**CS5373-Software Modelling and Architecture**

**Term Project: Highway Toll System**

**Team Members:**

Subhashchandra Muchchunoor -- R11850821

Raja Venkatesh Gurugubelli -- R11850660

Hemanth Reddy Nagireddy -- R11804804

Venkata Ramana Sathivada -- R11850665

Your group has been assigned to a project to design a client/server system for a Highway Toll

System. The system is developed in two phases as follows:

**Term Project: Phase I**

For Phase 1 of the project, you are required to develop an analysis model of the system. In

particular:

a) Develop a software system context class model depicted on a class diagram showing how the

system interfaces to the external environment. (3 pts)

Chart, radar chart

Description automatically generated

b) Develop a static model showing the entity classes in the system, the attributes of the classes, and the relationships between them. (3 pts)

Diagram

Description automatically generated

c) Develop interaction diagrams (one for each use case) using sequence or communication diagrams, depicting the sequence of interactions among the objects participating in each use case. Identify the object structuring criteria for the objects participating in each interaction diagram. (6 pts)

Use case 1:

Diagram

Description automatically generated

Use case 2:

Diagram

Description automatically generated

Use case 3:

Diagram

Description automatically generated

Use case 4,5,6,7:

Diagram

Description automatically generated with low confidence

d) For use cases involving entering and leaving the highway, show the statechart. Make sure that each statechart is consistent with the appropriate interaction diagram. (5 pts)

Use case: Enter Highway

Diagram

Description automatically generated

Use case: Exit highway

Diagram, schematic

Description automatically generated

**Term Project: Phase II**

For Phase II of the project, you are required to develop a Design Model for the Highway Toll System. In particular:

a) Develop an integrated communication diagram(s) showing all the objects and message interfaces in the system. (3 pts)

Diagram, schematic

Description automatically generated

b) Define the software architecture (depicted on a concurrent communication diagram) that shows the clients and server of the system. Define the message communication interfaces between the clients and server. (5 pts)

Diagram

Description automatically generated

c) Define the task architecture (depicted on concurrent communication diagrams) showing the concurrent tasks in each subsystem and the interfaces between the tasks. Describe the criteria used for task structuring. Define the message communication interfaces. (5 pts)

Diagram, schematic

Description automatically generated

**the criteria used for task structuring:**

**1. Account Registration interface**

<<Event driven>>

<<User interface>>

: account registration interface

The account registration interface activates upon the external input received from the driver, hence this interface will be event driven task

**2. Account Manager**

<<Demand driven>>

<<Business Logic>>

: Account Manger

The Account Manager Business logic will be activated from the message received from the input

Received from the account registration interface and performs required process, hence this

Interface will be demand driven task.

**3. Button interface**

<< Event driven>>

<<Input device interface>>

: Button interface

The button interface will be directly activated from the external input when the driver presses the button to get the ticket.

**4. Transponder detection sensor interface**

<<Event driven>>

<<Input device interface>>

: Transponder detection sensor interface

The transponder detection sensor interface will be activated from the external input received when vehicle approaches and it scans for the transponder. Hence it will be event driven task.

**5. Vehicle detection sensor interface**

<<Event driven>>

<<Input device interface>>

: Vehicle detection sensor interface

The vehicle detection sensor interface will be activated when a vehicle comes under the range of the sensor. Hence it is getting external input and getting active it will be event driven

**6. Ticket Reader interface**

<<Event Driven>>

<<Input device interface>>

: Ticket reader interface

The driver inserts the ticket in exiting the highway to calculate the task. This interface is activating with the input of the ticket. Therefore, it will be event driven.

**7. Credit card scanner interface**

<<Event driven>>

<<Input device interface>>

: Credit card scanner interface

Driver scans his credit card in the scanner and card scanner interface activates with the card input. Therefore, it will be event driven

**8. change calculator algorithm**

<<Demand driven>>

<<Algorithm>>

: change calculator

The change calculator algorithm will be activated when the driver enters the cash, and a control object sends an internal message to the change calculator algorithm. Therefore, it will be demand driven

**9. Driver interaction**

<<Event driven>>

<<User interaction>>

: Driver interaction

System displays toll amount to the user, then payment options will be displayed to select.

The driver selects the payment option which will be later sent to control object. Therefore, it will be event driven.

**10. Receipt printer interface**

<<Demand driven>>

<<Output device interface>>

: receipt printer interface

After payment the system will dispense a receipt. This will be activated from the internal message received from the control object. Hence it will be demand driven task.

**11. Bank operator interface**

<<Demand driven>>

<<Output device interface>>

: Bank operator interface

The bank operator interface will be activated after credit card is selected for payment and credit card is scanned. Then the operator object sends an internal message to bank operator interface for validation of charges. Therefore, it will be demand driven task.

**12. Toll operator interaction**

<<Event Driven>>

<<User interaction>>

: Toll operator interaction

The toll operator interaction will get toll displayed from the control object and payment type selected by the user. And operator need to enter details in case of cash payment selected. Therefore, it can be event driven.

**13. Traffic barrier interface**

<<Demand driven>>

<<Output device interface>>

: Traffic barrier interface

The traffic barrier interface directly works and gets activated from the input received from the control object. Therefore, it will be demand driven.

**14. Traffic light interface**

<<Demand driven>>

<<Output device interface>>

: Traffic light interface

The traffic light interface gets activates from the internal message received from the control. Hence it will be demand driven.

**15. Toll calculation algorithm**

<<Demand driven>>

<<Algorithm>>

: toll calculator

After ticket insertion or transponder detected at exit highway toll booth. A control object validates them and sends for toll calculation. This is an internal message, therefore it will be demand driven.

**16. Ticket printer interface**

<<Demand driven>>

<<output device interface>>

: ticket printer interface

Driver pushes a button at enter highway tollbooth when he doesn’t have a transponder. Then the system dispenses a ticket to the driver. The ticket printer interface activates when a message is sent from the control object. Therefore, it will be a demand driven.

**17. Toll control**

<<Demand driven>>

<<state dependent control>>

: toll control

Toll control object gets all the internal messages from the external objects and it co-ordinates them internally. Therefore, it will be demand driven.

**18. toll center alert**

<<Demand driven>>

<< User interface>>

: toll center alert

Toll center alert interface gets input from the control object when the tickets in the ticket dispenser are completely empty or low. It gets activated from the internal message. Therefore, it will be demand driven.

d) Develop a task interface specification for each concurrent task in the system. (2 pts)

1. **Account registration task**

**Name**: Account registration interface task

**Structuring task**: Role criteria: input/output user interface; concurrency criteria: event-driven

**Assumptions**: it will be accessed by driver’s personal device which is not part of the system.

**Task interface:**

**Task inputs-**

**Event input:** user login to the site and inputs the data for registration

**External input:** registrationDetailsInput

Synchronous message communication without reply:

-fill details for account creation

-validate form data

**Task outputs-**

**External output:** fill the form and validate it

Asynchronous message communication:

- account details are filled

- user form is validated

**Errors detected**: Kindly fill all mandatory fields indicated with \*

1. **Account manager task**

**Name:** Account manager task

**Structuring task**: Role criteria: business logic; concurrency criteria: demand-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** login and card details indicate the input

**External input:** LoginDetailsInput, CardDetailsInput

Synchronous message communication without reply:

--create account

-login

-add funds

**Task outputs-**

**External output:** create the account, login to the system, add funds

Asynchronous message communication:

- user account has been created

-logged into system

- funds are added

**Errors detected**: Invalid card or expired or insufficient funds in the card/ failed to create the use account, user already exist.

1. **Button interface task**

**Name:** Button interface task

**Structuring task**: Role criteria: input device interface; concurrency criteria: event-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** button press will go as external interrupt to the task

**External input:** buttonPressInput

Synchronous message communication without reply:

-Press button

**Task outputs-**

**External output:** button is pressed to get the ticket

Asynchronous message communication:

- Button is pressed

**Errors detected**: Login failed, Invalid card or expired or insufficient funds in the card

1. **Transponder detection sensor interface task**

**Name:** Transponder detection sensor interface task

**Structuring task**: Role criteria: input device interface; concurrency criteria: event-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** detect the transponder as input interrupt when vehicle is approached

**External input:** TransponderDetectionInput

Synchronous message communication without reply:

-detect the transponder

**Task outputs-**

**External output:** detected the transponder

Asynchronous message communication:

- Transponder is detected

**Errors detected**: NA

1. **Vehicle detection sensor interface task**

**Name:** Vehicle detection sensor interface task

**Structuring task**: Role criteria: input device interface; concurrency criteria: event-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** vehicle detection as input interrupt to task when vehicle is approached

**External input:** VehicleDetectionInput

Synchronous message communication without reply:

-detect vehicle

**Task outputs-**

**External output:** detected the vehicle

Asynchronous message communication:

- Arrival of vehicle is detected

**Errors detected**:

1. **Ticket reader interface task**

**Name:** Ticket reader interface task

**Structuring task**: Role criteria: input device interface; concurrency criteria: event-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** read the ticket as input interrupt to the task

**External input:** ticketReaderInput

Synchronous message communication without reply:

-read ticket

**Task outputs-**

**External output:** read ticket details

Asynchronous message communication:

- ticket details have been read

**Errors detected**: invalid ticket

1. **Credit card scanner interface task**

**Name:** Credit card scanner interface task

**Structuring task**: Role criteria: input device interface; concurrency criteria: event-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** credit card reader(scanner) as external interrupt to indicate that a card has been input.

**External input:** CreditCardScannerInput

Synchronous message communication with reply:

-read the credit card

**Task outputs-**

**External output:** read the card

Synchronous message communication with reply:

- card details have been read

**Errors detected**: invalid card detected, kindly try another card.

1. **Toll center control task**

**Name:** Toll center control task

**Structuring task**: Role criteria: control; concurrency criteria: demand-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** internal messages as input from different tasks with commands to perform.

**External input:** TollCenterInput

Synchronous message communication with reply:

-read internal commands

-execute the actions as per commands

-update the response back

**Task outputs-**

**External output:** read commands from different tasks and execute the actions

Synchronous message communication with reply:

- Appropriate actions have been performed as per internal commands from different tasks

**Errors detected**: Failed validations on any of the task inputs.

1. **Change calculator algorithm task**

**Name:** Change calculator algorithm task

**Structuring task**: Role criteria: algorithm; concurrency criteria: demand-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** internal messages from operator as input to calculate change algorithm.

**External input:** InternalMessagesInput

Synchronous message communication with reply:

-read the entered cash from internal message

-calculate the change

**Task outputs-**

**External output:** calculate the change

Synchronous message communication with reply:

- Change amount has been calculated

**Errors detected**: NA

1. **Ticket dispenser interface task**

**Name:** Ticket dispenser interface task

**Structuring task**: Role criteria: Output device interface; concurrency criteria: demand-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** internal message command by driver as input to the task

**External input:** tcketDispenserInput

Synchronous message communication without reply:

-Dispense the ticket

**Task outputs-**

**External output:** Dispenser prints the ticket

Asynchronous message communication:

-Ticket dispensed.

**Errors detected**:

1. **Toll calculator algorithm task**

**Name:** Toll calculator algorithm task

**Structuring task**: Role criteria: Algorithm; concurrency criteria: demand-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** command to calculate the toll charges as input

**External input:** calculateTollChargesInput

Synchronous message communication with reply:

-calculate toll charges

**Task outputs-**

**External output:** calculated charges

Asynchronous message communication:

-Toll charges calculated

**Errors detected**:

1. **Toll center alert task**

**Name:** Toll center alert task

**Structuring task**: Role criteria: User interface; concurrency criteria: demand-driven

**Assumptions**: when ticket count is zero, alert will be sent to the regional toll center

**Task interface:**

**Task inputs-**

**Event input:** Alert message will be sent as input to the task

**External input:** alertMessageInput

Synchronous message communication without reply:

-Alert the regional toll center

**Task outputs-**

**External output**: Send alert to the regional toll center

Asynchronous message communication:

-Alert has been sent to the regional toll center.

**Errors detected**: NA

1. **Traffic barrier interface task**

**Name:** Traffic barrier interface task

**Structuring task**: Role criteria: Output device interface; concurrency criteria: demand-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** Operator command as input to the traffic barrier

**External input:** TrafficBarrierInput

Synchronous message communication without reply:

-raise the barrier/ lower the barrier

**Task outputs-**

**External output:** raise/lower the barrier

Asynchronous message communication:

-Barrier is lowered/raised upon the signal.

**Errors detected**:

1. **Traffic light interface task**

**Name:** Traffic light interface task

**Structuring task**: Role criteria: Output device interface; concurrency criteria: demand-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** Traffic light signal will be given as input to the task

**External input:** TrafficLightInput

Synchronous message communication without reply:

-Turn on red/green traffic signal

**Task outputs-**

**External output:** Turn on red/green traffic signal

Asynchronous message communication:

- Turned on to red/green signal

**Errors detected**

1. **Toll operator interaction task**

**Name:** Toll operator interaction task

**Structuring task**: Role criteria: external input/output device; concurrency criteria: event-driven

**Assumptions**: Toll operator respond to driver

**Task interface:**

**Task inputs-**

**Event input:** Toll operator inputs command to the task in case of cash payment

**External input:** TollOperaterInput

Synchronous message communication with reply:

-Display dues and remaining change amount to driver

**Task outputs-**

**External output:** Toll operator display the dues, change, and collects cash from driver

Asynchronous message communication:

- Dues and change amount have been displayed

**Errors detected**: NA

1. **Bank operator interaction task**

**Name:** Bank operator interaction task

**Structuring task**: Role criteria: external system; concurrency criteria: demand-driven

**Assumptions**: Only bank operator can handle the transaction

**Task interface:**

**Task inputs-**

**Event input:** Payment authorization approval request will be sent to the Bank as input

**External input:** BankOperatorInput

Synchronous message communication with reply:

-Request payment authorization

**Task outputs-**

**External output:** Approve/Cancel the authorization request

Asynchronous message communication:

- Approved/Cancelled transaction

**Errors detected**: Insufficient fund, invalid card or expired.

1. **Receipt interface task**

**Name:** Receipt interface task

**Structuring task**: Role criteria: Output device interface; concurrency criteria: demand-driven

**Assumptions**: Payment success and operator provides receipt to the driver

**Task interface:**

**Task inputs-**

**Event input:** command will be sent for payment receipt

**External input:** PaymentReceiptInput

Synchronous message communication without reply:

-Print receipt

**Task outputs-**

**External output:** paymentreceipt is printed

Asynchronous message communication:

- Receipt printed

**Errors detected**:

1. **Driver Interaction task**

**Name:** Driver Interaction task

**Structuring task**: Role criteria: external input/output device; concurrency criteria: event-driven

**Assumptions**:

**Task interface:**

**Task inputs-**

**Event input:** Payment type will be selected by the driver as external interrupt

**External input:** PaymentTypeInput

Synchronous message communication with reply:

-Select payment type

**Task outputs-**

**External output:** driver selected the payment type

Synchronous message communication with reply:

- Selected the payment type

**Errors detected**:

**Assumptions:**

P = Primary key

F = Foreign key

**Use case: Purchase transponder**

(I) The driver will create a driver account; this data will be saved in driver account entity (driver id [P, F] (generated), Driver name, login id, password, email, address, phone number, driver’s license number, and vehicle license plate) and transponder account will be created separately in transponder entity (driver id [P, F], driver name, account id, balance). They both will connect using driver name as key.

Assuming a driver has a single driver account, and it is linked to only one transponder account.

**Use case: Add funds**

(I) When the driver adds funds to transponder account a transaction data will be created and stored in transaction entity (driver id[F], transaction id[P], credit card number, expiration data, amount) and funds are added to transponder account.

**Use case: Enter Highway**

(I) In step 3 alternative: assuming low tickets in toll both as completely empty and when the tickets are low it will alert the toll center and goes back to idle state.

(II) The data of Vehicle with transponder will be registered in Transponder Vehicle Travel Data Entity (Travel id (P), Transponder id (F), entry date, entry time, entry location)

(III) The data of Vehicle without transponder will be registered in non-Transponder Vehicle Travel Data Entity (Travel id (P), entry date, entry time, entry location)

**Use case: Exit Highway**

**Use case: Pay with transponder**

(I) For alternative case insufficient funds assuming the transponder have partial amount in their

account and the system display the due amount to the driver.so, he can choose another form

of payment to pay the balance amount.

**Use case: Pay with cash**

(I) Assuming driver selected the payment method as cash and handovers cash to toll operator.

The toll operator then enters the amount he received. Then the system displays the change to

be Returned.

(II) For alternative case insufficient funds operator records the license number offline and prints

receipt for acknowledgment, these details are saved in Toll billing data entity.

(III) In pay with cash payment method operator handles the traffic light and barrier manually.