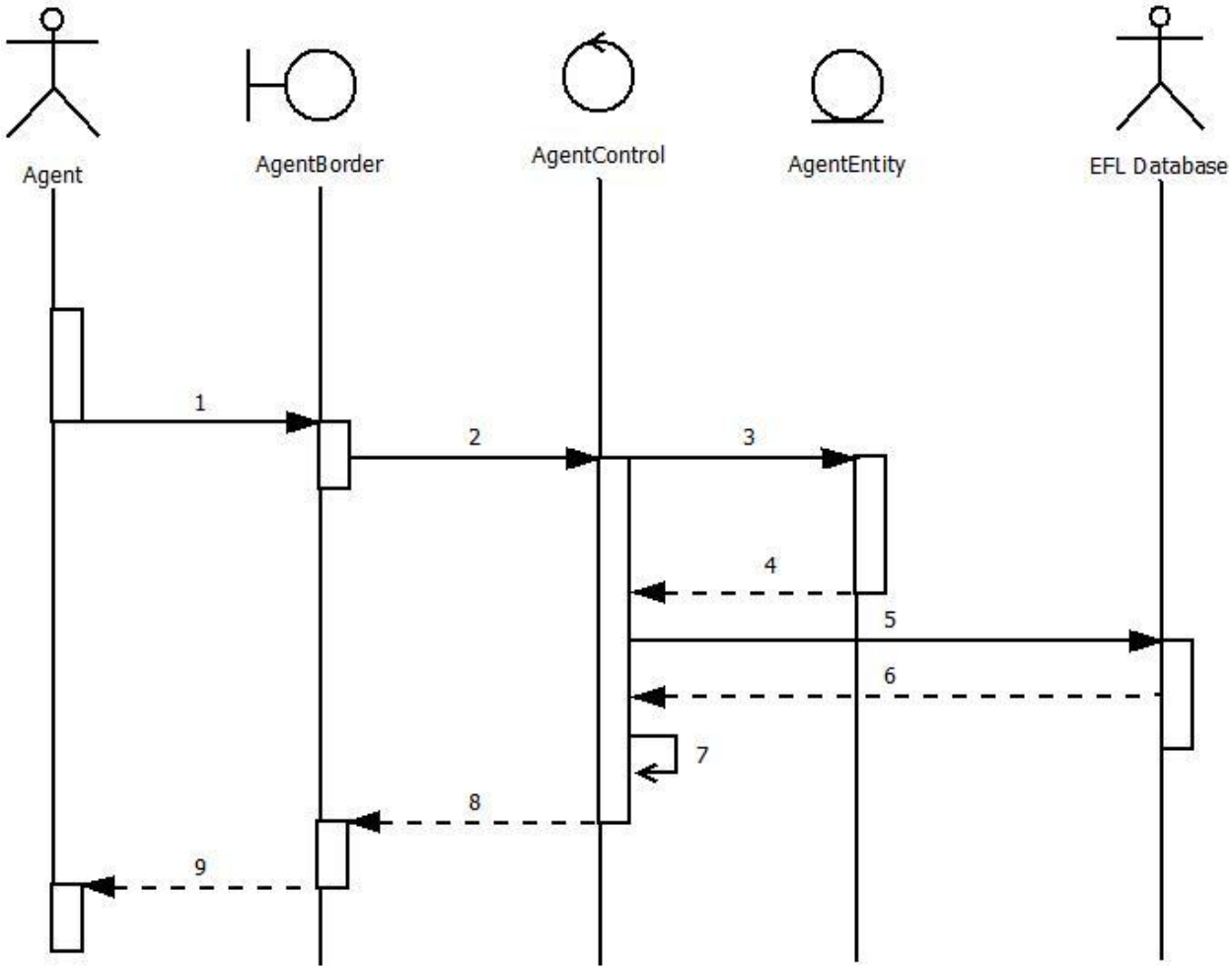
Sequence Diagrams

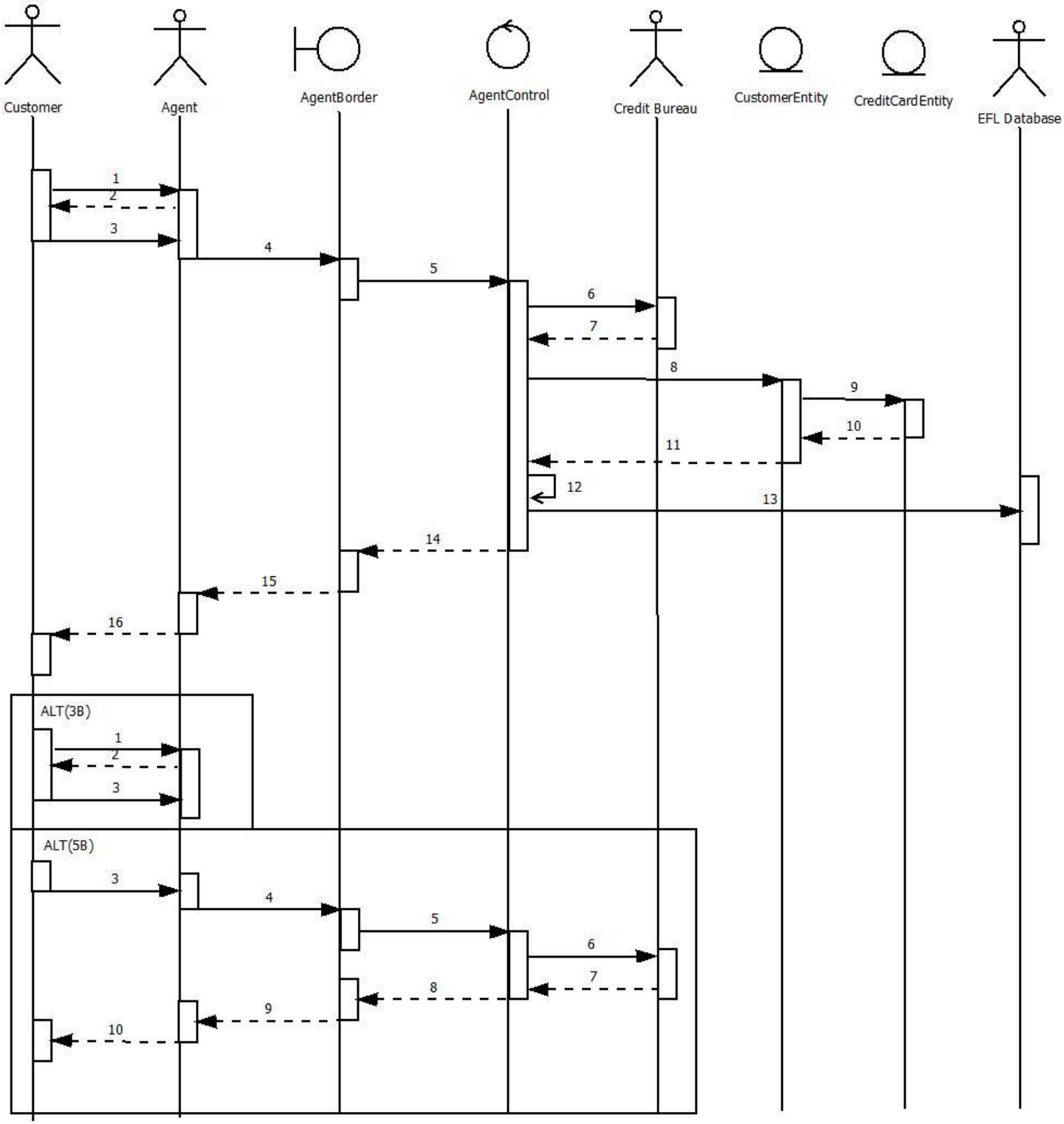
Agent Log-In



Agent Log-In(Normal)

1. Agent enters name and password
2. AgentBorder.LogIn()
3. AgentControl.LogIn()
4. AgentControl.agents.get(agentID)
5. AgentControl requests information from EFLDatabase
6. EFL Databse returns Agent Information
7. AgentControl verifies passwords are equal
8. AgentControl replies a successful update to AgentBorder
9. AgentBorder.MainView().setVisible(true)

Create New Customer Profile



Create New Customer Profile (Normal)

1. New Customer calls or enters the travel agency and would like to create an account
2. Agent explains fee structure and requests information from customer
3. Customer accepts services and provides information and credit card
4. Agent enters information into CreateAccount subscreen
5. AgentBorder.CreateNewCustomer()
6. AgentControl validates Credit Card number
7. Credit Information is cleared (valid)
8. new CustomerEntity()
9. new CreditCardEntity()
10. CreditCardEntity returns instance
11. CustomerEntity returns instance
12. AgentControl.CreateCustomerAccount()
13. AgentControl updates EFLDatabase
14. AgentControl provides fields to AgentBorder
15. AgentBorder.UpdateFields()
16. Agent Reports a successfully created account

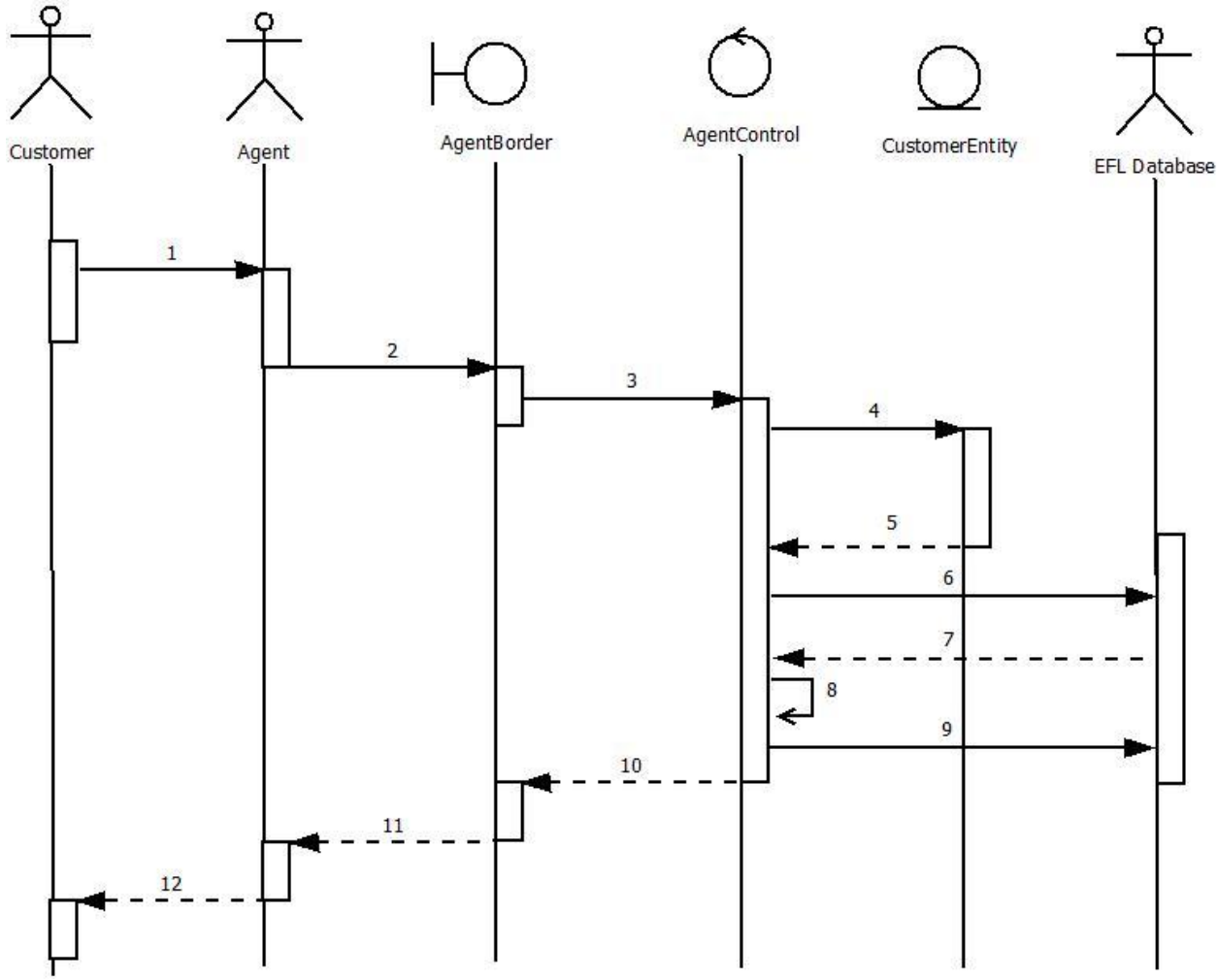
Sequence Diagram – Create New Customer Profile Scenario (Alt 3B)

1. New Customer calls or enters the travel agency and would like to create an account
2. Agent explains fee structure and requests information from customer
3. Customer rejects services

Sequence Diagram – Create New Customer Profile Scenario (Alt 5B)

3. Customer accepts services and provides information and credit card
4. Agent enters information into CreateAccount subscreen
5. AgentBorder.CreateNewCustomer()
6. AgentControl sends credit card info to be verified by Credit Bureau
7. Credit Information is invalid or unsatisfactory
8. AgentControl sends updated display to AgentBorder with error message
9. AgentBorder .DisplayErrorMessage()
10. Agent Reports an successfully created account

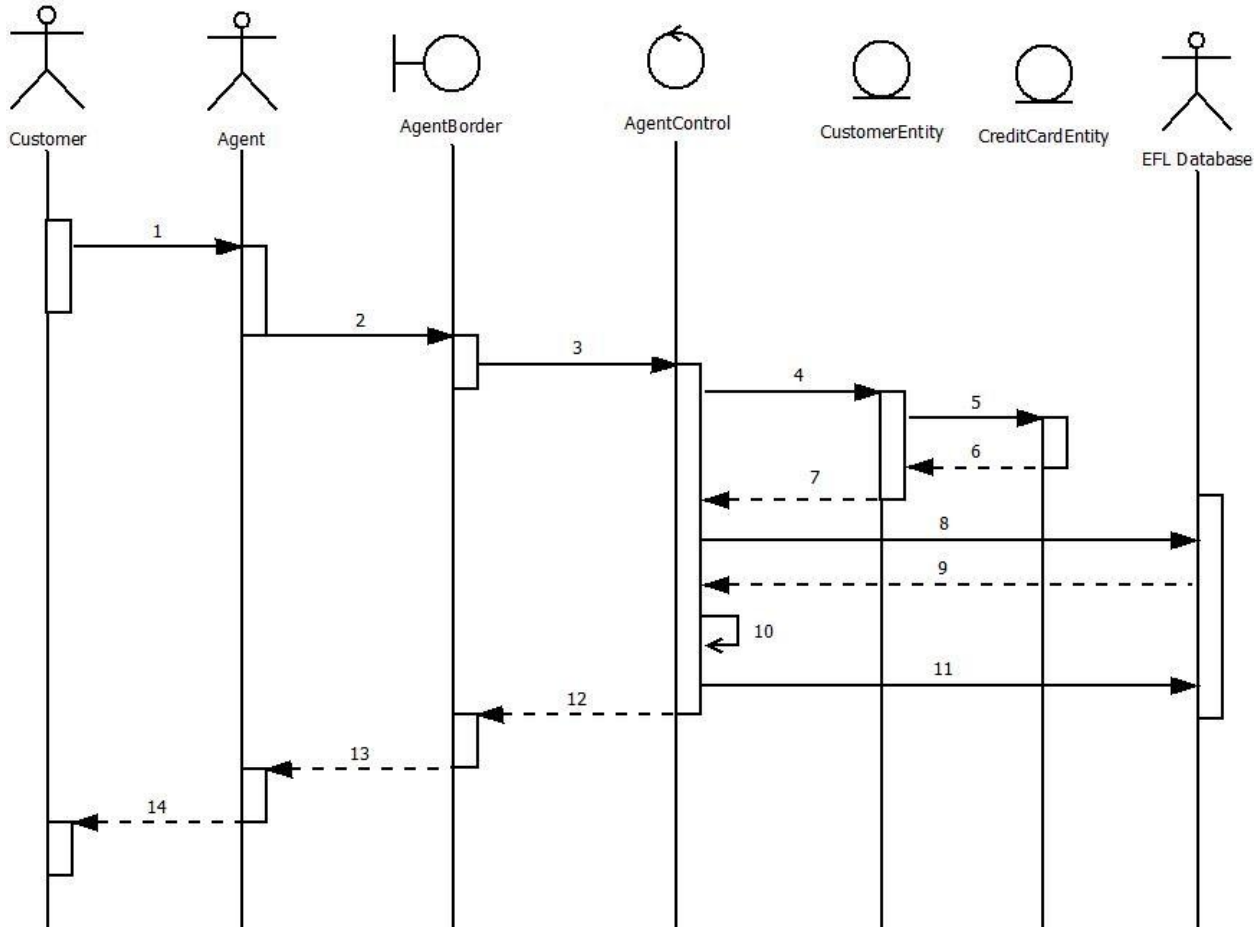
Add Credit to Customer Profile



Add Credit to Customer Profile

1. Customer brings in referral or Manager sees reason to provide credit to customer account
2. Agent requests to add credit to Customer Account giving the customers profile number
3. AgentBorder.AddCredit()
4. CustomerEntity()
5. CustomerEntity returns a new instance of itself
6. AgentControl requests for customer information from EFL Database
7. EFL Database returns customer information
8. AgentControl.AddCredit()
9. AgentControl sends information to the EFL Database
10. AgentControl returns successful operation
11. AgentBorder.DisplayMessage(“Add Credit Successful”)
12. Agent informs Customer the credit was added

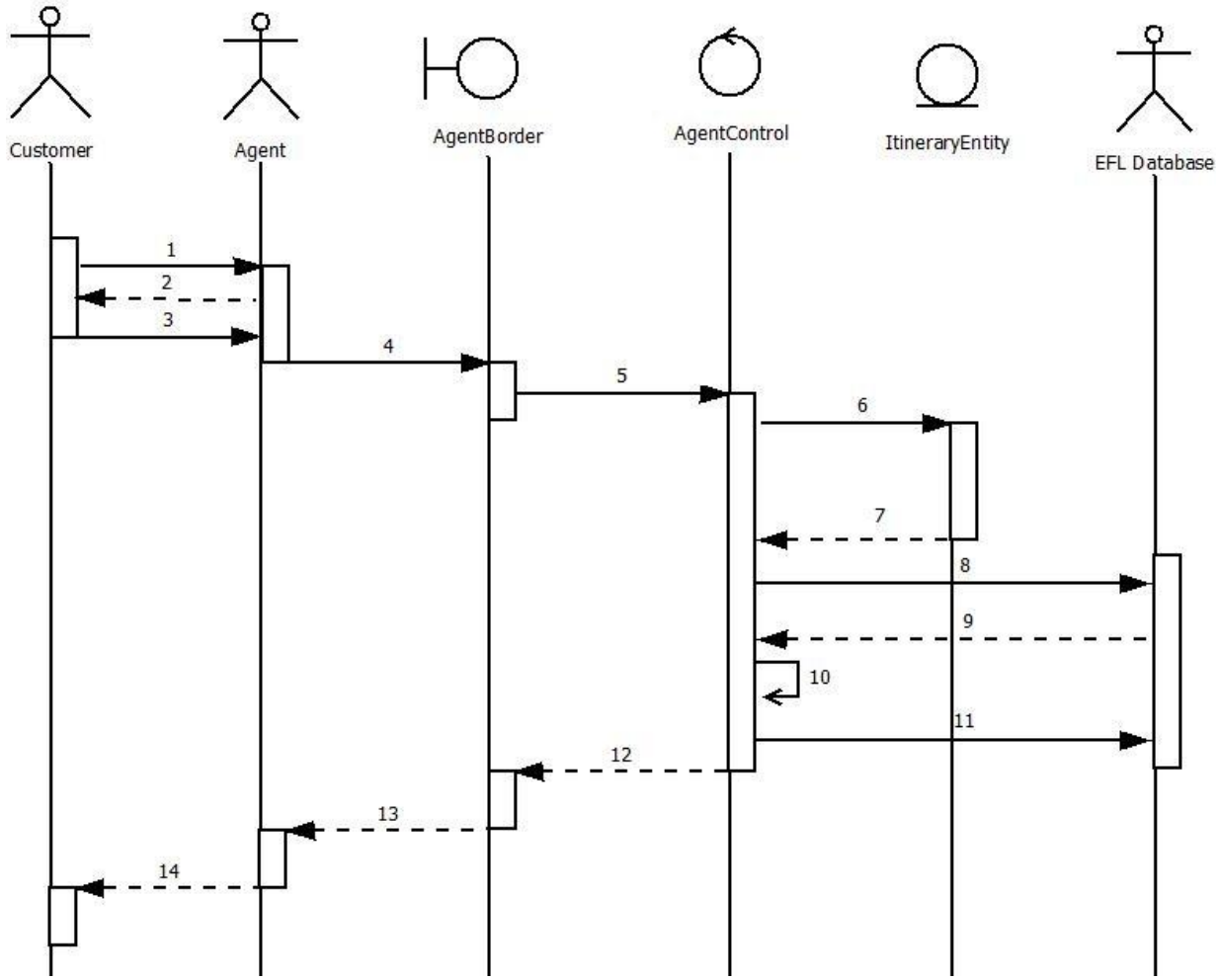
Manage Existing Customer Profile



Manage Existing Customer Profile(Normal)

1. Customer needs Agent to retrieve or update their profile with some new information
2. Agent inputs all necessary information into the terminal
3. AgentBorder.Manage Customer()
4. CustomerEntity()
5. CreditCardEntity ()
6. CreditCardEntity returns a new instance of itself
7. CustomerEntity returns a new instance of itself
8. AgentControl .CustomerLookUp(CustomerID)
   (Note: CustomerID is a name or phone number)
9. EFL Database returns information
10. AgentControl.UpdateCustomer(customerID)
11. AgentControl sends updated information to EFLDatabase
12. AgentControl returns updated information to the AgentBorder class
13. AgentBorder.UpdateFields()
14. Agent reports a successful update to Customer

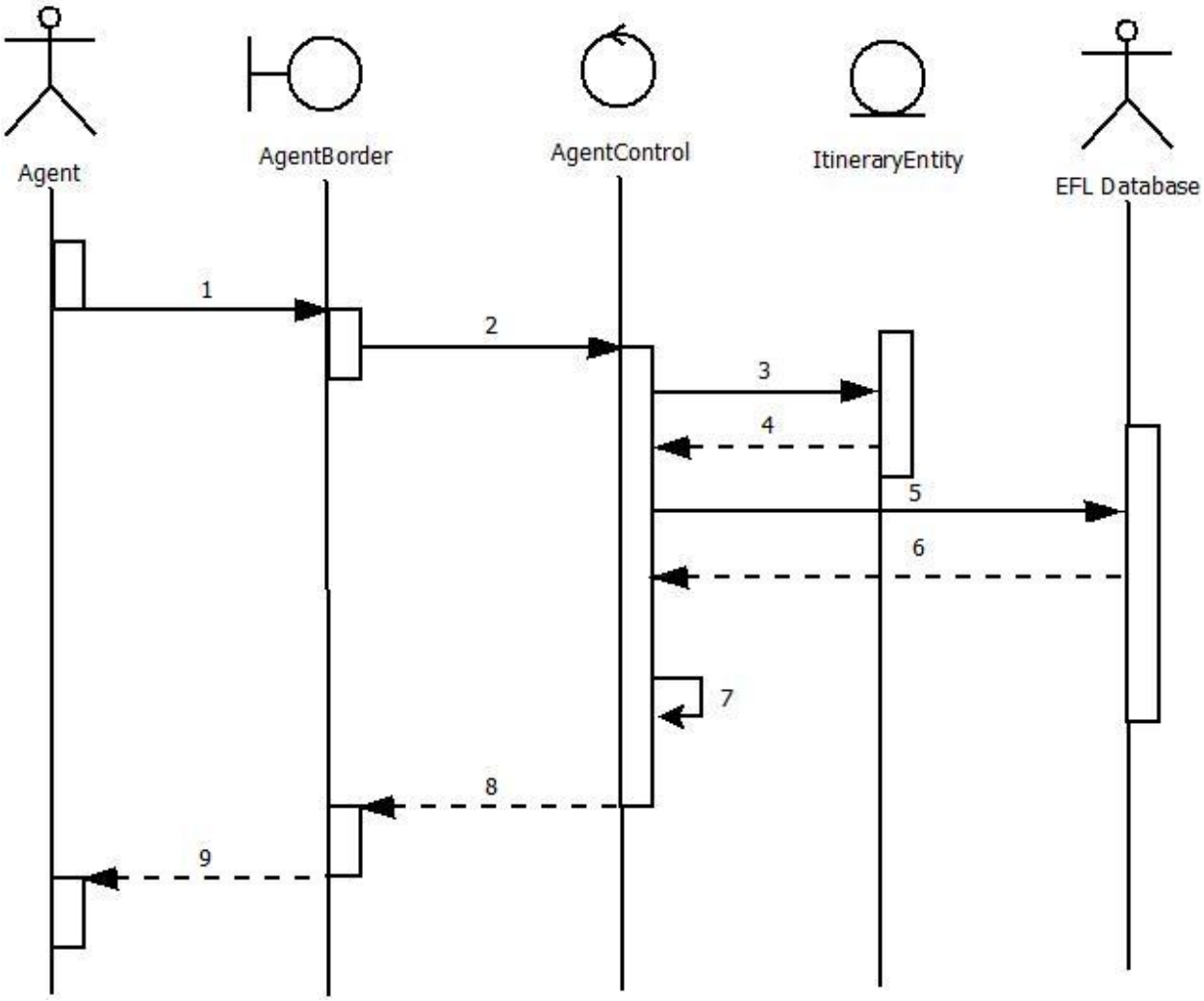
Create New Itinerary Case



Create New Itinerary Case

1. The Customer has requested to book a flight without a case already opened
2. The Agent requests information about the flight from the customer
3. The Customer gives the Agent the needed information
4. The Agent Inputs all the information in the AgentBorder Terminal
5. AgentBorder.CreateNewFlightItinerary()
6. ItineraryEntity()
7. ItineraryEntity returns a new instance of itself to AgentControl
8. AgentControl .SearchFlights()
9. EFLDatabase returns Information containing all the info for the Itinerary
10. AgentControl .CreateNewItinerary()
11. AgentControl sends Itinerary class to the EFLDatabase
12. AgentControl returns class info to AgentBorder
13. AgentBorder .UpdateFields
14. Agent Reports Back a successful Itinerary Creation

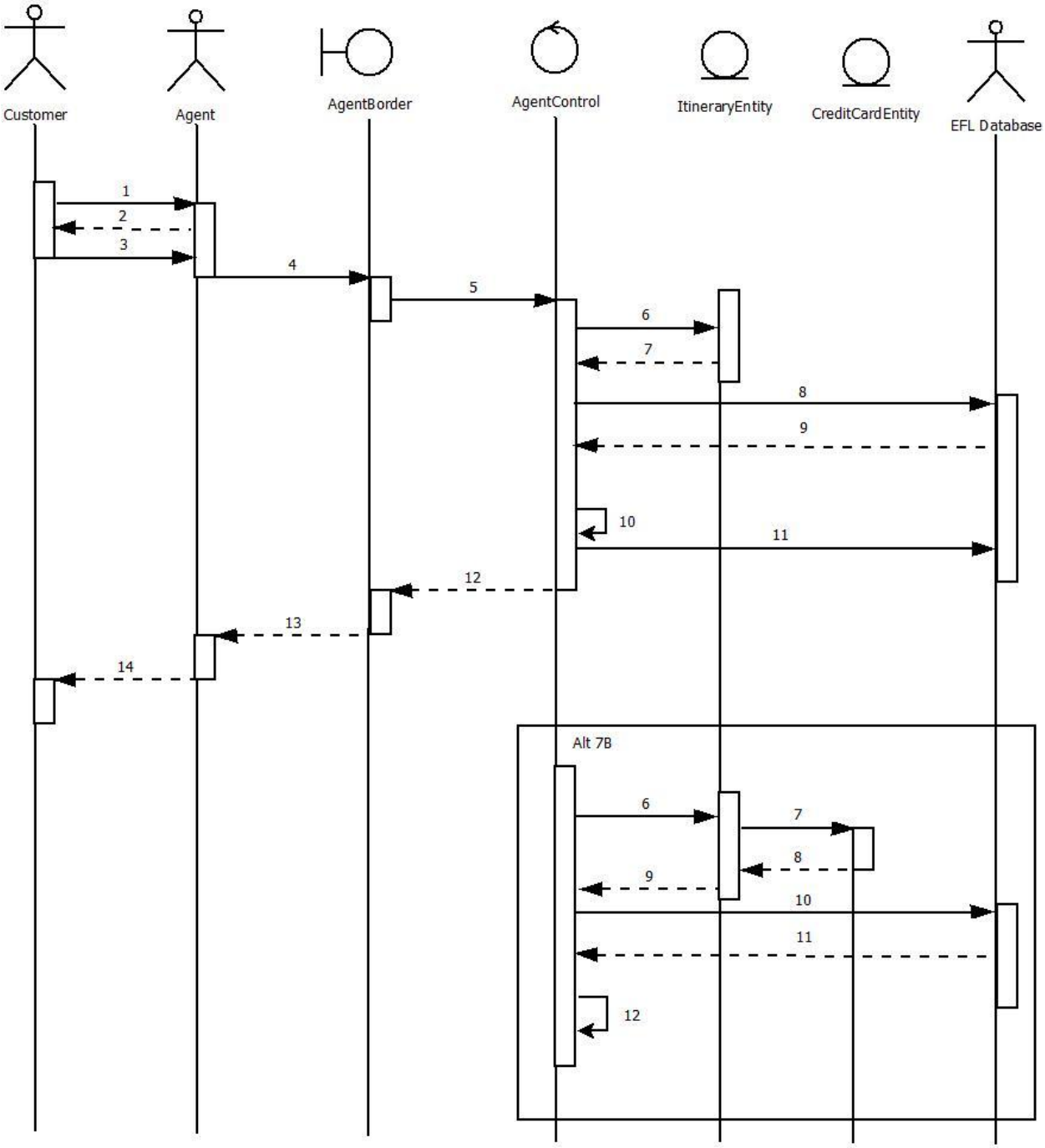
Produce Flight List



Produce Flight List(Normal)

1. Agent sends request to produce flight list by inserting customer and itineraryID
2. AgentBorder.ProduceFlightList()
3. ItineraryEntity()
4. ItineraryEntity returns a new instance of itself
5. AgentControl.eflDatabase.getCustomer(customerID).getItinerary(itineraryID)
6. EFL Database returns information to the AgentControl
7. AgentControl .SearchFlights(itineraries.getItinerary(itineraryID)
8. AgentControl. ProduceFlightList(flights)
9. AgentBorder.UpdateFields()

Reserve Flight



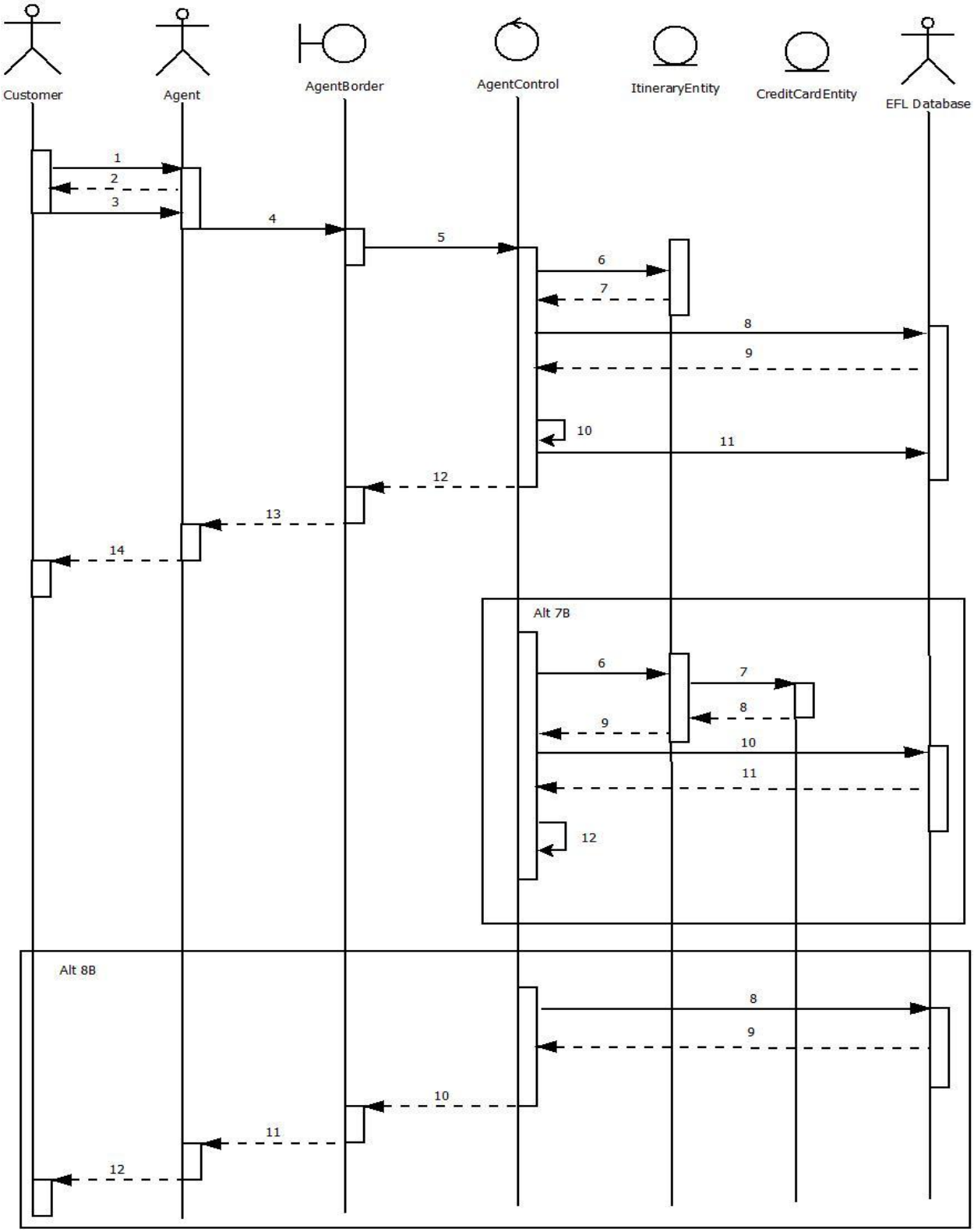
Reserve Flight(Normal)

1. Customer Wants to make a flight reservation
2. Agent requests specific information about the flight
3. Customer provides Agent with the information
4. The Agent Inputs all the information in the AgentBorder Terminal
5. AgentBorder.ReserveFlight()
6. ItineraryEntity()
7. ItineraryEntity returns a new instance of itself to AgentControl
8. AgentControl.eflDatabase.getCustomer(customerID).getItinerary(itineraryID)
9. EFLDatabase returns info containing all the info for the Itinerary
10. AgentControl.ResereFlight(flight)
11. AgentControl sends Itinerary info to the EFLDatabase
12. AgentControl returns info to AgentBorder
13. AgentBorder.UpdateFields()
14. Agent Reports Back a successful Itinerary Creation

Reserve Flight(7B)

6. ItineraryEntity()
7. CreditCardEntity()
8. CreditCardEntity returns a new instance of itself
9. ItineraryEntity returns a new instance of itself to AgentControl
10. AgentControl requests for the saved Itinerary for customer from EFLDatabase
11. EFLDatabase returns a text document containing all the info for the Itinerary
12. AgentControl updates all needed info into the ItineraryEntity

Manage Flight



Manage Flight(Normal)

1. Customer Wants to make a flight reservation
2. Agent requests specific information about the flight
3. Customer provides Agent with the information
4. The Agent Inputs all the information in the AgentBorder Terminal
5. AgentBorder.ModifyReservation()
6. ItineraryEntity()
7. ItineraryEntity returns a new instance of itself to AgentControl
8. AgentControl.eflDatabase.getCustomer(customerID).getItinerary(itineraryID)
9. EFLDatabase returns a text document containing all the info for the Itinerary
10. AgentControl.ModifyReservation(itinerary)
11. AgentControl sends Itinerary class to the EFLDatabase
12. AgentControl returns class instance to AgentBorder
13. AgentBorder .UpdateFields
14. Agent Reports Back a successful Itinerary Creation

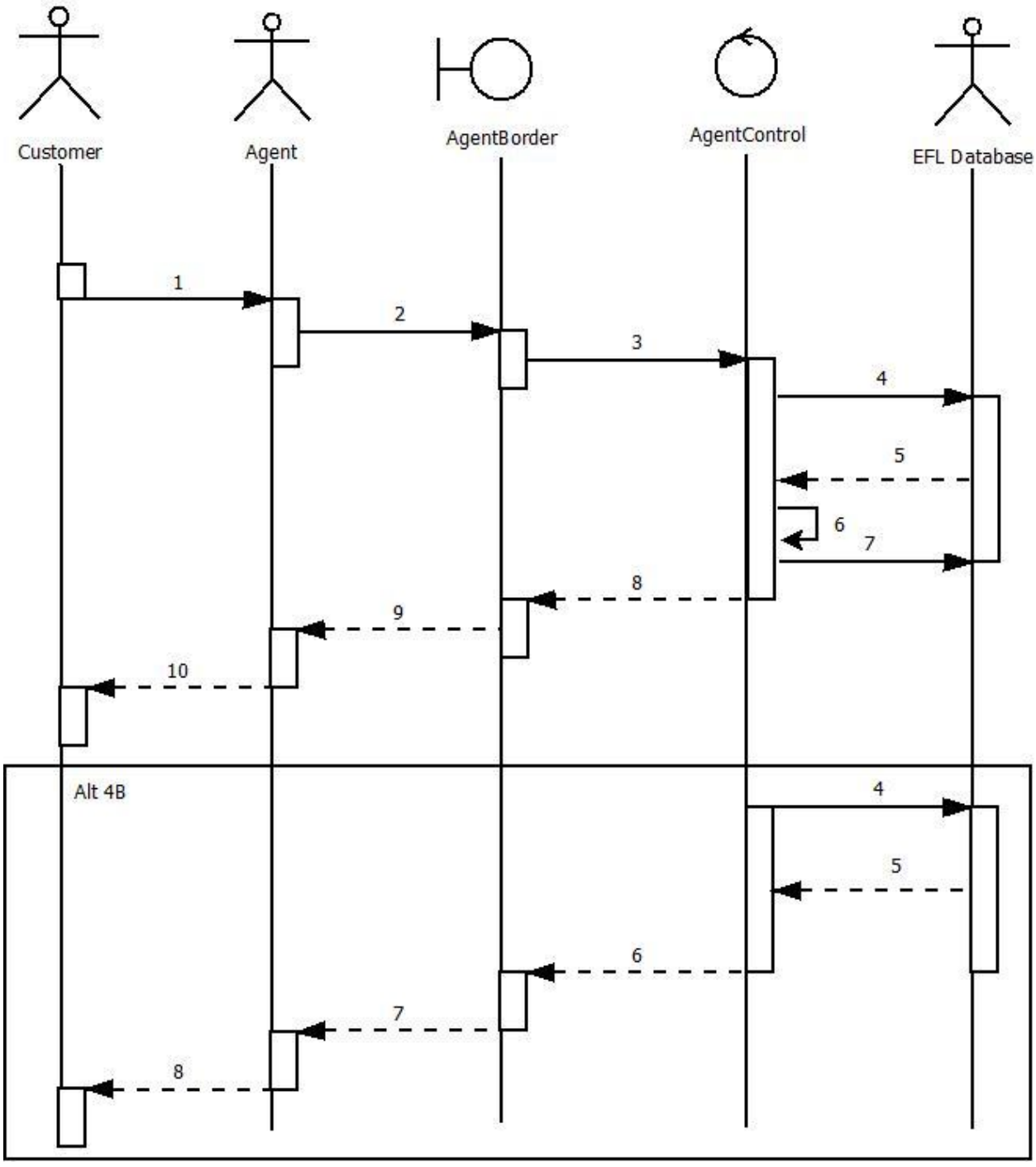
Manage Flight(7B)

6. ItineraryEntity()
7. CreditCardEntity()
8. CreditCardEntity returns a new instance of itself
9. ItineraryEntity returns a new instance of itself to AgentControl
10. AgentControl.eflDatabase.getCustomer(CustomerID).getItinerary(itineraryID)
11. EFLDatabase returns a text document containing all the info for the Itinerary
12. AgentControl updates all needed info into the ItineraryEntity

Manage Flight(8B)

8. AgentControl.itineraries.get(itineraryID)
9. EFL Database returns a null
10. AgentControl reports a failed operation to AgentBorder
11. AgentBorder.DisplayErrorMessage()
12. Agent tells customer that the Itinerary doesn't exist and offers to create a new one

Cancel Reservation



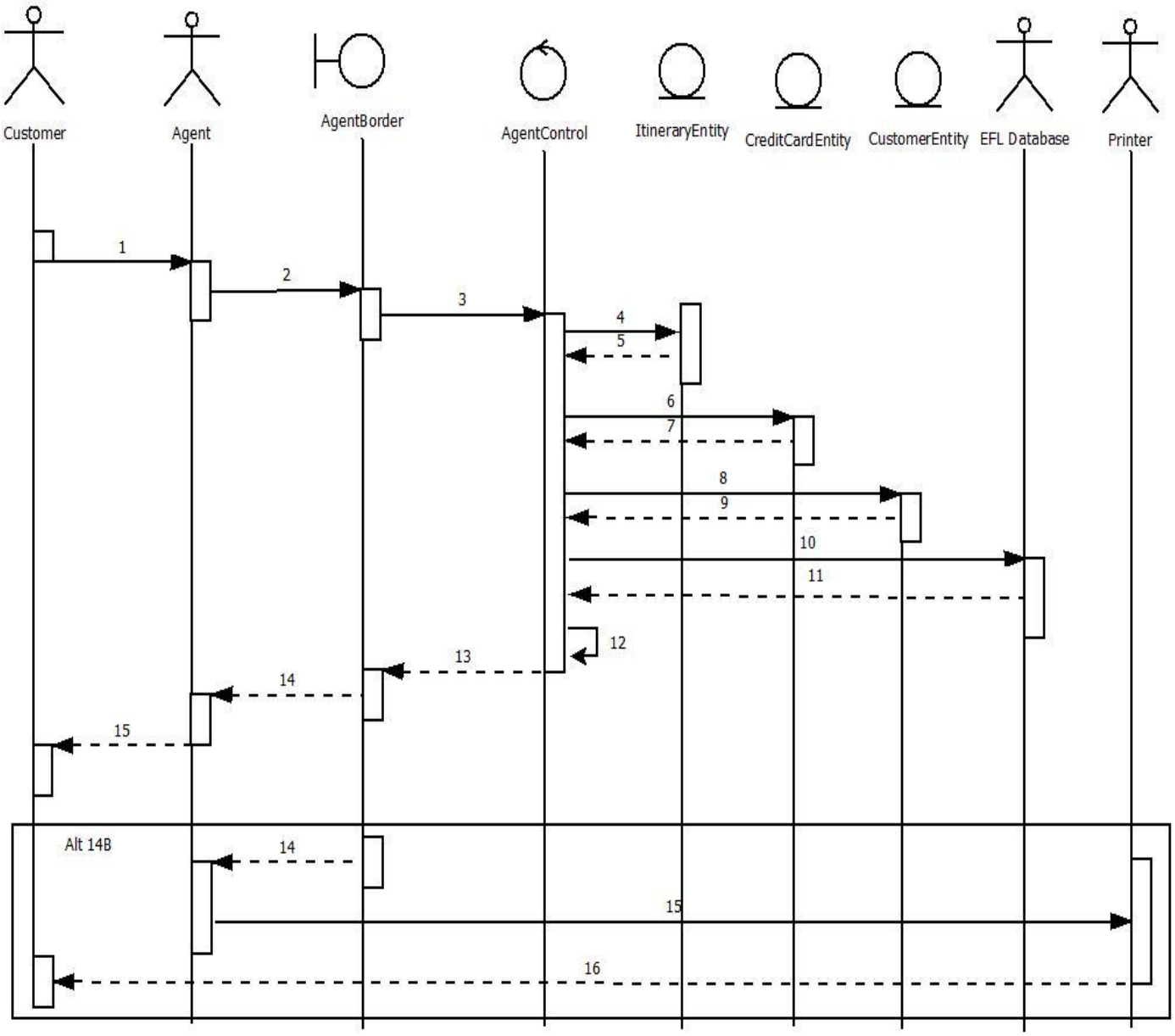
Cancel Reservation (Normal)

1. Customer Requests that their flight reservation is cancelled
2. Agent looks up the Itinerary to cancel by the customer and itinerary number in Agent Border
3. AgentBorder.CancelReservation()
4. AgentControl.eflDatabase.customer(customerID).itineraries.get(itineraryID)
5. EFL Database returns itinerary information
6. AgentControl.CancelReservation()
7. AgentControl sends updated information to the EFL Database
8. AgentControl returns successful cancellation to AgentBorder
9. AgentBorder.UpdateFields()
10. Agent reports to Customer a successful

Manage Flight(4B)

4. AgentControl.customer(customerID).itineraries.get(itineraryID)
5. EFL Database returns a null
6. AgentControl reports a failed operation to AgentBorder
7. AgentBorder.DisplayErrorMessage()
8. Agent tells customer that the Itinerary doesn't exist and offers to create a new one

Produce Flight Receipt



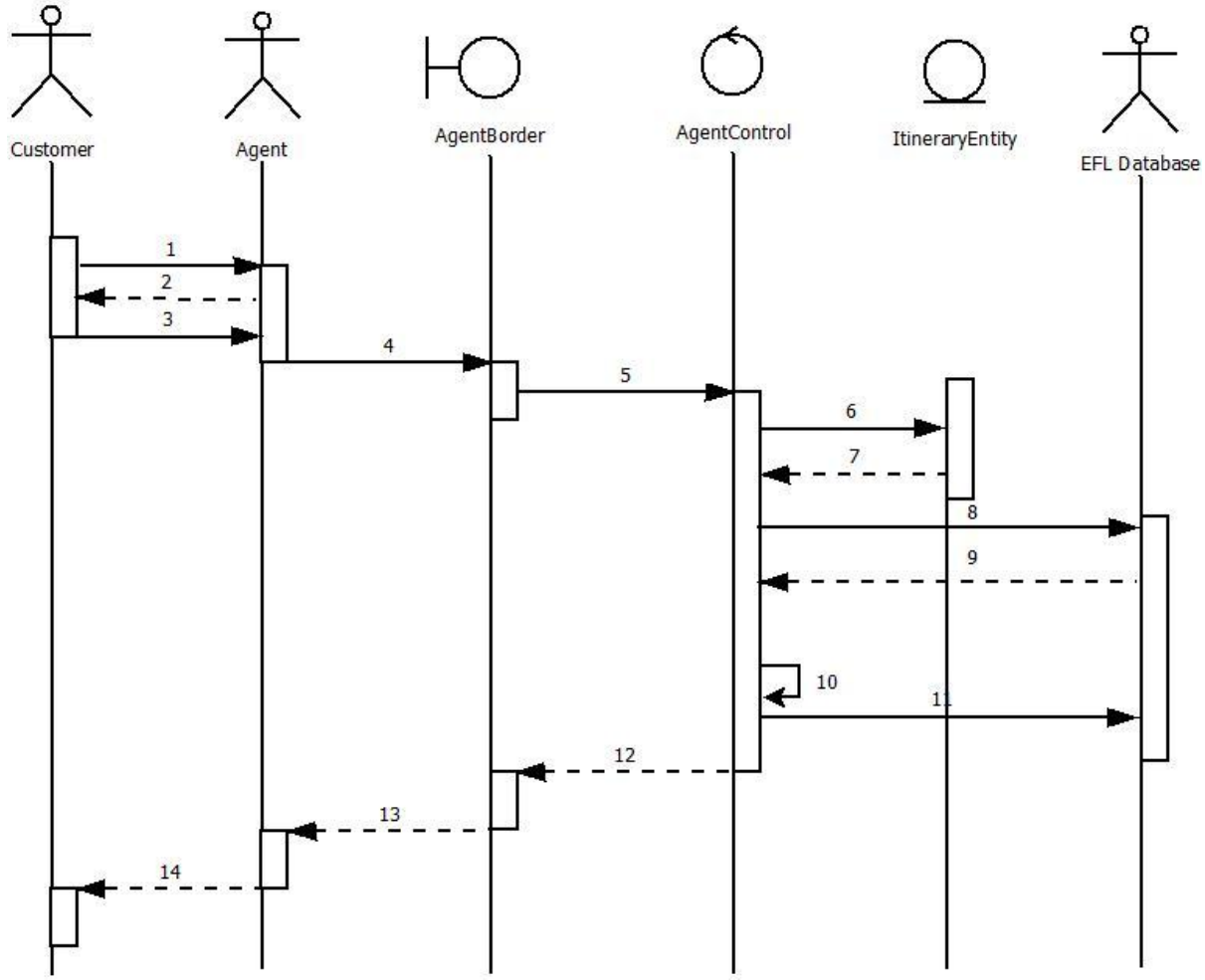
Produce Flight Receipt (Normal)

1. Customer Requests to print their flight reservation
2. Agent looks up the Itinerary to print by the customer and itinerary number in Agent Border
3. AgentBorder.ProduceFlightReceipt()
4. ItineraryEntity()
5. ItineraryEntity returns a new instance of itself
6. CreditCardEntity()
7. CreditCardEntity returns a new instance of itself
8. CustomerEntity()
9. CustomerEntity returns a new instance of itself
10. AgentControl.eflDatabase.getCustomer(customerID)
11. EFL Database returns itinerary information
12. AgentControl sets information to get returned as document flight reservation from itinerary
13. AgentControl .ProduceFlightReceipt(eflDatabase customer.itineraries.get(itineraryID))
14. AgentBorder.DisplayDocument()
15. Agent sends soft copy to customer via email

Produce Flight Receipt (14B)*

14. AgentBorder.DisplayDocument()
15. AgentBorder displays document and Agent chooses print
16. Printer prints the flight Receipt to the customer

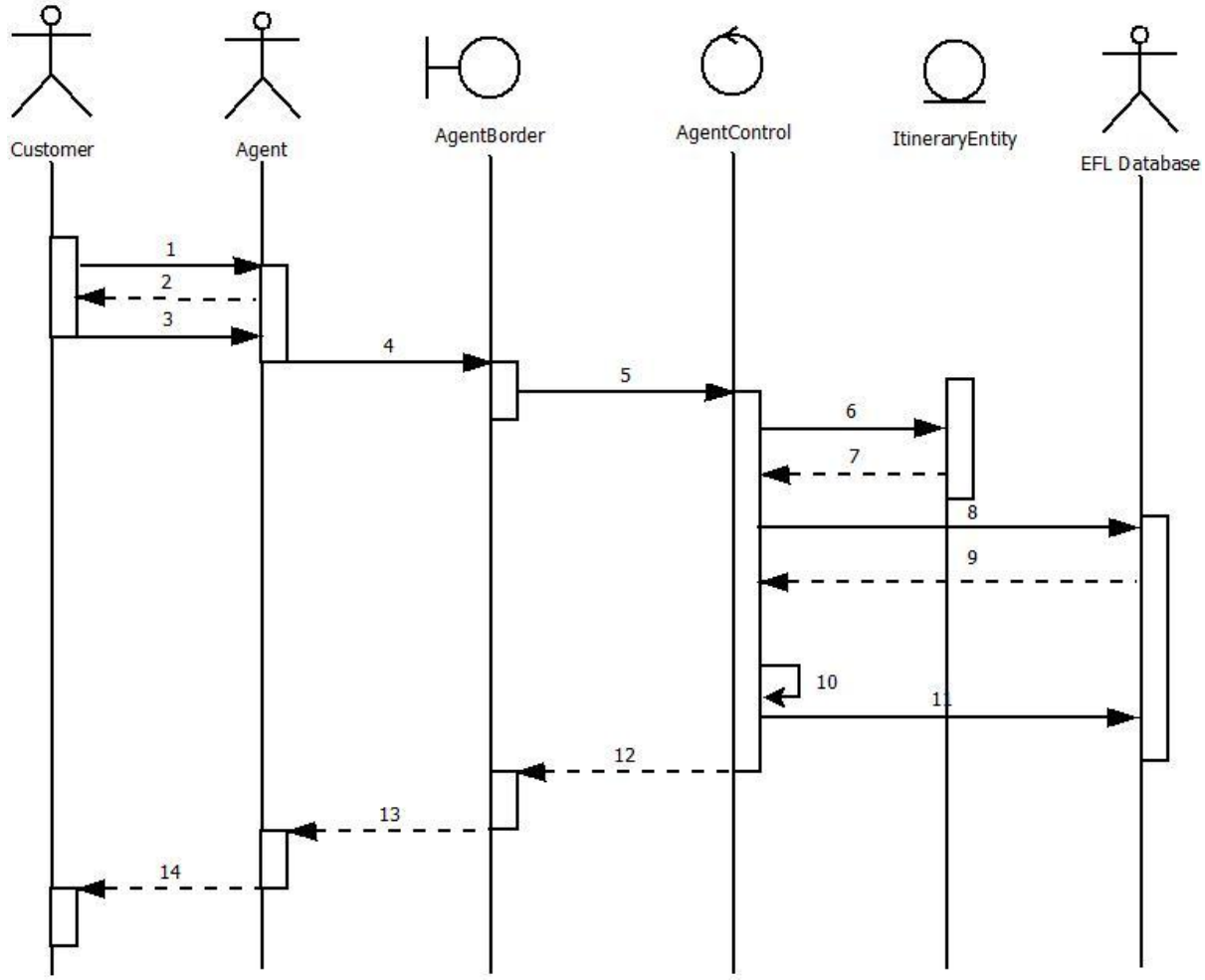
Create Price Watch



Create Price Watch(Normal)

1. The Customer doesn't like the prices for their flights
2. The Agent offers to create a price watch and requests their desired price
3. The Customer gives the Agent the needed information
4. The Agent Inputs all the information in AgentBorder
5. AgentBorder.CreatePriceWatch()
6. ItineraryEntity()
7. ItineraryEntity returns a new instance of itself to AgentControl
8. AgentControl.eflDatabase.customer(customerID).itineraries.get(itineraryID)
9. EFLDatabase returns a text document containing all the info for the Itinerary
10. AgentControl updates the price watch info into the ItineraryEntity instance
11. AgentControl sends Itinerary Case with updated price watch to the EFLDatabase
12. AgentControl returns info to AgentBorder
13. AgentBorder.UpdateFields()
14. Agent Reports Back a successful price watch creation

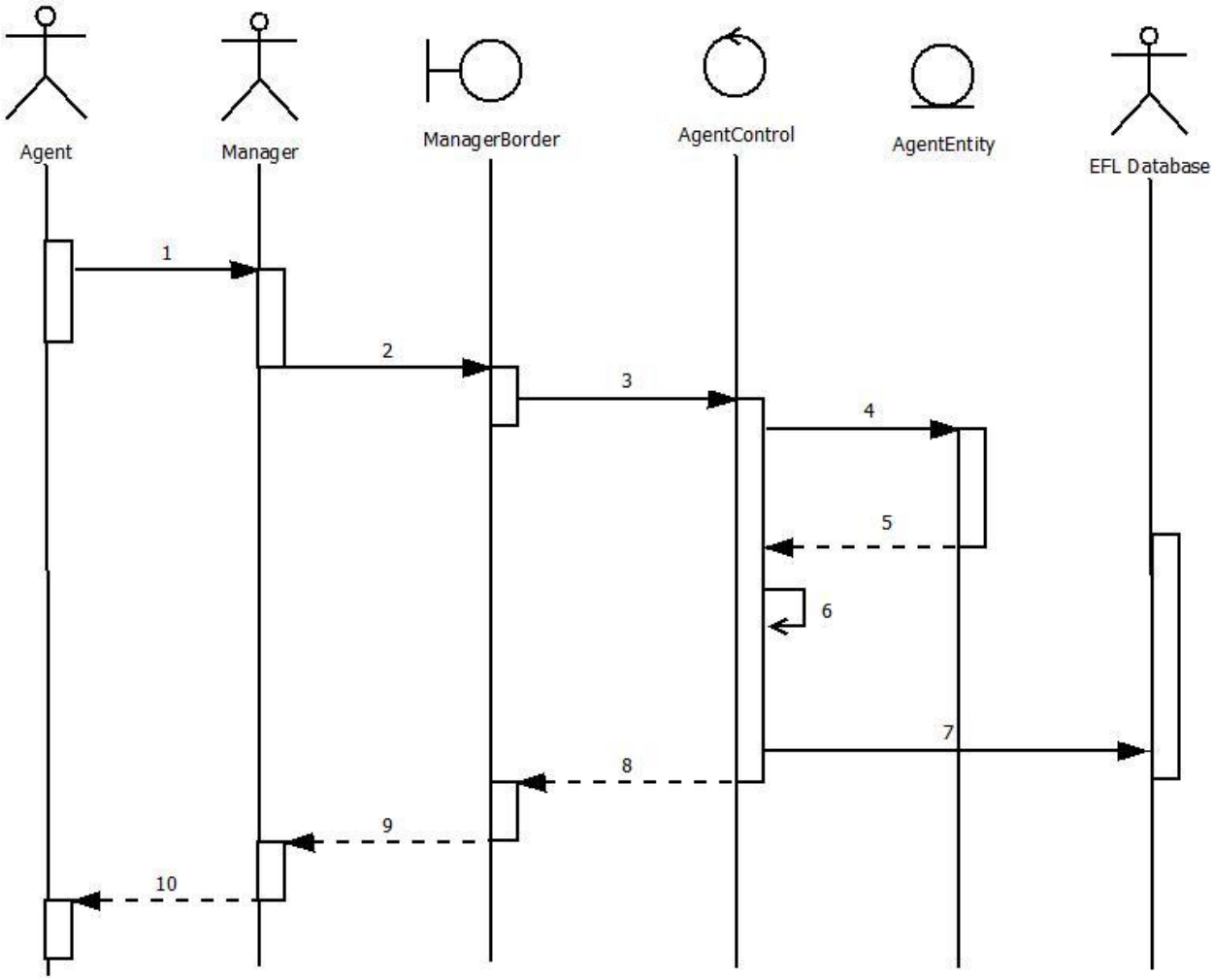
Cancel Price Watch



Cancel Price Watch(Normal)

1. The Customer Requests to book a price watch flight or wants to cancel price watch
2. The Agent requests the itinerary number
3. The Customer gives the Agent the needed information
4. The Agent Inputs all the information in AgentBorder
5. AgentBorder.CancelPriceWatch()
6. ItineraryEntity()
7. ItineraryEntity returns a new instance of itself to AgentControl
8. AgentControl.eflDatabase.customer(customerID).itineraries.get(itineraryID)
9. EFLDatabase returns a text document containing all the info for the Itinerary
10. AgentControl cancels the price watch in the ItineraryEntity instance
11. AgentControl sends Itinerary Case with updated information to the EFLDatabase
12. AgentControl returns class instance to AgentBorder
13. AgentBorder.DisplayMessage(“Price Watch Deleted”)
14. Agent Reports Back a successful price watch cancelation

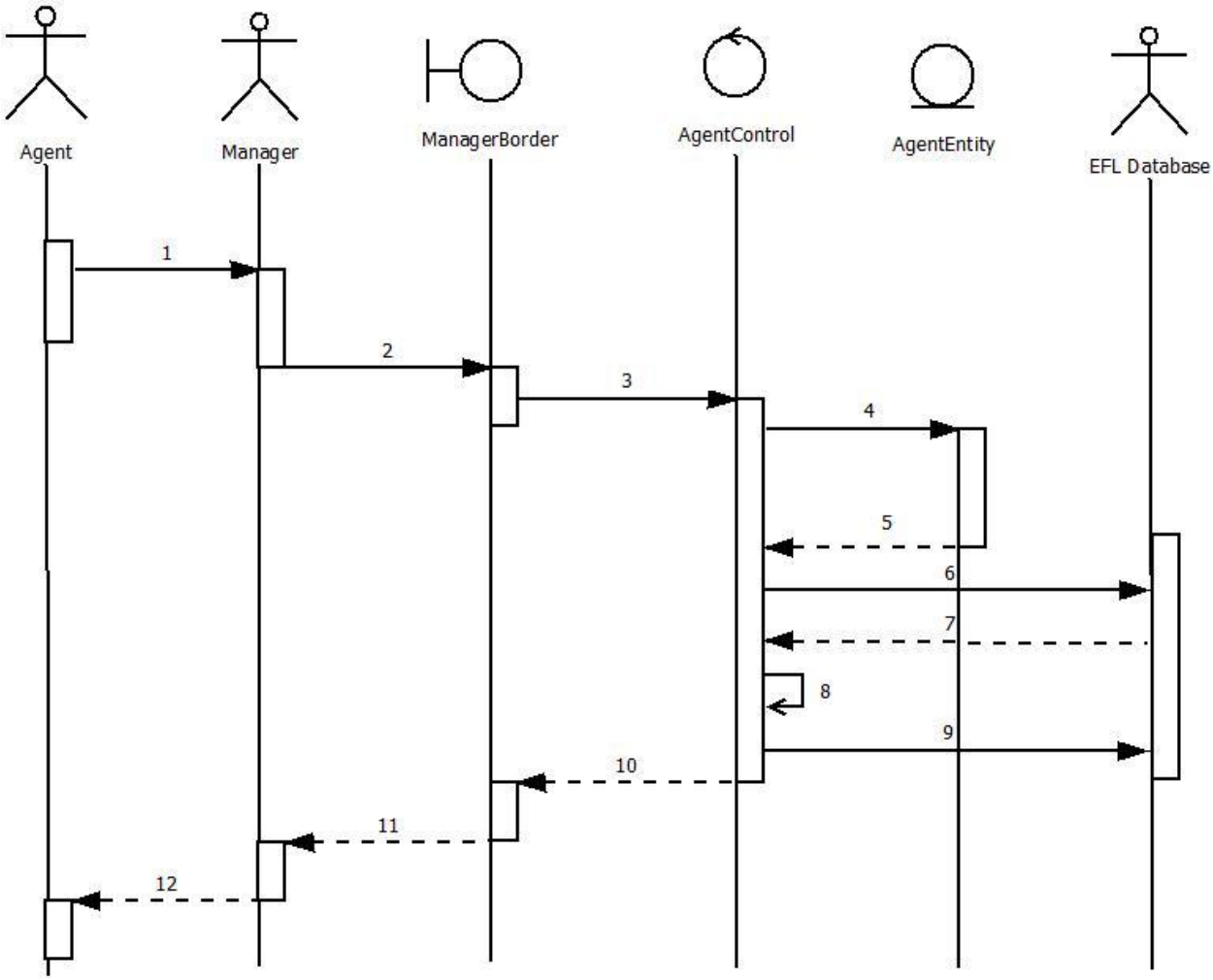
Create Agent Account



Create Agent Account(Normal)

1. Agent is a new hire and gives manager information to create an agent account
2. Manager enters information into ManagerBorder subscreen
3. ManagerControl.CreateNewAgent()
4. AgentEntity()
5. AgentEntity returns a new instance of itself to AgentControl
6. AgentControl.CreateNewAgent()
7. AgentControl sends agent info to EFLDatabase
8. AgentControl sends updated display to AgentBorder with customer instance
9. ManagerBorder.UpdatesFields()
10. Manager Reports a successfully created account

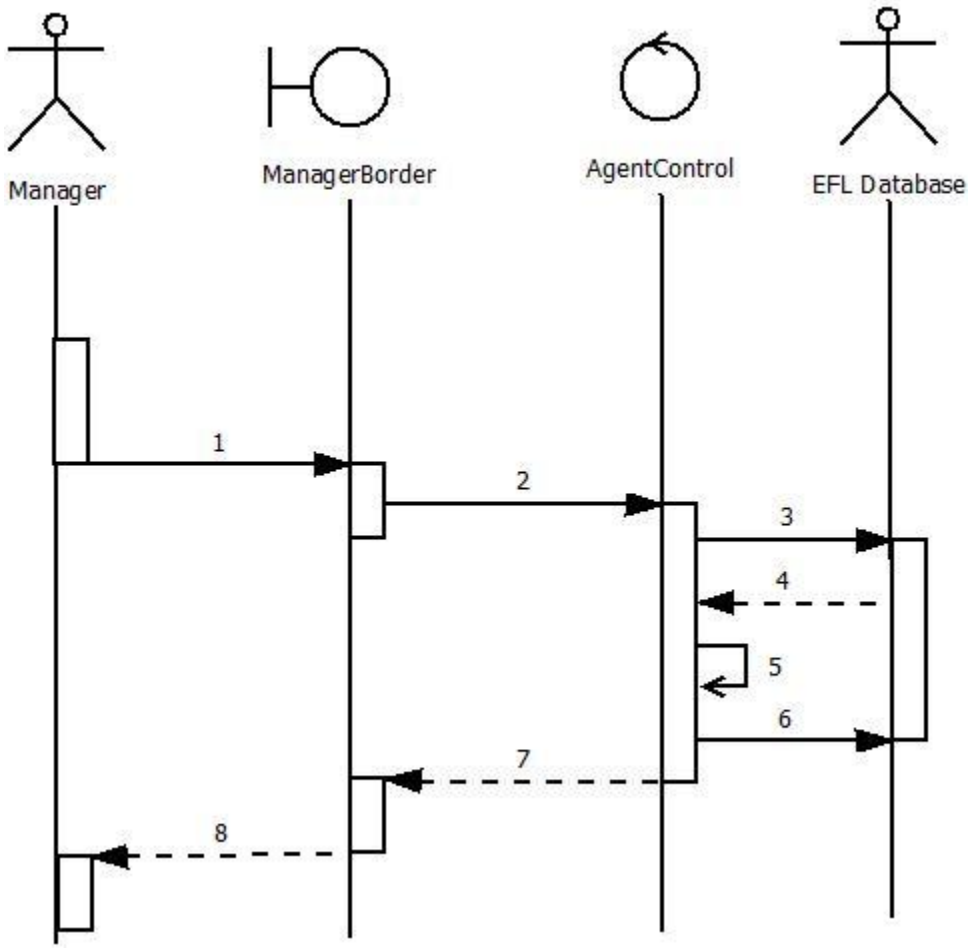
Manage Agent Account



Manage Agent Account(Normal)

1. Agent gives manager updated information to change their account
2. Manager enters information into ManagerBorder subscreen
3. ManagerBorder.ModifyAgentAccount()
4. AgentEntity()
5. AgentEntity returns a new instance of itself to AgentControl
6. AgentControl .eflDatabase.getAgentEntity(agentID)
7. EFl Database returns information to AgentControl
8. AgentControl.ModifyAgentAccount()
9. AgentControl sends agent info to EFLDatabase
10. AgentControl sends updated display to AgentBorder with customer instance
11. ManagerBorder .UpdateFields()
12. Manager Reports a successfully created account

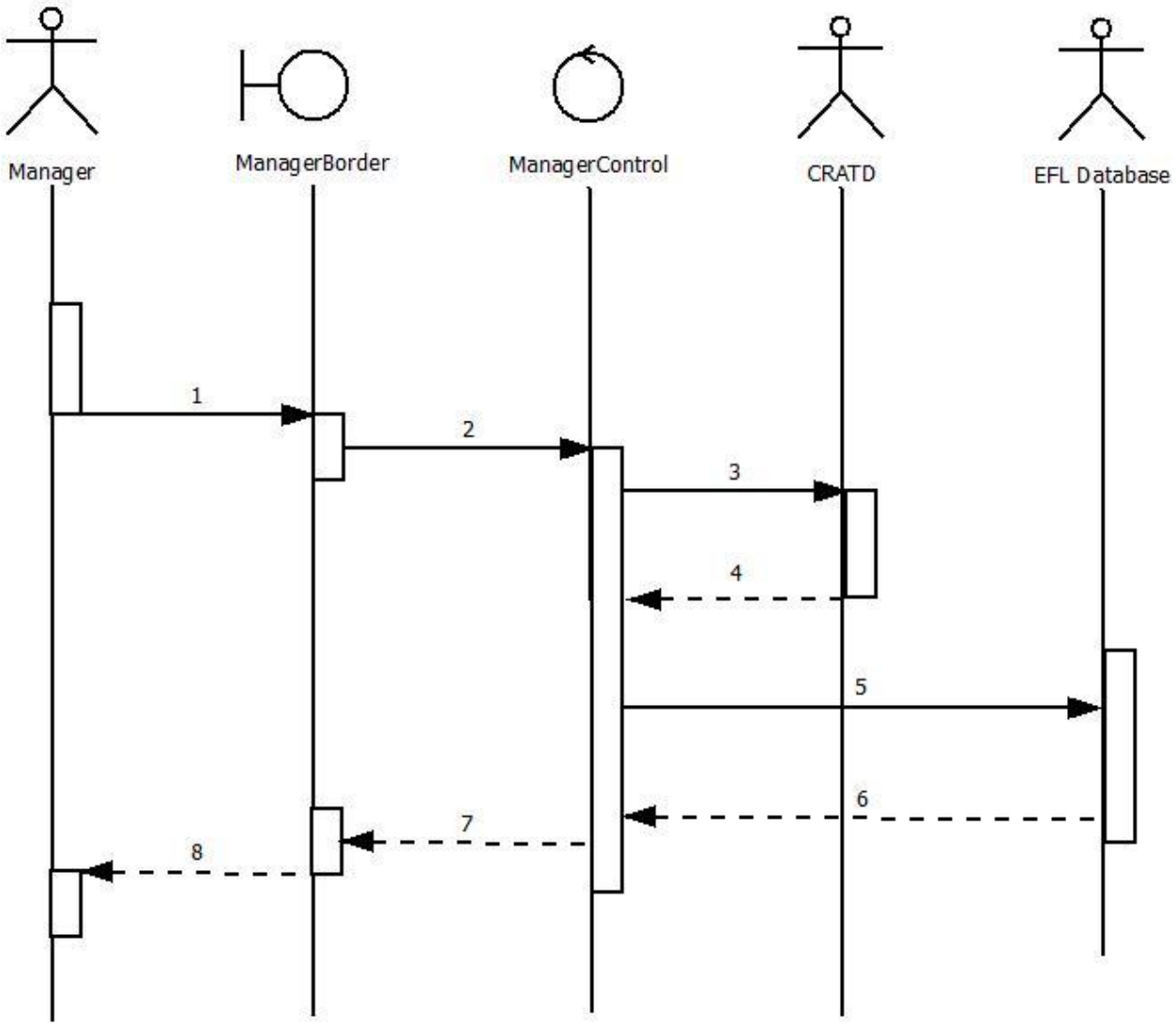
Produce Daily Reports



Produce Daily Reports(Normal)

1. Manager requests for daily reports to be computed
2. ManagerBorder .ProduceDailyReports()
3. AgentControl.eflDatabase
4. EFL Database returns requested information
5. AgentControl.ProduceDailyReports()
6. Updated reports are updated to the EFL Database
7. AgentControl returns information from Daily reports
8. ManagerBorder.UpdateFields()

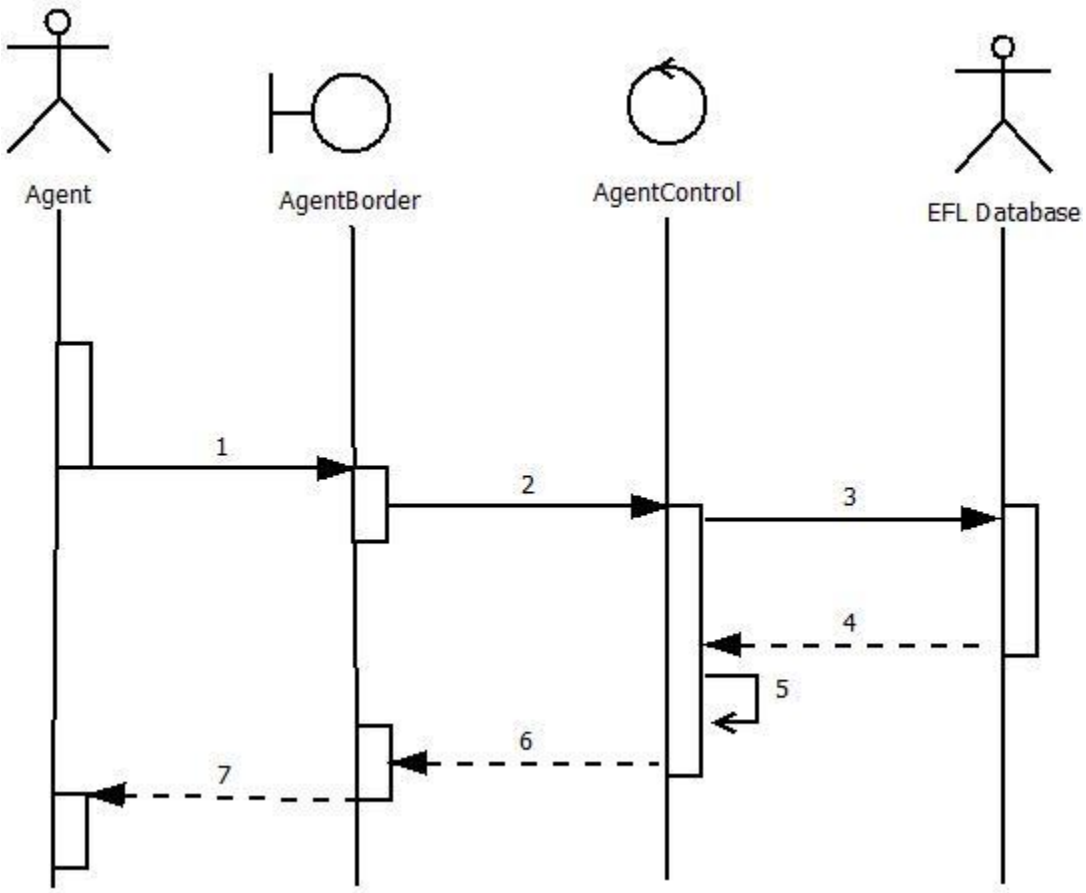
Get Updates From CRATD



Get Updates From CRATD (Normal)

1. Manager initiates downloading of updates from CRATD
2. ManagerBorder.GetCRATDUpdates()
3. ManagerControl .cratd.getUpdates ()
4. CRATD returns updates to the ManagerControl
5. ManagerControl.eflDatabase.airports.setFields()
6. EFL Database returns success message to the ManagerControl
7. ManagerControl returns success message to the ManagerBorder
8. ManagerBorder.DisplayMessage("CRATD Updates Successful")

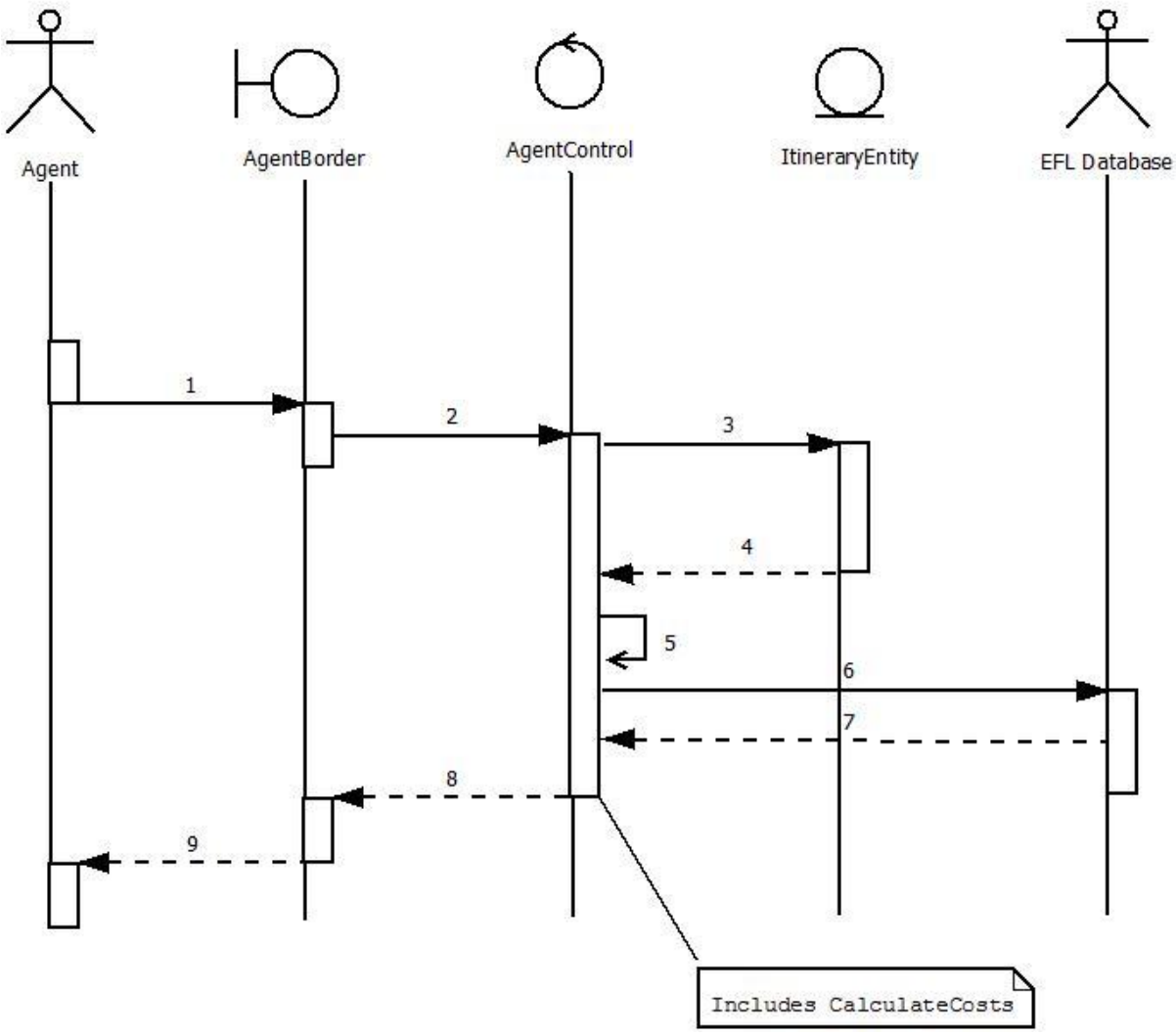
Calculate Costs



Calculate Costs (Normal)

1. Agent requests for a calculate costs action by submitting flight information into AgentBorder
2. AgentBorder.SearchFlights()
   (Note: Other methods that call Calculate Costs; SearchFlights() is the most common.)
3. AgentControl.eflDatabase.getFlights()
4. EFL Database returns information to AgentControl
5. AgentControl.ComputeCosts
6. AgentControl returns updated information to AgentBorder
7. AgentBorder.UpdateFields()

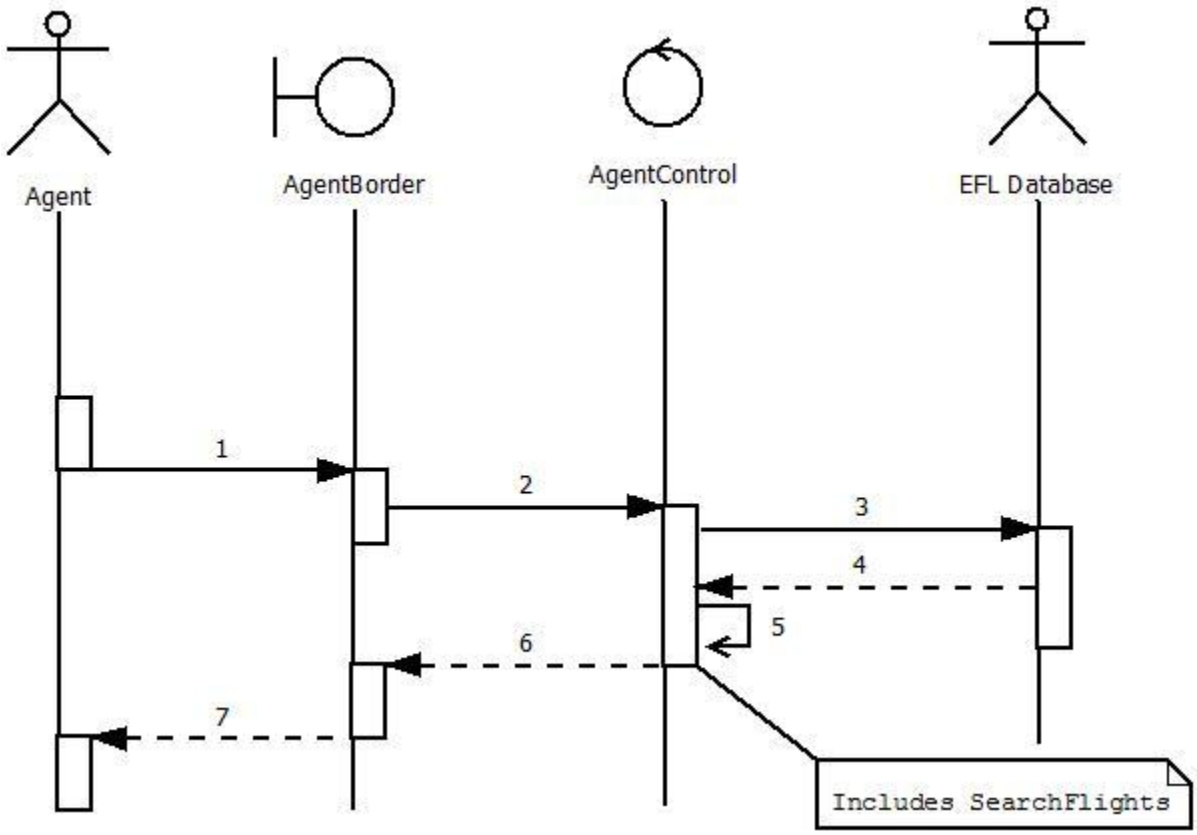
Search Flights



Search Flights (Normal)

1. Agent Requests for a list of current flights that matches an Itinerary given by the Itinerary number
2. Agent hits submit and AgentBorder forwards the information to AgentControl
3. ItineraryEntity()
4. ItineraryEntity returns a new instance of itself to AgentControl
5. AgentControl.SearchFlights(AgentControl.eflDatabase.getItinerary(itineraryID))
6. AgentControl.eflDatabase.getFlightsList()
7. EFL Database returns the list to AgentControl
8. AgentControl returns the list to AgentBorder
9. AgentBorder .UpdateFields()

Provide Met Watch Scenarios



Provide Met Watch Scenarios (Normal)

1. Agent requests all met Price Watch Scenarios to be listed
2. AgentBorder.ProvideMetWatchScenarios()
3. AgentControl.eflDatabase.getCustomer(customerID).getItinerary(itineraryID).getPriceWatch()
4. EFL Database returns a list of all current Prices Watches
5. AgentControl.ProvideMetWatchScenarios()
6. AgentControl returns a list of Met Price watches with flight information
7. AgentBorder .UpdateFields()