|  |  |
| --- | --- |
| **Use case name:** | Print Ticket |
| **Summary:** | Actor is issued with a ticket to enter the Parking Deck. |
| **Actor:** | Driver |
| **Dependency:** | none |
| **Precondition:** | Parking has at least one spot free to issue a ticket |
| **Description of main sequence:** |  |
|  | 1. Driver presses the “Print Ticket” button on the gate. 2. System performs a check to make sure there is available space inside the Parking Deck. 3. System generates a unique permit id and records the timestamp. 4. System generates a barcode with the information in step 2 embedded in it. 5. System prints the information generated in step 2 and 3 on the ticket. 6. System ejects the ticket for the actor. 7. Driver pulls out the ticket from the slot. 8. Include use case Open Gate. |
| **Alternatives:** | **Step 2**: If the system finds that the parking is full, the use case terminates.  **Step 2**: If the parking is closed due to some reason, the use case terminates.  **Step 4**: If the system is unable to generate a barcode from available information, the flow skips to Step 5.  **Step 5**: If the printer is out of ink, an alert is issued to Maintenance and the use case terminates.  **Step 7**: If the Driver does not take the ticket from the slot, the system pulls the ticket back in and declares the permit id invalid. Tickets are stored inside the machine and a notification is sent to Maintenance if the storage compartment becomes full. |
| **Postcondition:** | A ticket with a valid transaction id is printed to the Driver. |

|  |  |
| --- | --- |
| **Use case name:** | Open Gate |
| **Summary:** | The system opens the gate for the Driver |
| **Actor:** | Driver, Maintenance, Operator |
| **Precondition:** | The system has instructed the gate to open for a defined period of time. |
| **Description of main sequence:** |  |
|  | 1. The system issues request to open the gate for the Driver. 2. The system records the change in state of the gate. 3. The system keeps the gate open for a predefined period of time. 4. Include use case Close Gate. |
| **Alternatives:** | **Step 1**: If the system is unable to issue the request, the Operator can override the system command and open the gate from controls.  **Step 2**: If the system does not record a change in the gate state, a notification is issued to Maintenance. |
| **Postcondition:** | The gate opens for the car to pass through. |

|  |  |
| --- | --- |
| **Use case name:** | Close Gate |
| **Summary:** | The system closes the gate |
| **Actor:** | Driver, Maintenance, Operator |
| **Precondition:** | Use case Open Gate has been executed successfully |
| **Description of main sequence:** |  |
|  | 1. The system issues request to close the gate. 2. The system records the change in state of the gate. 3. The system keeps the gate closed. 4. Include use case Update Entry Display. |
| **Alternatives:** | **Step 1**: If the system is unable to issue the request, the Operator can override the system command and close the gate from controls.  **Step 2**: If the system does not record a change in the gate state, a notification is issued to Maintenance. |
| **Postcondition:** | The gate closes. |

|  |  |
| --- | --- |
| **Use case name:** | Update Entry Display |
| **Summary:** | The system updates the vacancy on the display of the Parking Deck |
| **Actor:** | Driver, Maintenance, Operator |
| **Precondition:** | Use case Close Gate has been executed successfully |
| **Description of main sequence:** |  |
|  | 1. The system updates the vacancy variable to reduce it by one if a car has just left the Parking Deck or increase it by one if a car has just entered the Parking Deck. 2. The system sends an update to the Entry Display and the use case terminates. |
| **Alternatives:** | **Step 1**: If the system is unable to update the vacancy variable, the display will show the last updated value of the variable.  **Step 2**: If the update is not received properly by the Entry Display, then it will show the last updated value of the variable. |
| **Postcondition:** | Display on the parking Deck entrance shows the exact amount of parking spaces available inside. |

|  |  |
| --- | --- |
| **Use case name:** | Make Payment |
| **Summary:** | The driver pays for parking car in the Parking Deck |
| **Actor:** | Driver, Operator |
| **Dependency:** | none |
| **Precondition:** | Driver has the entry ticket |
| **Description of main sequence:** |  |
|  | 1. Driver scans the ticket over a barcode reader OR hands the ticket to the Operator who scans it with a barcode reader. 2. System reads the ticket information i.e. permit id and time stamp. 3. System calculates the fee based on this information and the current timestamp. 4. System displays the fee to the Actors. 5. Driver hands the Operator with cash or credit card. 6. <payment> 7. System displays a thank you message and prints the receipt. 8. Include use case Open Gate. |
| **Alternatives:** | **Step 1**: If the barcode reader does not work, the Operator can enter the permit id manually into the system and the flow returns to Step 2.  **Step 3**: If the system calculates a fee that is greater than the max fee, then the fee is set to the max fee value and flow returns to Step 4.  **Step 3**: If the system is unable to calculate the fee, the Operator can enter the fee value into the system manually and the flow returns to Step 4.  **Step 7**: If the printer is out of ink, an alert is issued to Maintenance and the flow returns to step 8. |
| **Postcondition:** | The Driver can leave the Parking Deck after paying the fee. |

|  |  |
| --- | --- |
| **Use case name:** | Pay by Credit Card |
| **Summary:** | The Driver pays the fee by credit card |
| **Actor:** | Driver, Operator |
| **Dependency:** | Extends Make Payment |
| **Precondition:** | The Driver has presented the ticket but not yet paid for it |
| **Description of main sequence:** |  |
|  | 1. Driver presents the credit card and the Operator swipes the card on the machine. 2. The system reads card ID and expiration date. 3. The system sends transaction to authorization center containing card ID, expiration date and payment amount. 4. If authorization is approved, authorization center returns a positive confirmation. 5. System displays payment amount and confirmation. 6. Flow returns to the dependent use case. |
| **Alternatives:** | **Step 2:** If the system does not recognize the card, the use case terminates.  **Step 3**: If the system is unable to contact the authorization center, the use case terminates.  **Step 4**: If the system does not receive authorization, the use case terminates. |
| **Postcondition:** | Driver has paid the fee by credit card. |

|  |  |
| --- | --- |
| **Use case name:** | Pay by Cash |
| **Summary:** | The Driver pays the fee by cash |
| **Actor:** | Driver, Operator |
| **Dependency:** | Extends Make Payment |
| **Precondition:** | The Driver has presented the ticket but not yet paid for it |
| **Description of main sequence:** |  |
|  | 1. Driver presents the cash and the Operator enters the amount of cash that is presented into the system. 2. The system calculates the change. 3. The system displays the total amount due, cash payment and change on the screen to the Actors. 4. Flow returns to the dependent use case. |
| **Alternatives:** | **Step 2**: If the system is unable to calculate the change, the Operator can enter the amount of cash received and change due manually.  **Step 3**: If the system is unable to display the information, then the flow skips to step 4. |
| **Postcondition:** | Driver has paid the fee by cash. |