# Sri Lanka Institute of Information Technology


Assignment 01

KGL_01

# Information Systems and Data Modeling

# IT1090

# Group details

Group Number: KGL_01

Project Title: Laboratory Information Management System

| | Student ID | Student Name | Email | Contact Number |
|---|---|---|---|---|
| 1 | IT21257018 | Rathnayaka K.P.S.K | it21257018@my.sliit.lk | 0716378320 |
| 2 | IT21238444 | Wimalarathna D.M.A.T | it21238444@my.sliit.lk | 0742135589 |
| 3 | IT21256950 | Dayananda N.C.E | it21256950@my.sliit.lk | 0717155007 |
| 4 | IT21195570 | Herath H.M.K.C.B | it21195570@my.sliit.lk | 072 8625364 |
| 5 | IT21254420 | Dewpura D.D.W.C | It21254420@my.sliit.lk | 0764190521 |

# Table of Contents

## Hypothetical Scenario

The Laboratory Information Management System (LIMS) is a web-based system that support modern laboratory operations. This system has main two types of users who can access the system. They are registered user and the unregistered user(guest). The system allows both users to view the system but only a registered user can use the system for a lab test reservation.

An unregistered user can become a registered user by signing up to the system. The users who are registered can access the system by logging in to the system. After logging to the system, the user can make an appointment for a lab test. Once registered, the user can take the lab test according to the user requirements after paying the payment. The payment can be done by credit or master card. A user can view the lab test is done.

Admin and developer are responsible for maintaining the security, error handling, developing and updating of the system.

# Requirement Analysis

## Main requirements for the system

The Laboratory Information Management System (LIMS) is a reliable and well-planned web-based system that support modern laboratory operations. LMIS is a system that developed and implemented with all essential requirements of the system. There are two such set of requirements which are, functional requirement and non-functional requirements.

## Functional requirements

Functional requirements state how the users will interact with the system. The Laboratory Information Management System also includes a set of main functional requirements. There are as follows.

The website can be accessed by unregistered users, registered users, technician, custom care officials, developers and the system administrator. However, the accessible areas are different from each other.

## Registered / Unregistered user

Access: -

- Customers (registered / unregistered) only have access to the client side of the system.
- Customers can access website, view, and make lab test reservation.

## Registration: -

- Unregistered users can register to the system by providing appropriate details such as name, e-mail address, phone number, and creating and confirming a password for the new user account.
- Registered users can login to the system by providing username and password. If user forget the password, user should provide e-mail or phone number. The user has facility to reset the password.
- Registered users can edit their user profile.

## Register for a lab test: -

- Customers can select the lab test they want through the web platform.
- Customers can register for a lab test by entering username, password, e-mail, NIC number, mobile number, address, test type, consultant name, and consultancy.
- Customers are provided with detail information's of protocols follows when doing the lab test.

## Payment: -

- Customer can pay online via credit or master card.
- Customer should enter valid payment credentials and confirm payment.
- Customers receive a payment receipt via user e-mail as confirmation of the payment.

## Feedback and customer support: -

- Customers can rate and give feedback about lab tests and service of the website.
- Customers can contact custom care officials for inquiries and issues regarding reservations, payments, and other services.

## Technician

- Technician is allowed to access both client-side and server-side of the system.
- Technician can update/edit lab test information and add or remove lab test to/from the system.
- Technician can update/edit customer information.
- Technician can upload lab test reports to the relevant customer and update lab test list.
- Technician can view all reservation and payments records and lab test result history of customers and generate monthly reports containing said information.

## Developer

- Developer has access to both client-side and server-side of the system.
- Developer can maintain and update the system.
- Developer can edit the database.
- Developer is responsible for maintaining the security of the system.

## System Administrator

- Administrator has access to both client-side and server-side of the system.
- Administrator can update/edit customer information.
- Administrator can review and process a user- confirmed reservations.
- Administrator can manage financial procedure.

## Nonfunctional requirements

Non-functional requirements are the characteristics that maintain the system's implicit expectations or quality. Functional requirements as well as non-functional requirements are important for the system and its functionality.

### Availability

- The system should be available 24 hours for users.
- Must be able to access all websites and services on the system.

### Usability

- The system must meet the user's requirements as soon as possible.
- The system should be easy for the user to use and meet the requirements.

### Security

- The privacy of the user's personal information must be verified.
- Protecting the bank account information contained in the system.
- The system's special information should not be accessible to other users or unauthorized users.
- Main and verify the uniqueness of system user account password.

### Reliability requirements

- Reliability is required to maintain a service or system.
- Laboratory management system reliability should be 100 percent guaranteed.
- Reliability is very important as users share their extremely personal information with this system.

## Maintainability requirements

- Must be able to provide the services that users expect.
- Being able to provide the service without any problems when multiple users are expecting the same service at the same time.
- The maintain ability requirements allow the website to troubleshoot or add new functionality.

## Data Requirements

**Registered user**

- UserID
- U_address
- uMail
- age
- Uf_name
- Ul_name

**Payment**

- PayID
- Amount
- Ptype
- Acc_No
- PayTime
- PayDate

**Lab test**

- TestNo
- TestName
- Test_description
- Price
- UpdateDate
- UpdateTime

**Appointment**

- AppNo
- A_Date
- A_Time
- Lab

**Technician**

- TecID
- T_address
- Tf_name
- Tl_name
- Tmail
- Tmobile

**Lab report**

- ReportID
- ReportName
- Up_date
- Up_time

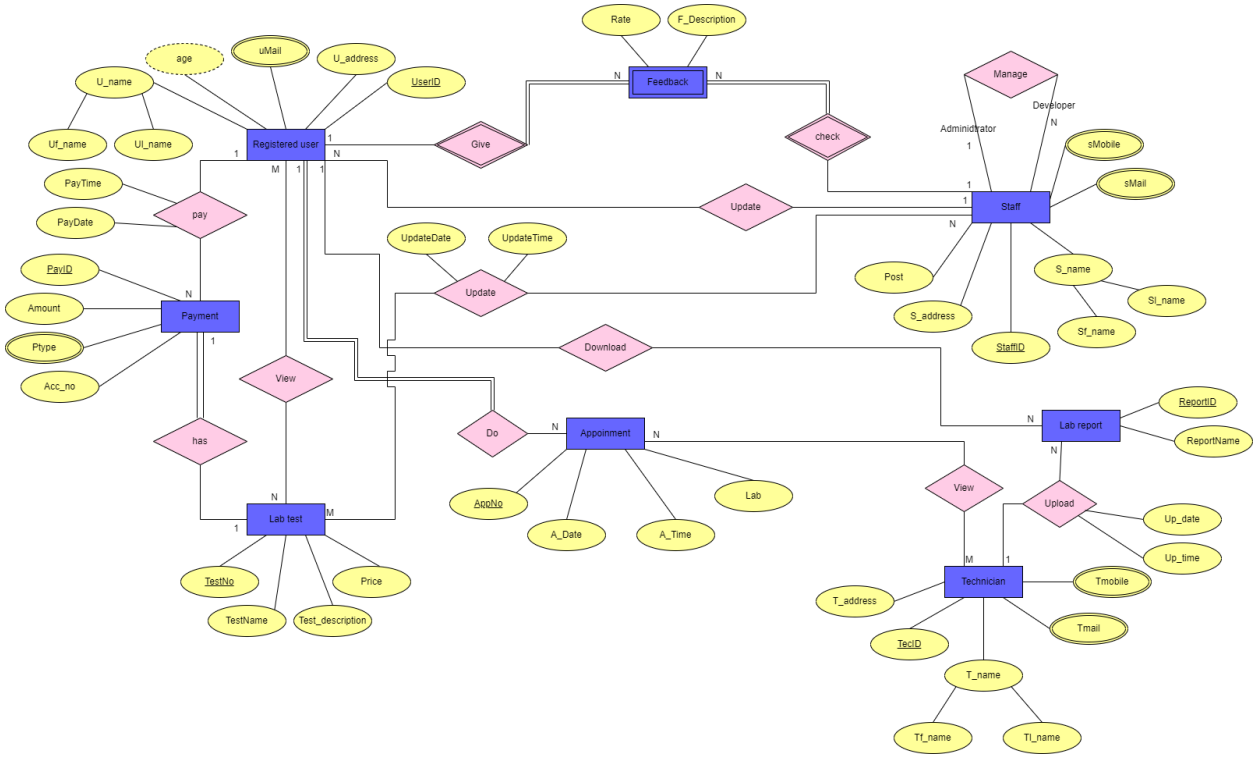
**Staff**

- StaffID
- S_address
- Post
- Sf_name
- Sl_name
- sMail
- sMobile

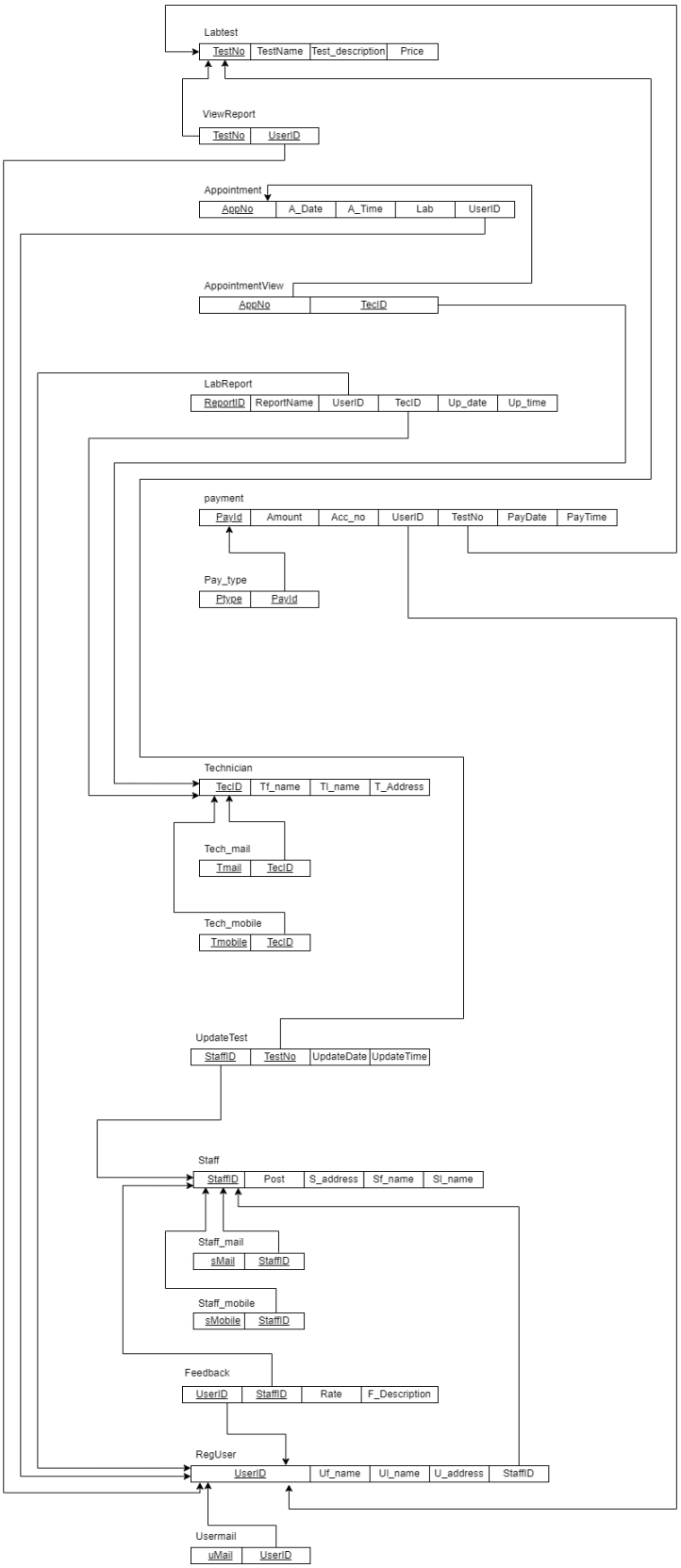
**Feedback**

- Rate
- F_Description

# Entity Relationship (ER) Diagram



# Relational Schema



```
--lab test table--
CREATE TABLE Labtest(
	TestNo INTEGER,
	TestName VARCHAR(50),
	Test_discription VARCHAR(200),
	Price INTEGER,

	constraint PK_Labtest
	primary key (testNo)
);

--Technician table--
CREATE TABLE Technician(
	TecID CHAR(6),
	Tf_name CHAR(10),
	Tl_name CHAR(20),
	T_Address VARCHAR(50),

	constraint PK_Technician
	primary key (TecID),

	constraint Technician_TecID_CK check (TecID LIKE '[t\T][0-9][0-9][0-9][0-9][0-9]')
);

--Staff table--
CREATE TABLE Staff(
	StaffID CHAR(6),
	Post VARCHAR(50),
	S_address VARCHAR(50),
	Sf_name CHAR(10),
	Sl_name CHAR(20),

	constraint PK_Staff
	primary key (StaffID),

	constraint Staff_StaffID_CK check (StaffID LIKE '[s\S][0-9][0-9][0-9][0-9][0-9]')
);

--RegUser table--
CREATE TABLE RegUser(
	UserID CHAR(6),
	Uf_name CHAR(10),
	Ul_name CHAR(20),
	U_Address VARCHAR(50),
	StaffID CHAR(6),

	constraint PK_RegUser
	primary key(UserID),

	constraint FK_RegUser
	foreign key(StaffID)
	references Staff(StaffID),

	constraint RegUser_UserID_CK check (UserID LIKE '[u\U][0-9][0-9][0-9][0-9][0-9]'),
      constraint RegUser_StaffID_CK check (StaffID LIKE '[s\S][0-9][0-9][0-9][0-9][0-9]')
);
```

```
--ViewReport table--
CREATE TABLE ViewReport(
       TestNo INTEGER,
       UserID CHAR(6),

       constraint PK_ViewReport
       primary key(TestNo,UserID),

       constraint FK_ViewReport
       foreign key(TestNo)
       references Labtest(TestNo),

       foreign key(UserID)
       references RegUser(UserID),

     constraint ViewReport_UserID_CK check (UserID LIKE '[u\U][0-9][0-9][0-9][0-9][0-9]')
);

--Appointment table--
CREATE TABLE Appointment(
       AppNo INTEGER,
       A_Date DATE,
       A_Time TIME,
       Lab VARCHAR(50),
       UserID CHAR(6),

       constraint PK_Appointment
        primary key(AppNo),

       constraint FK_Appointment
       foreign key(UserID)
       references RegUser(UserID),

    constraint Appointment_UserID_CK check (UserID like '[u\U][0-9][0-9][0-9][0-9][0-9]')
);

--AppointmentView table--
CREATE TABLE AppointmentView(
       AppNo INTEGER,
       TecID CHAR(6),

       constraint PK_AppointmentView
       primary key(AppNo,TecID),

       constraint FK_AppointmentView
       foreign key(AppNo)
       references Appointment(AppNo),

       foreign key(TecID)
       references Technician(TecID),

  constraint AppointmentView_TecID_CK check (TecID LIKE '[t\T][0-9][0-9][0-9][0-9][0-9]')
);
```

```
--LabReport table--
CREATE TABLE LabReport(
	ReportID CHAR(10),
	ReportName VARCHAR(50),
	UserID CHAR(6),
	TecID CHAR(6),
	Up_date DATE,
	Up_time TIME,

	constraint PK_LabReport
	primary key(ReportID),

	constraint FK_LabReport
	foreign key(UserID)
	references RegUser(UserID),

	foreign key(TecID)
	references Technician(TecID),

	constraint LabReport_ReportID_CK check (ReportID LIKE '[r\R][i\I][d\D][0-9][0-
9][0-9][0-9][0-9][0-9][0-9]'),
	constraint LabReport_UserID_CK check (UserID LIKE '[u\U][0-9][0-9][0-9][0-9][0-
9]'),
	constraint LabReport_TecID_CK check (TecID LIKE '[t\T][0-9][0-9][0-9][0-9][0-9]')
);



--payment table--
CREATE TABLE payment(
	PayID CHAR(10),
	Amount INTEGER,
	Acc_No INTEGER,
	UserID CHAR(6),
	TestNo INTEGER,
	PayDate DATE,
	payTime TIME,

	constraint PK_payment
	primary key(PayID),

	constraint FK_payment
	foreign key(UserID)
	references RegUser(UserID),

	foreign key(TestNo)
	references Labtest(TestNo),

	constraint payment_PayID_CK check (PayID LIKE '[p\P][i\I][d\D][0-9][0-9][0-9][0-
9][0-9][0-9][0-9]'),
	constraint payment_UserID_CK check (UserID LIKE '[u\U][0-9][0-9][0-9][0-9][0-9]')
);
```

```
--Pay_type table--
CREATE TABLE Pay_type(
	Ptype VARCHAR(50),
	PayID CHAR(10),

	constraint PK_Pay_type
	primary key(Ptype,PayID),

	constraint FK_Pay_type
	foreign key(PayID)
	references payment(PayID),

	constraint Pay_type_Ptype_CK check (Ptype IN ('Credit card','Master card')),
	constraint Pay_type_PayID_CK check (PayID LIKE '[p\P][i\I][d\D][0-9][0-9][0-9][0-
9][0-9][0-9][0-9]')
);



--Tech_mail table--
CREATE TABLE Tech_mail(
	Tmail CHAR(50),
	TecID CHAR(6),

	constraint PK_Tech_mail
	primary key(Tmail,TecID),

	constraint FK_Tech_mail
	foreign key(TecID)
	references Technician(TecID),

	constraint Tech_mail_Tmail_CK check (Tmail LIKE '%@%.%'),
	constraint Tech_mail_TecID_CK check (TecID LIKE '[t\T][0-9][0-9][0-9][0-9][0-9]')
);



--Tech_mobile table--
CREATE TABLE Tech_mobile(
	Tmobile INTEGER,
	TecID CHAR(6),

	constraint PK_Tech_mobile
	primary key(Tmobile,TecID),

	constraint FK_Tech_mobile
	foreign key(TecID)
	references Technician(TecID),

	constraint Tech_mobile_TecID_CK check (TecID LIKE '[t\T][0-9][0-9][0-9][0-9][0-
9]')
);
```

```
--UpdateTest table--
CREATE TABLE UpdateTest(
	StaffID CHAR(6),
	TestNo INTEGER,
	UpdateDate DATE,
	UpdateTime TIME,

	constraint PK_UpdateTest
	primary key(StaffID,TestNo),

	constraint FK_UpdateTest
	foreign key(StaffID)
	references Staff(StaffID),

	foreign key(TestNo)
	references Labtest(TestNo),

	constraint UpdateTest_StaffID_CK check (StaffID LIKE '[s\S][0-9][0-9][0-9][0-9][0-
9]')
);


--Staff_mail table--
CREATE TABLE Staff_mail(
	sMail VARCHAR(50),
	StaffID CHAR(6),

	constraint PK_Staff_mail
	primary key(sMail,StaffID),

	constraint FK_Staff_mail
	foreign key(StaffID)
	references Staff(StaffID),

	constraint Staff_mail_sMail_CK check (sMail LIKE '%@%.%'),
	constraint Staff_mail_StaffID_CK check (StaffID LIKE '[s\S][0-9][0-9][0-9][0-9][0-
9]')
);

--Staff_mobile table--
CREATE TABLE Staff_mobile(
	sMobile INTEGER,
	StaffID CHAR(6),

	constraint PK_Staff_mobile
	primary key(sMobile,StaffID),

	constraint FK_Staff_mobile
	foreign key(StaffID)
	references Staff(StaffID),

	constraint Staff_mobile_StaffID_CK check (StaffID LIKE '[s\S][0-9][0-9][0-9][0-
9][0-9]')
);
```

```
--Feedback table--
CREATE TABLE Feedback(
	UserID CHAR(6),
	StaffID CHAR(6),
	Rate INTEGER,
	F_Discription VARCHAR(200),

	constraint PK_Feedback
	primary key(UserID,StaffID),

	constraint FK_Feedback
	foreign key(StaffID)
	references Staff(StaffID),

	constraint Feedback_UserID_CK check (UserID LIKE '[u\U][0-9][0-9][0-9][0-9][0-
9]'),
	constraint Feedback_StaffID_CK check (StaffID LIKE '[s\S][0-9][0-9][0-9][0-9][0-
9]')
);



--Usermail table--
CREATE TABLE Usermail(
	uMail CHAR(20),
	UserID CHAR(6),

	constraint PK_Usermail
	primary key(uMail,UserID),

	constraint FK_Usermail
	foreign key(UserID)
	references RegUser(UserID),

	constraint Usermail_uMail_CK check (uMail LIKE '%@%.%'),
	constraint Usermail_UserID_CK check (UserID LIKE '[u\U][0-9][0-9][0-9][0-9][0-9]')
);
```

```
--Insertion of Sample Data--

Insert Into Labtest Values (0001,'PCR','A test to detect genetic material from a specific
organism, such as a virus.',7500.00);
Insert Into Labtest Values (0002,'Lipid profile','A blood test that can measure the
amount of cholesterol and triglycerides in your blood.',1200.00);
Insert Into Labtest Values (0003,'Fasting blood sugar','A fasting blood sugar test
measures sugar (glucose) in your blood.',400.00);
Insert Into Labtest Values (0004,'Electrolytes','A blood test that measures if there is
an electrolyte imbalance in the body.',900.00);
Insert Into Labtest Values (0005,'TSH','A test that measures the amount of thyroid
stimulating hormone (TSH) in blood',1500.00);


Insert Into Technician Values ('T00001','Kamal','Perera','No 12,Colombo 10');
Insert Into Technician Values ('T00002','Nimal','Herath','No
125,Baththaramulla,Koswaththa');
Insert Into Technician Values ('T00003','Sunil','Bandara','No 15,Vehara,Kurunegala');
Insert Into Technician Values ('T00004','Amal','Dissanayaka','No 04,New road,Negombo');
Insert Into Technician Values ('T00005','Nayana','Hettiarachchi','No 02,TownHall,Colomobo
07');


Insert Into Staff Values ('S00001','Admin','No
132,Pothuhara,Kurunegala','Shenal','Rathnayaka');
Insert Into Staff Values ('S00002','Developer','No 121,2nd
street,Homagama','Namal','Rajapaksha');
Insert Into Staff Values ('S00003','Manager','No
78,Bandarawatta,Kurunegala','Kamala','Disasekara');
Insert Into Staff Values ('S00004','Admin','No 89,Cotta Road,Colombo','Rayan','Perera');
Insert Into Staff Values ('S00005','Manager','No 23,Malay
Street,colombo','Aruna','Bandara');


Insert Into RegUser Values ('U00001','Dinesh','Fernando','No 08,Rodney
street,Colombo','S00001');
Insert Into RegUser Values ('U00002','Supun','Herath','No 25,Galle
Road,Rathmalana','S00002');
Insert Into RegUser Values ('U00003','Hashini','Herath','No 90,Rodney
street,Colombo','S00003');
Insert Into RegUser Values ('U00004','Kavindu','Jayalath','No
145,Palanwatta,Pannipitiya','S00004');
Insert Into RegUser Values ('U00005','Sanadali','Gunawardana','No 32,Thurstan
Road,Colombo','S00005');


Insert Into ViewReport Values (0001,'U00001');
Insert Into ViewReport Values (0002,'U00002');
Insert Into ViewReport Values (0003,'U00003');
Insert Into ViewReport Values (0004,'U00004');
Insert Into ViewReport Values (0005,'U00005');
```

```
Insert Into Appointment Values (01,'2022-06-01','09:00','Lab1','U00001');
Insert Into Appointment Values (02,'2022-06-01','11:00','Lab2','U00002');
Insert Into Appointment Values (03,'2022-06-02','14:00','Lab1','U00003');
Insert Into Appointment Values (04,'2022-06-02','10:00','Lab3','U00004');
Insert Into Appointment Values (05,'2022-06-03','11:30','Lab3','U00005');


Insert Into AppointmentView Values (01,'T00001');
Insert Into AppointmentView Values (02,'T00002');
Insert Into AppointmentView Values (03,'T00003');
Insert Into AppointmentView Values (04,'T00004');
Insert Into AppointmentView Values (05,'T00005');


Insert Into LabReport Values ('RID0000001','PCR_Report','U00001','T00001','2022-06-
05','09:00');
Insert Into LabReport Values ('RID0000002','Lipid
profile_Report','U00002','T00002','2022-06-05','11:30');
Insert Into LabReport Values ('RID0000003','Fasting blood
sugar_Report','U00003','T00003','2022-06-06','14:00');
Insert Into LabReport Values ('RID0000004','Electrolytes_Report','U00004','T00004','2022-
06-06','10:00');
Insert Into LabReport Values ('RID0000005','TSH_Report','U00005','T00005','2022-06-
07','11:30');

Insert Into payment Values ('PID0000001',7500.00,9835110,'U00001','0001','2022-05-
30','09:00');
Insert Into payment Values ('PID0000002',1200.00,1447528,'U00002','0002','2022-05-
28','10:00');
Insert Into payment Values ('PID0000003',400.00,8875895,'U00003','0003','2022-05-
27','13:00');
Insert Into payment Values ('PID0000004',900.00,7748585,'U00004','0004','2022-05-
31','14:30');
Insert Into payment Values ('PID0000005',1500.00,3895685,'U00005','0005','2022-05-
26','18:00');


Insert Into Pay_type Values ('Credit card','PID0000001');
Insert Into Pay_type Values ('Master card','PID0000002');
Insert Into Pay_type Values ('Credit card','PID0000003');
Insert Into Pay_type Values ('Credit card','PID0000004');
Insert Into Pay_type Values ('Master card','PID0000005');


Insert Into Tech_mail Values ('kamal@gmail.com','T00001');
Insert Into Tech_mail Values ('nimal@gmail.com','T00002');
Insert Into Tech_mail Values ('sunil@gmail.com','T00003');
Insert Into Tech_mail Values ('amal@gmail.com','T00004');
Insert Into Tech_mail Values ('nayana@gmail.com','T00005');
```

```
Insert Into Tech_mobile Values (0712365453,'T00001');
Insert Into Tech_mobile Values (0765283651,'T00002');
Insert Into Tech_mobile Values (0747351924,'T00003');
Insert Into Tech_mobile Values (0718264121,'T00004');
Insert Into Tech_mobile Values (0772937642,'T00005');


Insert Into UpdateTest Values ('S00001',0001,'2022-01-01','08:00');
Insert Into UpdateTest Values ('S00002',0002,'2022-01-01','08:00');
Insert Into UpdateTest Values ('S00003',0003,'2022-01-01','08:00');
Insert Into UpdateTest Values ('S00004',0004,'2022-01-01','08:00');
Insert Into UpdateTest Values ('S00005',0005,'2022-01-01','08:00');


Insert Into Staff_mail Values ('shenal@gmail.com','S00001');
Insert Into Staff_mail Values ('namal@gmail.com','S00002');
Insert Into Staff_mail Values ('kamala@gmail.com','S00003');
Insert Into Staff_mail Values ('rayan@gmail.com','S00004');
Insert Into Staff_mail Values ('aruna@gmail.com','S00005');


Insert Into Staff_mobile Values (0716354772,'S00001');
Insert Into Staff_mobile Values (0765344282,'S00002');
Insert Into Staff_mobile Values (0715562112,'S00003');
Insert Into Staff_mobile Values (0748890726,'S00004');
Insert Into Staff_mobile Values (0707873362,'S00005');


Insert Into Usermail Values ('dinesh@gmail.com','U00001');
Insert Into Usermail Values ('supun@gmail.com','U00002');
Insert Into Usermail Values ('hashini@gmail.com','U00003');
Insert Into Usermail Values ('kavindu@gmail.com','U00004');
Insert Into Usermail Values ('sanadali@gmail.com','U00005');

Insert Into Feedback Values ('U00001','S00001',5,'Extremely good');
Insert Into Feedback Values ('U00002','S00002',5,'Extremely good');
Insert Into Feedback Values ('U00003','S00003',4,'Very good');
Insert Into Feedback Values ('U00004','S00004',4,'Very good');
Insert Into Feedback Values ('U00005','S00005',3,'Good');
```

## Performance considerations

- The system should up and running 24 hours.
- Allow the user to log in and sign into the system at any time.
- Ability to provide services to the user at any time.
- Users can register for lab test at any time and retrieve their old lab reports.
- Registered users can update and delete their accounts at any time.
- Management can manage the customer details and staff details.
- Ability to update the system so that none of the information on the system is destroyed.
- Users can give feedback about the services of the system.
- The system should always update with new information.

## Security Requirements

- The privacy of the user's personal information must be verified.
- The user must enter and use a strong password when registering in the system.
- Use capital letters, simple letters numbers and symbols for password.
- User must validate all information when registering.
- Only registered users should be allowed to log in to the system, unauthorized access to the system should not allowed.
- The security of the user data contained in the system should be ensured and the data should not be accessed and accessed by an outsider.
- Enable access to user accounts and retrieval of information only by the user and relevant staff.
- Management can reply for the feedbacks.
- Users can read other feedbacks.
- Viruses coming from outside the system must be able to protect themselves from false information.

# Contribution of Project

| Student ID | Student Name | Individual contribution towards the project | Lessons learned by assignment |
|---|---|---|---|
| IT21257018 | Rathnayaka K.P.S.K | As the leader, I worked with the rest of the team to make project correctly. And I helped to create SQL Queries as well. My team members are very good. So, I always tried to make them happy. | I learned how to write SQL queries well.<br><br>I learned how to draw ER diagram and its advantages |
| IT21238444 | Wimalarathna D.M.A.T | Help to write requirements analysis, performance considerations and security requirements. Help to create ER diagram. | Learned how to draw and create er diagram. Learned how to show relationships in ER diagram. |
| IT21256950 | Dayananda N.C.E | Contribute to the drawing of the ER diagram. | I was able to draw an ER diagram and learn about its relations. |
| IT21195570 | Herath H.M.K.C.B | Helped in making of ER diagram and relational schema. | Learned how to draw a relational schema and got an idea about entities, attributes and relations. |
| IT21254420 | Dewpura D.D.W.C | Help to write requirement analysis and SQL queries. Help to draw ER diagram and Relational schema. | Learned how to draw an ER diagram and its relationships. Learned how to implement database and how to insert data into tables. |