Department of Public Safety (DPS) service Database Application

Mohammed Farees Patel

Syracuse university

School of Information Studies

IST 659 - Project Implementation Report

Table of Contents

Summary

Background:

The DPS is the campus police force serving the Syracuse University campus and providing various escort services for the SU students, staff, and faculty especially during the night, to ensure that members of the campus community arrive safely to their destinations. Ordering a DPS escort service, deals with calling the DPS helpline service or DPS office. There are a lot of problems associated with this current process to order an escort service. Firstly, to order a DPS escort service, he/she needs to declare their SUID number, start location and the destination location over the phone. This is highly time-consuming and there exist multiple errors due to human hearing. Secondly, an individual may have to wait for up to 90 minutes if he/she is at a safe location (such as an on-campus building or residence) because the escort may be working with another student. As a result, an individual has to wait for an indefinite time in hope for the escort service to show up.

Designed Solution:

A database solution will allow the DPS to directly collect the information through an application. Users can enter their SUID number, starting location and the destination location to order a DPS escort service. Hence allowing users to plan their journey well ahead of time. In addition, the proposed database system will also enable users to provide a feedback about the driver and the escort service for quality control. Similarly, drivers can report the user for any misbehavior while commuting. The database solution works as follows:

- The user enters his/her SUID number, name, email ID, contact number, starting location and destination location, the escort service type, and the departure time to book for a DPS escort service
- The above information generates a unique Booking ID which is then analyzed by the administration staff for feasibility and authenticity. Admin staff then provides their decision to either accept or reject the booking.
- If the booking is accepted. The a DPS car and a driver will be allotted for the specific Booking ID and based on the availability the waiting time is generated for the user

Entity and Attribute Table:

DATAOBJECT: **DPS_EscortService**	**This database contains all the tables and relations that together build the booking DPS Escort Service system for the SU students, staff and faculty.**
BookingUser	**Contains all user information of students, staff and the faculty who orders for a DPS escort service.**
UserID	PRIMARY KEY: Each BookingUser has a unique UserID to identify the specific user This will help in identifying that the user login is a BookingUser i.e. a user who wants to order the DPS service
UserPassword	Required for authenticating the user
FName	User's First Name
LName	User's Last Name
UserEmail	User email address
UserContact	User contact details for instance the phone number
UserAddress	This refers to the user's residential address
UserKind	Determines the role of the booking user. [Possible values: Student(Stud), Faculty(Facu) or Staff(Staf)]
Student	**Child entity of BookingUser. Contains Student information**
UserID	PRIMARY KEY: Each student has a unique BookingUserID to identify himself/herself.

	FOREIGN KEY: Associated with Primary Key of ‘BookingUser’ table. This will help in identifying that the BookingUser is a student.
StudentDesc	For additional student description
Faculty	**Child entity of BookingUser. Contains Faculty information**
UserID	PRIMARY KEY: Each faculty member has a unique BookingUserID to identify himself/herself. FOREIGN KEY: Associated with Primary Key of ‘BookingUser’ table. This will help in identifying that the BookingUser is a faculty member.
FacultyDesc	For additional faculty member description
Staff	**Child entity of BookingUser. Contains Staff information**
UserID	PRIMARY KEY: Each staff member has a unique BookingUserID to identify himself/herself. FOREIGN KEY: Associated with Primary Key of ‘BookingUser’ table.

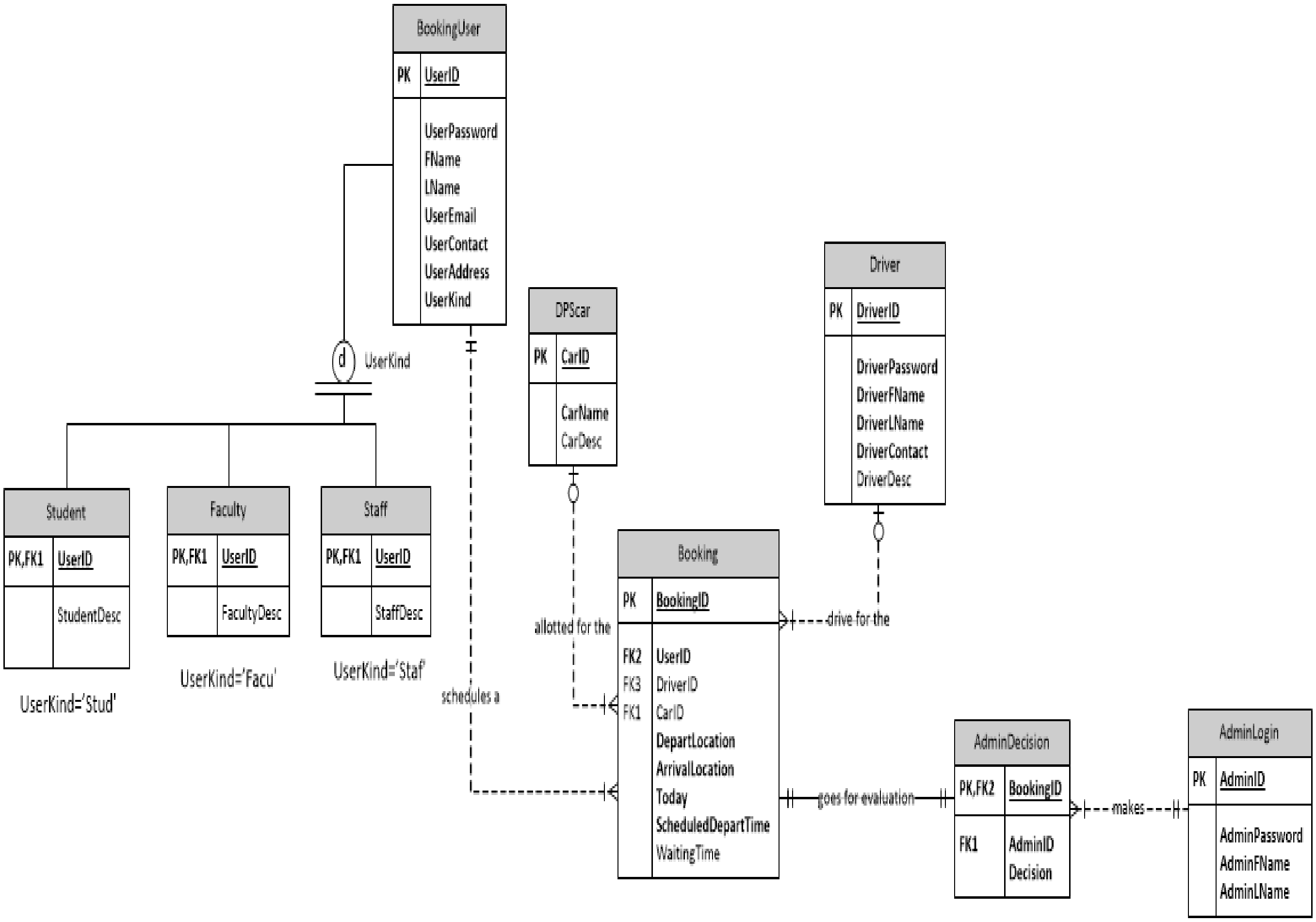
	This will help in identifying that the BookingUser is a staff member.
StaffDesc	For additional faculty staff member description
Driver	**Contains all the necessary driver information who drives the DPS car.**
DriverID	PRIMARY KEY: Each driver has a unique DriverID to identify himself/herself This will help in identifying that the user login is a Driver i.e. a user who drives the DPS car.
DriverPassword	Required for authenticating the user
DriverFName	Driver’s First Name
DriverLName	Driver’s Last Name
DriverContact	Driver’s contact details for instance the phone number
DriverDesc	Contains additional information i.e. available timings, specific commuting routes, etc.
AdminLogin	**Stores information of all the administrators who manages the bookings process and decisions.**
AdminID	PRIMARY KEY: Each admin member has a unique AdminID to identify himself/herself. This will help in identifying that the user login is a admin member i.e. one who manages the DPS booking processes and booking decisions
AdminPassword	Required for authenticating the user

AdminFName	Admin’s First Name
AdminLName	Admin’s Last Name
DPScar	**Stores information of all the cars that DPS drivers use for commuting**
CarID	PRIMARY KEY: Each car has a unique CarID which identifies the car (E.g.: car number plate)
CarName	Indicated the named and brand of the car
CarDesc	Stores additional information about the car
Booking	Stores information from the booking user regarding the booking for a DPS escort service
BookingID	PRIMARY KEY: Each booking has a unique BookingID number which identifies the specific booking made by a specific booking user.
UserID	FOREIGN KEY: Associated with Primary Key of ‘BookingUser’ table. Contains information of a booking user who schedules a booking. This indicates the booking made by either the student, faculty or staff.
DriverID	FOREIGN KEY: Associated with Primary Key of ‘Driver’ table. Contains information of a driver who is assigned to drive for the booking if the booking is accepted by the admin.

CarID	FOREIGN KEY: Associated with Primary Key of 'DPScar' table. Contains information of the car which is assigned to the driver for the booking if the booking is accepted by the admin.
DepartLocation	Contains information about the pickup location.
ArrivalLocation	Contains information about the destination location.
Today	Automatically inserts the date of booking
ScheduledDepartTime	Contains information about the pickup time.
WaitingTime	Contains information about the time required for the DPS to show up.
AdminDecision	**Stores information about the bookings decisions being confirmed or rejected by the admin**
BookingID	PRIMARY KEY: BookingID is the primary key. This key as a whole determines the booking decision i.e. whether the administration has accepted or rejected the user booking. FOREIGN KEY: Associated with Primary Key of 'Booking' table. The foreign key helps in identifying the booking decision for a specific booking ID.

AdminID	FOREIGN KEY: Associated with Primary Key of ‘AdminLogin’ table. The admin ID will help in identifying the booking decision made by the specific admin using his/her AdminID.
Decision	Uniquely identifies the status of the booking i.e. accepted/rejected using the specific BookingID

Visio Relational Data Model



BUSINESS RULES:

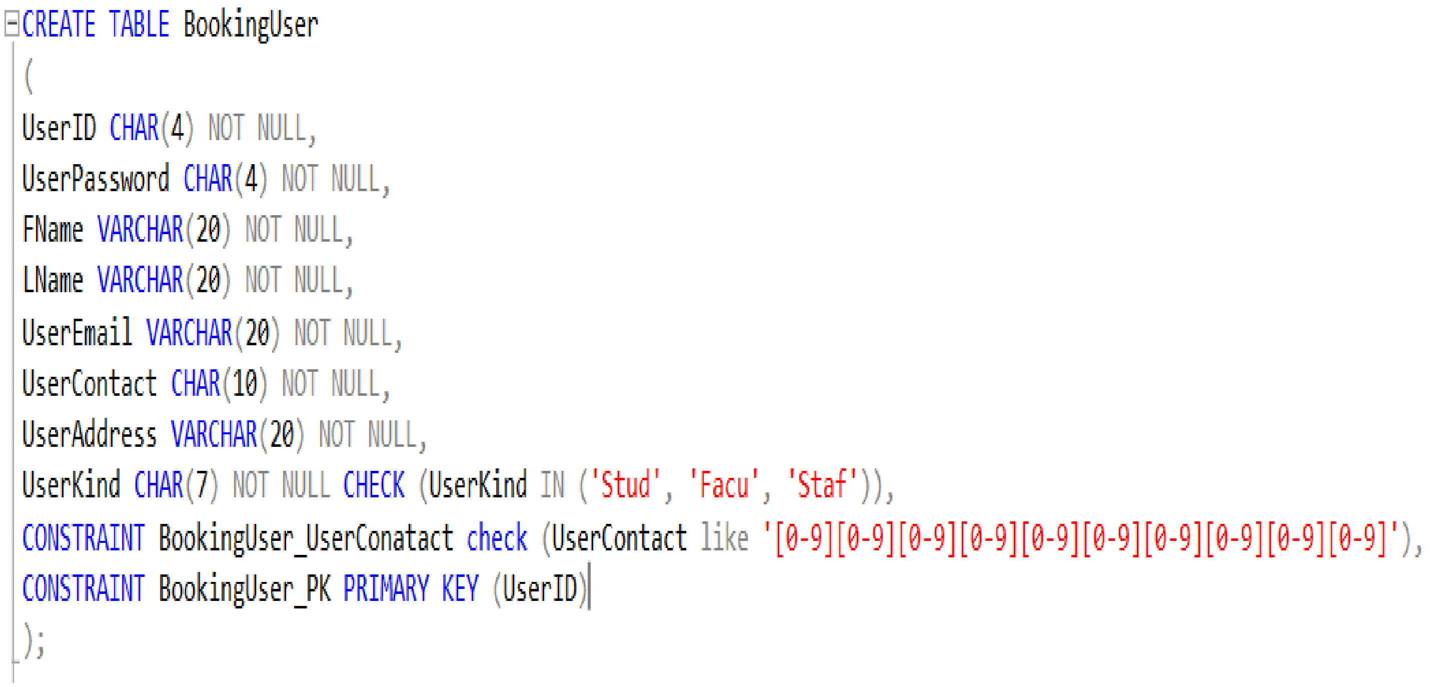
- Every user login must be either Administrator, Booking user or the Driver
- Booking user must be either Student (Stud), Faculty (Facu) or Staff (Staf)
- One booking user (Student, Faculty or Staff) can make only one booking at a time
- One booking decision can be made by only one admin
- Admin can either accept or reject the booking decision
- Only one driver and a car can be allotted for one booking

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.

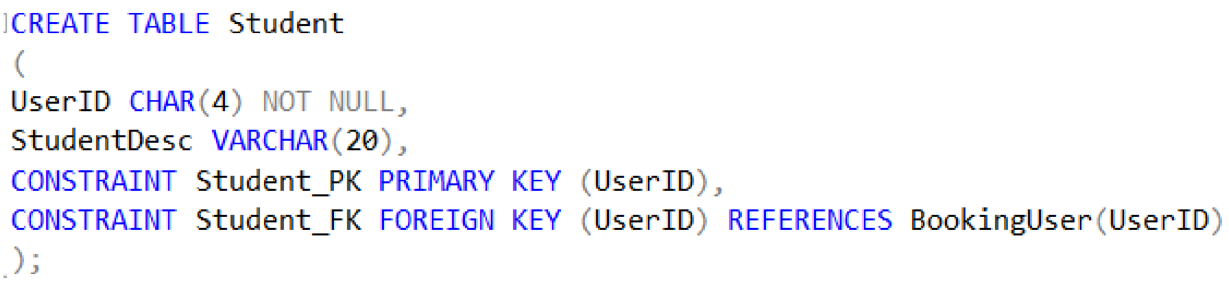
SQL Scripts for Creating and Inserting Sample Data:

CREATE: BookingUser



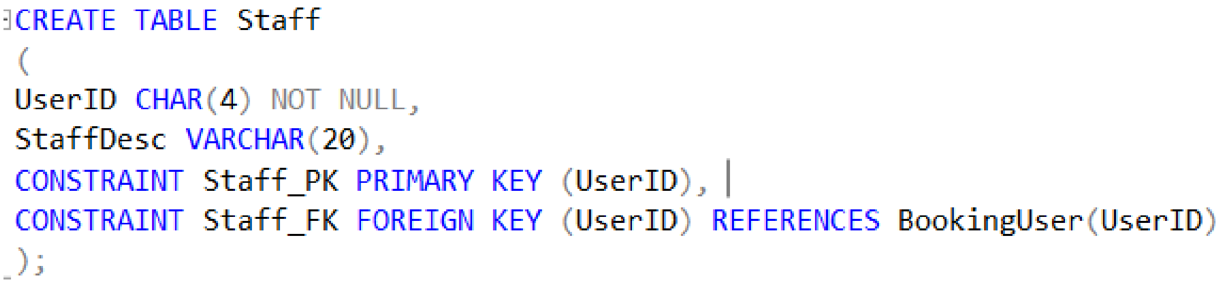
```
CREATE TABLE BookingUser
(
UserID CHAR(4) NOT NULL,
UserPassword CHAR(4) NOT NULL,
FName VARCHAR(20) NOT NULL,
LName VARCHAR(20) NOT NULL,
UserEmail VARCHAR(20) NOT NULL,
UserContact CHAR(10) NOT NULL,
UserAddress VARCHAR(20) NOT NULL,
UserKind CHAR(7) NOT NULL CHECK (UserKind IN ('Stud', 'Facu', 'Staf')),
CONSTRAINT BookingUser_UserConatact check (UserContact like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
CONSTRAINT BookingUser_PK PRIMARY KEY (UserID)
);
```

CREATE: Student



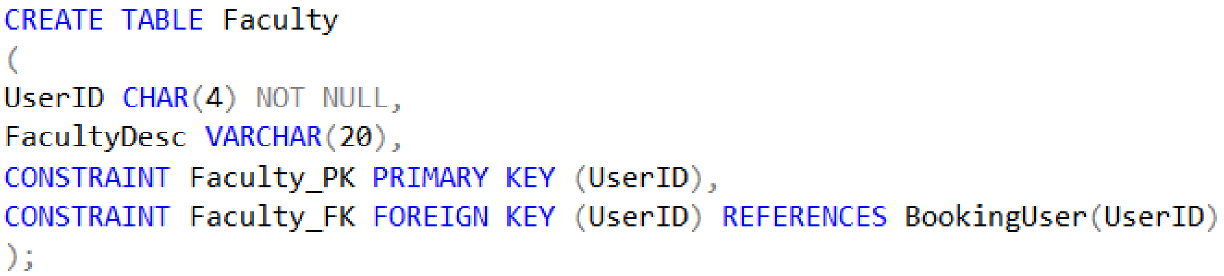
```
CREATE TABLE Student
(
UserID CHAR(4) NOT NULL,
StudentDesc VARCHAR(20),
CONSTRAINT Student_PK PRIMARY KEY (UserID),
CONSTRAINT Student_FK FOREIGN KEY (UserID) REFERENCES BookingUser(UserID)
);
```

CREATE: Staff



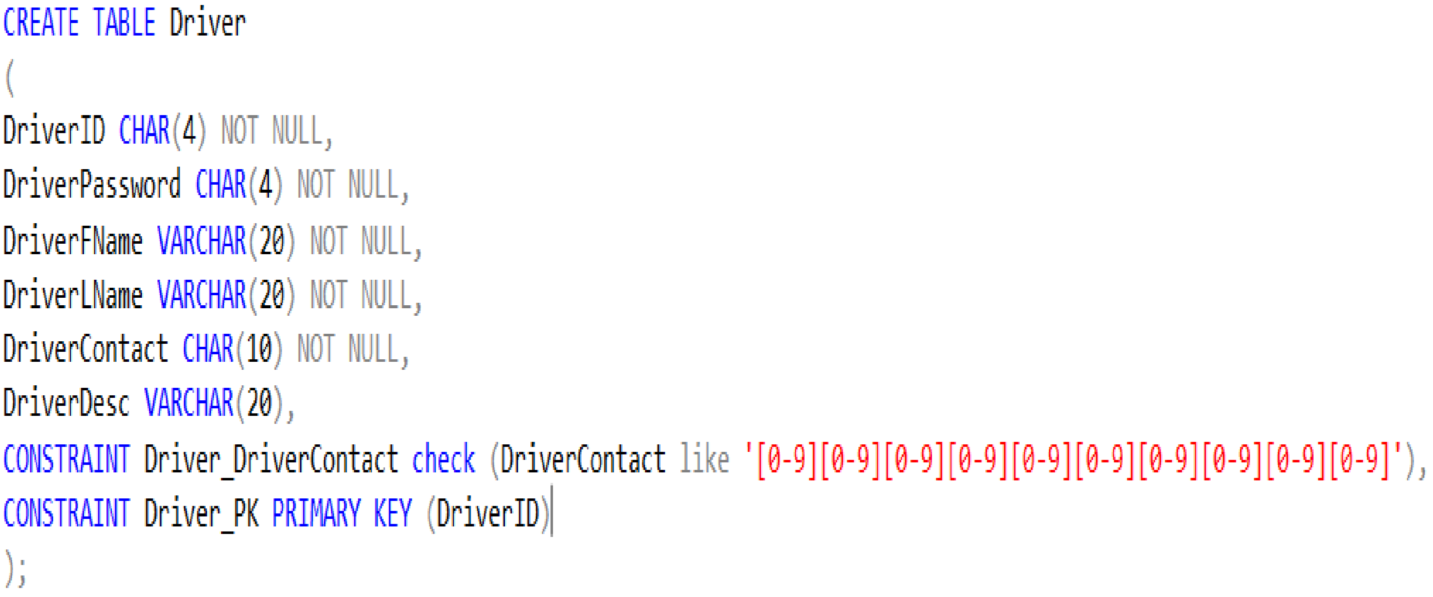
```
CREATE TABLE Staff
(
UserID CHAR(4) NOT NULL,
StaffDesc VARCHAR(20),
CONSTRAINT Staff_PK PRIMARY KEY (UserID),
CONSTRAINT Staff_FK FOREIGN KEY (UserID) REFERENCES BookingUser(UserID)
);
```

CREATE: Faculty



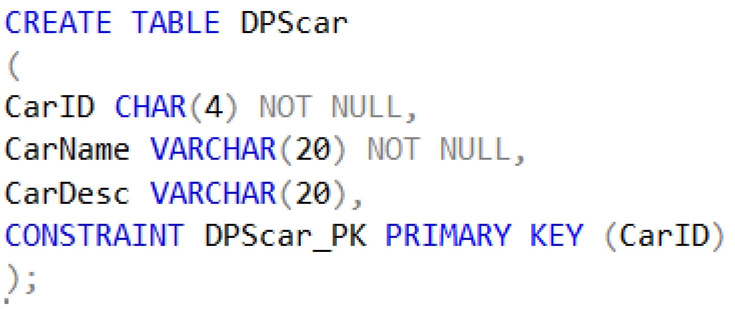
```
CREATE TABLE Faculty
(
UserID CHAR(4) NOT NULL,
FacultyDesc VARCHAR(20),
CONSTRAINT Faculty_PK PRIMARY KEY (UserID),
CONSTRAINT Faculty_FK FOREIGN KEY (UserID) REFERENCES BookingUser(UserID)
);
```

CREATE: Driver



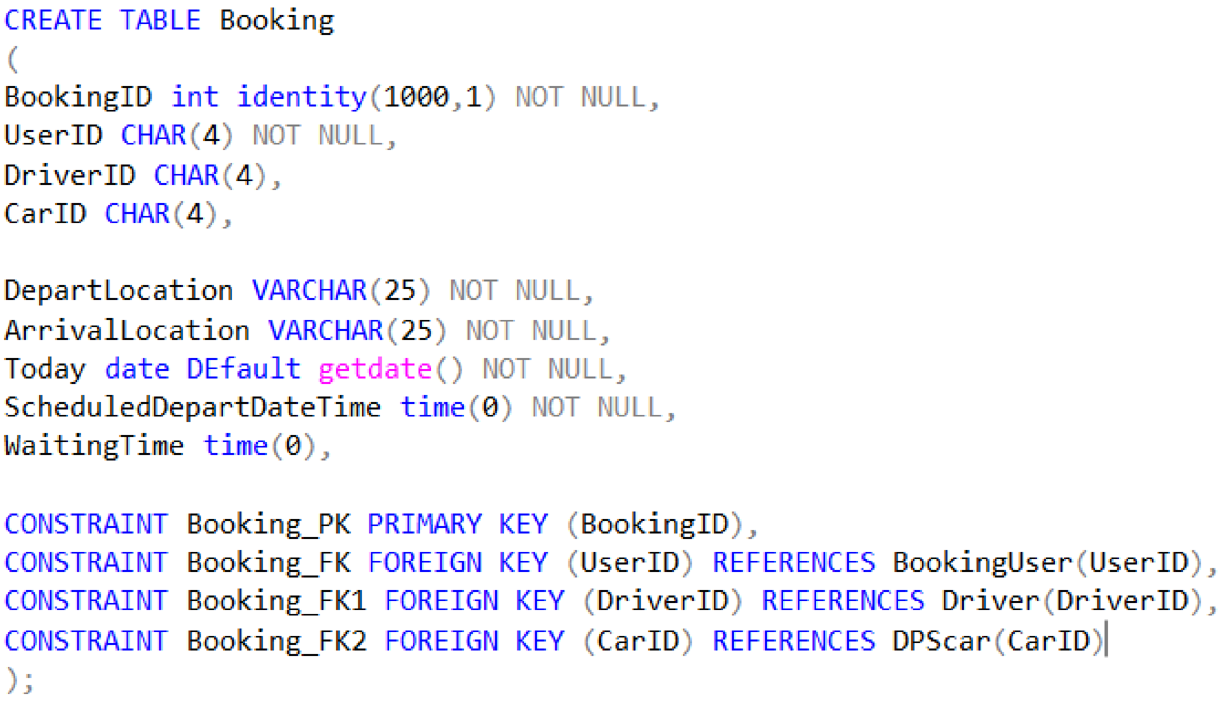
```
CREATE TABLE Driver
(
DriverID CHAR(4) NOT NULL,
DriverPassword CHAR(4) NOT NULL,
DriverFName VARCHAR(20) NOT NULL,
DriverLName VARCHAR(20) NOT NULL,
DriverContact CHAR(10) NOT NULL,
DriverDesc VARCHAR(20),
CONSTRAINT Driver_DriverContact check (DriverContact like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
CONSTRAINT Driver_PK PRIMARY KEY (DriverID)
);
```

CREATE: DPScar



```
CREATE TABLE DPScar
(
CarID CHAR(4) NOT NULL,
CarName VARCHAR(20) NOT NULL,
CarDesc VARCHAR(20),
CONSTRAINT DPScar_PK PRIMARY KEY (CarID)
);
```

CREATE: Booking

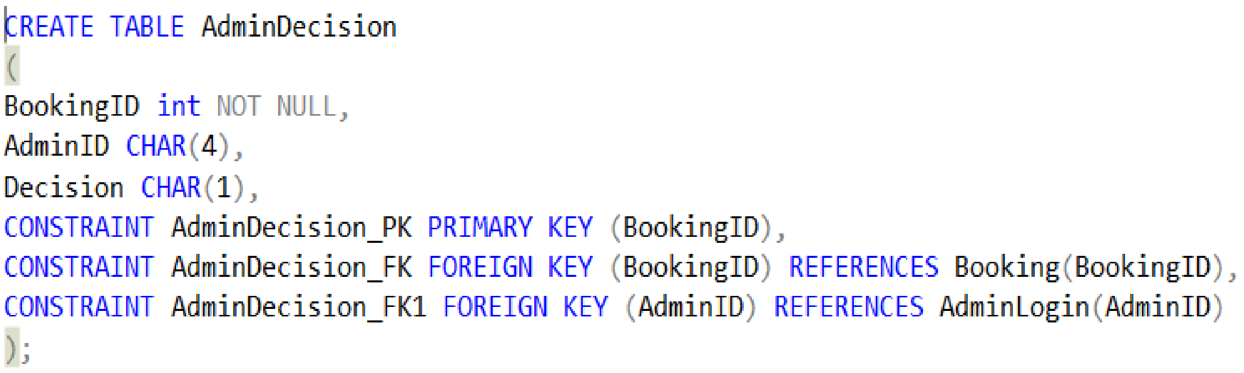


```
CREATE TABLE Booking
(
BookingID int identity(1000,1) NOT NULL,
UserID CHAR(4) NOT NULL,
DriverID CHAR(4),
CarID CHAR(4),

DepartLocation VARCHAR(25) NOT NULL,
ArrivalLocation VARCHAR(25) NOT NULL,
Today date DEfault getdate() NOT NULL,
ScheduledDepartDateTime time(0) NOT NULL,
WaitingTime time(0),

CONSTRAINT Booking_PK PRIMARY KEY (BookingID),
CONSTRAINT Booking_FK FOREIGN KEY (UserID) REFERENCES BookingUser(UserID),
CONSTRAINT Booking_FK1 FOREIGN KEY (DriverID) REFERENCES Driver(DriverID),
CONSTRAINT Booking_FK2 FOREIGN KEY (CarID) REFERENCES DPScar(CarID)
);
```

CREATE: AdminDecision



```
CREATE TABLE AdminDecision
(
BookingID int NOT NULL,
AdminID CHAR(4),
Decision CHAR(1),
CONSTRAINT AdminDecision_PK PRIMARY KEY (BookingID),
CONSTRAINT AdminDecision_FK FOREIGN KEY (BookingID) REFERENCES Booking(BookingID),
CONSTRAINT AdminDecision_FK1 FOREIGN KEY (AdminID) REFERENCES AdminLogin(AdminID)
);
```

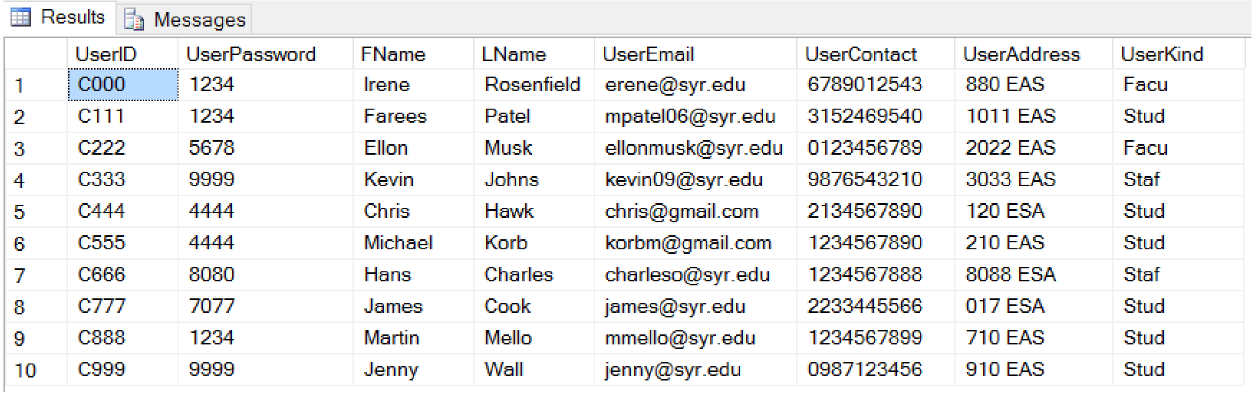
CREATE: AdminLogin

```
CREATE TABLE AdminLogin
(
AdminID CHAR(4),
AdminPassword CHAR(4),
AdminFName VARCHAR(20),
AdminLName VARCHAR(20),
CONSTRAINT AdminLogin_PK PRIMARY KEY (AdminID)
);
```

Insert Data in BookingUser Table:

```
INSERT INTO BookingUser VALUES ('C111','1234','Farees','Patel','mpatel06@syr.edu', '3152469540','1011 EAS','Stud')
INSERT INTO BookingUser VALUES ('C222','5678','Ellon','Musk','ellonmusk@syr.edu', '0123456789','2022 EAS','Facu')
INSERT INTO BookingUser VALUES ('C333','9999','Kevin','Johns','kevin09@syr.edu', '9876543210','3033 EAS','Staf')
INSERT INTO BookingUser VALUES ('C444','4444','Chris','Hawk','chris@gmail.com', '2134567890','120 ESA','Stud')
INSERT INTO BookingUser VALUES ('C555','4444','Michael','Korb','korbm@gmail.com', '1234567890','210 EAS','Stud')
INSERT INTO BookingUser VALUES ('C888','1234','Martin','Mello','mmello@syr.edu', '1234567899','710 EAS','Stud')
INSERT INTO BookingUser VALUES ('C666','8080','Hans','Charles','charleso@syr.edu', '1234567888','8088 ESA','Staff')
INSERT INTO BookingUser VALUES ('C777','7077','James','Cook','james@syr.edu', '2233445566','017 ESA','Stud')
INSERT INTO BookingUser VALUES ('C999','9999','Jenny','Wall','jenny@syr.edu', '0987123456','910 EAS','Stud')
INSERT INTO BookingUser VALUES ('C000','1234','Irene','Rosenfield','erene@syr.edu', '6789012543','880 EAS','Facu')
```

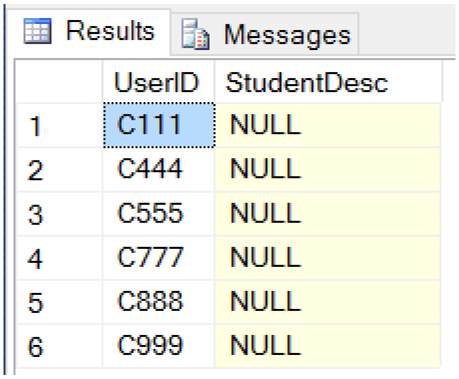
```
SELECT * from BookingUser
```



	UserID	UserPassword	FName	LName	UserEmail	UserContact	UserAddress	UserKind
1	C000	1234	Irene	Rosenfield	erene@syr.edu	6789012543	880 EAS	Facu
2	C111	1234	Farees	Patel	mpatel06@syr.edu	3152469540	1011 EAS	Stud
3	C222	5678	Ellon	Musk	ellonmusk@syr.edu	0123456789	2022 EAS	Facu
4	C333	9999	Kevin	Johns	kevin09@syr.edu	9876543210	3033 EAS	Staf
5	C444	4444	Chris	Hawk	chris@gmail.com	2134567890	120 ESA	Stud
6	C555	4444	Michael	Korb	korbm@gmail.com	1234567890	210 EAS	Stud
7	C666	8080	Hans	Charles	charleso@syr.edu	1234567888	8088 ESA	Staf
8	C777	7077	James	Cook	james@syr.edu	2233445566	017 ESA	Stud
9	C888	1234	Martin	Mello	mmello@syr.edu	1234567899	710 EAS	Stud
10	C999	9999	Jenny	Wall	jenny@syr.edu	0987123456	910 EAS	Stud

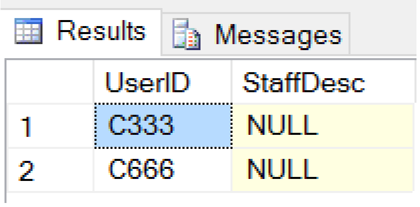
Automatic insert into Student, Faculty and Staff Table due to Trigger:

```
SELECT * from Student
```



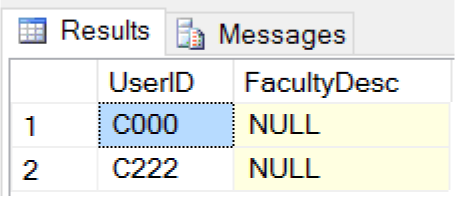
	UserID	StudentDesc
1	C111	NULL
2	C444	NULL
3	C555	NULL
4	C777	NULL
5	C888	NULL
6	C999	NULL

```
SELECT * from Staff
```



	UserID	StaffDesc
1	C333	NULL
2	C666	NULL

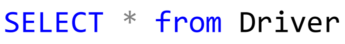
```
SELECT * from Faculty
```



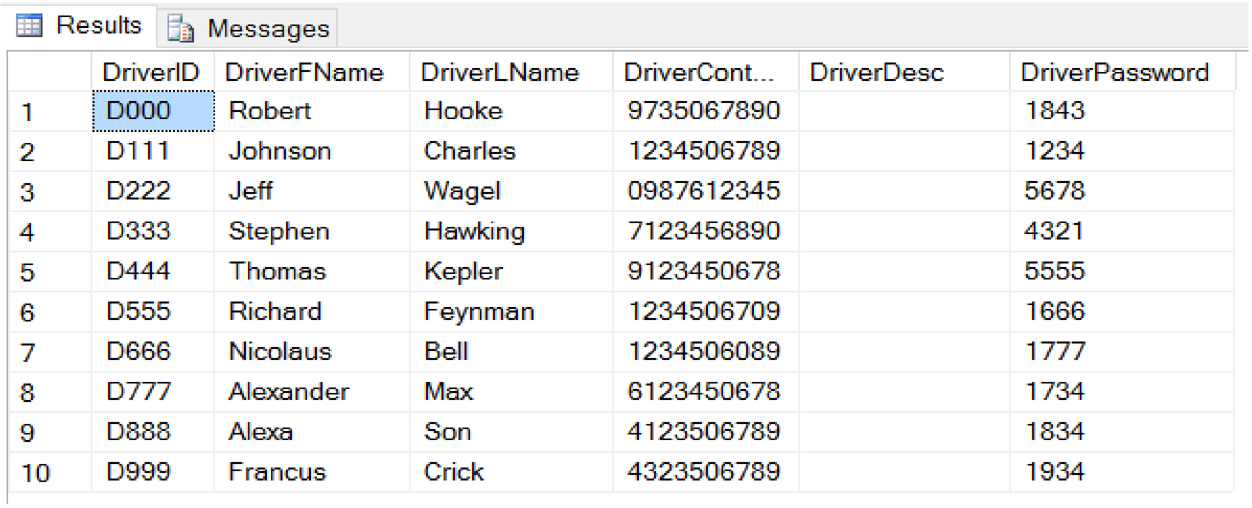
	UserID	FacultyDesc
1	C000	NULL
2	C222	NULL

Insert Data in Driver Table:

```
Insert into Driver Values ('D111','Johnson','Charles','1234506789','','1234')
Insert into Driver Values ('D222','Jeff','Wagel','0987612345','','5678')
Insert into Driver Values ('D333','Stephen','Hawking','7123456890','','4321')
Insert into Driver Values ('D444','Thomas','Kepler','9123450678','','5555')
Insert into Driver Values ('D555','Richard','Feynman','81234506709','','1666')
Insert into Driver Values ('D666','Nicolaus','Bell','71234506089','','1777')
Insert into Driver Values ('D777','Alexander','Max','6123450678','','1734')
Insert into Driver Values ('D888','Alexa','Son','4123506789','','1834')
Insert into Driver Values ('D999','Francus','Crick','4323506789','','1934')
Insert into Driver Values ('D000','Robert','Hooke','973506789','','1843')
```

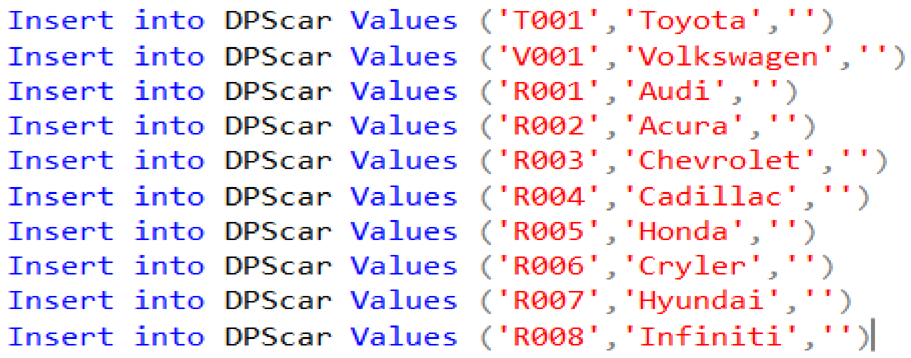


```
SELECT * from Driver
```

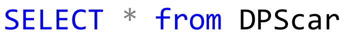


	DriverID	DriverFName	DriverLName	DriverCont...	DriverDesc	DriverPassword
1	D000	Robert	Hooke	9735067890		1843
2	D111	Johnson	Charles	1234506789		1234
3	D222	Jeff	Wagel	0987612345		5678
4	D333	Stephen	Hawking	7123456890		4321
5	D444	Thomas	Kepler	9123450678		5555
6	D555	Richard	Feynman	1234506709		1666
7	D666	Nicolaus	Bell	1234506089		1777
8	D777	Alexander	Max	6123450678		1734
9	D888	Alexa	Son	4123506789		1834
10	D999	Francus	Crick	4323506789		1934

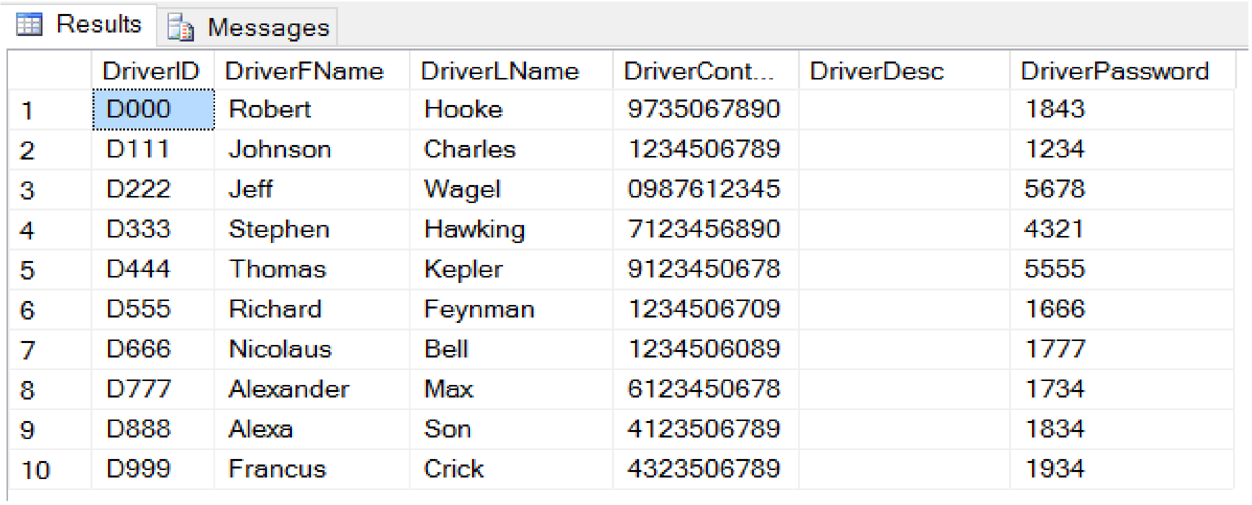
Insert Data in DPScar Table:



```
Insert into DPScar Values ('T001','Toyota','')
Insert into DPScar Values ('V001','Volkswagen','')
Insert into DPScar Values ('R001','Audi','')
Insert into DPScar Values ('R002','Acura','')
Insert into DPScar Values ('R003','Chevrolet','')
Insert into DPScar Values ('R004','Cadillac','')
Insert into DPScar Values ('R005','Honda','')
Insert into DPScar Values ('R006','Cryler','')
Insert into DPScar Values ('R007','Hyundai','')
Insert into DPScar Values ('R008','Infiniti','')
```

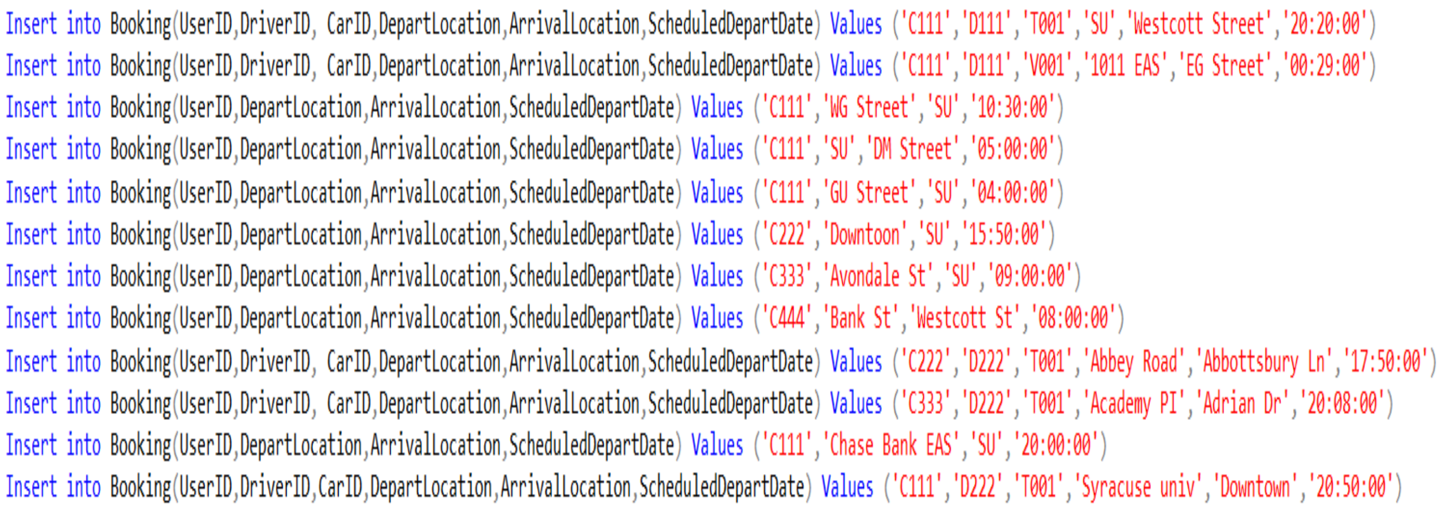


```
SELECT * from DPScar
```

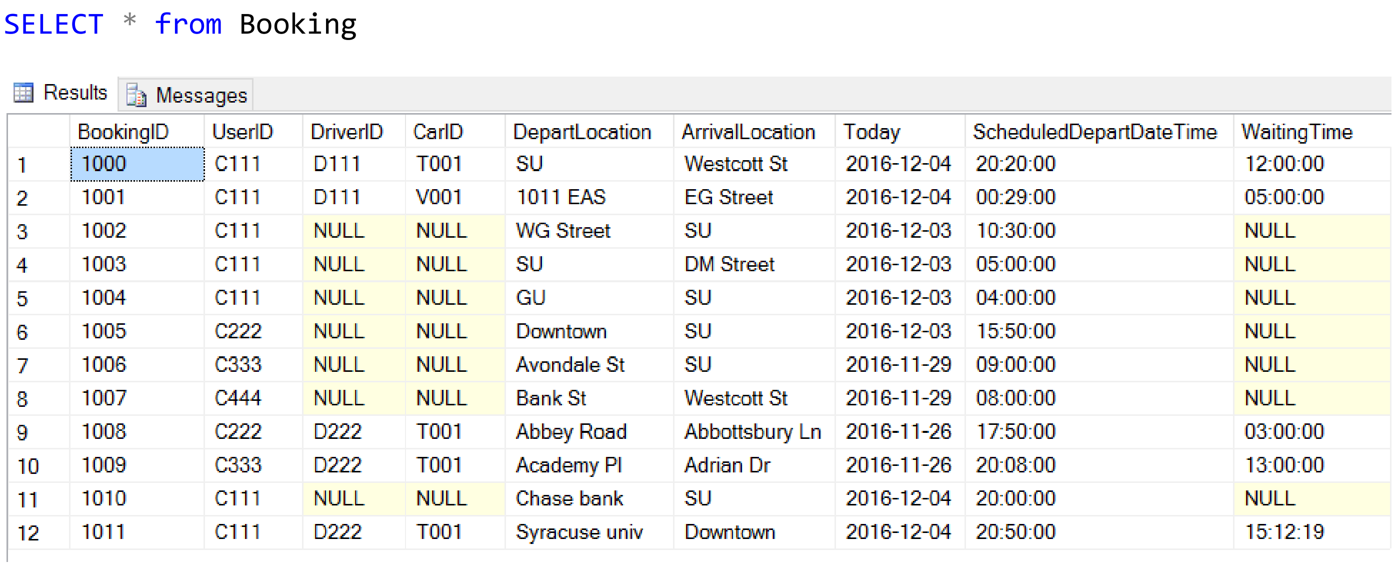


	DriverID	DriverFName	DriverLName	DriverCont...	DriverDesc	DriverPassword
1	D000	Robert	Hooke	9735067890		1843
2	D111	Johnson	Charles	1234506789		1234
3	D222	Jeff	Wagel	0987612345		5678
4	D333	Stephen	Hawking	7123456890		4321
5	D444	Thomas	Kepler	9123450678		5555
6	D555	Richard	Feynman	1234506709		1666
7	D666	Nicolaus	Bell	1234506089		1777
8	D777	Alexander	Max	6123450678		1734
9	D888	Alexa	Son	4123506789		1834
10	D999	Francus	Crick	4323506789		1934

Insert Data in Booking Table:



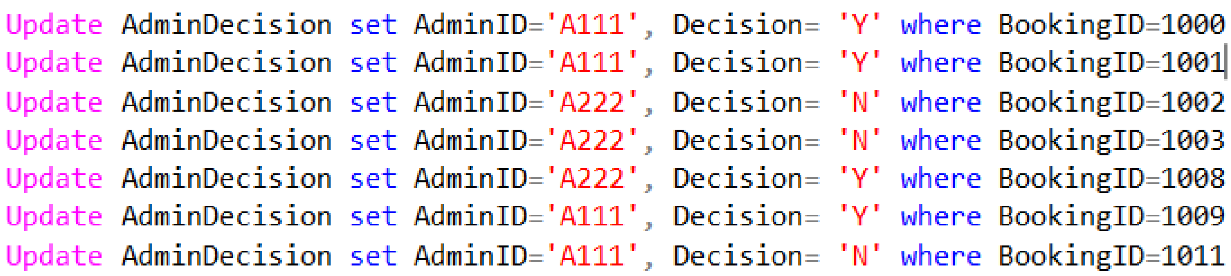
```sql
Insert into Booking(UserID,DriverID, CarID,DepartLocation,ArrivalLocation,ScheduledDepartDate) Values ('C111','D111','T001','SU','Westcott Street','20:20:00')
Insert into Booking(UserID,DriverID, CarID,DepartLocation,ArrivalLocation,ScheduledDepartDate) Values ('C111','D111','V001','1011 EAS','EG Street','00:29:00')
Insert into Booking(UserID,DepartLocation,ArrivalLocation,ScheduledDepartDate) Values ('C111','WG Street','SU','10:30:00')
Insert into Booking(UserID,DepartLocation,ArrivalLocation,ScheduledDepartDate) Values ('C111','SU','DM Street','05:00:00')
Insert into Booking(UserID,DepartLocation,ArrivalLocation,ScheduledDepartDate) Values ('C111','GU Street','SU','04:00:00')
Insert into Booking(UserID,DepartLocation,ArrivalLocation,ScheduledDepartDate) Values ('C222','Downtoon','SU','15:50:00')
Insert into Booking(UserID,DepartLocation,ArrivalLocation,ScheduledDepartDate) Values ('C333','Avondale St','SU','09:00:00')
Insert into Booking(UserID,DepartLocation,ArrivalLocation,ScheduledDepartDate) Values ('C444','Bank St','Westcott St','08:00:00')
Insert into Booking(UserID,DriverID, CarID,DepartLocation,ArrivalLocation,ScheduledDepartDate) Values ('C222','D222','T001','Abbey Road','Abbottsbury Ln','17:50:00')
Insert into Booking(UserID,DriverID, CarID,DepartLocation,ArrivalLocation,ScheduledDepartDate) Values ('C333','D222','T001','Academy PI','Adrian Dr','20:08:00')
Insert into Booking(UserID,DepartLocation,ArrivalLocation,ScheduledDepartDate) Values ('C111','Chase Bank EAS','SU','20:00:00')
Insert into Booking(UserID,DriverID,CarID,DepartLocation,ArrivalLocation,ScheduledDepartDate) Values ('C111','D222','T001','Syracuse univ','Downtown','20:50:00')
```



```sql
SELECT * from Booking
```

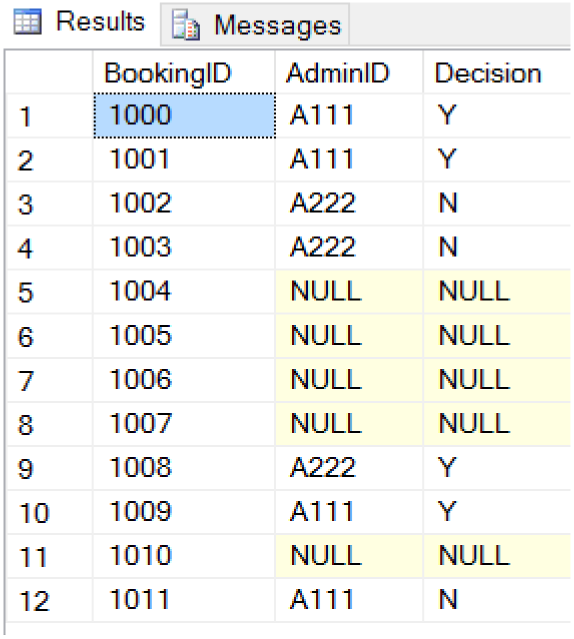
	BookingID	UserID	DriverID	CarID	DepartLocation	ArrivalLocation	Today	ScheduledDepartDateTime	WaitingTime
1	1000	C111	D111	T001	SU	Westcott St	2016-12-04	20:20:00	12:00:00
2	1001	C111	D111	V001	1011 EAS	EG Street	2016-12-04	00:29:00	05:00:00
3	1002	C111	NULL	NULL	WG Street	SU	2016-12-03	10:30:00	NULL
4	1003	C111	NULL	NULL	SU	DM Street	2016-12-03	05:00:00	NULL
5	1004	C111	NULL	NULL	GU	SU	2016-12-03	04:00:00	NULL
6	1005	C222	NULL	NULL	Downtown	SU	2016-12-03	15:50:00	NULL
7	1006	C333	NULL	NULL	Avondale St	SU	2016-11-29	09:00:00	NULL
8	1007	C444	NULL	NULL	Bank St	Westcott St	2016-11-29	08:00:00	NULL
9	1008	C222	D222	T001	Abbey Road	Abbottsbury Ln	2016-11-26	17:50:00	03:00:00
10	1009	C333	D222	T001	Academy Pl	Adrian Dr	2016-11-26	20:08:00	13:00:00
11	1010	C111	NULL	NULL	Chase bank	SU	2016-12-04	20:00:00	NULL
12	1011	C111	D222	T001	Syracuse univ	Downtown	2016-12-04	20:50:00	15:12:19

Insert Data in AdminDecision Table (Need to update the values due to active trigger):



```sql
Update AdminDecision set AdminID='A111', Decision= 'Y' where BookingID=1000
Update AdminDecision set AdminID='A111', Decision= 'Y' where BookingID=1001
Update AdminDecision set AdminID='A222', Decision= 'N' where BookingID=1002
Update AdminDecision set AdminID='A222', Decision= 'N' where BookingID=1003
Update AdminDecision set AdminID='A222', Decision= 'Y' where BookingID=1008
Update AdminDecision set AdminID='A111', Decision= 'Y' where BookingID=1009
Update AdminDecision set AdminID='A111', Decision= 'N' where BookingID=1011
```

```
SELECT * from AdminDecision
```

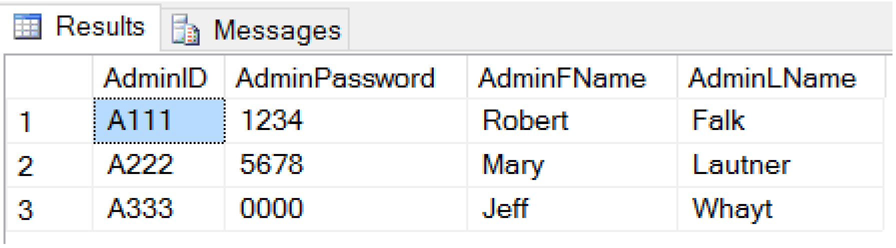


	BookingID	AdminID	Decision
1	1000	A111	Y
2	1001	A111	Y
3	1002	A222	N
4	1003	A222	N
5	1004	NULL	NULL
6	1005	NULL	NULL
7	1006	NULL	NULL
8	1007	NULL	NULL
9	1008	A222	Y
10	1009	A111	Y
11	1010	NULL	NULL
12	1011	A111	N

Insert Data in AdminLogin Table:

```
Insert into AdminLogin values ('A111','1234','Robert','Falk')
Insert into AdminLogin values ('A222','5678','Mary','Lautner')
Insert into AdminLogin values ('A333','0000','Jeff' ,'Whayt')
```

```
SELECT * from AdminLogin
```



	AdminID	AdminPassword	AdminFName	AdminLName
1	A111	1234	Robert	Falk
2	A222	5678	Mary	Lautner
3	A333	0000	Jeff	Whayt

MAJOR DATA QUESTIONS:

Since the current system for booking a DPS escort service is a manual system, a database application solution will turn to a completely online system to manage and process booking efficiently.

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

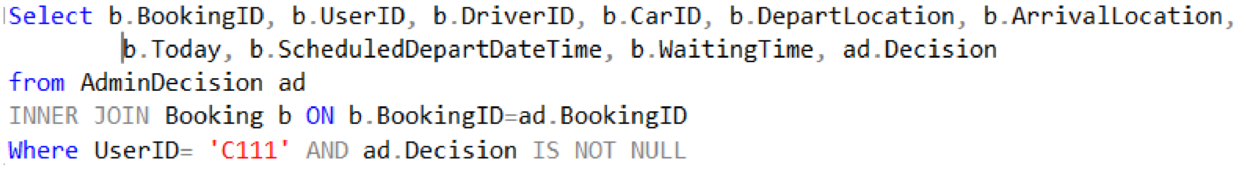
- SU Users (students/staff/faculty)
- DPS administrators
- Drivers

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

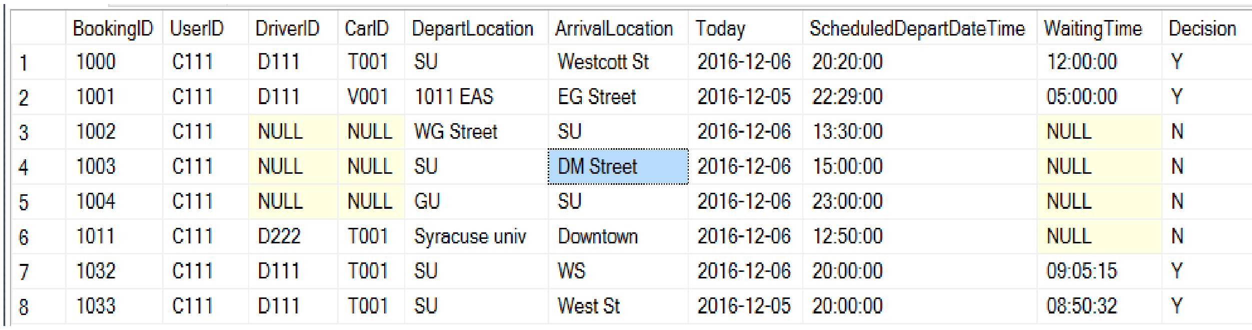
1. Why SU Users (students/staff/faculty) query the database

- **SU users can query the database to view their previous booking**

  The SU users of my database application can view all the bookings they have made in the past. Users can view all their bookings that were either approved or disapproved by the administrator. This section will enlist only the past bookings of the user who has login to the system successfully. Below is the SQL query to demonstrate this function. Let's assume the UserID to be C111.



```
Select b.BookingID, b.UserID, b.DriverID, b.CarID, b.DepartLocation, b.ArrivalLocation,
        b.Today, b.ScheduledDepartDateTime, b.WaitingTime, ad.Decision
from AdminDecision ad
INNER JOIN Booking b ON b.BookingID=ad.BookingID
Where UserID= 'C111' AND ad.Decision IS NOT NULL
```

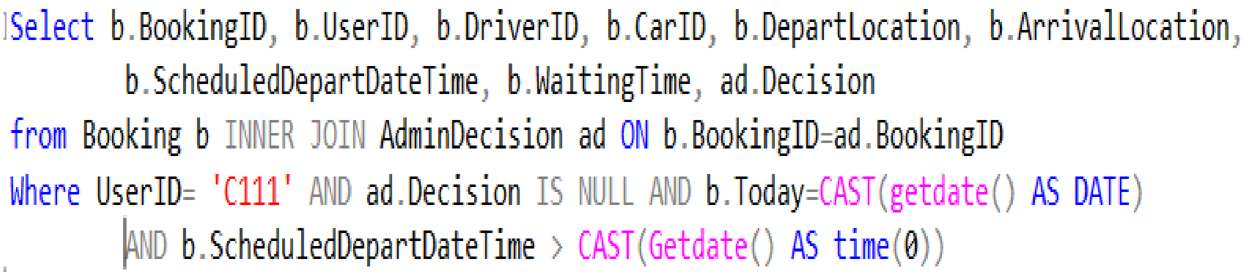


	BookingID	UserID	DriverID	CarID	DepartLocation	ArrivalLocation	Today	ScheduledDepartDateTime	WaitingTime	Decision
1	1000	C111	D111	T001	SU	Westcott St	2016-12-06	20:20:00	12:00:00	Y
2	1001	C111	D111	V001	1011 EAS	EG Street	2016-12-05	22:29:00	05:00:00	Y
3	1002	C111	NULL	NULL	WG Street	SU	2016-12-06	13:30:00	NULL	N
4	1003	C111	NULL	NULL	SU	DM Street	2016-12-06	15:00:00	NULL	N
5	1004	C111	NULL	NULL	GU	SU	2016-12-06	23:00:00	NULL	N
6	1011	C111	D222	T001	Syracuse univ	Downtown	2016-12-06	12:50:00	NULL	N
7	1032	C111	D111	T001	SU	WS	2016-12-06	20:00:00	09:05:15	Y
8	1033	C111	D111	T001	SU	West St	2016-12-05	20:00:00	08:50:32	Y

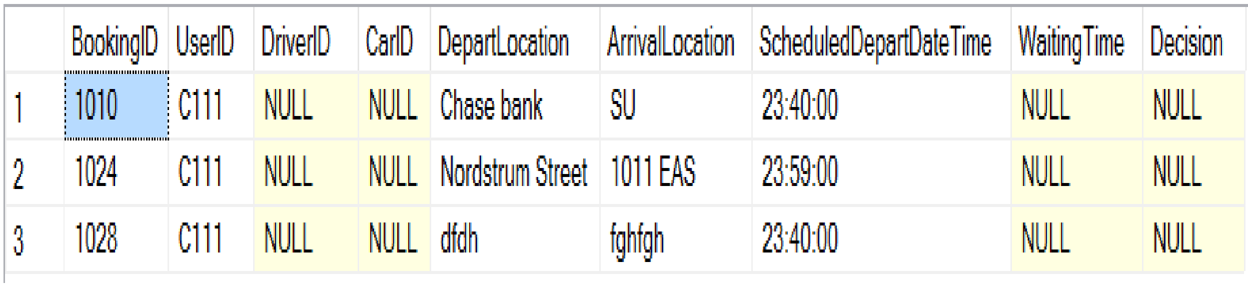
- **SU users can query the database to order a DPS escort service**

Users query the database to book an escort service by entering their UserID, Pickup-location and destination location. And if the booking is approved the waiting time is calculated for the user.

- **SU users can query the database to find bookings that are yet to be evaluated** Students, Staff or Faculty can order an escort service and can view all their bookings that are under processing and have not received any decision. A booking record under this section of the application will indicate that the booking request has been received by the admin but the admin has been yet to evaluate the booking.

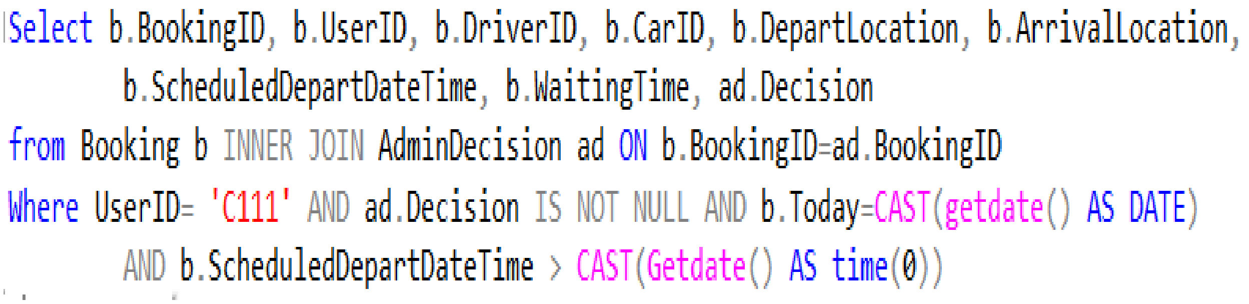


```
Select b.BookingID, b.UserID, b.DriverID, b.CarID, b.DepartLocation, b.ArrivalLocation,
        b.ScheduledDepartDateTime, b.WaitingTime, ad.Decision
from Booking b INNER JOIN AdminDecision ad ON b.BookingID=ad.BookingID
Where UserID= 'C111' AND ad.Decision IS NULL AND b.Today=CAST(getdate() AS DATE)
        AND b.ScheduledDepartDateTime > CAST(Getdate() AS time(0))
```

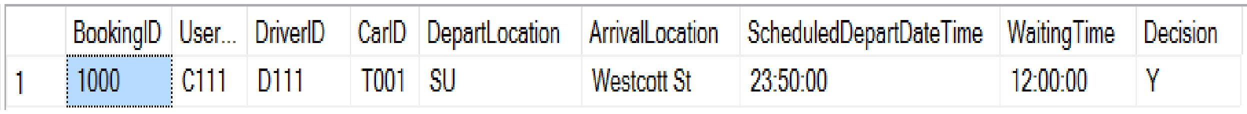


	BookingID	UserID	DriverID	CarID	DepartLocation	ArrivalLocation	ScheduledDepartDateTime	WaitingTime	Decision
1	1010	C111	NULL	NULL	Chase bank	SU	23:40:00	NULL	NULL
2	1024	C111	NULL	NULL	Nordstrum Street	1011 EAS	23:59:00	NULL	NULL
3	1028	C111	NULL	NULL	dfdh	fghfgh	23:40:00	NULL	NULL

- **SU users query the database to find bookings that were approved by admin** Students, Staff or Faculty can order an escort service and can view all their bookings that are processed and have received a decision. A booking record under this section of the application will indicate that the booking request has been evaluated by the admin and the admin has either approved or disapproved the booking.



```
Select b.BookingID, b.UserID, b.DriverID, b.CarID, b.DepartLocation, b.ArrivalLocation,
        b.ScheduledDepartDateTime, b.WaitingTime, ad.Decision
from Booking b INNER JOIN AdminDecision ad ON b.BookingID=ad.BookingID
Where UserID= 'C111' AND ad.Decision IS NOT NULL AND b.Today=CAST(getdate() AS DATE)
        AND b.ScheduledDepartDateTime > CAST(Getdate() AS time(0))
```

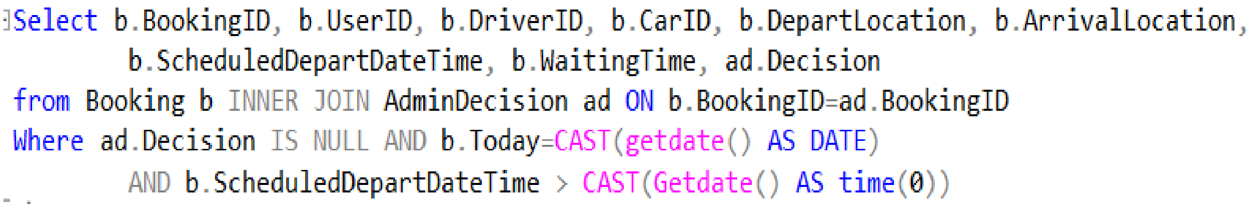


	BookingID	User...	DriverID	CarID	DepartLocation	ArrivalLocation	ScheduledDepartDateTime	WaitingTime	Decision
1	1000	C111	D111	T001	SU	Westcott St	23:50:00	12:00:00	Y

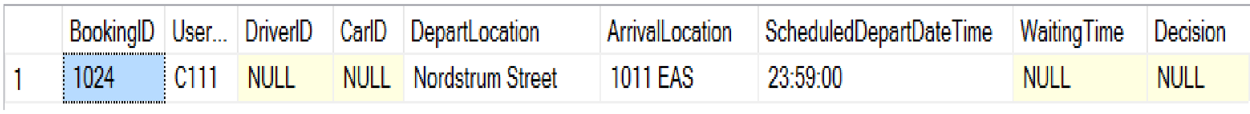
2. Why DPS Administrators query the database

- **DPS administrators query the database to manage and process user bookings**
  Administrators can monitor the bookings by reviewing the ‘BookingID’ and analyze the feasibility and authenticity of the booking and eventually can make decisions to either accept or reject the user booking request

Before making a decision:

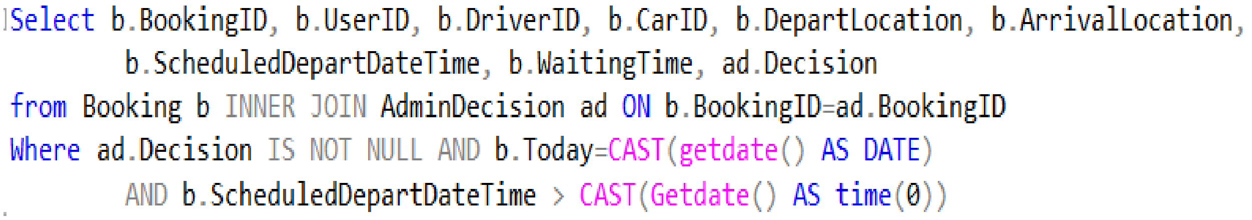


```
Select b.BookingID, b.UserID, b.DriverID, b.CarID, b.DepartLocation, b.ArrivalLocation,
        b.ScheduledDepartDateTime, b.WaitingTime, ad.Decision
from Booking b INNER JOIN AdminDecision ad ON b.BookingID=ad.BookingID
Where ad.Decision IS NULL AND b.Today=CAST(getdate() AS DATE)
        AND b.ScheduledDepartDateTime > CAST(Getdate() AS time(0))
```

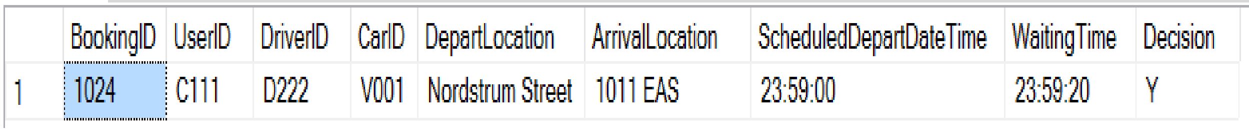


	BookingID	User...	DriverID	CarID	DepartLocation	ArrivalLocation	ScheduledDepartDateTime	WaitingTime	Decision
1	1024	C111	NULL	NULL	Nordstrum Street	1011 EAS	23:59:00	NULL	NULL

After making a decision:

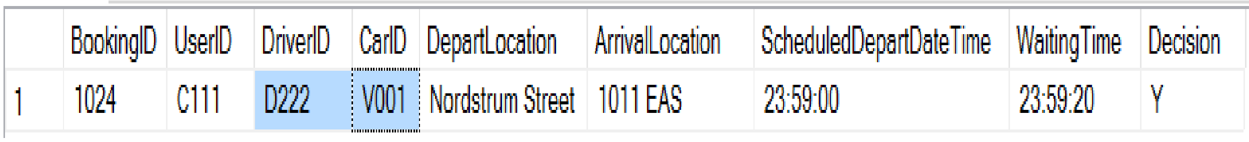


```
Select b.BookingID, b.UserID, b.DriverID, b.CarID, b.DepartLocation, b.ArrivalLocation,
        b.ScheduledDepartDateTime, b.WaitingTime, ad.Decision
from Booking b INNER JOIN AdminDecision ad ON b.BookingID=ad.BookingID
Where ad.Decision IS NOT NULL AND b.Today=CAST(getdate() AS DATE)
        AND b.ScheduledDepartDateTime > CAST(Getdate() AS time(0))
```



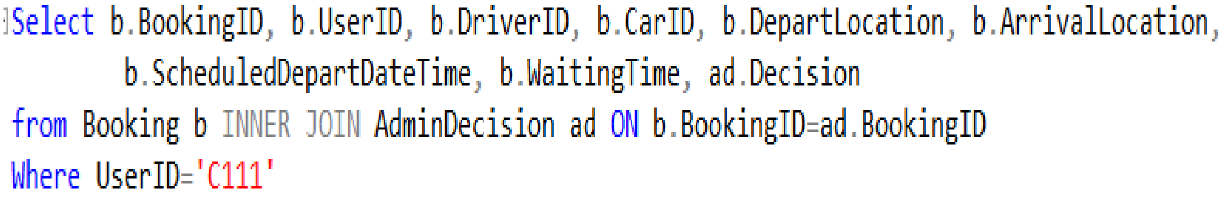
	BookingID	UserID	DriverID	CarID	DepartLocation	ArrivalLocation	ScheduledDepartDateTime	WaitingTime	Decision
1	1024	C111	D222	V001	Nordstrum Street	1011 EAS	23:59:00	23:59:20	Y

- **Administrators can allot a driver and a car for a ‘BookingID’**
  Administrator once approved the booking then has to allot a driver and a car to that BookingID. The admin does so by entering the DriverID and CarID for the specific BookingID.

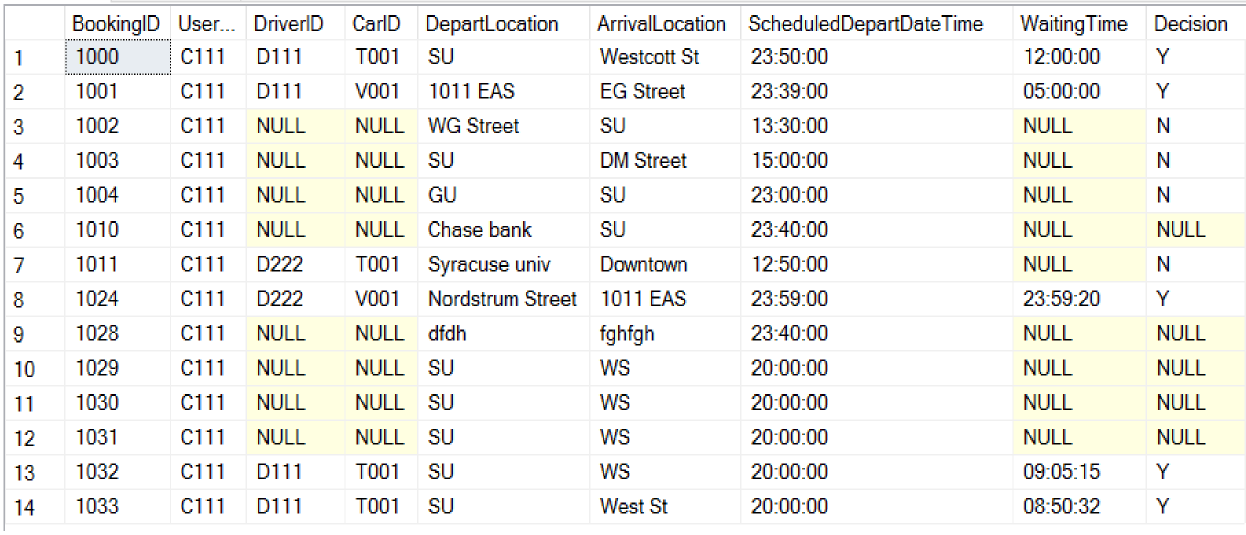


	BookingID	UserID	DriverID	CarID	DepartLocation	ArrivalLocation	ScheduledDepartDateTime	WaitingTime	Decision
1	1024	C111	D222	V001	Nordstrum Street	1011 EAS	23:59:00	23:59:20	Y

- **Admin can search for user and booking information**
  Admin has the privilege to search for user and booking records for various purposes. Admin has to enter the UserID or BookingID for retrieving the specific user or booking information. Moreover, admin can generate reports on the bookings that were approved or disapproved by the admin for the day.
  Search by UserID: (Example: UserID: C111)

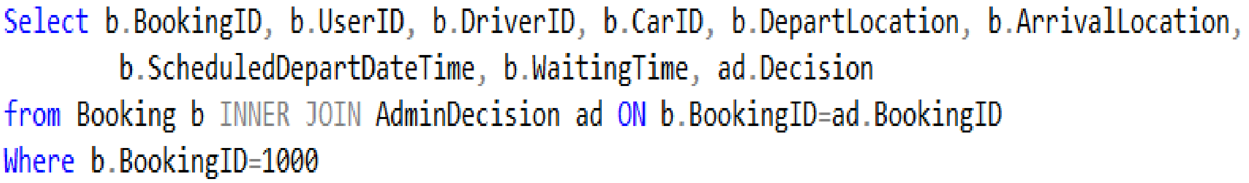


```
Select b.BookingID, b.UserID, b.DriverID, b.CarID, b.DepartLocation, b.ArrivalLocation,
        b.ScheduledDepartDateTime, b.WaitingTime, ad.Decision
from Booking b INNER JOIN AdminDecision ad ON b.BookingID=ad.BookingID
Where UserID='C111'
```

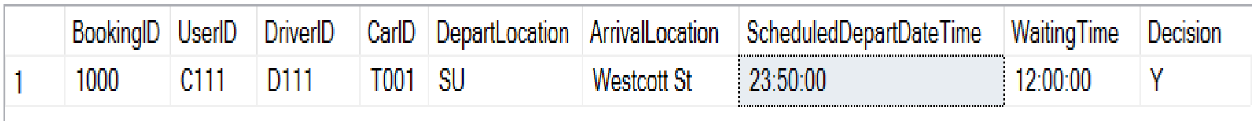


	BookingID	User...	DriverID	CarID	DepartLocation	ArrivalLocation	ScheduledDepartDateTime	WaitingTime	Decision
1	1000	C111	D111	T001	SU	Westcott St	23:50:00	12:00:00	Y
2	1001	C111	D111	V001	1011 EAS	EG Street	23:39:00	05:00:00	Y
3	1002	C111	NULL	NULL	WG Street	SU	13:30:00	NULL	N
4	1003	C111	NULL	NULL	SU	DM Street	15:00:00	NULL	N
5	1004	C111	NULL	NULL	GU	SU	23:00:00	NULL	N
6	1010	C111	NULL	NULL	Chase bank	SU	23:40:00	NULL	NULL
7	1011	C111	D222	T001	Syracuse univ	Downtown	12:50:00	NULL	N
8	1024	C111	D222	V001	Nordstrum Street	1011 EAS	23:59:00	23:59:20	Y
9	1028	C111	NULL	NULL	dfdh	fghfgh	23:40:00	NULL	NULL
10	1029	C111	NULL	NULL	SU	WS	20:00:00	NULL	NULL
11	1030	C111	NULL	NULL	SU	WS	20:00:00	NULL	NULL
12	1031	C111	NULL	NULL	SU	WS	20:00:00	NULL	NULL
13	1032	C111	D111	T001	SU	WS	20:00:00	09:05:15	Y
14	1033	C111	D111	T001	SU	West St	20:00:00	08:50:32	Y

Search by BookingID: (Example: UserID: 1000)



```
Select b.BookingID, b.UserID, b.DriverID, b.CarID, b.DepartLocation, b.ArrivalLocation,
        b.ScheduledDepartDateTime, b.WaitingTime, ad.Decision
from Booking b INNER JOIN AdminDecision ad ON b.BookingID=ad.BookingID
Where b.BookingID=1000
```



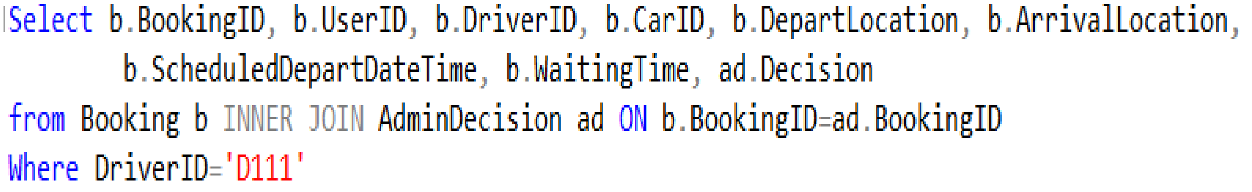
	BookingID	UserID	DriverID	CarID	DepartLocation	ArrivalLocation	ScheduledDepartDateTime	WaitingTime	Decision
1	1000	C111	D111	T001	SU	Westcott St	23:50:00	12:00:00	Y

3. Why the Drivers query the database

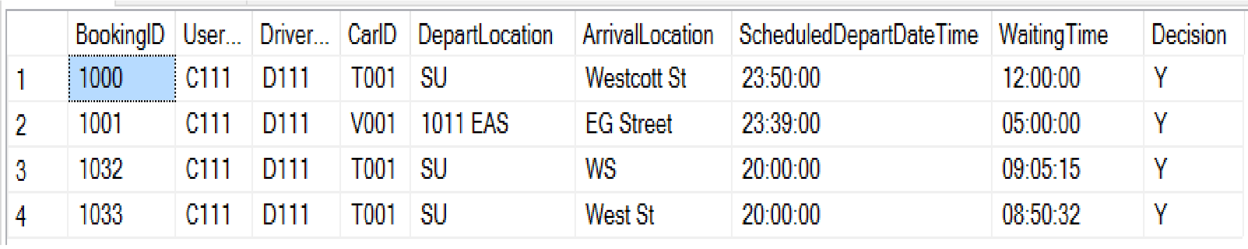
   - **Drivers can query the database to know whom they have to escort**

     Drivers can login using their username and password and know the BookingID which has been allotted to them by the DPS escort service. They can also view their past bookings and people that they have escorted in the past.

Example: DriverID = 'D111'



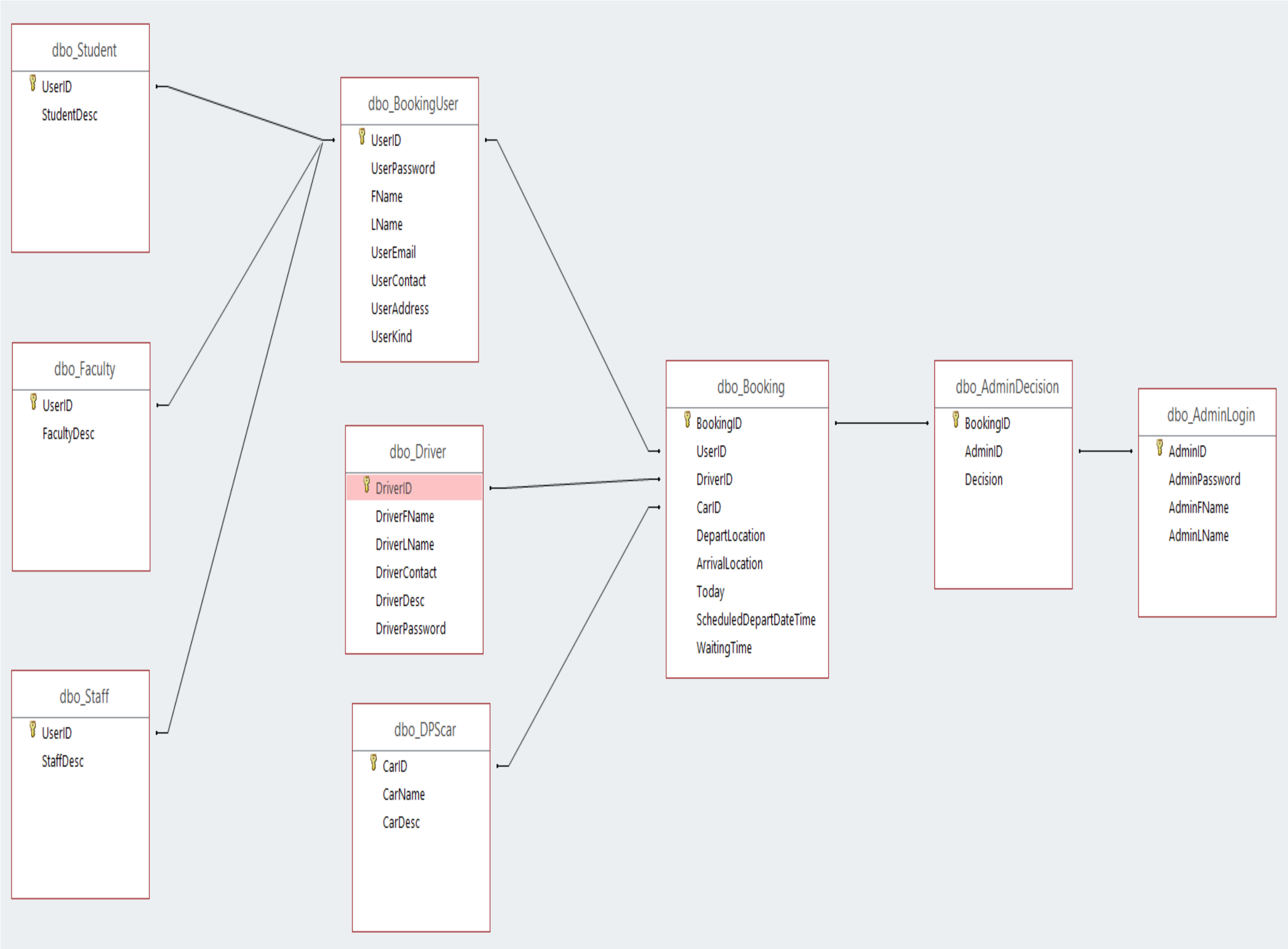
```
Select b.BookingID, b.UserID, b.DriverID, b.CarID, b.DepartLocation, b.ArrivalLocation,
        b.ScheduledDepartDateTime, b.WaitingTime, ad.Decision
from Booking b INNER JOIN AdminDecision ad ON b.BookingID=ad.BookingID
Where DriverID='D111'
```



	BookingID	User...	Driver...	CarID	DepartLocation	ArrivalLocation	ScheduledDepartDateTime	WaitingTime	Decision
1	1000	C111	D111	T001	SU	Westcott St	23:50:00	12:00:00	Y
2	1001	C111	D111	V001	1011 EAS	EG Street	23:39:00	05:00:00	Y
3	1032	C111	D111	T001	SU	WS	20:00:00	09:05:15	Y
4	1033	C111	D111	T001	SU	West St	20:00:00	08:50:32	Y

MS-ACCESS RELATIONSHIP DIAGRAM:

The relationship diagram is as under:

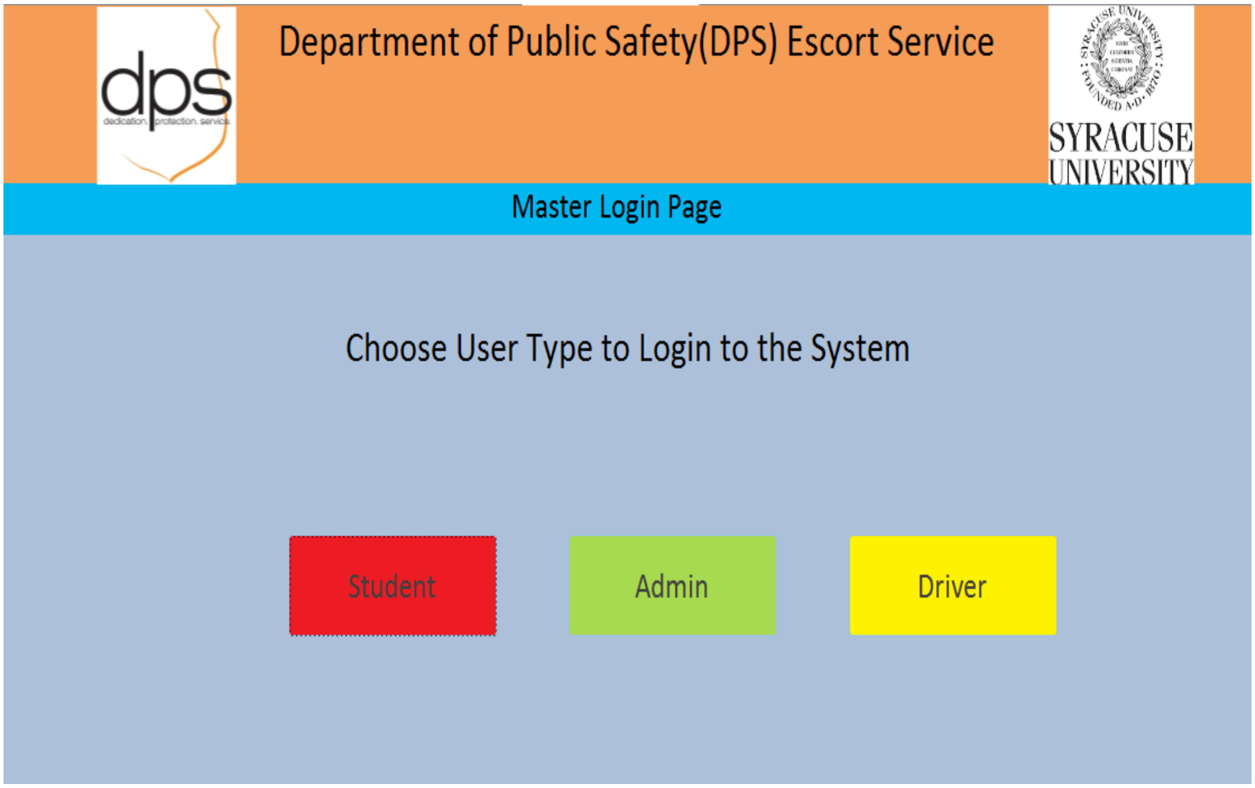


FORMS:

DPS service database application has three types of users:

1. Students/Faculty/Staff
2. Administrator
3. Driver

Master Login Form



Department of Public Safety(DPS) Escort Service

Master Login Page

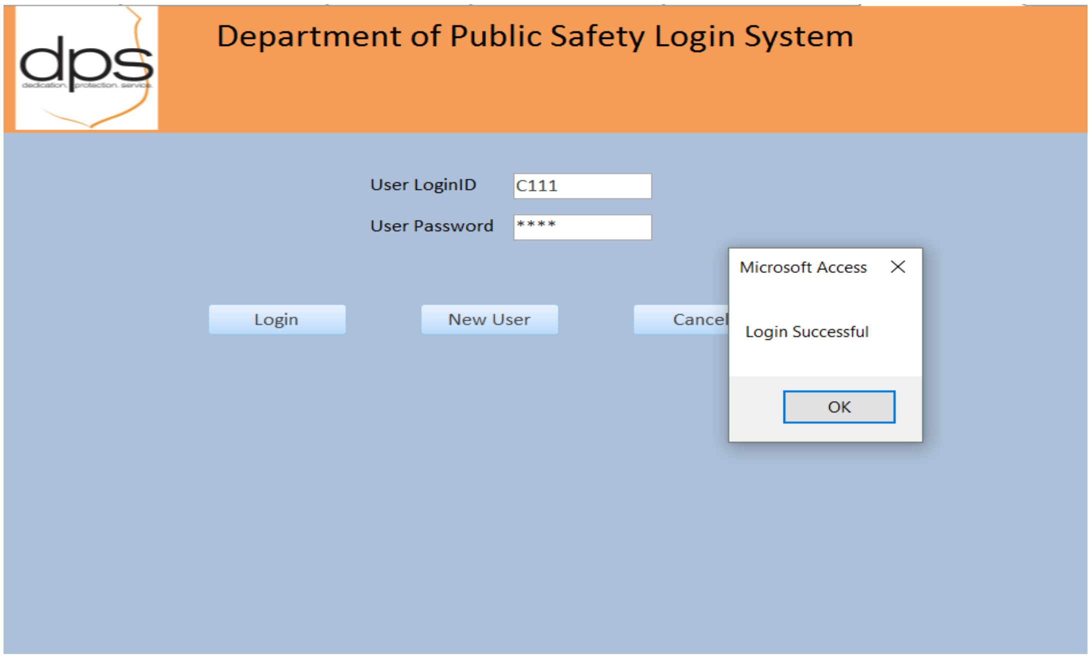
Choose User Type to Login to the System

The above master login page enables users to choose their type of login.

1. Students/Faculty/Staff

Login Page

The below form is a login form for students, staff and faculty and it provides an interface for users to use DPS services.



Department of Public Safety Login System

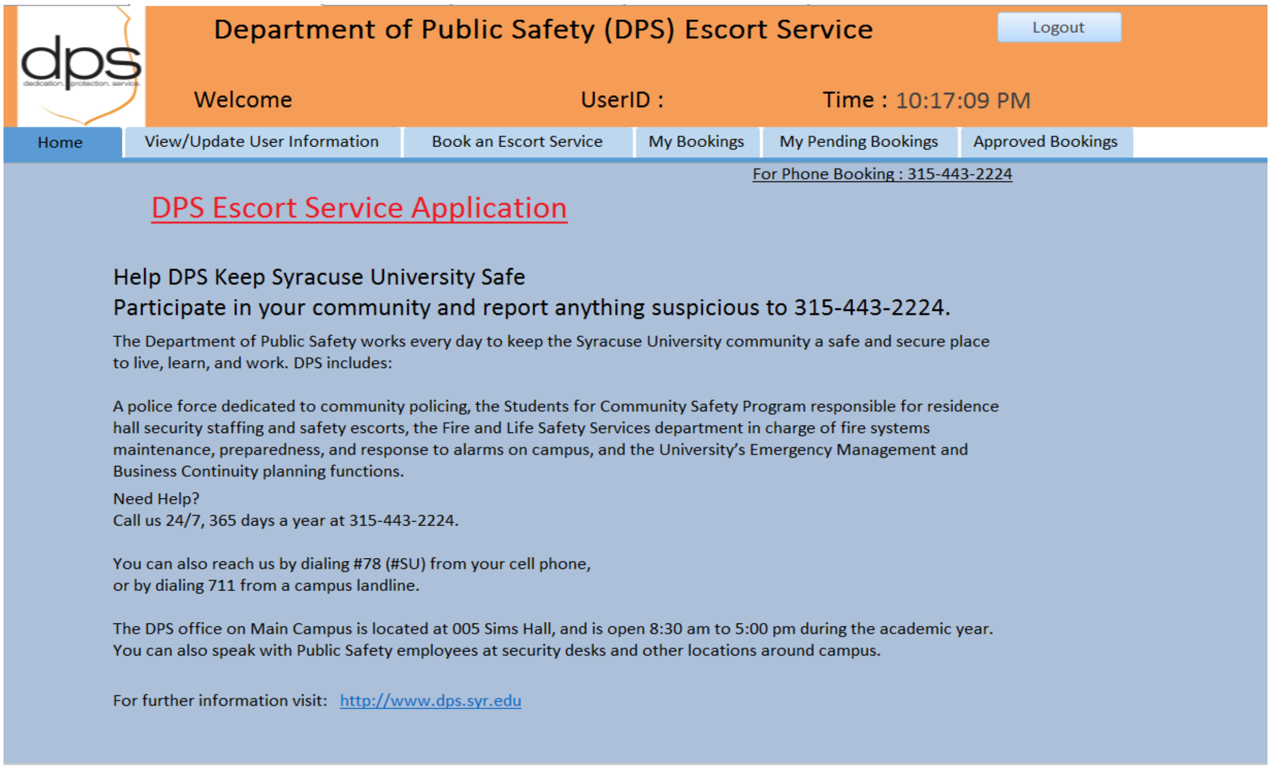
User LoginID C111

User Password ****

Login Successful

User Panel (Homepage):

The below form is a user panel for students, staff and faculty. Also, it provides an overview and the necessary information about the DPS service.


dps
dedication. protection. service.
Department of Public Safety (DPS) Escort Service
Logout
Welcome
UserID :
Time : 10:17:09 PM
Home
View/Update User Information
Book an Escort Service
My Bookings
My Pending Bookings
Approved Bookings
For Phone Booking : 315-443-2224

DPS Escort Service Application

Help DPS Keep Syracuse University Safe
Participate in your community and report anything suspicious to 315-443-2224.

The Department of Public Safety works every day to keep the Syracuse University community a safe and secure place to live, learn, and work. DPS includes:

A police force dedicated to community policing, the Students for Community Safety Program responsible for residence hall security staffing and safety escorts, the Fire and Life Safety Services department in charge of fire systems maintenance, preparedness, and response to alarms on campus, and the University's Emergency Management and Business Continuity planning functions.

Need Help?
Call us 24/7, 365 days a year at 315-443-2224.

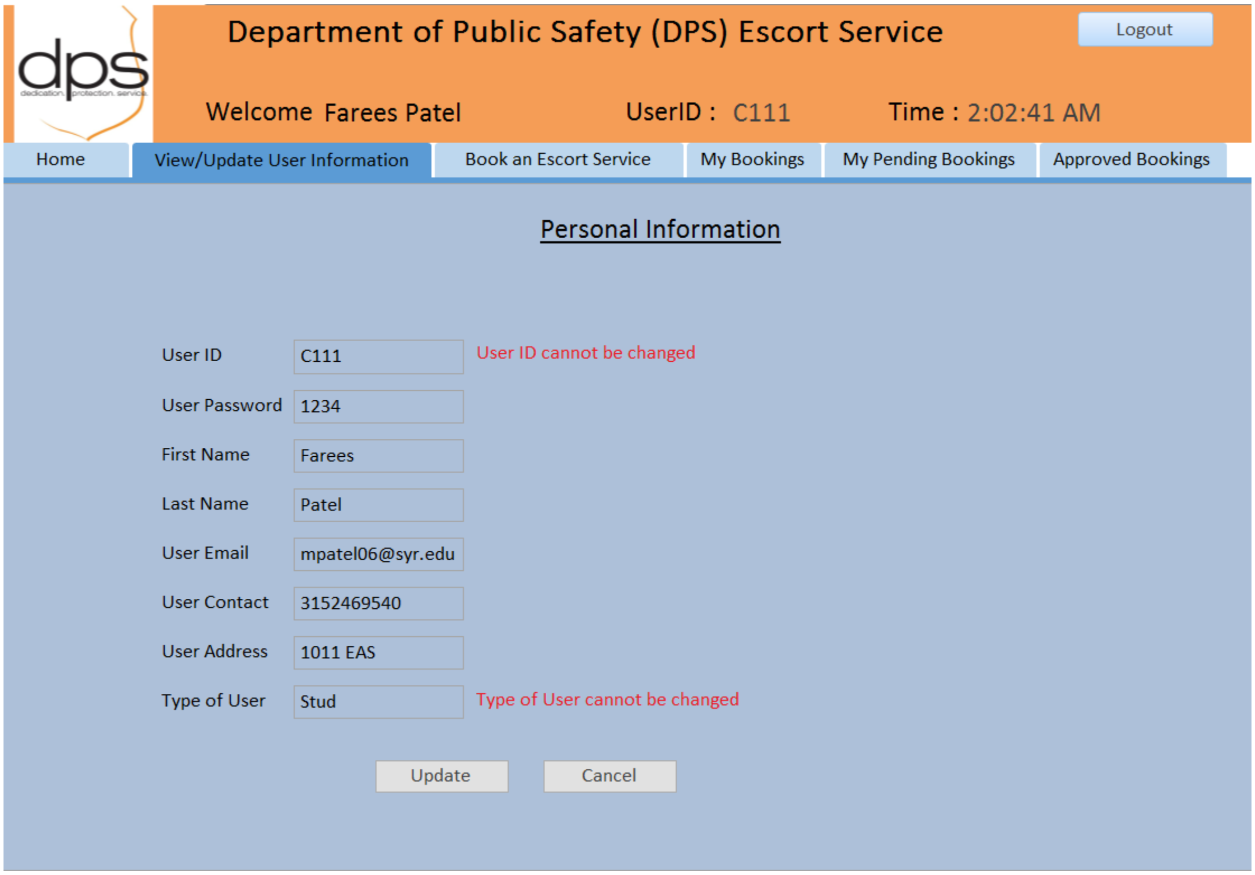
You can also reach us by dialing #78 (#SU) from your cell phone,
or by dialing 711 from a campus landline.

The DPS office on Main Campus is located at 005 Sims Hall, and is open 8:30 am to 5:00 pm during the academic year.
You can also speak with Public Safety employees at security desks and other locations around campus.

For further information visit: http://www.dps.syr.edu

View/Update and 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.



Department of Public Safety (DPS) Escort Service

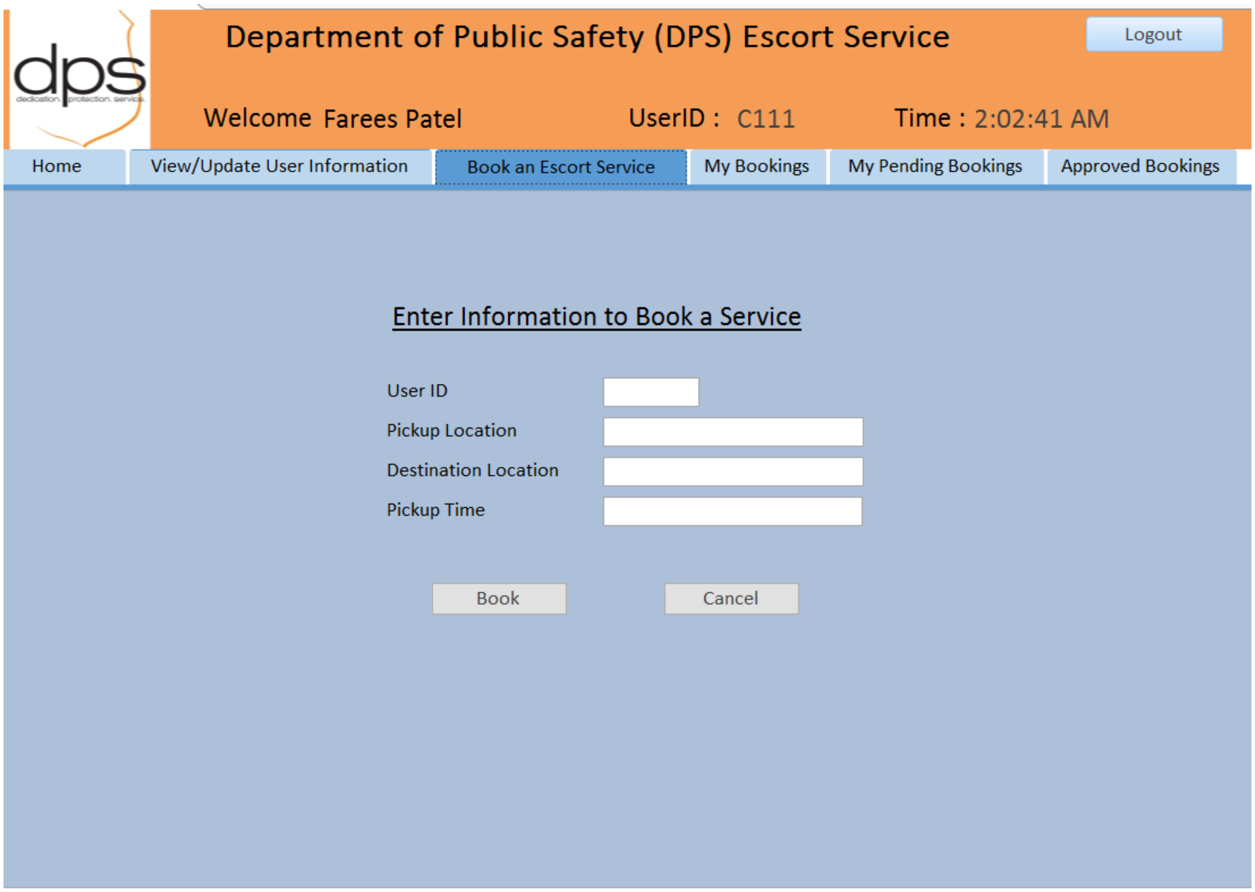
Welcome Farees Patel UserID : C111 Time : 2:02:41 AM

Personal Information

User ID	C111	User ID cannot be changed
User Password	1234	
First Name	Farees	
Last Name	Patel	
User Email	mpatel06@syr.edu	
User Contact	3152469540	
User Address	1011 EAS	
Type of User	Stud	Type of User cannot be changed

Book an escort service:

The below form enables users (students, staff or faculty) to book an escort service. This form makes it easy for users to book an escort service because the user just has to mention his/her respective user along with the pickup location, destination location and the schedule pickup time. Entering all information and booking will send the booking information to the admin for approval/disapproval.



Department of Public Safety (DPS) Escort Service

Welcome Farees Patel UserID : C111 Time : 2:02:41 AM

Enter Information to Book a Service

User ID

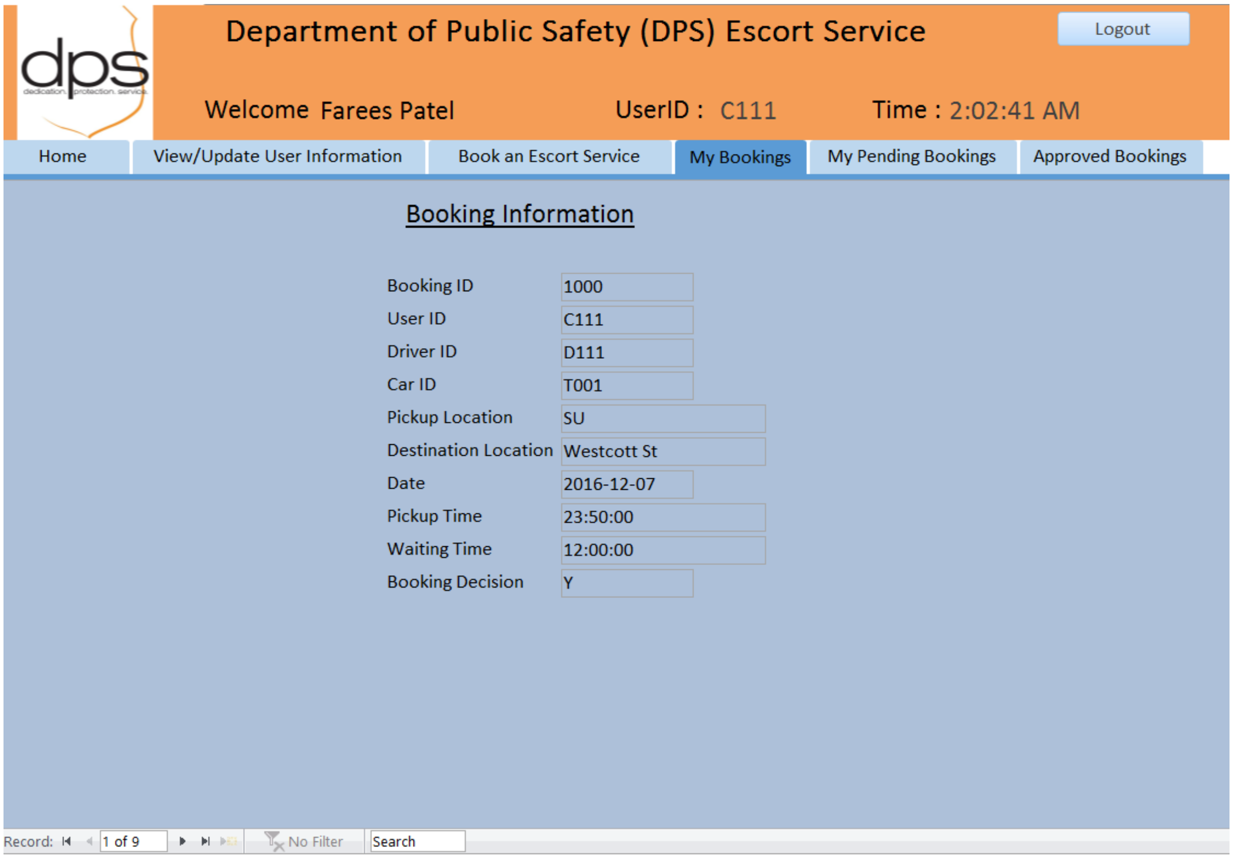
Pickup Location

Destination Location

Pickup Time

My Bookings:

The below form will enlist all the bookings that the specific users requested in the past. Since, my application has a high level of abstraction, the bookings tab will only contain booking record that is related to the specific user who has login to the system.



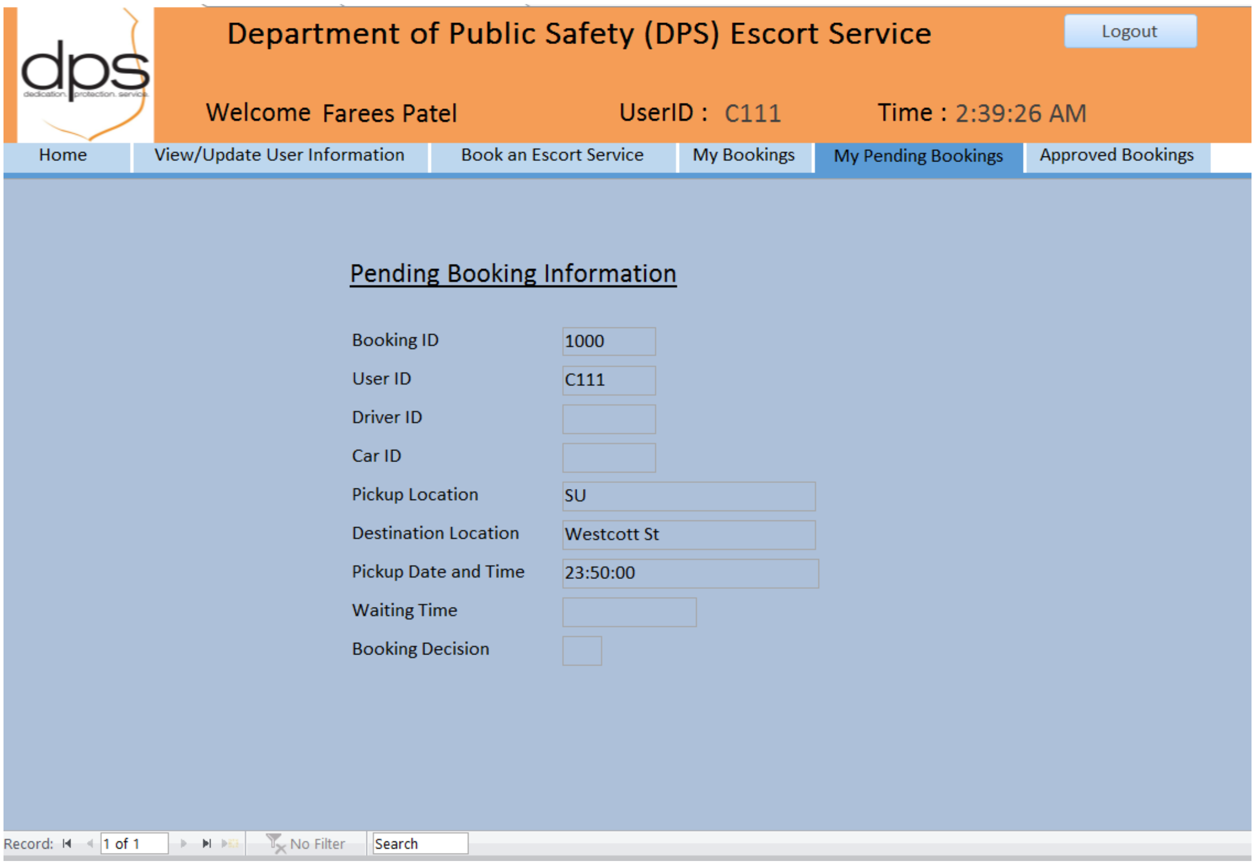
Welcome Farees Patel UserID : C111 Time : 2:02:41 AM

Booking Information

Booking ID	1000
User ID	C111
Driver ID	D111
Car ID	T001
Pickup Location	SU
Destination Location	Westcott St
Date	2016-12-07
Pickup Time	23:50:00
Waiting Time	12:00:00
Booking Decision	Y

My Pending Bookings:

The pending bookings tab will enlist all the bookings that are under processing. All the bookings that user has requested but still has not been either approved or disapproved by the administrator. Certain fields of the booking records are void for example the driver ID and car ID are void because only when the booking is approved, the admin is going to allot a driver and a car to the booking hence entering the respective Driver ID and the Car ID.

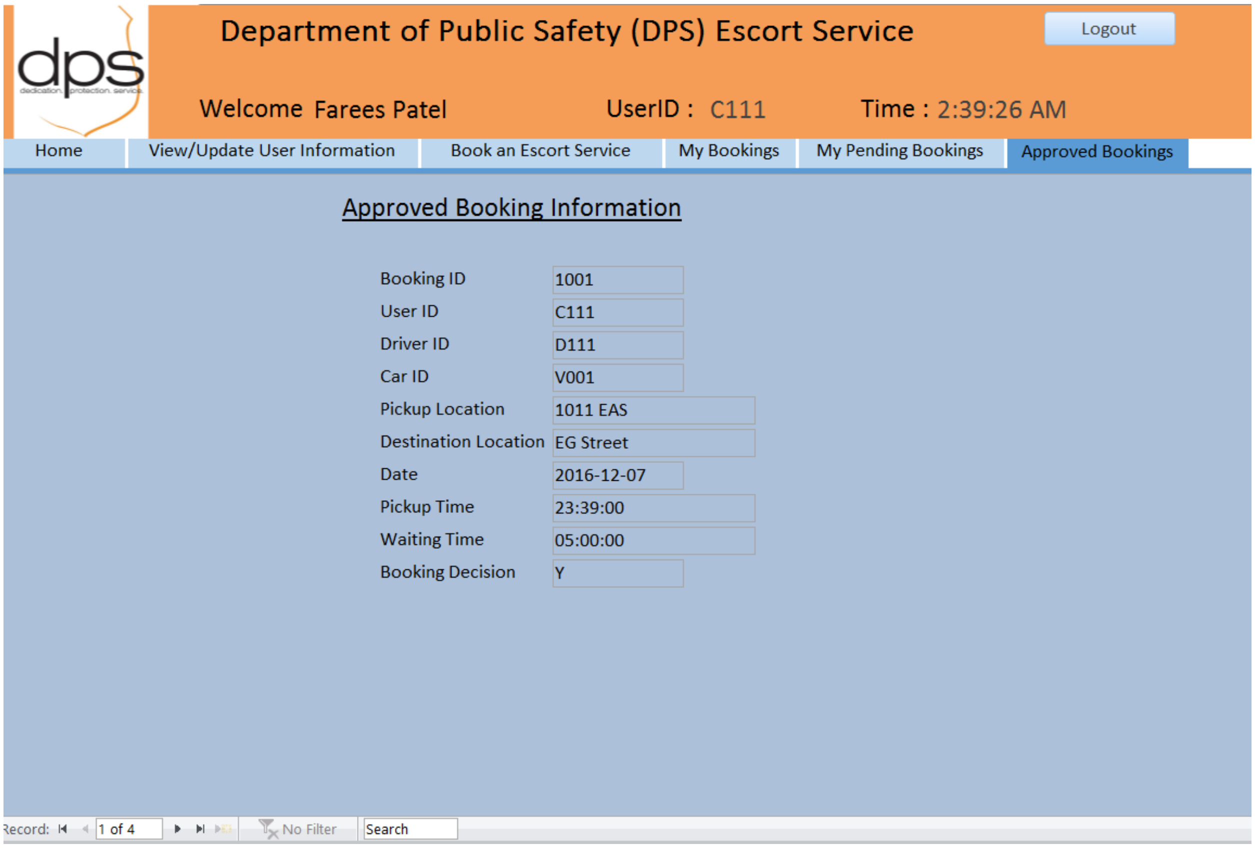


Pending Booking Information

Booking ID	1000
User ID	C111
Driver ID	
Car ID	
Pickup Location	SU
Destination Location	Westcott St
Pickup Date and Time	23:50:00
Waiting Time	
Booking Decision	

Approved Bookings:

The approved bookings tab will enlist all the bookings that are processed and are approved by the administrator. Every fields of the booking record have a significant value for example the driver ID and car ID are present because the booking is approved and the admin is going to allot a driver and a car to the booking request. The waiting time for the booking is automatically calculated but the admin can also manually enter the waiting time if there is any delay due to the unavailability of either driver or the car.



Welcome Farees Patel UserID : C111 Time : 2:39:26 AM

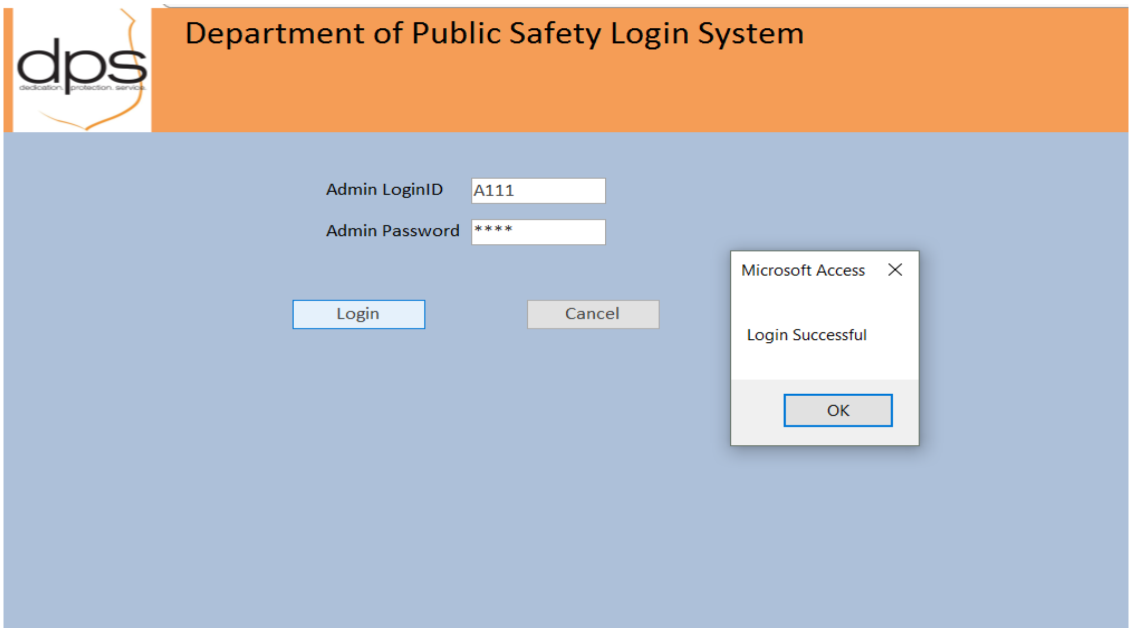
Approved Booking Information

Booking ID	1001
User ID	C111
Driver ID	D111
Car ID	V001
Pickup Location	1011 EAS
Destination Location	EG Street
Date	2016-12-07
Pickup Time	23:39:00
Waiting Time	05:00:00
Booking Decision	Y

2. Administrator

Admin Login Page:

The below form is a login form for students, staff and faculty and it provides an interface for administrators to use evaluate bookings for DPS escort services.



Department of Public Safety Login System

Admin LoginID A111

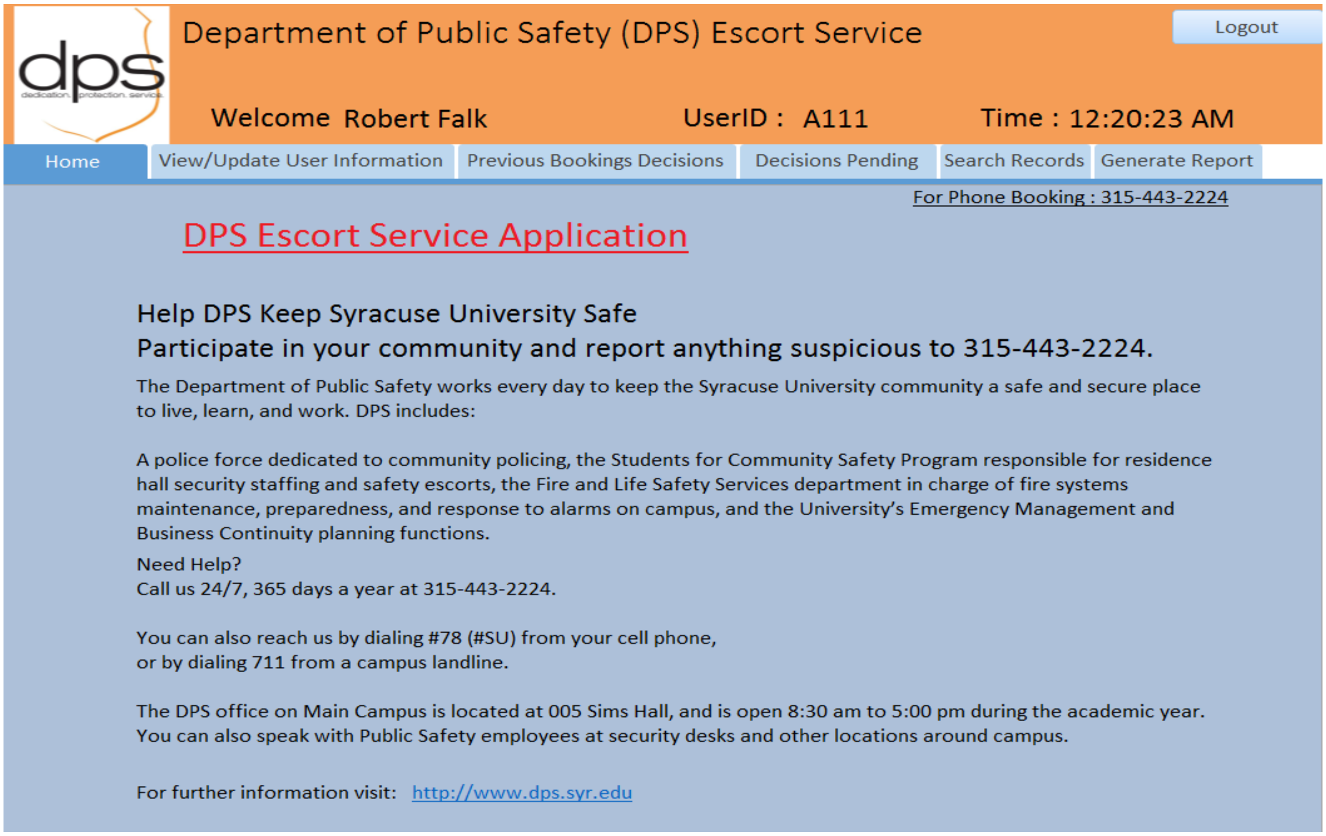
Admin Password ****

Microsoft Access

Login Successful

User Panel (Homepage):

The below form is a user panel for the admin. Also, it provides an overview and the necessary information about the DPS service.


dps
dedication. protection. service.
Logout
Home
View/Update User Information
Previous Bookings Decisions
Decisions Pending
Search Records
Generate Report

Department of Public Safety (DPS) Escort Service

Welcome Robert Falk UserID : A111 Time : 12:20:23 AM

For Phone Booking : 315-443-2224

DPS Escort Service Application

Help DPS Keep Syracuse University Safe
Participate in your community and report anything suspicious to 315-443-2224.

The Department of Public Safety works every day to keep the Syracuse University community a safe and secure place to live, learn, and work. DPS includes:

A police force dedicated to community policing, the Students for Community Safety Program responsible for residence hall security staffing and safety escorts, the Fire and Life Safety Services department in charge of fire systems maintenance, preparedness, and response to alarms on campus, and the University's Emergency Management and Business Continuity planning functions.

Need Help?
Call us 24/7, 365 days a year at 315-443-2224.

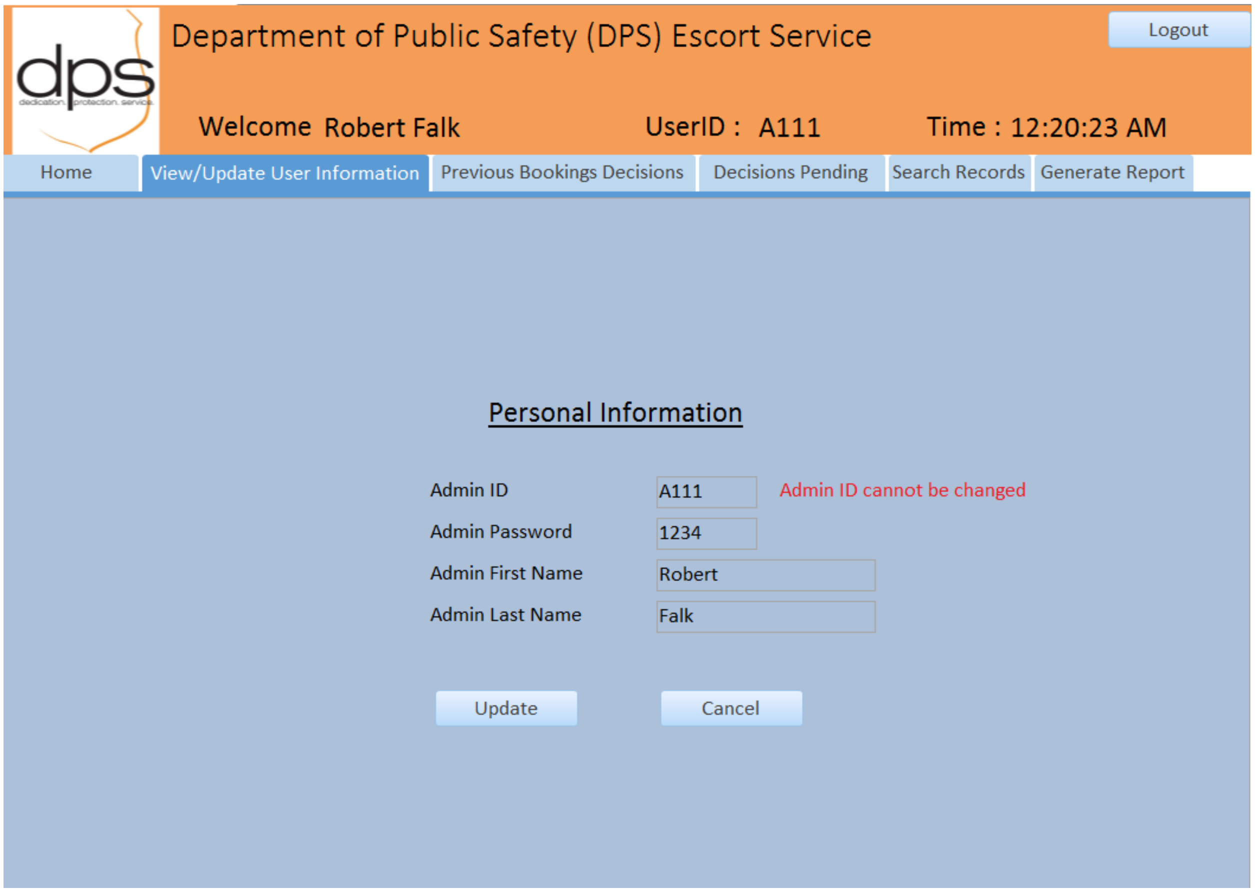
You can also reach us by dialing #78 (#SU) from your cell phone,
or by dialing 711 from a campus landline.

The DPS office on Main Campus is located at 005 Sims Hall, and is open 8:30 am to 5:00 pm during the academic year. You can also speak with Public Safety employees at security desks and other locations around campus.

For further information visit: http://www.dps.syr.edu

View/Update and User Information:

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



Department of Public Safety (DPS) Escort Service

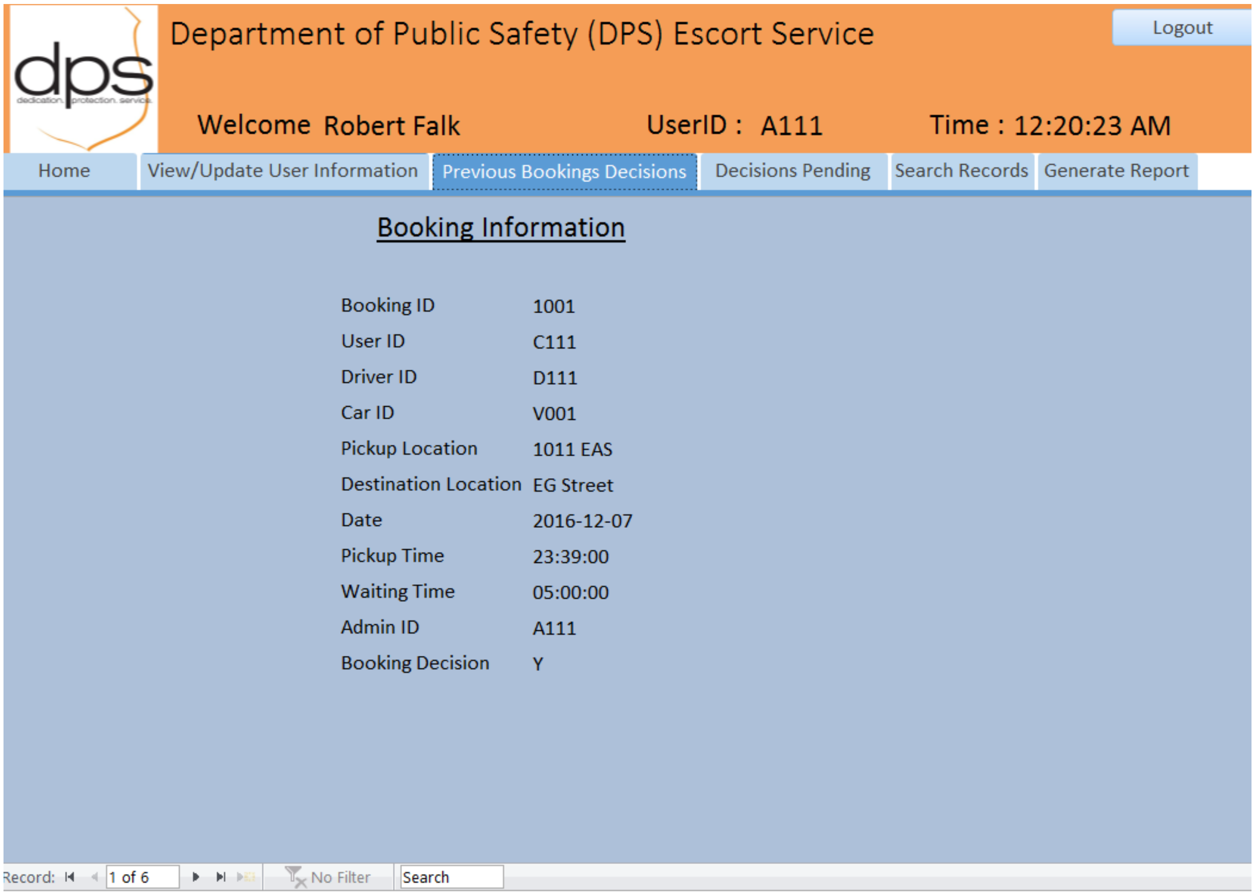
Welcome Robert Falk UserID : A111 Time : 12:20:23 AM

Personal Information

Admin ID	A111	Admin ID cannot be changed
Admin Password	1234	
Admin First Name	Robert	
Admin Last Name	Falk	

Previous Bookings Decisions:

The below form will enlist all the bookings that the specific administrator has either approved or disapproved in the past. Since, my application has a high level of abstraction, the ‘previous bookings tab’ will only contain booking record that is related to the specific administrator who has login to the system.

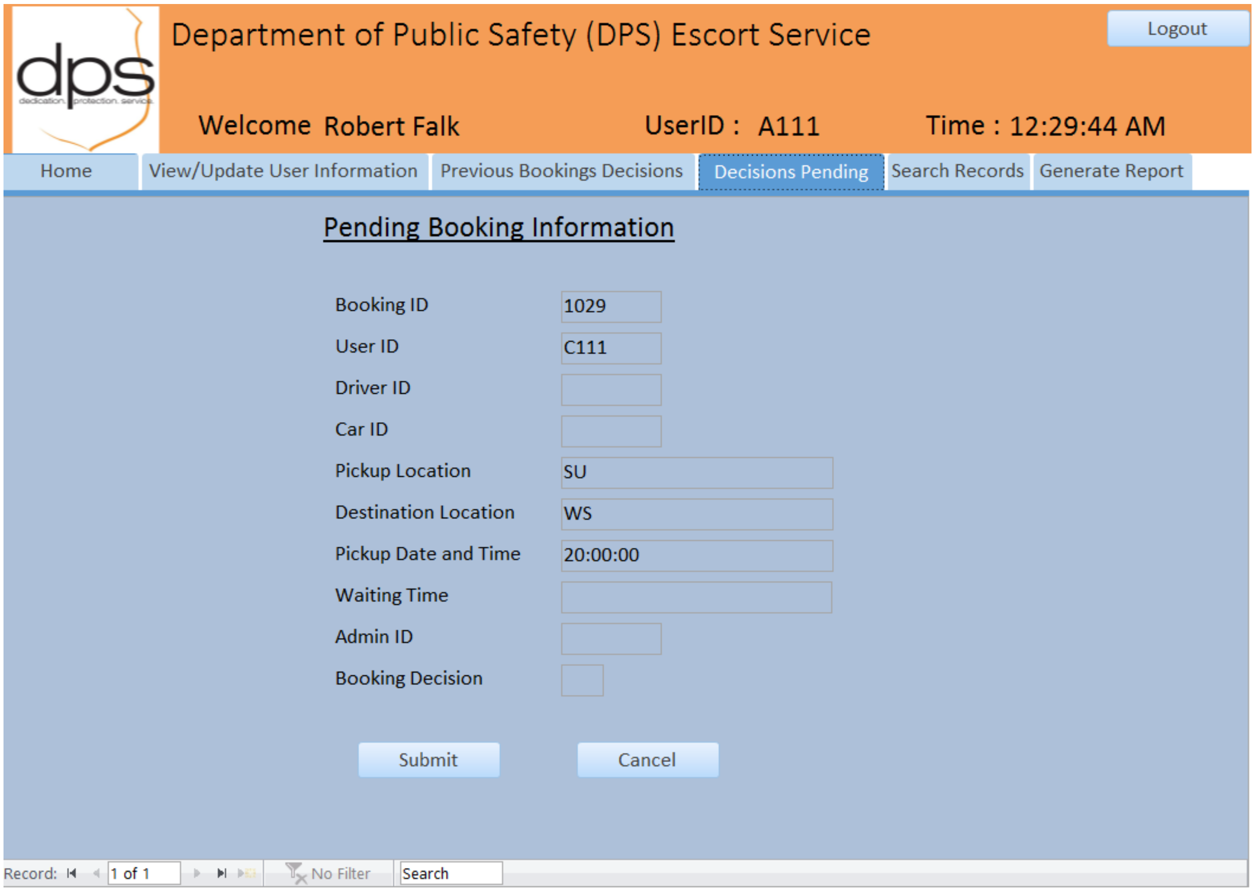


Booking Information

Booking ID	1001
User ID	C111
Driver ID	D111
Car ID	V001
Pickup Location	1011 EAS
Destination Location	EG Street
Date	2016-12-07
Pickup Time	23:39:00
Waiting Time	05:00:00
Admin ID	A111
Booking Decision	Y

Decisions Pending:

The ‘decisions pending’ tab will enlist all the bookings that the admin has to process. All the bookings that user has requested but still has not been either approved or disapproved by the administrator will be enlisted under this tab. Here, certain fields of the booking records are void for example the driver ID and car ID are void because only when the booking is approved, the admin is going to allot a driver and a car to the booking hence entering the respective Driver ID and the Car ID.



Department of Public Safety (DPS) Escort Service

Welcome Robert Falk UserID : A111 Time : 12:29:44 AM

Pending Booking Information

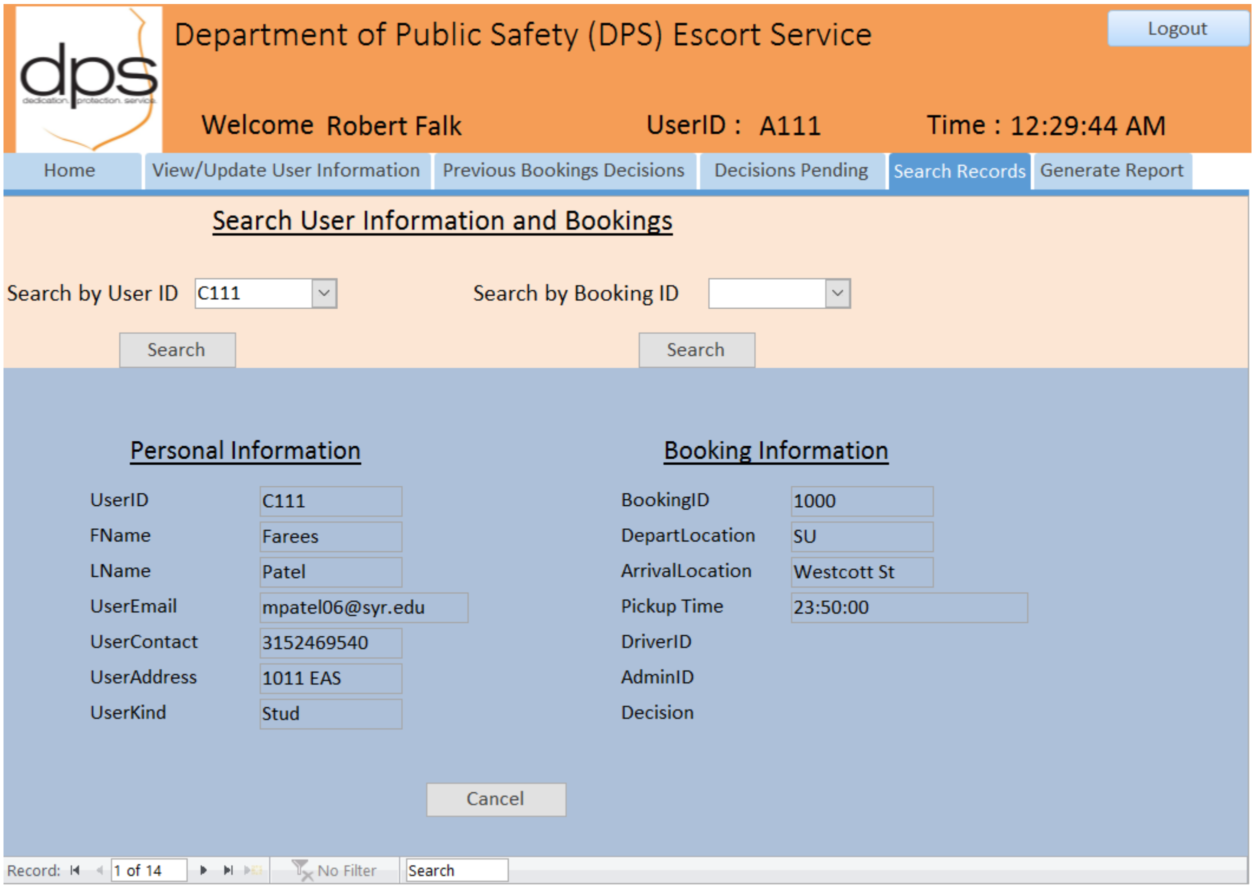
Booking ID	1029
User ID	C111
Driver ID	
Car ID	
Pickup Location	SU
Destination Location	WS
Pickup Date and Time	20:00:00
Waiting Time	
Admin ID	
Booking Decision	

Record: 1 of 1

Search Records:

This form will enable the administrator to search for any specific user by their unique User ID to get the user's personal information or bookings. It will enlist all the bookings and the personal information that is ties to the specific User ID. Moreover, the admin can also search for any specific booking by entering the unique Booking ID. This will enlist all the booking information and the details about the user who has requested that booking.

Search by User ID:



Department of Public Safety (DPS) Escort Service

Welcome Robert Falk UserID : A111 Time : 12:29:44 AM

Search User Information and Bookings

Search by User ID C111 Search by Booking ID

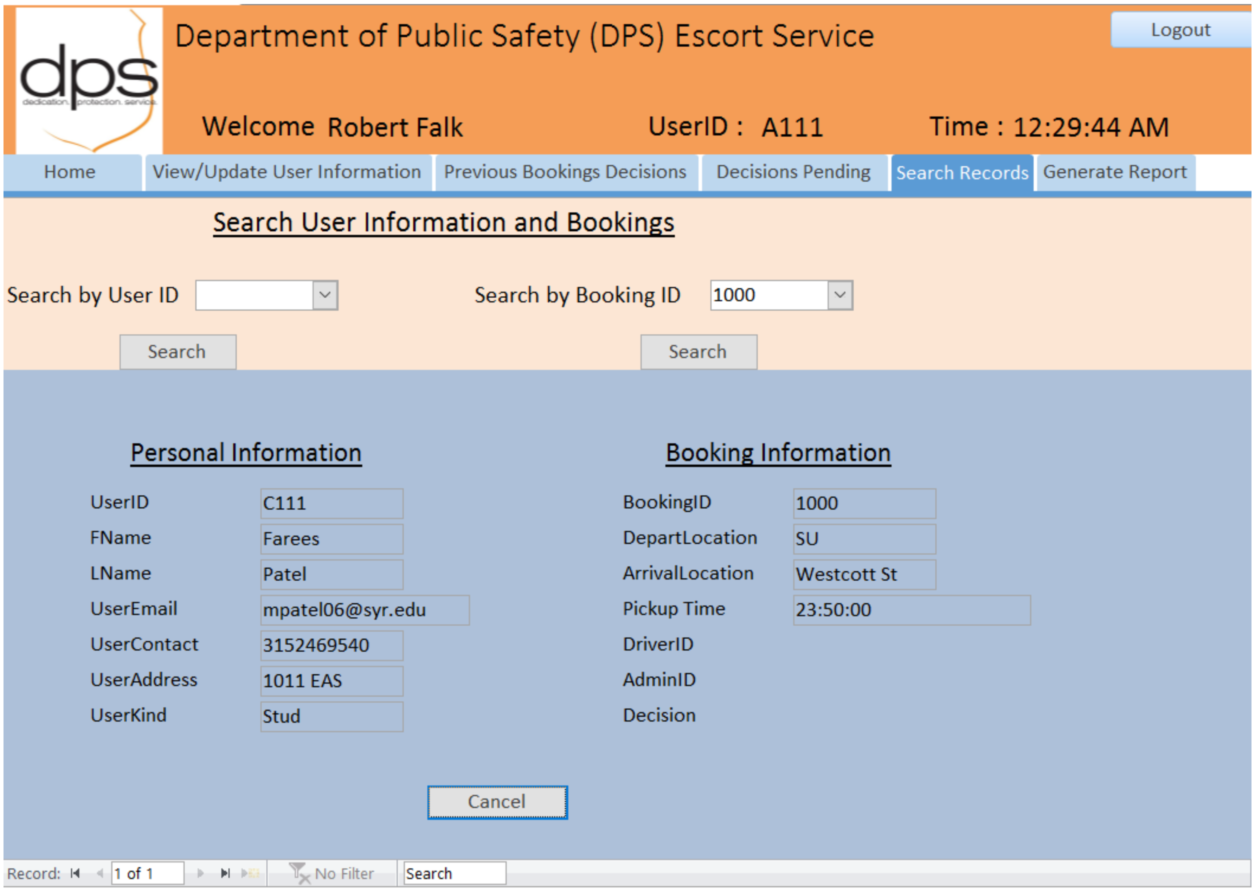
Personal Information

UserID	C111
FName	Farees
LName	Patel
UserEmail	mpatel06@syr.edu
UserContact	3152469540
UserAddress	1011 EAS
UserKind	Stud

Booking Information

BookingID	1000
DepartLocation	SU
ArrivalLocation	Westcott St
Pickup Time	23:50:00
DriverID	
AdminID	
Decision	

Search by Booking ID:



Department of Public Safety (DPS) Escort Service

Welcome Robert Falk UserID : A111 Time : 12:29:44 AM

Search User Information and Bookings

Search by User ID

Search by Booking ID 1000

Personal Information

UserID	C111
FName	Farees
LName	Patel
UserEmail	mpatel06@syr.edu
UserContact	3152469540
UserAddress	1011 EAS
UserKind	Stud

Booking Information

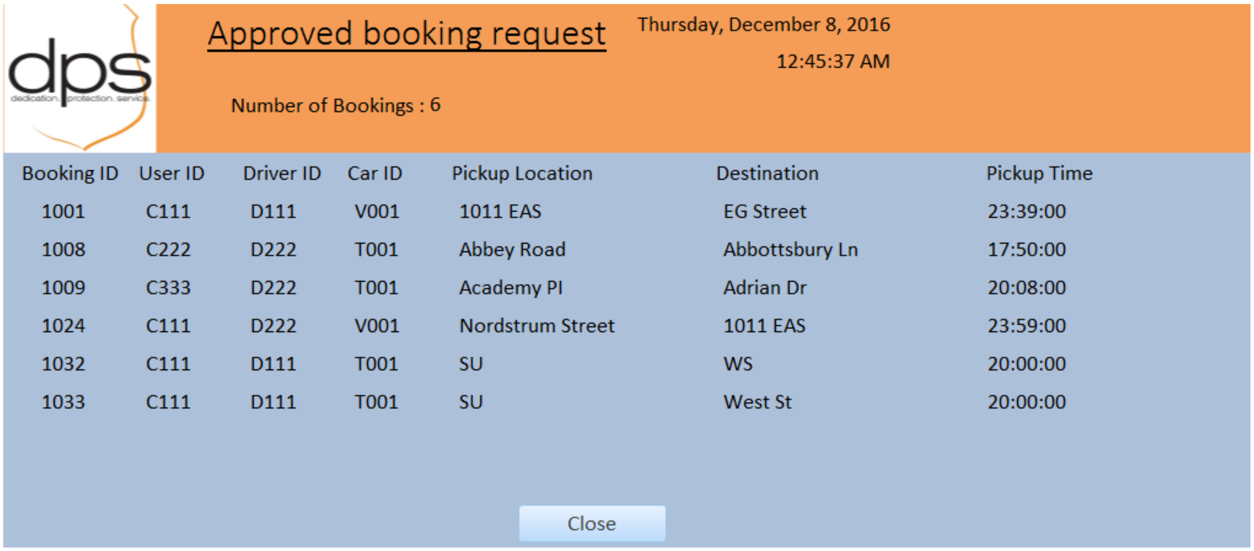
BookingID	1000
DepartLocation	SU
ArrivalLocation	Westcott St
Pickup Time	23:50:00
DriverID	
AdminID	
Decision	

Generate Reports:

This form will mention all the reports that the admin can generate. My application will help the admin to generate four reports. The first three reports uses aggregate function to evaluate the count of the users under various conditions.

(a) Report 1: Bookings approved for the day

This report enlists all the bookings for the day that were approved by the administrators.



Approved booking request

Thursday, December 8, 2016

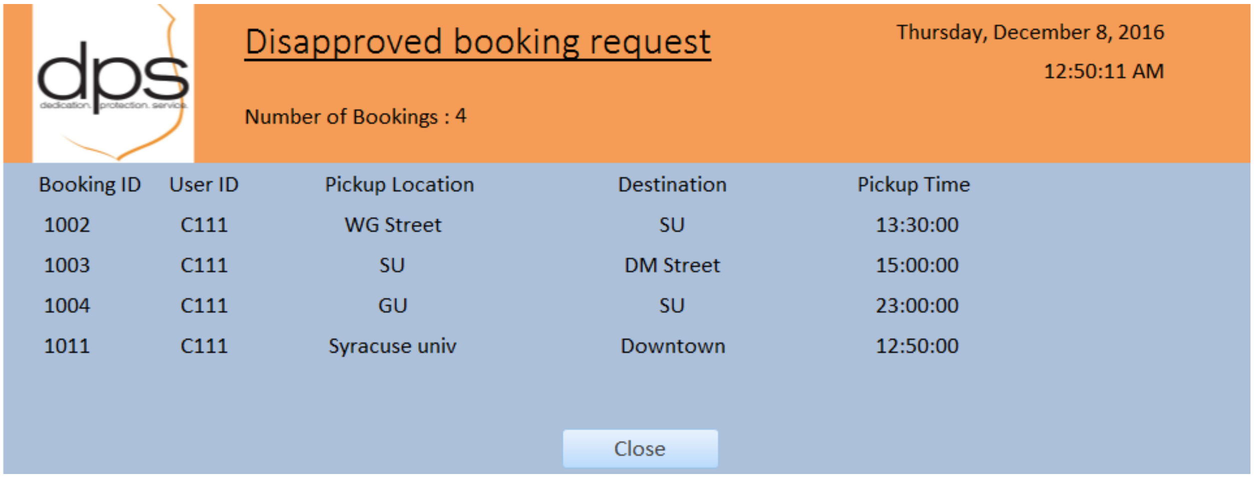
12:45:37 AM

Number of Bookings : 6

Booking ID	User ID	Driver ID	Car ID	Pickup Location	Destination	Pickup Time
1001	C111	D111	V001	1011 EAS	EG Street	23:39:00
1008	C222	D222	T001	Abbey Road	Abbottsbury Ln	17:50:00
1009	C333	D222	T001	Academy Pl	Adrian Dr	20:08:00
1024	C111	D222	V001	Nordstrum Street	1011 EAS	23:59:00
1032	C111	D111	T001	SU	WS	20:00:00
1033	C111	D111	T001	SU	West St	20:00:00

(b) Report 2: Bookings disapproved for the day

This report enlists all the bookings for the day that were disapproved by the administrators.



Disapproved booking request

Thursday, December 8, 2016

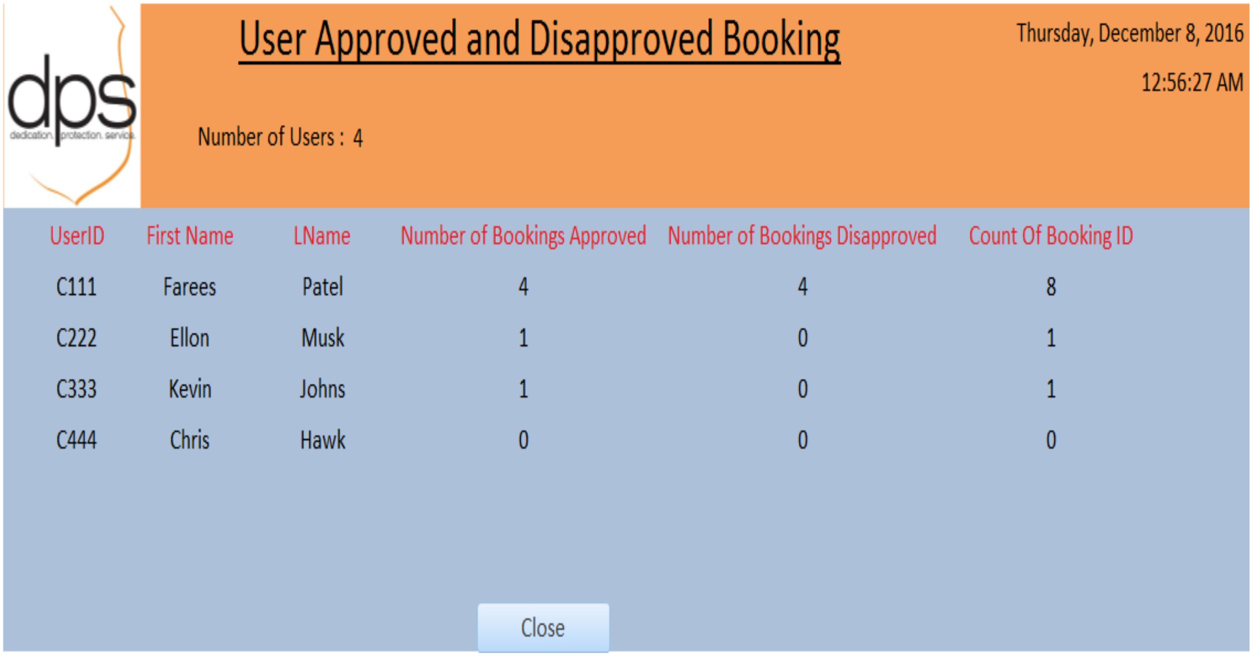
12:50:11 AM

Number of Bookings : 4

Booking ID	User ID	Pickup Location	Destination	Pickup Time
1002	C111	WG Street	SU	13:30:00
1003	C111	SU	DM Street	15:00:00
1004	C111	GU	SU	23:00:00
1011	C111	Syracuse univ	Downtown	12:50:00

(c) Report 3: User approved and disapproved Booking

This report enlists all the count for the bookings for the day that were approved and disapproved by the administrators.



User Approved and Disapproved Booking

Thursday, December 8, 2016

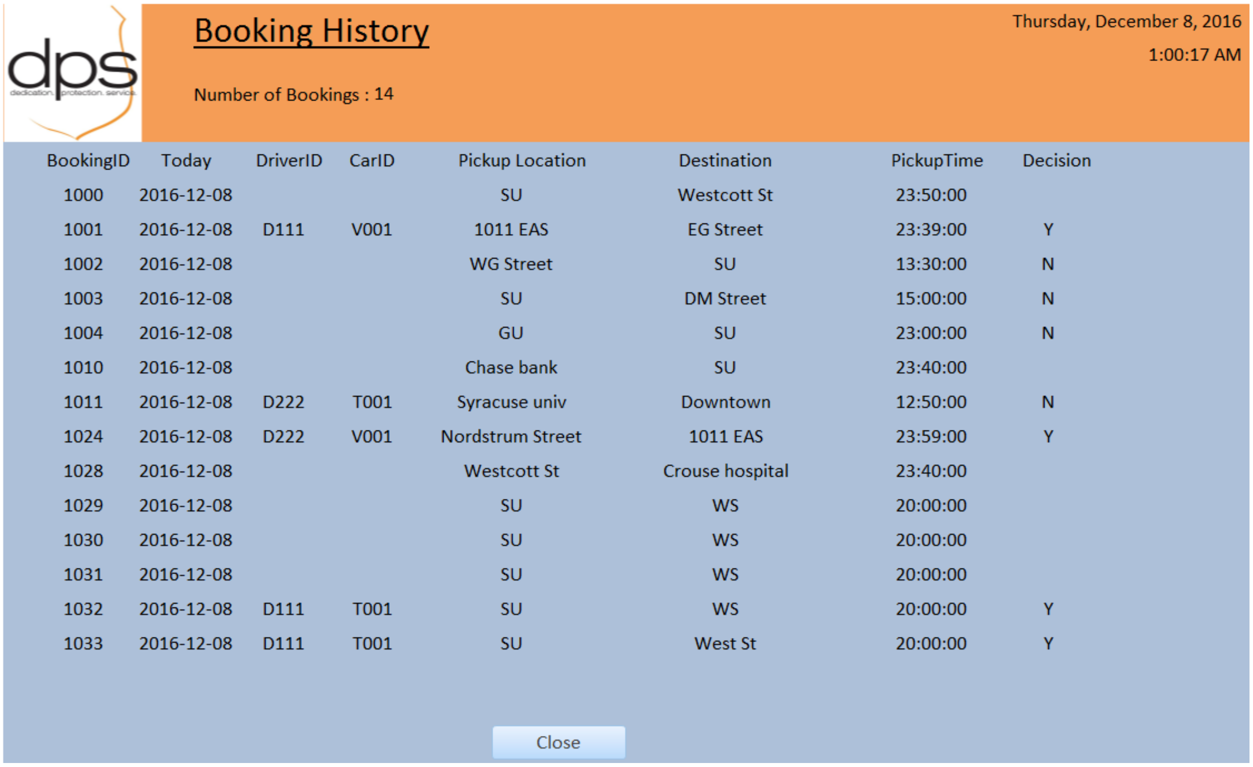
12:56:27 AM

Number of Users : 4

UserID	First Name	LName	Number of Bookings Approved	Number of Bookings Disapproved	Count Of Booking ID
C111	Farees	Patel	4	4	8
C222	Ellon	Musk	1	0	1
C333	Kevin	Johns	1	0	1
C444	Chris	Hawk	0	0	0

(d) Report 4: Booking history of a user

This report enlists all the bookings that the specific user has made. The admin has to enter the User ID of the user to get the detail information of all the bookings tied to that user.



Booking History

Thursday, December 8, 2016

1:00:17 AM

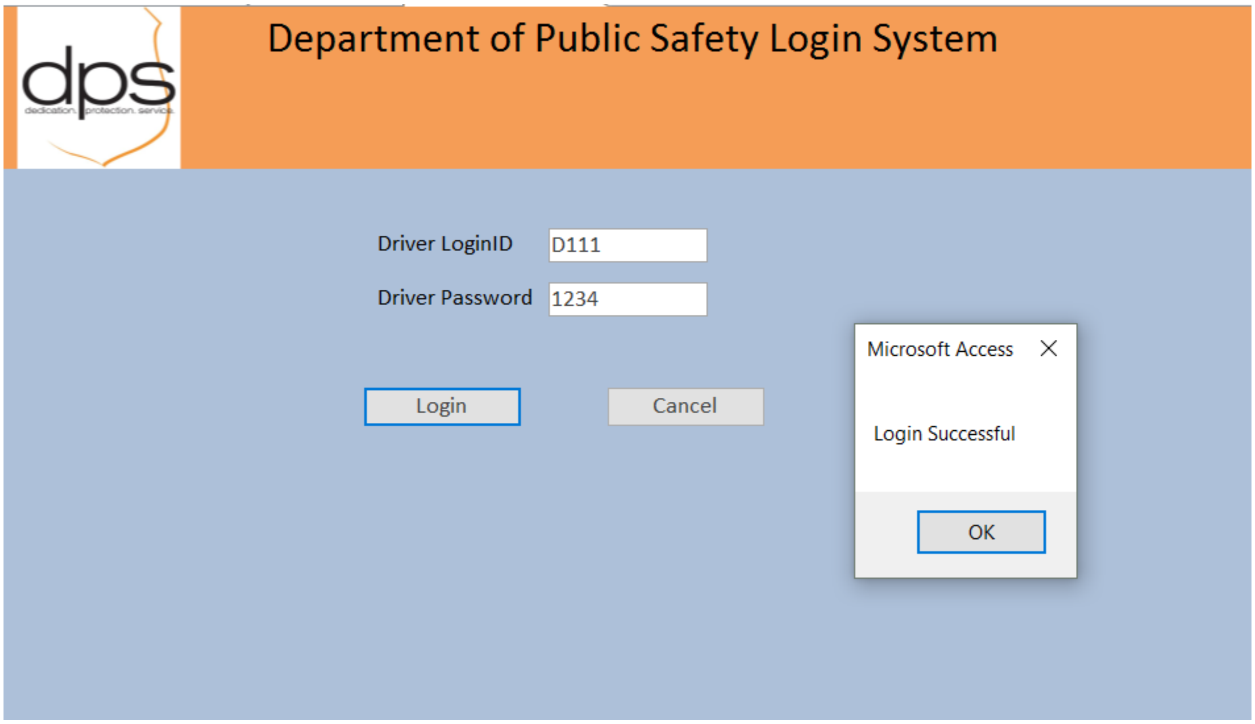
Number of Bookings : 14

BookingID	Today	DriverID	CarID	Pickup Location	Destination	PickupTime	Decision
1000	2016-12-08			SU	Westcott St	23:50:00	
1001	2016-12-08	D111	V001	1011 EAS	EG Street	23:39:00	Y
1002	2016-12-08			WG Street	SU	13:30:00	N
1003	2016-12-08			SU	DM Street	15:00:00	N
1004	2016-12-08			GU	SU	23:00:00	N
1010	2016-12-08			Chase bank	SU	23:40:00	
1011	2016-12-08	D222	T001	Syracuse univ	Downtown	12:50:00	N
1024	2016-12-08	D222	V001	Nordstrum Street	1011 EAS	23:59:00	Y
1028	2016-12-08			Westcott St	Crouse hospital	23:40:00	
1029	2016-12-08			SU	WS	20:00:00	
1030	2016-12-08			SU	WS	20:00:00	
1031	2016-12-08			SU	WS	20:00:00	
1032	2016-12-08	D111	T001	SU	WS	20:00:00	Y
1033	2016-12-08	D111	T001	SU	West St	20:00:00	Y

3. Driver

Driver Login Page:

The above form is a login form for students, staff and faculty and it provides an interface for users to use DPS services.



Department of Public Safety Login System

Driver LoginID D111

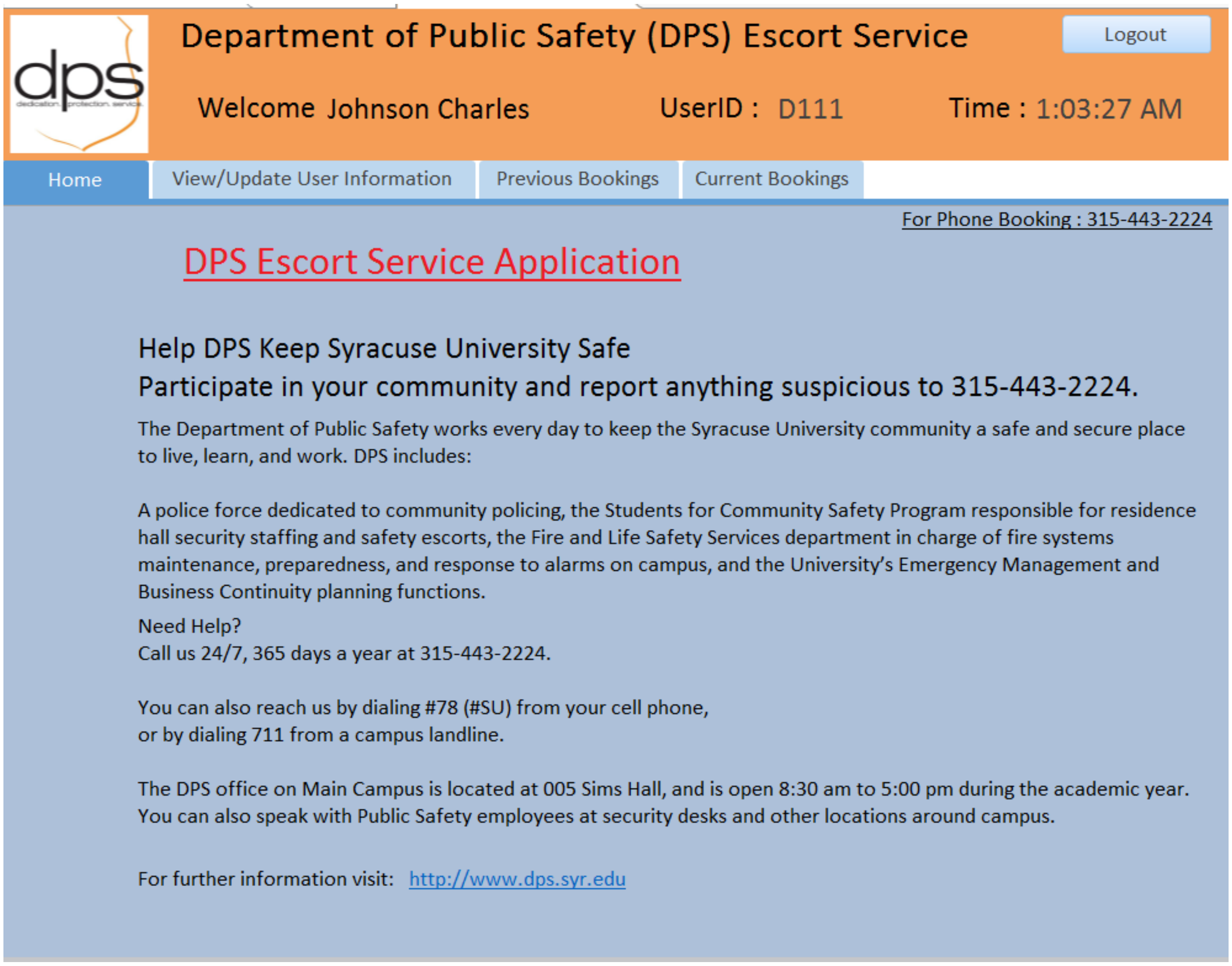
Driver Password 1234

Microsoft Access

Login Successful

User Panel (Homepage):

The above form is a user panel for the driver. Also, it provides an overview and the necessary information about the DPS service.



Department of Public Safety (DPS) Escort Service

Welcome Johnson Charles UserID : D111 Time : 1:03:27 AM

For Phone Booking : 315-443-2224

DPS Escort Service Application

Help DPS Keep Syracuse University Safe
Participate in your community and report anything suspicious to 315-443-2224.

The Department of Public Safety works every day to keep the Syracuse University community a safe and secure place to live, learn, and work. DPS includes:

A police force dedicated to community policing, the Students for Community Safety Program responsible for residence hall security staffing and safety escorts, the Fire and Life Safety Services department in charge of fire systems maintenance, preparedness, and response to alarms on campus, and the University's Emergency Management and Business Continuity planning functions.

Need Help?
Call us 24/7, 365 days a year at 315-443-2224.

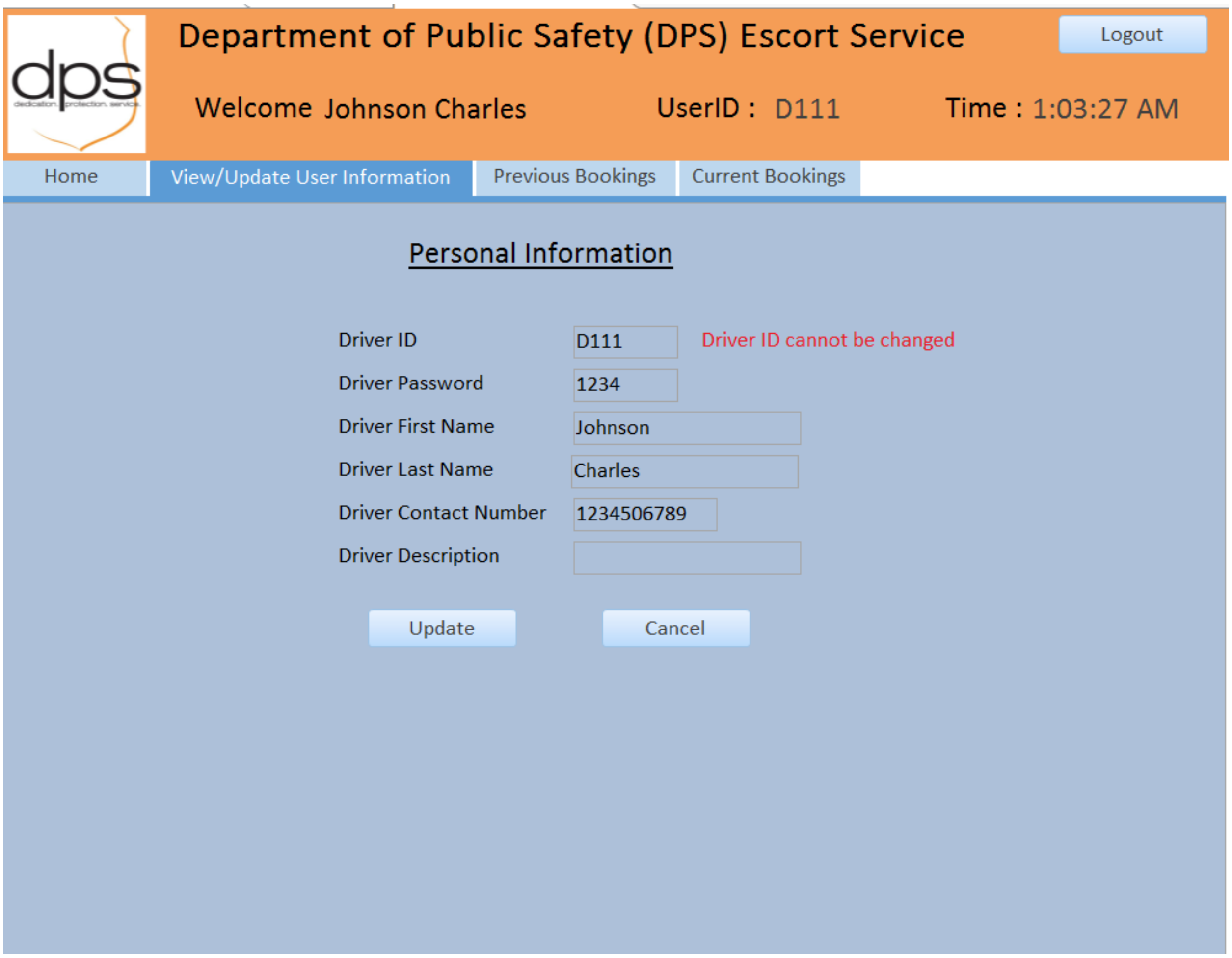
You can also reach us by dialing #78 (#SU) from your cell phone,
or by dialing 711 from a campus landline.

The DPS office on Main Campus is located at 005 Sims Hall, and is open 8:30 am to 5:00 pm during the academic year. You can also speak with Public Safety employees at security desks and other locations around campus.

For further information visit: http://www.dps.syr.edu

View/Update and User Information:

The below form allows drivers to view or update their personal information, however, the driver does not have the rights to change their Driver ID. Driver can change their email and contact numbers to update their information.



Department of Public Safety (DPS) Escort Service

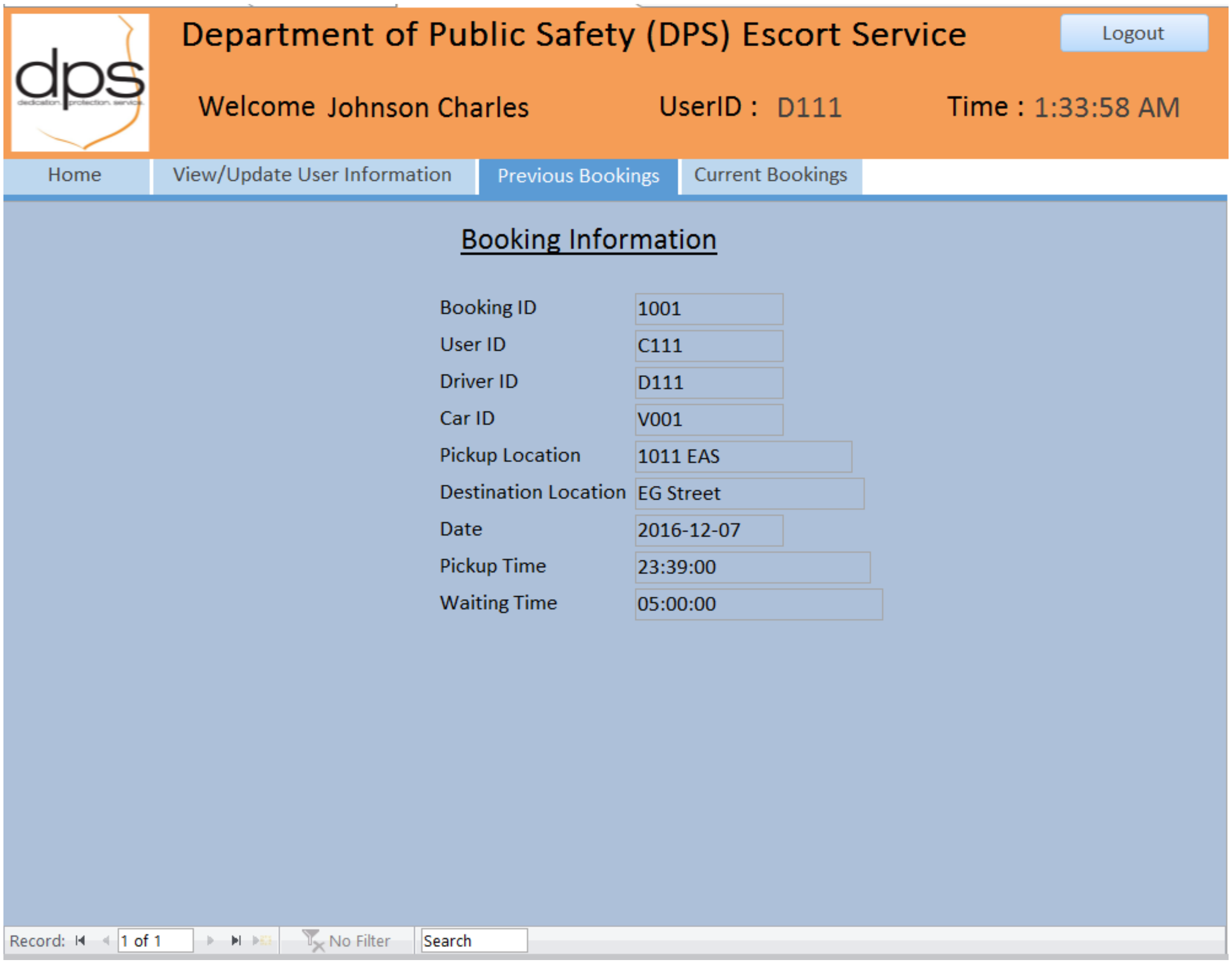
Welcome Johnson Charles UserID : D111 Time : 1:03:27 AM

Personal Information

Driver ID	D111	Driver ID cannot be changed
Driver Password	1234	
Driver First Name	Johnson	
Driver Last Name	Charles	
Driver Contact Number	1234506789	
Driver Description		

Previous Booking:

The below form allows drivers to view all the bookings that they have served in the past. Since, my application has a high level of abstraction, the ‘previous bookings tab’ will only contain booking record that is related to the specific driver who has login to the system.



Department of Public Safety (DPS) Escort Service

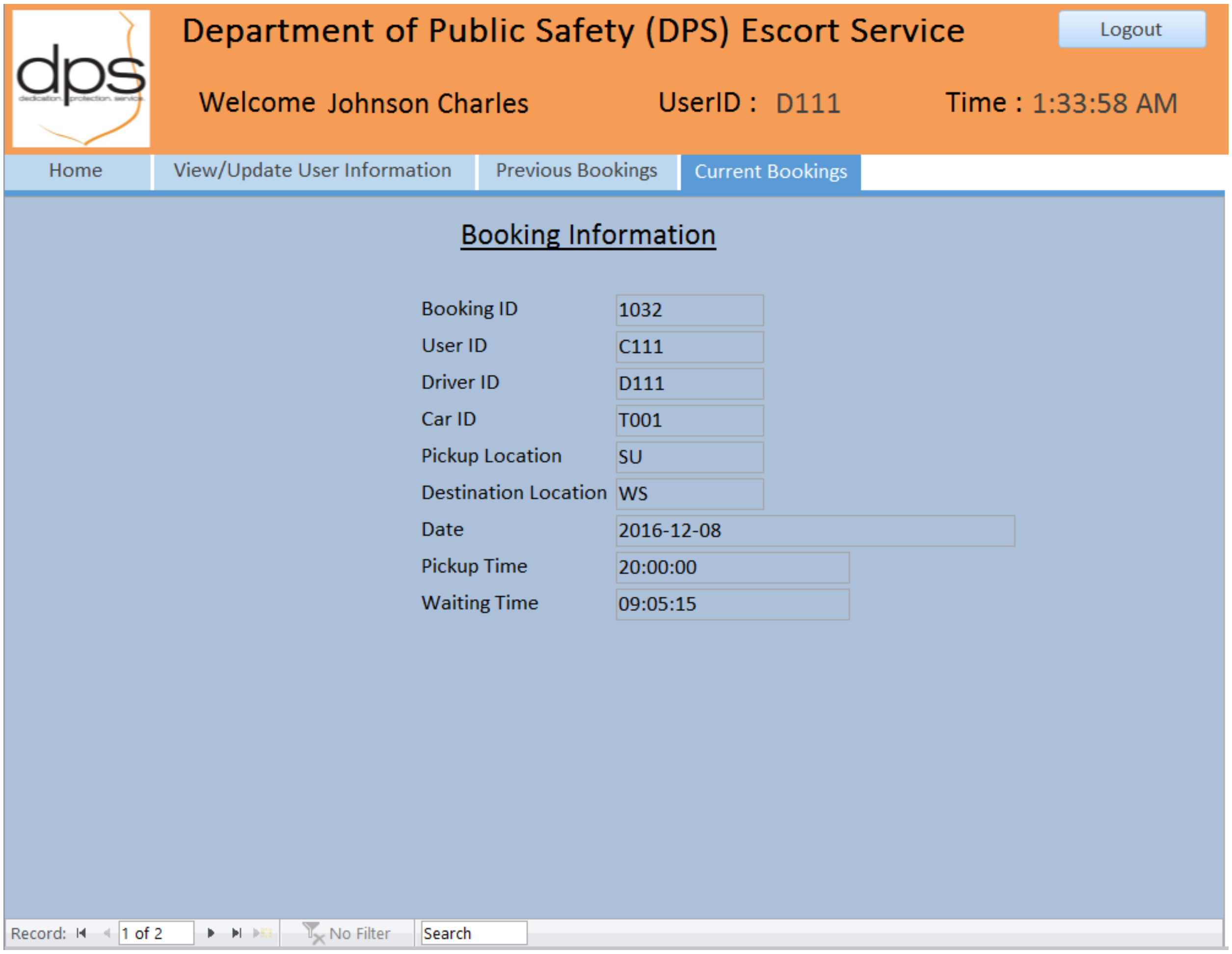
Welcome Johnson Charles UserID : D111 Time : 1:33:58 AM

Booking Information

Booking ID	1001
User ID	C111
Driver ID	D111
Car ID	V001
Pickup Location	1011 EAS
Destination Location	EG Street
Date	2016-12-07
Pickup Time	23:39:00
Waiting Time	05:00:00

Current Booking:

The below form allows drivers to view all the bookings that they have to serve. All the bookings will have all the information that the driver needs in order to escort the passenger to their destination. Since, my application has a high level of abstraction, the ‘previous bookings tab’ will only contain booking record that is related to the specific driver who has login to the system.



Booking Information

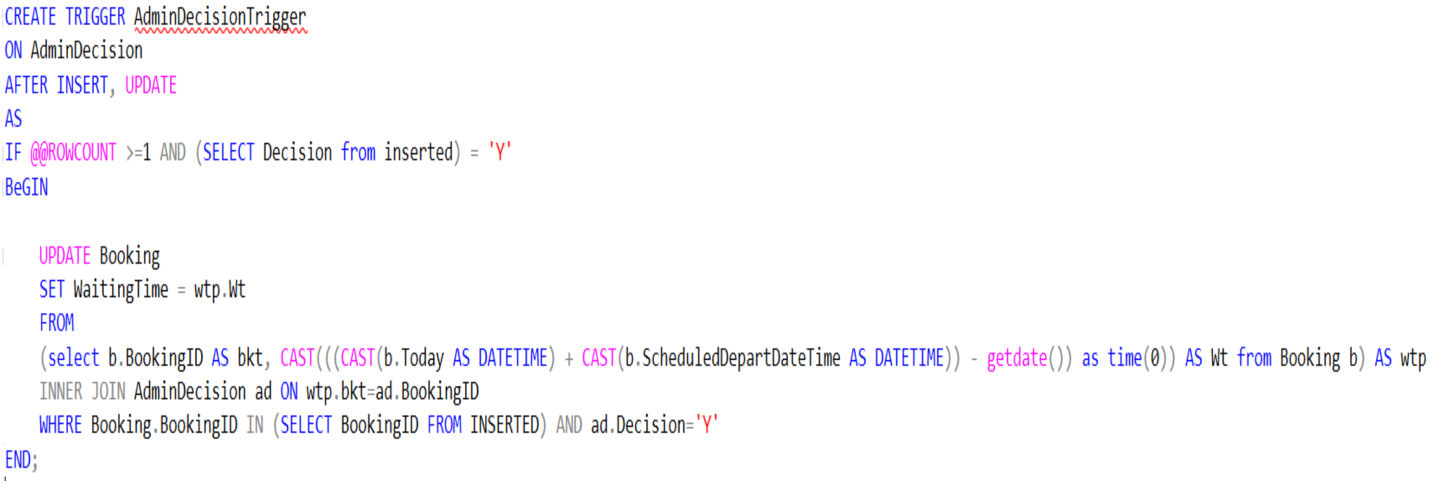
Booking ID	1032
User ID	C111
Driver ID	D111
Car ID	T001
Pickup Location	SU
Destination Location	WS
Date	2016-12-08
Pickup Time	20:00:00
Waiting Time	09:05:15

TRIGGER:

I implemented a trigger to notify the user for the time that he has to wait for the DPS escort service to show up.

- Logic: Whenever the admin approves the booking by entering ‘Y’ in the ‘AdminDecision’ table the waiting time column in the booking table for that specific user is updated.

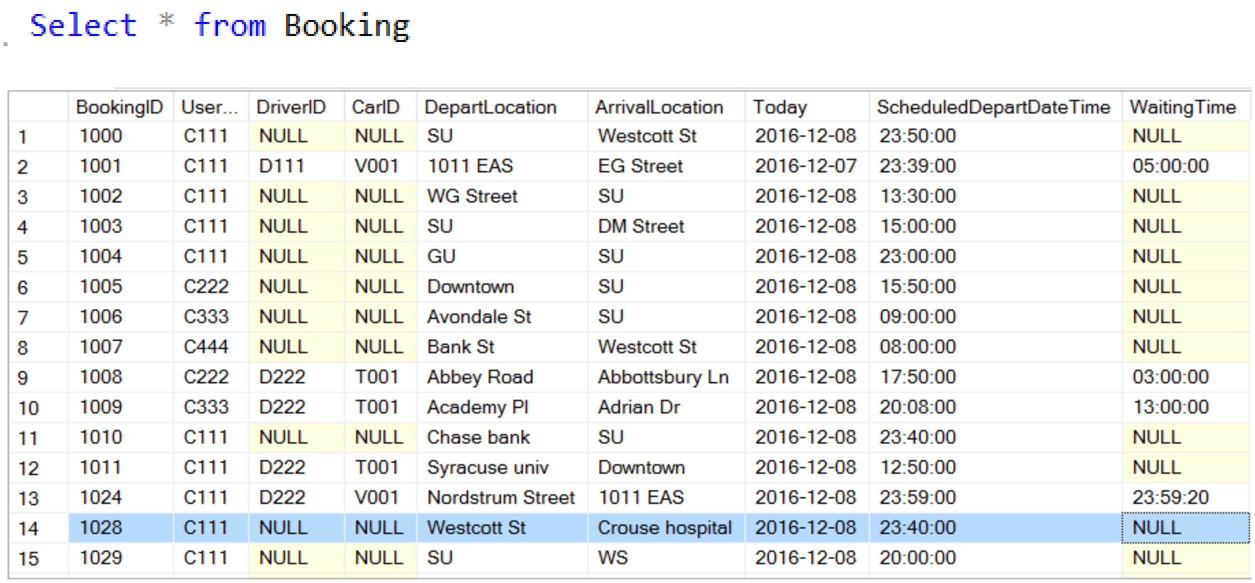
Trigger code:



```
CREATE TRIGGER AdminDecisionTrigger
ON AdminDecision
AFTER INSERT, UPDATE
AS
IF @@ROWCOUNT >=1 AND (SELECT Decision from inserted) = 'Y'
BeGIN

    UPDATE Booking
    SET WaitingTime = wtp.Wt
    FROM
    (select b.BookingID AS bkt, CAST(((CAST(b.Today AS DATETIME) + CAST(b.ScheduledDepartDateTime AS DATETIME)) - getdate()) as time(0)) AS Wt from Booking b) AS wtp
    INNER JOIN AdminDecision ad ON wtp.bkt=ad.BookingID
    WHERE Booking.BookingID IN (SELECT BookingID FROM INSERTED) AND ad.Decision='Y'
END;
```

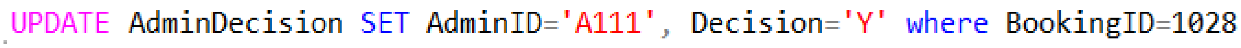
Before running the trigger:



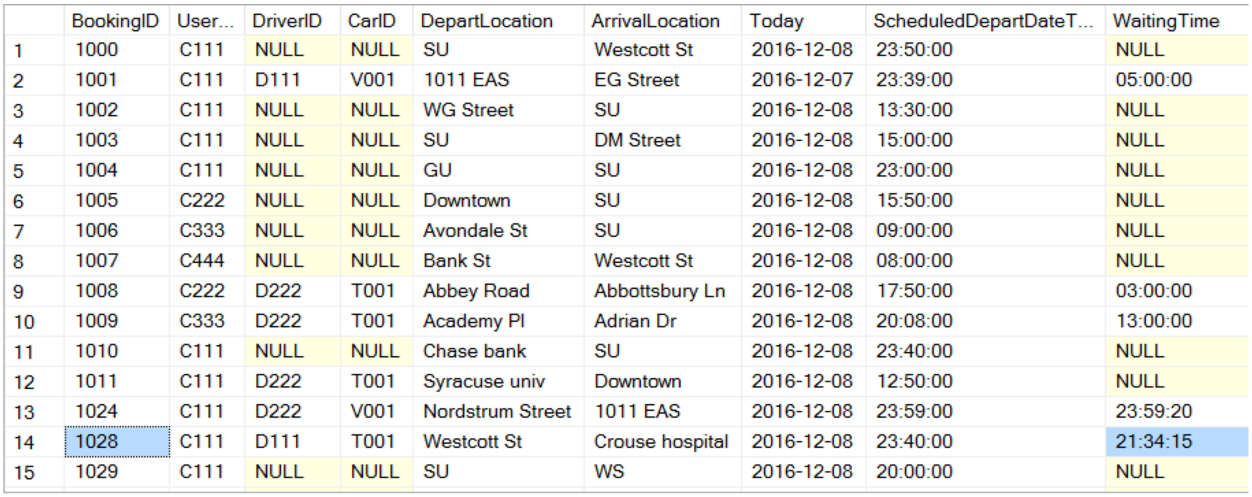
```
Select * from Booking
```

	BookingID	User...	DriverID	CarID	DepartLocation	ArrivalLocation	Today	ScheduledDepartDateTime	WaitingTime
1	1000	C111	NULL	NULL	SU	Westcott St	2016-12-08	23:50:00	NULL
2	1001	C111	D111	V001	1011 EAS	EG Street	2016-12-07	23:39:00	05:00:00
3	1002	C111	NULL	NULL	WG Street	SU	2016-12-08	13:30:00	NULL
4	1003	C111	NULL	NULL	SU	DM Street	2016-12-08	15:00:00	NULL
5	1004	C111	NULL	NULL	GU	SU	2016-12-08	23:00:00	NULL
6	1005	C222	NULL	NULL	Downtown	SU	2016-12-08	15:50:00	NULL
7	1006	C333	NULL	NULL	Avondale St	SU	2016-12-08	09:00:00	NULL
8	1007	C444	NULL	NULL	Bank St	Westcott St	2016-12-08	08:00:00	NULL
9	1008	C222	D222	T001	Abbey Road	Abbottsbury Ln	2016-12-08	17:50:00	03:00:00
10	1009	C333	D222	T001	Academy Pl	Adrian Dr	2016-12-08	20:08:00	13:00:00
11	1010	C111	NULL	NULL	Chase bank	SU	2016-12-08	23:40:00	NULL
12	1011	C111	D222	T001	Syracuse univ	Downtown	2016-12-08	12:50:00	NULL
13	1024	C111	D222	V001	Nordstrum Street	1011 EAS	2016-12-08	23:59:00	23:59:20
14	1028	C111	NULL	NULL	Westcott St	Crouse hospital	2016-12-08	23:40:00	NULL
15	1029	C111	NULL	NULL	SU	WS	2016-12-08	20:00:00	NULL

During the execution of the trigger:



```
UPDATE AdminDecision SET AdminID='A111', Decision='Y' where BookingID=1028
```



	BookingID	User...	DriverID	CarID	DepartLocation	ArrivalLocation	Today	ScheduledDepartDateT...	WaitingTime
1	1000	C111	NULL	NULL	SU	Westcott St	2016-12-08	23:50:00	NULL
2	1001	C111	D111	V001	1011 EAS	EG Street	2016-12-07	23:39:00	05:00:00
3	1002	C111	NULL	NULL	WG Street	SU	2016-12-08	13:30:00	NULL
4	1003	C111	NULL	NULL	SU	DM Street	2016-12-08	15:00:00	NULL
5	1004	C111	NULL	NULL	GU	SU	2016-12-08	23:00:00	NULL
6	1005	C222	NULL	NULL	Downtown	SU	2016-12-08	15:50:00	NULL
7	1006	C333	NULL	NULL	Avondale St	SU	2016-12-08	09:00:00	NULL
8	1007	C444	NULL	NULL	Bank St	Westcott St	2016-12-08	08:00:00	NULL
9	1008	C222	D222	T001	Abbey Road	Abbottsbury Ln	2016-12-08	17:50:00	03:00:00
10	1009	C333	D222	T001	Academy Pl	Adrian Dr	2016-12-08	20:08:00	13:00:00
11	1010	C111	NULL	NULL	Chase bank	SU	2016-12-08	23:40:00	NULL
12	1011	C111	D222	T001	Syracuse univ	Downtown	2016-12-08	12:50:00	NULL
13	1024	C111	D222	V001	Nordstrum Street	1011 EAS	2016-12-08	23:59:00	23:59:20
14	1028	C111	D111	T001	Westcott St	Crouse hospital	2016-12-08	23:40:00	21:34:15
15	1029	C111	NULL	NULL	SU	WS	2016-12-08	20:00:00	NULL