

CS5373 TERM PROJECT REPORT

A HIGHWAY TOLL SYSTEM

Submitted by Group 13

Veena Posani - (R11844472)
Sai Charani Arumilli - (R11849421)
Mugdha Ektare – (R11841641)
Aditya Chundi – (R11797338)

Under the guidance of

Michael (Eonsuk) Shin, Ph.D.
Associate Professor



Texas Tech University
Department of Computer Science
Box 43104
Lubbock, TX 79409–3104

Table of Contents

Problem Description

Phase 1

1. Software context class diagram
2. Conceptual static model for highway toll system – Entity Classes
3. Interaction diagrams using sequence or communication diagram
4. State charts for use cases involving enter and exit highway

Phase 2

1. Integrated communication diagram
2. Software Architecture Diagram
3. Task Architecture
4. Task Interface Specifications

HIGHWAY TOLL SYSTEM: PROBLEM DESCRIPTION

Drivers who use the toll road may pay the tolls by using a transponder placed in their vehicle or by paying with cash or with credit cards at selected tollbooths.

Drivers wishing to use transponders purchase the transponders through the regional toll center. This regional toll center maintains driver accounts in a database including the owner, vehicle information, account balance, and toll use history. Drivers purchasing a transponder must pay ahead for toll fees by credit card. Accounts are reduced by the toll amount incurred at the end of each trip. The toll amount to be paid depends on the length of the trip.

For access to the toll highway, there is a ticket issuing booth with a barrier. The driver presses a button to receive the ticket. The driver with a transponder does not need to receive the ticket. To leave the toll highway, there is a human-operated booth with a barrier.

All tollbooths consist of an operational light to indicate whether the tollbooth is available for use; a vehicle-approaching sensor (placed 50 feet in front of the tollbooth); a vehicle-passed sensor; a traffic light to indicate whether the vehicle has been authorized to pass through the tollbooth; a transponder sensor. Additionally, all tollbooths contain a vehicle barrier that is raised and lowered to enforce payment. Exit tollbooths have a display to indicate the required toll amount. Exit tollbooths have a card reader device, which can read both tollbooth-issued tickets and credit cards, and an operator who collects the appropriate toll and authorizes a car to pass. After the driver inserts the ticket into the card reader or the system detects a transponder on a vehicle, the entry point and toll amount (which are read off the ticket) will be displayed on the operator's and driver's displays.

The traffic light at each tollbooth is initially red, and the barrier is lowered. When a vehicle approaches the tollbooth, the vehicle sensor detects the vehicle's presence. Drivers not possessing a transponder or having insufficient funds on their transponder must use pay with cash or with credit cards. In each of these cases, the toll is calculated based on entry and exit times, and entry and exit locations. The driver can pay by credit card or pay an operator who manually collects the required toll and authorizes the vehicle to pass. If a toll payment is successful, the light will be turned green, and the barrier will be raised. The traffic light is switched to red, and the barrier is lowered after the car passes.

PHASE 1

1. SYSTEM CONTEXT CLASS DIAGRAM

Objects used in highway toll system:

Actors:

1. Driver
2. Vehicle
3. Bank
4. Operator
5. Regional toll center

External Objects:

- | | |
|----------------------------|------------------------------|
| 1. Driver UI | 10. Exit Traffic Light |
| 2. Card Reader | 11. Entry Barrier |
| 3. Driver Display | 12. Exit Barrier |
| 4. Operational light | 13. Entry Transponder Sensor |
| 5. Entry traffic light | 14. Entry Passing sensor |
| 6. Ticket button | 15. Entry Approaching Sensor |
| 7. Ticket Printer | 16. Exit Transponder Sensor |
| 8. Receipt Printer | 17. Exit Passing sensor |
| 9. Exit Approaching Sensor | 18. Operator Display |

System:

1. Highway Toll System

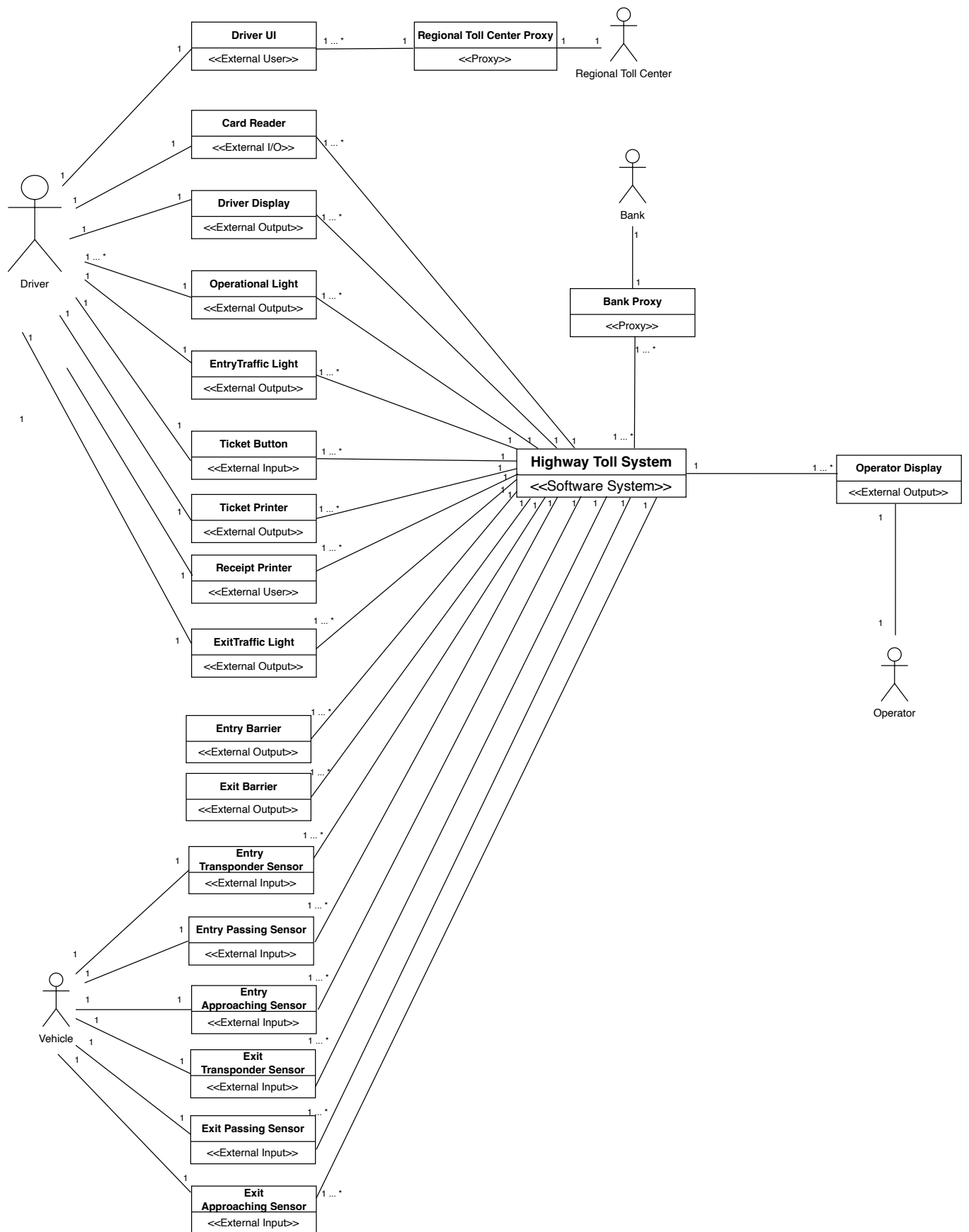
Proxies:

1. Regional Toll Center Proxy
2. Bank Proxy

Assumptions made:

1. Driver and external object interactions are only for one instance/scenario
2. Vehicle and external object interactions are only for one instance/scenario

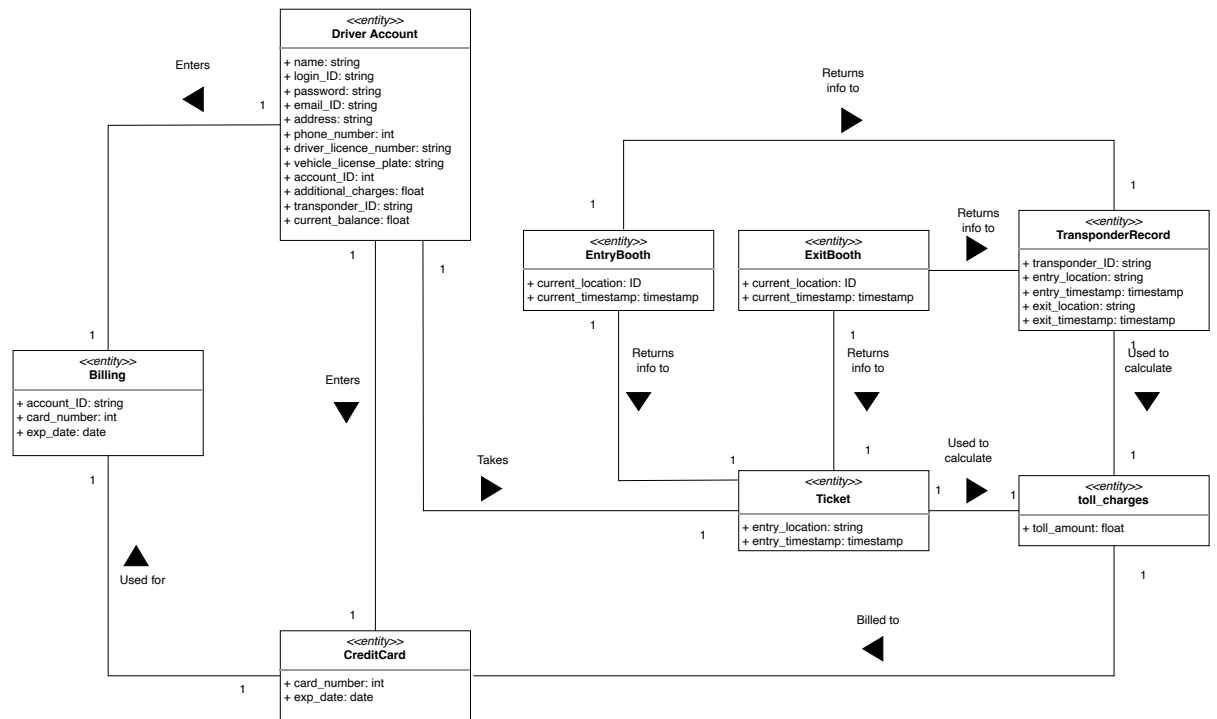
The diagram in next page is the software system context class model depicted on a class diagram showing how the system interfaces to the external environment:



2. CONCEPTUAL STATIC MODEL FOR HIGHWAY TOLL SYSTEM -ENTITY CLASS

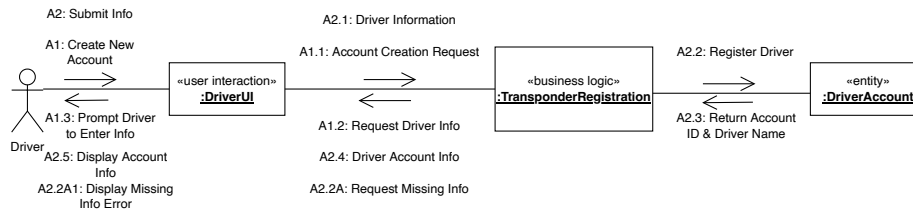
Assumptions made:

1. Entity booth stores the number of tickets available in the ticket printer for alternative sequence in use case.



3. INTERCATION DIAGRAMS USING SEQUENCE OR COMMUNICATION DIAGRAM

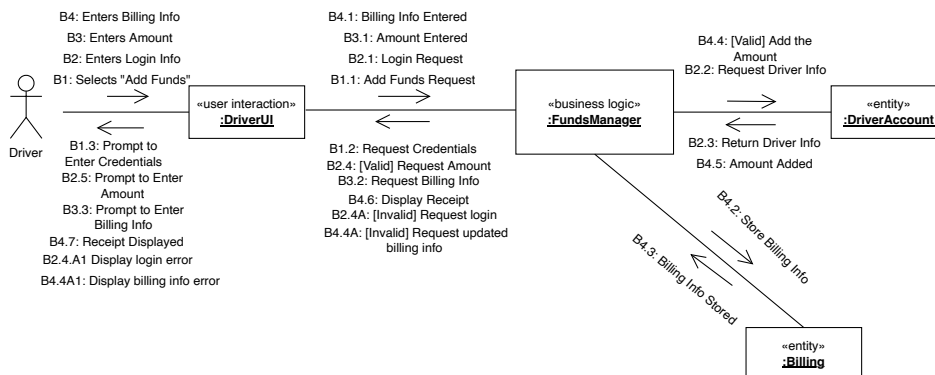
Use Case 1: Purchase Transponder



Use Case 2: Add Funds

Assumptions made:

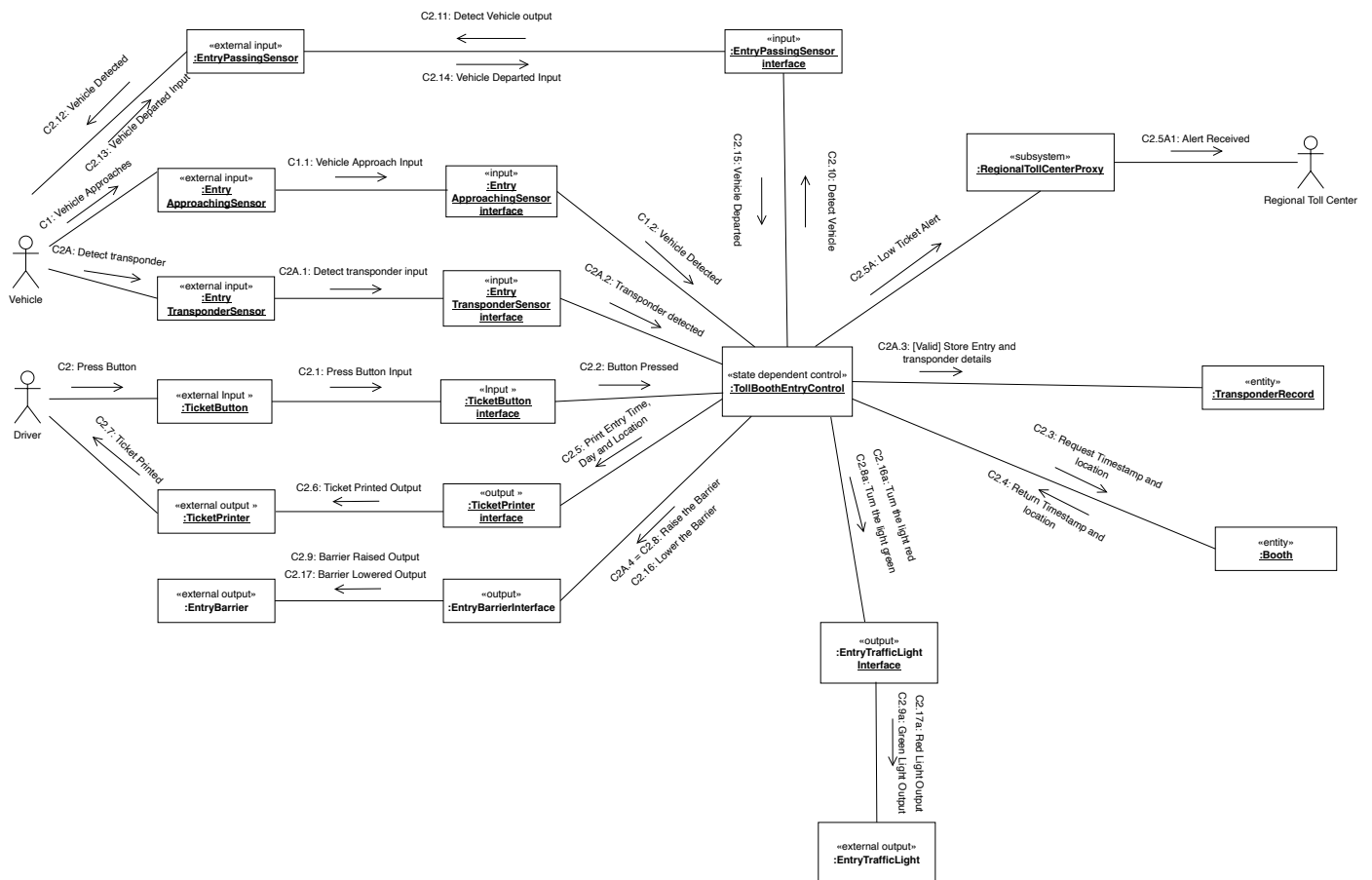
1. Only one transponder exits per driver account
2. Assuming customer entered multiples of 10\$ in enter amount.



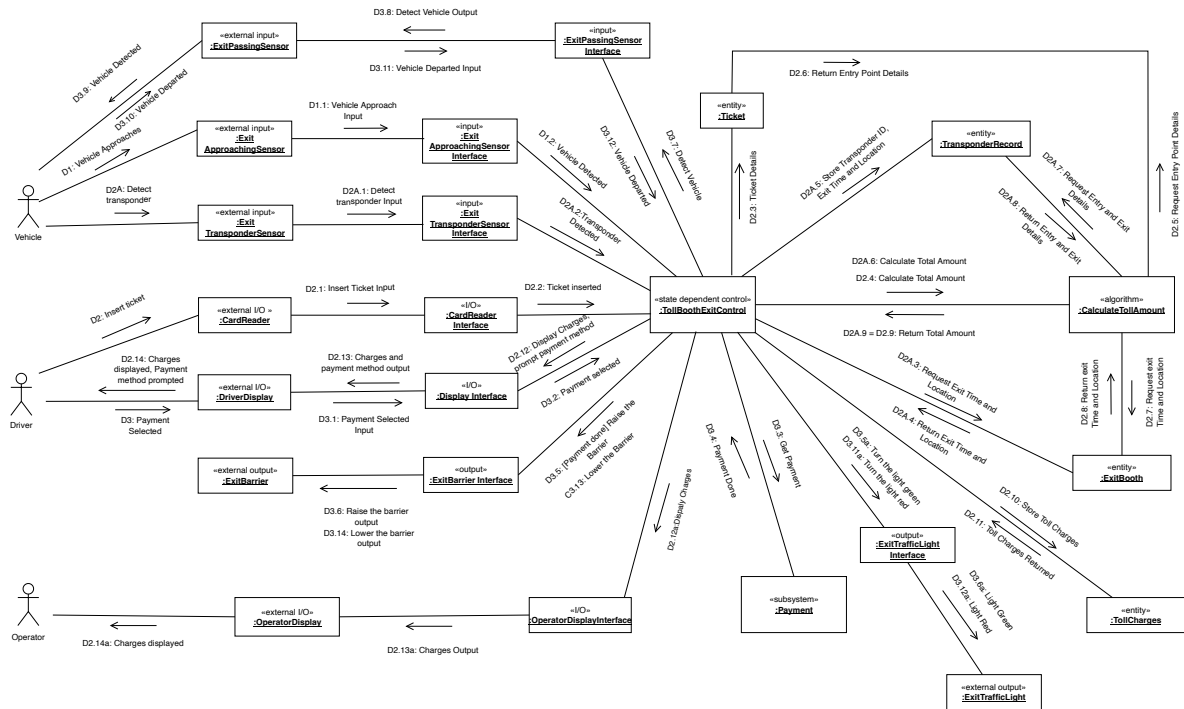
Use Case 3: Entry Highway

Assumptions made:

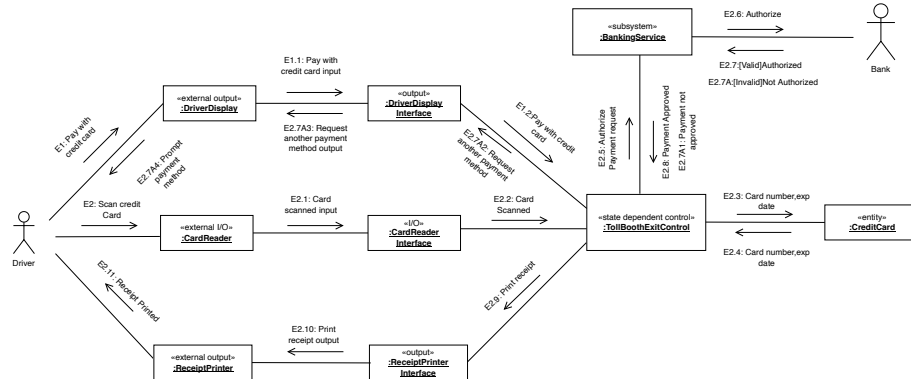
1. For low on tickets alert alternative step, entry booth has number of tickets and system checks this entity to get number of tickets left count when it requests time and location.
2. Post condition assumption – if system running low on tickets, driver won't get ticket.



Use Case 4: Exit Highway



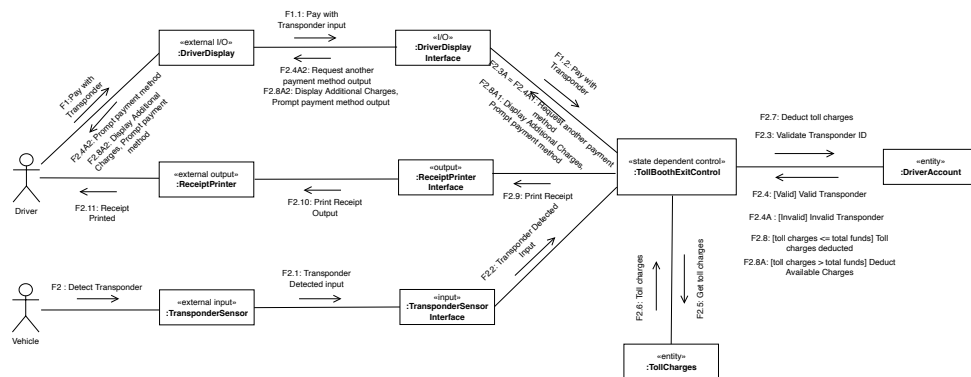
Use Case 5: Pay with Credit Card



Use Case 6: Pay with Transponder

Assumptions made:

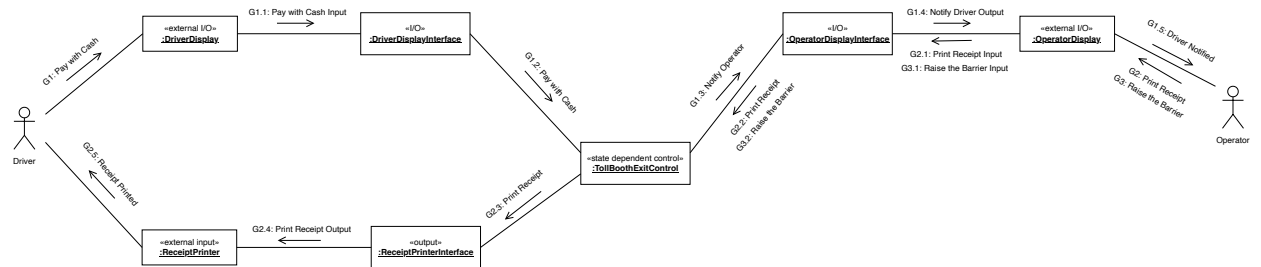
1. Only one active transponder exits per driver account



Use Case 7: Pay with Cash

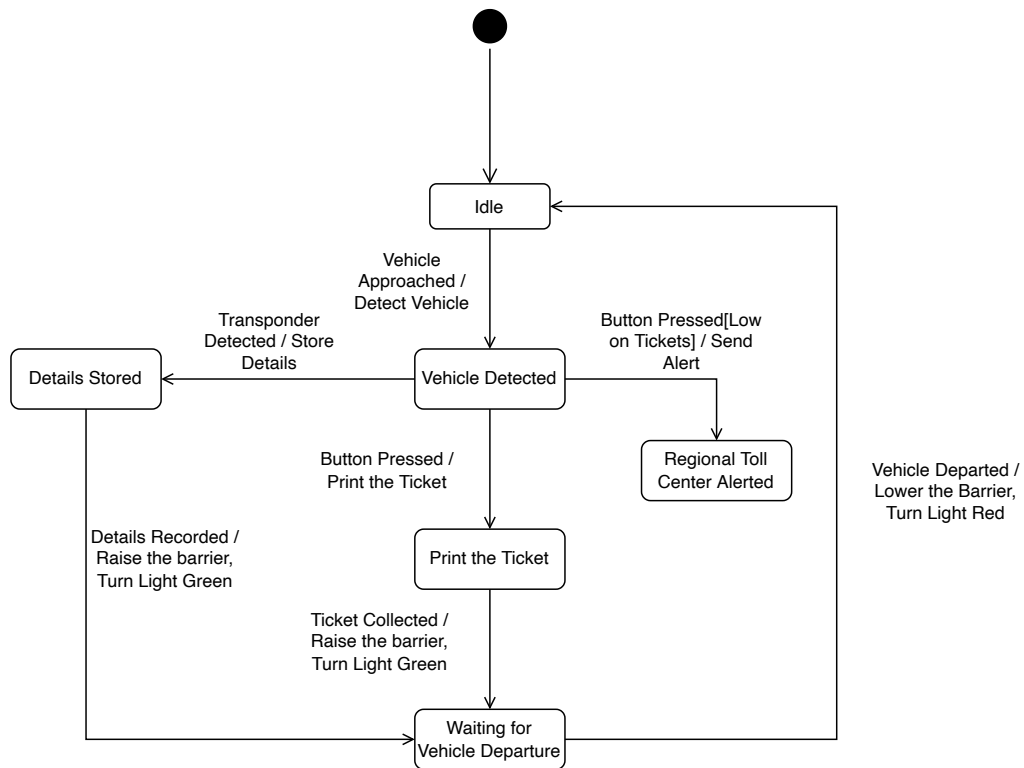
Assumptions made:

1. Notify operator that customer selected pay with cash.

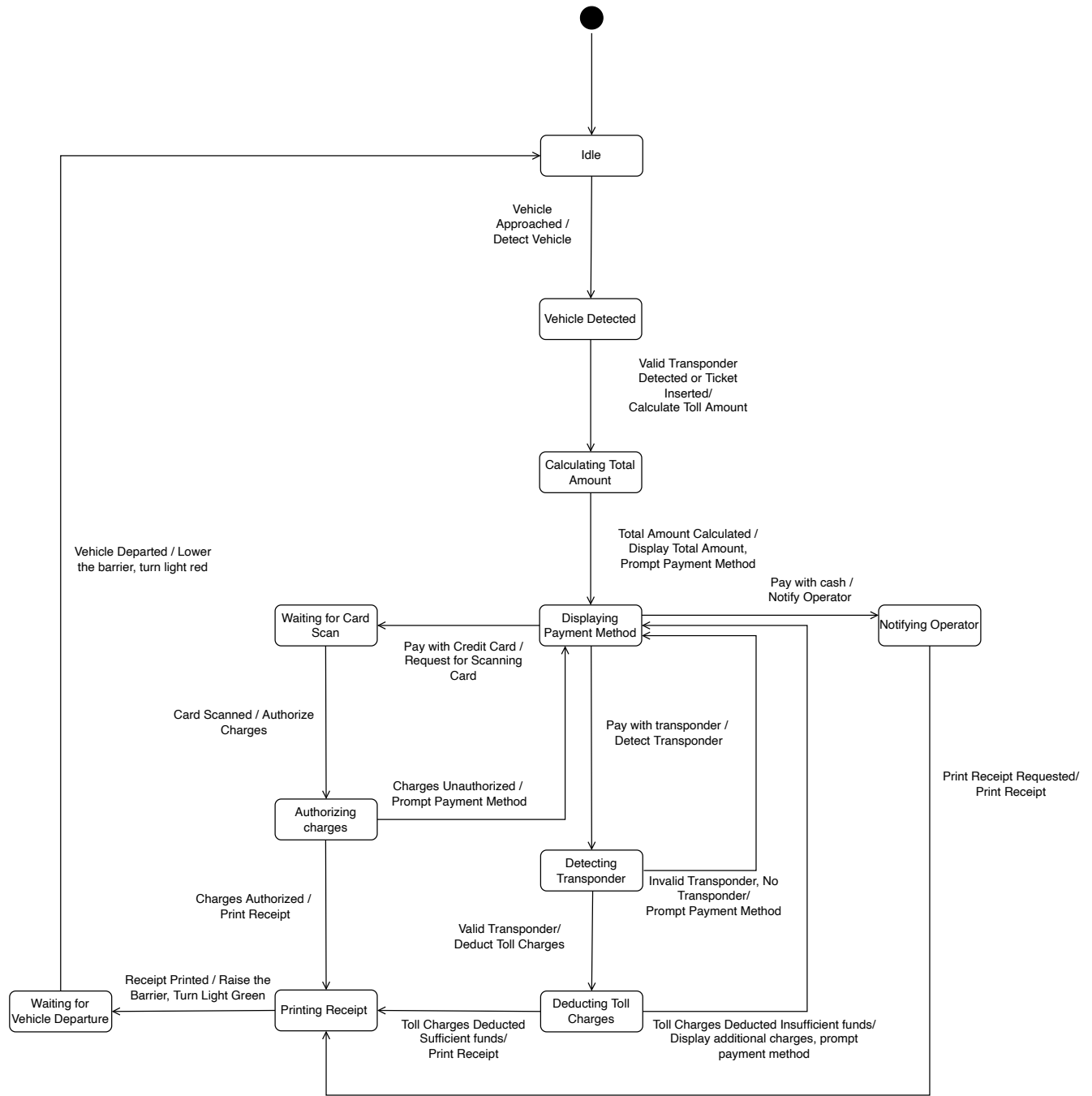


4. STATECHARTS FOR USECASES INVOLVING ENTER AND EXIT HIGHWAY

State Chart for Enter Highway:

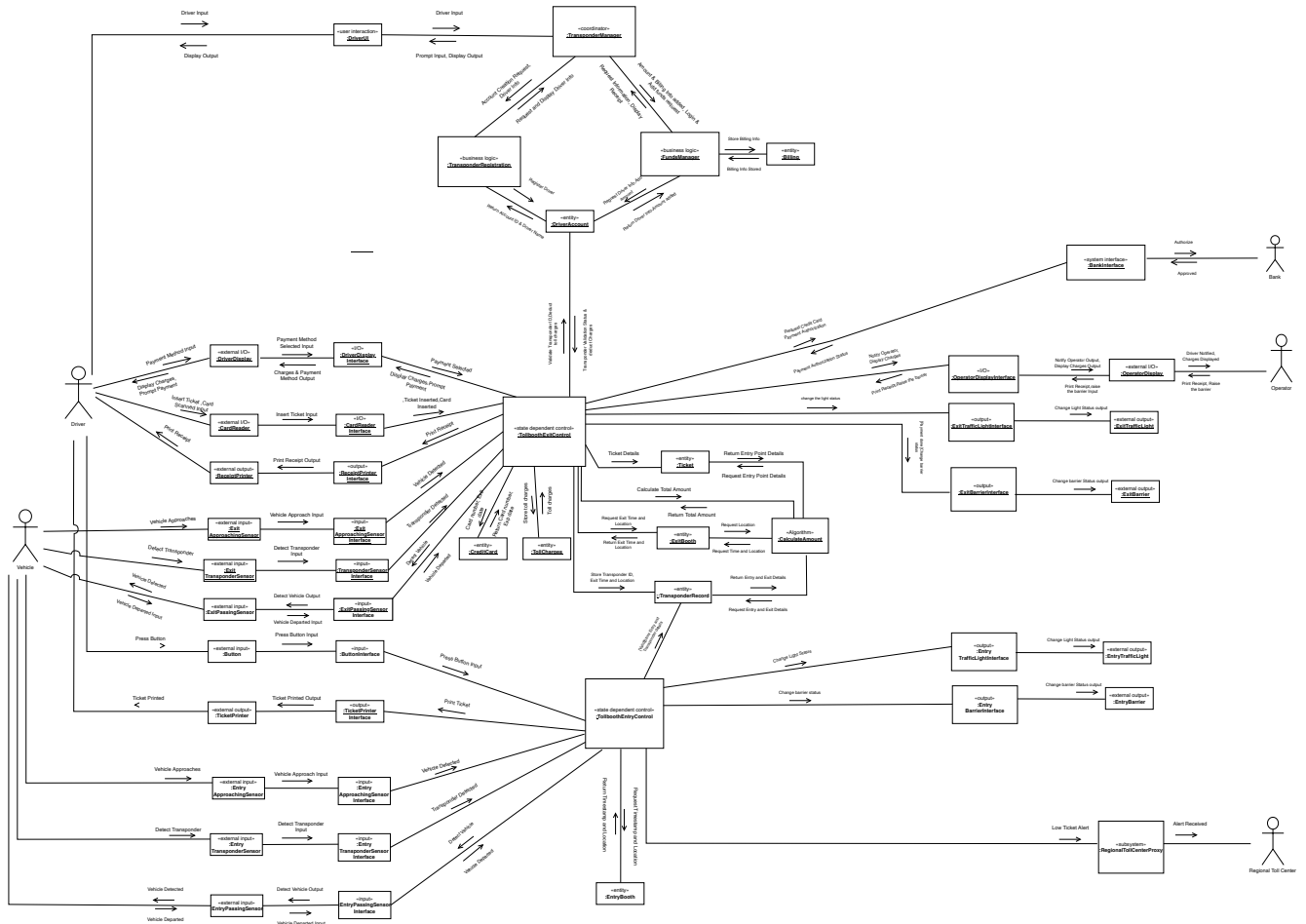


State Chart for Exit Highway:



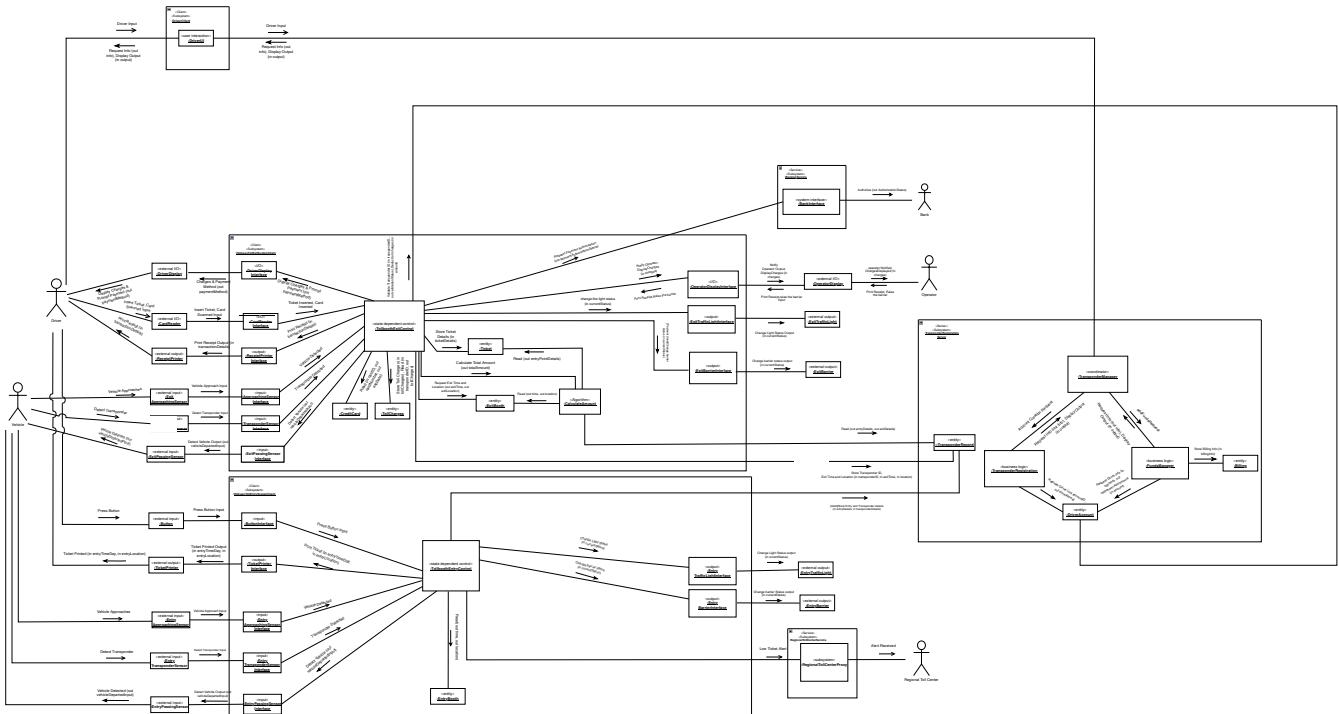
PHASE 2

1. INTEGRATED COMMUNICATION DIAGRAM

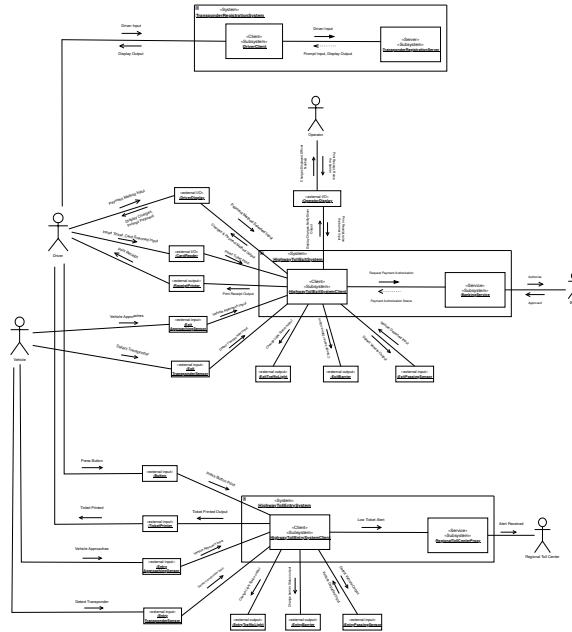


2. SOFTWARE ARCHITECTURE DIAGRAM

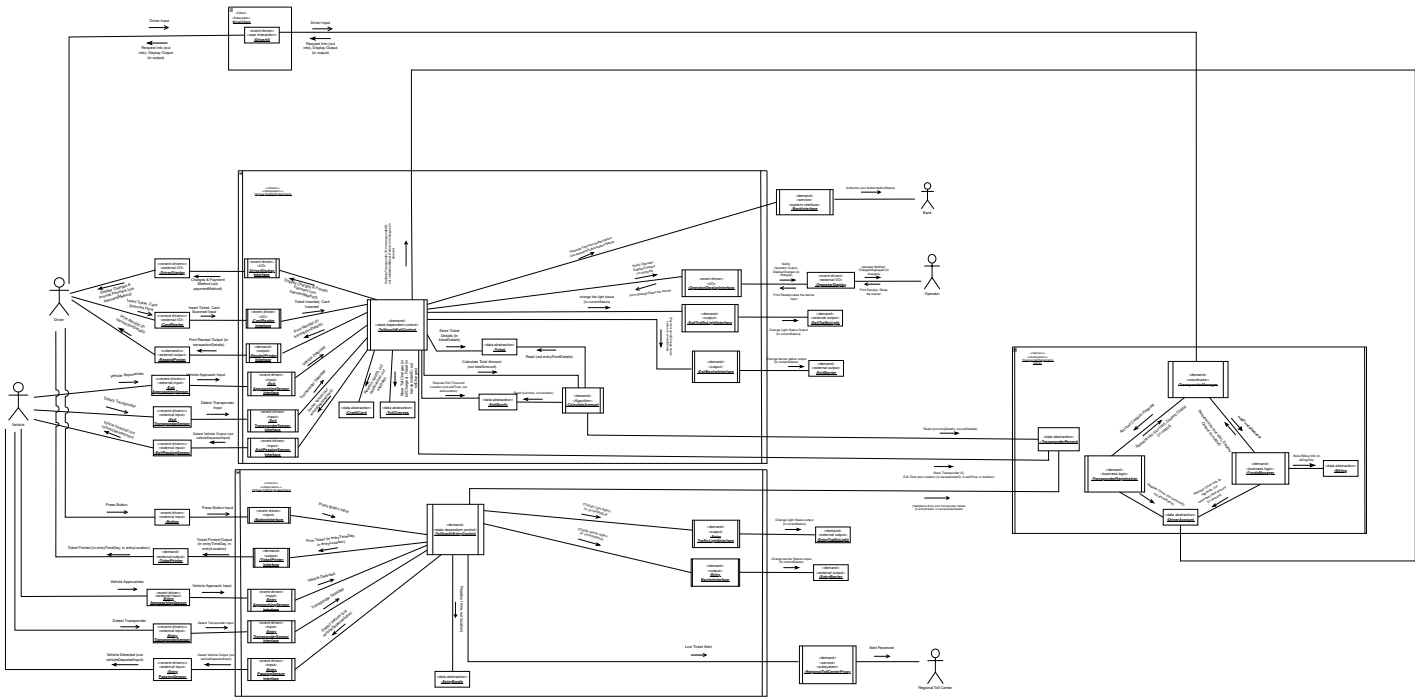
Software Architecture showing Client and Server of the System:



Message Communication Interfaces Between Client and Server:



3. TASK ARCHITECTURE



CRITERIA USED FOR TASK STRUCTURING:

Event Driven Input task:

- Exit Approaching Sensor Interface
- Exit Transponder Sensor Interface
- Exit Passing Sensor Interface
- Button Interface
- Entry Transponder Sensor Interface
- Entry Passing Sensor Interface

Event Driven Input/Output task:

- Driver Display Interface
- Card Reader Interface
- Operator Display Interface

Demand Driven Output task:

- Receipt Printer Interface
- Exit Traffic Light Interface
- Exit Barrier Interface
- Ticket Printer Interface
- Entry Traffic Light Interface
- Entry Barrier Interface

Demand Driven State Dependent Control task:

- Tool booth Exit Control
- Tool booth Entry Control

Demand Driven Algorithm task:

- Calculate Amount

Demand Driven Co-Ordinator task:

- Transponder Manager

Demand Driven BusinessLogic task:

- Transponder Registration
- Funds Manager

Event Driven User Interaction task:

- DriverUI

Demand Driven Service System Interface task:

- Bank Interface

Demand Driven Service Subsystem task:

- Regional Toll Center Proxy

4. TASK INTERFACE SPECIFICATIONS

Name: DriverDisplayInterface

Structuring Criteria: Role Criterion: input/output; Concurrency Criterion: event driven

Task Interface:

Task Inputs:

Event Input: Payment method selected from driver.

External Input: Payment Method

Synchronous message communication with reply:

- Messages - Prompt Payment

Reply: Payment Method

Synchronous message communication without reply:

- Display Charges

Task Outputs:

External Output: Charges, Prompt Payment Method

Message replies as described previously.

Name: CardReaderInterface

Structuring Criteria: Role criterion: input; Concurrency criterion: event driven

Task Interface:

Task Inputs:

Event Input: Driver inserts either a ticket or card.

External Input: Insert Ticket, Card Scanned Input

Task Outputs:

External Output: None

Asynchronous message communication:

- Ticket Inserted

- Card Inserted

Name: ReceiptPrinterInterface

Structuring Criteria: Role Criterion: output; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

Event Input: None

External Input: None

Synchronous message communication without reply:

- Messages:
 - Print Receipt

Task Outputs:

External Output: Print Receipt Output

Name: ExitApproachingSensorInterface

Structuring Criteria: Role Criterion: input; Concurrency Criterion: event driven

Task Interface:

Task Inputs:

Event Input: Vehicle approaches near to exit toll booth.

External Input: Vehicle Approach Input

Task Outputs:

External Output: None

Asynchronous message communication:

- Messages:
 - Vehicle Detected
-

Name: ExitTransponderSensorInterface

Structuring Criteria: Role Criterion: input; Concurrency Criterion: event driven

Task Interface:

Task Inputs:

Event Input: Detecting transponder from the vehicle approached.

External Input: Detect Transponder Input

Task Outputs:

External Output: None

Asynchronous message communication:

- Messages:
 - Transponder Detected
-

Name: ExitPassingSensorInterface

Structuring Criteria: Role Criterion: input; Concurrency Criterion: event driven

Task Interface:

Task Inputs:

Event Input: Detecting passing vehicle

External Input: Vehicle departed

Synchronous message communication with reply:

- Messages:

- Detect Vehicle

Reply: vehicle departed

Task Outputs:

External Output: None

Message replies as described previously.

Name: TransponderManager [Transponder Registration Server]

Structuring Criteria: Role Criterion: coordinator; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

External Input: none

Asynchronous message communication:

- Messages:
 - Driver Input

Synchronous message communication without reply:

- Messages:
 - Display Output
- Synchronous message communication with reply:
 - Messages:
 - Request Driver Information
 - Reply – Driver information
- Request Add funds information
- Reply – Amount and billing information

Task Outputs:

External Output: none

Message replies as described previously.

Synchronous message communication without Reply:

- Messages:
 - Account Creation Request
 - AddFundsRequest
 - Display Output

Name: TransponderRegistration [Transponder Registration Server]

Structuring Criteria: Role Criterion: business logic; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

External Input: None

Synchronous message communication without reply:

- Messages:
 - Account Creation Request

Task Outputs:

External Output: None

Synchronous message communication with Reply:

- Messages:
 - Request Info
- Reply: Driver Information
 - Register Driver

Reply : return AccountId, DriverName

Synchronous message communication without Reply:

- Messages:
 - Display Output

Name: FundsManager [Transponder Registration Server]

Structuring Criteria: Role Criterion: business logic; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

External Input: None

Synchronous message communication without reply:

- Add Funds Request

Task Outputs:

External Output: None

Synchronous message communication with Reply:

- Messages: Request Driver Information

Reply – return Driver Info

Synchronous message communication without Reply:

- Messages:
 - Store Billing Info
 - Add amount
-

Name: CalculateAmount

Structuring Criteria: Role Criterion: Algorithm; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

External Input: None

Synchronous message communication with reply:

- Messages -
 - Calculate Total Amount:

Reply – Calculated Toll amount

Task Outputs:

Message replies as described previously

External Output: None

Synchronous message communication with Reply:

- Messages:
 - Ticket Entry point Details

Output – Ticket Entry time and location

- Transponder details

Output – Transponder Entry and exit details

- Exit booth details

Output – Read exit time and location

Name: OperatorDisplayInterface

Structuring Criteria: Role Criterion: input/output; Concurrency Criterion: event driven

Task Interface:

Task Inputs:

Event Input: Once driver completed paying by cash, Operator requests system to print receipt to driver and raise the barrier.

External Input: Print Receipt, Raise the barrier input

Synchronous message communication without reply:

- Messages:
 - Notify Operator
 - DisplayCharges

Task Outputs:

External Output: Notify Operator, Display total charges

Asynchronous message communication:

- Messages:
 - Print Receipt
 - Raise the barrier
-

Name: ExitTrafficLightInterface

Structuring Criteria: Role Criterion: output; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

External Input: None

Synchronous message communication without reply:

- Messages:
 - change the light status

Task Outputs:

External Output: Change traffic light status output

Name: ExitBarrierInterface

Structuring Criteria: Role Criterion: output; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

External Input: None

Synchronous message communication without reply:

- Messages:
 - Change barrier status

Task Outputs:

External Output: Change barrier status output

Name: BankInterface

Structuring Criteria: Role Criterion: system service; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

External Input: None

Synchronous message communication with reply:

- Messages:
 - Request Payment Authorization

Input parameters – Validate CreditCard

Reply – CreditCard Payment Authorized

Task Outputs:

External Output: Authorize the Credit card payment

Name: ButtonInterface

Structuring Criteria: Role Criterion: input; Concurrency Criterion: event driven

Task Interface:

Task Inputs:

Event Input: When driver presses the button

External Input: Press Button

Task Outputs:

External Output: None

Asynchronous message communication:

- Messages:
 - Press Button Input
-

Name: TicketPrinterInterface

Structuring Criteria: Role Criterion: output ; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

External Input: None

Synchronous message communication without reply:

- Messages:
 - Print Ticket

Task Outputs:

External Output: Ticket Printed Output

Name: EntryApproachingSensorInterface

Structuring Criteria: Role Criterion: input; Concurrency Criterion: event driven

Task Interface:

Task Inputs:

Event Input: Vehicle approaches near to entry tool booth.

External Input: Vehicle Approach Input

Task Outputs:

External Output: None

Asynchronous message communication:

- Messages:
 - Vehicle Detected
-

Name: TollBoothExitControl

Structuring Criteria: Role Criterion: state dependent control; Concurrency Criterion: demand

Task Interface:

Task Inputs:

External Input: None

Task Outputs:

External Output: None

Asynchronous message communication:

- Messages:
 - Vehicle Detected
-

Name: DriverUI

Structuring Criteria: Role Criterion: user interaction; Concurrency Criterion: event driven

Task Interface:

Task Inputs:

Event Input: Driver request to create account and add funds in transponder.

External Input: Driver Input

Synchronous message communication with reply:

- Messages:
 - Request information
- Reply – driver information response

Task Outputs:

Message replies as described previously.

Name: EntryTransponderSensorInterface

Structuring Criteria: Role Criterion: input; Concurrency Criterion: event driven

Task Interface:

Task Inputs:

Event Input: Detecting transponder from the vehicle approached.

External Input: Detect Transponder Input

Task Outputs:

External Output: None

Asynchronous message communication:

- Messages:
 - Transponder Detected
-

Name: EntryPassingSensorInterface

Structuring Criteria: Role Criterion: input; Concurrency Criterion: event driven

Task Interface:

Task Inputs:

Event Input: Detecting passing vehicle

External Input: None

Synchronous message communication with reply:

- Messages:
 - detect vehicle

Reply – vehicle departed.

Task Outputs:

External Output: Detect vehicle output

Name: EntryTrafficLightInterface

Structuring Criteria: Role Criterion: output; Concurrency Criterion: demand

Task Interface:

Task Inputs:

External Input: None

Synchronous message communication without reply:

- Messages:
 - change the light status

Task Outputs:

External Output: Change traffic light status output

Name: EntryBarrierInterface

Structuring Criteria: Role Criterion: output; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

External Input: None

Synchronous message communication without reply:

- Messages:
 - Change barrier status

Task Outputs:

External Output: Change barrier status output

Name: RegionalTollCenterProxy

Structuring Criteria: Role Criterion: system service; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

External Input: None

Asynchronous message communication:

- Messages:
 - Low ticket alert

Task Outputs:

External Output: Alert received

Name: TollboothExitControl

Structuring Criteria: Role Criterion: state dependent control; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

External Input: None

Asynchronous message communication:

- Messages:
 - Ticket inserted
 - Card inserted
 - Vehicle detected
 - Transponder detected
 - Print receipt command from operator
 - Raise the barrier command from operator

Task Outputs:

External Output: none

Synchronous message communication without reply:

- Messages:
 - Print receipt
 - Store toll charges
 - Store ticket details
 - Change barrier status
 - Change light status
 - Notify Operator
 - Display Charges to operator
 - Deduct toll charges
 - Store Transponder ID, Exit time and location

Synchronous message communication with reply:

- Messages:
 - Validate Transponder ID
 - Reply – Validation status
 - Display charges and prompt payment method
 - Reply – Payment method
 - Detect vehicle
 - Reply – Vehicle departed input
 - Request Credit Card details
 - Reply – Card number and expiration date
 - Request toll charges
 - Reply – Toll Charges
 - Request Exit time and location
 - Reply – Exit time and location
 - Calculate total amount
 - Reply Total amount
 - Request Payment authorization
 - Reply – Payment authorization status
-

Name: TollboothEntryControl

Structuring Criteria: Role Criterion: state dependent control; Concurrency Criterion: demand driven

Task Interface:

Task Inputs:

External Input: None

Asynchronous message communication:

- Messages:
 - Press button input
 - Vehicle detected
 - Transponder detected

Task Outputs:

External Output: none

Asynchronous message communication:

- Messages:
 - Low ticket alert

Synchronous message communication without reply:

- Messages:
 - Print ticket
 - Store entry and transponder details
 - Change barrier status
 - Change light status

Synchronous message communication with reply:

- Messages:
 - Detect vehicle
- Reply – Vehicle departed input
 - Request Entry time and location
- Reply – Entry time and location