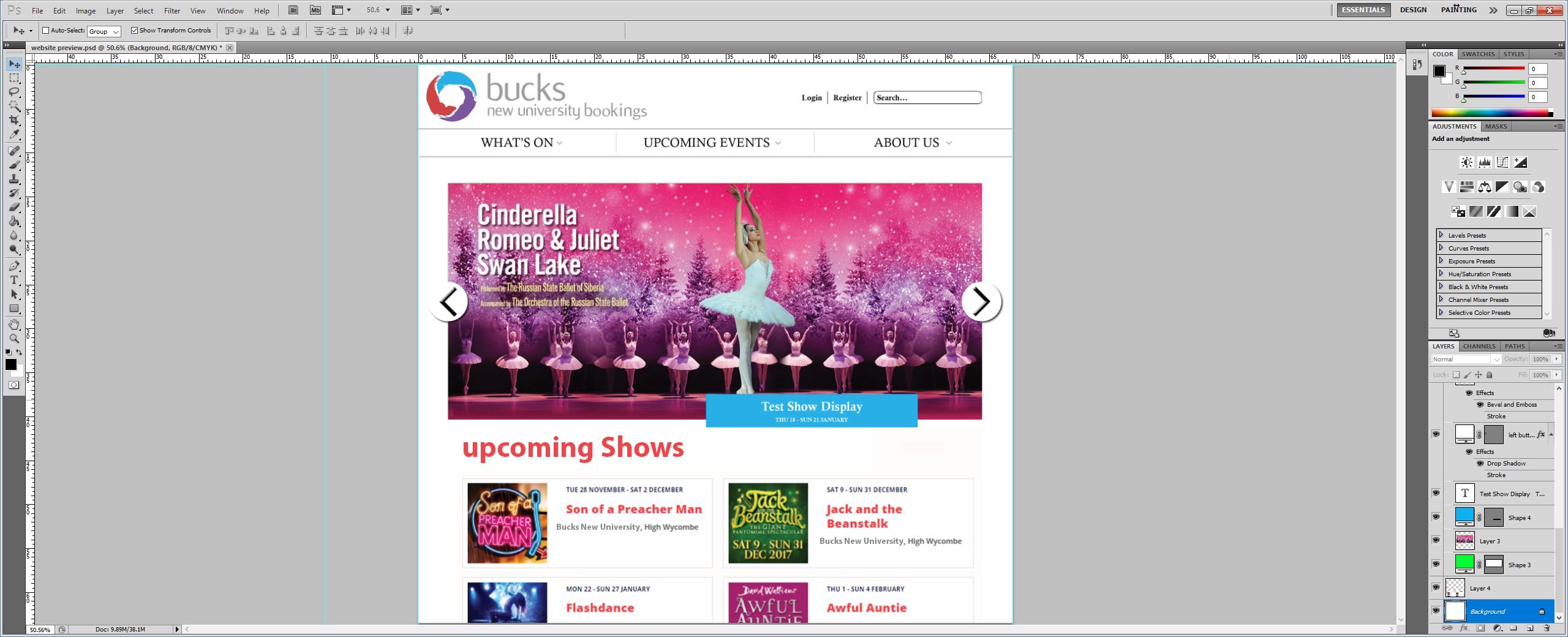
##### Annotated Screenshots

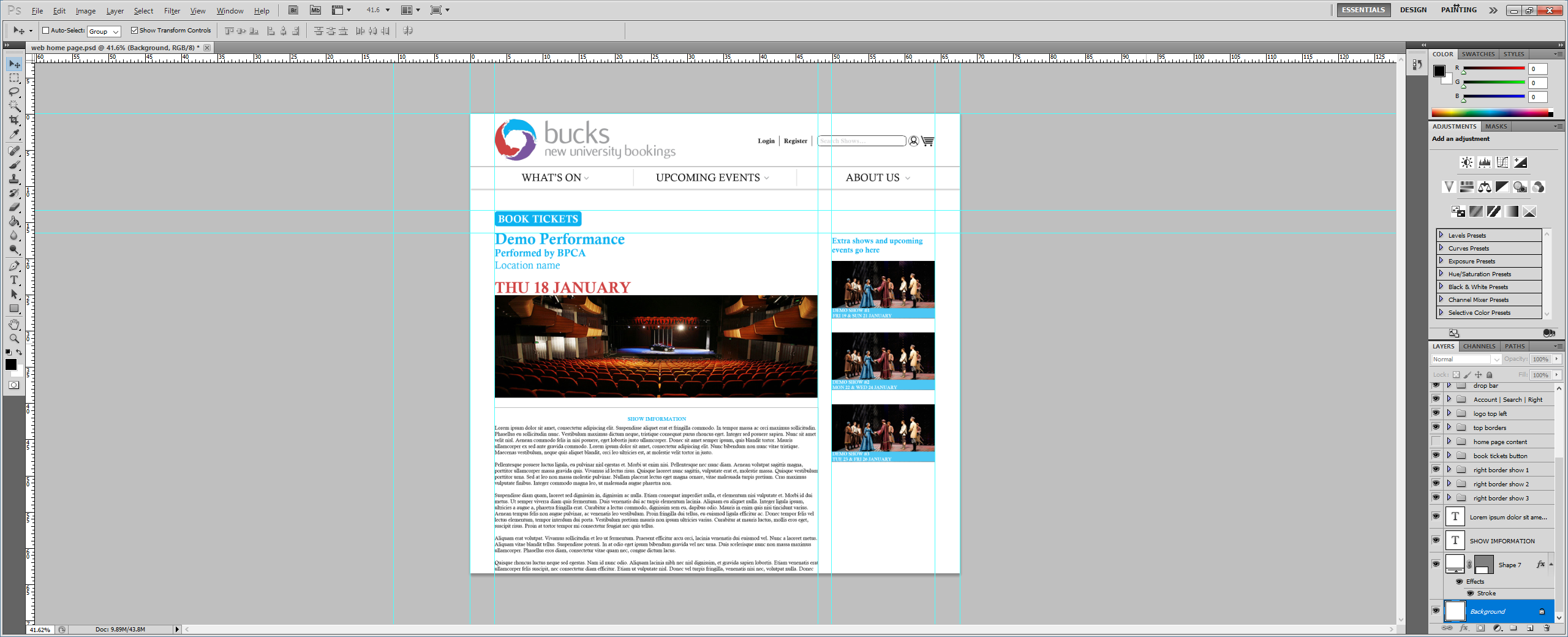
##### Home page



###### Screenshot Description Table

|  |  |
| --- | --- |
| No. | Annotation Description |
| 1 | The accounts bar will exist across the whole navigation section of the application and once successfully logged in will change “Login | Register” to “Account” while maintaining the search function from every page during navigation.  When log in is clicked it will load the below login screen which will also display another registration option. |
| 2 | The navigation bar allows the user to navigate shows in certain ways.  “WHAT’S ON” will present a list of current shows that are currently or soon to be showing.  “UPCOMING EVENTS” will show a full list of currently scheduled shows in date order, and also allow the user to search from the selection by either show name or show date, this can also be done in the “WHAT’S ON” tab.  “ABOUT US” is a page dedicated to the BCPA team and University to give a synopsis of the team and any relevant information they would like to share with the public. |
| 3 | This is a scrolling screen that will display current popular or prime events and draw extra attention to desired events. |
| 4 | This is a less detailed version of the “WHAT’S ON” tab which will just display a small amount of upcoming shows in date order, the user can’t search from this section. |

##### Display Selected Show Details



View available dates

###### Screenshot Description Table

|  |  |
| --- | --- |
| No. | Annotation Description |
| 1 | This button will allow the user to proceed with booking tickets for the current selected show. |
| 2 | This will display the selected show date and allow the user to view alternatively available dates. |
| 3 | This section is customised to show relevant, promotional or additional shows that the venue is actively promoting. |
| 4 | This section is a description of the selected show outlining show synopsis, cast and other key information. |

Requirement Analysis

This is the Requirement Analysis section which will be covering; Class diagrams; Communication diagrams and Analysis models

### Class Diagrams and Models

The below classes demonstrate the available methods to each user type, the majority will be globally inherited with a few unique to individual user types, however the majority of data is stored in individual seat objects.

#### Class Names

##### Non-Person Classes:

Show, Seating Layout, Seat, Promotion, Ticket

##### Person Classes:

Person, Venue Manager, Agent, Customer

#### Class Structures (Left-most class is parent class)

##### Non-Person Classes:

Show 🡪 Seating Layout 🡪 Seat 🡪 Promotion 🡪 Ticket

##### Person Classes:

Person 🡪 Venue Manager 🡪 Agent 🡪 Customer

#### Class Data & Methods Draft

##### Non-Person Classes:

|  |  |
| --- | --- |
| **Show** | |
| ***Data*** | ***ShowID, ShowDateTime, ShowName, ShowLengthMins, SeatLayoutLocation, ShowDiscount, ShowBasePrices, ShowSaveLocation, ShowState*** |
| ***Methods*** | **setShowName(), setShowDateTime(), setShowLength(), setShowBasePrices(), setShowDiscount(), setShowSaveLocation(), setSeatingLayoutLocation(), setShowState(), getShowName(), getShowDateTime(), getShowLength(), getShowBasePrices(), getShowDiscount(), getShowSaveLocation(), getSeatingLayoutLocation(), getShowState(), loadSeatingLayout(), loadShow(), initialise()** |

↓

|  |  |
| --- | --- |
| **Seating Layout** | |
| ***Data*** | ***MaxSeatsPerCustomer, SeatLayoutImageLocation, SeatLayoutImageSizePx, SeatImagePositionsPx, SeatDataLocation*** |
| ***Methods*** | ***setMaxSeatsPerCustomer*()*, setSeatLayoutImageLocation*()*, getImageSize*()*, setSeatLayoutImageSizePx*()*, setSeatImagePositionsPx*()*, setSeatDataSetLocation*()*, getMaxSeatsPerCustomer(), getSeatLayoutImageLocation(), getImageSize(), getSeatLayoutImageSizePx(), getSeatImagePositionsPx(), getSeatDatagetLocation(), loadSeatDataSet*()*, saveSeatingLayout*()*, loadExistingLayout*()*,* initialise()** |

↓

|  |  |
| --- | --- |
| **Seat** | |
| ***Data*** | ***SeatPrice, SeatTimer, SeatImagePositionID, SeatPromotionSaveLocation, SeatStatus, ControllingAgent*** |
| ***Methods*** | **setSeatPrice(), setSeatTimer(), setSeatImagePositionID(), setSeatPromotionSaveLocation(), loadPromotion(), setSeatControllingAgent(), setSeatStatus(), getSeatPrice(), getSeatTimer(), getSeatImagePositionID(), getSeatPromotionSaveLocation(), getSeatControllingAgent(), getSeatStatus(), playTimerAlarm(), saveSeat(), loadExistingSeat(), initialise()** |

↓

|  |  |
| --- | --- |
| **Promotion** | |
| ***Data*** | ***promotionName, promotionType, promoPriceReductionPercentages, fixedPromotionalTicketPrices*** |
| ***Methods*** | **setPromoName(), setPromoType(), setPriceReductionPercentages(), setPromoPriceReductionPercentages(), getPromoName(), getPromoType(), getPriceReductionPercentages(), getPromoPriceReductionPercentages(), savePromotion(), loadExistingPromotion(), initialise()** |

↓

|  |  |
| --- | --- |
| **Ticket** | |
| ***Data*** | ***TicketID, TicketOwner, TicketText, TicketState*** |
| ***Methods*** | ***setTicketOwner*()*, setTicketText*()*, setTicketState*()*, getTicketOwner(), getTicketText(), getTicketState(), saveTicketImage*()*, createTicketImage*()*, initialise*()** |

##### Person Classes:

Person, Venue Manager, Agent, Customer

|  |  |
| --- | --- |
| **Person** | |
| Data | Title, FirstName, MiddleName, LastName, DOB, Gender, PhoneNumber, EmailAddress, AddressL1, AddressL2, AddressTown, AddressPostCode, AddressCounty, AddressCountry, Password, UserLevel |
| Methods | setTitle(), setFirstName(), setMiddleName(), setLastName(), setDOB(), setGender(), setPhoneNumber(), setEmailAddress(), setAddressL1(), setAddressL2(), setAddressTown(), setAddressPostCode(), setAddressCounty(), setAddressCountry(), setPassword(), SetUserLevel(), getTitle(), getFirstName(), getMiddleName(), getLastName(), getDOB(), getGender(), getPhoneNumber(), getEmailAddress(), getAddressL1(), getAddressL2(), getAddressTown(), getAddressPostCode(), getAddressCounty(), getAddressCountry(), getPassword(), getUserLevel() |

↓

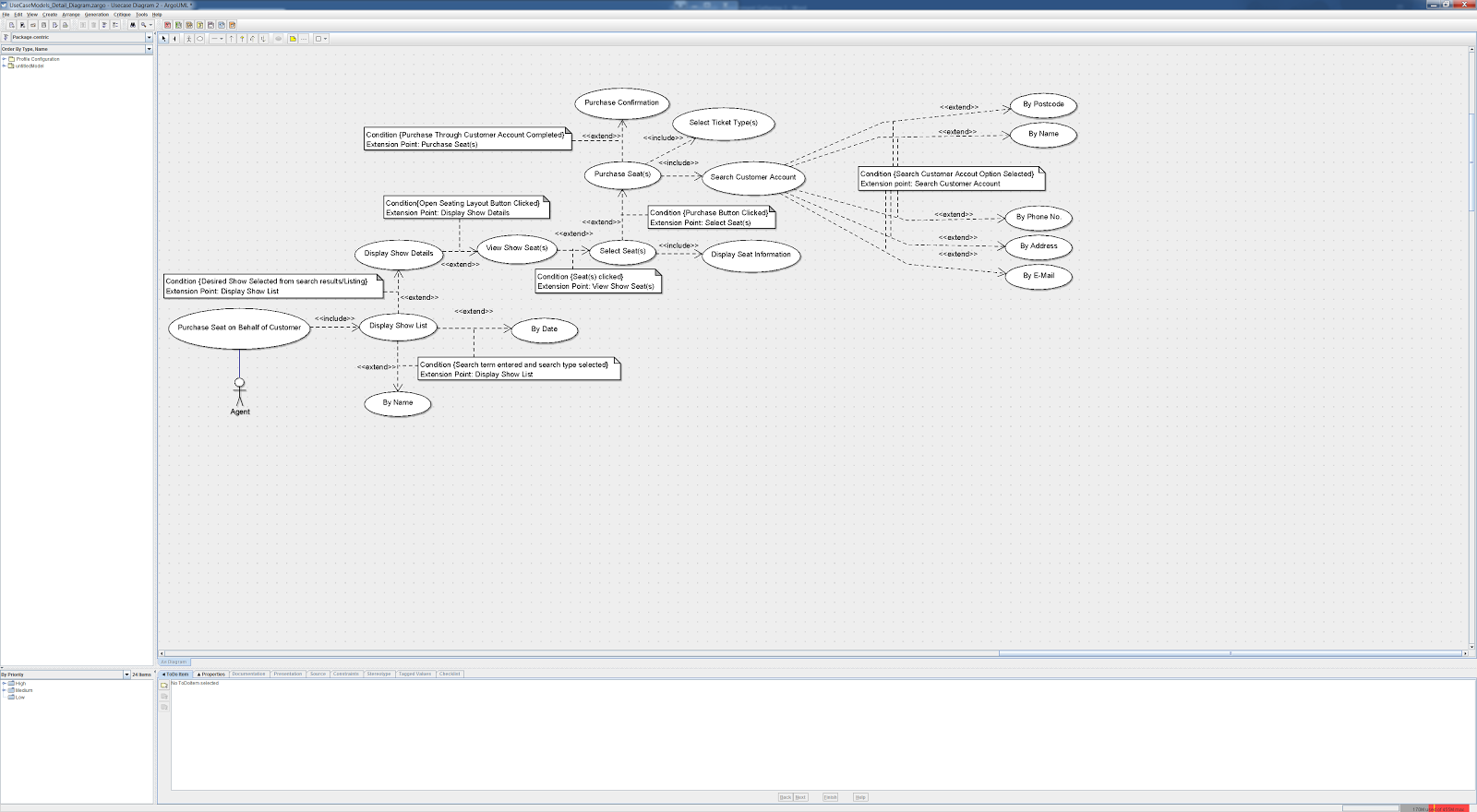
|  |  |
| --- | --- |
| Venue Manager | |
| Data | LoginHistory |
| Methods | setLoginHistory(), getLoginHistory(), |

↓

|  |  |
| --- | --- |
| Agent | |
| Data | BookingHistory, ActiveAssignedSeats |
| Methods | setBookingHistory(), getBookingHistory(), setActiveAssignedSeats, getActiveAssignedSeats() |

↓

|  |  |
| --- | --- |
| Customer | |
| Data | CardNo, CVC2, ExpiryDate, CardName, CardBillingAddress, DefaultPaymentMethod |
| Methods | setCardNo(), setCVC2(), setExpiryDate(), setCardName(), setCardBillingAddress(), setDefaultPaymentMethod(), getCardNo(), getCVC2(), getExpiryDate(), getCardName(), getCardBillingAddress(), getDefaultPaymentMethod() |

****

**Purchase Seat on Behalf of Customer** use case with extension

**The Agent will select the “Purchase Seat on Behalf of Customer” button which will take him to a new window to view shows by Date; name or by default, display all shows in date order.**

**This will display a list of shows that the Agent can then select from to display show details.**

**Once the desired show is selected the Agent may then select to view the seating layout where he can select the customer’s desired seats, up to a predetermined maximum (set by venue manager upon show creation) which will then display a box by the selected seats showing the price tiers of the selected seats; seat status and assigned agent.**

**Once the Agent has finalised the seats he must then select the purchase seat(s) button which will request him to assign the ticket types (I.E. Student, Senior, Child).**

**Once all seat types are assigned the Agent must search for the customer by; Postcode; Name; Phone Number; Address or Email Address.**

**Once the account is selected, the Agent can continue with entering Payment information to finalize the purchase, which will display purchase information such as order number, and automatically email the customer a digital receipt and order confirmation.**

|  |  |  |
| --- | --- | --- |
| **No.** | **Requirement** | **Use Case(s)** |
| **1** | **To record names, times, base price, descriptions and venue information for each show.** | **Add new show** |
| **2** | **To edit existing show information such as name date time description and base price.** | **Edit show details** |
| **3** | **To create a venue seating plan reflective of the venue layout.** | **Create seating layout** |
| **4** | **To create and save individual seat objects to hold information such as, location, price, status, assigned agent and any applicable discounts for each showing.** | **Create seat** |
| **5** | **To save seating layouts to be used as templates for future shows.** | **Save seating layout** |
| **6** | **To load previously saved seating layouts.** | **Load seating layout** |
| **7** | **To add promotional offers to individual seats.** | **Add promotion** |
| **8** | **To save promotions for future use.** | **Save promotion** |
| **9** | **To load previously saved promotions.** | **Load promotion** |
| **10** | **To add discounts to individual showings of a show.** | **Add discount** |
| **11** | **To save discounts for future use.** | **Save discount** |
| **12** | **To load previously saved discounts.** | **Load discount** |
| **13** | **To create agent accounts and assign them Agent permissions to execute privileged tasks.** | **Assign agent permission** |
| **14** | **To assign agents certain seats in each show, allocating what seats they can and can not sell.** | **Assign agent to seat** |
| **15** | **To allow agents to fulfil purchase requests on behalf of the customer.** | **Purchase ticket on behalf of customer** |
| **16** | **To allow venue managers to cancel tickets and or refund ticket cost.** | **Refund/cancel ticket** |
| **17** | **To allow agents and venue managers to find customer accounts, display account information and purchase history information.** | **Find customer account** |
| **18** | **To allow customers to register accounts and store mandatory account information such as; name; billing info; payment info; contact info etc.** | **Create customer account** |
| **19** | **To allow customers to edit and update their saved account information.** | **Edit customer account** |
| **20** | **To display a list of all available shows in date order.** | **Display all shows** |
| **21** | **To search shows by name and or date range.** | **Search show(s)** |
| **22** | **To search for the best seats based on price range.** | **Search seat(s) with price range** |
| **23** | **To select desired seats.** | **Select seat(s)** |
| **24** | **To hold desired seats.** | **Hold seat(s)** |
| **25** | **To allow the user to log into their account with their sign up email address and password.** | **Login** |
| **26** | **To allow the user to safely log out of their account.** | **Logout** |