Department of Public Safety (DPS) service Database Application

Mohammed Farees Patel

Syracuse university

School of Information Studies

IST 659 - Project Design Report

Executive 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
- The driver and user can both provide their respective feedback

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.**
UserLogin	**Contains login credentials for users.** **Reason for making a separate table: Provides better security.**
UserID	PRIMARY KEY: Each user will have a unique userID to identify the user
Login_username	
Login_password	
UserType	Determines the role of the user. [Possible values: BookingUser(B), Administration(A) or Driver(D)]
BookingUser	**Child entity of UserLogin. Contains all user information of students, staff and the faculty who orders for a DPS escort service.**
BookingUserID	PRIMARY KEY: Each BookingUser has a unique userID to identify the specific user FOREIGN KEY: Associated with Primary Key of 'UserLogin' table. This will help in identifying that the user login is a BookingUser i.e. a user who wants to order the DPS service
BookingUserFirstName	
BookingUserLastName	
BookingUserEmail	
BookingUserContact	
BookingUserAddress	
BookingUserKind	Determines the role of the booking user. [Possible values: Student(1), Faculty(2) or Staff(3)]

Student	**Child entity of BookingUser. Contains Student information**
BookingUserID	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**
BookingUserID	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**
BookingUserID	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	**Child entity of UserLogin. Contains all the necessary driver information who drives the DPS car.**

DriverID	PRIMARY KEY: Each driver has a unique DriverID to identify himself/herself FOREIGN KEY: Associated with Primary Key of 'UserLogin' table. This will help in identifying that the user login is a Driver i.e. a user who drives the DPS car.
DriverFirstName	
DriverLastName	
DriverContact	
DriverDesc	Contains additional information i.e. available timings, specific commuting routes, etc.
Administration	**Child entity of UserLogin. 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. FOREIGN KEY: Associated with Primary Key of 'UserLogin' table. 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
AdminFirstName	
AdminLastName	
DPScar	**Stores information of all the cars that DPS drivers use for commuting**
CarNum	PRIMARY KEY: Each car has a unique CarNum which identifies the car (E.g.: car number plate)

CarName	
CarDesc	
TotalCars	Indicates the total number of cars operating for the DPS Escort Service
AvailCars	Indicates the number of cars available for bookings
DPS_EscortService	**Stores information about all the users, drivers, administrators, cars and bookings for the DPS Escort Service.**
DPSserviceID	PRIMARY KEY: DPS has a unique service number which identify Syracuse University's (SU's) DPS service
DriverID	FOREIGN KEY: Associated with Primary Key of 'Driver' table. Contains all the driver information in the parent Table. This indicates the total number of drivers working for the DPS Escort Service.
CarNum	FOREIGN KEY: Associated with Primary Key of 'DPScar' table. Contains all the car information in the parent Table. This indicates the total number of cars operating for the DPS Escort Service.
AdminID	FOREIGN KEY: Associated with Primary Key of 'Administration' table. Contains all the admin information in the parent Table. This indicates the total number of administrators who manages the DPS Escort Service.

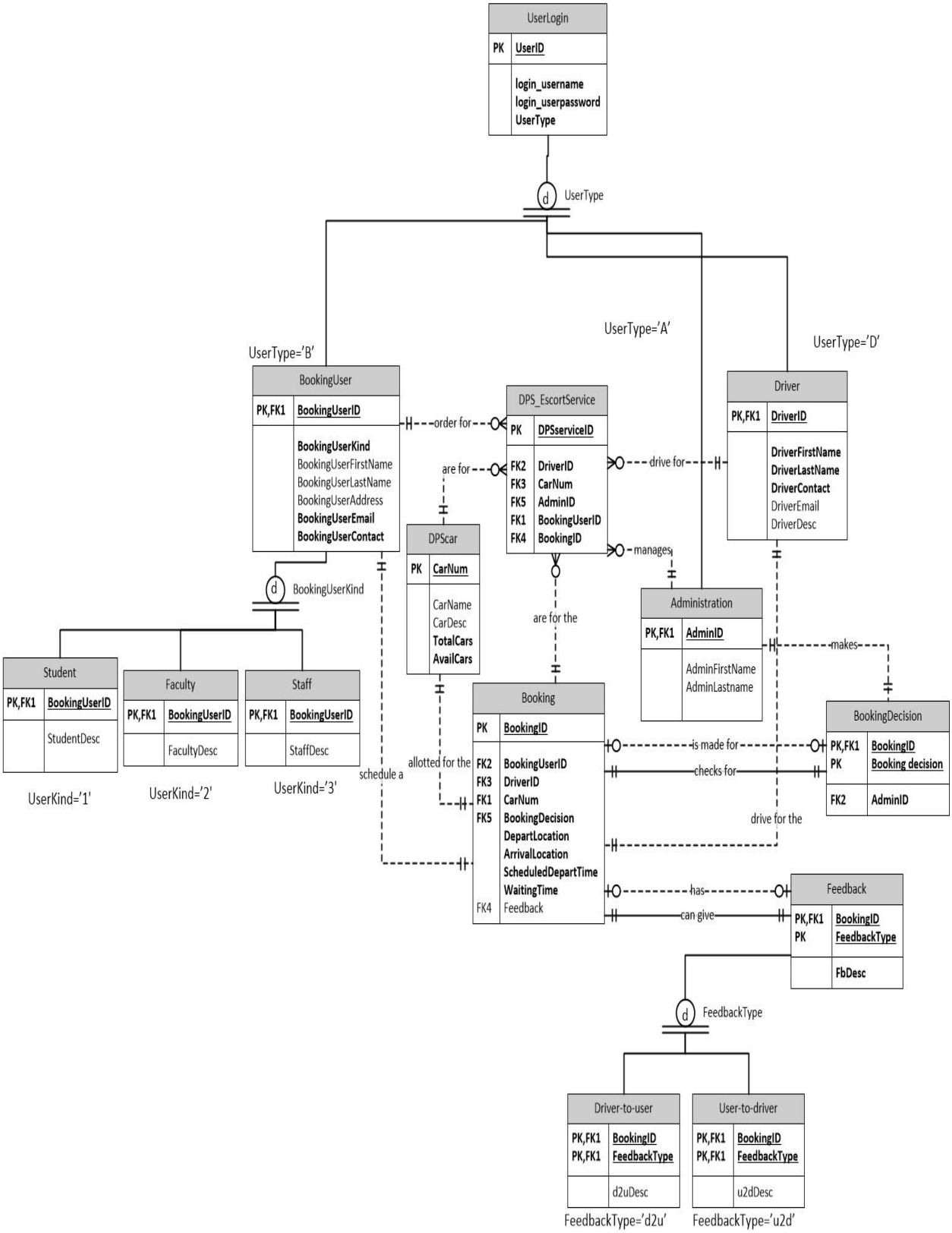
BookingUserID	FOREIGN KEY: Associated with Primary Key of 'BookingUser' table. Contains all the booking user information in the parent Table. This indicates the total number of booking users (i.e. students, faculty or staff) who orders for a DPS Escort Service.
BookingID	FOREIGN KEY: Associated with Primary Key of 'Booking' table. Contains all the booking information in the parent Table. This indicates the total number of accepted or rejected bookings by an users (i.e. students, faculty or staff) who orders for a DPS Escort Service.
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.
BookingUserID	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.

CarNum	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.
BookingDecision	FOREIGN KEY: Associated with Primary Key of 'BookingDecision' table. Contains information regarding the status of the booking i.e. booking accepted/rejected by the admin.
DepartLocation	Contains information about the pickup location.
ArrivalLocation	Contains information about the destination location.
ScheduledDepartTime	Contains information about the pickup time.
WaitingTime	Contains information about the time required for the DPS to show up.
Feedback	FOREIGN KEY: Associated with Primary Key of 'Feedback' table. Contains information about the feedback given by either driver to the user or vice versa. This also may contains one/two feedback for every booking.
BookingDecision	**Stores information about the bookings decisions being confirmed or rejected by the admin**
BookingID	PRIMARY KEY: BookingID is a part of the composite primary key. The composite 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.
BookingDecision	PRIMARY KEY: Uniquely identifies the status of the booking i.e. accepted/rejected using the specific BookingID
AdminID	FOREIGN KEY: Associated with Primary Key of 'Administration' table. The admin ID will help in identifying the booking decision made by the specific admin using his/her AdminID.
Feedback	**Stores the feedback information either given by driver to the user or given by the user to the driver**
BookingID	PRIMARY KEY: BookingID is a part of the composite primary key. The composite key as a whole determines the feedback for the respective BookingID FOREIGN KEY: Associated with Primary Key of 'Booking' table. The foreign key helps in identifying the feedback for a specific booking ID.
FeedbackType	PRIMARY KEY: FeedbackType uniquely identifies the feedback with the help of a composite key which comprises of BookingID and FeedbackType Determines the type of the feedback. [Possible values: driver to user(d2u) or user to driver (u2d)]
FbDesc	Contains information about the description of the feedback.
Driver-to-user	**Child entity of Feedback. Contains driver-to-user information**

BookingID	PRIMARY KEY: Uniquely identifies the driver-to-user booking feedback using the specific BookingID FOREIGN KEY: Associated with the Primary Key of 'Feedback' Table
FeedbackType	PRIMARY KEY: Stores 'd2u' character to identify that the feedback was given from the driver to the user. FOREIGN KEY: Associated with the Primary Key of 'Feedback' Table
d2uDesc	Description about the feedback (E.g. maximum length/characters of the feedback)
User-to-driver	**Child entity of Feedback. Contains user-to-driver information**
BookingID	PRIMARY KEY: Uniquely identifies the user-to-driver booking feedback using the specific BookingID FOREIGN KEY: Associated with the Primary Key of 'Feedback' Table
FeedbackType	PRIMARY KEY: Stores 'u2d' character to identify that the feedback was given from the user to the driver. FOREIGN KEY: Associated with the Primary Key of 'Feedback' Table
u2dDesc	Description about the feedback (E.g. maximum length/characters of the feedback)

Relational Data Model



BUSINESS RULES:

- Every user login must be either admin (A), booking user (B) or the driver (D)
- Booking user must be either Student (1), Faculty (2) or Staff (3)
- 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 car can be allotted for one booking
- Feedback can be provided by either the driver to the user (d2u) or by the user to the driver (u2d)
- Each booking may have at most two feedback

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
   - Function:
     - SU users can query the database to order a DPS escort service
     - Users can login using their username and password and select their 'UserType' as BookingUser (B).

- Users can then enter their SUID number, name, address, email and their contact details, and then select their 'BookingUserKind' depending upon whether they are students (1), faculty (2) or Staff (3)
- Users can book for a DPS escort service by then entering their 'DepartLocation', 'ArrivalLocation' and 'ScheduledDepartTime'.
- Users can also provide their valuable feedback about the driver and the service using the database application

- Constraints:
  - This segment of the users will have a partial access over the database system, where they can either book a service or provide their valuable feedback

2. Why DPS Administrators query the database
   - Function:
     - DPS administrators can query the database to manage and process bookings for the DPS escort service
     - Administrators can login using their username and password and select their 'UserType' as Administrators (A)
     - Administrators can monitor the bookings by reviewing the 'BookingID' and analyze the feasibility and authenticity of the booking
     - Administrators can make decision to either accept or reject the user booking request
     - Administrators can allot a driver and a car for a 'BookingID' and can read feedback from the users and drivers to improve the service of their escort service
   - Constraints:
     - This segment of the users will have a full access over the database system, where they can manage the whole database
3. Why the Drivers query the database
   - Function:
     - Drivers can query the database to know the 'BookingID' which has been allotted to them by the DPS escort service
     - Drivers can login using their username and password and select their 'UserType' as Drivers (D).

  - Drivers can know the ‘DepartLocation’, ‘ArrivalLocation’ and ‘ScheduledDepartTime’.
  - Drivers can also provide their valuable feedback about the user through the database application

- Constraints:
  - This segment of the users will have a partial access to the database system where they can know the current and the next passenger’s transit information and have the privilege to report the user for any misbehavior.