SYRACUSE UNIVERSITY

FOOD SERVICES

Rahul Narpatraj Rathod
Syracuse University
School of Information Studies
IST 659 – Project Implementation Report

Table of Contents

Background

SU Food Services is committed to serving students of Syracuse University with healthy, flavorful, and nutritious food. You can find them at every corner on campus with dining centers, food courts, cafes, catering, and vending to meet all your dining needs. They look forward to being a part of your daily activities on the Syracuse University campus. Currently, SU Food Service offers many services on campus for students and staff of Syracuse University. I am primarily focusing on solving the problems of dining halls which are across every corner of the campus.

Project Summary

This project focuses on designing a database for the Syracuse University Food Services. Currently, SU Food Service offers many services on campus for students and staff of Syracuse University. I am primarily focusing on solving the problems of dining halls which are across every corner of the campus.

The system will store the information about different dining halls, commercial and student employees working there, their shifts/timings and number of hours they worked at their workplace, also it will show information about dropped shifts and picked shifts such Who dropped the shift? Why he/she dropped the shift? What is the time of dropped shift? Where was the shift? Who picked up the shift?

Student employees and commercial workers have different shifts that they need to be picked up according to their schedule. Thus, there is an absence of a system where these students have to physically be present to pick these shifts up. Moreover, every time a student wants to sub or pick a shift he has to mail the SU Food Service Management. Thus, everything is manual and there is no level of customization to this data.

For this, we need to develop a database application that will change the process of sub shifts. Through this application, the student employees who are willing to drop their shift can upload their shift details and reason on the application which can be accessed by management and student employees. As a result, this will eliminate the huge number of emails that all student employees receive. So, if a student wants a sub shift he can directly login to the application and see what shifts

are available. This will eliminate the cumbersome of sub shift mails that popup on student's mail ids.

The report comprises of the database requirement in form of business rules, relationship between the various entities and attributes and fully attributed relational data models (ERD). The report also addresses the major queries and information that may be required by various users of this system.

Entity and Attribute Table

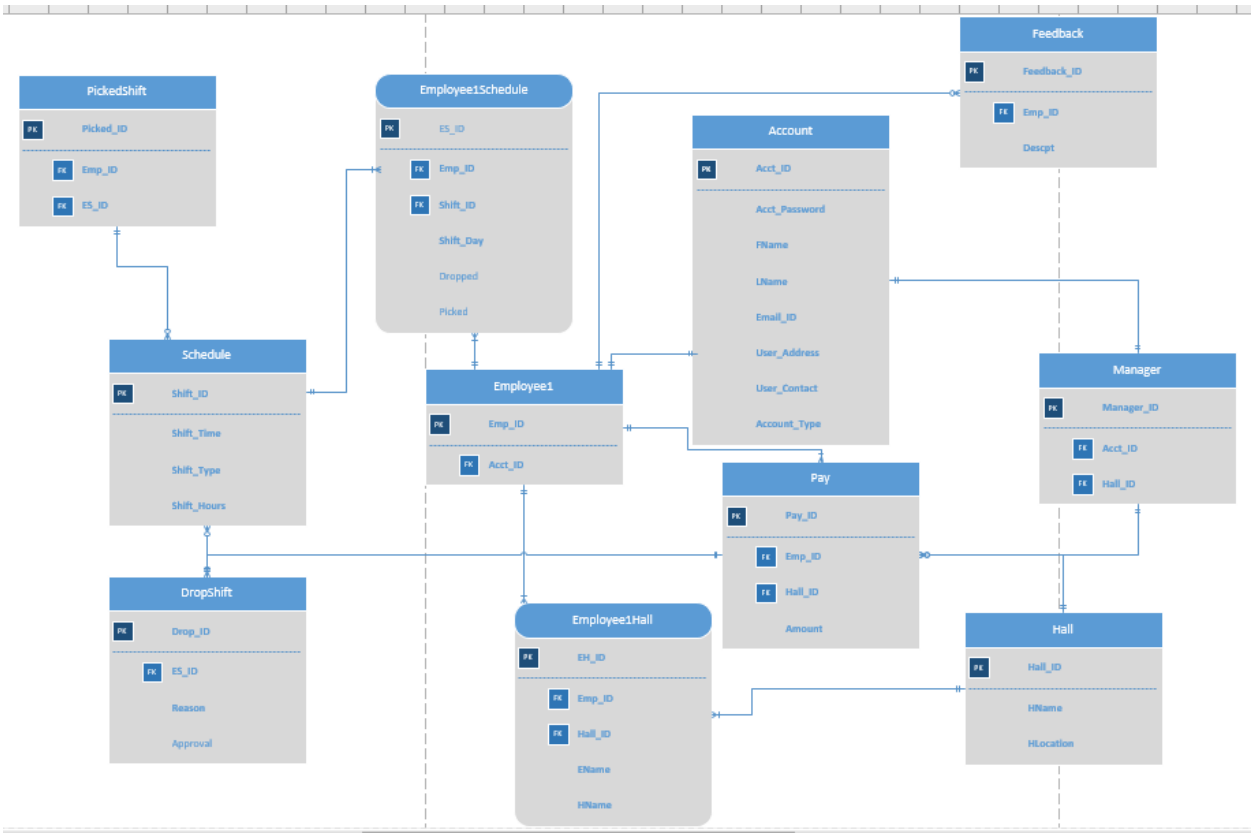
Objects	Description
1. Account	Contains all the user information of Employees and Managers working in different halls
a. Acct_ID	Primary Key uniquely identify the specific user
b. Acct_Password	Required for authenticating the user
c. FName	User's First Name
d. LName	User's Last Name
e. Email_ID	User's Email Address
f. User_Address	This refers to User's residential address
g. User_Contact	User's contact details for instance the phone number

h. Account_Type	Determines the type of Account for instance Student employee or Manager
2. Dining Hall	Can be any dining hall across the campus
a. Hall_ID	Primary Key uniquely identifies the hall
b. HName	Name of Hall
c. HLocation	Location of Hall
3. Employee	Employees working at respective dining halls
a. Emp_ID	Primary Key uniquely identifies the employee
b. Acct_ID	Foreign Key that will help to identify is the user an employee
4. Manager	Manager of respective dining halls
a. Manager_ID	Primary Key uniquely identifies the manager
b. Acct_ID	Foreign Key that will help to identify is the user a manager
c. Hall_ID	Foreign Key that will help to identify that manager works in which hall
5. Schedule	Schedule of the Hall
a. Shift_ID	Primary Key uniquely identifies the shift
a. Shift_time	Timing of the Shift
b. Shift_type	Type of Shift
6. Payroll	It contains payment information of employees
a. Pay_ID	Primary Key uniquely identifies the payment id
b. Emp_ID	Foreign Key, primary key for the employee

	payment
c. Hall_ID	Foreign Key, primary key for the employee payment
d. Amount	Weekly payment amount of employees working
7. Employee Hall	It contains information on which hall does each employee work
a. EH_ID	Primary Key
b. Emp_ID	Foreign Key identifies the employee
c. Hall_ID	Foreign Key identifies the hall
d. EName	Name of Employee
e. HName	Name of Dining Hall
8. Employee Schedule	It contains information of schedule of employees
a. ES_ID	Primary Key uniquely identifies employee schedule
b. Emp_ID	Foreign Key identifies the employee
c. Shift_ID	Foreign Key identifies the shift
d. Shift_Day	Day of Shift for instance Monday or Friday
e. Dropped	It gives information whether the shift of employee is dropped
f. Picked	It gives information whether the dropped shift of employee is picked
9. Drop Shift	Shift Dropped by Employees
a. Drop_ID	Primary Key uniquely identifies which shift is dropped
b. ES_ID	Foreign Key identifies the scheduled shift of

	employee
c. Reason	Reason for dropping the shift
10. Picked Shift	Shifts that are picked or available
a. Pick_ID	Primary Key uniquely identifies which shift are picked or available
b. Emp_ID	Foreign Key identifies employees who picks the shift
c. ES_ID	Foreign Key identifies the schedule of employee who picked the shift
11. Feedback	Feedback given by employee
a. Feedback_ID	Primary Key uniquely identifies feedback
b. Emp_ID	Foreign Key uniquely identifies employee
c. Description	Description of feedback

Vision Entity-Relationship Diagram



Business Rule:

- Every user login must be either manager, commercial or student employee
- An employee should be a commercial or student employee who is affiliated with university.
- An employee must have at-least two shifts in a dining hall.
- A student employee can work no more than 20 hours.
- Mode of payroll with the student could be only direct deposit or cheque.
- An employee can drop a shift two weeks prior to dropped shift date
- Manager will approve the drop shift request before it is available for pick-up
- An employee can write feedback about its own workplace.

Database infrastructure:

The database infrastructure is based on a client-server model. The SQL-server is used as the database engine and the MS Access is used as the interface design tool. Data is inserted, updated, deleted and queried from the SQL-server database with the help of forms on access. Useful data stored on SQL database can also be viewed with the help of reports that are generated through MS Access.

SQL Script for Creating and Inserting Sample Data:

CREATE: Account

```
Create Table Account
(
Acct_ID Char(4) Not Null,
Acct_Password Char(4) Not Null,
FName VarChar(10) Not Null,
LName VarChar(10) Not Null,
Email_ID VarChar(20) Not NUll,
User_Address VarChar(20) Not Null,
User_Contact Char(10) Not Null,
Account_Type Char(8) Not Null Check(Account_Type IN('Employee','Manager')),
Constraint Account_PK Primary Key (Acct_ID),
Constraint Account_User_Contact check (User_Contact like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
);
```

CREATE: Hall

```
Insert Into Hall Values ('H001','Graham','Mt. Olympus Drive')
Insert Into Hall Values ('H002','Saddler','100 Irving Ave')
Insert Into Hall Values ('H003','Shaw','201 Euclid Ave')
Insert Into Hall Values ('H004','Ernie','619 Comstock Avenue')
Insert Into Hall Values ('H005','Brockway','401 Van Buren St.')
```

CREATE: Employee1

```
Create Table Employee1sni
(
Emp_ID Char(4) Not Null,
Acct_ID Char(4) Not NUll,
Constraint Employee1_PK Primary Key(Emp_ID),
Constraint Employee1_FK Foreign Key (Acct_ID) References Account(Acct_ID)
);
```

CREATE: Employee1Hall

```
Create Table Employee1Hall
(
EH_ID char(4) Not Null,
Emp_ID char(4) Not Null,
Hall_ID char(4) Not Null,
EName Varchar(10) Not Null,
HName Varchar(10) Not Null,
Constraint Employee1Hall_PK Primary Key (EH_ID),
Constraint Employee1Hall_FK Foreign Key (Emp_ID) References Employee1(Emp_ID),
Constraint Employee1Hall_FK1 Foreign Key (Hall_ID) References Hall(Hall_ID)
);
```

CREATE: Manager

```
Create Table Manager
(
Manager_ID Char(4) Not Null,
Acct_ID Char(4) Not NUll,
Hall_ID char(4) Not Null,
Constraint Manager_PK Primary Key(Manager_ID),
Constraint Manager_FK Foreign Key (Acct_ID) References Account(Acct_ID),
Constraint Manager_FK Foreign Key (Hall_ID) References Hall(Hall_ID)
);
```

CREATE: Payroll

```
Create Table Payroll
(
Pay_ID Char(4) Not Null,
Emp_ID Char(4) Not Null,
Hall_ID Char(4) Not NUll,
Amount int Not Null,
Constraint Payroll_PK Primary Key(Pay_ID),
Constraint Payroll_FK Foreign Key(Emp_ID) References Employee1(Emp_ID),
Constraint Payroll_FK Foreign Key(Hall_ID) References Hall(Hall_ID)
);
```

CREATE: Schedule

```
Create Table Schedule
(
Shift_ID char(4) Not Null,
Shift_Time Varchar(10) Not Null,
Shift_Type Varchar(10) Not Null,
Shift_Hours int Not Null,
Constraint Schedule_PK Primary Key(Shift_ID)
);
```

CREATE: Employee1Schedule

```
Create Table Employee1Schedule
(
ES_ID char(4) Not Null,
Emp_ID Char(4) Not Null,
Shift_ID Char(4) Not Null,
Shift_Day VarChar(10) Not Null,
Dropped Varchar(10),
Picked varchar(10),
Constraint Employee1Schedule_PK Primary Key(ES_ID),
Constraint EmployeeSchedule_FK Foreign Key (Emp_ID) References Employee1(Emp_ID),
Constraint EmployeeSchedule_FK1 Foreign Key (Shift_ID) References Schedule(Shift_ID)
);
```

CREATE: DropShift

```
Create Table DropShift
(
Drop_ID char(4) Not Null,
Shift_ID char(4) Not Null,
Emp_ID char(4) Not Null,
Reason Varchar(20) Not Null,
Constraint DropShift_PK Primary Key(Drop_ID),
Constraint DropShift_FK Foreign Key (ES_ID) References Employee1Schedule(ES_ID)
);
```

CREATE: PickedShift

```
Create Table PickedShift
(
Picked_ID char(4) Not Null,
Shift_ID char(4) Not Null,
Emp_ID char(4) Not Null,
Constraint PickedShift_PK Primary Key(Picked_ID),
Constraint PickedShift_FK Foreign Key (Emp_ID) References Employee1(Emp_ID),
Constraint PickedShift_FK1 Foreign Key (ES_ID) References Employee1Schedule(ES_ID)
);
```

CREATE: Feedback

```
Create Table Feedback
(
Feedback_ID char(4) Not Null,
Emp_ID char(4) Not Null,
Descpt Varchar(50)
Constraint FeedBack_PK Primary Key (Feedback_ID),
Constraint Feedback_FK Foreign Key (Emp_ID) References Employee1(Emp_ID)
);
```

Insert Data in Account Table

```
Insert Into Account Values ('C111','1234','Rahul','Rathod','rnrathod@syr.edu','1011 EAS', '3156799859', 'Employee')
Insert Into Account Values ('C222','5678','Ellon','Musk','ellonmusk@syr.edu','2022 EAS', '0123456789', 'Manager')
Insert Into Account Values ('C333','9999','Kevin','Johns','kevinjohnsd@syr.edu','3033 EAS', '9876543210', 'Employee')
Insert Into Account Values ('C444','4444','Chris','Hawk','chawk@syr.edu','120 EAS', '3155669859', 'Employee')
Insert Into Account Values ('C555','1234','Martin','Mello','mmello@syr.edu','111 EAS', '3156799679', 'Employee')
Insert Into Account Values ('C666','8000','Hans','Charles','hcharles@syr.edu','101 EAS', '8796799859', 'Manager')
Insert Into Account Values ('C777','7077','James','Wall','james@syr.edu','999 EAS', '7956796859', 'Employee')
Insert Into Account Values ('C888','8888','Jenny','Wall', 'jwall@syr.edu','879 EAS', '3256799234', 'Manager')
Insert Into Account Values ('C000','1234','Irene','Roshan','rirene@syr.edu','731 EAS', '679812345', 'Employee')
Insert Into Account Values ('C999','9999','Raj','Nisar','rajnisar@syr.edu','312 EAS', '3157654321', 'Employee')
```

SELECT * FROM Account

	Acct_ID	Acct_Password	FName	LName	Email_ID	User_Address	User_Contact	Account_Type
1	C000	1234	Irene	Roshan	rirene@syr.edu	731 EAS	679812345	Employee
2	C110	1111	Kathy	Hellinger	khellinger@syr.edu	123 EAS	3145687489	Manager
3	C111	1234	Rahul	Rathod	mrathod@syr.edu	1011 EAS	3156799859	Employee
4	C220	2200	Stephen	Brandt	brandt@syr.edu	345 EAS	3145867489	Manager
5	C222	5678	Ellon	Musk	ellonmusk@syr.edu	2022 EAS	0123456789	Manager
6	C333	9999	Kevin	Johns	kevinjohnsd@syr.edu	3033 EAS	9876543210	Employee
7	C444	4444	Chris	Hawk	chawk@syr.edu	120 EAS	3155669859	Employee
8	C555	1234	Martin	Mello	mmello@syr.edu	111 EAS	3156799679	Employee
9	C666	8000	Hans	Charles	hcharles@syr.edu	101 EAS	8796799859	Manager
10	C777	7077	James	Wall	james@syr.edu	999 EAS	7956796859	Employee
11	C888	8888	Jenny	Wall	jwall@syr.edu	879 EAS	3256799234	Manager
12	C999	9999	Raj	Nisar	rajnisar@syr.edu	312 EAS	3157654321	Employee

Insert Data in Hall Table

```
Insert Into Hall Values ('H001','Graham','Mt. Olympus Drive')
Insert Into Hall Values ('H002','Saddler','100 Irving Ave')
Insert Into Hall Values ('H003','Shaw','201 Euclid Ave')
Insert Into Hall Values ('H004','Ernie','619 Comstock Avenue')
Insert Into Hall Values ('H005','Brockway','401 Van Buren St.')
```

SELECT * FROM Hall

	Hall_ID	HName	HLocation
1	H001	Graham	Mt. Olympus Drive
2	H002	Saddler	100 Irving Ave
3	H003	Shaw	201 Euclid Ave
4	H004	Ernie	619 Comstock Avenue
5	H005	Brockway	401 Van Buren St.

Automatic Insert Data in Employee1 Table due to trigger

```
SELECT * FROM Employee1
```

	Emp_ID	Acct_ID
1	E001	C000
2	E002	C111
3	E003	C333
4	E004	C444
5	E005	C555
6	E006	C777
7	E007	C999

Insert Data in Employee1Hall Table

```
Insert Into Employee1Hall Values ('EH01','E001','H005','Irene Roshan','Brockway')
Insert Into Employee1Hall Values ('EH02','E002','H004','Rahul Rathod','Erine')
Insert Into Employee1Hall Values ('EH03','E003','H005','Kevin Johns','Brockway')
Insert Into Employee1Hall Values ('EH04','E004','H003','Chris Hawk','Shaw')
Insert Into Employee1Hall Values ('EH05','E004','H001','Chirs Hawk','Graham')
Insert Into Employee1Hall Values ('EH06','E005','H004','Martin Mella','Saddler')
Insert Into Employee1Hall Values ('EH07','E006','H002','James Wall','Graham')
Insert Into Employee1Hall Values ('EH08','E007','H004','Raj Nisar','Erine')
Insert Into Employee1Hall Values ('EH09','E001','H005','Irene Roshan','Graham')
Insert Into Employee1Hall Values ('EH10','E003','H004','Kevin Johns','Erine')
Insert Into Employee1Hall Values ('EH11','E006','H002','James Wall','Saddler')
```

```
SELECT * FROM Employee1Hall
```

	EH_ID	Emp_ID	Hall_ID	EName	HName
1	EH01	E001	H005	Irene Roshan	Brockway
2	EH02	E002	H004	Rahul Rathod	Emie
3	EH03	E003	H005	Kevin Johns	Brockway
4	EH04	E004	H003	Chris Hawk	Shaw
5	EH05	E004	H001	Chris Hawk	Graham
6	EH06	E005	H002	Martin Mello	Saddler
7	EH07	E006	H001	James Wall	Graham
8	EH08	E007	H004	Raj Nisar	Emie
9	EH09	E001	H005	Irene Roshan	Graham
10	EH10	E003	H004	Kevin Johns	Emie
11	EH11	E006	H002	James Wall	Saddler

Automatic Insert Data in Manager Table due to trigger

SELECT * FROM Manager

	Manager_ID	Acct_ID	Hall_ID
1	M001	C222	H005
2	M002	C666	H002
3	M003	C888	H003
4	M004	C110	H001
5	M005	C220	H004

Insert Data in Payroll Table

```
Insert Into Payroll Values('P001', 'E001','H005','70')
Insert Into Payroll Values('P002','E002','H004','195')
Insert Into Payroll Values('P003','E003','H005','0')
Insert Into Payroll Values('P004','E004','H003','100')
Insert Into Payroll Values('P005','E004','H001','100')
Insert Into Payroll Values('P006','E005','H002','65')
Insert Into Payroll Values('P007','E006','H001','85')
Insert Into Payroll Values('P008','E001','H005','40')
Insert Into Payroll Values('P009','E003','H004','90')
Insert Into Payroll Values('P010','E006','H002','120')
```

SELECT * FROM Payroll

	Pay_ID	Emp_ID	Hall_ID	Amount
1	P001	E001	H005	70
2	P002	E002	H004	195
3	P003	E003	H005	0
4	P004	E004	H003	100
5	P005	E004	H001	100
6	P006	E005	H002	65
7	P007	E006	H001	85
8	P008	E001	H005	40
9	P009	E003	H004	90
10	P010	E006	H002	120

Insert Data in Schedule Table

```
Insert Into Schedule Values ('S001','6:00am -10:00am','Checker','4H')
Insert Into Schedule Values ('S002','6:00am -10:00am','Utilities','4H')
Insert Into Schedule Values ('S003','6:00am -10:00am','Cooking','4H')
Insert Into Schedule Values ('S004','6:00am -10:00am','Dishroom','4H')
Insert Into Schedule Values ('S005','6:00am -10:00am','HotLine','4H')
Insert Into Schedule Values ('S006','10:00am -2:00pm','Checker','4H')
Insert Into Schedule Values ('S007','10:00am -2:00pm','Utilities','4H')
Insert Into Schedule Values ('S008','10:00am -2:00pm','Cooking','4H')
Insert Into Schedule Values ('S009','10:00am -2:00pm','Dishroom','4H')
Insert Into Schedule Values ('S010','10:00am -2:00pm','HotLine','4H')
Insert Into Schedule Values ('S011','2:00pm -6:00pm','Checker','4H')
Insert Into Schedule Values ('S012','2:00pm -6:00pm','Utilities','4H')
Insert Into Schedule Values ('S013','2:00pm -6:00pm','Cooking','4H')
Insert Into Schedule Values ('S014','2:00pm -6:00pm','Dishroom','4H')
Insert Into Schedule Values ('S015','2:00pm -6:00pm','HotLine','4H')
Insert Into Schedule Values ('S016','6:00pm -10:00pm','Checker','4H')
Insert Into Schedule Values ('S017','6:00pm -10:00pm','Utilities','4H')
Insert Into Schedule Values ('S018','6:00pm -10:00pm','Cooking','4H')
Insert Into Schedule Values ('S019','6:00pm -10:00pm','Dishroom','4H')
Insert Into Schedule Values ('S020','6:00pm -10:00pm','HotLine','4H')
```

SELECT * FROM Schedule

	Shift_ID	Shift_Time	Shift_Type	Shift_Hours
1	S001	6:00am -10:00am	Checker	4
2	S002	6:00am -10:00am	Utilities	4
3	S003	6:00am -10:00am	Cooking	4
4	S004	6:00am -10:00am	Dishroom	4
5	S005	6:00am -10:00am	HotLine	4
6	S006	10:00am -2:00pm	Checker	4
7	S007	10:00am -2:00pm	Utilities	4
8	S008	10:00am -2:00pm	Cooking	4
9	S009	10:00am -2:00pm	Dishroom	4
10	S010	10:00am -2:00pm	HotLine	4
11	S011	2:00pm -6:00pm	Checker	4
12	S012	2:00pm -6:00pm	Utilities	4
13	S013	2:00pm -6:00pm	Cooking	4
14	S014	2:00pm -6:00pm	Dishroom	4
15	S015	2:00pm -6:00pm	HotLine	4
16	S016	6:00pm -10:00pm	Checker	4
17	S017	6:00pm -10:00pm	Utilities	4
18	S018	6:00pm -10:00pm	Cooking	4
19	S019	6:00pm -10:00pm	Dishroom	4
20	S020	6:00pm -10:00pm	HotLine	4

Insert Data in Employee1Schedule Table

```
Insert Into Employee1Schedule Values ('ES01','E001','S004','Monday')
Insert Into Employee1Schedule Values ('ES02','E001','S005','Tuesday')
Insert Into Employee1Schedule Values ('ES03','E001','S013','Thursday')
Insert Into Employee1Schedule Values ('ES05','E002','S004','Wednesday')
Insert Into Employee1Schedule Values ('ES06','E002','S005','Friday')
Insert Into Employee1Schedule Values ('ES07','E002','S013','Thursday')
Insert Into Employee1Schedule Values ('ES08','E003','S008','Wednesday')
Insert Into Employee1Schedule Values ('ES09','E003','S003','Monday')
Insert Into Employee1Schedule Values ('ES10','E003','S011','Thursday')
Insert Into Employee1Schedule Values ('ES11','E004','S006','Monday')
Insert Into Employee1Schedule Values ('ES12','E004','S010','Sunday')
Insert Into Employee1Schedule Values ('ES13','E004','S012','Saturday')
Insert Into Employee1Schedule Values ('ES14','E005','S003','Wednesday')
Insert Into Employee1Schedule Values ('ES15','E005','S006','Sunday')
Insert Into Employee1Schedule Values ('ES16','E006','S005','Thursday')
Insert Into Employee1Schedule Values ('ES17','E006','S010','Saturday')
Insert Into Employee1Schedule Values ('ES18','E006','S013','Friday')
Insert Into Employee1Schedule Values ('ES19','E007','S004','Saturday')
```

```
SELECT * FROM Employee1Schedule
```

	ES_ID	Emp_ID	Shift_ID	Shift_Day	Dropped	Picked
1	ES01	E001	S004	Monday	Yes	Yes
2	ES02	E001	S005	Tuesday	NULL	NULL
3	ES03	E001	S013	Thursday	NULL	NULL
4	ES05	E002	S004	Wednesday	NULL	NULL
5	ES06	E002	S005	Friday	NULL	NULL
6	ES07	E002	S013	Thursday	NULL	NULL
7	ES08	E003	S008	Wednesday	NULL	NULL
8	ES09	E003	S003	Monday	NULL	NULL
9	ES10	E003	S011	Thursday	NULL	NULL
10	ES11	E004	S006	Monday	NULL	NULL
11	ES12	E004	S010	Sunday	NULL	NULL
12	ES13	E004	S012	Saturday	NULL	NULL
13	ES14	E005	S003	Wednesday	NULL	NULL
14	ES15	E005	S006	Sunday	NULL	NULL
15	ES16	E006	S005	Thursday	NULL	NULL
16	ES17	E006	S010	Saturday	NULL	NULL
17	ES18	E006	S013	Friday	NULL	NULL
18	ES19	E007	S004	Saturday	NULL	NULL

Insert Data in DropShift Table

```
Insert Into DropShift Values('D001','ES06','Sick','Yes');
Insert Into DropShift Values('D002','ES03','Project Meeting','Yes')
Insert Into DropShift(Drop_ID,ES_ID,Reason) Values('D003','ES12','Change Schedule')
Insert Into DropShift Values('D004','ES01','Vacation','No');
Insert Into DropShift(Drop_ID,ES_ID,Reason) Values('D005','ES02','Sick')
Insert Into DropShift Values('D006','ES13','Studies','Yes');
Insert Into DropShift Values('D007','ES09','Project Meeting','Yes');
```

SELECT * FROM DropShift

	Drop_ID	ES_ID	Reason	Approval
1	D001	ES06	Sick	Yes
2	D002	ES03	Project Meeting	Yes
3	D003	ES12	Change Schedule	NULL
4	D004	ES01	Vacation	No
5	D005	ES02	Sick	NULL
6	D006	ES13	Studies	Yes
7	D007	ES09	Project Meeting	Yes
8	D008	ES01	Sick	Yes

Insert Data in PickedShift Table

```
Insert Into PickedShift values('P001','E002','ES01')
Insert Into PickedShift values('P002','E003','ES11')
Insert Into PickedShift values('P003','E002','ES13')
Insert Into PickedShift values('P004','E001','ES09')
Insert Into PickedShift values('P005','E007','ES02')
```

SELECT * FROM PickedShift

	Picked_ID	Emp_ID	ES_ID
1	P001	E002	ES01
2	P002	E003	ES11
3	P003	E002	ES13
4	P004	E001	ES09
5	P005	E007	ES02

Insert Data in Feedback Table

```
Insert Into FeedBack Values ('F001','E002','Great Working Enivornment')
Insert Into FeedBack Values ('F002','E003','Lot of Work')
Insert Into FeedBack Values ('F003','E004','Flexible Shift Scheduling')
Insert Into FeedBack Values ('F004','E005','Best Manager')
Insert Into FeedBack Values ('F005','E001','Strict rules')
Insert Into FeedBack Values ('F006','E006','Great Place and Good Food')
```

SELECT * FROM Feedback

	Feedback_ID	Emp_ID	Descpt
1	F001	E002	Great Working Enivornment
2	F002	E003	Lot of Work
3	F003	E004	Flexible Shift Scheduling
4	F004	E005	Best Manager
5	F005	E001	Strict rules
6	F006	E006	Great Place and Good Food

Major Data Questions

Since the current system for employee scheduling is a manual system, a database application solution will turn to a completely online system to manage employee scheduling efficiently.

The users of my database application can be segregated into three segments:

- Student and Commercial employees
- Managers

Following list shows segregation of what data questions each role needs answered by the proposed system and how we achieve it.

Why SU Food Service Employees query the database

- SU employees can query the database to view their schedule of the week. The SU users of my database application can view all the shifts he/she going to work in the week. Users can view all their schedule that were dropped but not picked by other employee or disapproved by the administrator. This section will enlist only the schedule of the user who has login to the system successfully. Below is the SQL query to demonstrate this function. Let's assume the Emp_ID to be E002.

```
Select  s.Shift_ID,es.Emp_ID, es.Shift_Day, s.Shift_Time, s.Shift_Type,s.Shift_Hours
From Employee1Schedule es
Inner Join Schedule s ON s.Shift_ID = es.Shift_ID
Where es.Emp_ID='E002' AND es.Picked IS Null;
```

Shift_ID	Emp_ID	Shift_Day	Shift_Time	Shift_Type	Shift_Hours
S004	E002	Wednesday	6:00am -10:00am	Dishroom	4
S005	E002	Friday	6:00am -10:00am	HotLine	4
S013	E002	Thursday	2:00pm -6:00pm	Cooking	4

- SU Employees can query the database to view the shifts that are dropped by other employees. User of my database can view all the shifts that are available to be picked and approved by the manager. This section will enlist only the schedule of the user who has login to the system successfully. Below is the SQL query to demonstrate this function.

```
Select d.Drop_ID , es.ES_ID , s.Shift_ID, s.Shift_Time, s.Shift_Type,s.Shift_Hours, es.Shift_Day
From Employee1Schedule es
Inner Join Schedule s On s.Shift_ID = es.Shift_ID
Inner Join DropShift d ON d.ES_ID = es.ES_ID
Where Approval='Yes'
```

Drop_ID	ES_ID	Shift_ID	Shift_Time	Shift_Type	Shift_Hours	Shift_Day
D001	ES06	S005	6:00am -10:00am	HotLine	4	Friday
D002	ES03	S013	2:00pm -6:00pm	Cooking	4	Thursday
D006	ES13	S012	2:00pm -6:00pm	Utilities	4	Saturday
D007	ES09	S003	6:00am -10:00am	Cooking	4	Monday
D008	ES01	S004	6:00am -10:00am	Dishroom	4	Monday

- SU Employees can query the database to view the pay amount of the week. User of my database can view all the pay amount he/she is going to receive from different halls he/she is working. This section will enlist only the schedule of the user who has login to the system successfully. Below is the SQL query to demonstrate this function. Let's assume the Emp_ID to be E001.

```
Select Pay_ID, Emp_ID , Hall_ID ,Amount
From Payroll
where Emp_ID='E001'
```

Pay_ID	Emp_ID	Hall_ID	Amount
P001	E001	H005	70
P008	E001	H005	40

Why SU Food Service Managers query database

- Manager can query database to process and manage the dropped shifts. Manager can monitor the shifts that are been dropped and analyze the authenticity of the reason for dropping the shift and eventually can make decisions to either accept or reject the request to drop shift.

```
Select d.Drop_ID , es.ES_ID , s.Shift_ID, s.Shift_Time, s.Shift_Type,s.Shift_Hours, es.Shift_Day ,d.Reason,d.Approval
From Employee1Schedule es
Inner Join Schedule s On s.Shift_ID = es.Shift_ID
Inner Join DropShift d ON d.ES_ID = es.ES_ID
```

Drop_ID	ES_ID	Shift_ID	Shift_Time	Shift_Type	Shift_Hours	Shift_Day	Reason	Approval
D001	ES06	S005	6:00am -10:00am	HotLine	4	Friday	Sick	Yes
D002	ES03	S013	2:00pm -6:00pm	Cooking	4	Thursday	Project Meeting	Yes
D003	ES12	S010	10:00am -2:00pm	HotLine	4	Sunday	Change Schedule	NULL
D004	ES01	S004	6:00am -10:00am	Dishroom	4	Monday	Vacation	No
D005	ES02	S005	6:00am -10:00am	HotLine	4	Tuesday	Sick	NULL
D006	ES13	S012	2:00pm -6:00pm	Utilities	4	Saturday	Studies	Yes
D007	ES09	S003	6:00am -10:00am	Cooking	4	Monday	Project Meeting	Yes
D008	ES01	S004	6:00am -10:00am	Dishroom	4	Monday	Sick	Yes

Manager can query the shifts which are yet to be approved.

Before making decision

Drop_ID	ES_ID	Shift_ID	Shift_Time	Shift_Type	Shift_Hours	Shift_Day	Reason	Approval
D003	ES12	S010	10:00am -2:00pm	HotLine	4	Sunday	Change Schedule	NULL
D005	ES02	S005	6:00am -10:00am	HotLine	4	Tuesday	Sick	NULL

After giving decision

Drop_ID	ES_ID	Shift_ID	Shift_Time	Shift_Type	Shift_Hours	Shift_Day	Reason	Approval
D003	ES12	S010	10:00am -2:00pm	HotLine	4	Sunday	Change Schedule	No
D005	ES02	S005	6:00am -10:00am	HotLine	4	Tuesday	Sick	Yes

- Manager can query database to view the schedule of the hall. User of my database can view all the different employees that will be working on that day in the hall. Below is the SQL query to demonstrate this function. Let's assume the Hall_ID to be H005.

```
Select eh.Hall_ID ,eh.HName, es.Emp_ID , eh.EName , s.Shift_ID , es.Shift_Day , s.Shift_Time , s.Shift_Type, s.Shift_Hours
From Employee1Schedule es
Inner Join Schedule s On s.Shift_ID = es.Shift_ID
Inner Join Employee1Hall eh ON eh.Emp_ID = es.Emp_ID
Where eh.Hall_ID ='H005';
```

Hall_ID	HName	Emp_ID	EName	Shift_ID	Shift_Day	Shift_Time	Shift_Type	Shift_Hours
H005	Brockway	E001	Irene Roshan	S004	Monday	6:00am -10:00am	Dishroom	4
H005	Graham	E001	Irene Roshan	S004	Monday	6:00am -10:00am	Dishroom	4
H005	Brockway	E001	Irene Roshan	S005	Tuesday	6:00am -10:00am	HotLine	4
H005	Graham	E001	Irene Roshan	S005	Tuesday	6:00am -10:00am	HotLine	4
H005	Brockway	E001	Irene Roshan	S013	Thursday	2:00pm -6:00pm	Cooking	4
H005	Graham	E001	Irene Roshan	S013	Thursday	2:00pm -6:00pm	Cooking	4
H005	Brockway	E003	Kevin Johns	S008	Wednesday	10:00am -2:00pm	Cooking	4
H005	Brockway	E003	Kevin Johns	S003	Monday	6:00am -10:00am	Cooking	4
H005	Brockway	E003	Kevin Johns	S011	Thursday	2:00pm -6:00pm	Checker	4

- Manager can search for employee information .Admin has the privilege to search for user and payment records for various purposes. Admin has to enter the Emp_ID for retrieving the specific user. Below is the SQL query to demonstrate this function.

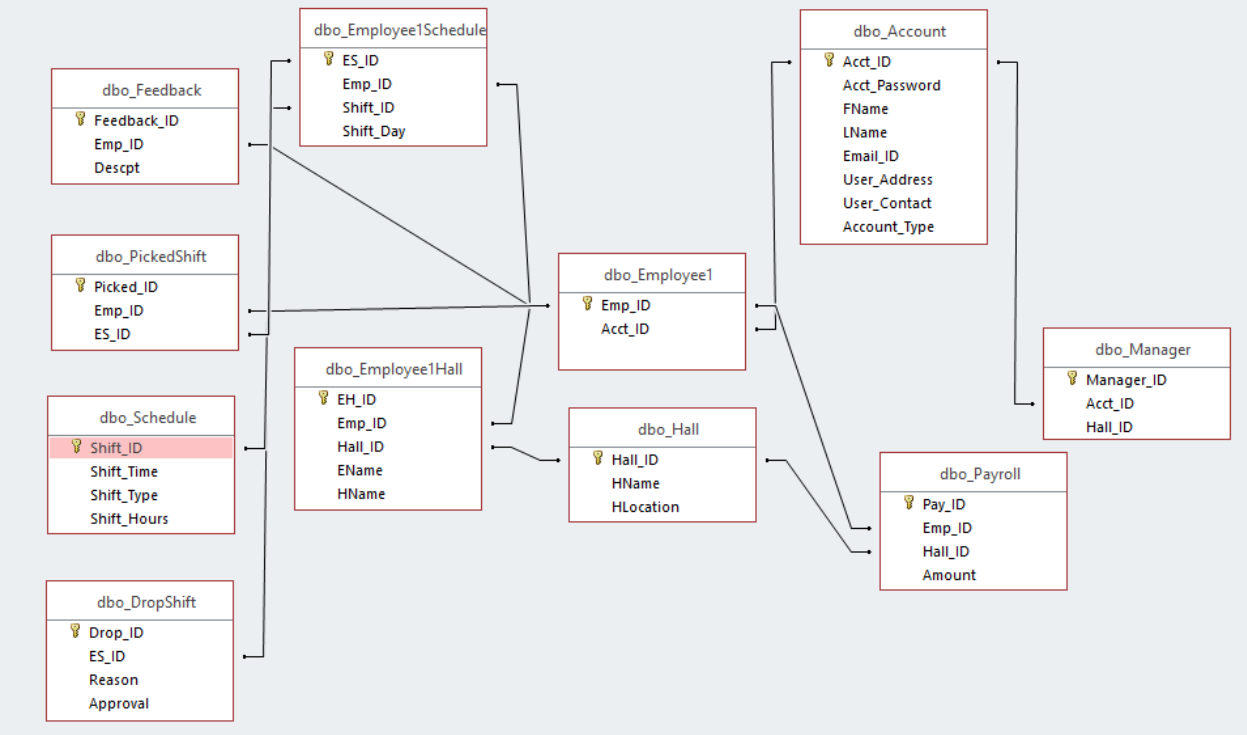
```
Select a.FName ,a.LName , a.Email_ID ,a.User_Address , a.User_Contact, p.Hall_ID , p.Amount
From Account a
Inner Join Employee1 e On e.Acct_ID = a.Acct_ID
Inner Join Payroll p On p.Emp_ID = e.Emp_ID
```

FName	LName	Email_ID	User_Address	User_Contact	Hall_ID	Amount
Irene	Roshan	rirene@syr.edu	731 EAS	679812345	H005	70
Rahul	Rathod	rrathod@syr.edu	1011 EAS	3156799859	H004	195
Kevin	Johns	kevinjohnsd@syr.edu	3033 EAS	9876543210	H005	0
Chris	Hawk	chawk@syr.edu	120 EAS	3155669859	H003	100
Chris	Hawk	chawk@syr.edu	120 EAS	3155669859	H001	100
Martin	Mello	mmello@syr.edu	111 EAS	3156799679	H002	65
James	Wall	james@syr.edu	999 EAS	7956796859	H001	85
Irene	Roshan	rirene@syr.edu	731 EAS	679812345	H005	40
Kevin	Johns	kevinjohnsd@syr.edu	3033 EAS	9876543210	H004	90
James	Wall	james@syr.edu	999 EAS	7956796859	H002	120

Manager can query data for specific user. Let's assume Emp_ID is E006

Emp_ID	FName	LName	Email_ID	User_Address	User_Contact	Hall_ID	Amount
E006	James	Wall	james@syr.edu	999 EAS	7956796859	H001	85
E006	James	Wall	james@syr.edu	999 EAS	7956796859	H002	120

MS-Access Relationship Diagram



Forms

SU Food Service Database Application has two types of users:

1. Employee
2. Manager

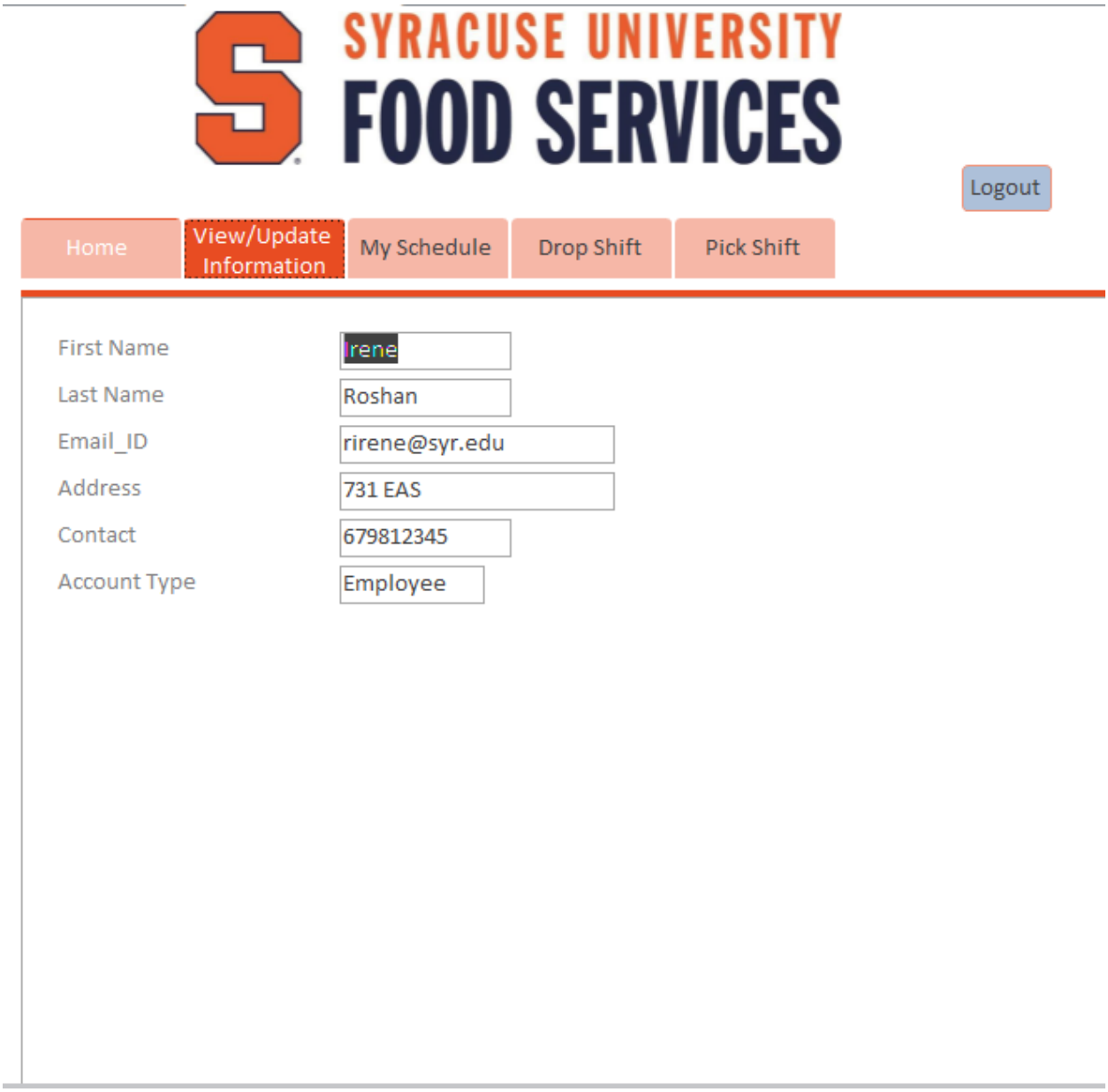
Master Login Form

The below form is a login form for students and commercial employees and managers and it provides an interface for users to plan their schedule. The above master login page enables users to choose their type of login.

User Panel of Employee

View/Update User Information

The below form allows users to view or update their personal information, however, the user does not have the rights to change their User ID and the type of User. Users can change their email and contact numbers to update their information.

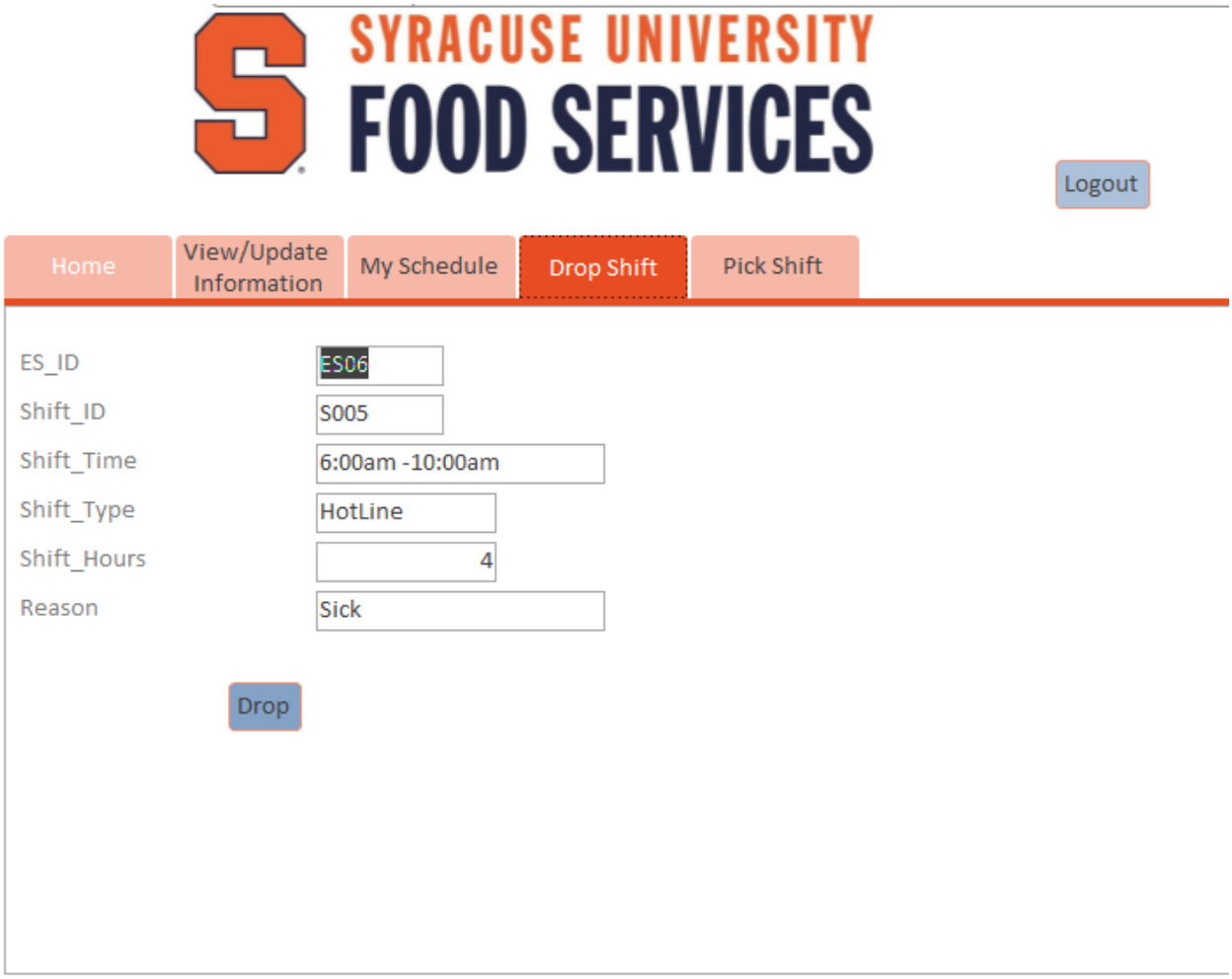


My Schedule

The 'My Schedule' will enlist the updated schedule of the user. The above form give details about the Shift ID, Day of Shift, Shift Timing, Shift Type and Hours Shift.

Drop Shift

The below form enable the user to drop the shift. In order to drop the shift the user has to fill the details of the shift he/she wants to drop. ES_ID, Shift_ID, Shift_Time, Shift_Type, Shift_Hours, Reason. In this form all the fields are mandatory.

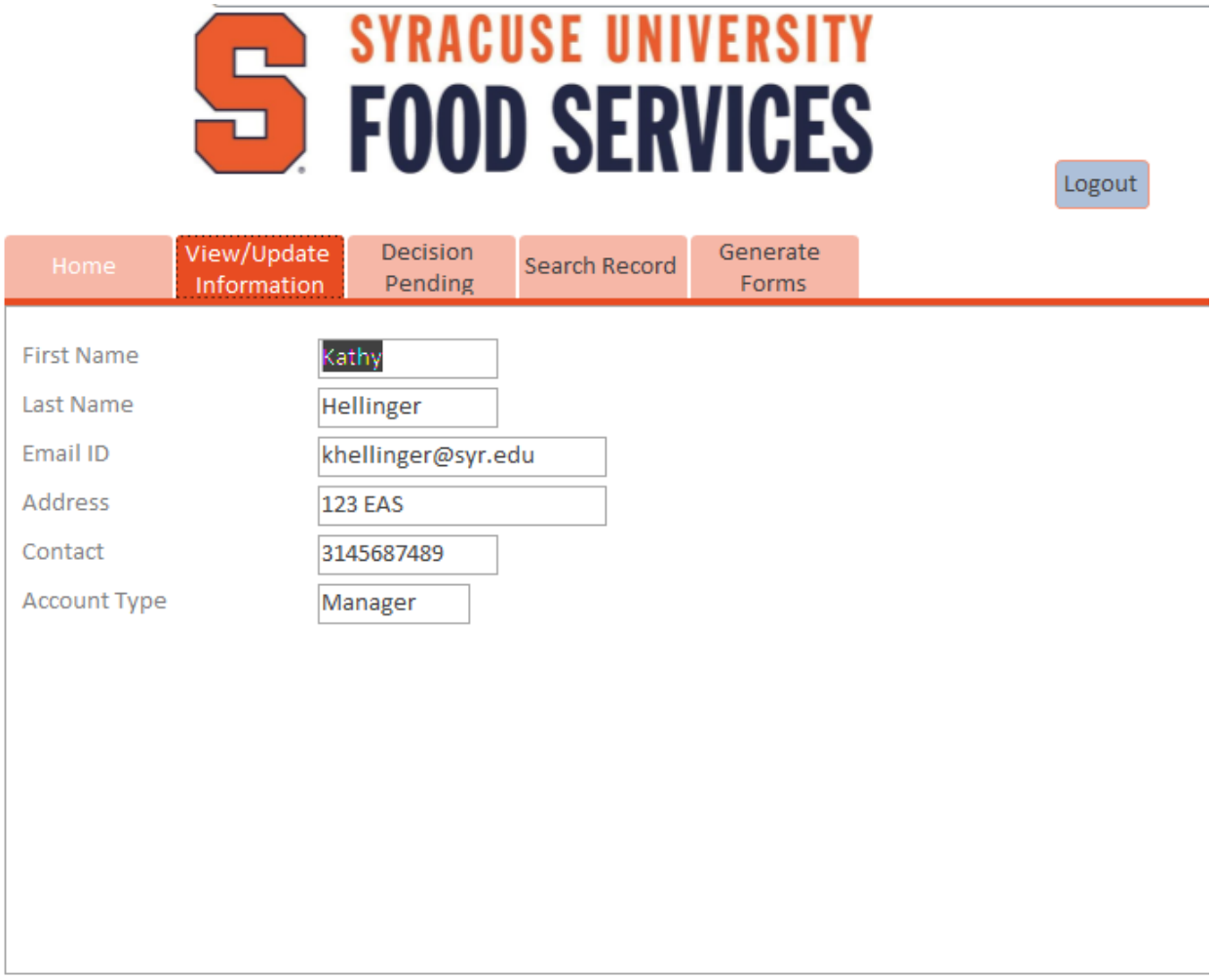


Pick Shift

The below form enable the user to pick any shift he/she wants if the shift is available. In order to pick a shift the user will have to fill the ES_ID and Shift_ID. If the shift is available it will be added to the PickedShift table and if the shift is not available it will shows an error message 'No Data Found'.

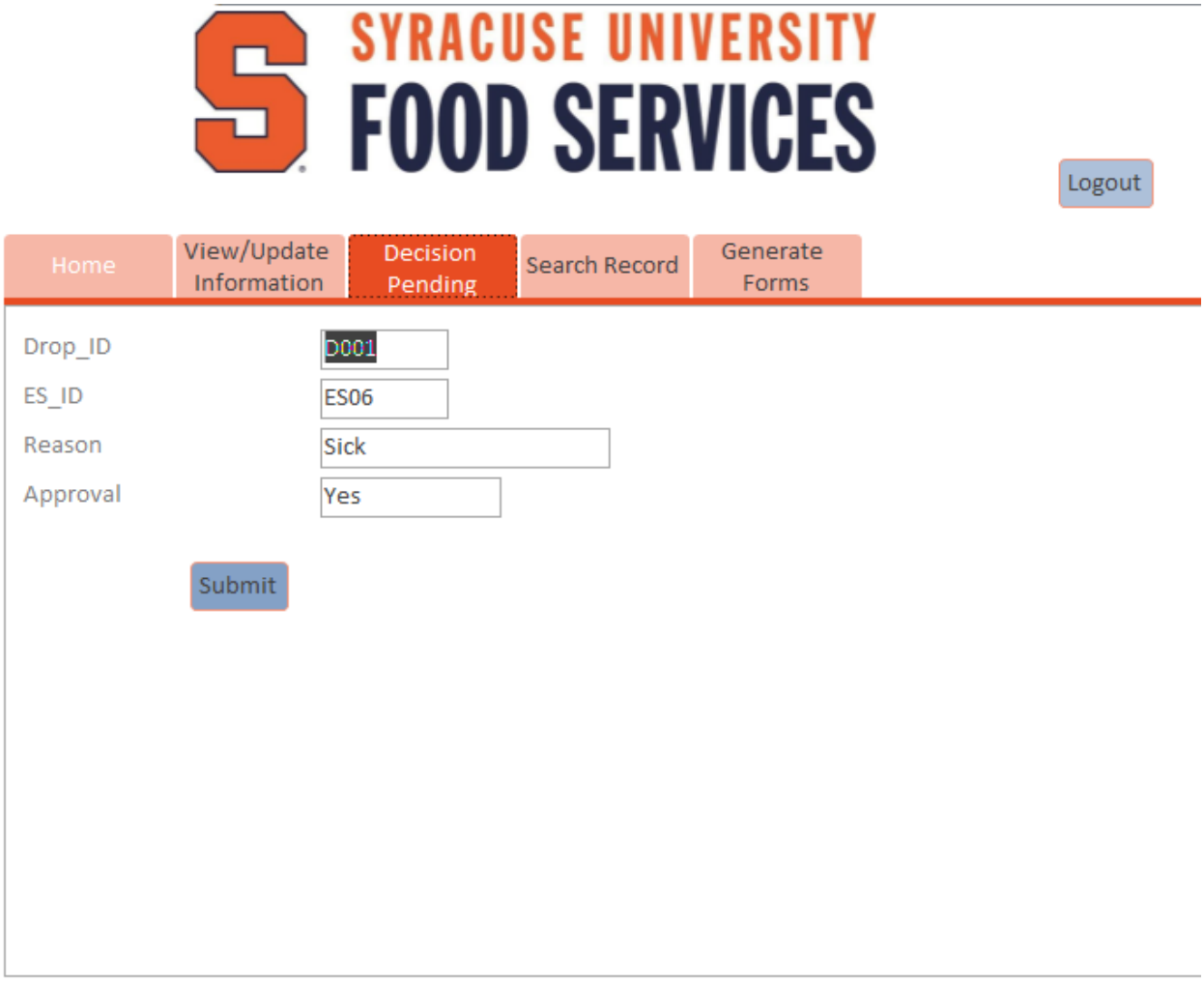
User Panel of Manager

The below form allows manager to view or update their personal information, however, the admin does not have the rights to change their Acct_ ID. Manager can change their email and contact numbers to update their information.



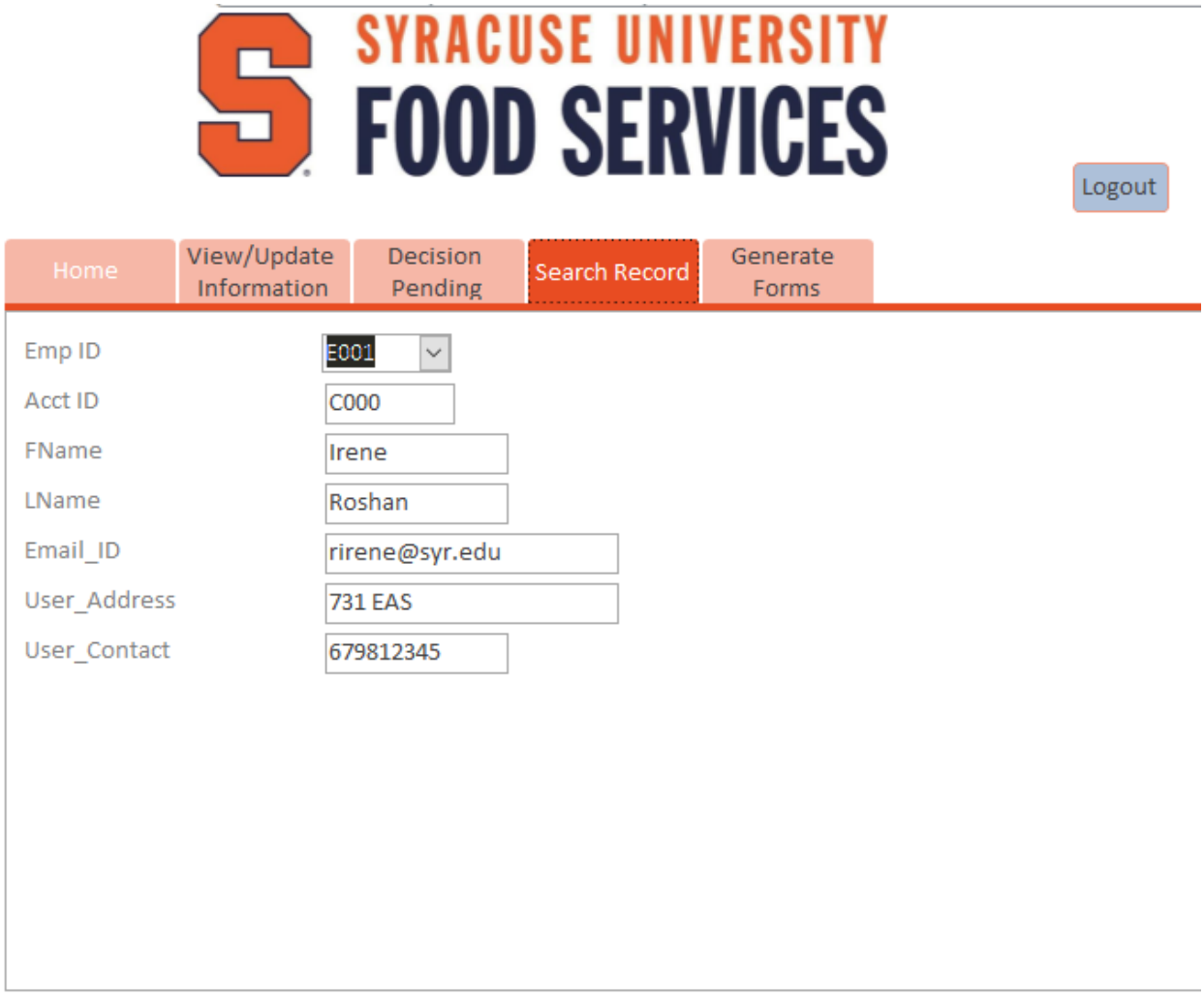
Decision Pending

The 'decisions pending' tab will enlist all the dropped shifts that the manager has to process. All the shifts that user has requested but still has not been either approved or disapproved by the manager will be enlisted under this tab.



Search Record

This form will enable the manager to search for any specific user by their unique Emp_ID to get the user's personal information. It will enlist all the personal information that is ties to the specific Emp_ID.



Generate Reports:

This database application will allow the manager to generate various reports.. My application will help the admin to generate three reports. All reports uses aggregate function to evaluate the count of the users under various conditions.

Report 1: How much a hall spends weekly on employees?

This report enlists all the halls and the amount they spend on student employees.

Query1

Hall_ID	SumOfAmount
H001	185
H002	185
H003	100
H004	285
H005	110

Report 2: How much employee earn weekly from all hall

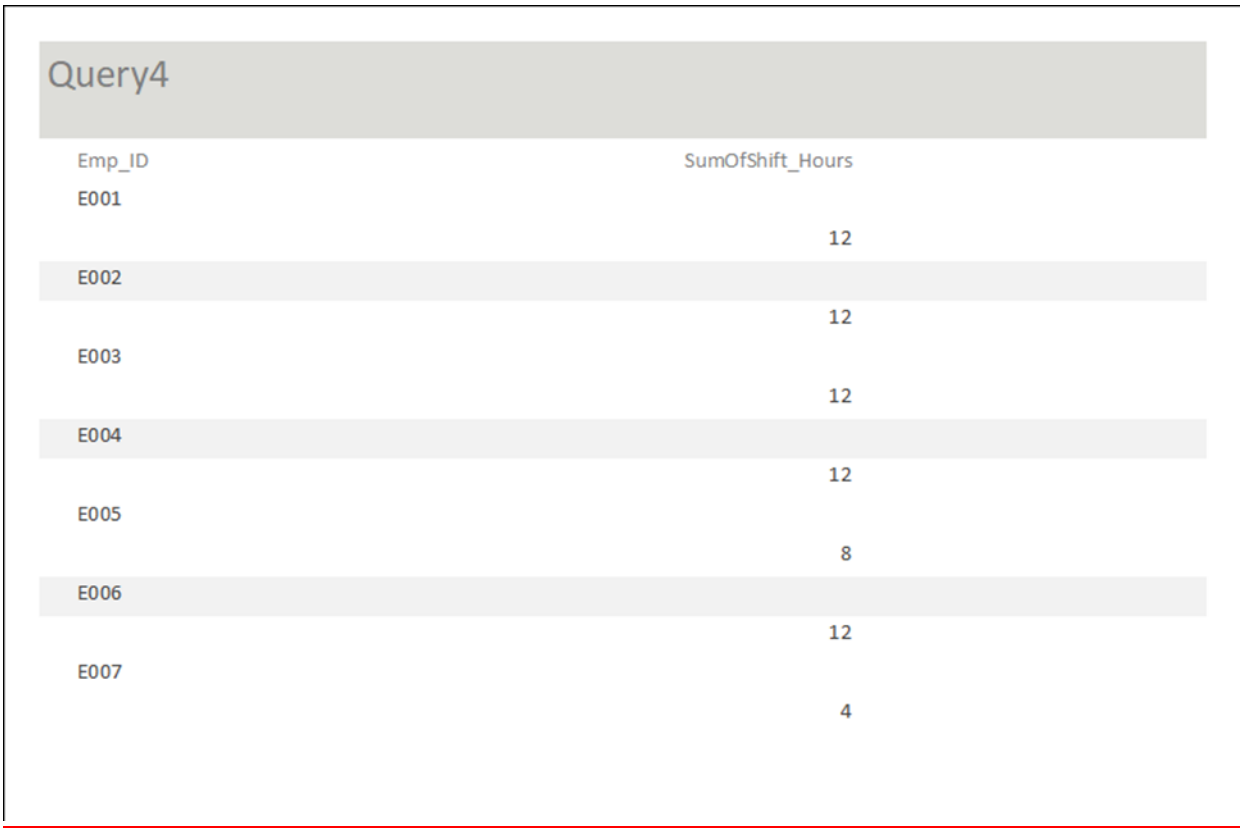
This report enlists all the employees and the amount each employee earned in a week.

Query2

Emp_ID	SumOfAmount
E001	220
E002	195
E003	180
E004	400
E005	65
E006	410

Report 3: How many hours a particular employee is working

This report enlist all the employees and the total number of hours they are working in a week.



Query4

Emp_ID	SumOfShift_Hours
E001	12
E002	12
E003	12
E004	12
E005	8
E006	12
E007	4

Triggers

I implemented a trigger to update the schedule of employee if he/she has dropped shift and it was approved by manager and also update the schedule of employee who picked a dropped shift.

Logic: Whenever the manager approves the drop request by entering 'Yes' in the Employee1Schedule table and 'DropShift' table and if someone picks that shift than Emp_ID of for that specific Employee1Schedule is updated

```
Create Trigger update_employee_schedule
On Employee1Schedule
After Insert ,Update
As
If @@ROWCOUNT >= AND (Select Picked From Employee1Schedule) = 'Yes'
Begin

    Update Employee1Schedule
    Set Emp_ID = (select Emp_ID From PickedShift)
    From
    (Select es.ES_ID, es.Emp_ID,es.Shift_ID from Employee1Schedule es
    Inner Join PickedShift p On p.ES_ID = es.ES_ID
    Where es.ES_ID IN (Select ES_ID from Employee1Schedule) AND es.Picked = 'Yes'
End;
```

Before running the trigger

ES_ID	Emp_ID	Shift_ID	Shift_Day	Dropped	Picked
ES01	E001	S004	Monday	Yes	Yes
ES02	E001	S005	Tuesday	Yes	NULL
ES03	E001	S013	Thursday	NULL	NULL
ES05	E002	S004	Wednesday	NULL	NULL
ES06	E002	S005	Friday	NULL	NULL
ES07	E002	S013	Thursday	NULL	NULL
ES08	E003	S008	Wednesday	NULL	NULL
ES09	E003	S003	Monday	NULL	NULL
ES10	E003	S011	Thursday	NULL	NULL
ES11	E004	S006	Monday	NULL	NULL
ES12	E004	S010	Sunday	NULL	NULL
ES13	E004	S012	Saturday	NULL	NULL
ES14	E005	S003	Wednesday	NULL	NULL
ES15	E005	S006	Sunday	NULL	NULL

After running the trigger

```
Insert Into PickedShift Values ('P006','E003','ES02');
```

ES_ID	Emp_ID	Shift_ID	Shift_Day	Dropped	Picked
ES01	E001	S004	Monday	Yes	Yes
ES02	E003	S005	Tuesday	Yes	Yes
ES03	E001	S013	Thursday	NULL	NULL
ES05	E002	S004	Wednesday	NULL	NULL
ES06	E002	S005	Friday	NULL	NULL
ES07	E002	S013	Thursday	NULL	NULL
ES08	E003	S008	Wednesday	NULL	NULL
ES09	E003	S003	Monday	NULL	NULL
ES10	E003	S011	Thursday	NULL	NULL
ES11	E004	S006	Monday	NULL	NULL
ES12	E004	S010	Sunday	NULL	NULL
ES13	E004	S012	Saturday	NULL	NULL
ES14	E005	S003	Wednesday	NULL	NULL