 UNIVERSITY OF CALGARY	Course Number: SENG 609.22	Course Name: Agent-based Software Engineering
	Session: Fall, 2003	Department: Electrical and Computer Engineering
		Document Type: Project Report

Ranching Agent Assistant System

Eduard Dantsiguer
Kendra Hamilton
SENG 609.22
December 2003

Table of Contents

Table of Contents	1
Abstract	2
Business Case.....	2
System Description	2
Assumptions.....	3
System Requirements.....	3
System Design Document.....	4
System Architecture.....	4
Agent Internal Architecture	6
Discovery overview	6
Communication protocol	7
Agent Use Cases	8
Detailed Design.....	16
Sequence Diagrams and Walkthroughs	21
Inter Agent Messages.....	26
Data Design.....	30
ER Diagram	30
Data Definition.....	30
Desirable Future Enhancements	35
Bibliography	36

Abstract

Ranching is increasingly more competitive and time consuming. In today's world ranchers can use every advantage they can get to improve their difficult financial conditions. Managing the heard manually is not a simple task. It takes a lot of time and effort to research the best times and places to purchase and sell cattle. It is not trivial to find time and cost effective methods of transporting cattle. With numerous medical advances taking place, one needs to stay abreast of the situation to best protect the heard against possible deceases.

This paper presents an overview, analysis and design of RAAS system which aids ranchers in researching information relevant to the ranching profession and transacting over the Internet.

Business Case

Ranching has long played an important part in Alberta's economy. In 1885, ranching exports contributed eleven million dollars annually to the economy [1]. Today, the cattle industry contributes 3.3 billion dollars annually based on 5 million head of cattle and 21 million hectares of land [2]. However, as ranching is becoming increasingly commoditized, ranchers need to be aware of current cattle prices and trends, environmental trends and forecasts, changes in laws and tariffs related to shipping and exports, and disease control mechanisms.

Increasingly, the Internet is providing an ideal mechanism for ranchers to research the latest issues which impact their profession, to place transactions on cattle, and to procure transportation for their cattle [3]. Unfortunately, the day-to-day demands of the ranching business leave little time for ranchers to research and analyse these issues. We propose the RAAS (Ranching Agent Assistant System) to enable ranchers to more effectively use information and transaction services available via the Internet in order to maximize their profit and minimize the time needed in research and transacting.

System Description

- RAAS is a multi agent system to help ranchers improve profit and minimize research and transact time.
- RAAS will provide the following services: the rancher will be able to buy cattle via the RAAS system, the rancher will be able to sell cattle via the RAAS system, the RAAS system will arrange transportation of cattle which was bought or sold via the system, the RAAS system will regularly research medical information services for information on cattle which may be of concern to the rancher.
- RAAS stores information on individual rancher preferences, using this information in determining which cattle auction services and transportation services to deal with. As well, RASS uses personal preferences and user feedback to determine what types of medical literature and news the rancher would be interested in.
- RAAS deals with the government, research groups, news agencies, shipping companies, and auctions via existing agent services available through the Internet.

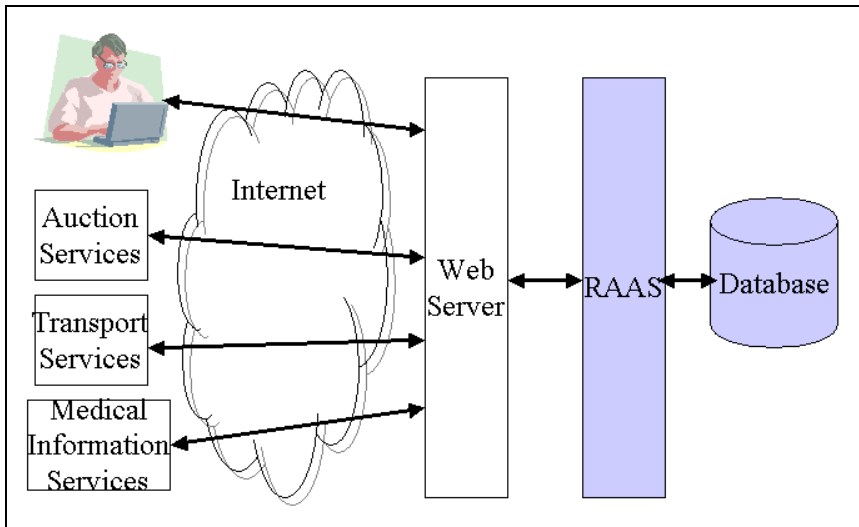


Figure 1 System Diagram

Assumptions

- System maintains a user profile on each rancher (service requester) of their personal preferences such as:
 - Preferred auction house to buy cattle from or sell cattle to
 - Preferred transportation company to use for transporting cattle
 - Type of medical information which is of interest
- The user interacts with the system to buy, sell, or transport cattle through the user interface. The user inputs information concerning the cattle (such as number, type, weight, age) and minimum or maximum price. The system shall locate the auction house and transportation company via agent services provided on the Web but shall consider user preferences in determining the final choice.
- When purchasing or selling cattle via the RAAS system, the user must accept the transportation services provided. There is not an option to buy or sell cattle and not transport it.

System Requirements

It is important to note that this section is not meant to present an extensive list of requirements for RAAS system. Instead, it is meant as a list of major requirements that are central to desired system operation.

1. RAAS shall allow rancher to purchase cattle at an auction
2. RAAS shall allow rancher to sell cattle at an auction
3. RAAS shall assist in selecting the best auction house for buying and selling cattle
4. RAAS shall attempt to minimize price paid and maximize price received during cattle transactions
5. RAAS shall allow to transport cattle to required destination
6. RAAS shall assist in finding the best transportation and optimal transport cost and schedule
7. RAAS shall actively search for new medical advances that become available
8. RAAS shall notify the user of any new medical advances that become available

System Design Document

System Architecture

- In order to achieve the customer requirements, the RAAS system will a architectural layer between the individual rancher's existing browser and existing agent services available over the Internet such as auction house service provider agents and transportation service provider agents.
- RAAS will query directory service agents for list of service provider agents which provide specific services such as auctions, transportations, or medical information.
- Of the list of service provider agents returned by the directory service, RAAS will determine which service provider agents to request quotations from. Based on the response from the request for quotation and the rancher's personal preferences, RASS will determine which service provider agents to arrange services with.
- Prior to using RAAS for purchasing, selling, transporting, or researching cattle, the rancher must create an initial user preference profile using the customer service agent.

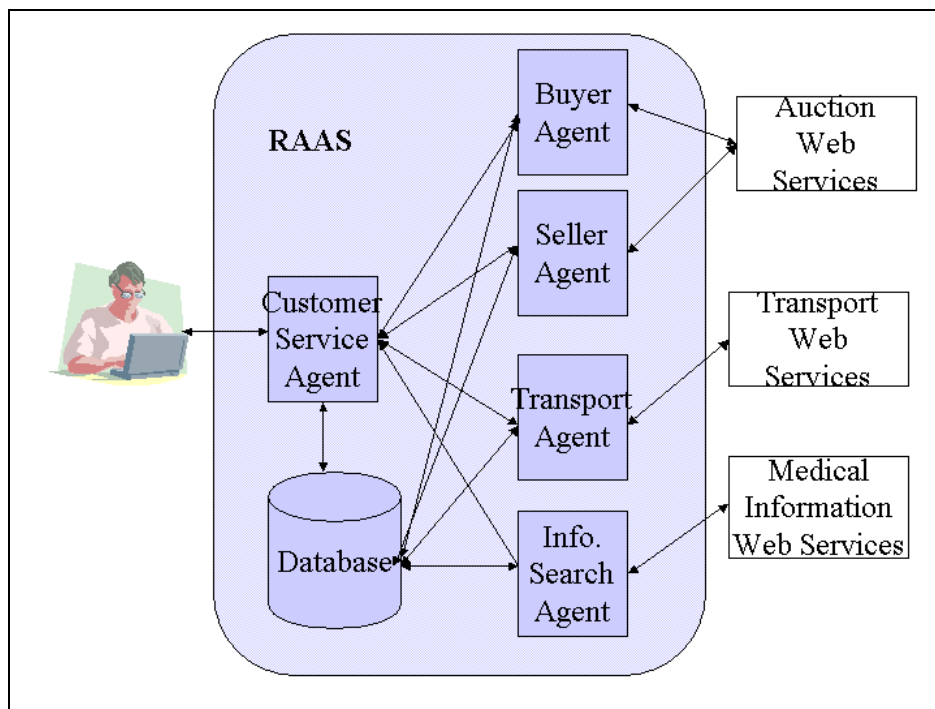


Figure 2 System Architecture

Roles

Customer Service Role

The Customer Service role is responsible for interfacing through a GUI interface with individual ranchers, responding to their requests, and keeping a profile on their buying, selling, transportation, and medical information preferences.

If the rancher requests that cattle be bought or sold, the Customer Service role is responsible for forwarding this request to the Buyer or Seller role

along with customer preference details. When the Buyer or Seller role responds with the details of the sale, the Customer Service role is responsible for notifying the rancher with the details of the deal. It is responsible for contacting the agent tagged with the Cattle Transporter role to arrange transportation services for the cattle that has been bought or sold.

If the Medical Search role has found news of potential interest to the rancher, the Customer Service role is responsible for presenting the information to the rancher.

Buyer Role

The Buyer role intercepts requests for purchasing cattle from the Customer Service role. The Buyer role registers a request for a list of cattle auction service agents from a predefined Service Directory agent on the Web. Once the Buyer role has received a reply from the Service Directory agent, the Buyer role issues bids with Cattle Auction Service Provider agents. If the Buyer successfully wins a bid, it replies to the Customer Service Agent with the details of the sale.

Seller Role

The Seller role intercepts requests for sale of cattle from the Customer Service role. The Seller role registers a request for a list of cattle auction service agents from a predefined Service Directory agent on the Web. Once the Seller role has received a reply from the Service Directory agent, the Buyer role issues bids with Cattle Auction Service Provider agents. If the Cattle Auction Service Provider accepts the terms of the sale, the Seller role will notify the Customer Service role with the details of the sale.

Cattle Transport Role

The Cattle Transport agent intercepts requests from the agent responsible for the Customer Service role for transportation of cattle. The Cattle Transport Agent registers a request for a list of transportation service provider agents with a predefined Service Directory agent on the Web. Based on details provided by the Customer Service agent and the list provided by the Service Directory agent, the Cattle Transport issues a request for quotation to the Cattle Transport Service Provider agents. Once the Cattle Transport agent has received replies to the request for quotation, it chooses the appropriate Cattle Service Transportation Provider and schedules the actual transportation. The Cattle Service Transportation Provider is responsible to let the Customer Service Agent know the transportation details.

Medical Search Agent

The Medical Search agent is responsible for proactively searching out new medical information on cattle which may be of interest to the rancher. The Medical Search agent interfaces with a Medical Service Directory on the Web in order to receive a list of Web sites which might have applicable information. Based on the ranchers preferences, the Medical Search agent reviews r recent news and reports the information back to the Customer Service agent for presentation to the rancher.

Agent Internal Architecture

Figure 3 represents the internal architecture of agents in the RAAS system.

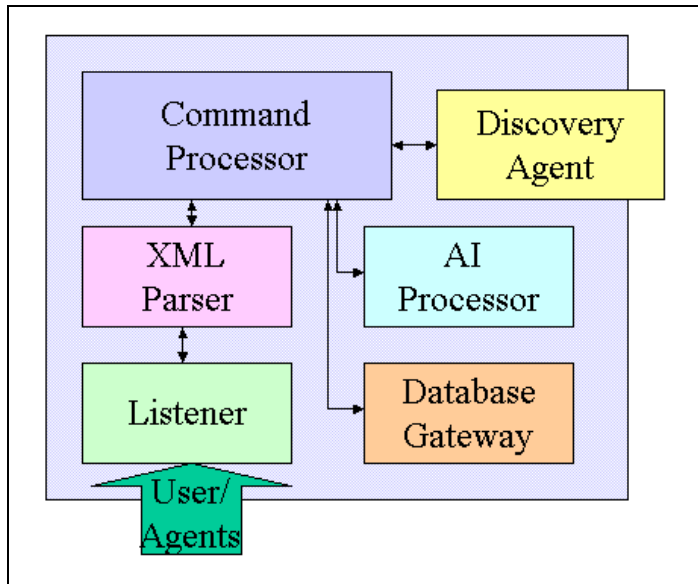


Figure 3 Agent internal architecture

The agent internal architecture is made up of the following components:

1. The **listener** is responsible for listening to a port for incoming messages from the user or other agents.
2. The **XML parser** is responsible for parsing and interpreting messages. It assumes that all messages are in an agreed upon document type definition.
3. The **command processor** receives the interpreted XML messages and invokes appropriate methods
4. The **database gateway** is responsible for handling communication to and from the database.
5. The **AI processor** is responsible for solving problems based on AI methods such as fuzzy logic or predicate logic reasoning.
6. The **discovery agent** is an external service which provides service discovery.

Discovery overview

In order for the Medical Information search agent to be able to find information relevant to the rancher's interests, it must be able to discover and communicate with services which have appropriate information. WDSL^[5] (Web Service Definition Language) defines an XML grammar to describe services as endpoints which can communicate. THE WDSL document is constructed from seven elements:

1. Types - data type definitions
2. Message - an abstract definition of the data being sent
3. Operation - an abstract description of the service provided
4. Port Type - an abstract description of operations provided
5. Binding - the protocol and data format specification of a specific port type
6. Port - a specific combination of binding and network address
7. Service - a set of related endpoints.

Communication protocol

SOAP[4], the chosen communication protocol for RAAS is a lightweight means of exchanging information between objects. Of particular benefit to the multiagent system with mobile agents, SOAP can be used in a distributed and decentralized environment. SOAP messages are constructed from three sections:

1. The *envelope* provides a format for describing the format of a message, who should act on the message, and how act on the message.
2. The *encoding rules* used to express application defined datatypes.
3. A *convention* to represent remote procedure calls and responses.

Due to the reliance of RAAS on external agents which are accessible via the Internet, the SOAP protocol will be used to encode an HTML header and XML message body. XML messages are further defined later in this document.

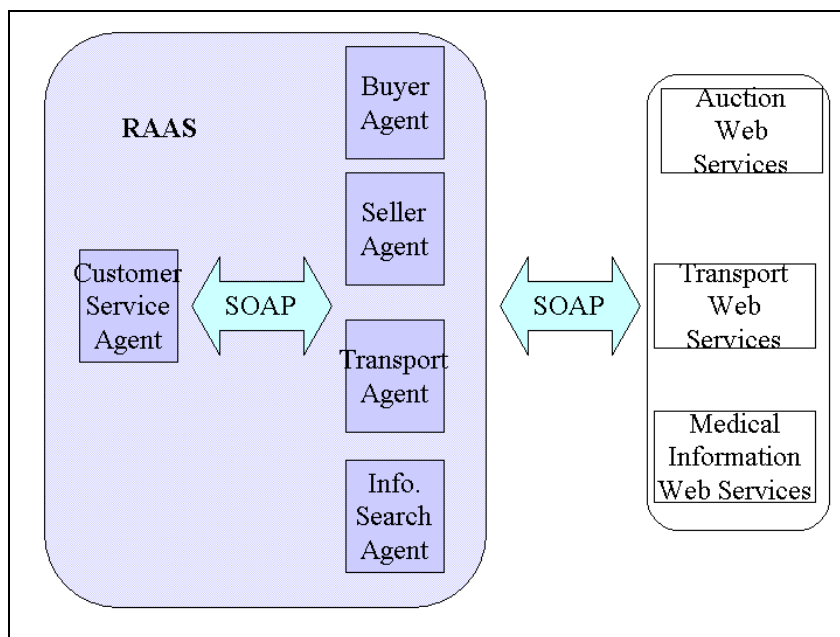


Figure 4 Communication Protocol

Agent Use Cases

Before the system can be designed, it needs to be better understood. Through a series use cases, this improved understanding will be achieved.

Overall Use Case

Overall, high-level use case is illustrated in Figure 5. It will then be broken down into more detailed use cases.

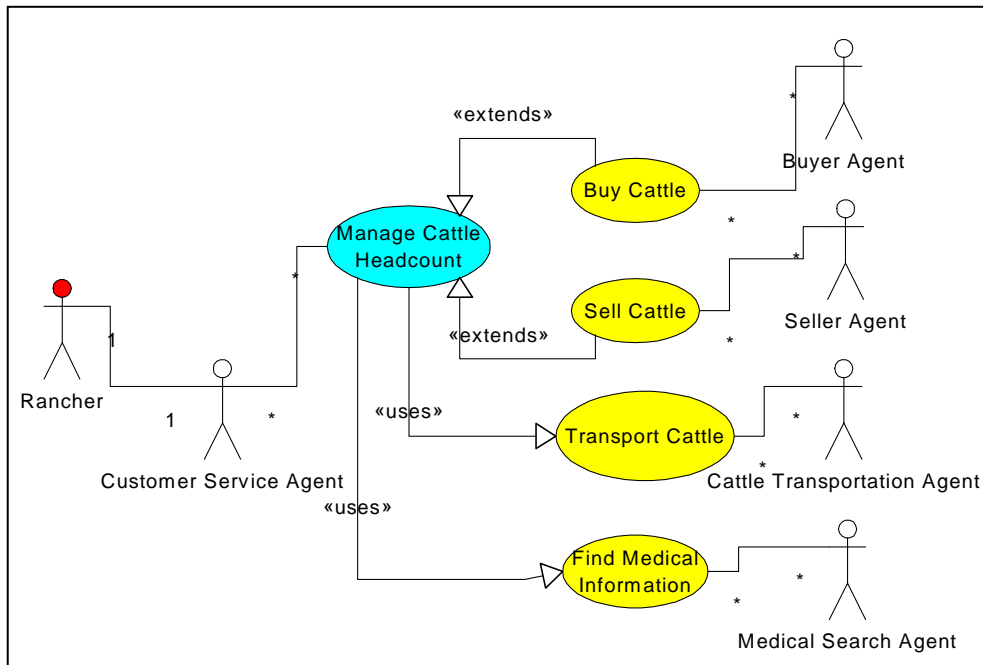


Figure 5 Overall use case

Brief Description	High-level view of main functionality
Precondition(s)	All agents exist
Post Condition(s)	
Process Steps	
1	Rancher needs to manage his/her heard. This includes buying more cattle, selling some existing cattle, transporting cattle that has been bought/sold, staying current with medical developments to ensure the health of the heard. Rancher will make these requests to the Customer Service Agent, which will handle the rest.
2	Customer Service Agent will delegate rancher's request
Exceptions	
	None
Relationships	
Initiating	Rancher
Collaborating	Transportation Service Directory Provider, Transportation Service Provider, Customer Service Agent

Data Requirements	
Data Required:	Action to take: buy/sell/transport Quantity of cattle involved Purchase/sale price

Buy Cattle Use Case

A more detailed use case diagram for purchasing cattle is presented in Figure 6, below. Here, activities that compose "Buy Cattle" action are illustrated

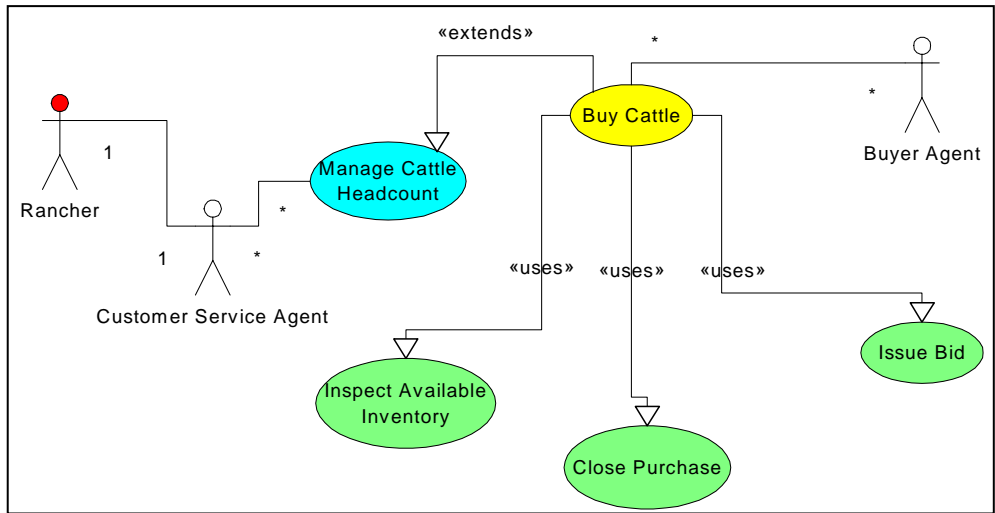


Figure 6 Buy Cattle Use Case

Brief Description	Purchase of cattle
Precondition(s)	Rancher instructs the system that he wants to purchase cattle.
Post Condition(s)	Purchase is either completed or no buyers are found.
Process Steps	
1	The Customer Service agent makes a request for purchase of cattle by providing the number of cattle, the type of cattle, the age of the cattle, the purchase date, and optionally a maximum allowed price.
2	The Buyer Agent makes a request to the Auction Services Directory Agent for a list of cattle Auction Service Providers.
3	The Auction Services Directory Agent replies to the Buyer agent with a list of Auction Service Providers and inspects available inventory.
4	The Buyer Agent sends out requests for minimum bids allowed to the Auction Service Provider Agents; specifying the number of cattle, the type of cattle, the age of the cattle, the purchase date, and a reference number.
5	The Buyer Agent receives replies from the Auction Service Provider Agents specifying the purchase reference number and the minimum bid.

6	The Buyer Agent decides which Auction Service Provider to use based on the minimum bid and the rancher's preferred Auction Service Provider.
7	The Buyer Agent sends a message to the chosen Auction Service Provider specifying the number of cattle, type of cattle, age of cattle, purchase reference number, and bid.
8	The Buyer Agent receives a reply from the Auction Service Provider agent specifying the purchase reference number and either a request for a higher bid, an award of the sales, or a decline of the sale, and the current bid price. The deal is closed and Customer Service Agent is notified.
Exceptions	
	No cattle to purchase.
Relationships	
Initiating	Customer Service Agent
Collaborating	Auction Service Directory Provider, Auction Service Provider, Cattle Transport Agent
Data Requirements	
Data Required:	Data Required for the Buyer Agent: Purchase id Service Requester Auction Service Provider Number of cattle Type of cattle Age of cattle Total weight of cattle Maximum purchase price Purchase price Source location Pickup date Delivery location Last possible delivery date

Sell Cattle Use Case

A more detailed use case diagram for selling cattle is presented in Figure 7, below. Here, activities that compose "Sell Cattle" action are illustrated.

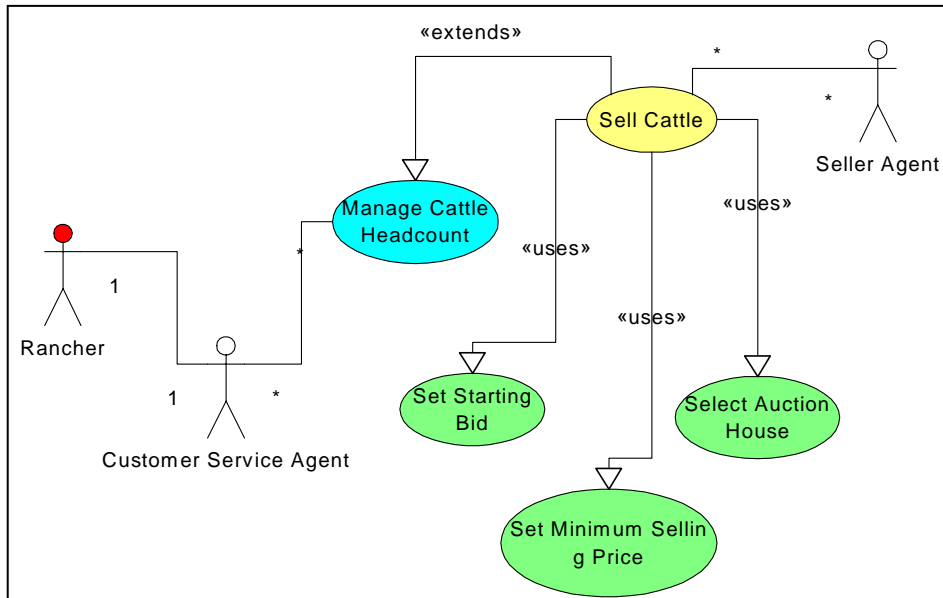


Figure 7 Sell Cattle Use Case

Brief Description	Sale of cattle
Precondition(s)	Rancher instructs the system that he has cattle for sale.
Post Condition(s)	Sale is either completed or no buyers are found.
Process Steps	
1	The Customer Service agent makes a request for sale of cattle by providing the number of cattle, the type of cattle, the age of the cattle, the purchase date, and a minimum acceptable price.
2	The Seller Agent makes a request to the Auction Services Directory Agent for a list of cattle Auction Service Providers.
3	The Auction Services Directory Agent replies to the Seller agent with a list of Auction Service Providers.
4	The Seller Agent sends out a starting bid price to the Auction Service Provider Agents; specifying the number of cattle, the type of cattle, the age of the cattle, the purchase date, and a reference number.
5	The Seller Agent receives replies from the Auction Service Provider Agents specifying the sale reference number and the minimum bid.
6	The Seller Agent decides which Auction Service Provider to use based on the minimum bid and the rancher's preferred Auction Service Provider.
7	The Seller Agent sends a message to the chosen Auction Service Provider specifying the number of cattle, type of cattle, age of cattle, sale reference number, and minimum sale price.

8	The Seller Agent receives a reply from the Auction Service Provider agent specifying the sale reference number and either a decline of the sale or indication of success. The deal is then closed and Customer Service Agent is notified.
Exceptions	
	No interested buyers.
Relationships	
Initiating	Customer Service Agent
Collaborating	Auction Service Directory Provider, Auction Service Provider, Cattle Transporter Agent
Data Requirements	
Data Required:	Data Required for the Seller Agent: Purchase id Service Requester Auction Service Provider Number of cattle Type of cattle Age of cattle Total weight of cattle Minimum sale price Sale price Source location Pickup date Delivery location Last possible delivery date

Transport Cattle Use Case

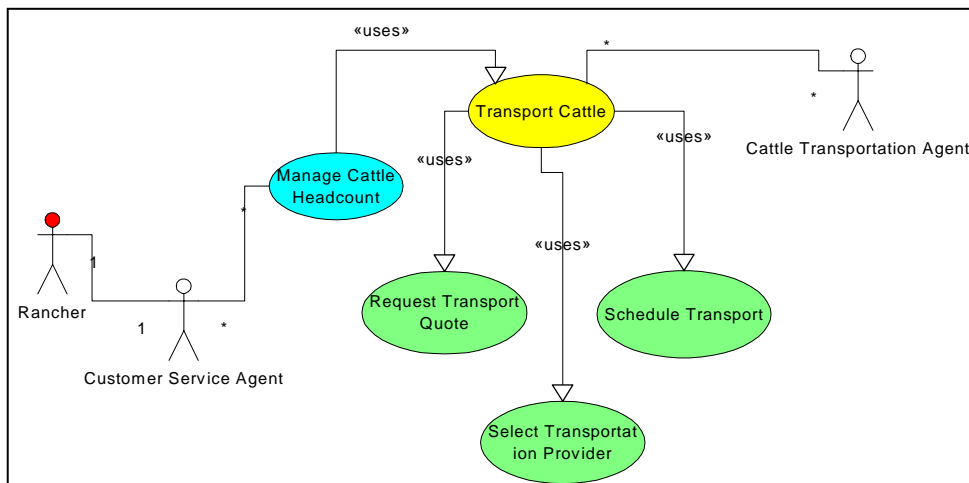


Figure 8 Transport Cattle Use Case

A more detailed use case diagram for transporting cattle is presented in Figure 8, above. Here, activities that compose "Transport Cattle" action are illustrated

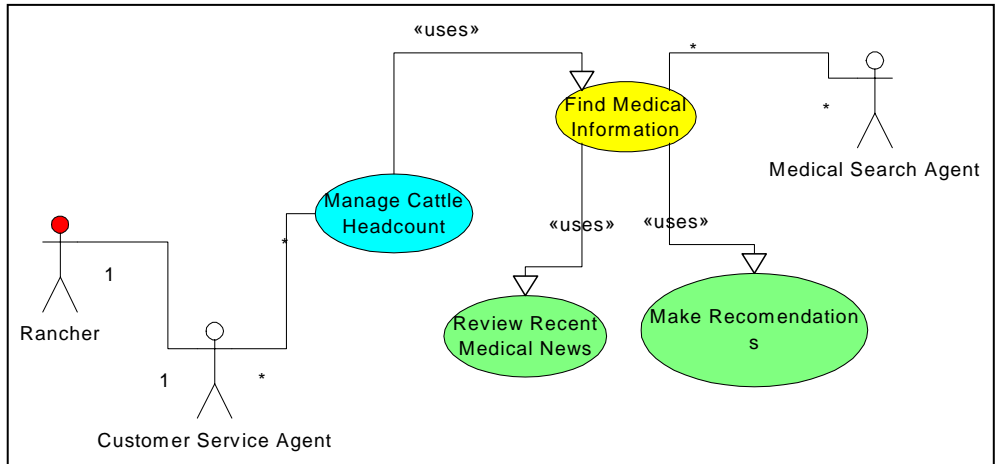
Brief Description	Transportation of cattle
Precondition(s)	Cattle has been purchased or sold
Post Condition(s)	Cattle transport scheduled
Process Steps	

1	The Customer Service Agent makes a request for transportation by providing the number of cattle, the total weight of the cattle, the source location, the delivery location, the date of pickup, the last possible delivery date, and optionally a maximum allowed price.
2	The Cattle Transportation Agent makes a request to the Transport Services Directory Agent for a list of Transportation Service Providers which can transport cattle and which can transport between the source and delivery locations.
3	The Transport Services Directory Agent replies to the Cattle Transport agent with a list of Transport Service Providers which can transport cattle and which can transport between the source and delivery locations.
4	The Cattle Transport Agent sends out requests for quotes to the Transport Service Provider Agents; specifying the number of cattle, the total weight of the cattle, the source location, the delivery location, the date of pickup, the last possible delivery date, and a reference number.
5	The Cattle Transport Agent receives replies from the Transport Service Provider Agents specifying the transport reference number and the price.
6	The Cattle Transport Agent decides which Transport Service Provider to use based on the price and the rancher's preferred Transport Service Provider.
7	The Cattle Transport Agent sends a message to the chosen Transport Service Provider specifying the transport identifier, service transport reference id, and acceptance of the terms of the quote.
8	The Cattle Transport Agent receives a message from the Transport Service Provider agent specifying the transportation id, service provider reference id, source pickup date and time, and delivery date and time.
9	The Cattle Transport Agent will send a message to the Customer Service Agent confirming transportation services and specifying the sales/purchase id, the transport service provider, the transport service reference, the number of cattle being transported, the total weight of the cattle, the pickup date and time, the delivery and time, the pickup source location, the delivery location, and the price.
Exceptions	
	No service available to desired location.
Relationships	
Initiating	Customer Service Agent
Collaborating	Transportation Service Directory Provider, Transportation Service Provider, Customer Service Agent
Data Requirements	

Data Required:	Data Required for the Cattle Transporter Agent: Transportation id Service provider reference id Sales/Purchase id Number of cattle Total weight of cattle Source location Delivery location Date/Time of pickup Last possible delivery date Date/Time of delivery Maximum allowed price Transportation Service Requester Transportation Service Provider Price for transportation
----------------	---

Medical Research Use Case

Figure 9 Medical Research Use Case



Brief Description	Research for new medical information.
Precondition(s)	None
Post Condition(s)	New medical information found
Process Steps	
1	Medical Search Agent searches for new cattle information by querying Medical Service Directory for a list of information service providers.
2	Medical Service Directory responds with a list of medical information service providers.
3	Medical Search Agent uses this list of providers to search for new medical information.
4	Medical Search Agent returns any new information that it finds to Customer Service Agent.
Exceptions	
	No new medical information is available.
Relationships	
Initiating	Medical Search Agent
Collaborating	Medical Service Directory Provider, Medical Information Agent, Customer Service Agent

Data Requirements	
Data Required:	Data Required for the Medical Search Agent: Type of medical information required

Detailed Design

Before looking into details of each class involved in this system, it is important to mention how these classes relate to each other and what dependencies exist among them. Figure 10, below, shows the overall system class hierarchy. Please note that such details as methods and attributes of all classes involved are omitted at this stage. They will be described in more detail later in this document.

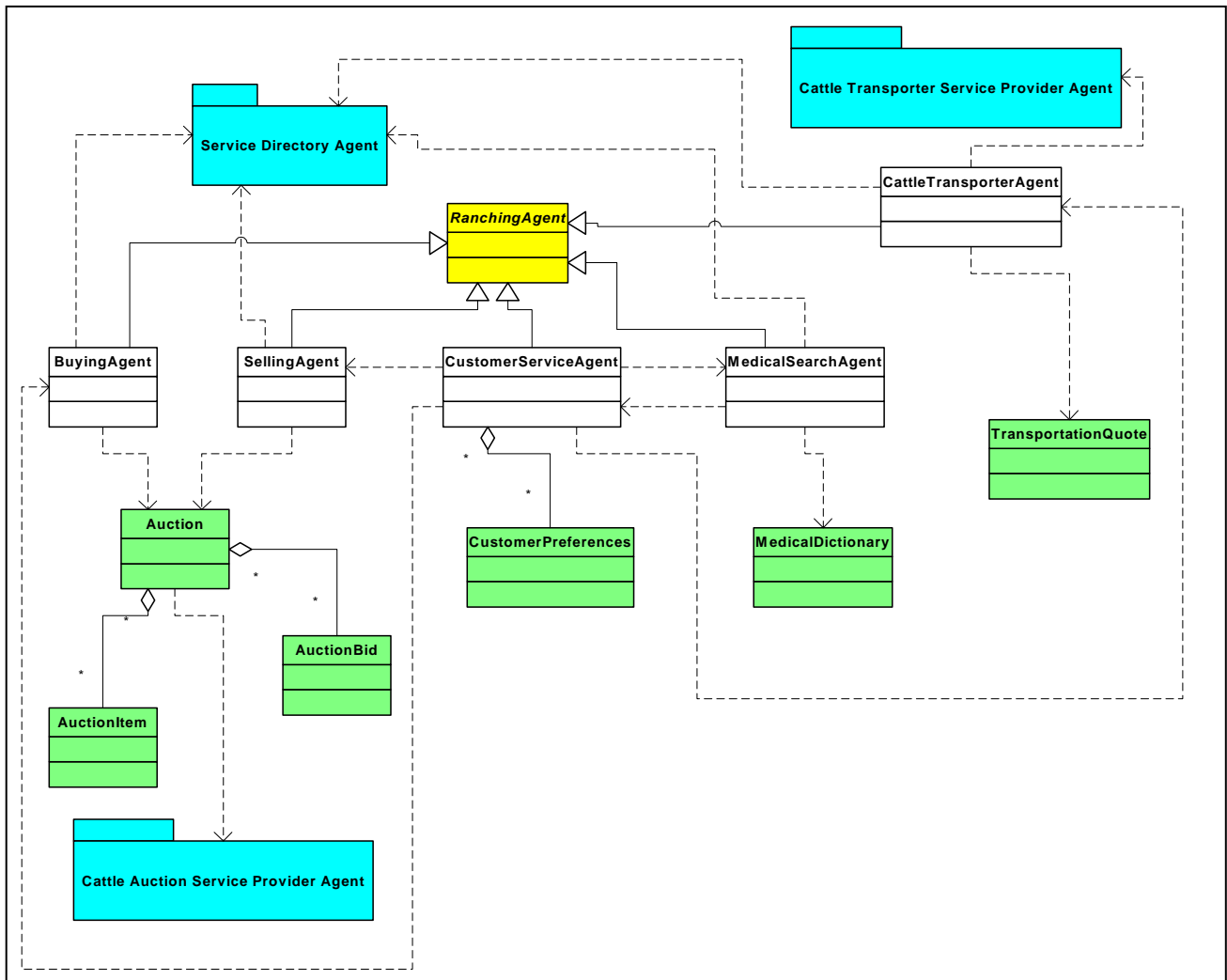


Figure 10 System Class Hierarchy

Agents illustrated in blue are external to the system. The system interacts with them and delegates tasks to them but has no visibility into their structure.

Classes in white are the five agents that compose RAAS. Classes in green are support classes that agent rely on to accomplish their roles. The next five sections will provide a more detailed descriptions of the classes mentioned thus far.

RanchingAgent Class

RanchingAgent (illustrated in figure 11, below) is the base class for all agents of RAAS. It is implemented as a pure virtual class, meaning that all classes that inherit from it will have to overwrite all methods defined in RanchingAgent.

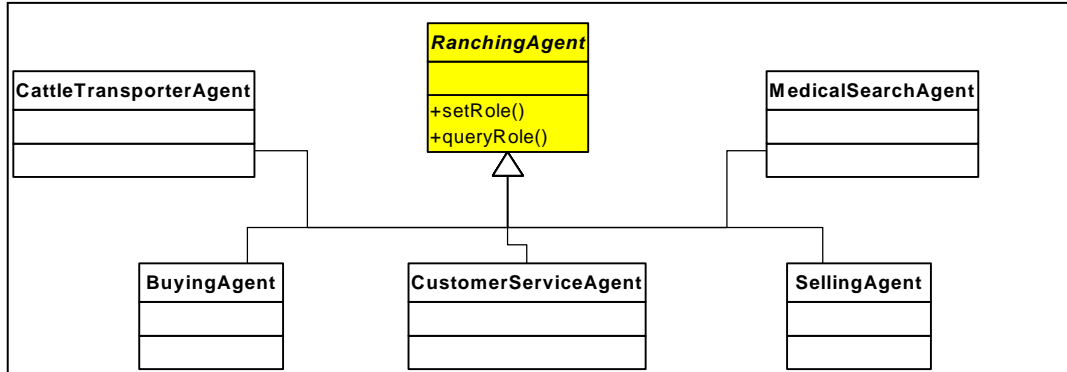


Figure 11 RanchingAgent Class

CustomerServiceAgent Class

CustomerServiceAgent class inherits from RanchingAgent class. CustomerServiceAgent class is illustrated in Figure 12, below.

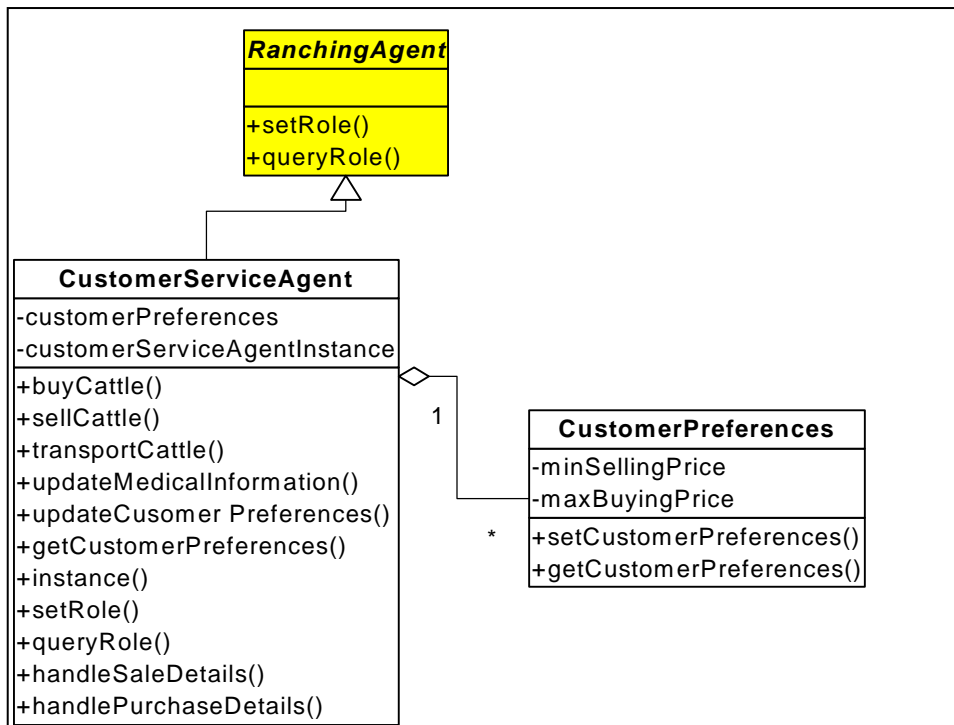


Figure 12 CustomerServiceAgent Class

CustomerServiceAgent includes an object of CustomerPreferences class to store rancher’s preferences when it comes to cattle management (buying/selling/transporting/researching). CustomerServiceAgent class is implemented following the Singleton Design Pattern, meaning that there will only

be a single instance of this class. "instance" method is provide to retrieve a pointer to this object.

The main roles of CustomerServiceAgent class are as follows:

- To help other agents to accomplish their tasks via information sharing.
- To interact with the user (rancher) by accepting his commands and providing feedback of what cattle being purchased or sold, transport information and new medical breakthroughs.
- To delegate tasks to other agents to accomplish rancher's requests.

CustomerServiceAgent class provides the following interfaces to the rancher:

- Buy cattle
- Sell cattle
- Transport cattle
- Update customer preferences

CustomerServiceAgent class provides the following interfaces to the BuyerAgent class (described in a later section):

- Provide user preferences upon request
- Handle notification of cattle being purchased

CustomerServiceAgent class provides the following interfaces to the SellerAgent class (described in a later section):

- Provide user preferences upon request
- Handle notification of cattle being sold

CustomerServiceAgent class provides the following interfaces to the TransportAgent class (described in a later section):

- Provide user preferences upon request
- Handle notification of transportation arrangements being scheduled

CustomerServiceAgent class provides the following interfaces to the MedicalAgent class (described in a later section):

- Provide user preferences upon request
- Handle notification of new medical information being discovered

BuyerAgent Class

BuyerAgent class inherits from RanchingAgent class. BuyerAgent class is illustrated (together with SellerAgent class) in Figure 13, below.

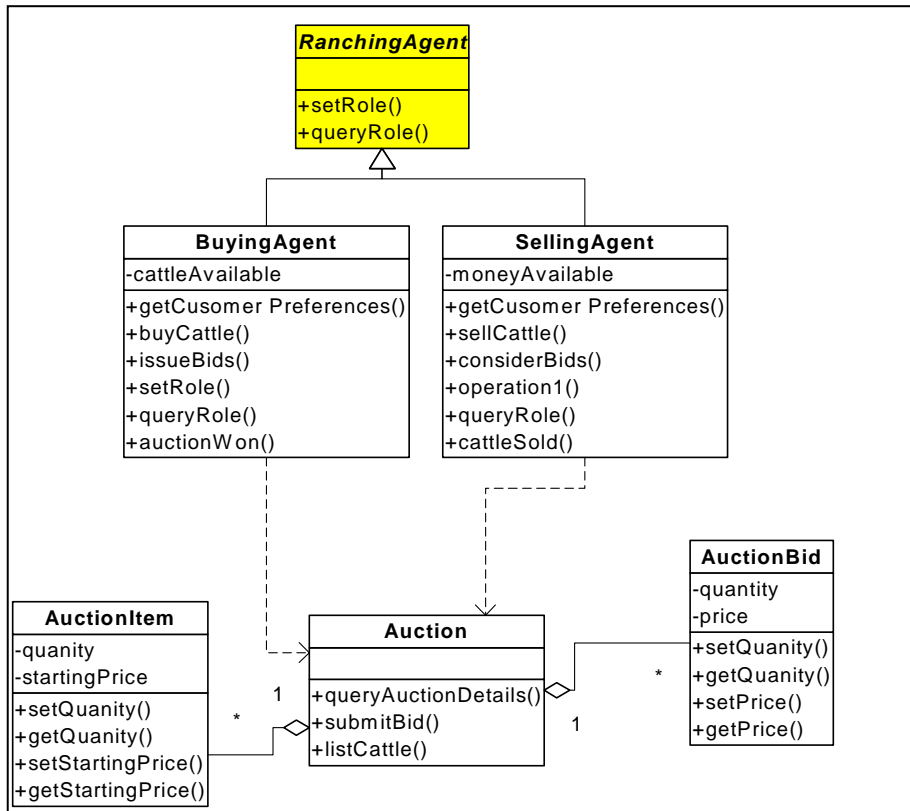


Figure 13 BuyerAgent/SellerAgent Classes

BuyerAgent class interacts with Auction class to assist in capturing information about the selected auction house and placing bids of cattle.

The main roles of BuyerAgent class are as follows:

- Facilitate purchase of cattle
- Communicate with Directory Services Agent and Cattle Service Provider Agent to get the best deal
- Notify Service Provider Agent when the purchase is made

SellerAgent Class

SellerAgent class inherits from RanchingAgent class. SellerAgent class is illustrated (together with BuyerAgent class) in Figure 13, above.

SellerAgent class interacts with Auction class to assist in capturing information about the selected auction house and listing cattle up for bid.

The main roles of SellerAgent class are as follows:

- Facilitate sale of cattle
- Communicate with Directory Services Agent and Cattle Service Provider Agent to get the most money for cattle
- Notify Service Provider Agent when the sale is made

TransportAgent Class

TransportAgent class inherits from RanchingAgent class. TransportAgent class is illustrated in Figure 14.

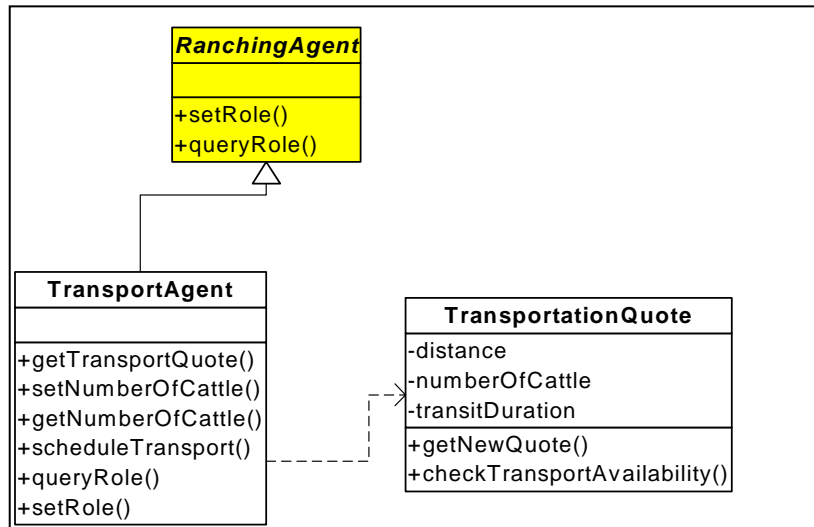


Figure 14 TransportAgent Class

TransportAgent class interacts with TransportationQuote class to assist in getting quotes for transporting cattle to desired location.

The main roles of SellerAgent class are as follows:

- Facilitate transport of cattle
- Communicate with Directory Services Agent and Cattle Transporter Service Provider Agent to get the best price and schedule for transporting cattle
- Notify Service Provider Agent when transport arrangements are made
- Request transport quotes to get the best deal

MedicalSearchAgent Class

MedicalSearchAgent class inherits from RanchingAgent class. MedicalSearchAgent class is illustrated in Figure 15.

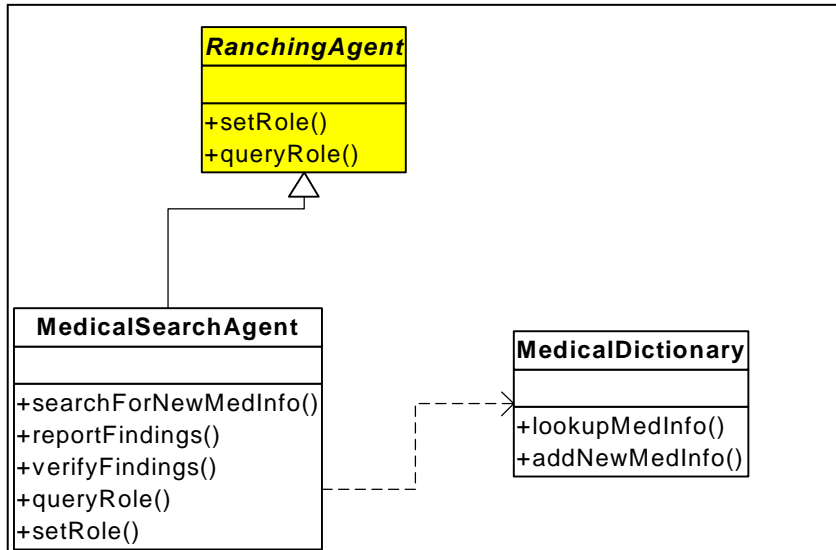


Figure 15 MedicalSearchAgent Class

MedicalSearchAgent class interacts with MedicalDictionary class to assist in assessing relevance of medical information that it finds.

The main roles of SellerAgent class are as follows:

- Proactively search for relevant medical developments
- Communicate with Directory Services Agent to locate providers of medical information
- Notify Service Provider Agent when medical information of interest is found

Sequence Diagrams and Walkthroughs

With classes being defined and class interactions briefly introduced, this sections takes a detailed look at interactions that take place when major system functionality is being performed. Please note that this section does not aim to demonstrate every possible system interaction. It is assumed that interactions not addressed by this section are sufficiently trivial to be omitted.

Purchasing Cattle

This section presents a sequence diagram of what interactions take place when a rancher indicates that he wants more cattle to be purchased. It is assumed that the quantity, type, price and other necessary criteria for cattle to be purchased are specified by the rancher and stored in PurchaseEvents object. These interactions are illustrated in Figure 16, below.

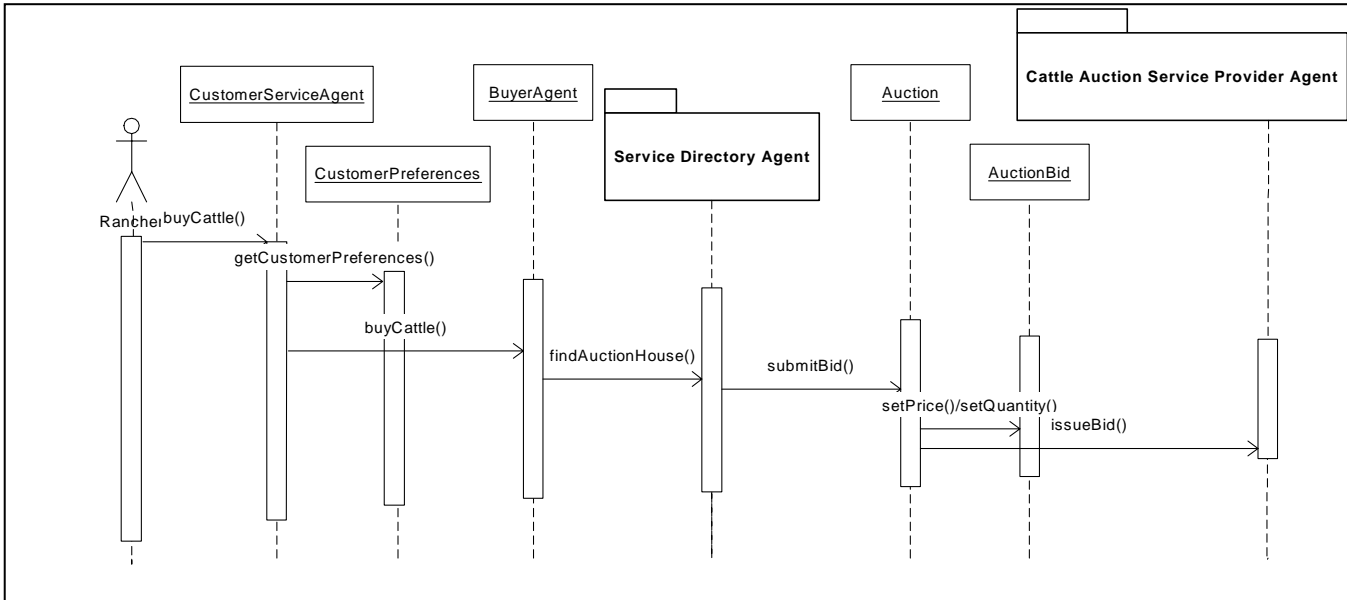


Figure 16 Purchasing Cattle Sequence Diagram

Selling Cattle

This section presents a sequence diagram of what interactions take place when a rancher indicates that he wants to sell some of his cattle. These interactions are illustrated in Figure 17, below.

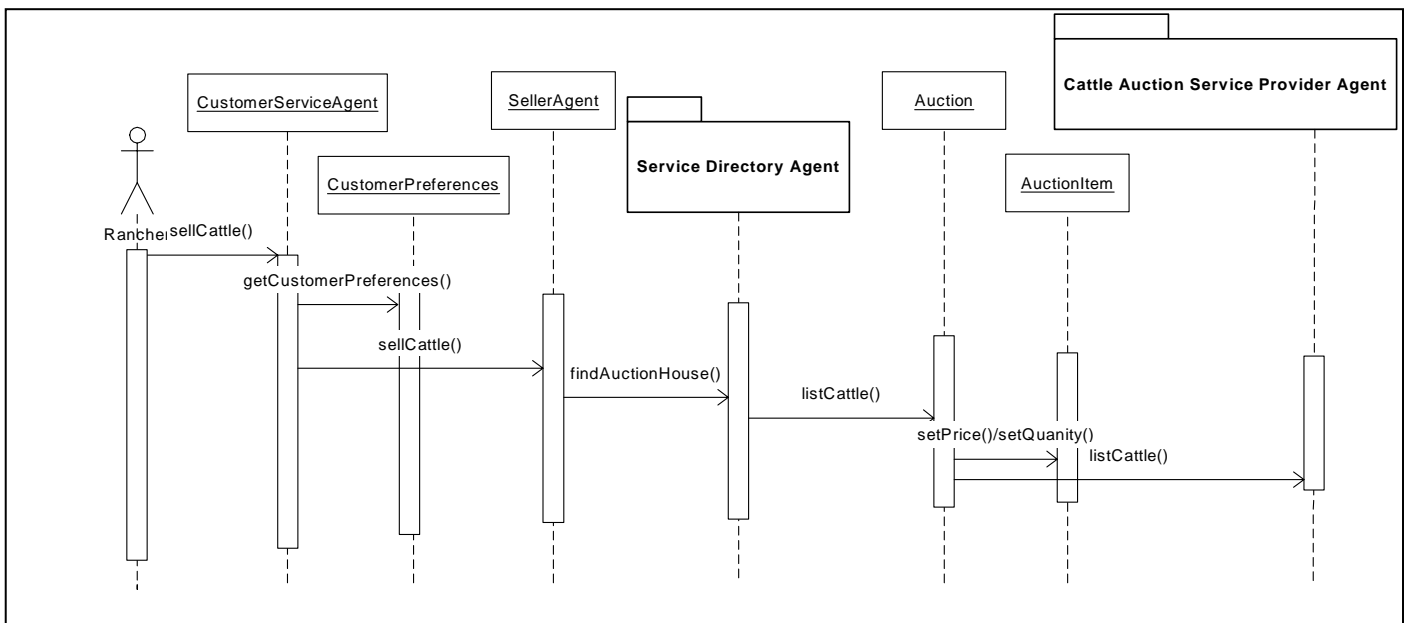


Figure 17 Selling Cattle Sequence Diagram

It is assumed that the quantity, type, price and other necessary criteria for cattle to be sold are specified by the rancher and stored in SalesEvents object.

Cattle Purchased

The bid for cattle has been placed and won. The cattle is purchased but the Customer Service Agent needs to be notified. Sequence diagram (Figure 18) below illustrates interactions involved in this process.

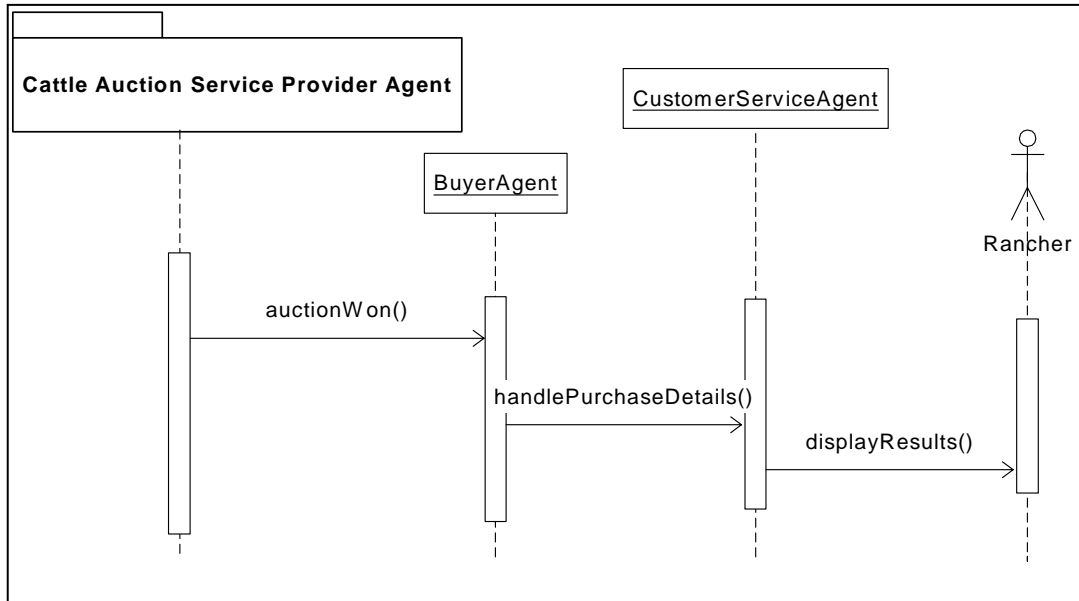


Figure 18 Cattle Purchased Sequence Diagram

Cattle Sold

Cattle has been placed up for bid and sold. The cattle is sold but the Customer Service Agent needs to be notified. Sequence diagram (Figure 19) below illustrates interactions involved in this process.

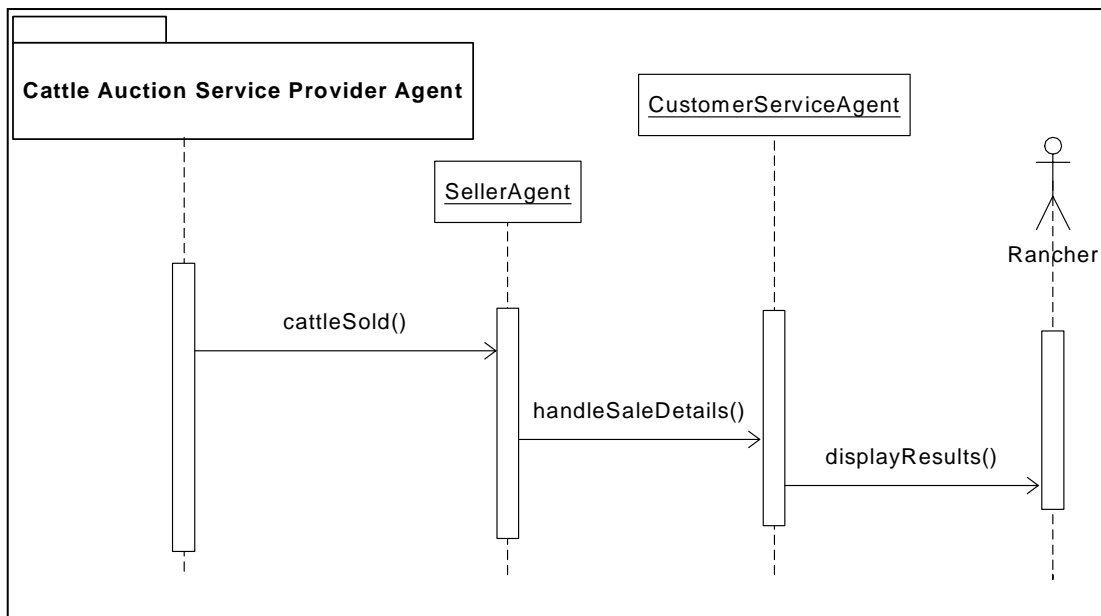


Figure 19 Cattle Sold Sequence Diagram

Cattle Transportation

Customer Service Agent has been notified that cattle has either been purchased or sold. It now needs to ensure that arrangements to transport the cattle are made. Sequence diagram (Figure 20) below illustrates interactions involved in this process.

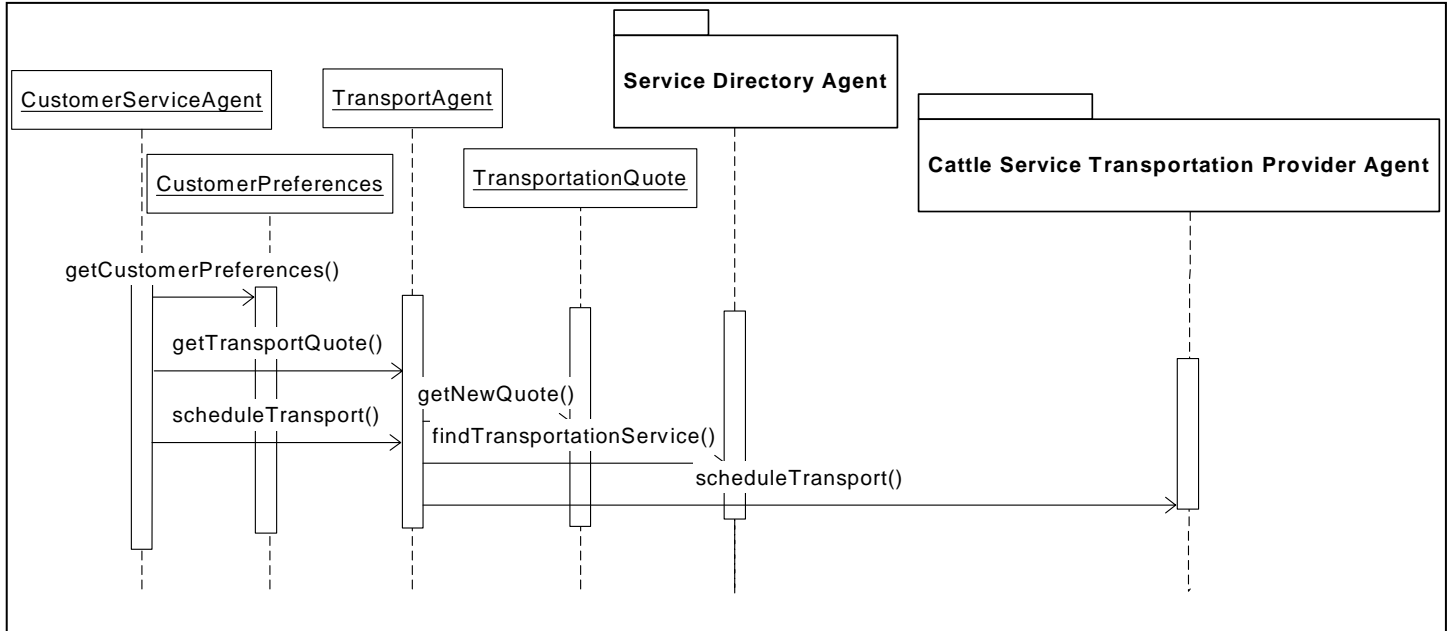


Figure 20 Cattle Transportation Sequence Diagram

Medical Search

Medical Search Agent was proactively looking for new relevant medical information. It found something that it thinks is worthy to report to the rancher. Sequence diagram (Figure 21) below illustrates interactions involved in this process.

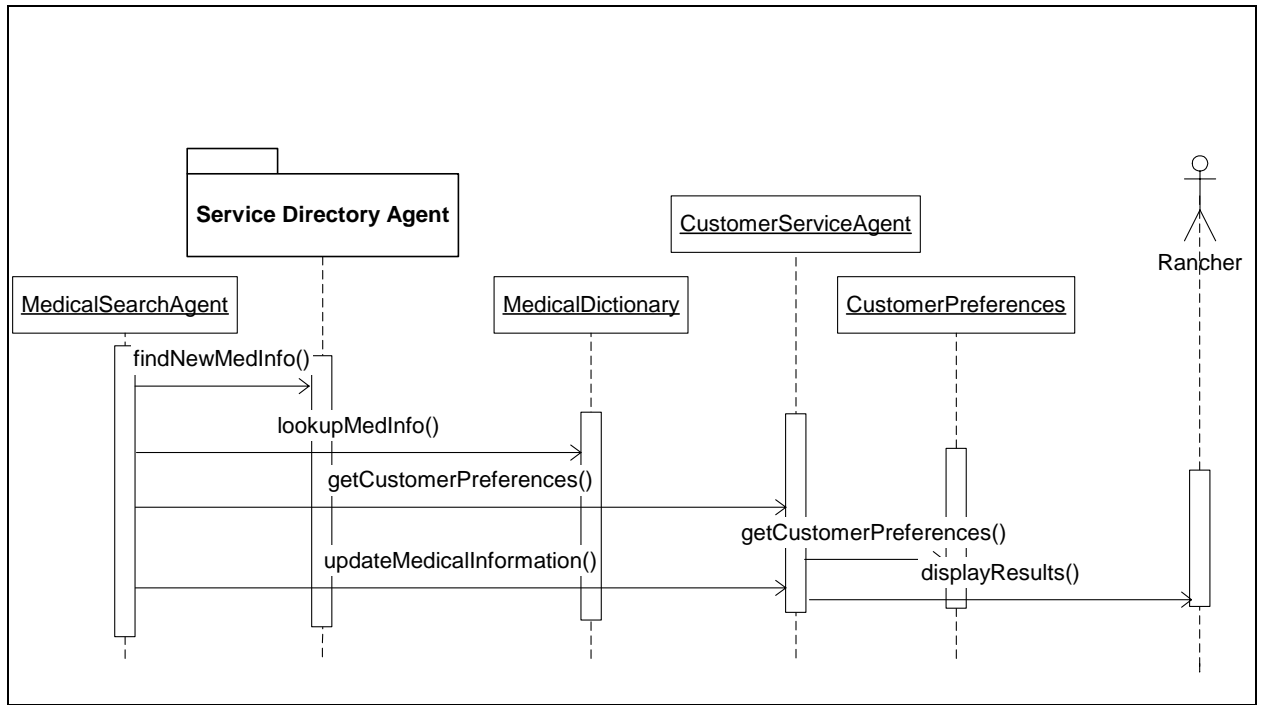


Figure 21 Medical Search Sequence Diagram

Inter Agent Messages

The following inter-agent messages are implemented in XML using the SOAP protocol:

BuyCattle

Request	<pre><buycattle> <purchaseid>integer</purchaseid> <servicerequesterid>integer</servicerequesterid> <numberofcattle>integer</numberofcattle> <typeofcattle>string</typeofcattle> <ageofcattle>integer</ageofcattle> <maxprice>float</maxprice> <creditcardnumber>string</creditcardnumber> <nameoncreditcard>string</nameoncreditcard> <creditcardtype>string</creditcardtype> <expirymonthyear>string</expirymonthyear> </buycattle></pre>
Response	<pre><buycattle> <purchaseid>integer</purchaseid> <servicerequesterid>integer</servicerequesterid> <cattlebought>char</cattlebought> <serviceproviderid>integer</serviceproviderid> <serviceproviderrefid>string</serviceproviderrefid> <numberofcattle>integer</numberofcattle> <typeofcattle>string</typeofcattle> <ageofcattle>integer</ageofcattle> <totalweight>float</totalweight> <price>float</price> </buycattle></pre>

SellCattle

Request	<pre><sellcattle> <salesid>integer</salesid> <servicerequesterid>integer</servicerequesterid> <numberofcattle>integer</numberofcattle> <typeofcattle>string</typeofcattle> <ageofcattle>integer</ageofcattle> <minprice>float</minprice> <creditcardnumber>string</creditcardnumber> <nameoncreditcard>string</nameoncreditcard> <creditcardtype>string</creditcardtype> <expirymonthyear>string</expirymonthyear> </sellcattle></pre>
Response	<pre><sellcattle> <salesid>integer</salesid> <servicerequesterid>integer</servicerequesterid> <cattlesold>char</cattlesold> <serviceproviderid>integer</serviceproviderid> <serviceproviderrefid>string</serviceproviderrefid></pre>

	<pre> <numberofcattle>integer</numberofcattle> <typeofcattle>string</typeofcattle> <ageofcattle>integer</ageofcattle> <totalweight>float</totalweight> <price>float</price> </sellcattle> </pre>
--	--

GetTransportQuote

Request	<pre> <gettransportquote> <transportid>integer</transportid> <servicerequesterid>integer</servicerequesterid> <serviceproviderid>integer</serviceproviderid> <numberofcattle>integer</numberofcattle> <typeofcattle>string</typeofcattle> <ageofcattle>integer</ageofcattle> <totalweight>float</totalweight> <dateofpickup>string</dateofpickup> <lastdeliverydate>string</lastdeliverydate> <addressofpickup>string</addressofpickup> <addressofdelivery>string</addressofdelivery> </gettransportquote> </pre>
Response	<pre> <gettransportqoute> <transportid>integer</purchaseid> <servicerequesterid>integer</servicerequesterid> <transportavailable>char</transportavailabale> <serviceproviderid>integer</serviceproviderid> <serviceproviderrefid>string</serviceproviderrefid> <price>float</price> </gettransportquote> </pre>

ScheduleTransport

Request	<pre> <scheduletransport> <transportid>integer</transportid> <servicerequesterid>integer</servicerequesterid> <serviceproviderid>integer</serviceproviderid> <serviceproviderrefid>string</serviceproviderrefid> <numberofcattle>integer</numberofcattle> <typeofcattle>string</typeofcattle> <ageofcattle>integer</ageofcattle> <totalweight>float</totalweight> <dateofpickup>string</dateofpickup> <lastdeliverydate>string</lastdeliverydate> <addressofpickup>string</addressofpickup> <addressofdelivery>string</addressofdelivery> </scheduletransport> </pre>
Response	<pre> <scheduletransport> <transportid>integer</purchaseid> <servicerequesterid>integer</servicerequesterid> <transportavailable>char</transportavailabale> <serviceproviderid>integer</serviceproviderid> </pre>

	<pre> <serviceproviderrefid>string</serviceproviderrefid> <datetimeofpickup>string</datetimeofpickup> <datetimeofdelivery>string</datetimeofdelivery> <price>float</price> </scheduletransport> </pre>
--	--

GetCustomerPreferences

Request	<pre> <getcustomerpreferences> <servicerequesterid>integer</servicerequesterid> <serviceprovidertype>string</serviceprovidertype> </getcustomerpreferences> </pre>
Response	<pre> <getcustomerpreferences> <servicerequesterid>integer</servicerequesterid> <serviceprovidertype>string</serviceprovidertype> <serviceproviders> <serviceproviderid>integer</serviceproviderid> <pricerating>integer</pricerating> <overallrating>integer</overallrating> <acceptableflag>char</acceptableflag> ... n ... </serviceproviders> </getcustomerpreferences> </pre>

GetMedicalPreferences

Request	<pre> <getmedicalpreferences> <servicerequesterid>integer</servicerequesterid> </getmedicalpreferences> </pre>
Response	<pre> <getmedicalpreferences> <servicerequesterid>integer</servicerequesterid> <keywords> <keyword>string</keyword> <mandatoryflag>char</mandatoryflag> <overallrating>integer</overallrating> ... n ... </keywords> </getmedicalpreferences> </pre>

UpdateMedicalInformation

Message	<pre> <updatemedicalinformation> <servicerequesterid>integer</servicerequesterid> <serviceproviderid>integer</serviceproviderid> <retrievaldatetime>string</retrievaldatetime> <textsubject>string</textsubject> </pre>
---------	---

	<code><textcontent>blob</textcontent></code> <code></updatemedicalinformation></code>
Response	<code><updatemedicalinformation></code> <code><textreceived>char</textreceived></code> <code></updatemedicalinformation></code>

Data Design

ER Diagram

The following figure is the ER diagram for the RAAS system.

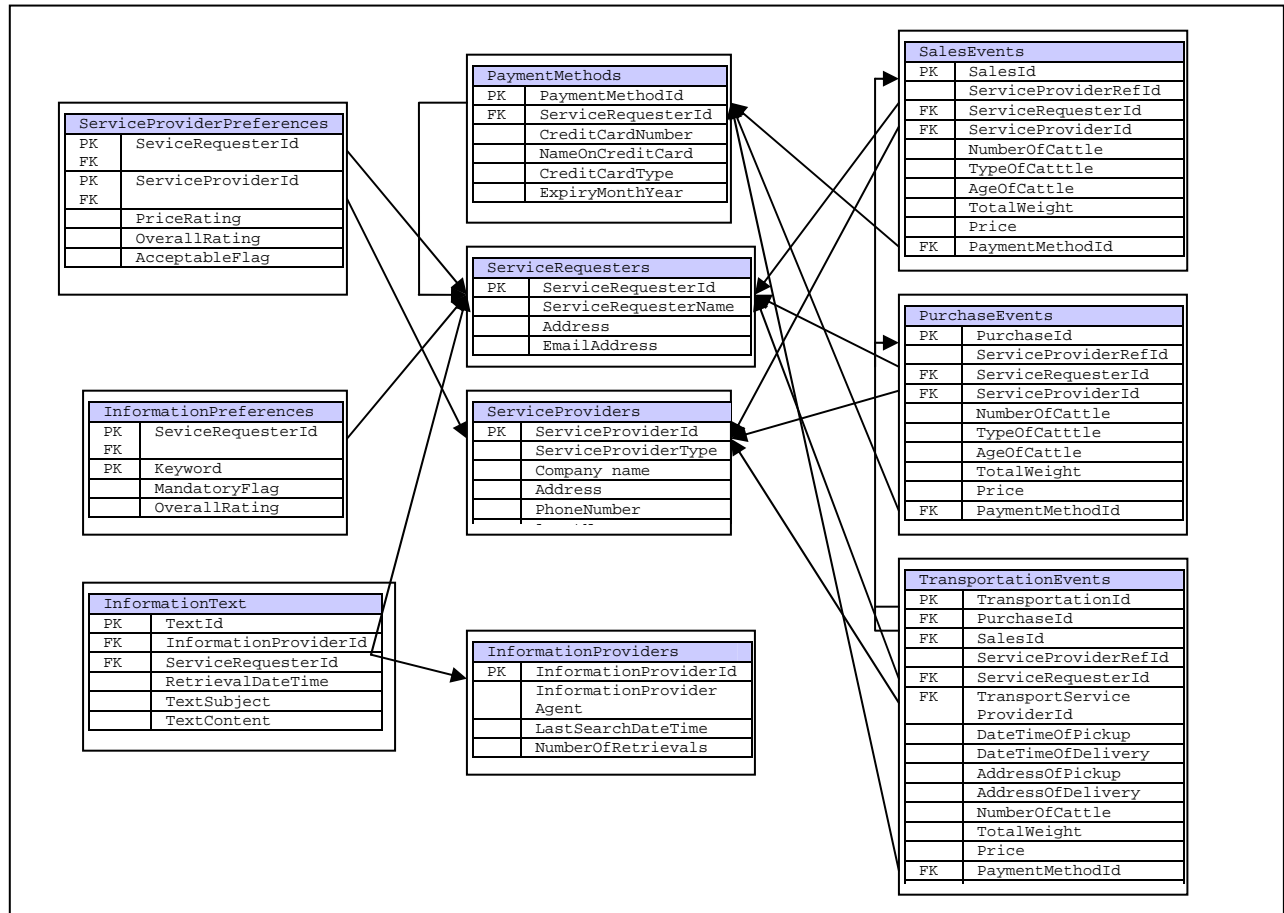


Figure 22 RAAS System ERD

Data Definition

The information on ranchers using the system is stored in the ServiceRequesters table.

Field	Description	Type
ServiceRequesterId	A unique identifier specifying the company using the system.	Integer
ServiceRequesterName	Name of the company using the system.	Varchar(100)
Address	Address of the service requester.	Varchar(200)
EmailAddress	The email address which can be used to contact	Varchar(200)

	the company using the system.	
--	-------------------------------	--

The information on the service requester's credit to make or receive payment is in the PaymentMethods table.

Field	Description	Type
PaymentMethodId	A unique identifier specifying a credit card used to make or receive payment.	Integer
ServiceRequesterId	A unique identifier specifying the service requester making or receiving payment.	Integer
CreditCardNumber	Credit card number	Varchar(20)
NameOnCreditCard	The name of the individual or company which is embossed on the credit card.	Varchar(200)
CreditCardType	Credit card type	Varchar(50)
ExpiryMonthYear	Month and year the credit card expires in MMY format	Char(4)

The information on cattle purchases is stored in the PurchaseEvents table.

Field	Description	Type
PurchaseId	A unique identifier specifying a purchase event.	Integer
ServiceProviderRefId	A reference identifier for the purchase from the Service Provider.	Varchar(100)
ServiceRequesterId	A unique identifier specifying the company initiating the purchase	Integer
ServiceProviderId	An identifier of the auction company.	Integer
NumberOfCattle	Number of cattle purchased.	Integer
TypeOfCattle	The type of the cattle being purchased.	Varchar(50)
AgeOfCattle	The age of the cattle being purchased in months.	Integer
TotalWeight	Total weight of the cattle purchased.	Float
Price	Price of the cattle	Float
PaymentMethodId	Payment method identifier	Integer

The information on cattle sales is stored in the SalesEvents table.

Field	Description	Type
-------	-------------	------

SalesId	A unique identifier specifying a sales event.	Integer
ServiceProviderRefId	A reference identifier for the sales from the Service Provider.	Varchar(100)
ServiceRequesterId	A unique identifier specifying the company initiating the sales	Integer
ServiceProviderId	An identifier of the auction company.	Integer
NumberOfCattle	Number of cattle sold.	Integer
TypeOfCattle	The type of the cattle being sold.	Varchar(50)
AgeOfCattle	The age of the cattle being sold in months.	Integer
TotalWeight	Total weight of the cattle sold.	Float
Price	Price of the cattle	Float
PaymentMethodId	Payment method identifier	Integer

The information on companies which have provided cattle transportation or auction services is stored in the ServiceProviders table.

Field	Description	Type
ServiceProviderId	A unique identifier of the company providing transportation or auction services	Integer
ServiceProviderType	A means of identifying the type of service provider such as auction house or transportation.	Varchar(10)
Company name	The name of the company providing transportation services.	Varchar(100)
Address	The address of the company providing services.	Varchar(200)
PhoneNumber	The contact phone number of the company providing services in (999)999-9999 format.	Char(13)
AgentName	The name of the agent used to interface with the service provider.	Varchar(200)

The information on cattle transportation is stored in the TransportEvents table.

Field	Description	Type
TransportationId	A unique identifier specifying a transport event.	Integer
PurchaseId	An identifier specifying a purchase event.	Integer
SalesId	An identifier specifying	Integer

	a sales event.	
ServiceProviderRefId	A reference identifier for the transport from the Service Provider.	Varchar(100)
ServiceRequesterId	A unique identifier specifying the company requesting transportation	Integer
TransportServiceProviderId	An identifier of the company providing transportation services	Integer
DateTimeOfPickup	Date the cattle will be picked up	Date/Time
DateTimeOfDelivery	Date the cattle will be delivered	Date/Time
AddressOfPickup	Address where the cattle will be picked up.	Varchar(200)
AddressOfDelivery	Address where the cattle will be delivered	Varchar(200)
NumberOfCattle	Number of cattle to be transported.	Integer
TotalWeight	Total weight of the cattle to be delivered.	Float
Price	Price for transportation	Float
PaymentMethodId	Payment method identifier	Integer

The medical information sources are stored in the InformationProvider table

Field	Description	Type
InformationProviderId	An internal identifier of the source agent which provided the information	Integer
InformationProviderAgent	An means of identifying the Information Provider Agent on the Web	Varchar(200)
LastSearchDateTime	A date/timestamp specifying when the last time this Information Provider was queried.	Date/Time
NumberOfRetrievals	The number of articles which have been retrieved from the Information Provider	Integer

The medical information is stored in the InformationText table.

Field	Description	Type
TextId	A unique identifier.	Integer
InformationProviderId	An identifier of the source agent which provided the information	Integer
ServiceRequesterId	A unique identifier specifying the company requesting transportation	Integer
RetrievalDateTime	A date/timestamp	Date/Time

	specifying when the information was retrieved from the source	
TextSubject	A short subject line describing the content of the text	Varchar(200)
TextContent	Content of the information text.	Blob

The service requester medical information preferences are stored in the InformationPreferences table.

Field	Description	Type
SeviceRequesterId	An identifier of the service requester,	Integer
Keyword	A keyword which should be found in the author, title, subject, list of keywords, publisher, or abstract of medical text searched.	Varchar(200)
MandatoryFlag	A Y or N flag indicating whether or not it is mandatory that this keyword appear in the author, title, subject, list of keywords, publisher, or abstract of medical text searched.	Char(1)
OverallRating	An overall rating of how important it is that this keyword appear in the author, title, subject, list of keywords, publisher, or abstract of medical text searched. The rating is out of 10 where 0 is low and 10 is high.	Integer

The service requester preference information on service providers is stored in the ServiceProviderPreferences table.

Field	Description	Type
SeviceRequesterId	An identifier of the service requester,	Integer
ServiceProviderId	An identifier of the service provider.	Integer
PriceRating	A rating of this service providers prices in general to other service providers of a similar type. The rating is out of 10 where 0 is low and 10 is high.	Integer
OverallRating	An overall satisfaction	Integer

	rating out of 10 where 0 is low and 10 is high.	
AcceptableFlag	A Y or N flag indicating whether the service requester is willing to do business with the service provider.	Char(1)

Desirable Future Enhancements

This document does not design for the following suggested enhancements though they should be considered in future releases:

- The ability of ranchers to keep details of the cattle stock in the system which would reduce the possibility of data input error when entering requests for to buy, sell, or transport cattle.
- The ability to bill users for the use of this system
- The ability for ranchers to schedule transportation independent of sales or purchases.
- The ability of the system to search on legal and economic information regarding the ranching business.

Bibliography

1. http://collections.ic.gc.ca/pasttopresent/rural_life/ranching_economy.html
2. Bird, Michael, "Strong Economics"
<http://www.albertabeef.org/rancHER/pdf/economics.pdf>
3. Swigert, Steve, "Ranching Information on the Web" February 2002
<http://www.noble.org/ag/Economics/RanchingInfoWeb/>
4. Box, Don et al., World Wide Web Consortium, "Simple Object Access Protocol (SOAP) 1.1", May 2000, <http://www.w3.org/TR/SOAP/>
5. Christensen, Erik, et al., World Wide Web Consortium, "Web Services Description Language (WSDL) 1.1", March 2001, <http://www.w3.org/TR/wsd1>