

Name: Maunee Patel (214139869)

## 1.0 Midterm Class Diagram

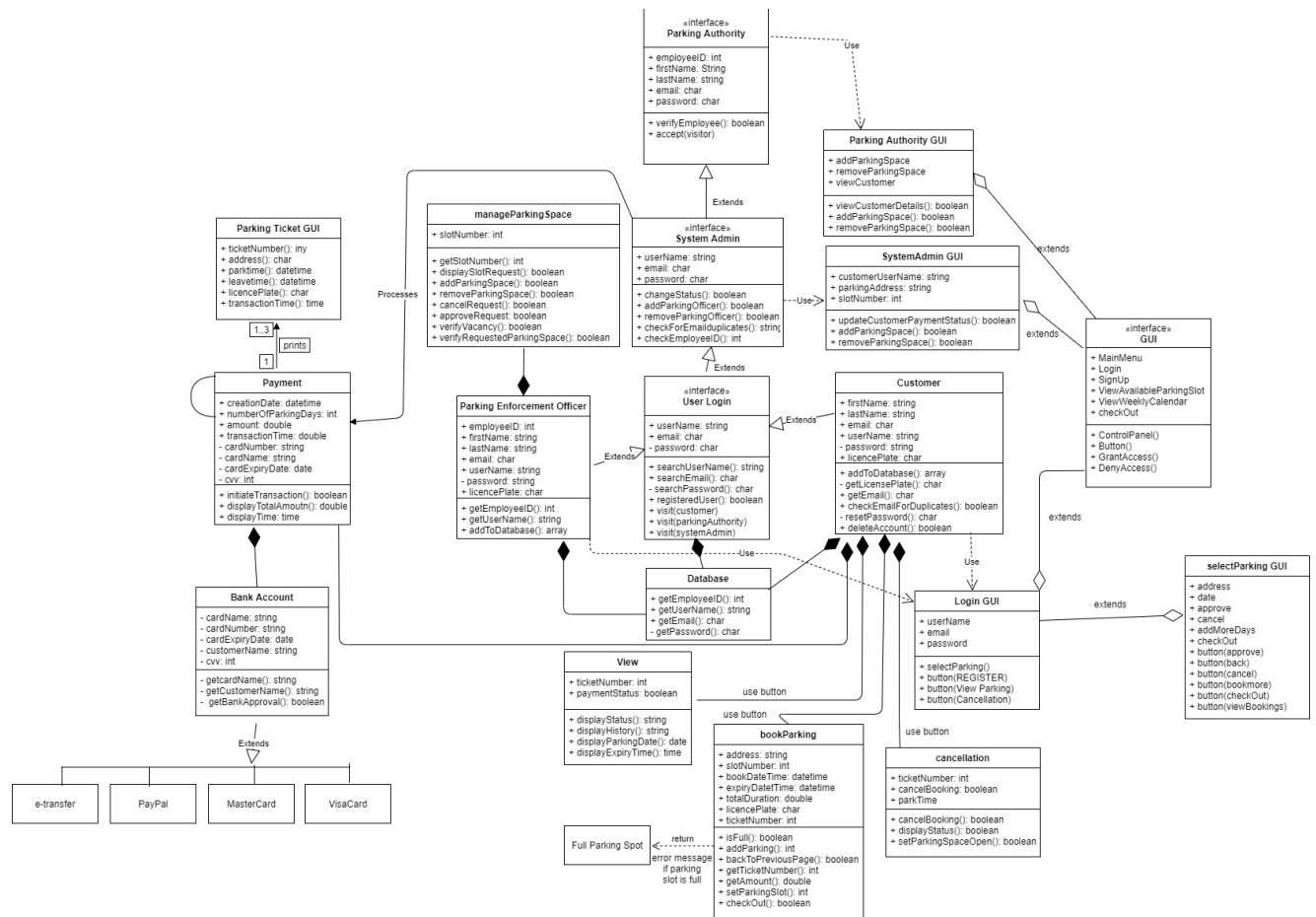


Figure 1: Midterm Class Diagram

### 1.1 Changes Made

- Same design pattern used.

### 1.2 Difference Between Final and Initial Diagrams

- userLogin is the main screen GUI which allows all 4 types of users to login. Then depending on what type of user they login as, they are able to see the various GUIs.
- GUI interface is now called userLogin GUI and it does not include viewAvailableParkingSpot, weekly calendar and checkout. That is done on another GUI window.

```

classDiagram
    class ParkingAuthority {
        +employeeID: int
        +firstName: string
        +lastName: string
        +email: char
        +password: char
        +verifyEmployee(): boolean
        +accept(visitor)
    }
    class ParkingAuthorityGUI {
        +addParkingSpace
        +removeParkingSpace
        +viewCustomer
        +viewCustomerDetails(): boolean
        +addParkingSpace(): boolean
        +removeParkingSpace(): boolean
    }
    class SystemAdmin {
        +username: string
        +email: char
        +password: char
        +changeStatus(): boolean
        +addParkingOfficer(): boolean
        +removeParkingOfficer(): boolean
        +checkForEmailDuplicates(): string
        +checkEmployee(): int
    }
    class SystemAdminGUI {
        +customerUsername: string
        +parkingAddress: string
        +slotNumber: int
        +updateCustomerPaymentStatus(): boolean
        +addParkingSpace(): boolean
        +removeParkingSpace(): boolean
    }
    class Customer {
        +firstName: string
        +lastName: string
        +email: char
        +username: string
        +password: string
        +licensePlate: char
        +addToDatabase(): array
        +getLicensePlate(): char
        +getEmail(): char
        +checkEmailForDuplicates(): boolean
        +resetPassword(): char
        +deleteAccount(): boolean
    }
    class UserLogin {
        +loginAs
        +username: string
        +email: char
        +password: char
        +searchByUsername(): string
        +searchByEmail(): char
        +searchByPassword(): char
        +registeredUser(): boolean
        +visit(customer)
        +visit(parkingAuthority)
        +visit(systemAdmin)
    }
    class ParkingEnforcementOfficer {
        +employeeID: int
        +firstName: string
        +lastName: string
        +email: char
        +username: string
        +password: string
        +licensePlate: char
        +getEmployeeID(): int
        +getUsername(): string
        +addToDatabase(): array
    }
    class ParkingTicketGUI {
        +ticketNumber(): int
        +address(): char
        +parkDate(): datetime
        +leaveTime(): datetime
        +licensePlate(): char
        +transactionTime(): time
    }
    class Payment {
        +creationDate: datetime
        +numberOfParkingDays: int
        +amount: double
        +transactionTime: double
        +cardNumber: string
        +cardName: string
        +cardExpiryDate: date
        +customerName: string
        +cvv: int
        +initiateTransaction(): boolean
        +displayTotalAmount(): double
        +displayTime: time
    }
    class BankAccount {
        +cardName: string
        +cardNumber: string
        +cardExpiryDate: date
        +customerName: string
        +cvv: int
        +getCardName(): string
        +getCustomerName(): string
        +getBankApproval(): boolean
    }
    class LoginGUI {
        +username
        +email
        +password
        +selectParking()
        +button(REGISTER)
        +button(View Parking)
        +button(Cancellation)
    }
    class selectParkingGUI {
        +address
        +date
        +approve
        +cancel
        +addNumberOfDays
        +button(approve)
        +button(back)
        +button(cancel)
        +button(bookmore)
        +button(checkOut)
        +button(viewBookings)
    }
    class bookParking {
        +address: string
        +slotNumber: int
        +bookDate: datetime
        +expiryDate: datetime
        +totalDuration: double
        +licensePlate: char
        +ticketNumber: int
        +isFull(): boolean
        +addParking()
        +backToPreviousPage(): boolean
        +if parking slot is full
        +getAmount(): double
        +setParkingSlot(): int
        +checkOut(): boolean
    }
    class cancellation {
        +ticketNumber: int
        +cancelBooking: boolean
        +partTime
        +cancelBooking(): boolean
        +displayStatus(): boolean
        +setParkingSpaceOpen(): boolean
    }
    class FullParkingSlot {
        +return
        +error message
    }

    ParkingAuthority <|-- ParkingAuthorityGUI
    ParkingAuthority <|-- SystemAdmin
    SystemAdmin <|-- SystemAdminGUI
    Customer <|-- UserLogin
    UserLogin <|-- ParkingEnforcementOfficer
    ParkingEnforcementOfficer <|-- ParkingTicketGUI
    ParkingTicketGUI <|-- Payment
    Payment <|-- BankAccount
    BankAccount <|-- LoginGUI
    LoginGUI <|-- selectParkingGUI
    selectParkingGUI <|-- bookParking
    bookParking <|-- cancellation
    cancellation <|-- FullParkingSlot

    ParkingAuthority --> ParkingAuthorityGUI : Use
    ParkingAuthority --> SystemAdmin : Use
    SystemAdmin --> SystemAdminGUI : Use
    SystemAdmin --> Customer : Use
    Customer --> UserLogin : Use
    UserLogin --> ParkingEnforcementOfficer : Use
    ParkingEnforcementOfficer --> ParkingTicketGUI : Use
    ParkingTicketGUI --> Payment : Use
    Payment --> BankAccount : Use
    BankAccount --> LoginGUI : Use
    LoginGUI --> selectParkingGUI : Use
    selectParkingGUI --> bookParking : Use
    bookParking --> cancellation : Use
    cancellation --> FullParkingSlot : Use

    ParkingAuthority --> ParkingEnforcementOfficer : Extends
    SystemAdmin --> UserLogin : Extends
    Customer --> UserLogin : Extends
    UserLogin --> ParkingEnforcementOfficer : Extends
    ParkingEnforcementOfficer --> ParkingTicketGUI : Extends
    Payment --> BankAccount : Extends
    BankAccount --> LoginGUI : Extends
    LoginGUI --> selectParkingGUI : Extends
    selectParkingGUI --> bookParking : Extends
    bookParking --> cancellation : Extends
    cancellation --> FullParkingSlot : Extends

    ParkingAuthority --> ParkingEnforcementOfficer : Association
    ParkingEnforcementOfficer --> ParkingTicketGUI : Association
    ParkingTicketGUI --> Payment : Association
    Payment --> BankAccount : Association
    BankAccount --> LoginGUI : Association
    LoginGUI --> selectParkingGUI : Association
    selectParkingGUI --> bookParking : Association
    bookParking --> cancellation : Association
    cancellation --> FullParkingSlot : Association

    ParkingAuthority --> ParkingEnforcementOfficer : Aggregation
    ParkingEnforcementOfficer --> ParkingTicketGUI : Aggregation
    ParkingTicketGUI --> Payment : Aggregation
    Payment --> BankAccount : Aggregation
    BankAccount --> LoginGUI : Aggregation
    LoginGUI --> selectParkingGUI : Aggregation
    selectParkingGUI --> bookParking : Aggregation
    bookParking --> cancellation : Aggregation
    cancellation --> FullParkingSlot : Aggregation

    ParkingAuthority --> ParkingEnforcementOfficer : Composition
    ParkingEnforcementOfficer --> ParkingTicketGUI : Composition
    ParkingTicketGUI --> Payment : Composition
    Payment --> BankAccount : Composition
    BankAccount --> LoginGUI : Composition
    LoginGUI --> selectParkingGUI : Composition
    selectParkingGUI --> bookParking : Composition
    bookParking --> cancellation : Composition
    cancellation --> FullParkingSlot : Composition
  
```

Element	Coverage	ed Instructions	ed Instructions	tal Instructions	
expressParkingApp		27.1 %	750	2,021	2,771
src		27.1 %	750	2,021	2,771
parking		19.6 %	442	1,811	2,253
test		59.5 %	308	210	518
cancellationTest.java		39.8 %	86	130	216
userLoginTest.java		68.5 %	174	80	254
bookParkingTest.java		100.0 %	48	0	48

## 5.0 System (Explanation with Visuals)

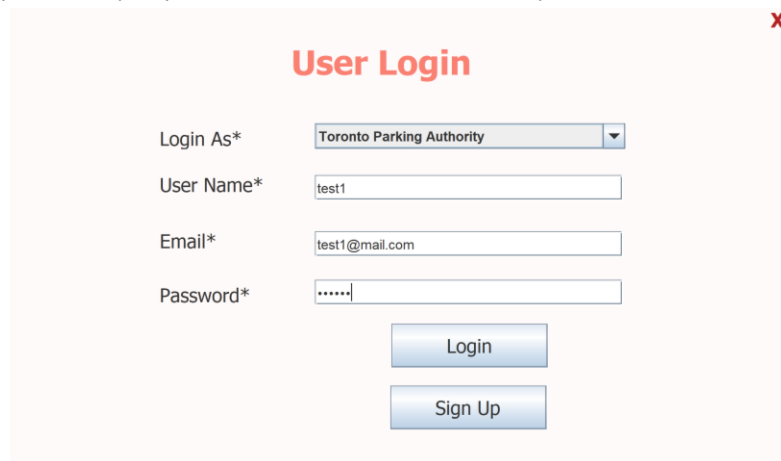
A screenshot of a 'User Login' window. The title bar is red with a close button 'X'. The window has a light pink background. It contains four input fields: 'Login As\*' with a dropdown menu showing 'Toronto Parking Authority', 'User Name\*' with 'test1', 'Email\*' with 'test1@mail.com', and 'Password\*' with masked characters '.....'. Below the fields are two buttons: 'Login' and 'Sign Up'.

Figure 4: Main Login

- Login As allows user to login as Toronto Parking Authority, System Admin, Parking Enforcement Officer, or Customer.
- The input data is then matched with the database to see if there is an existing user. If there is, then the user can login. (I have a small error in this. The dialog message box runs for every data line in the text file so you need to keep clicking "ok" and once the match is found, it will automatically lead you to next window.
- If the user wants to register: user inputs their details and clicks on the sign up button.

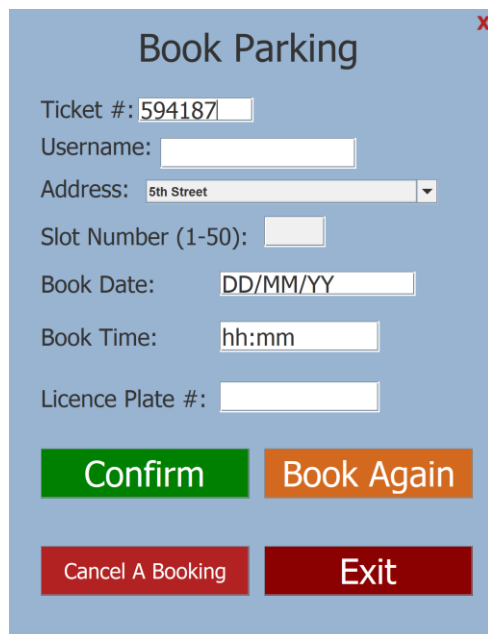
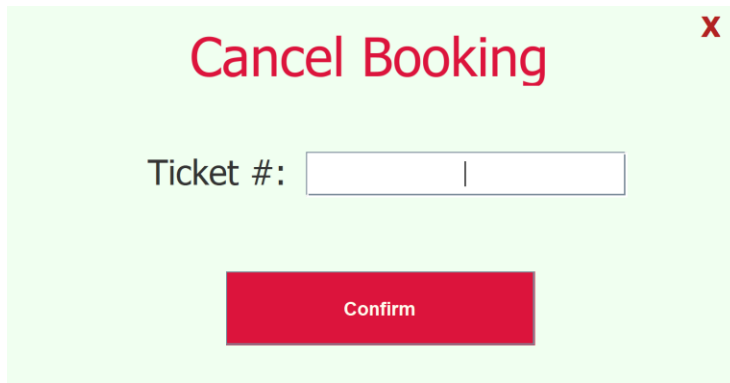
A screenshot of a 'Book Parking' window. The title bar is blue with a close button 'X'. The window has a light blue background. It contains several input fields: 'Ticket #:' with '594187', 'Username:', 'Address:' with a dropdown showing '5th Street', 'Slot Number (1-50):', 'Book Date:' with 'DD/MM/YY', 'Book Time:' with 'hh:mm', and 'Licence Plate #:'. At the bottom are four buttons: 'Confirm' (green), 'Book Again' (orange), 'Cancel A Booking' (red), and 'Exit' (dark red).

Figure 5: Book a parking slot

- Figure 5 is the window that opens after a successful login.
- An automated ticket number is given.
- User inserts username, address, slot number, book date and time and their licence plate number.
- Then click on confirm to get their confirmation message and ticket.
- If user wants to book again, this window is reloaded and a new ticket number is given.
- User can cancel a booking and they are led to a new window (Figure 6).
- The Exit button is to cancel and exit current input data.

A screenshot of a 'Cancel Booking' window. The window has a light green background. At the top center, the title 'Cancel Booking' is written in a large, bold, red font. In the top right corner, there is a small red 'X' icon. Below the title, the text 'Ticket #:' is followed by a white text input field with a thin black border. At the bottom center, there is a solid red rectangular button with the word 'Confirm' written in white text.

## Cancel Booking

Ticket #:

Confirm

*Figure 6: Window to cancel booking*

- User can type in their ticket number which gets matched in the database and the booking is deleted from database.

## 6.0 Specifications (From Midterm)

### 1.0 Assumptions

- Payment status is stored in a database where the system admin can verify.
- System admin already exists in the system and has master login.
- Customers can book up to 3 parking slots.
- Parking officer must login as a customer in order to book a payment. They can use their userName to login (not employeeID).
- When a parking officer is added, the system automatically assigns a unique employeeID.
- getEmployeeID() allows system admin to get the officer's ID number. This can allow the admin to pull out employee information from the employee database.
- There is one database which has sub-database for employees and customers and parking slot information.
- Slots 1 to 5 in all parking areas are handicap parking.

### 2.0 Interface

#### 2.1 Parking authority

- Parking authority has their own login for security.
- Authority can verify employee and can accept new login registers.

#### 2.2 System Admin

- Has their own login for security.
- Can update status of user payment for parking space.
- Can add or remove parking enforcement officers from the system. It finds employee information from database by getting employeeID number. Also checks for email duplicates for employees before creating parking enforcement officer.
- Can change customer's status to paid or unpaid.

#### 2.3 User Login

- This is the main login in interface. Both employee and customers go through this.
- It is also connected to the information database.
- User login will check database and will then allow a branch into either customer or parking enforcement officer. This is done by matching the userName and email.
- Password is used for security.

#### 2.4 GUI

- This is the main GUI interface. It stores all the information for other GUI classes.
- This interface holds the main control panel of the system look.

### 3.0 Classes

#### 3.1 Parking Enforcement Officer

- Must login through normal login page.
- Information gets stored in database.

- To book a parking slot, employee can use their userName which allows them to login like a “customer”. The officer’s portal has access to both customer and officer tabs.

### 3.1.1 Manage Parking Space

- Allows parking enforcement officer to manage parking space by adding or removing them.
- They can add, remove, cancel and grant requests.
- Each officer has an employeeID to ensure that they are an authenticated parking enforcement officer.
- They must verify that the parking is vacant before removing it from the system or granting requests.

### 3.2 Customer

- Once a customer logs in, they see 3 buttons for registration, view already booked parking and cancellation. This is in the Login GUI

#### 3.2.1 BookParking

- This checks for the inserted parking area, parking slot, time and date the customer inserts.
- If the slot is full, there is a “Full Parking Spot” message displayed.
- If the slot is available, the user can book and is given a transactionNumber. The customer can then book 2 more spaces and is given transactionNumbers for each.
- Then the customer can get a total amount of their transaction.
- When the customer checks out, the message is passed to the systemAdmin who verifies the payment and the payment is made using customer’s preferred method. This connects directly to the customer’s bank account. The transaction is verified by making sure the card name, expiry date and the CVV number is correct. Then the customer gets a digital copy of the total transactions which includes slot numbers, transactionNumber(s)/ticketNumber(s), licence plate, parktime/expirytime.

#### 3.2.2 View

- If the customer selects the view button, they are able to view all their transactions.

#### 3.2.3 Cancellation

- If the customer selects cancellation, the customer can select their transactions according to their ticketNumber(s) and cancel their booking.
- The system checks to ensure the booking is canceled 24 hours before parkTime.
- If requirement is met and cancellation is done, the system then opens that parking space by sending a message to the system admin.

### 3.3 Database

- Holds user information (employee/customer) for verification when a user logs into the system and when a user creates a new account.