

**Ho Chi Minh University of Technology  
Faculty of Computer Science and Engineering**



**SOFTWARE ENGINEERING**

**Project report #4**

## **SMART PARKING SYSTEM**

Student's Name: Nguyen Anh Hoang Phuc

Student's ID: 1752041

Project: Smart Parking

## **List of contents**

	<b>Page</b>
1. Module interface	3
2. Class diagram	3
3. Method description	4
4. Sequence diagram	8
5. Activity diagram	9
6. Design pattern	9
7. Demonstration	10

# 1. Module interface:

## 1.1 SearchEngine interface:

- (public) InputDestinationName(DName: string): void: Users input the place where they want to arrive
- (public) GetParkingName(): string: System return the parking area name which is recommended
- (public) SetParkingName(PName): void: System assigns value of PName of class MapView to value of PName of class SeachDestination.
- (public) SetCoordinate(X\_Coordinate: double[], Y\_Coordinate: double[]): void: System exchanges the destination name into coordinate with X\_Coordinate[0] and Y\_Coordinate[0]; System exchanges the list of parking names into coordinates with X\_Coordinate[1..N] and Y\_Coordinate[1..N]
- (public) SetNewCoordinateOnMap(X\_Coordinate: double[], Y\_Coordinate: double[]): void: System marks each parking coordinate on the map with the figures in order to users can have a quick look of all parking areas near their destination.

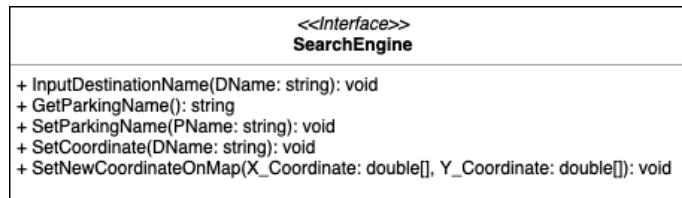


Figure 1: SearchEngine interface

## 1.2 BookingEngine interface:

- (public) Inform(bookininfo: BookingInfo): void: Users fill the booking form.



Figure 2: BookingEngine interface

# 2. Class diagram:

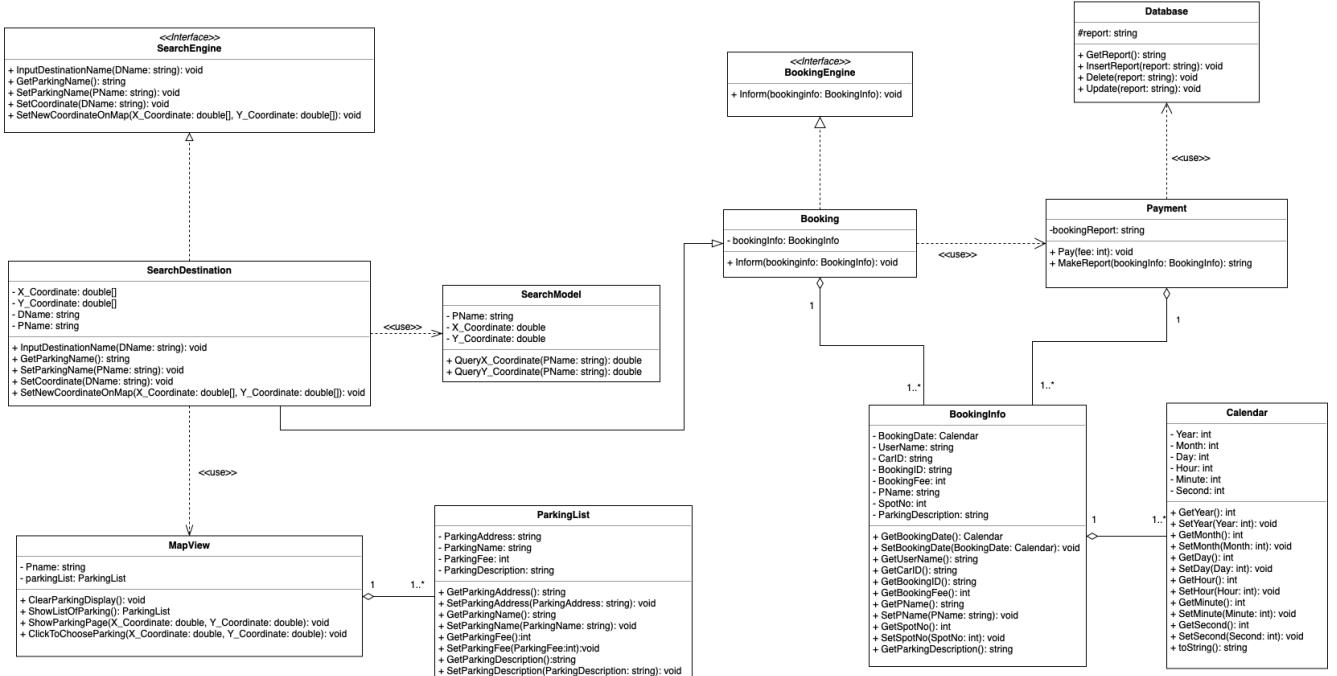


Figure 3: Class diagram of booking subsystem

### 3. Method description:

- Class: SearchDestination
- Attributes:
  - (private) X\_Coordinate: double[]: list of X coordinate with X\_Coordinate[0] is X coordinate of destination; X\_Coordinate[1..N] are X coordinates of parking areas near destination.
  - (private) Y\_Coordinate: double[]: list of Y coordinate with Y\_Coordinate[0] is Y coordinate of destination; Y\_Coordinate[1..N] are Y coordinates of parking areas near destination.
  - (private) PName: name of choiced parking area.
  - (private) DName: name of destination input.
- Methods:

Method name	Purpose
(public) InputDestinationName(DName: string): void	Users input the place where they want to arrive
(public) GetParkingName(): string	System return the parking area name which is recommended
(public) SetParkingName(PName: string): void	System assigns value of PName of class MapView to value of PName of class SeachDestination.
(public) SetCoordinate(X_Coordinate: double[], Y_Coordinate: double[]): void	System exchanges the destination name into coordinate with X_Coordinate[0] and Y_Coordinate[0]; System exchanges the list of parking names into coordinates with X_Coordinate[1..N] and Y_Coordinate[1..N]
(public) SetNewCoordinateOnMap(X_Coordinate: double[], Y_Coordinate: double[]): void	System marks each parking coordinate on the map with the figures in order to users can have a quick look of all parking areas near their destination.

- Class: MapView
- Attributes:
  - (private) PName: name of choiced parking area.
  - (private) parkingList: ParkingList: List of parking areas displayed on map
- Methods:

Method name	Purpose
(public) ClearParkingDisplay(): void	System resets the map
(public) ShowListOfParking(): ParkingList	System returns the list of parking areas near destination
(public) ShowParkingPage(X_Coordinate: double, Y_Coordinate: double): void	System displays parking page
(public) ClickToChooseParking(X_Coordinate: double, Y_Coordinate: double): void	When users click one parking, system will open that parking page

- Class: ParkingList
- Attributes:
  - (private) ParkingID: string: ID of the parking area
  - (private) ParkingName: string: Name of the parking area
  - (private) ParkingFee: int: fee of parking area
  - (private) ParkingDescription: string: brief description of the parking area

- Methods:

Method name	Purpose
(public) GetParkingAddress(): string	System returns the parking address
(public) SetParkingAddress(ParkingAddress: string): void	System assigns parameter ParkingAddress to ParkingAddress of class ParkingList
(public) GetParkingName(): string	System returns the parking area name
(public) SetParkingName(ParkingName: string): void	System assigns parameter ParkingName to ParkingName of class ParkingList.
(public) GetParkingFee(): int	System returns the parking fee
(public) SetParkingFee(ParkingFee: int): void	System assigns parameter ParkingFee to ParkingFee of class ParkingList.
(public) GetParkingDescription(): string	System returns the parking brief description
(public) SetParkingDescription(ParkingDescription: string): void	System assigns parameter ParkingDescription to Parkingscription of class ParkingList.

- Class: SearchModel

- Attributes:

- (private) PName: string: Name of place need be encoded into the coordinate
- (private) X\_Coordinate: double: X coordinate after encode
- (private) Y\_Coordinate: double: Y coordinate after encode

- Methods:

Method name	Purpose
(public) QueryX_Coordinate(PName: string): double	System queries and returns X coordinate of parameter PName from database
(public) QueryY_Coordinate(PName: string): double	System queries and returns Y coordinate of parameter PName from database

- Class: Booking

- Attribute:

- (private) bookingInfo: BookingInfo

- Method

Method name	Purpose
(public) Inform(bookingInfo: BookingInfo): void	System assigns parameter bookingInfo to bookingInfo of class Booking

- Class: BookingInfo

- Attributes:

- (private) BookingDate: Calendar: the date which users will park their car
- (private) UserID: string: ID of car owner
- (private) CarID: string: ID of car will be booked
- (private) BookingID: string: ID of booking
- (private) BookingFee: int: fee of final booking
- (private) PName: string: Parking area is booked
- (private) SpotNo: int: Spot is booked
- (private) ParkingDescription: string: Detail description of parking area

- Methods:

Method name	Purpose
(public) GetBookingDate(): Calendar	System returns the date which users will park their car
(public) SetBookingDate(BookingDate: calendar): void	System assigns parameter BookingDate to BookingDate of class BookingInfo
(public) GetUserName(): string	System returns name of car owner
(public) GetCarID(): string	System returns ID of car
(public) GetBookingID(): string	System returns ID of booking
(public) GetBookingFee(): int	System returns the fee of booking
(public) GetPName(): string	System returns the name of parking booked
(public) SetPName(PName: string): void	System assigns parameter PName to PName of class BookingInfo
(public) GetSpotNo(): int	System returns spot position booked
(public) SetSpotNo(SpotNo:int): void	System assigns parameter SpotNo to SpotNo of class BookingInfo
(public) GetParkingDescription(): string	System returns the detail description of parking booked

- Class: Payment
- Attribute:
  - (private) bookingReport: string: report file of booking
- Method

Method name	Purpose
(public) Pay(fee: int): void	Record the payment to database
(public) MakeReport(bookingInfo: BookingInfo): string	System exchange parameter bookingInfo into bookingReport of class Payment

- Class: Database
- Attribute:
  - (protect) report: string: report file of booking
- Method

Method name	Purpose
(public) GetReport(): string	System returns report file of booking
(public) InsertReport(report: string): void	System inserts new report file of booking to database
(public) DeleteReport(report: string): void	System finds and removes that report file of booking in database
(public) UpdateReport(report: string): void	System finds and updates that report file of booking in database

- Class: Calendar
- Attributes:
  - (private) year: int: year of booking time
  - (private) month: int: month of booking time
  - (private) day: int: day of booking time
  - (private) hour: int: hour of booking time
  - (private) minute: int: minute of booking time
  - (private) second: int: second of booking time

- Methods:

Method name	Purpose
(public) GetYear(): int	System returns year of booking time
(public) SetYear(Year:int): void	System assigns parameter Year to Year of class Calendar
(public) GetMonth(): int	System returns Month of booking time
(public) SetMonth(Month: int): void	System assigns parameter Month to Month of class Calendar
(public) GetDay(): int	System returns day of booking time
(public) SetDay(Day: int): void	System assigns parameter Day to Day of class Calendar
(public) GetHour(): int	System returns hour of booking time
(public) SetHour(Hour: int): void	System assigns parameter Hour to Hour of class Calendar
(public) GetMinute(): int	System returns minute of booking time
(public) SetMinute(Minuter: int): void	System assigns parameter Minute to Minute of class Calendar
(public) GetSecond(): int	System returns second of booking time
(public) SetSecond(Second: int): void	System assigns parameter Second to Second of class Calendar
(public) toString(): string	System returns time to string

#### 4. Sequence diagram:

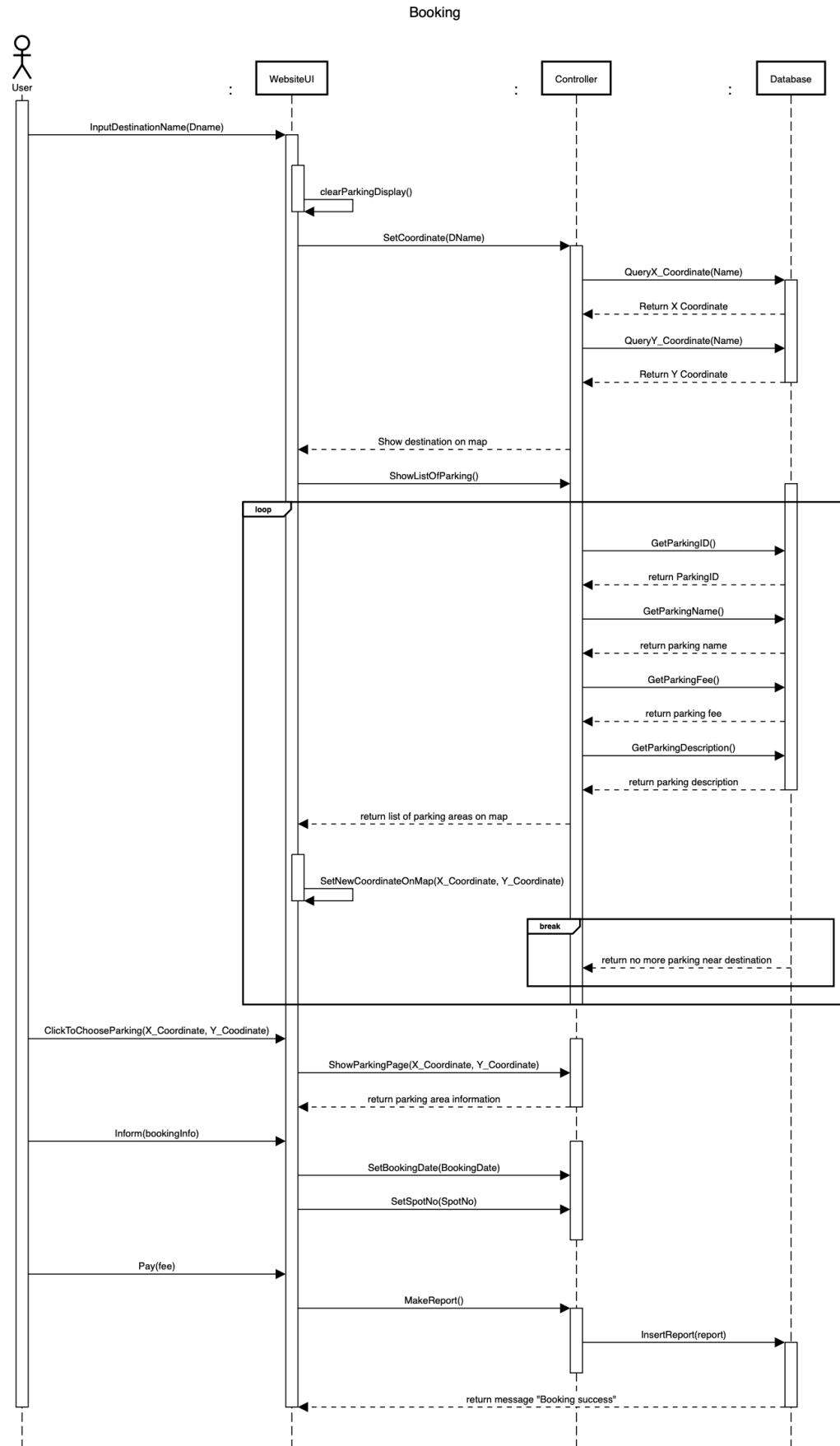


Figure 4: Sequence diagram of booking subsystem

## 5. Activity diagram:

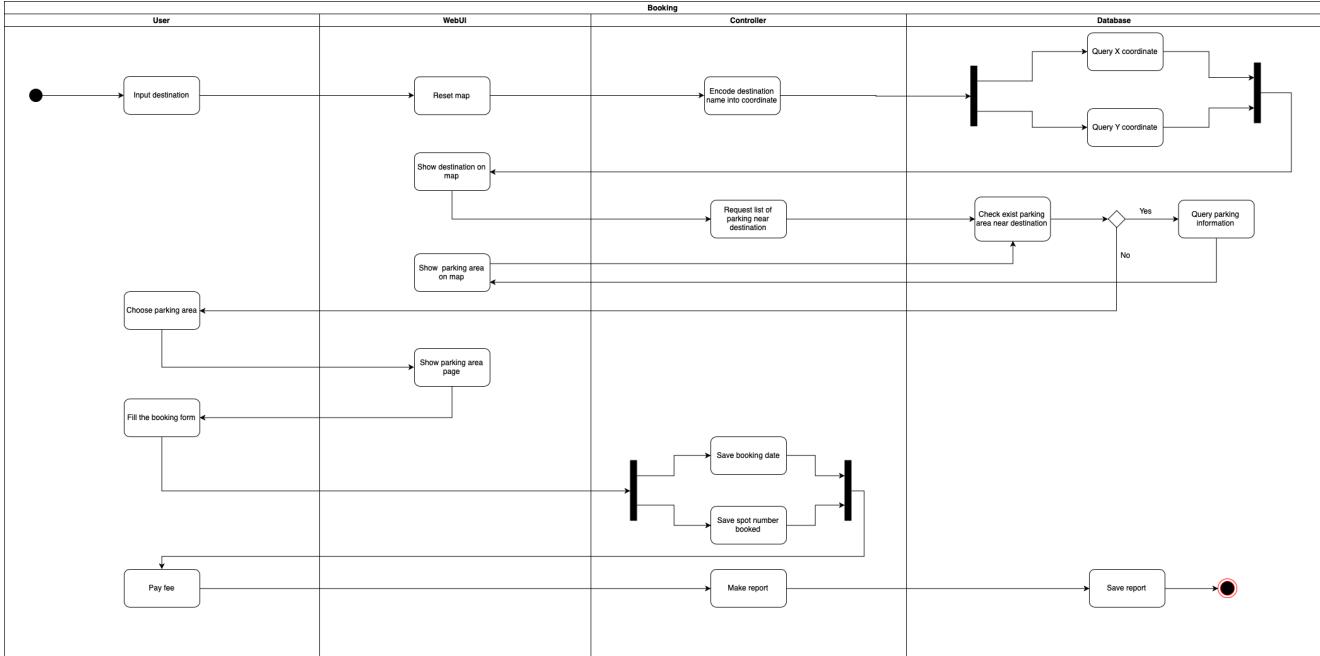


Figure 5: Activity diagram of booking subsystem

## 6. Design pattern:

In booking subsystem, I use MVC design pattern:

- View: I use class MapView, Booking, Payment to interact with user

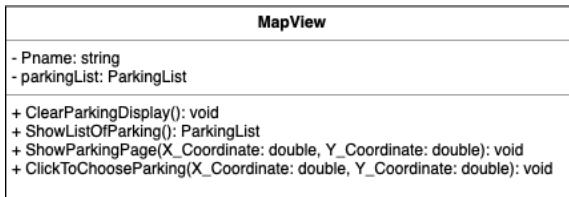


Figure 6: Class MapView



Figure 7: Class Booking

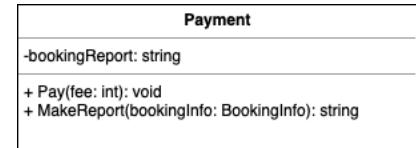


Figure 8: Class Payment

- Controller: I use class SearchDestination, BookingInfo, Calendar to process data and store temporary data

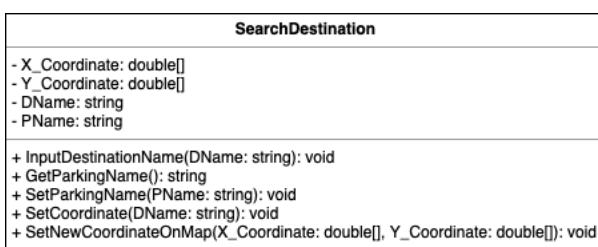


Figure 9: Class SearchDestination

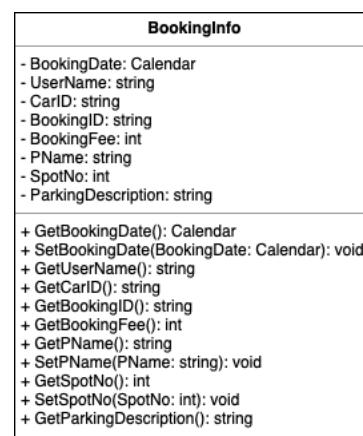


Figure 10: Class BookingInfo

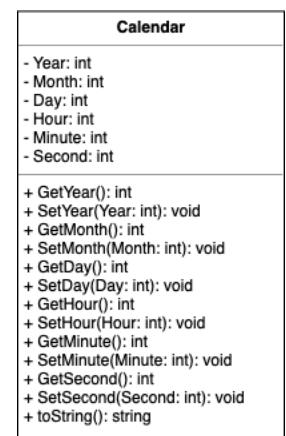


Figure 11: Class Calendar

- Model: I use class SearchModel, ParkingList, Database to interact with database

SearchModel
- PName: string - X_Coordinate: double - Y_Coordinate: double
+ QueryX_Coordinate(PName: string): double + QueryY_Coordinate(PName: string): double

Figure 12: Class SearchModel

ParkingList
- ParkingAddress: string - ParkingName: string - ParkingFee: int - ParkingDescription: string
+ GetParkingAddress(): string + SetParkingAddress(ParkingAddress: string): void + GetParkingName(): string + SetParkingName(ParkingName: string): void + GetParkingFee(): int + SetParkingFee(ParkingFee: int): void + GetParkingDescription(): string + SetParkingDescription(ParkingDescription: string): void

Figure 13: Class ParkingList

Database
#report: string
+ GetReport(): string
+ InsertReport(report: string): void
+ Delete(report: string): void
+ Update(report: string): void

Figure 14: Class Database

## 7. Demonstration:

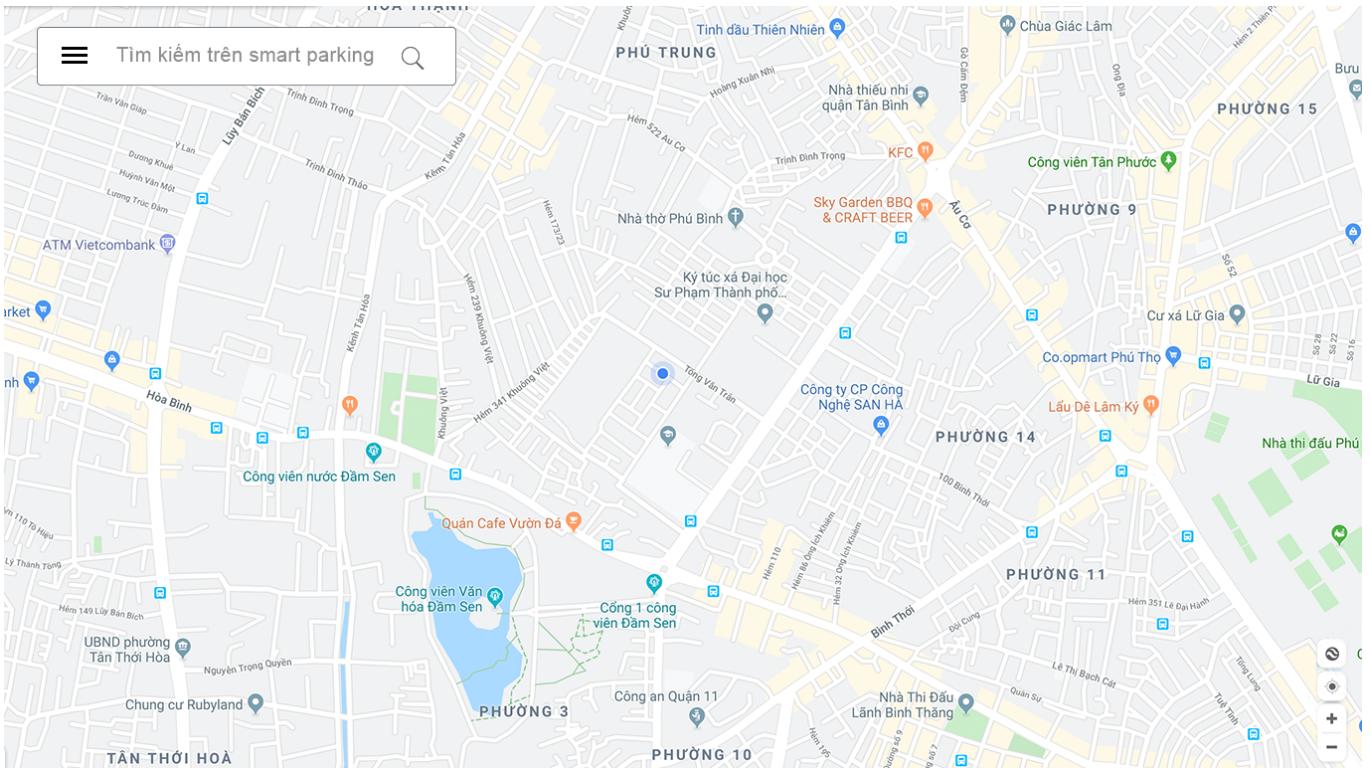


Figure 15: Map screen

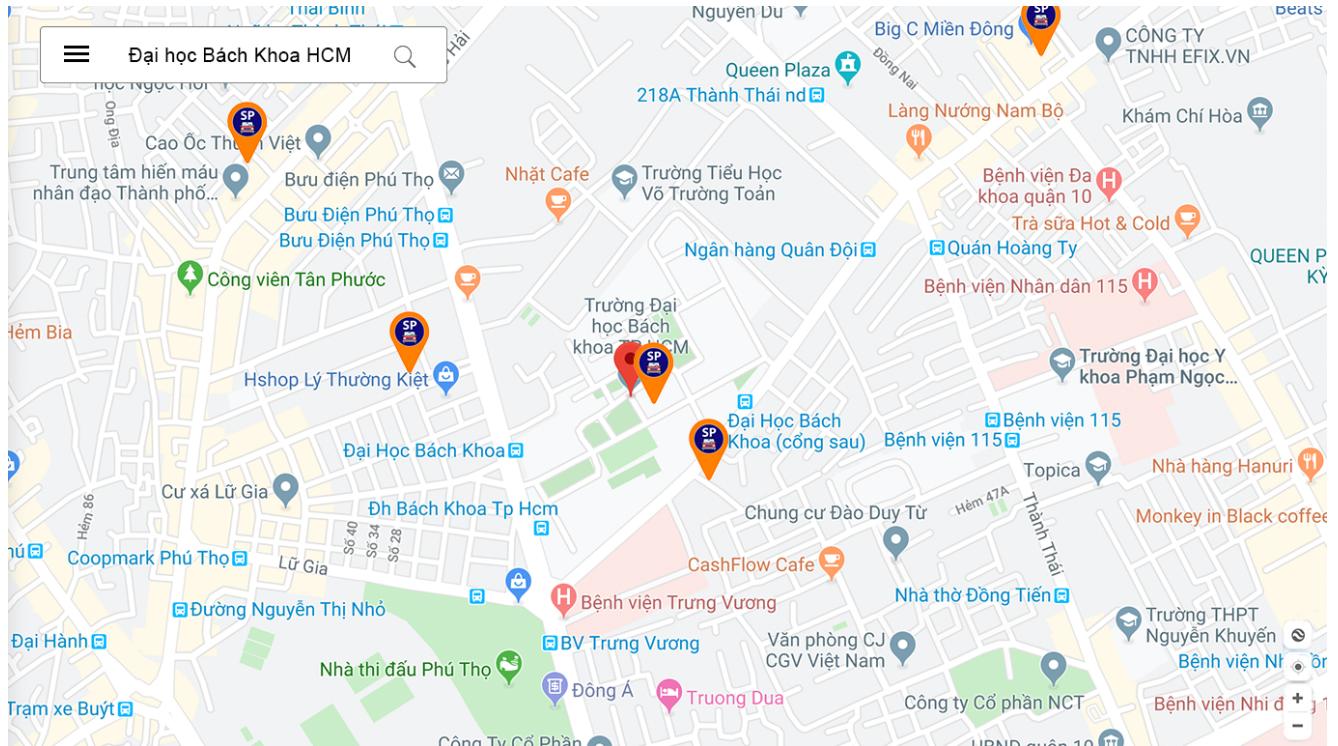


Figure 16: Search destination is HCM University of Technology

Figure 17: Parking page of HCMUT Parking

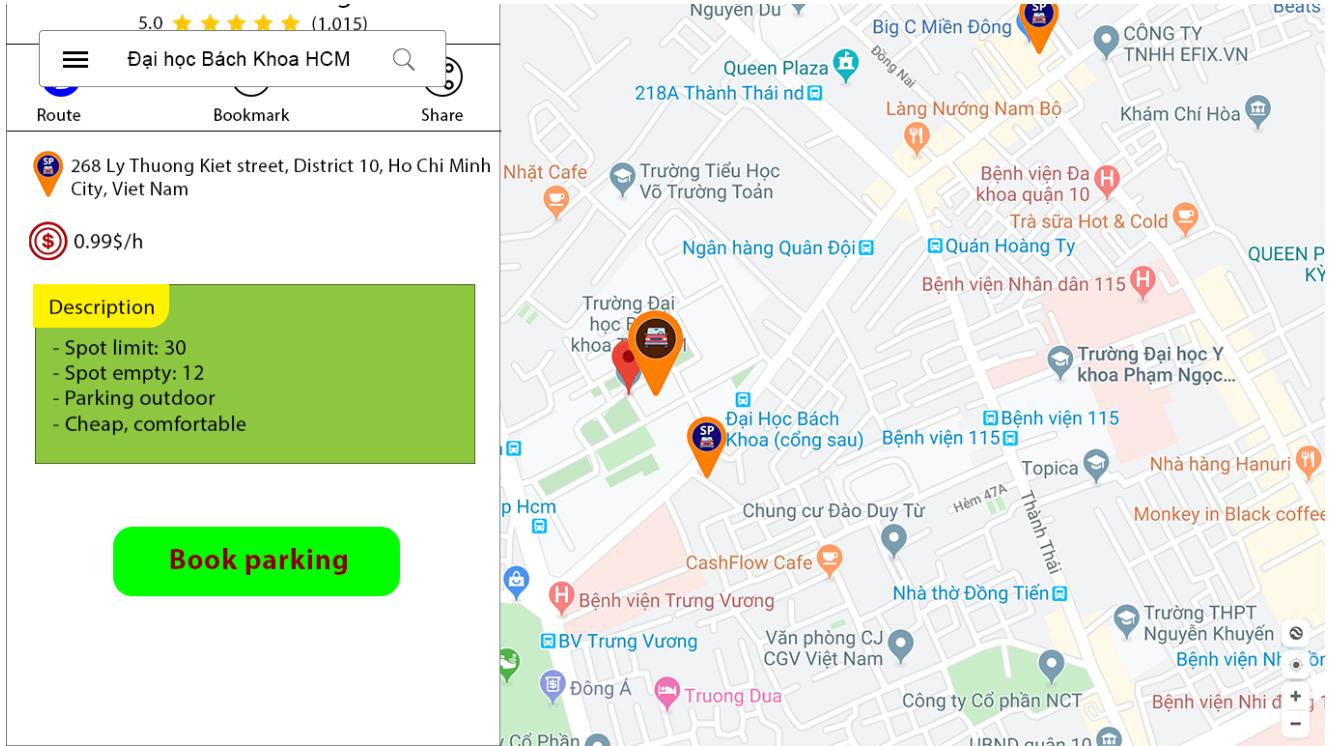


Figure 18: Click Book parking

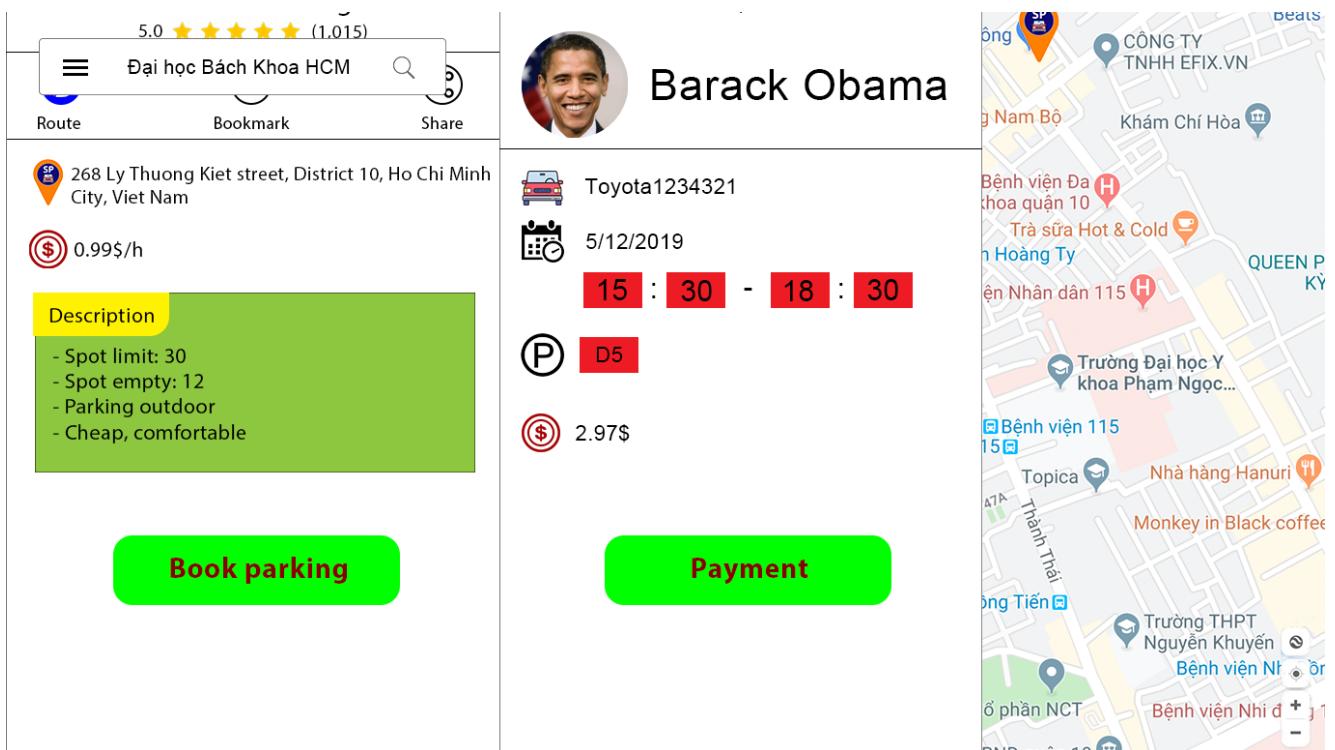


Figure 20: Fill the booking form and pay fee