

Wolf Parking Management System

For Wolf Parking

CSC 540 Database Management Systems
Project Report 1

Daniel Buchanan(dsbuchan), Ophelia Sin(oysin), Aadil Tajani(atajani), Manali Teke(mteke)
September 26th, 2023

Assumptions:

1. Phone Number is unique for all drivers.
2. Student and Employee drivers will have a Phone Number.
3. Space Number is unique within a Parking Lot and a Zone is unique within a Parking Lot.
4. Parking Lot name is unique.
5. Parking manager is one of the administrators.
6. Any permit with a particular space type can park in a regular space. However, to access non regular space types, drivers have to hold a permit for that specific space type. For example, only a driver with a handicap parking permit can park in a handicap space.
7. A parking lot can contain multiple zones. A driver can park in the specified parking lot and zone in their permits.
8. A parking lot can have multiple types of spaces.
9. Each permit can have multiple lots and zones.
10. Administrators manage the billing processes.
11. On creation of a citation, payment status is set to unpaid by default and is available to the driver with the citation information and any status change reflects to all(driver, security, and administrator).
12. Drivers can get a new permit upon exhausting their respective limits only after getting a permit of the same type deleted.
13. Security can look up if a vehicle has a valid permit to note a violation and generate a citation.
14. The citation number is unique for every citation.
15. Each permit can only have one space type.

1. Problem Statement:

We will design a Wolf Parking Management System for administrators of the on-campus parking services to maintain parking lot information, zone information, space information, permit information, vehicle information, driver information, and citation information.

We are performing four major types of tasks:

- 1) information processing,
- 2) assigning zones to each parking lot and a type to a given space,
- 3) generating and maintaining citations, and
- 4) creating reports that include total citations, list of zones in each parking lot, the number of cars that are currently in violation and other statistics.

Since the information will be accessed, created, updated, and deleted simultaneously by multiple users, we opted to use a database system . There are multiple entities (like parking lots, zones, spaces, etc.) that are related to each other and handling the relationships becomes easier with databases. A database lets us store and manage large amounts of data in a structured manner, while preventing data corruption and enforcing data consistency. For example, a database system prevents parking in the wrong parking spot and ensures violations of permits can be identified accurately. Moreover, control mechanisms of a database ensures that only authorized users can perform specific operations on sensitive data. It will also allow us

to fetch related information easily by just using keys which is fast and most importantly, eliminates redundancy in data storage.

2. Intended Users:

Administrators: have access to all the information pertaining to the parking system including drivers, permits, vehicles, citations, and parking spots. They can add parking lots to the system, assign zones and spaces to the lots, assign a parking permit to a driver, change the availability of a space, and check if a car has a valid permit in their lot.

Security: can create/update/delete citations to vehicles that violate parking regulations, and check permits and vehicle information.

Drivers: can view their citations, pay their citations, and view their permit information.

3. Five Main Entities:

- a. **Driver:** Name, status, phone number
- b. **Parking Lot:** name, address
- c. **Permit Information:** permit ID, lot, start date, expiration date, expiration time, permit type
- d. **Vehicle:** car license number, model, year, color, manufacturer
- e. **Citation Information:** Citation number, citation date, citation time, fee, category, payment status

4. Tasks and Operations- Realistic Situations:

- Situation 1: Sam, a student, parks his car in a student handicap space using a regular permit. The security officer notices the violation and generates a citation. The citation includes details such as the date, time, lot where the car is parked, car license number, model, color, category, and the fee, setting status unpaid. Sam later gets the option to pay or appeal it through the system.
- Situation 2: At the end of the month, the university's parking administration team needs to generate a report on all citations issued. They generate a monthly citation report to analyze the number of citations given, the types of violations, and the total fees collected. This report helps in analyzing parking enforcement effectiveness and revenue generation.

5. Application Program Interfaces:

Information Processing:

- Enter/update/delete basic information about drivers, parking lots, zones, spaces, and permits.
 - enterDriverInfo(phone, univId, name, status)
return confirmation

- updateDriverInfo(phone, univId, name, status)
return confirmation
* If NULL value for any of the fields, then they will not be updated
- deleteDriverInfo(phone)
return confirmation
- enterParkingLotInfo(name, address)
return confirmation
- updateParkingLotInfo(name, address)
return confirmation
* If NULL value for any of the fields, then they will not be updated
- deleteParkingLotInfo(name)
return confirmation
- enterZoneInfo(zoneId)
return confirmation
- deleteZoneInfo(zoneId)
return confirmation
- enterSpaceInfo(spaceNumber, zoneId, parkingLotName, spaceType, availabilityStatus)
return confirmation
- updateSpaceInfo(spaceNumber, zoneId, parkingLotName, spaceType, availabilityStatus)
return confirmation
* If NULL value for any of the fields, then they will not be updated
- deleteSpaceInfo(spaceNumber, parkingLotName)
return confirmation
* If NULL value for any of the fields, then they will not be deleted
- *Assign zones to each parking lot and a type to a given space.*
 - assignZone(parkingLotName, zoneId)
return confirmation
* If NULL value for any of the fields, then they will not be updated, and if already assigned, change assignment to new data provided if data valid.
 - assignSpaceType(parkingLotName, spaceNumber)
return confirmation

* If NULL value for any of the fields, then they will not be updated, and if already assigned, change assignment to new data provided if data is valid.

- *Citation appeals can be requested and citation payment status updated accordingly.*
 - `acceptAppeal(citationNumber)`
return updated paymentStatus
 - `rejectAppeal(citationNumber)`
return paymentStatus

Maintaining permits and vehicle information for each driver:

- Assign permits to drivers according to their status.
 - `assignPermit(phone, status, permitID)`
return confirmation
* If NULL value for any of the fields, then they will not be entered
- Enter/update permit information and vehicle ownership information, including remove or add vehicles.
 - `createEnterPermitInfo(permitID, lot, zoneID, spaceType, startDate, expDate, expTime, permitType)`
return confirmation
* If NULL value for any of the fields, then they will not be entered
 - `updatePermitInfo(permitID, lot, zoneID, spaceType, startDate, expDate, expTime, permitType)`
return confirmation
* If NULL value for any of the fields, then they will not be updated
 - `deletePermitInfo(permitID)`
return confirmation
 - `enterVehicleInfo(licenseNumber, model, year, color, manufacturer, permitID)`
return confirmation
* If NULL value for any of the fields, then they will not be entered
 - `updateVehicleInfo(licenseNumber, model, year, color, manufacturer, permitID, phone)`
return confirmation
* If NULL value for any of the fields, then they will not be updated
 - `deleteVehicleInfo(licenseNumber)`
return confirmation
 - `getPermitInfo(identifierType, identifierValue)`
return permit records if exists using respective
identifier(permitID/phone/carLicenseNumber) else return confirmation

Generating and maintaining citations:

- Generate/maintain appropriate information for each citation. Before generating a citation, detect parking violations by checking if a car has a valid permit in the lot. Drivers have the ability to pay or appeal citations.
 - `checkValidPermit(carLicenseNumber)`
return True if permit is valid, False if it is not valid

- enterCitation(citationID, citationDate, category, citationTime, paymentStatus, licenseNumber, parkingLotName, fee)
 - returns true if citation was successfully created, false otherwise
 - * If NULL value for any of the fields, then they will not be entered
- payUpdateCitation(citationNumber)
 - return confirmation about payment and update the status as paid if payment goes through else status remains unchanged
- appealCitation(citationID)
 - return confirmation
- getCitation(licenseNumber)
 - return confirmation

Reports:

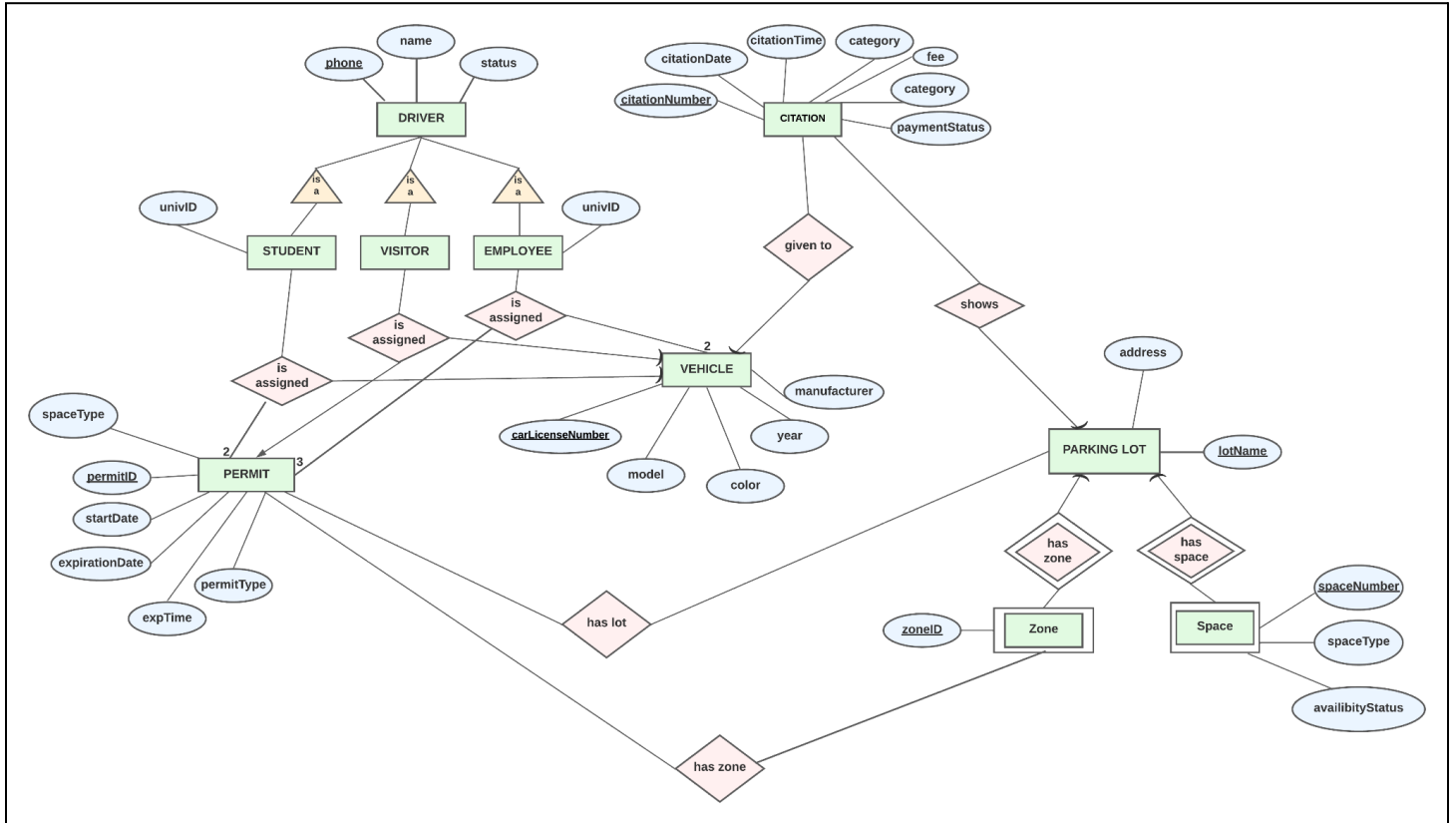
- Generate a report for citations.
 - getLotCitations(parkingLotName, startDate, endDate)
 - return number of citations in all zones in the lot
 - getAllLotZones()
 - return list of zones for each lot as tuple pairs
 - getAllCitations()
 - return number of cars that are currently in violation
 - getEmployeePermits(zoneID, lotName)
 - return number of employees having permits in the zone
 - getPermitInfo(permitID, phone number)
 - return permit information
 - *If both fields are empty, then the request will not be processed.
 - getSpaceAvailability(spaceType, zoneID, parkingLotName)
 - return space number
 - * If NULL value for any of the fields, then the request will not be processed
 - getPermitInfo(phone)
 - return a list of permits of the user

6. Description of Views:

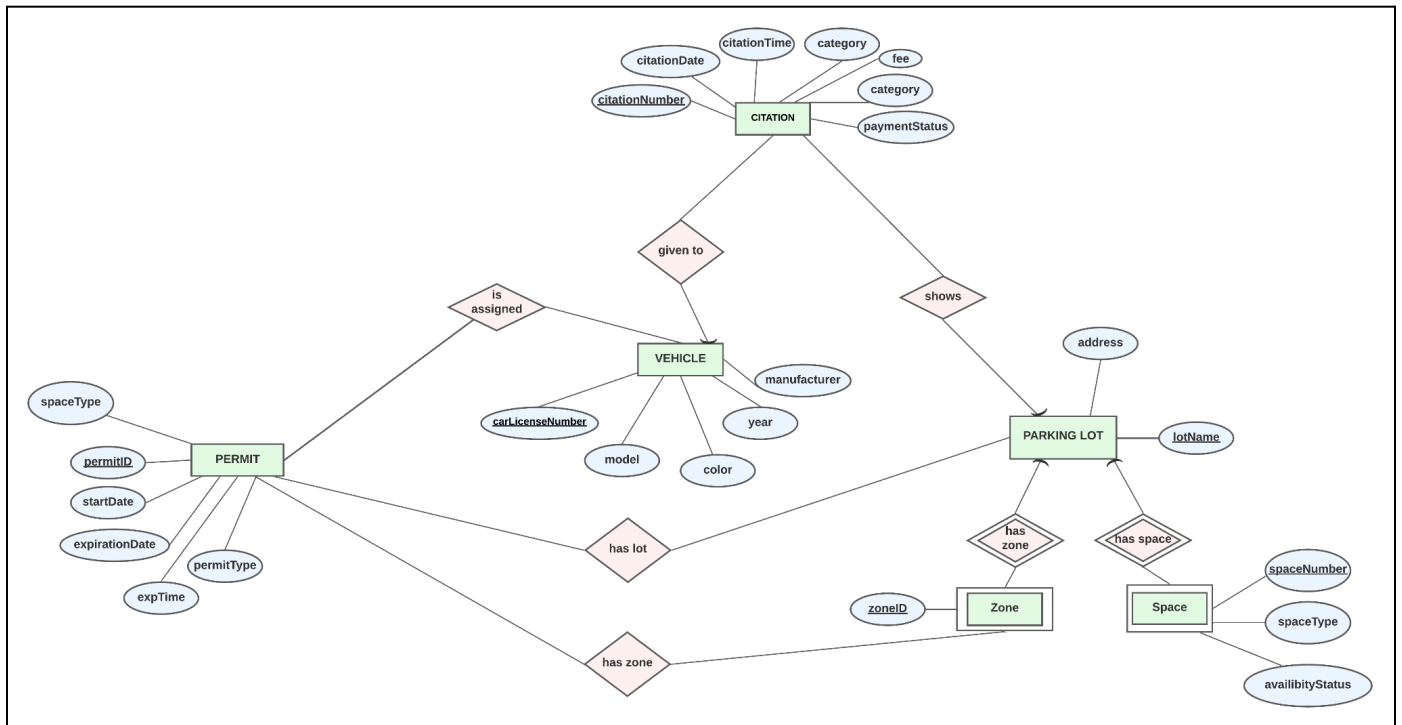
- **Administrators:** can view Driver Information, Parking Lot Information, Information on Zones and Spaces, Permit Information, and Citation information. They need an overall view of all the entities, so that they can manage the parking lots, assign permits to drivers, review citation appeals and manage billings for citations.
- **Security:** can view Citation Information, Vehicle Information, Permit Information, Space Information, Zone Information, and Lot Information to view validity of the vehicle permits and generate citations.
- **Driver:** can view Citation Information assigned to their own vehicles, Lot Information, and their own Permit Information.

7. Local E/R Diagrams:

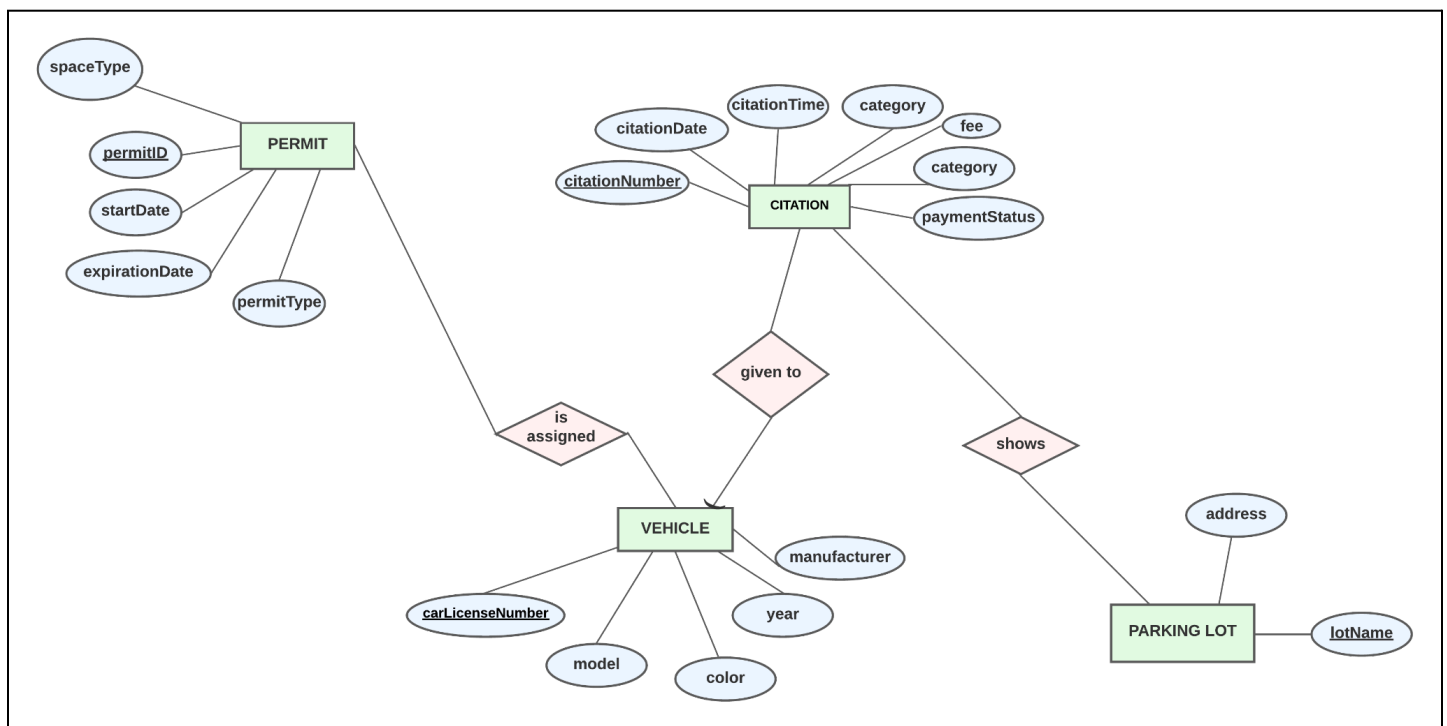
A. Administrator View :



B. Security:



C. Driver :



8. Description of Local E/R diagram:

- A driver will use a phone number as the key, as UnivID can only be used to identify Students and Employees, not Visitors.
- The Driver entity will have sub-entities Visitor, Student, and Employee to reflect the unique relations each sub-entity will have with Permits and Vehicles.
- Each visitor can have at most one permit (pointed arrow). This is the case as visitors are not allowed additional permits beyond the visitor permit.
- Each visitor can have exactly one vehicle in the parking system.
- Each student can have exactly one (rounded arrow) vehicle. Even though students can get at most two permits (1 any and 1 special events/park and ride), each one of those permits must be for exactly one vehicle.
- Each student can have at most two permits. Each student can have one normal permit, and then one other permit for special events.
- Each employee can have at most two vehicles.
- Each employee can have at most 3 permits. Each employee can have two normal permits, and then an additional permit for special events.
- A citation will be uniquely identified by the citation number (citationNumber) as the citation number should be unique for every generated citation.
- Each citation will be given to exactly one vehicle. A vehicle is also able to have many citations as vehicles can have multiple violations.
- Space is a weak entity set that uses space number (spaceNumber) and parking lot name (name) as the key. Space is a weak entity as it cannot exist without a lot, and is only identifiable with the name of the lot it belongs to.
- Zone is a weak entity set that uses zone id (zoneId) and parking lot name (name) as the key. Because permits are only valid for certain zones that belong to certain lots, zones should be identified with lots. Zones also can only exist with a lot.
- Permits use unique ids (permitID) that function as keys. This is so that permits are distinguishable between drivers, and can hold unique information including start date, expiration time, and permit type.
- Each permit will have at most one driver, who can be a student, visitor, or employee. This is so drivers cannot share the same permit with other drivers, and thus have multiple vehicles on one permit.
- Permits will have attribute space type (spaceType) to allow drivers to indicate which type of space ("regular", "handicap", "electric", etc.) they are allowed to park in.
- Each zone can be identified with many permits and each permit can contain multiple zones. This is because each permit has a type, and that type is given a set of zones and lots.
- Each citation will have exactly one lot where the citation occurred. Each lot can have many citations as multiple citations can occur in a single lot.
- Each lot can be assigned to many permits and each permit can have many lots. This relation exists as every zone can belong to multiple lots, so multiple lots can belong to one permit. The lot must also be specified with the permit to fully identify the zones the driver can park in.

9. Local Relational Schemas:

Admin View:

Driver(phone, name, status)

Student(phone, univId)

Employee(phone, univId)

Visitor(phone)

Zone(zoneID, lotName)

Space(spaceNumber, zoneID, lotName, spaceType, availabilityStatus)

CitationInformation(citationNumber, citationDate, citationTime, category, fee, paymentStatus)

Vehicle(carLicenseNumber, model, year, color, manufacturer)

Permit(permitId, spaceType, startDate, expirationDate, expirationTime, permitType)

ParkingLot(lotName, address)

IsAssigned(phone, permitId, carLicenseNumber)

GivenTo(citationNumber, carLicenseNumber)

Shows(citationNumber, lotName)

HasLot(permitId, lotName)

HasZone(permitId, zoneId, lotName)

Security View:

Zone(zoneID, lotName)

Space(spaceNumber, zoneID, lotName, spaceType, availabilityStatus)

CitationInformation(citationNumber, citationDate, citationTime, category, fee, paymentStatus)

Vehicle(carLicenseNumber, model, year, color, manufacturer)

Permit(permitId, spaceType, startDate, expirationDate, expirationTime, permitType)

ParkingLot(lotName, address)

IsAssigned(permitId, carLicenseNumber)

GivenTo(citationNumber, carLicenseNumber)

Shows(citationNumber, lotName)

HasLot(permitId, lotName)

HasZone(permitId, zoneId, lotName)

Driver View:

CitationInformation(citationNumber, citationDate, citationTime, category, fee, paymentStatus)

Vehicle(carLicenseNumber, model, year, color, manufacturer)

Permit(permitId, spaceType, startDate, expirationDate, expirationTime, permitType)

ParkingLot(lotName, address)

IsAssigned(permitId, carLicenseNumber)

GivenTo(citationNumber, carLicenseNumber)

Shows(citationNumber, lotName)

10. Local Schema Documentation:

Entity Sets to Relations:

- The entity sets in our diagram were made into relations, with the attributes the same for Driver, Permit, Citation, Parking Lot, Zone and Space.
- The Entity sets that are subsets of Driver were made into relations based on the E/R approach to avoid redundancy and save table space by mentioning parent Ids and subclass attributes.

Combining Many-One Relationships:

- In the following places an attribute was made by combining a many-to-one relationship. This eliminates redundancy and decreases the overhead that many tables cause. It makes queries quicker.
 - Weak entity set Space was made into a relation with all its attributes plus the LotName attribute, which is needed.
 - Weak entity Zone was also made into relation with its only attribute and key Zoneld with LotName as it is a weak entity.

Relationships to Relations:

- Strong relationships IsAssigned, GivenTo, Shows, HasLot, HasZone from the E/R diagrams have each been turned into relations in our schema. Their attributes in the schema are the keys of the entities they represent.
- HasZone weak relationship between Zone and Permit also has the additional attribute of LotName, which it gets from Zone as it is a weak entity on Parking Lot.
- HasSpace weak relationship between Space and Parking Lot gets covered in the entity Space as Space is a weak entity.