should display/ send the Reservation Request to the Admin and store details.
- System should provide ability to Upload Feedbacks and Experiences.
- System Should send feedback uploading Request to the admin for authorization and store the details/data.
- System should send the Reservation Cancellation Request to the admin and store details.
- System should store/ save modified User Account details.

- System should provide ability for Users to contact via social medias or send a message to the assigned officers mentioning the Requirement.

2. **Tour Guide (can access the back end of the system).**

User Requirement-

- Tour can login to the system by entering login credentials.
- Tour guide replies the messages send by the users and Guests.

System Requirement-

- System should validate login credentials entered by the tour guide.
- System Display the messages Query list to the tour guide.
- System allows for the coordination between tour guide and the Guest/Users.

3. **Administrator and Manager (can access the back end of the system).**

User requirements –

- Administrator signs into the website by providing required login credentials.
- Administrator can approve the reservations.
- Administrator can add and remove staff accounts.
- Administrator can activate and deactivate user account.
- Administrator can approve cancelled Reservations and remove marked reservation from calendar and arrange the refund.
- Administrator can update safari details (map, about safari and accommodation, images, video, packages)
- Administrator can add and remove share experiences.
- Administrator checks member feedbacks, contacts, and reviews.

- Manager can login to the system by entering login credentials.
- Manager can generate reports (Donation reports, Reservations reports and Hotel booking reports).
- Manager can manage Donations and funds.
- Manager can manage salaries of employees.

System requirements –

- System should validate the user login credentials.
- System should delete details of the deleted safari details by the user.
- System should delete details of the members deleted by the user.
- System should update the details of the downgraded users and modified accounts in the database.
- System stores the member feedbacks, contacts, and reviews and display them.
- System should approve cancelled reservations and remove marked reservation from calendar and refund by the user.
- System should validate login Credentials entered by the manager.
- System should generate Reports such as Donation report, Reservation Report, Hotel booking report for the manager.

4. **Developer (can access the back end of the system).**

User Requirements-

- Developer signs to the Website by entering login Credentials.
- Developer can Develop facilities of the website.
- Developer can fix the bugs and errors of the existing system.
- Developer can update the wildlife Quest System features.
- Developer can check the security of the system and protect the system from virus treats.

System Requirements-

- System Validate user login Credentials
- System notifies bugs and errors available in the existing system.
- System updates the website where bugs and errors are fixed.
- System displays the system features.
- System allows new updates.
- System can store all the system features.

3.1.2 Non-Functional Requirements

Non-functional requirements simply known as quality attributes. It describes the characteristics of the system that are not directly concerned with specific functionality. Non-functional requirements may be more critical than functional requirements. If these are not achieved, the system may be useless.

Speed

- The system must have good speed.
- The system can access more users at same time without any failures.

Availability

- The system should be available in 24/7.

User friendly

- The system should be accessible to users with low IT literacy.

Reliability

- The system must have ability to detect the invalid user credentials.

Security

- The system should have ability to prevent unauthorized access, misuse, forgery and secure user data.
- Also, by providing unique user ID and password, no one can access the system by using any other’s user ID and password.

Scalability

- The system should be able to handle a higher workload on-demand.

Performance

- Admin can add, edit, remove, update properties
- Any number of users can be able to access the system at the same time and the response of the system regarding to the user requests will be very high.

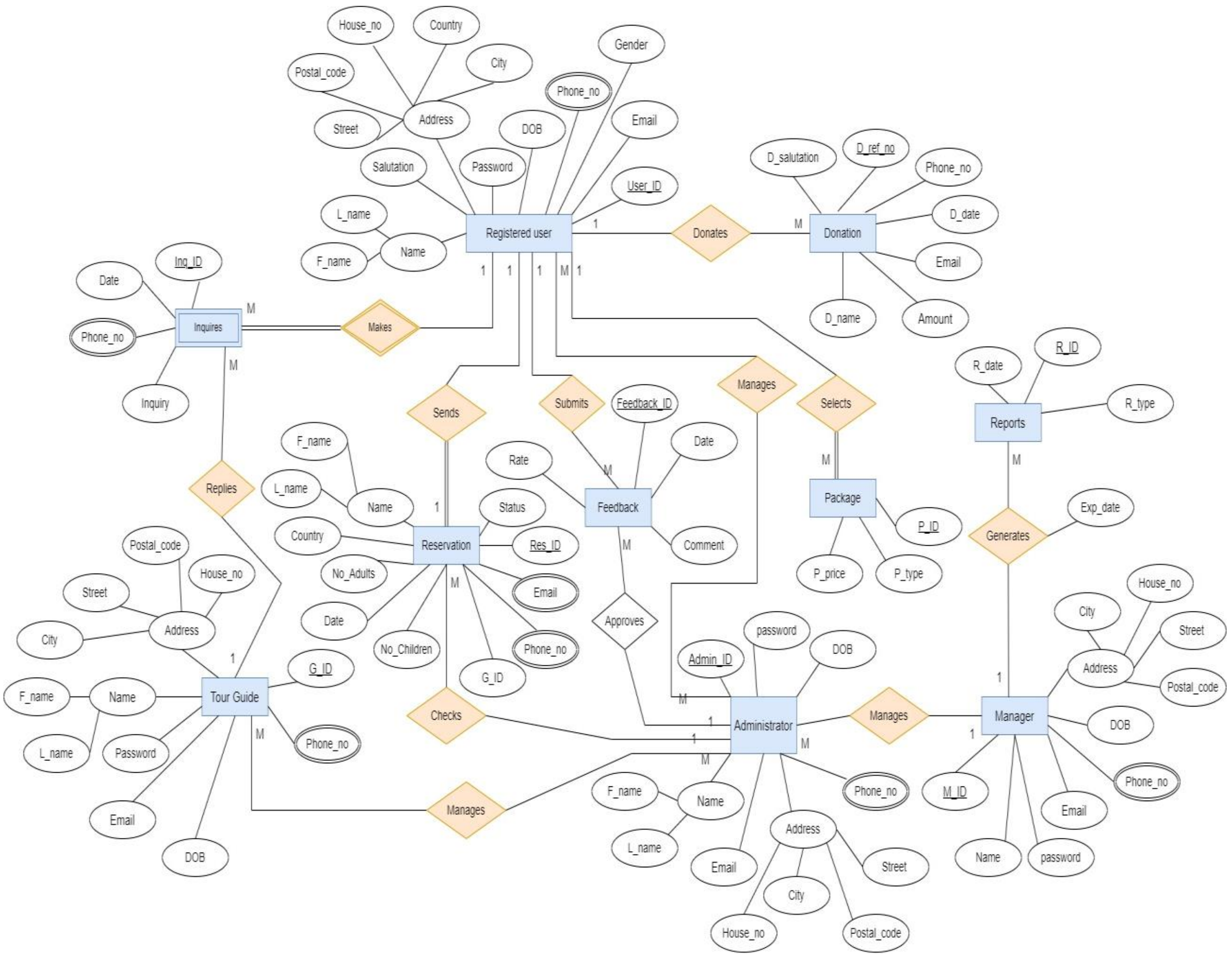
3.2 Data Requirements

- Registered User
  - User_ID
  - Salutation
  - F_name
  - L_name
  - House_no
  - Street
  - Postal_code
  - City
  - Country
  - Password
  - DOB
  - Gender
  - Phone_no
  - email
- Tour Guide
  - G_ID
  - F_name
  - L_name
  - House_no
  - Street
  - Postal_code
  - City
  - Password
  - DOB

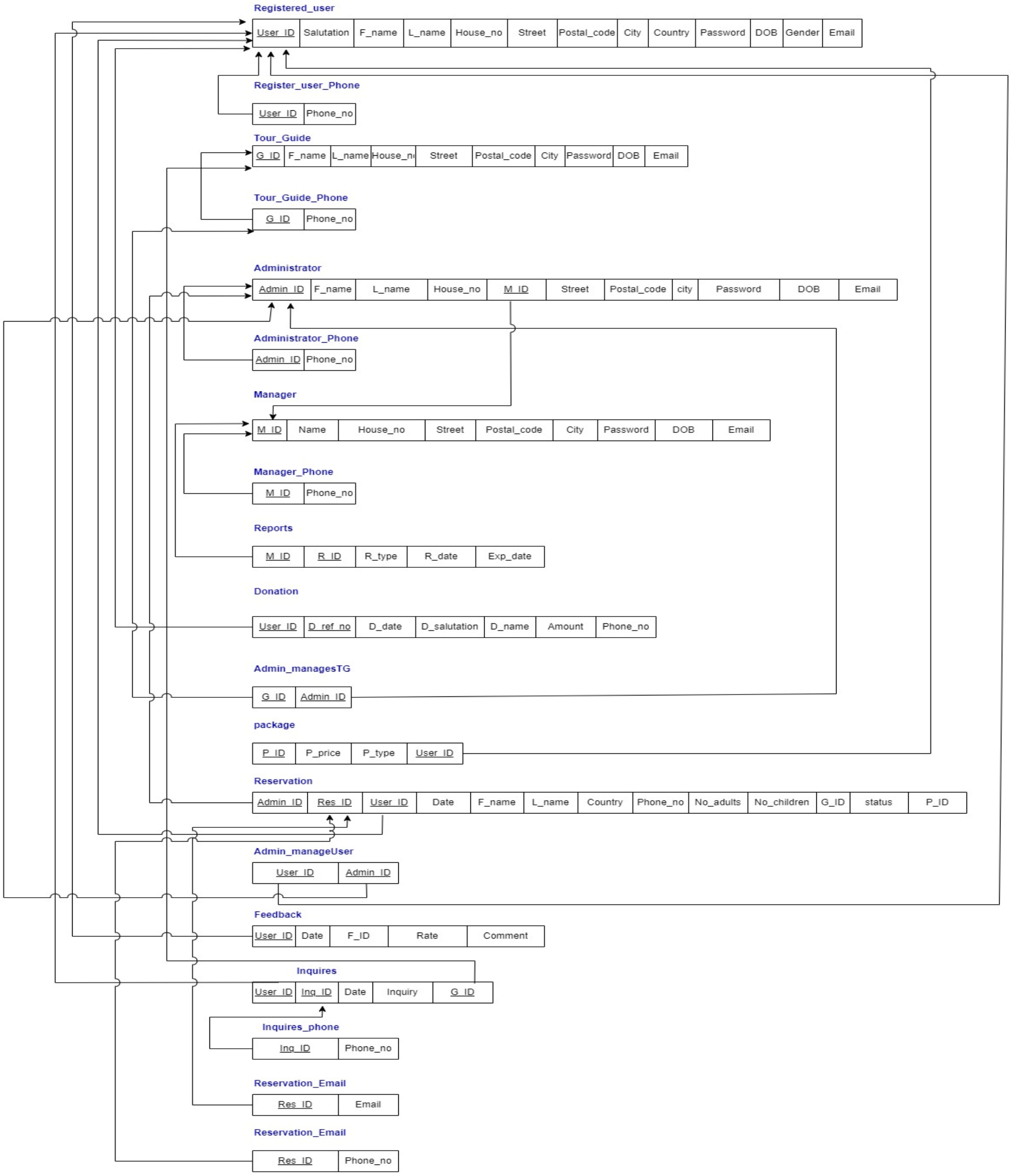
  - Phone_no
  - email
- Administrator
  - Admin_ID
  - F_name
  - L_name
  - House_no
  - Street
  - Postal_code
  - City
  - Password
  - DOB
  - Phone_no
  - email
- Manager
  - M_ID
  - Name
  - House_no
  - Street
  - Postal_code
  - City
  - Password
  - DOB
  - Phone_no
  - email
- Donation
  - D_ref_no
  - D_date
  - D_salutation
  - D_name
  - Amount
  - Phone_no
  - Email

- Reports
  - R_type
  - R_ID
  - R_date
- Package
  - P_price
  - P_ID
  - P_type
- Reservation
  - Res_ID
  - Date
  - F_name
  - L_name
  - Country
  - Phone_no
  - Email
  - No_adult
  - No_children
  - G_ID
  - Status
  - Rate
  - Comment
- Feedback
  - Feedback_ID
  - Date
  - comment
  - rate
- Inquires
  - Inq_ID
  - Date
  - Phone_no
  - Inquiry

4 Entity Relationship (ER Diagram)



5 Relational Schema



Registered_user

User_ID	Salutation	F_name	L_name	House_no	Street	Postal_code	City	Country	Password	DOB	Gender	Email

Register_user_Phone

User_ID	Phone_no

Tour_Guide

G_ID	F_name	L_name	House_n	Street	Postal_code	City	Password	DOB	Email

Tour_Guide_Phone

G_ID	Phone_no

Administrator

Admin_ID	F_name	L_name	House_no	M_ID	Street	Postal_code	city	Password	DOB	Email

Administrator_Phone

Admin_ID	Phone_no

Manager

M_ID	Name	House_no	Street	Postal_code	City	Password	DOB	Email

Manager_Phone

M_ID	Phone_no

Reports

M_ID	R_ID	R_type	R_date	Exp_date

Donation

User_ID	D_ref_no	D_date	D_salutation	D_name	Amount	Phone_no

Admin_managesTG

G_ID	Admin_ID

package

P_ID	P_price	P_type	User_ID

Reservation

Admin_ID	Res_ID	User_ID	Date	F_name	L_name	Country	Phone_no	No_adults	No_children	G_ID	status	P_ID

Admin_manageUser

User_ID	Admin_ID

Feedback

User_ID	Date	F_ID	Rate	Comment

Inquires

User_ID	Inq_ID	Date	Inquiry	G_ID

Inquires_phone

Inq_ID	Phone_no

Reservation_Email

Res_ID	Email

Reservation_Email

Res_ID	Phone_no

6 SQL Queries

6.1 Data Base Create

```
/*Table registered user*/
CREATE TABLE Registered_user
(
       User_ID varchar(15) not null,
       Salutation varchar(6) not null,
       F_name varchar(40) not null,
       L_name varchar(40) not null,
       House_no varchar(30) not null,
       Street varchar(30) not null,
       Postal_code varchar(15) not null,
       City varchar(35) not null,
       Country varchar(20) not null,
       Password varchar(15) not null,
       DOB date not null,
       Gender varchar(10) not null,
       Email varchar(40) CHECK (Email LIKE '%_@__%.__%') not null,

       CONSTRAINT REG_USER_PK PRIMARY KEY(USer_ID)
);

 /*Table registered user phone*/
CREATE TABLE Registered_user_Phone
(
       User_ID varchar(15) not null,
       Phone_no decimal(10,0) not null,

       CONSTRAINT REG_USER_PHONE_PK PRIMARY KEY(USer_ID),
       CONSTRAINT REG_USER_PHONE_FK FOREIGN KEY(USer_ID) References Registered_user
(USer_ID)
);

/*Table Tour_Guide*/
CREATE TABLE Tour_Guide
(
       G_ID varchar(15) not null,
       F_name varchar(40) not null,
       L_name varchar(40) not null,
       House_no varchar(30) not null,
       Street varchar(30) not null,
       Postal_code varchar(15) not null,
       City varchar(35) not null,
       Password varchar(15) not null,
       DOB date not null,
       Email varchar(40) CHECK (Email LIKE '%_@__%.__%') not null,
       CONSTRAINT TOUR_GUIDE_PK PRIMARY KEY(G_ID)
);

CREATE TABLE Tour_Guide_Phone /*Table Tour Guide Phone*/
(
       G_ID varchar(15) not null,
       Phone_no decimal(10,0) not null,

       CONSTRAINT TOUR_GUIDE_PHONE_PK PRIMARY KEY(G_ID),
       CONSTRAINT TOUR_GUIDE_PHONE_FK FOREIGN KEY(G_ID) References Tour_Guide (G_ID)
);

/*Administrator Table */
CREATE TABLE Administrator (

    Admin_ID varchar(15) not null,
    F_name varchar(40) not null,
    L_name varchar(40) not null,
    House_no varchar(30) not null,
    M_ID varchar(15) not null,
    Street varchar(30) not null,
    Postal_code varchar(15) not null,
    city varchar(35) not null,
    Password varchar(15) not null,
    DOB date not null,
    Email varchar(35) CHECK (Email LIKE '%_@_%._%') not null,

    constraint Administrator_PK PRIMARY KEY (Admin_ID)

);


/*Administrator Phone numbers Table */
CREATE TABLE Administrator_Phone (

    Admin_ID varchar(15) not null,
    Phone_no decimal (10, 0) not null,

    CONSTRAINT Administrator_Phone_PK PRIMARY KEY (Admin_ID),
    CONSTRAINT Administrator_Phone_FK FOREIGN KEY (Admin_ID) REFERENCES
Administrator(Admin_ID)

);



/*Manager table */
CREATE TABLE Manager (

    M_ID varchar(15) not null,
       Name varchar(40) not null,
    House_no varchar(30) not null,
    Street varchar(30) not null,
    Postal_code varchar(15) not null,
    city varchar(35) not null,
    Password varchar(15) not null,
    DOB date not null,
    Email varchar(35) CHECK (Email LIKE '%_@_%._%') not null,

    CONSTRAINT Manager_PK PRIMARY KEY (M_ID)

);

/*Manager phone numbers Table */
CREATE TABLE Manager_Phone (

    M_ID varchar(15) not null,
    Phone_no decimal (10, 0) not null,

    CONSTRAINT Manager_Phone_PK PRIMARY KEY (M_ID),
    CONSTRAINT Manager_Phone_FK FOREIGN KEY (M_ID) REFERENCES Manager(M_ID)

);


/* Report Table*/
CREATE TABLE Reports (
    M_ID varchar(15) not null,
    R_ID varchar(15) not null,
       R_type varchar(20) not null,
    R_date date not null,
    Exp_date date not null,

    CONSTRAINT Reports_PK PRIMARY KEY(R_ID),
    CONSTRAINT Reports_FK FOREIGN KEY(M_ID) REFERENCES Manager(M_ID)

);


/*Donation Table*/
CREATE TABLE Donation (
   User_ID VARCHAR(15) not null,
   D_ref_no VARCHAR(15) not null,
   D_date DATE not null,
   D_salutation VARCHAR(6) not null,
   D_name VARCHAR(15) not null,
   Amount FLOAT not null,
   Phone_no decimal(10,0) not null,

   CONSTRAINT Donation_PK PRIMARY KEY (D_ref_no),
   CONSTRAINT Donation_FK FOREIGN KEY (User_ID) REFERENCES Registered_user(User_ID)
);
```

```
/* Reservation Table details */
create table Reservation(
       Admin_ID varchar(15) NOT NULL,
       Res_ID varchar(15) NOT NULL,
       User_ID varchar(15) NOT NULL,
       Date date NOT NULL,
       F_name varchar(30) NOT NULL,
       L_name varchar(30) NOT NULL,
       Country varchar(40) NOT NULL,
       Phone_no int NOT NULL,
       No_adults int NOT NULL,
       No_children int NOT NULL,
       G_ID varchar(15)NOT NULL,
       status varchar(30) NOT NULL,
       P_ID varchar(15) NOT NULL,
       constraint reservation_PK primary key (Res_ID),
       constraint reservation_FK foreign key (Admin_ID) references
Administrator(Admin_ID),
       constraint reservation1_FK foreign key (User_ID) references
Registered_user(User_ID),
       constraint Phone_no_CHK CHECK (Phone_no BETWEEN 0000000000 AND 9999999999),
);
/*Admin manager Table*/
CREATE TABLE Admin_managesTG (

    G_ID varchar(15) not null,
    Admin_ID varchar(15) not null,

    CONSTRAINT Admin_managesTG_PK PRIMARY KEY (Admin_ID),
    CONSTRAINT Admin_managesTG_FK FOREIGN KEY (G_ID) REFERENCES Tour_Guide_Phone(G_ID)

);


/* Packages Table*/
CREATE TABLE Packages (
    P_ID VARCHAR(15) not null,
    P_price FLOAT not null,
    P_type VARCHAR(15) not null,
    User_ID VARCHAR(15) not null,

    CONSTRAINT Packages_PK PRIMARY KEY(P_ID),
    CONSTRAINT Packages_FK FOREIGN KEY(User_ID) REFERENCES Registered_user(User_ID)

);

/*Admin manages user Table*/
CREATE TABLE Admin_manageUser (

    User_ID varchar(15) not null,
    Admin_ID varchar(15) not null,
    CONSTRAINT Admin_manageUser_PK PRIMARY KEY (User_ID),
       CONSTRAINT Admin_manageUser_FK1 FOREIGN KEY (User_ID) REFERENCES
Registered_user(User_ID),
       CONSTRAINT Admin_manageUser_FK2 FOREIGN KEY (Admin_ID) REFERENCES
Administrator(Admin_ID),

);


/* Feedback table */

create table Feedback(
       User_ID varchar(15) NOT NULL,
       Date date NOT NULL,
       F_ID varchar (15) NOT NULL,
       Rate float NOT NULL,
       Comment varchar(300),
       constraint Feedback_pk primary key (F_ID),
       constraint Feedback_FK foreign key(User_ID) references Registered_user(User_ID)
);


/*Inquires table*/

create table Inquires(
       User_ID varchar(15) NOT NULL,
       Inq_ID varchar(15) NOT NULL,
       date date NOT NULL,
       Inquiry varchar(500) NOT NULL,
       G_ID varchar(15)NOT NULL,
       constraint inquires_pk primary key (Inq_ID),
       constraint inquires_FK foreign key(User_ID) references Registered_user(User_ID),
       constraint inquires2_FK foreign key(G_ID) references Tour_guide(G_ID),
);


/* Inquiree_phone Table */

create table Inquiree_phone(
Inq_ID varchar(15)NOT NULL,
phone_no int ,
CONSTRAINT inquiree_phone_PK PRIMARY KEY (Inq_ID),
constraint inquiree_phone_FK foreign key(Inq_ID ) references Inquires(Inq_ID ),
);


/* Reservation Email Table */

create table Reservation_Email(
Res_ID varchar(15) NOT NULL,
Email varchar(50)NOT NULL,
CONSTRAINT Reservation_Email_PK PRIMARY KEY (Res_ID),
constraint Reservation_Email_FK foreign key(Res_ID) references Reservation(Res_ID),
constraint Reservation_Email_CHK CHECK (Email LIKE '%@%%'),
);


/* Reservation phone number Table */

create table Reservation_phone(
Res_ID varchar(15) NOT NULL,
Phone_no int ,
CONSTRAINT Reservation_phone_PK PRIMARY KEY (Res_ID),
constraint Reservation_phone_FK foreign key(Res_ID ) references Reservation(Res_ID),
);
```

6.2 Data Store in Data Base

```
/* Add table details */



/* Registered_user table details*/

INSERT INTO Registered_user VALUES ('W0001', 'Mr', 'Kamal', 'Samantha', 'No.56/2', 'Galle
road','dfaffad', 567, 'Sri lanka', '****', '1985-10-08','Male', 'kamal@gmail.com' );
INSERT INTO Registered_user VALUES ('W0002', 'Mr', 'Madura', 'Dimantha', 'No.9/7',
'Nugegoda road','Nugegoda', 347, 'Sri lanka', '****', '1985-10-08','Male',
'madura@gmail.com' );
INSERT INTO Registered_user VALUES ('W0003', 'Ms', 'Sakuni', 'Sansala', 'No.36/8',
'Gothami road','hdhkjhf', 963, 'Australia', '****', '1985-10-08','Male',
'sakuni@gmail.com' );
INSERT INTO Registered_user VALUES ('W0004', 'Mr', 'Sithum', 'Apsara', 'No.25/47',
'Mahinda road','dsfjhadsf', 347, 'Canada', '****', '1985-10-08','Male',
'dimantha@gmail.com' );
INSERT INTO Registered_user VALUES ('W0005', 'Mrs', 'Malki', 'Jayathilaka', 'No.5/12',
'Rajakaruna road','dfkajfa', 741, 'Sri lanka', '****', '1985-10-08','Male',
'malki@gmail.com' );


/*Register_user_Phone table details*/
INSERT INTO Registered_user_Phone VALUES ('W0001', 0747456123);
INSERT INTO Registered_user_Phone VALUES ('W0002', 0767845684);
INSERT INTO Registered_user_Phone VALUES ('W0003', 0787456128);
INSERT INTO Registered_user_Phone VALUES ('W0004', 0779854623);
INSERT INTO Registered_user_Phone VALUES ('W0005', 0115875247);
```

```
/*Tour_Guide Table details*/

INSERT INTO Tour_Guide VALUES ('G0001', 'Waruna' , 'Deshappriya', 'No.8/12', 'Mahameuna
Road', 964, 'Kandy', 'fkjdij', '1991-4-8', 'Deshappriya@gmail.com'  );
INSERT INTO Tour_Guide VALUES ('G0002', 'Silwa' , 'Sinhawanshe', 'No.8/45', 'Hewagama
Road', 654,'Jaffna', 'ejijijk', '1985-9-15', 'Hewa@gmail.com'  );
INSERT INTO Tour_Guide VALUES ('G0003', 'Waruna' , 'Wijerathna', 'No.18', 'Nelligala
Road', 964, 'Kandy', 'fkjdij', '1981-4-4', 'Wijerathna@gmail.com'  );
INSERT INTO Tour_Guide VALUES ('G0004', 'Nelum' , 'Sliva', 'No.1/62', 'Gegoda Road', 964,
'Kandy', 'fkjdij', '1991-4-8', 'silva@gmail.com'  );
INSERT INTO Tour_Guide VALUES ('G0005', 'Kamal' , 'Rathnayaka', 'No.1/962', 'kesbawa
Road', 954, 'Borella', 'xeedj', '1991-5-21', 'Kamal@gmail.com'  );


/*Tour_Guide_Phone Table details*/

INSERT INTO Tour_Guide_Phone VALUES ('G0001', 0759638252);
INSERT INTO Tour_Guide_Phone VALUES ('G0002', 0755475561);
INSERT INTO Tour_Guide_Phone VALUES ('G0003', 0765245322);
INSERT INTO Tour_Guide_Phone VALUES ('G0004', 0754278232);
INSERT INTO Tour_Guide_Phone VALUES ('G0005', 0786424585);


/*Administator Table details*/

INSERT INTO Administrator
(Admin_ID,F_name,L_name,House_no,M_ID,Street,Postal_code,city,Password,DOB,Email) VALUES
('A0001', 'sithum', 'munasinhe', 'No.273', 'M0001', 'Kota road', 586, 'Sithawa',
'suhjwghuy' ,'1993-03-02', 'munasinhe@gmail.com');
INSERT INTO Administrator
(Admin_ID,F_name,L_name,House_no,M_ID,Street,Postal_code,city,Password,DOB,Email) VALUES
('A0002' ,'hethum', 'hathurusinhe', 'No.87', 'M0002', 'King road', 865, 'Pasyala',
'swdefdf' ,'1999-03-04', 'hathurusinhe@gmail.com');
INSERT INTO Administrator
(Admin_ID,F_name,L_name,House_no,M_ID,Street,Postal_code,city,Password,DOB,Email) VALUES
('A0003', 'Sethmi' ,'ijesinghe', 'No.67', 'M0003','Mathil road', 745, 'jaffna',
'shhysuj', '1994-03-05', 'Wijesinghe@gmail.com');
INSERT INTO Administrator
(Admin_ID,F_name,L_name,House_no,M_ID,Street,Postal_code,city,Password,DOB,Email) VALUES
('A0004', 'Nirmal', 'alupahana' ,'No.54', 'M0004', 'Nilmal road', 632, 'Kalutara',
'sdauijm', '1997-02-24', 'kalupahana@gmail.com');
INSERT INTO Administrator
(Admin_ID,F_name,L_name,House_no,M_ID,Street,Postal_code,city,Password,DOB,Email) VALUES
('A0005', 'Jayantha', 'jayathilaka', 'No.90', 'M0005', 'Kahatagas road', 953,
'Athurugiriya', 'sdauijm', '1989-07-09', 'ayathilaka@gmail.com');
```

```
/*Administrator_Phone Table details*/

INSERT INTO Administrator_Phone VALUES ('A0001', 0769576321);
INSERT INTO Administrator_Phone VALUES ('A0002', 0798745624);
INSERT INTO Administrator_Phone VALUES ('A0003', 0769645872);
INSERT INTO Administrator_Phone VALUES ('A0004', 0789643425);
INSERT INTO Administrator_Phone VALUES ('A0005', 0719755466);

/*      Manager table details*/

INSERT INTO Manager VALUES ('M0001', 'lalith kothalawala', 'No.98', 'Gamunu road', 654,
'Malabe', 'shuidh', '1996-05-30', 'kothalawala@gmail.com');


/* Manager_Phone table details*/

INSERT INTO Manager_Phone VALUES ('M0001', 0119632856);


/* Reports table details */

INSERT INTO Reports VALUES  ('M0001', 'R0001', 'Don_report','2019-12-11', '2019-12-31');
INSERT INTO Reports VALUES  ('M0001', 'R0002', 'Res_report','2019-12-21', '2019-12-31');
INSERT INTO Reports VALUES  ('M0001', 'R0003', 'Res_report','2020-12-19', '2020-12-31');
INSERT INTO Reports VALUES  ('M0001', 'R0004', 'Don_report','2020-12-28', '2020-12-31');


/*Donation table details*/
INSERT INTO Donation VALUES ('W0001', 'D0001', '2019-05-08', 'Mr', 'Kamal' , 10000 ,
0712546489);
INSERT INTO Donation VALUES ('W0002', 'D0002', '2019-06-12', 'Mr', 'Madura' , 20000 ,
0725487683);
INSERT INTO Donation VALUES ('W0003', 'D0003', '2019-05-07', 'Ms', 'Sakuni' , 50000 ,
0776451367);
INSERT INTO Donation VALUES ('W0004', 'D0004', '2020-07-30', 'Mr', 'sithum', 100000,
0706895426);
INSERT INTO Donation VALUES ('W0005', 'D0005', '2021-10-01', 'Mrs', 'Malki', 90000,
0714465488);


/*Admin Admin_managesTG */
INSERT INTO Admin_managesTG VALUES ( 'G0002', 'A0003');
INSERT INTO Admin_managesTG VALUES ('G0003', 'M0001');
INSERT INTO Admin_managesTG VALUES ('G0004', 'M0002');
INSERT INTO Admin_managesTG VALUES ('G0001', 'M0004');
```

```
/*Packages table details*/
INSERT INTO Packages VALUES ('P0001', 50000, 'PT0001', 'W0001');
INSERT INTO Packages VALUES ('P0002', 75000, 'PT0002', 'W0002');
INSERT INTO Packages VALUES ('P0003', 90000, 'PT0003', 'W0003');
INSERT INTO Packages VALUES ('P0004', 85000, 'PT0003', 'W0004');
INSERT INTO Packages VALUES ('P0005', 67000, 'PT0003', 'W0005');

/*Reservation table details*/
INSERT INTO Reservation VALUES ('A0001', 'RE0001', 'W0001', '2019-06-19', 'Kamal',
'Samantha', 'Sri lanka', 0712546489, 1, 2, 'G0001', 'Approved', 'P0001');
INSERT INTO Reservation VALUES ('A0002', 'RE0002', 'W0002', '2019-12-12', 'Madura',
'Dimantha', 'Sri lanka', 0725487683, 2, 1, 'G0002', 'Approved', 'PT0002');
INSERT INTO Reservation VALUES ('A0003', 'RE0003', 'W0003', '2020-03-29', 'Sakuni',
'Sansala',  'Australia', 0776451367, 4, 3, 'G0003', 'Approved', 'P0003');
INSERT INTO Reservation VALUES ('A0004', 'RE0004', 'W0004', '2021-06-06', 'Sithum',
'Apsara',  'Canada', 0706895426, 3, 2, 'G0004', 'Approved', 'P0004');
INSERT INTO Reservation VALUES ('A0005', 'RE0005', 'W0005', '2022-01-18', 'Malki',
'Jayathilaka', 'Sri lanka', 0714465488, 2, 2, 'G0005', 'Cancelled', 'P0005');

/* Admin_manageUser table details*/
INSERT INTO Admin_manageUser VALUES ('W0001', 'A0001');
INSERT INTO Admin_manageUser VALUES ('W0002', 'A0002');
INSERT INTO Admin_manageUser VALUES ('W0003', 'A0003');
INSERT INTO Admin_manageUser VALUES ('W0004', 'A0004');
INSERT INTO Admin_manageUser VALUES ('W0005', 'A0005');

/*Feedback table details*/
INSERT INTO Feedback VALUES ('W0001', '2019-08-11', 'F0001', 5, 'Nice journey');
INSERT INTO Feedback VALUES ('W0002', '2019-11-25', 'F0002', 3, 'Good Service');
INSERT INTO Feedback VALUES ('W0003', '2020-02-06','F0003', 4, 'Perfect place');
INSERT INTO Feedback VALUES ('W0004', '2020-04-13', 'F0004', 5, 'Wonderful Expericence');
INSERT INTO Feedback VALUES ('W0005', '2021-04-27', 'F0005', 4, 'Enjoyful');

/*Inquires*/
INSERT INTO Inquires VALUES('W0001','IN0001','2022-01-01','can i know about package
details?','G0001');
INSERT INTO Inquires VALUES('W0002','IN0002','2022-01-03','how to requests for requests
cancelation?','G0002');
INSERT INTO Inquires VALUES('W0003','IN0003','2022-01-08','let me know available dates
for reservation?','G0003');
INSERT INTO Inquires VALUES('W0004','IN0004','2022-02-03','can i donate funds without
registering for the system?','G0004');
INSERT INTO Inquires VALUES('W0005','IN0005','2022-03-09','can we share our experiences
in your website and how to do that?','G0005');

/*Inquires_Phone*/
INSERT INTO Inquiree_phone VALUES('IN0001',0753243154);
INSERT INTO Inquiree_phone VALUES('IN0002',0767861245);
INSERT INTO Inquiree_phone VALUES('IN0003',0789800780);
INSERT INTO Inquiree_phone VALUES('IN0004',0779123456);
INSERT INTO Inquiree_phone VALUES('IN0005',0789088990);
```

```
/*Reservation_Email*/
INSERT INTO Reservation_Email VALUES('RE0001','kamal@gmail.com');
INSERT INTO Reservation_Email VALUES('RE0002','madura@gmail.com');
INSERT INTO Reservation_Email VALUES('RE0003','sakuni@gmail.com');
INSERT INTO Reservation_Email VALUES('RE0004','dimantha@gmail.com');
INSERT INTO Reservation_Email VALUES('RE0005','malki@gmail.com');

/*Reservation_Phone*/
INSERT INTO Reservation_Phone VALUES('RE0001',0753243154);
INSERT INTO Reservation_Phone VALUES('RE0002',0767861245);
INSERT INTO Reservation_Phone VALUES('RE0003',0789800780);
INSERT INTO Reservation_Phone VALUES('RE0004',0779123456);
INSERT INTO Reservation_Phone VALUES('RE0005',0789088990);
```

7 Performance Requirement

A major role is played by Performance Requirements to make the system successful. They are as follow,

- System must be active 24 hours, 365 days for a Registered user to access the system without any inconvenience.
- A Registered User can access the system numerous times by entering his/ her login credentials.
- Login process and loading the pages must be done within few seconds.
- Speed and Usability are the performance requirements for this system.
- Registered user can view safari details.
- Registered user can edit or delete his/her account details.
- System loads within a minimum time.
- Administrator can add or remove shared experiences and feedbacks, Approve Reservation Requests and cancel reservation requests, manage User accounts, edit/ update safari details.
- System allows Guests to Donate funds without login to the system.
- System allows Tour Guide to respond the Users messages.
- Developer can Develop new function to the website.
- Developer can update the system.
- Developer needs to detect bugs and errors to fix them.
- Design user friendly user interface.
- Users must be able to access the website at anytime using any device or browser.
- System provide ability for Administrator to manage the staff accounts.

8 Security Requirements

- Personal details of users should be encrypted before send to the database.
- Unauthorized users should be unable to access restricted features.
- The database should have a backup of all the data in the system.
- The password of user account must be a strength password which is included uppercase letters, lowercase letters, numbers, and special characters.
- For one email address, there should be only one user account.
- Database server must maintain with redundant server.
- Only the administrators can access and modify the data of the system.