

6. Behavioral Modeling-Use case Diagram

Use case diagrams model the behavior of a system and help to capture the requirements of the system. Use case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

6.1 Use case diagram for Airport check-in and security screening business model

Design and development a use case diagram for Airport check-in and security screening business model

Hints

```
/**
 * Design the Static Deployment view with Use case diagram for Airport check-in and
 security screening
 */To design a Use Case Diagram:
Select first an element where a new Use Case Diagram to be contained as a child.
Select Model | Add Diagram | Use Case Diagram in Menu Bar or select Add Diagram | Use
Case Diagram in Context Menu.
To design a Use Case Subject:
Select Use Case Subject in Toolbox.
Drag on the diagram as the size of Use Case Subject.
To design an Actor:
Select Actor in Toolbox.
Drag on the diagram as the size of Actor.
To design an Actor (model element only) by Menu:
Select an Element where a new Actor to be contained.
Select Model | Add | Actor in Menu Bar or Add | Actor in Context Menu.
Name Expression: Edit name expression.
Syntax of Name Expression
expression ::= [ '<<' stereotype '>>' ] [ visibility ] name
stereotype ::= (identifier)
visibility ::= '+' | '#' | '-' | '~'
name ::= (identifier)
To design an Include:
Select Include in Toolbox.
Drag from a Use Case and drop on another Use Case (to be included).
To design an Extend:
Select Extend in Toolbox.
Drag from a Use Case (to be extended) and drop on another Use Case. Business actors
are Passenger, Tour Guide, Minor (Child), Passenger with Special Needs (e.g.
with disabilities), all playing external roles in relation to airport business.

Business use cases are Individual Check-In, Group Check-In (for groups of tourists),
Security Screening, etc. - representing business functions or processes taking place
in airport and serving the needs of passengers.

Business use cases Baggage Check-in and Baggage Handling extend Check-In use cases,
because passenger might have no luggage, so baggage check-in and handling are optional.
```

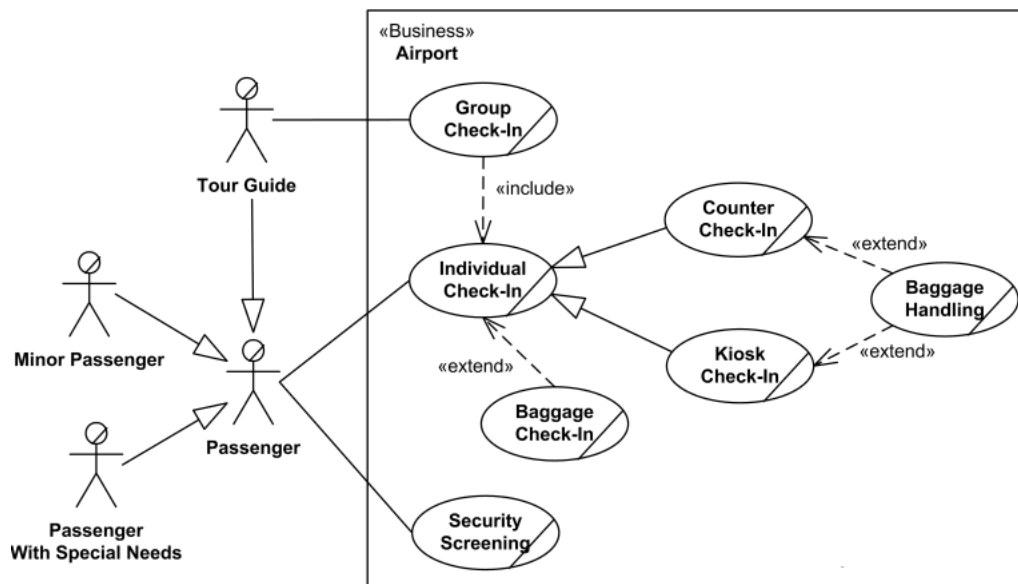


Fig.6.1 Use case diagram for Airport check-in and security screening business model

6.2 Use case diagram for Transportation and Logistics

Design and development a use case diagram for Ticket Vending Machine. Ticket vending machine, i.e. vending machine that sells and produces tickets to commuters, is a subject of the example use case diagram. This kind of a machine is a combination of both hardware and software, and it is only a part of the whole system selling tickets to the customers. So, we will use «Subsystem» stereotype.

Hints

/**

* Design the usecase view with Use case diagram for *Transportation and Logistics* */

Ticket vending machine provides Purchase Ticket use case for the Commuter and Bank actors.

The ultimate goal of the Commuter in relation to our ticket vending machine is to buy a ticket. So, there is Purchase Ticket use case. Purchasing ticket might involve a bank, if payment is to be made using a debit or credit card. So we are also adding another actor - Bank. Both actors.

participating in the use case are connected to the use case by association. Use case behaviors may be described in a natural language text (opaque behavior), which is current common practice, or by using UML behavior diagrams. UML tools should allow binding behaviors to the described use cases. Example of such binding of the Purchase Ticket usecase to the behavior represented by activity

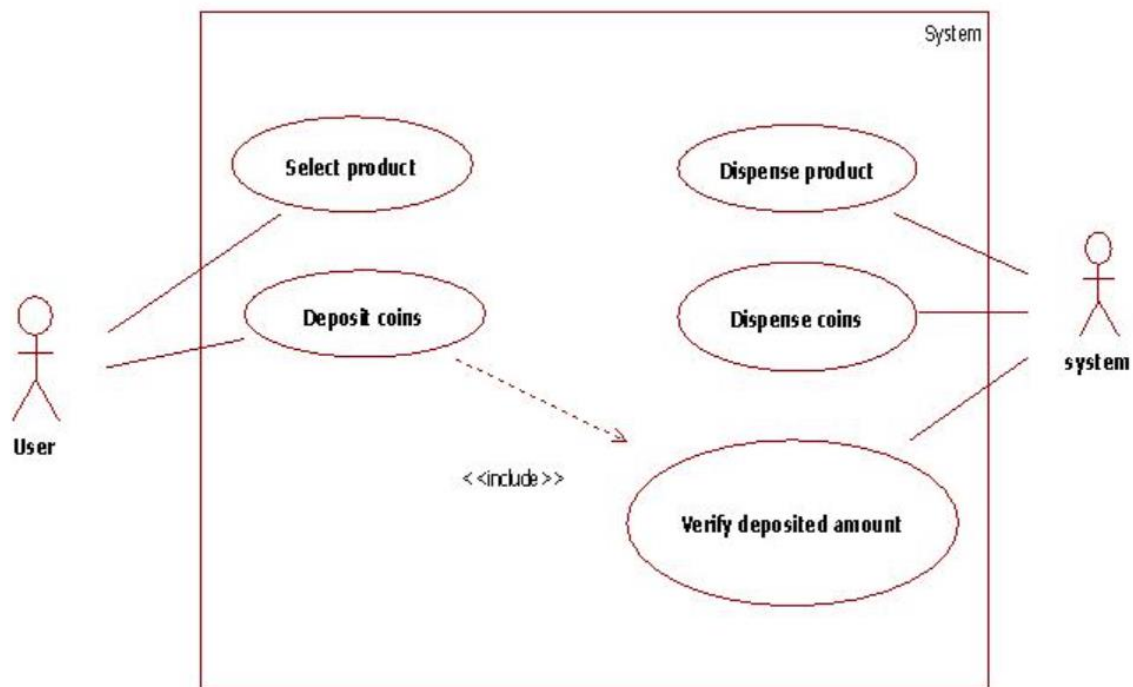


Fig.6.2 Use case diagram for Transportation and Logistics

6.3 Use case diagram for Credit Card Processing System

Design and development a use case diagram for Ticket Vending Machine. Credit Card Processing System (aka Credit Card Payment Gateway) is a subject, i.e. system under design or consideration. Primary actor for the system is a Merchant's Credit Card Processing System. The merchant submits some credit card transaction request to the credit card payment gateway on behalf of a customer. Bank which issued customer's credit card is actor which could approve or reject the transaction. If transaction is approved, funds will be transferred to merchant's bank account.

Hints

```

/**
 * Design the use case view with Use case diagram for Credit card Processing System*/

```

Authorize and Capture use case is the most common type of credit card transaction. The requested amount of money should be first authorized by Customer's Credit Card Bank, and if approved, is further submitted for settlement. During the settlement funds approved for the credit card transaction are deposited into the Merchant's Bank account.

In some cases, only authorization is requested and the transaction will not be sent for settlement. In this case, usually if no further action is taken within some number of days, the authorization expires. Merchants can submit this request if they want to verify the availability of funds on the customer's credit card, if item is not currently in stock, or if merchant wants to review orders before shipping.

Capture (request to capture funds that were previously authorized) use case describes several scenarios when merchant needs to complete some previously authorized

transaction - either submitted through the payment gateway or requested without using the system, e.g using voice authorization.

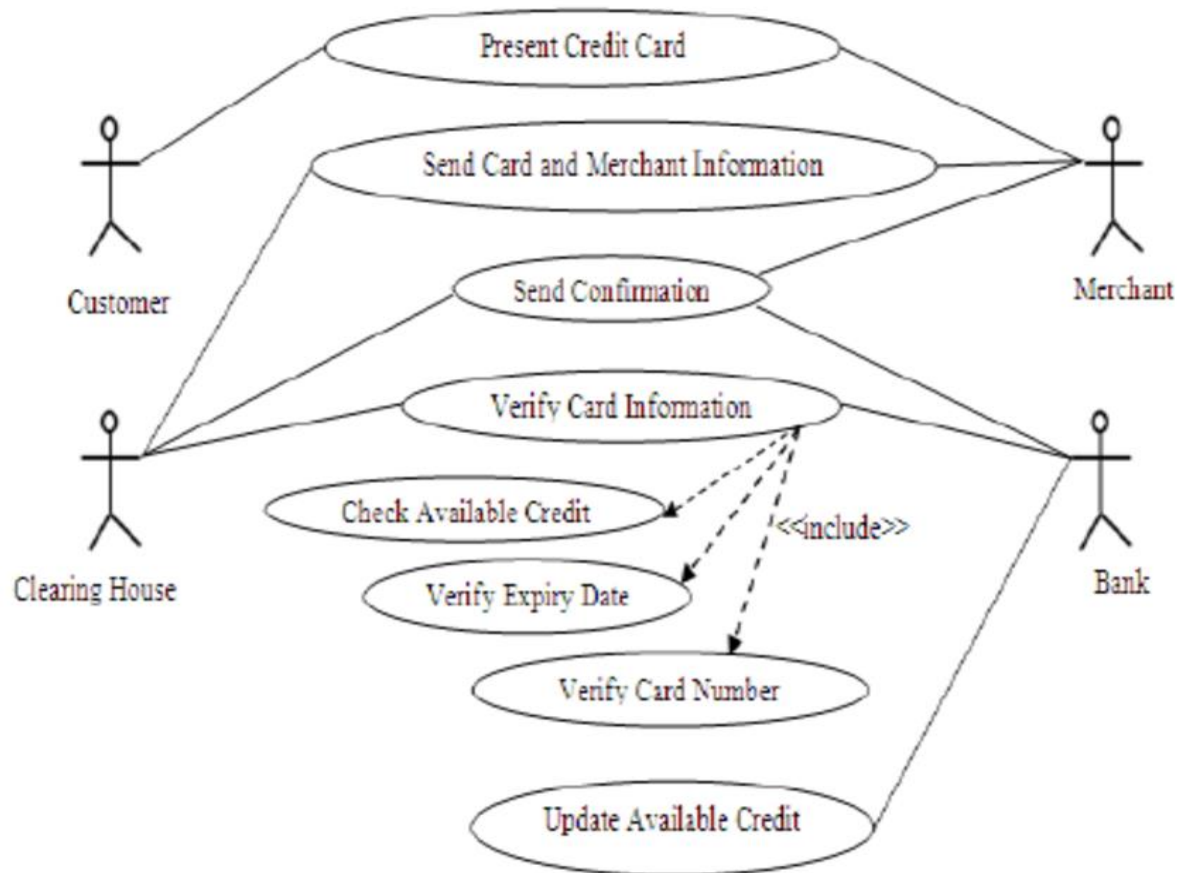


Fig.6.3 Use case diagram for Credit Card Processing System

6.4 Use case diagram for Radiology Diagnostic Reporting

Design and development a use case diagram for Radiology Diagnostic Reporting. The Simple Image and Numeric Report (SINR) [IHE Radiology Integration Profile, IHE RAD TF Vol. 1, Rev. 11.0] facilitates the growing use of digital dictation, voice recognition, and specialized reporting packages, by separating the functions of diagnostic reporting into discrete actors for creation, management, storage and report viewing. Separating these functions while defining transactions to exchange the reports between them enables a vendor to include one or more of these functions in an actual system. The IHE Technical Framework (TF) identifies IHE Actors - functional components of a healthcare enterprise from the point of view of their interactions in distributed healthcare environment.

Hints

```
/**  
 * Design the usecase view with Use case diagram for Radiology Diagnostic Reporting  
 System*/
```

Radiology diagnostic reporting UML use case diagram example for Simple Image and Numeric Report (SINR) IHE Radiology Integration Profile

The Simple Image Report allows documents with multiple sections (with headings) containing report text and references to relevant images. Some text items of these documents may also be related to specific images. This allows a reading physician to identify one or more images from which their conclusions were inferred.

Reports are processed and modified by the Report Manager IHE actor. This involves adding and changing report data as well as verifying draft reports

In the Report Issuing transaction, the Report Manager transmits either an unchanged draft DICOM SR or a new modified DICOM SR to the Report Repository.

The Report Repository provides permanent storage of DICOM Structured Reports. It also allows reports to be queried and retrieved throughout the enterprise by Report Readers.

The External Report Repository Access actor is a gateway to obtain other enterprise department reports, such as Laboratory and Pathology, from within the Imaging department.

In the Structured Report Export [RAD-28] transaction, the Report Manager transmits verified Structured Reports as unsolicited HL7 observations to the Enterprise Report Repository

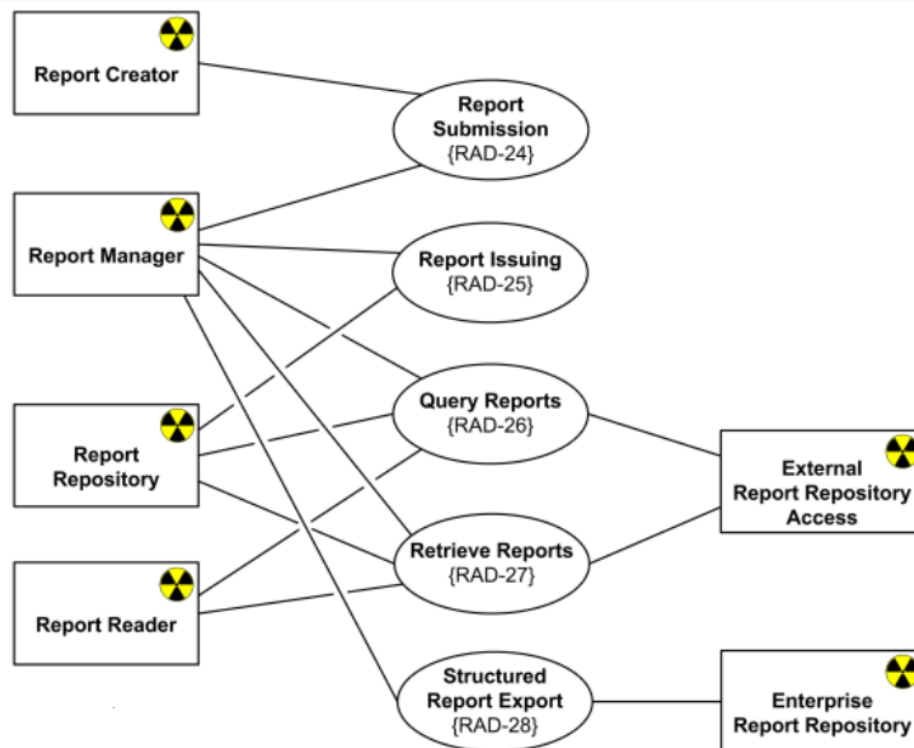


Fig.6.4 Use case diagram for Radiology Diagnostic Reporting

6.5 Use case diagram for Software Protection and Licensing

Design and development a use case diagram for Software Protection and Licensing. Sentinel License Development Kit (Sentinel LDK) is a Software Digital Rights Management (DRM) solution by SafeNet Inc. that delivers strong copy protection, protection for Intellectual Property (IP), and secure and flexible licensing. Sentinel LDK separates licensing and production processes (implemented with Sentinel EMS) from the software protection process (implemented with Sentinel Licensing API or Sentinel LDK Envelope). Sentinel EMS is a web-based graphical application provided as part of Sentinel LDK that is used to perform a range of functions required to manage the licensing, production, distribution, customer support, and maintenance of protected applications. This application is a role-based application designed to manage the business activities required to implement and maintain Sentinel LDK in the organization which needs to protect its software. Sentinel EMS Server maintains a database containing a wide range of information, including data related to product features, licenses, sales, orders, and customers.

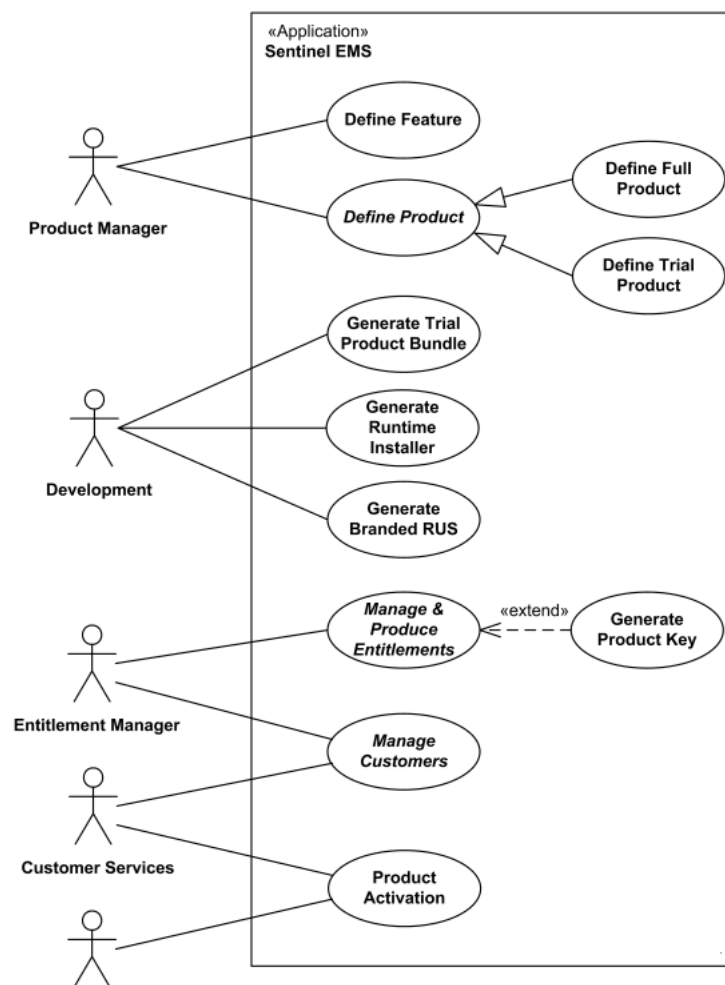


Fig.6.5 Use case diagram for Software Protection and Licensing