Bucks Centre for Performing Arts:  
Online Ticketing System

By Thomas Golbourn, Student ID: 21614843.  
Harvey Rush, Student ID: 21518237

# Requirement Gathering

## Introduction

The Bucks Centre for Performing Arts would like to enable their customers to order tickets through the internet. They have presented us with a document outlining the system they want. This document contains all the known requirements of the system from a functional perspective. Some of these functions lend themselves to non-functions as well. Here we will be outlining the requirements presented in the document given, both functional and non-functional, so that we can begin analysing them.

## Capturing Requirements

In this case, most of the requirement capture has been done for us, as we have been given a very specific outline of how the software should operate in a document. However, where there are gaps, or ambiguity in the document, requirement capture techniques can be used. These techniques can be broken down into 5 categories; interview, observation, document sampling, background reading, and questionnaire.

### Background Reading

In this project background reading is not entirely needed if the project is built to be scalable to suit a large theatre organisation with multiple venues, shows, and thousands of users. However, if building it for a smaller scale, then background reading would need to be undertaken to gauge how small a scale might be acceptable. We can also use background reading as a technique to learn about existing online ticket systems for theatres. This can help us avoid obstacles others have faced in the past. A major concern about background reading is that sometimes relevant or up to date information can be difficult to find, and can even have red herrings within them. In any case due to the unfamiliarity with the organisation background reading should take place to begin to understand the fact-finding questions we’ll need to ask, and to understand the business we’ll be working with.

#### Results of Background Reading

The first part of the background reading was discovering any online presence, and how they currently book tickets through that. Their website is <http://bucksperformingarts.blogspot.co.uk/> and is quite simplistic. From this site we can see their most recent show used the Courtyard Theatre in London, but they have used Wycombe Town Hall, and multiple other venues in the past. This means we will absolutely have to accommodate multiple venues in the software. Their most recent method of ticket booking was via the Courtyard Theatre’s use of TicketWeb, with an alternative to phone and book tickets directly with the theatre. However, they have used other theatre’s own payment methods, such as Norden Farm theatre’s website, in the past. This could be the reason why they need their own system, as the inconsistent booking methods are not working for them.

### Interviewing

In this project interviewing is a readily available tool, as there are regular meetings with the client, who is also well-versed in computer science and programming. With just one point of contact there may be bias, but at least there shouldn’t be any conflicting accounts. Under normal circumstances it would be costly and time-consuming to conduct the interviews and transcribe the audio or compile the notes in a meaningful and useful way. However due to the regularity of the meetings, the knowledge of the client, and the university funding them, these disadvantages do not apply quite as much. Due to this interviewing is one of the most powerful tools at our disposal in this project.

#### Interview Questions & Answers

Interview questions are shown in bold italics as headings with the paraphrased and reformatted answer indented afterwards. These were paraphrased and written down on-the-fly and agreed upon with the client during the interview as acceptable answers to the questions.

##### The document isn’t clear on which ‘customer information’ an agent must provide to purchase a ticket on a customer’s behalf, could you clarify which information should be given and how consent should be handled?

Consent from the user will be asked for when signing up, prior to entering credit card or personal information. This consent will explicitly state that agents may book on their behalf at their request whenever the requested tickets become available. Furthermore, it should state that the payment will be automated and can be subject to a processing time leading up to and including within 24 hours of the event, starting from the payment date. An agent needs to be able to find the customer in the system via any identifier or search term, such as the customer’s name or address. A unique identifier such as an email address, customer id number, or username, should also be available to use. Once an agent has identified the customer they should be able to make the purchase for them without knowing any further details about the customer. The customer should then be notified of this via email.

##### The document has no mention of target operating systems or hardware, do you have a preference? Are there hardware or software considerations on the server end we need to account for, or systems in place we need to integrate with?

The target operating system for the customer, venue manager, and agent is Windows 7, Windows 8, and Windows 10. The server should be hosted on a Windows Server 2012 machine. Payment systems will be set-up via a phoneline connected specialist machine. This machine has a local address on the network with the server and takes inputs at a rate of up to 10,000 per second in the following format:   
 [Unique ID (Int), CardNumber (Int), CVC Code (Int), Expiry (Int (MMYY), Charge (Float (####.##))]  
These inputs simply need to be parsed to a plain text file, line by line, on the server machine, in a windows network accessible location. The specialist card machine will read the inputs from there once it has been configured by the BCPA staff. Once it is running it will generate a file named ‘output.txt’ and a file named ‘log.txt’ in the same directory as the input file. Within the output file the results of operations parsed to input will be shown in the following format: [Unique ID (Int), DateTime (DateTime), Status (String)]  
The log file will show these inputs and outputs in this format: [[InputData], [OutputData]]  
The email system should use a text file in this same location called ‘email.txt’. This should be fed data line by line, after each line is read it will be deleted, the format specified is:  
[Email Address (Str), First Name (Str), Last Name (Str), Ticket Info (Any Multi-Line String)), MessageTemplateName (Str (Available Types: ‘PurchaseConfirmed’, ‘PurchaseProcessed’))]

##### Roughly how many employees does the BCPA have, and expect to have during the software’s expected lifecycle?

Less than 50 on both accounts.

##### Roughly what peak number of concurrent users do you think the software see during it’s expected lifecycle?

Between 100 and 600 concurrent users.

##### Who will be the Data Protection Officer should we need one, and will they ensure and finance the projects adherence to all current and future data privacy and e-commerce laws?

Yes, we will cover the cost of adherence to all data privacy and e-commerce law during the software’s development and lifecycle, and I am already the appointed Data Protection Officer for this software.

#### Interview Conclusions

The key take-aways are: Mail and credit card charging systems are easy to use and access. Data privacy and e-commerce law adherence will be overseen by the client. The software should handle over 600 users at once.

## Non-Functional Requirements

Non-functional requirements are requirements that are concerned with system design rather than use. For example, the use of the ticket system is to enable customer purchases, however, the design of the system also determines several key parts of the program. Here I have created a list of non-functional requirements that should be considered, with the reasons why.

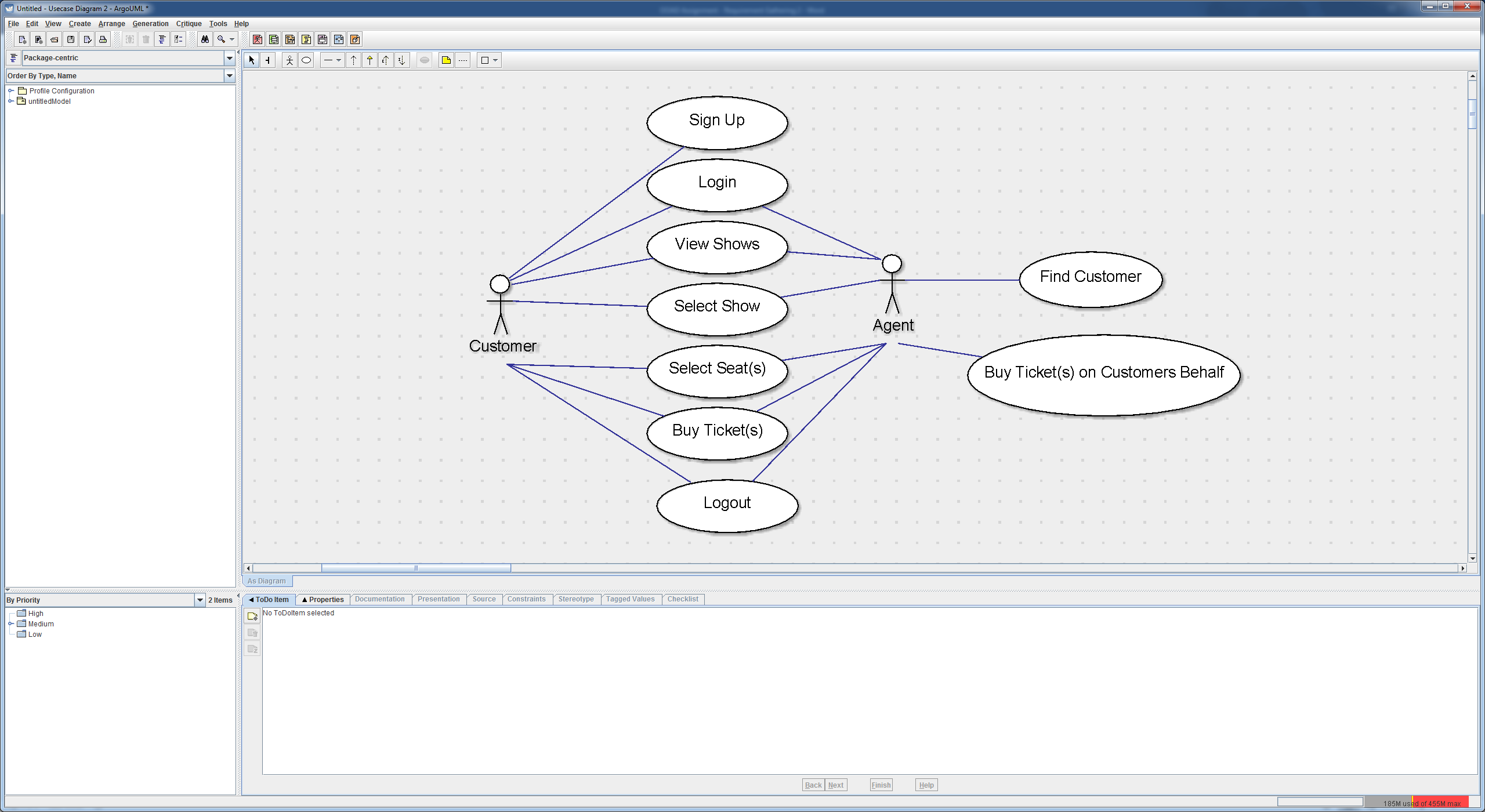
1. **Security**: Since the program requires users to create accounts with personal and financial details, and since agents and venue managers will be able to access these details, security measures are needed. If an unauthorised user managed to access one customer, or several customer accounts via an agent or venue manager account, the result would be a disaster. Customers would no longer trust the software with their financial or personal details.
2. **Quality**: If the program is of low quality, then users may lament having to use the program and call for its replacement. Customers especially may not trust the program with their information if it appears the program is of inadequate quality.
3. **Usability:** If the software is difficult to use for any user, they will be dissatisfied with it. In the venue manager’s case, where there are a lot of ways to configure a show, usability should be considered strongly. If it is not considered, then the program could easily end up feeling ‘bloated’, inundated with buttons and text, and hard to understand.
4. **Privacy:** Since Britain is still currently operating under the European Union law, the European Union’s laws must be considered. Some of these laws pertain to data privacy, control, and processing, this concerns how data can be collected, stored, used, and transported. Compliance with such laws, directives, and regulations, should be ensured during the design, build, and use of the software. Britain also uses its own laws, or clauses within EU laws, pertaining to data privacy, storage, transportation, and processing, that must be considered alongside EU law. In May of 2018 new EU legislation is coming into practice, this will also need to be considered and adhered to, along with any laws introduced by Britain following Brexit.
5. **Accountability:** Since the requirements dictate that the agent is allowed to make purchases on a customer’s behalf, actions taken within the software should trigger mechanisms that record or report such actions. This is to make the software safe to use in an online environment. If a security breach, error, or user mistake occurred and there was no accountability then discovering where, when, and how it occurred could be difficult.
6. **Configurability:** Since the software outline document does not include any seating layouts or price structures to work with, and theatres can vary these between shows, we need to ensure that the software is configurable to meet the theatre’s unique properties. These configurations will not be set-up by a software engineer, but instead by the venue manager. As such the software needs to be easily configurable on the user-level.
7. **Reliability:** If the software is unreliable, a multitude of problems can occur. Tickets may not go on sale at the right time, or the right price, seats may be oversold, users may not be able to access or use parts of or the entire system. These will hurt the business and the consumers trust in the business. This will have a ripple effect, and our software solutions may no longer be as trusted by other businesses, which could lose us clients. As such reliability is incredibly important. If a lift breaks half the times it is used, the stairs will be taken instead, regardless of how good the lift is.

## Requirements Model

### First Iteration, Basic Use Cases

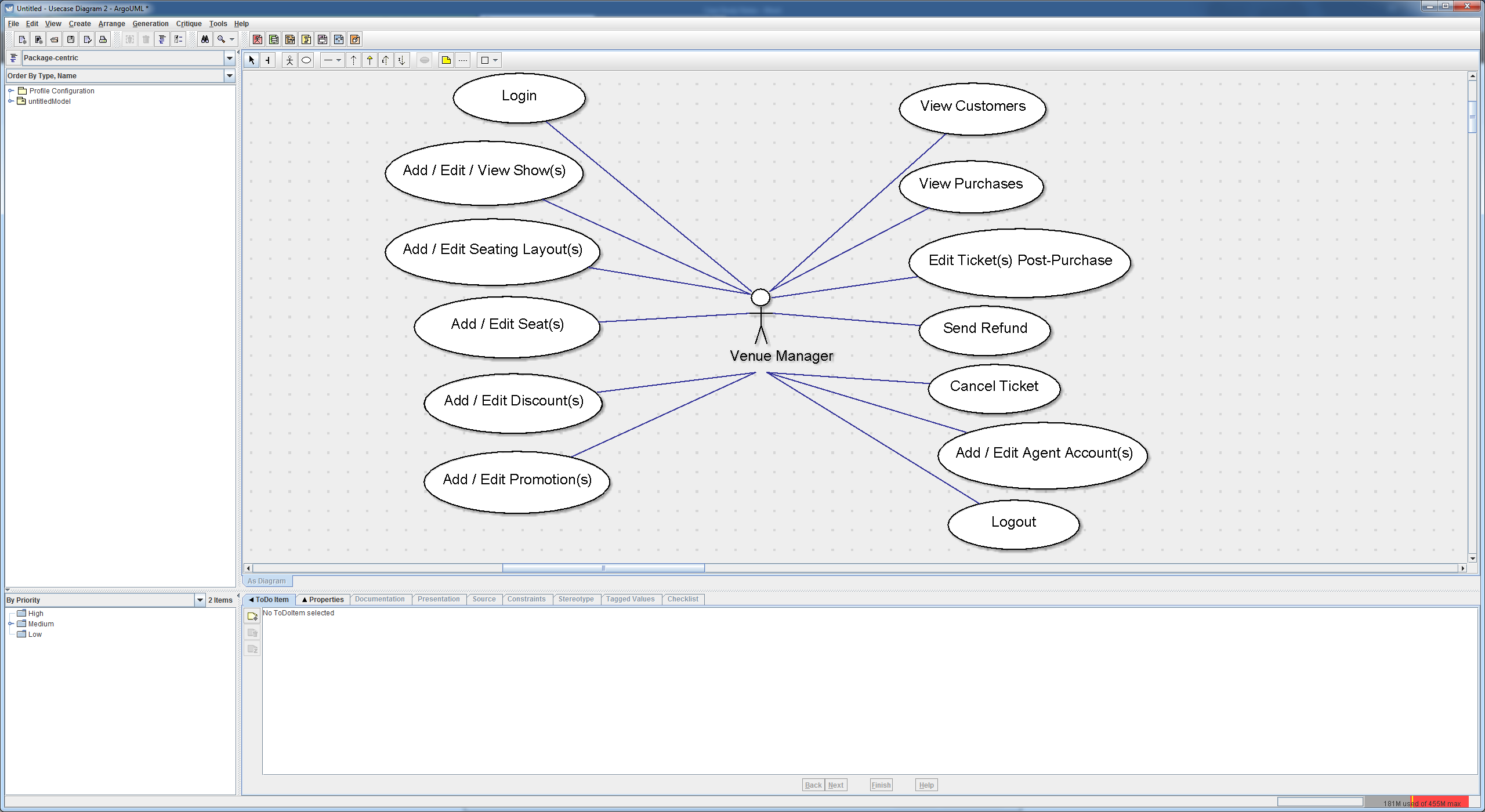
|  |  |
| --- | --- |
| Actors | Descriptions |
| Venue Manager | The venue manager is responsible for the creation and editing of shows, including their prices, seating layouts, promotions, and discounts. They are also responsible for processing any refunds or amending tickets, as well as management of agent accounts. |
| Agent | The agent is responsible for booking tickets on behalf of customers. |
| Customer | The customer is able to access shows, select seats, and purchase tickets. |

#### Customer and Agent



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Use Cases | Descriptions | | Sign Up | To allow the customer to execute any purchase requests they must first create an account via the sign up button, entering the critical information that will be needed to later complete a purchase. | | Login | Once a user has an account they can log in with their sign up email address and password, allowing them to access and edit account information and view order history. | | View Shows | The customer can view available shows via one of the search options either by show name, date or a list of all current shows. | | Select Show | The customer can then select their desired show to begin viewing extended show details and ticket information. | | Select Seat(s) | A seating plan for the venue will be shown, displaying all seats with information on availability and price ranges, they can then be selected by the customer to create a purchase request to be executed by the agent. | | Buy Ticket(s) | The customer must then enter their card details or select saved card details to be passed on to execute the order once available. | | Logout | The customer is then free to log out and their part of the purchase process is now complete. | | Find Customer | The agent will now locate and load customer details for his allocated list of purchases. | | Buy Ticket(s) on Customers Behalf | Once the customer is loaded and tickets are ready to be purchased, the agent will complete the purchase on the customer’s behalf and the customer will be automatically notified. | |  |

#### Venue Manager



|  |  |
| --- | --- |
| Use Cases | Descriptions |
| Login | The venue manager will log in with his email and password, which are allocated special privileges and a separate interface from agents for management uses. |
| Add / Edit / View Show(s) | In the venue managers interface he will have access to creating new shows, editing existing shows or simply viewing shows the way an agent would.  Creating a new show requires entering details such as, show title; date; time; show description; ticket cost. |
| Add / Edit Seating Layout(s) | Once a show is created the venue manager can begin creating, editing or loading a seating layout to be applied to the show, the seating layout will reflect the venue layout and dictates the maximum seats per customer. |
| Add / Edit Seat(s) | Inside the seating layout, individual seats can then be added which hold the following information, seat price; seat name; seat ID; seat reserve timer; promotion ID; agent ID; seat booking status and seat position. |
| Add/ Edit Discount(s) | Discounts are added when certain criteria is met however can also be applied to the desired show as a whole.  Criteria can include, ticket type or volume of tickets purchased. |
| Add / Edit Promotion(s) | Promotions are controlled at two levels, show and seat. A show promotion is applied to all seats and can be saved, loaded and reused across multiple shows.  A seat promotion can be added at seat level to individual seats for desired reasons, e.g. clearance of sections. |
| View Customers | The venue manager can view customer details by searching from a list of available criteria such as, name; address; phone number or email address. |
| View Purchases | The venue manager can view purchase details either from customer purchase records when viewing a customer or via a provided purchase ID. |
| Edit Ticket(s) Post-Purchase | A ticket can be edited post purchase to reallocate customer seat location in certain circumstances. |
| Send Refund | If required, when viewing purchases the venue manager can issue a refund to the customer without affecting the customers’ ability to use the ticket or cancelling the seat. |
| Cancel Tickets | Should the customer express that they can no longer attend the event, the venue manager can choose to refund and cancel the ticket, reallocating it for purchase again. |
| Add / Edit Agent Account(s) | Agents must be given permissions by the venue manager once they have created their account through the standard email registration system. |
| Logout | The venue manager can log out of his account when desired. |

### First Iteration, Detailed Use Cases

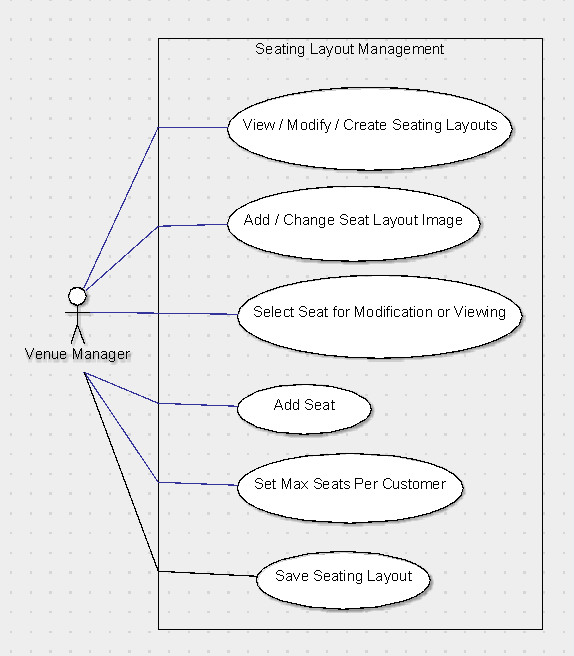
The following use cases are ‘zoomed in’ looks at the basic use cases. They are sorted by the actor that they apply to.

#### Venue Manager

The following use cases apply to the venue manager actor only.

##### Seating Layout Management

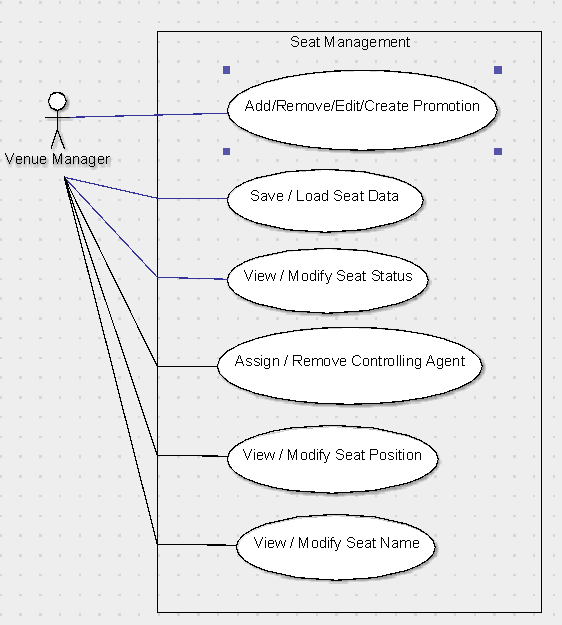
The seating layout management dialog has two parts, one is the select / create / delete dialog box, which is shown prior to the main interface which enables you to edit a seating layout. The select / create / delete dialog shows only a list box with 3 buttons: Create Layout, Modify Layout, and Delete Layout. The selection is done via a list box / drop down box that shows all existing seating layouts, sorted by ID, shown with both ID and name.



|  |  |
| --- | --- |
| Use Cases | Descriptions |
| View / Modify / Create Seating Layouts | This shows a dialog with a list box that contains all saved seating layouts, sorted by ID number, with the text showing the ID number and the layout name. To select a seating layout from here the user simply presses ‘Edit Seating Layout’ with the desired seating layout to edit selected in the list box. There will also be a button in this dialog that allows the user to create a new seating layout, and a button that enables the user to delete the selected seating layout. |
| Add / Change Seat Layout Image | This shows an open file dialog box that lets the user select a seating layout image from a location on the local computer. Once an image is selected the user must click a confirmation button and then the seating layout image will be applied to the currently selected seating layout. |
| Select Seat for Modification or Viewing | In the seating layout seats appear as objects that can be clicked, if the user selects a seat they will be given with options to change and view its properties, such as position, name, promotion ID, or controlling agent ID. |
| Add Seat | When the user clicks the seating layout in an empty space, a new seat will be created, a dialog will show allowing the manager to input the seats details. This is the same dialog that shows when modifying a seat. |
| Set Max Seats per Customer | An integral part of all seating layouts is that there is a maximum amount of seats each customer can purchase. This must be set for each seating layout. This is done via the seating layout interface with a small labelled text box. |
| Save Seating Layout | This option is a simple button in the main seating layout interface that saves the seating layout that is currently open. |

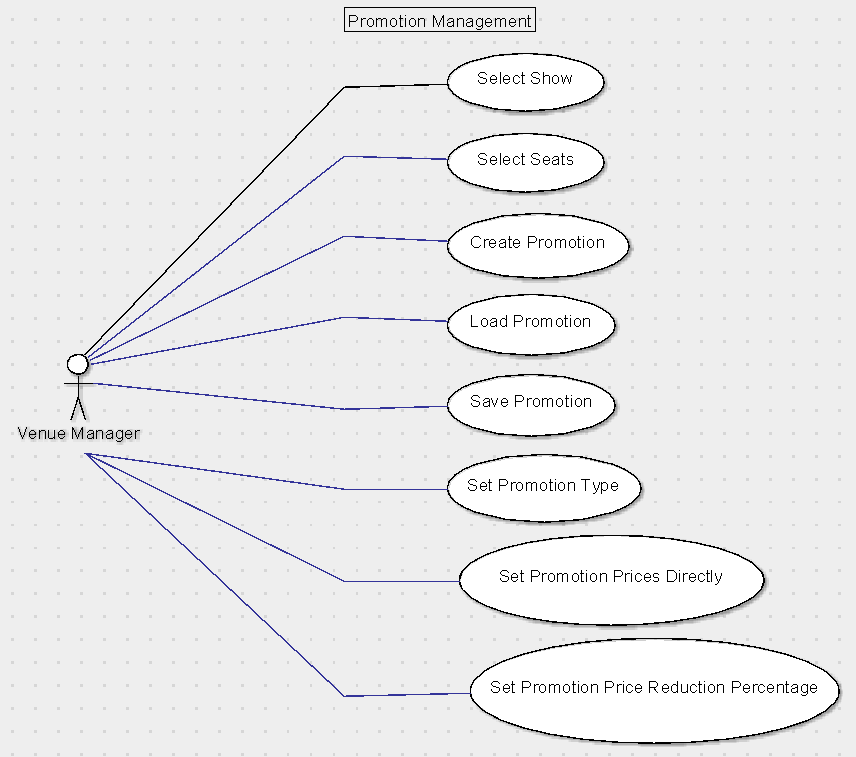
##### Seat Management

Seat management is done via the seat management dialog as well as the seating layout dialog. Within the seating layouts seats can be selected, created, deleted, and moved. Within the seat management dialog seats can be renamed, moved, assigned promotions or controlling agents, have their status changed, and have their data saved or loaded.

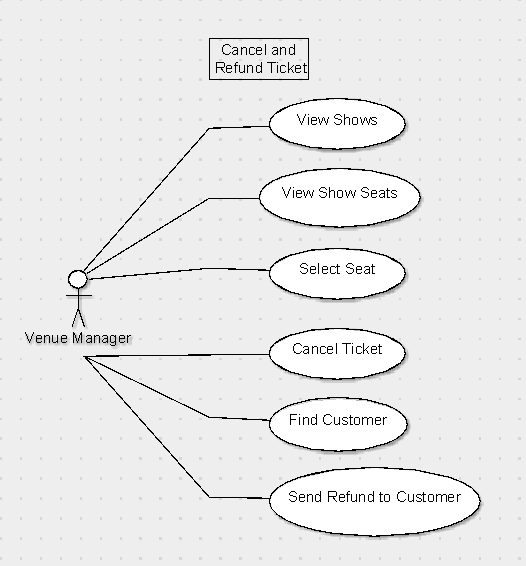


|  |  |
| --- | --- |
| Use Cases | Descriptions |
| Add / Remove / Create Promotion | This option opens the promotion selection dialog, which also includes buttons to add a selected promotion to the seat, create a new promotion, delete an existing promotion, and modify an existing promotion. |
| Save / Load Seat Data | This shows a dialog which enables the user to select from a list of existing saved seat data, and also enables the user to save the current seat data with a new name into the list, or delete a selected option from the list. |
| Assign / Remove Controlling Agent | This lets the user select from a list of agents, in that list is the ‘No Agent’ option as well. This is shown through a list box. |
| View / Modify Seat Status | This lets the user select from a drop down list of available status types, these are ‘Available’, ‘Held’, and ‘Booked’. |
| View / Modify Seat Position | This can be done in one of two ways, either by selecting and then dragging the seat, or multiple seats, across the seating layout to the new position, or by opening the seat management interface where the seat position can be set manually in pixel X and Y position. |
| View / Modify Seat Name | This can only be done through the seat management dialog which is opened by double clicking on a seat. Inside the dialog box there is a text box which shows the current seat name, and there is also an apply button to save the changes, and a cancel button to ignore the changes. |

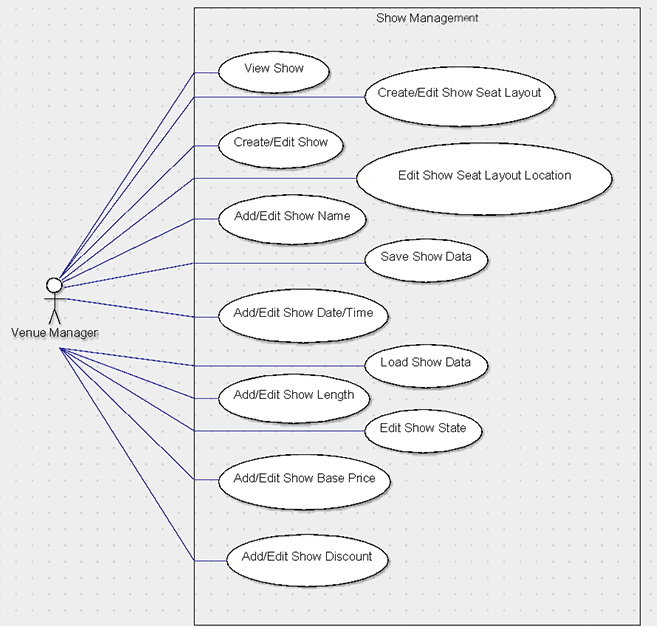
##### Promotion Management



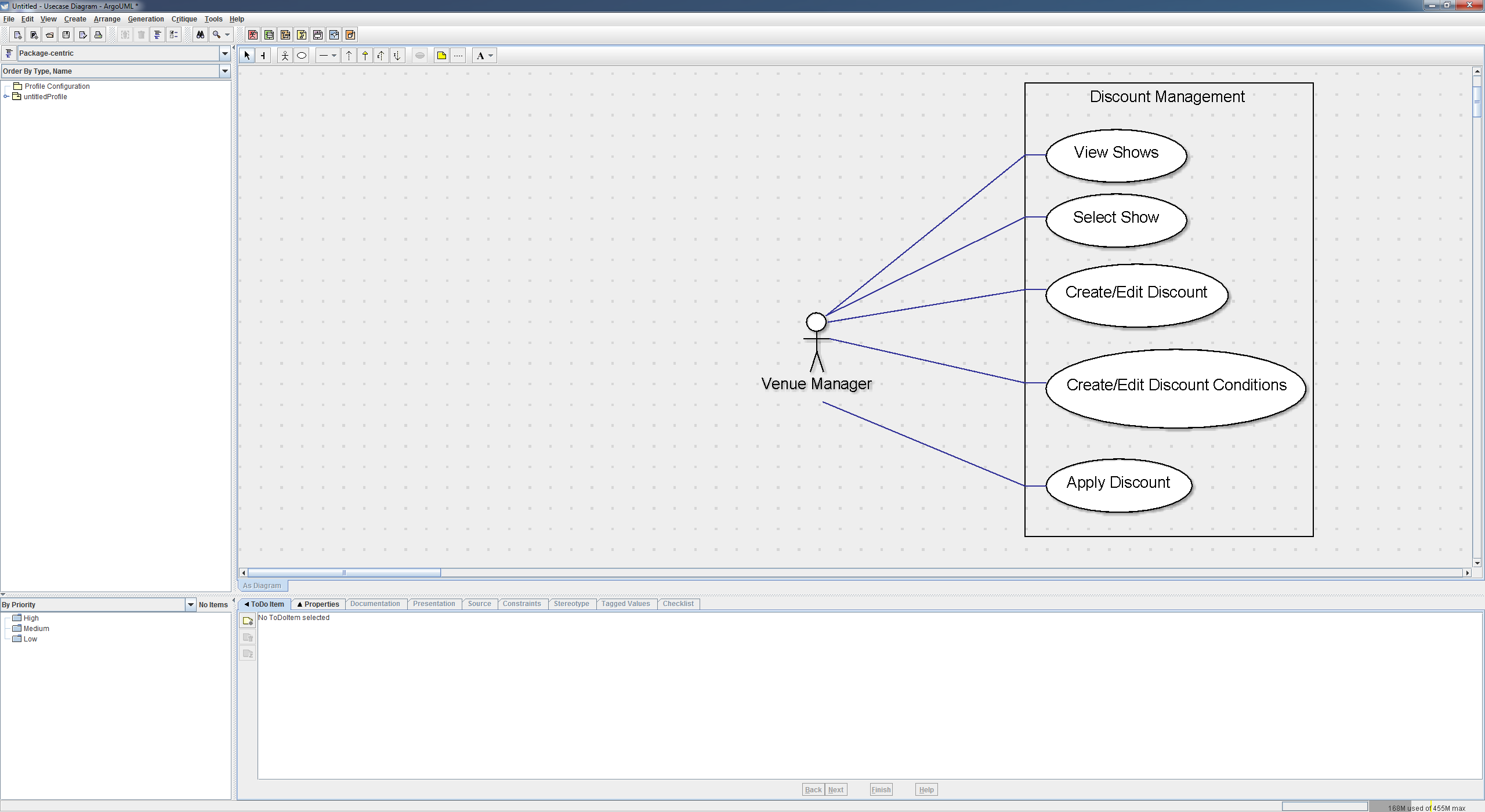
##### Cancel and Refund Ticket



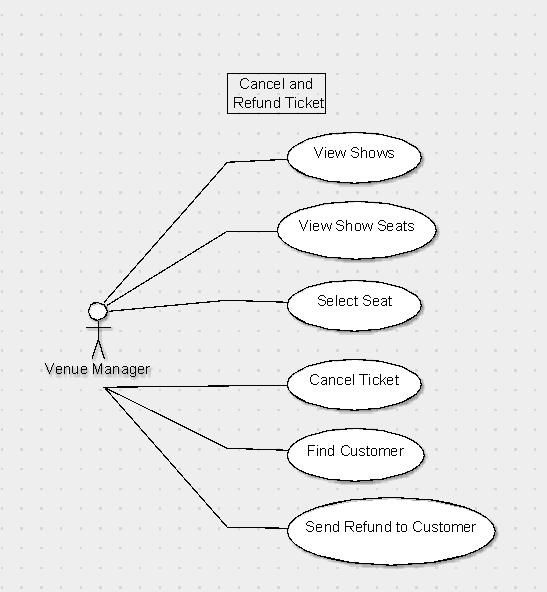
##### Show Management



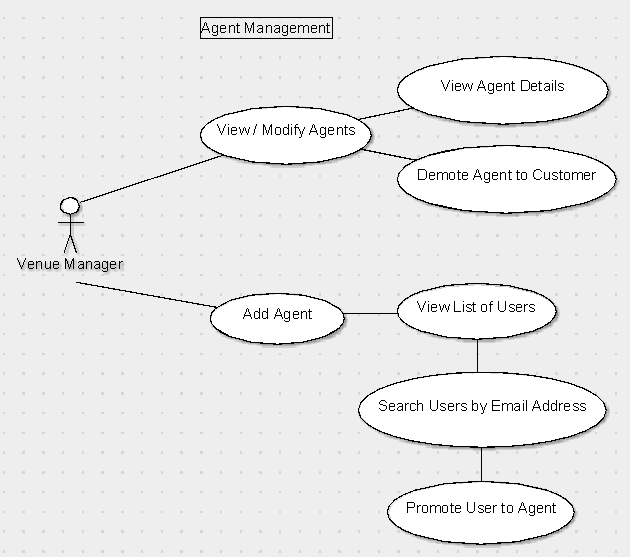
##### Discount Management



##### Cancel and Refund Ticket



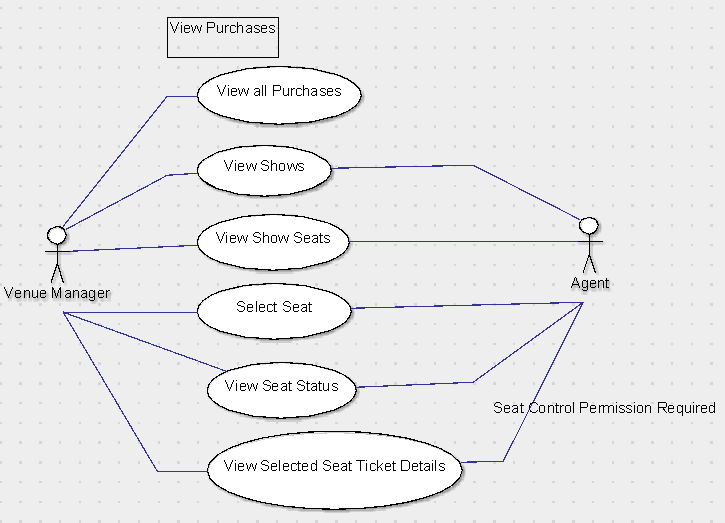
##### Agent Management



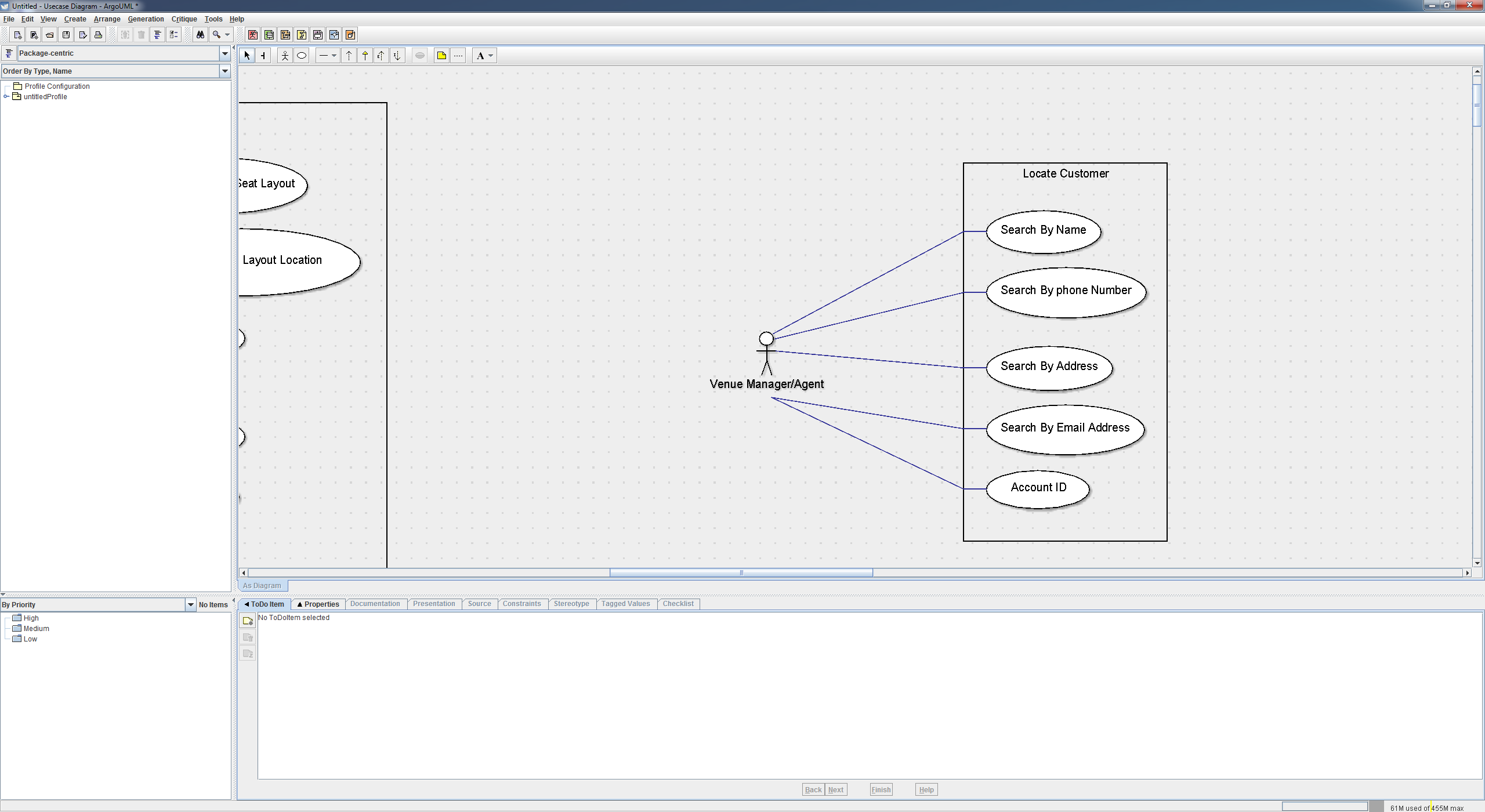
#### Venue Manager & Agent Combined

The following are use cases that apply to both the venue manager and the agent actors.

##### View Purchases



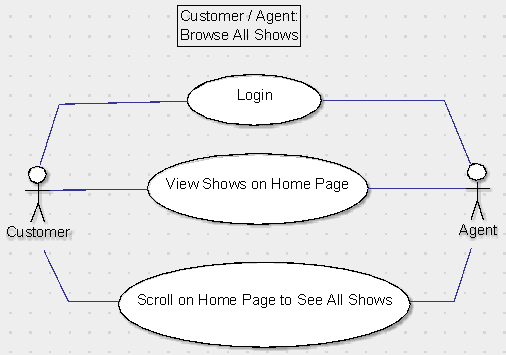
##### Locate Customer



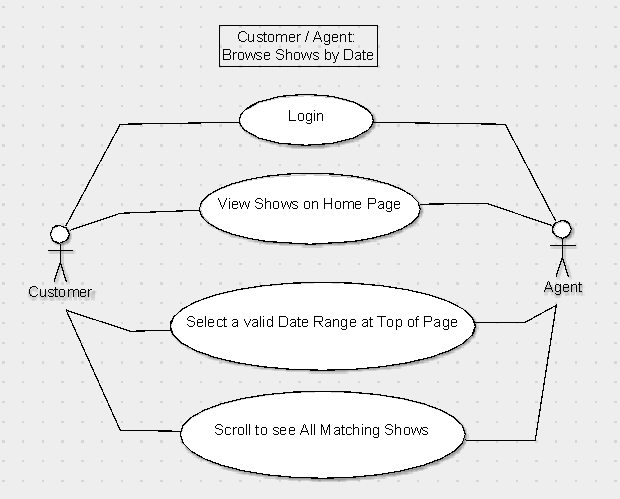
#### Customer & Agent Combined

The following are use cases that apply to both the customer and the agent actors.

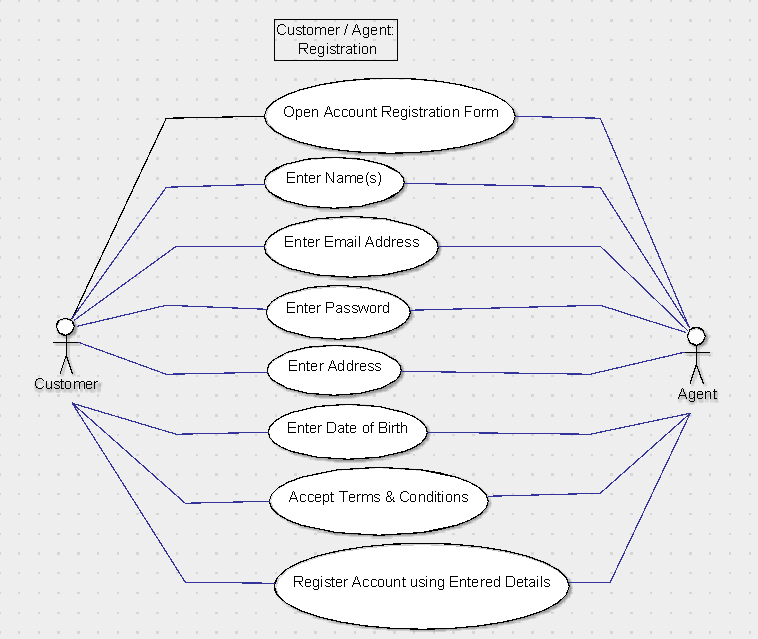
##### Browse All Shows



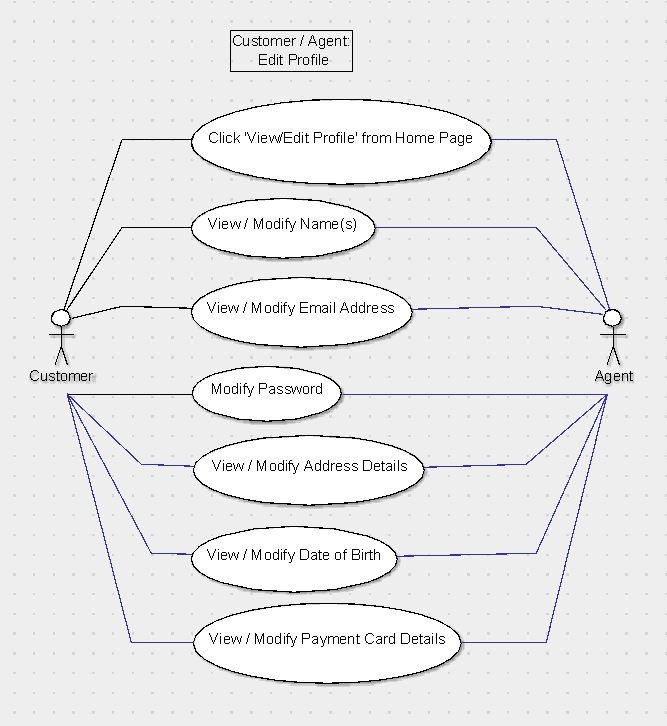
##### Browse Shows by Date



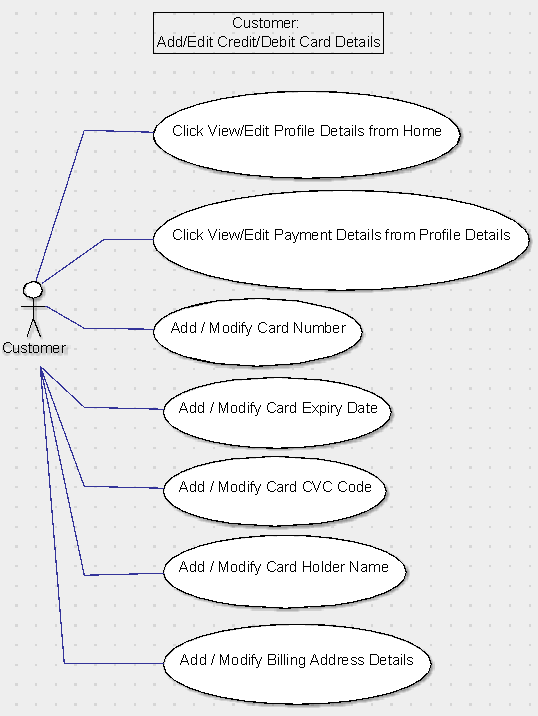
##### Register Account



##### Edit Profile



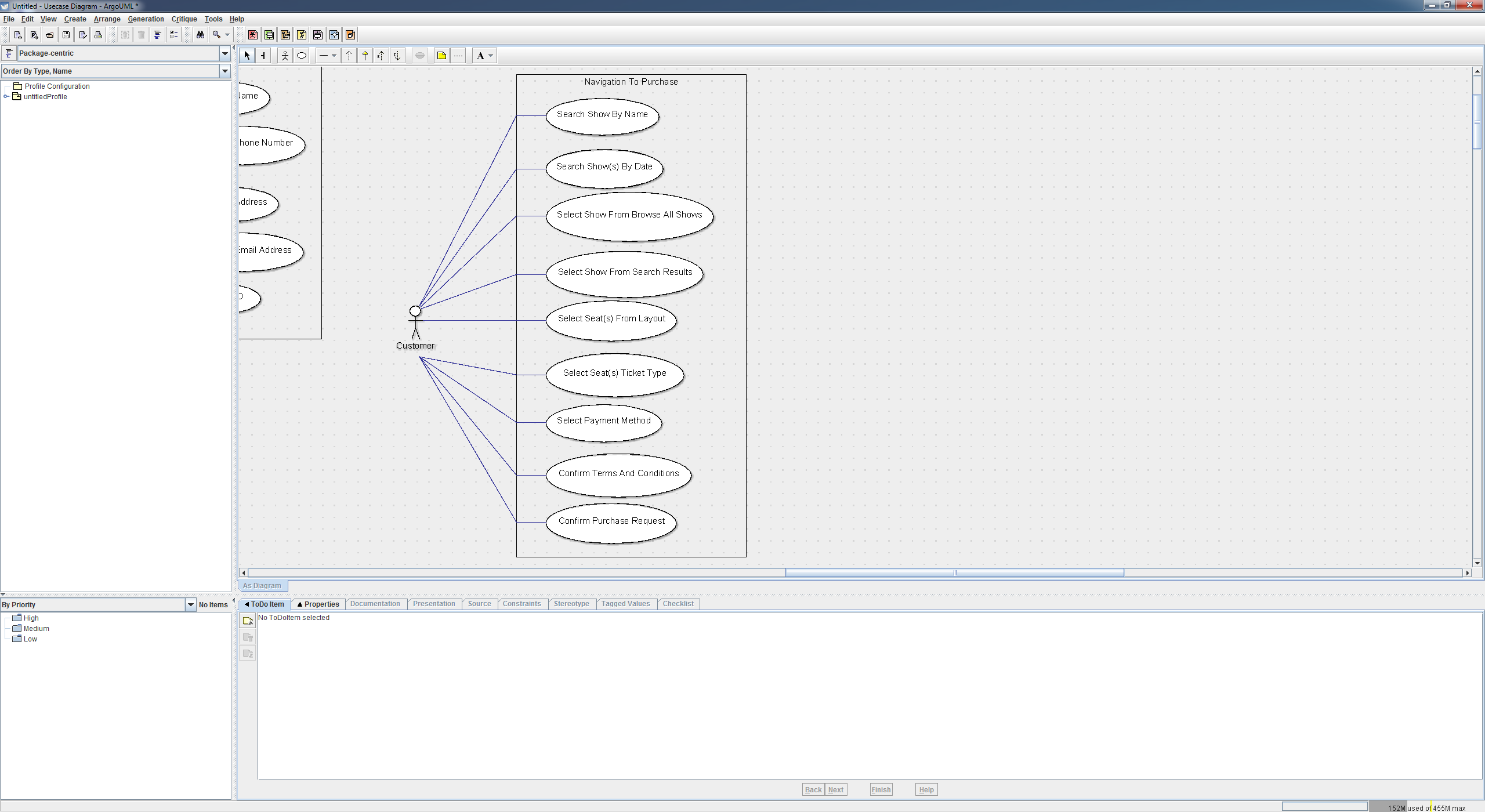
##### Add / Edit Credit/Debit Card Details



#### Customer

The following apply to the customer actor only.

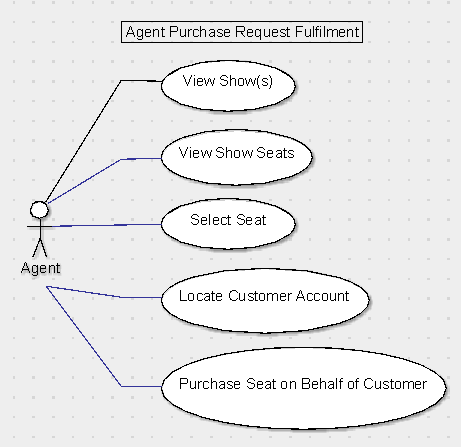
##### Navigate To Purchase



#### Agent

The following apply to the agent actor only.

##### Agent Purchase Request Fulfilment



|  |  |
| --- | --- |
| Use Cases | Descriptions |
| View Show(s) | This is the default dialog that opens after an agent logs in, it displays a list of shows in date order and has an option to search for shows by date range or view the upcoming shows. |
| View Show Seats | After a show has been selected the user can open the seating layout dialog by clicking the ‘Open Seating Layout’ button. This opens the seating layout screen which allows the agent to view seats position and status on a map through coloured seating icons, as well as select seats within his control to view their details (ticket prices, discounts, promotions, seat id, and seat name) or purchase them on behalf of a customer. |
| Select Seat | A seat can be selected with other seats, each selection is done by one click. This use case describes the result of that click. When a user selects a seat it adds the seat to the selected list. There is a text display to the right of the layout with all of the selected seats information regarding status, price, name, and ID number. Held or booked seats cannot be selected, neither can seats the agent does not control, clicking these seats does nothing. It will not add them the selected seats list. After seats have been selected that are valid for purchase the ‘Purchase Seat(s)’ button is activated. |
| Purchase Seat on Behalf of Customer | After an agent has selected valid seat or seats, they can click the purchase button This opens a new dialog showing the current purchase details, such as the seats selected, the ticket types, the cost after discounts and promotions have been applied with details of such discounts and promotions, the show details, and the customer account it is being purchased through. To complete the purchase they need to go through the ‘Locate Customer Account’ use case. They will also need to within the purchase interface, select the ticket type for each seat from Adult, Student, or Child. |
| Locate Customer Account | This is an interface that has a simple search bar, with a list box to select the type of search to perform, from postcode, name, phone number, and email address. Once a single account is selected, an OK button is available, which confirms the selection. |