CMP-5012B
SOFTWARE ENGINEERING
GROUP PROJECT STAGE 1
Group 25

Booking System for Events

Group members:

Martin Siddons – 100225776
Chris Sutton – 006214363
Lena Almatrodi – 100255110
Aurelie Sing Fat – 100249917

Date: 17/03/2020

Table of Contents

1. Project Description

1.1 Main objective and overview

Scenario 3: Booking System

The system is a box office ticketing application appropriate for many different types of event. For stage 1, the objective was to allow an Events Coordinator to log into the system and set up a number of events which they or their colleagues could manage, and have an additional mode of access for Agents who would be accepting bookings for those events either in person or over the phone. All data is stored in and retrieved from an offline database.

1.2 Analysis of software systems (Booking Websites)

1.2.1 Weezevent.com

UI:
- Clear interface, immediately takes you to event creation screen upon first login
- Pleasant colour scheme
- Sensible form layout
- Good fonts
- Sidebar on dashboard with drop-down menus
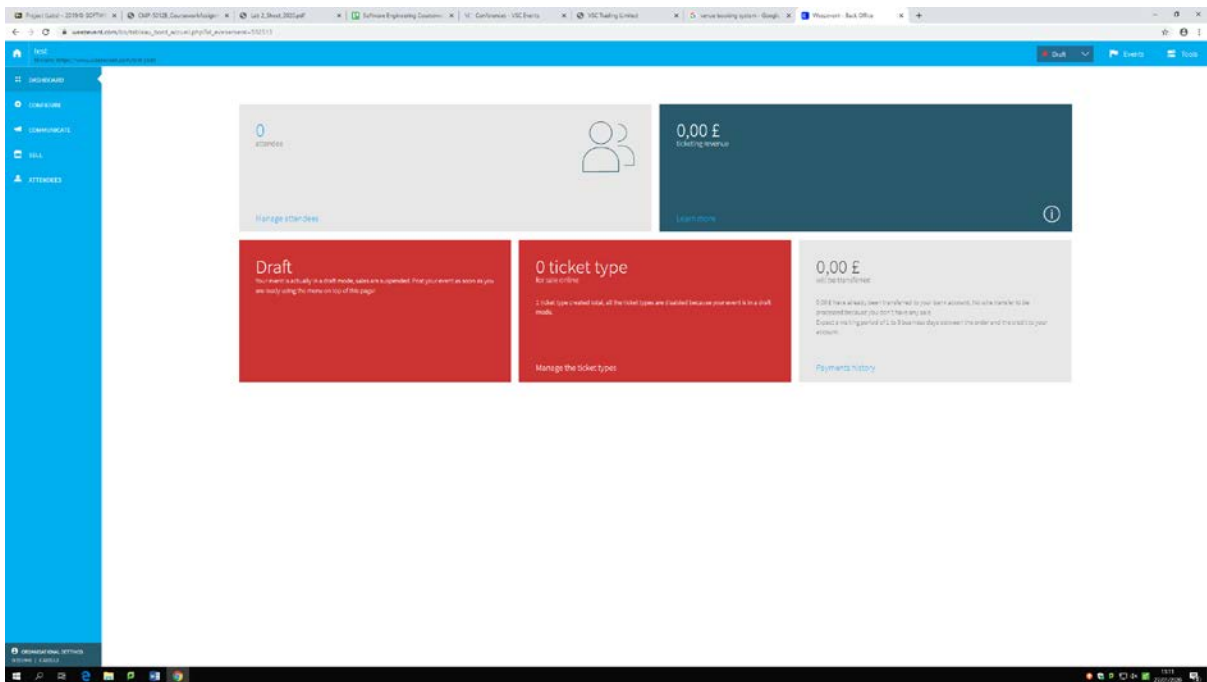- Some pages could make better use of space
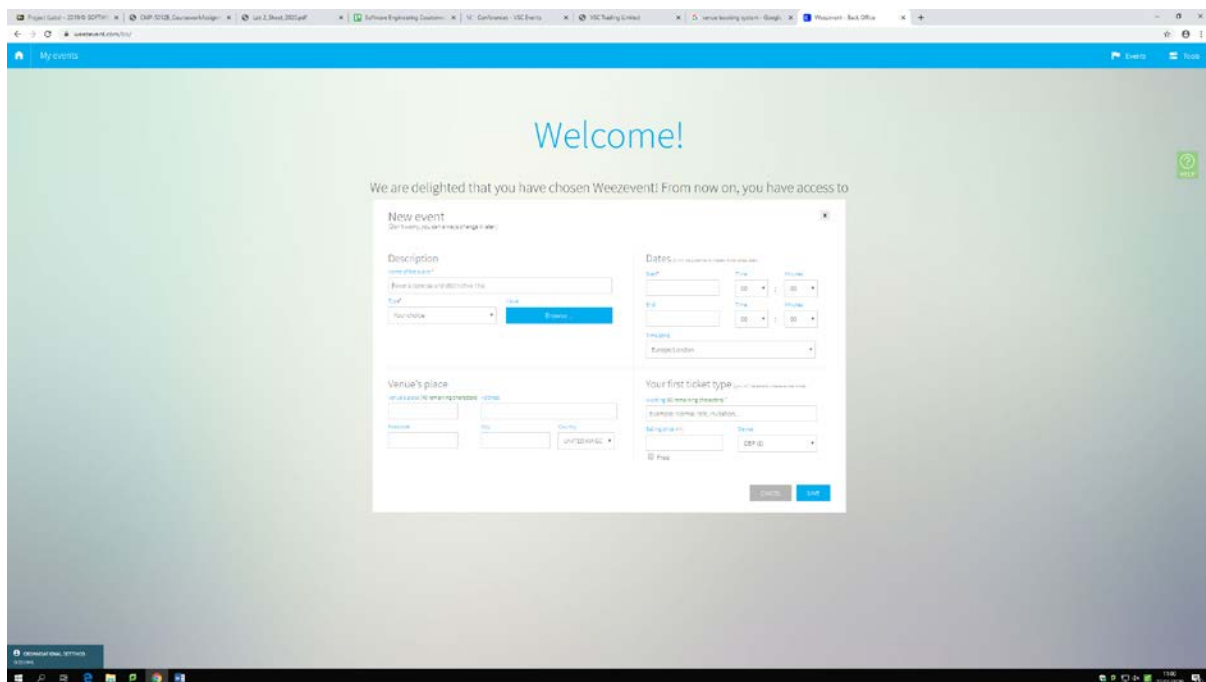
Event attributes:
- Description
- Location (includes non-uk addresses)
- Date (calendar pop-up)
- Ticket type (eg. Normal rate, invitation, back stage pass)
- Visual (eg. Picture or video to go along with event)
- Website

Ticket attributes:
- Wording
- Price
- Type

Attendee attributes:
- Name
- Email address
- Ticket issued

Notes:

Events are created in a 'draft mode' and can then be published when everything is finalised. Back-end lists number of attendees for event, revenue gained, ticket types. You can upload a design for your tickets, personalised for each ticket type. Notifications are available by email when a ticket is booked/when a quota is reached/when the event is sold out etc. You can create discount codes which are automatically applied to your tickets when they are booked. Search engine/social media referencing (needs research)? You can send invites via email for advertisement/spam/if you have a mailing list. Reports can be generated including a list of emails of those who took part in your event, your current profit for the event. You can generate and print physical tickets for sale. Payments are handled through visa and MasterCard.

Missing features:

There doesn't appear to be any way to manage account level access to data (admin, event organiser etc.).

1.2.2 Ticketsource.com

UI:
- Offers a tour of the system when you log in for the first time
- Navigation bar with dropdown menus
- Fair use of colour (customisable)
- Clear layout, if a little dated
- Could make better use of screen space
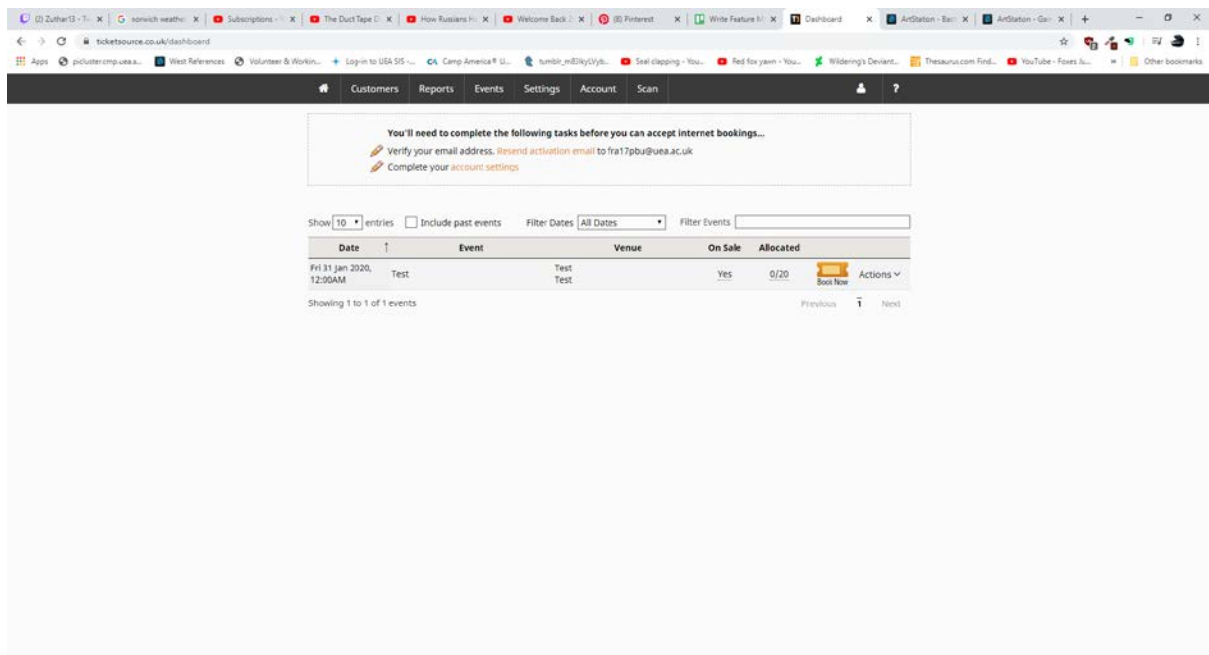
Event attributes:
- Description
- Location (includes non-uk addresses)
- Category(drop down list)
- Genre
- Date (calendar pop-up)
- Ticket type (e.g. Normal rate, invitation, back stage pass)
- Visual (e.g. Picture or video to go along with event)
- Website
- Restricted? (set max number of tickets one user can buy)

Ticket attributes:
- Wording
- Price
- Type
- Group/single
- Seat/area

Attendee attributes:
- Name
- Email address
- Ticket issued
- Membership number
- Preferred contact method

Notes:

The look and feel of the box office pages can be customised to suit any website. Payment processing is possible both through the site and through a Stripe (3rd party processor) plugin. Booking fees can be configured and applied when a customer purchases a ticket. The dashboard offers a generated link to a public event listing page. It can generate a 'buy tickets' html button for embedding in customer's own site. A ticket sales report can be generated and printed, as can a 'front of house report' which provides booking references against customer details for checking attendance manually. The system is integrated into a physical ticket scanning app for validation at the door. You can add sundry options which are also available for purchase (drinks/food etc.). There is an option to stop ticket sales some period before start time. The system uses mail chimp to contact customers for confirmation.

1.2.3 Bookinglive.com

UI:
- Adjusts to the size of the window or device being used.
- Logical layout with items in boxes, does not overwhelm the user.
- Shows all information booking manager would want to see on the front page
- Tutorial system, 'Achievements' and clickable question marks allow the user to understand the system quickly.
- Side menu with icons expands over the dashboard with wordings when hovered.
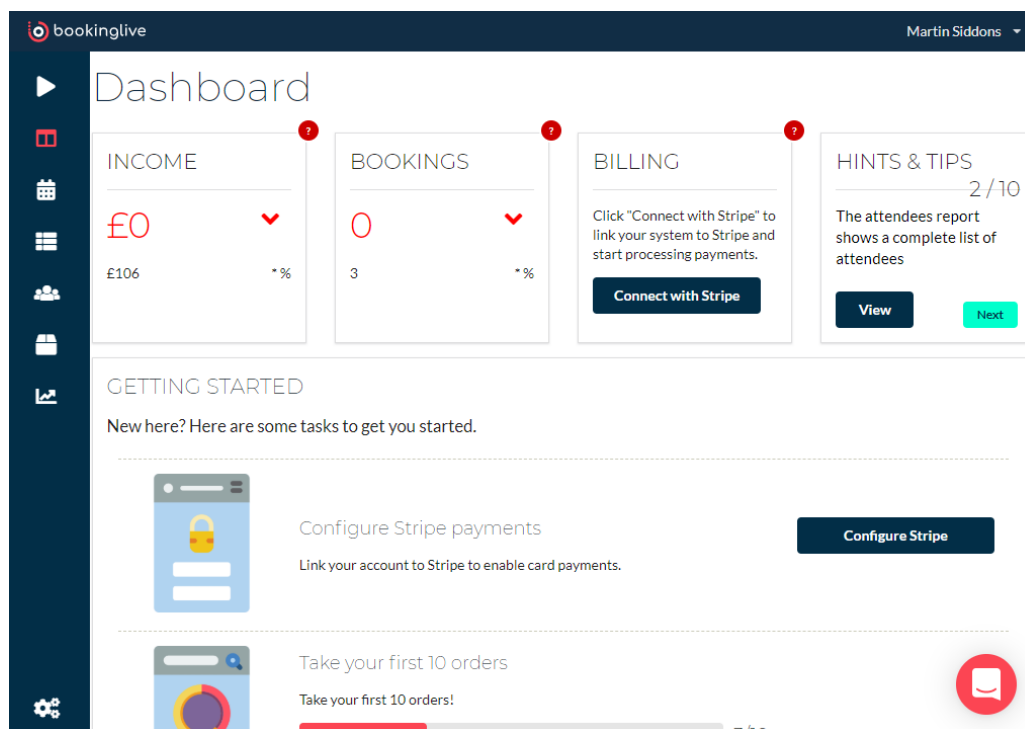
Event attributes:
- Accessed through "Items" page, unusual wording doesn't make sense.
- Allows adding and removing different ticket types
- Can share a direct link to customers to book via minisite or embed iFrame.
- Can book one event to have multiple times and dates very easily.

Ticket attributes:
- Order ID, date, status (paid, completed etc.), total amount paid
- Name, email, phone
- Event booked (with date, time and cost)
- Details of attendees
- Transaction details.

Attendee attributes:
- Unique ID, creation date, name, email
- List of attendances (date, time, cost, paid status)

Notes:
Allows Stripe of cash payment and allows Stripe account to be registered to the bookinglive account to allow for payment automation.

1.2.4 Eventbrite.co.uk

UI:
- Fully adjustable based on screen size, could use more of the screen on full screen though.
- Left-side menu with textual descriptions is visible at all times.
- Dashboard displays everything you would need to know at a glance.
- Has in-built designer system for designing unique minisite.
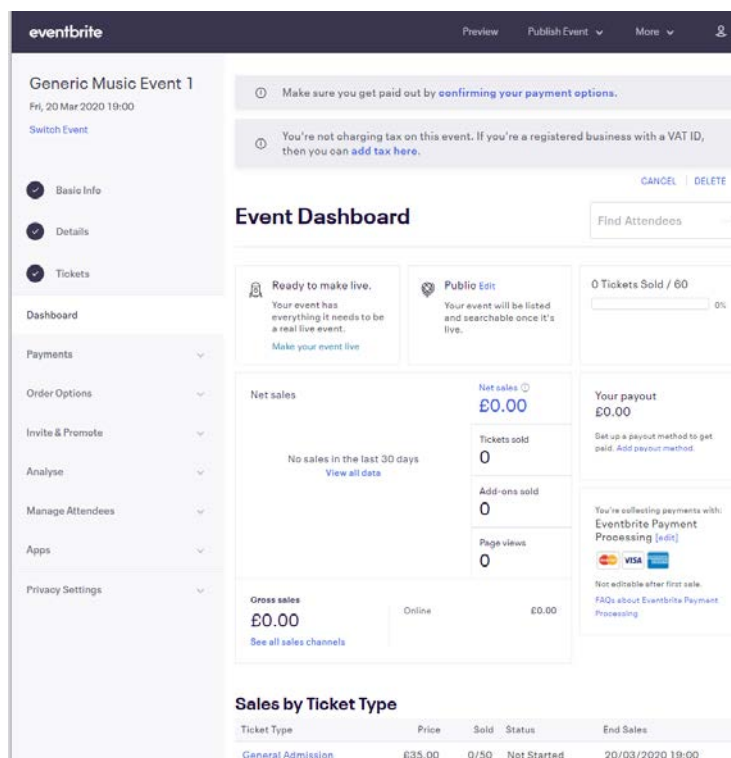
Event attributes:
- Can add text description, image, video to event.
- Define the type of event and add keyword tags, specify organiser info.
- Location, date time, whether it's repeating or not
- Can specify a huge amount of info for tickets (name, description, price etc)

Ticket attributes:
- Price, total limit, title.
- Can define more than one, including adding sundries separately.
- Can choose to take information from just the buyer or all attendees.
- Allows for customers to print their tickets, sent by email or via app.

Attendee attributes:
- Once the event is live, attendees can be seen via an attendee list and the organiser can add attendees manually.
- Can send bulk emails to all attendees
- View orders etc

Notes:

Starts off by asking you questions about the event you're booking so it can set up the system for you, and the venue can be either physical or virtual. The system allows for linking to an actual venue in Google Maps too. There is a huge amount of customisation available down to specifying taxes, local currency, different entry times per ticket, group registration etc. Can also use one ticket as a template for creating tickets for future events. There is Facebook integration and a minisite available, and the site even has the option to pay for ads on social media and affiliate programmes. This site allows users to browse and easily find events in their area and is the most feature-rich of the sites reviewed so far.

1.2.5 RSVPify.com

UI:

- Not as dynamic as other sites but still does work on mobile.
- Dashboard clear and easy to read.
- Set size for left-hand menu (viewable only on setup area)
- Good use of simple colours – grey, white and purple.
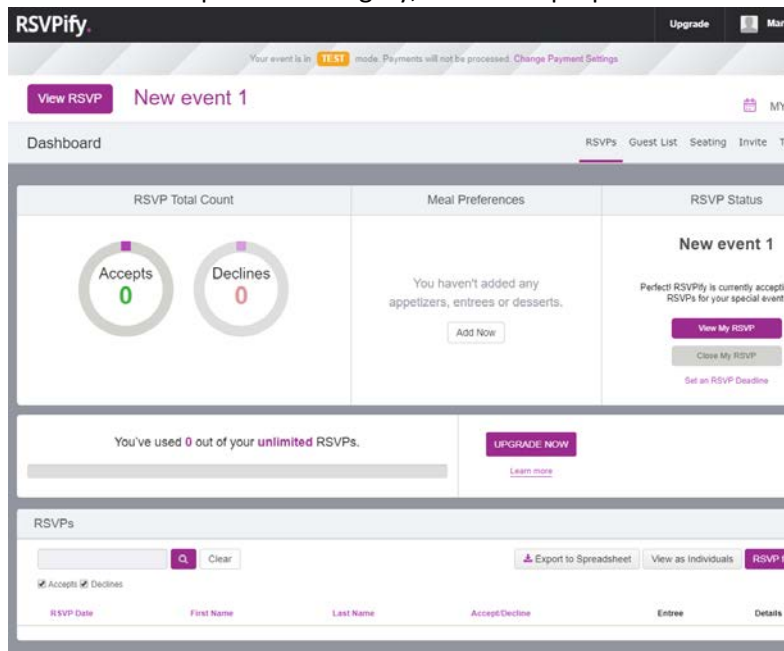
Event attributes:

- Can specify custom URL for minisite.
- Can send out email invitations and share links to the event page, as well as generate a QR code to send people to the event page.
- Heavily customisable landing page and confirmation page for customers.

Ticket attributes:

- Can configure multiple different tickets.
- Can add on details for dinner menu options.
- Can choose to pass on fees + taxes or absorb them.

Attendee attributes:

- Can view all attendees, including setting up a guest list to track how many have looked at your event page and declined.
- Can request details from all attendees or just the person booking.
- Can have the system notify event coordinator when a new RSVP is received
- Will email the attendee with details of their ticket as well as an .ics calendar invitation.

Notes:

This site is designed more for dinner parties /weddings etc. than live events and so has more specific customisation options such as being able to ask attendees questions, adding meal and seating details etc. It has a very in-depth customisation module for the pages the customer will see, like Eventbrite. It still has things in common with other booking systems, like being able to link people to the booking page, being able to see attendees etc.

1.3 Feature Matrix of Similar Systems

Feature	WeezEvent	TicketSource	Eventbrite	RSVPify	BookingLive
Create / Define Venue	✓	✓	✓	✓	✓
Create Event at venue	✓	✓	✓	✓	✓
Pick date and time for event	✓	✓	✓	✓	✓
Ability to add different ticket types for different seats/ areas	✓	✓	✓	✓	✓ (as upsell)
Ability to add different tickets for different ages	X	X	✓	✓	✓
Ability to add a visual for the event	✓	X	✓	✓	✓
Can add YouTube video to event	X	X	✓	X	✓
Ability to see revenue taken from tickets so far	✓	X	✓	✓	✓
Allows widget integration into existing site	✓	✓	✓	✓ (when not taking payment)	✓ (iframe)
Allows sales / advertising via Facebook integration	✓	✓	✓	✓	X
Gives subsite / minisite to advertise and allow bookings	✓	X	✓	✓	✓
Ability to customise the design of the ticket	✓	X	X	✓	X
Generate and print tickets for physical sale	✓	✓	X	X	X
System holds account details of attendees	✓	✓	✓	✓	✓
System sends confirmation to users when ticket is booked	✓	✓ (mailchimp)	✓	✓	✓
Email to event organiser when ticket booked / quota reached / sold out	✓	?	?	?	?
Allows creation of discount codes	✓	X	X	✓	X
Send invites by email	✓	X	✓	✓	X
Allows reports to be made	✓	✓	✓	✓	✓
Integration with Stripe / Paypal	X	✓ (Stripe)	✓ (Paypal)	✓ (Stripe)	✓ (Stripe)
Allows payment with Debit / Credit Card	✓	✓	✓	✓	✓
Configurable booking fee	✓	✓	✓	✓	X
Allows booking co-ordinator (BCO) to see attendees	✓	✓	✓	✓	✓
Allows BCO to view graphs on event attendance etc.	✓	✓	✓	✓	X (table only)

Allows BCO to add attendees to system	?	?	✓	✓	✓
Provides on-the-door ticket validation	X	✓	X	✓ (QR)	X
Seating plan management	X	✓	X	✓	X
Togglable ability to allow customers make donation to charity	X	✓	✓	✓	X (could be added as upsell)
Ability to check details are correct before publication	✓	X	✓	✓	✓
Can set a maximum number of tickets to sell	✓	✓	✓	✓	✓
Allows additional costs to be added to ticket (food/drink)	X	✓	✓	✓	✓
Ability to set a time and date to stop selling tickets	X	✓	✓	X	✓
Booking interface for customers to book their own tickets	✓	X	✓	✓	✓
Links event venue to Google Maps location	X	X	✓	X	X
Event designer system for minisite	X	X	✓	X	X
Can limit the number of ticket sales per transaction	X (default only)	X (default limit)	✓	X	X (10 by default)
Can set different entry times per ticket	X	X	✓	X	X
Minisite traffic analytics	X	X	✓	X	✓
Allows a set refund policy	X	X	✓	X	X
Can set up surveys to go out to attendees	X	X	✓	X	X
Customisable URL for minisite	X	X	X	✓	X
Specific settings for dinner can be toggled on	X	X	X	✓	X
Ability to ask guests custom questions (hotel, travel plans)	X	X	X	✓	X
Ability to send a .ics calendar invite to attendees	X	X	X	✓	X

1.4 Glossary

Customer – User who contacts an Agent in order to make a booking for an Event.

Event coordinator (EC) – User responsible for adding, editing, and deleting Events from the System. User in charge of the organisation of an event.

Agent – User responsible for liaising with Customers and creating Bookings.

Booking – Process of booking an event, undertaken by an Agent - confirmation is received after request is accepted by the System.

Event – Concert, Party, Premiere etc.

Venue – Location booked for event.

Account login – validate email and password.

Database – Records all information relating to Events, Users and Bookings.

Ticket – Physical or digital information given to a Customer which allows access to an Event.

Ticket Type – Description of a type of Ticket attached to an event e.g. VIP, Standard.

Sundry – an additional item attached to an Event, available for purchase by a Customer when making a Booking.

1.5 MoSCoW Analysis

Must Have
- Ability for EC to create, edit, and delete an event.
- System to notify customers by email when a booking has been made.
- Agent must have ability to make, edit and delete bookings for customers.

Should Have
- Ability for EC and Agent to view all events and see them in detail.
- Ability for EC to view how much total revenue has been generated by bookings.
- Ability for EC to view how much revenue has been generated per event.
- Ability for EC to generate reports of various criteria and print them in PDF form.
- Ability for EC to add sundry items to an event which can be added to a booking.
- Ability for EC to specify several different types of ticket for an event.
- Ability for EC to specify the maximum number of tickets which can be sold per event.
- The system should have the ability to generate and print/email tickets.
- Each booking should be able to contain several tickets.
- System should retain customer data for subsequent bookings.
- Some level of security and account management.
- Different levels of access for EC and Agent.

Could Have
- Ability for EC to design a custom ticket template to be printed/emailed to customer.
- Notify customers automatically by email if an event they have booked has been edited.
- Ability for EC to create discount codes which are automatically applied to a booking.

- The system could generate and distribute marketing emails to customers held within the database.
- The system could handle payment processing through some third party API.

Won't Have
- Online integration with event websites.
- The customer will not be able to make their own bookings online.
- The customer will not be able to view their own bookings or booking history.
- Any kind of registration page—not necessary in stage 1.
- Google Maps integration when showing event locations.
- Credit card processing/data handling.

2. User Stories and Use Case

2.1 User Stories

2.1.1 Events Coordinator

To do/task list	#: 1	HIGH
As an Events Coordinator,		
I want to add an event and edit existing events.		
So that I can generate tickets and offer them to customers and keep my customers and my advertisements up to date if an event changes.		

To do/task list	#: 2	LOW
As an Events Coordinator,		
I want to design a ticket.		
So that I can offer a custom ticket to my customers that represents the look and feel of my event.		

To do/task list	#: 3	MED
As an Events Coordinator,		
I want to see how much revenue I have generated, both globally and from a particular event.		
So that I can keep track of my finances and evaluate how well events are performing.		

To do/task list	#: 4	HIGH
As an Events Coordinator,		
I want to notify my customer's by email when they have made a booking and serve them a receipt when payment is received.		
So that I and my customers will be aware that they have booked successfully and will have a record of when and where the event is.		

To do/task list	#: 5	LOW
As an Events Coordinator,		
I want to notify my customers if an event that they have booked is edited.		
So that they are kept up to date on the event details, and can cancel if they want to.		

To do/task list	#: 6	LOW
As an Events Coordinator,		
I want to create discount codes for my tickets and distribute them to customers.		
So that I can entice more business and use them in my advertisements.		

To do/task list	#: 7	MED
As an Events Coordinator,		
I want to generate reports, including ticket sales, customers who are/have attended an event, a 'front of house' report, and profits by event.		
So that I can stay up to date on the performance of my events and plan for future events.		

To do/task list	#: 8	LOW
As an Events Coordinator,		
I want to customise the look and feel of the customer-facing booking/payment pages.		
So that they will be in-line with the style of my website.		

To do/task list	#: 9	MED
As an Events Coordinator,		
I want to add sundry items to the event and booking process.		
So that customers can purchase additional items before they arrive at the event, and I will be aware of how many additional items I will need.		

To do/task list	#: 10	MED
As an Events Coordinator,		
I want to be able to specify different types of ticket per event		
So that I can charge different prices depending on which seats are sold or the age of the ticketholder.		

To do/task list	#: 11	MED
As an Events Coordinator,		
I want to be able to specify a maximum number of tickets to be sold.		
So that I don't overbook an event.		

To do/task list	#: 12	LOW
As an Events Coordinator,		
I want to generate and print tickets from the system.		
So that we can sell tickets at the door of the event.		

2.1.2 Agent

To do/task list	#: 1	HIGH
As an Agent,		
I want to see all information about existing events		
So that I can provide details of the event when customers call for enquiry.		

To do/task list	#: 2	HIGH
As an Agent,		
I want to have access to the database		
So that I can make the new bookings for customers.		

To do/task list	#: 3	HIGH
As an Agent,		
I want to edit bookings		
So that I can provide customers facilities if they want to make changes to their bookings.		

To do/task list	#: 4	LOW
As an Agent,		
I want to see the number of tickets remaining for an event		
So that I can keep track of my sales and how well events are performing.		

To do/task list	#: 5	MED
As an Agent,		
I want to be able to send emails to customers		
So that I can advertise our upcoming events		

2.1.3 Customer

To do/task list	#: 1	HIGH
As a customer,		
I want to be able to create an account.		
So that I don't have to enter my details every time I book a ticket.		

To do/task list	#: 2	HIGH
As a customer,		
I want to be able to pay before getting to the venue.		
So that I don't have to carry cash, making it more convenient.		

To do/task list	#: 3	HIGH
As a customer,		
I want to receive the receipt via email.		
So that I can have a copy of the payment transaction.		

To do/task list	#: 4	LOW
As a customer,		
I want the website to have a great user interface.		
So that I can easily create and cancel bookings made.		

To do/task list	#: 5	MED
As a customer,		
I want to be able to contact a real person.		
So that if I have any questions or issues regarding the booking, it can easily be solved.		

To do/task list	#: 6	HIGH
As a customer,		
I want to be able to book for multiple people at the same time.		
So that the payment is all booked under one account.		

To do/task list	#: 7	LOW
As a customer,		
I want to be able to view previous bookings made.		
So that I can easily purchase the same event again on a different date.		

To do/task list	#: 8	LOW
As a customer,		
I want to be able to read Frequently Answered Questions (FAQ's).		
So that if I have a minor issue, I can easily get answers.		

2.2 Textual Use Case

Name: Agent makes a new booking

Version: 1.0

Goal: Save booking data to the database, linking a customer and an event.

Summary: Customer contacts Agent with inquiry about an event. Agent takes Customer details and saves them to the database. Agent then makes a booking with the requested details

Actors: Customer, Agent

Stakeholders: Customer, Agent, Events Coordinator

Preconditions: Event Coordinator and Agent have accounts in the database. Event Coordinator has created a valid event. Agent has logged into the system.

Triggers: Agent receives a phone call/in-person request from the customer.

Happy path:
- Customer inquires about making a booking on a certain event.
- Agent navigates to event list page.
- Agent double clicks on the requested event.
- The system presents the event details screen.
- Agent navigates to bookings tab.
- Agent clicks on 'Make new Booking' button.
- The system presents the new booking screen.
- Agent requests customer username, password, and address.
- Customer provides details.
- Agent enters details into text fields and clicks 'Add new Customer' button.
- System saves new customer to database.
- Agent selects new customer.
- Agent types in requested number of tickets of the requested type.
- Agent types in requested number of sundries of the requested type.
- Agent clicks on 'Save Booking' button.
- System presents a success message.
- System saves new booking to database.
- System generates confirmation email and sends it to customer.

Unhappy path:
- Customer inquires about making a booking on a certain event.
- Agent navigates to event list page.
- Agent double clicks on the requested event.
- System cannot locate requested event in database.
- Error message presented.
- Failure logged.

Unhappy path:
- Customer inquires about making a booking on a certain event.
- Agent navigates to event list page.

- Agent double clicks on the requested event.
- The system presents the event details screen.
- Agent navigates to bookings tab.
- Agent clicks on 'Make new Booking' button.
- System fails to correctly present empty booking page template.
- Error message presented.
- Failure logged.

Unhappy path:
- Customer inquires about making a booking on a certain event.
- Agent navigates to event list page.
- Agent double clicks on the requested event.
- The system presents the event details screen.
- Agent navigates to bookings tab.
- Agent clicks on 'Make new Booking' button.
- The system presents the new booking screen.
- Agent requests customer username, password, and address.
- Customer provides details.
- Agent enters details into text fields and clicks 'Add new Customer' button.
- System fails to save new customer to database.
- Error message presented.
- Failure logged.

Unhappy path:
- Customer inquires about making a booking on a certain event.
- Agent navigates to event list page.
- Agent double clicks on the requested event.
- The system presents the event details screen.
- Agent navigates to bookings tab.
- Agent clicks on 'Make new Booking' button.
- The system presents the new booking screen.
- Agent requests customer username, password, and address.
- Customer provides details.
- Agent enters details into text fields and clicks 'Add new Customer' button.
- System saves new customer to database.
- Agent selects new customer.
- Agent types in requested number of tickets of the requested type.
- Agent types in requested number of sundries of the requested type.
- Agent clicks on 'Save Booking' button.
- System fails to save new booking to database.
- Error message presented.
- Failure logged.

Unhappy path:
- Customer inquires about making a booking on a certain event.
- Agent navigates to event list page.
- Agent double clicks on the requested event.
- The system presents the event details screen.
- Agent navigates to bookings tab.
- Agent clicks on 'Make new Booking' button.

- The system presents the new booking screen.
- Agent requests customer username, password, and address.
- Customer provides details.
- Agent enters details into text fields and clicks 'Add new Customer' button.
- System saves new customer to database.
- Agent selects new customer.
- Agent types in requested number of tickets of the requested type.
- Agent types in requested number of sundries of the requested type.
- Agent clicks on 'Save Booking' button.
- System presents a success message.
- System saves new booking to database.
- System fails to generate confirmation email and send it to customer.
- Error message presented.
- Failure logged.

Post conditions: System database contains customer and booking details. Customer is notified by email that the booking has been made successfully.

Notes: Error messages are populated with data returned from the JVM or the database to provide details.

Author and date: Chris Sutton, 17th March, 2020

2.3 Personas

Adam Marshall

<u>Demographics</u>

Age: 43

Location: Thorpe St. Andrew, Norwich

Education: Business Management

Occupation: Event Organiser, Ideal Events Inc.

Family: Wife, two sons

"I want to modernise the way we manage events and customers at Ideal Events."

<u>Goals</u>

- To offer an excellent event management service to my customers
- To ease the workload of my agents
- To extend the marketing reach of Ideal Events Inc.

<u>Frustrations</u>

- I have to spend a great deal of time correcting errors/making edits in our records.
- I can't offer an automated way of online booking to my customers.
- My agents have to waste time manually collating reports for me to view the health of the business.

<u>Bio</u>

John Doe is a hardworking father, husband and director of Ideal Events Inc. He is extroverted, friendly, and a capable leader, but his patience is lacking. Although he remains in close contact with several loyal customers, the company's current slow method of taking bookings for events makes him feel that they are losing business to others with more modern methods. He is eager to drag his company (employees and all) into the 21st century, even if he himself is unsure exactly what that will require. His agents aren't quite as gung-ho in their approach, so there is a small rift growing between their traditional outlook, and his more forward-thinking one. Nonetheless, he is held in high regard by his employees broadly for his commitment to them and the company. This opinion extends to his customers, who would like to stay with him if he can provide an up to date service.

Jessica Williams

<u>Demographics</u>

Age: 29

Location: Norwich

Occupation: Agent, Ideal Events Inc.

Language: English and French

Family: Parents

<u>Goals</u>

• To hit her target of number of bookings each month
• To offer excellent services to customers
• To get promoted by the end of the year

<u>Frustrations</u>

• I always have to be in the office to take bookings and often gets tired to take calls from customers.
• When the call service line breaks in between a call with a customer
• When there are miscommunications with the events coordinator

<u>Bio</u>

Born in Norwich, Jessica attended Eaton Primary School and Hellesdon High School. She graduated from UEA with a degree in Management and Marketing. Jessica loves dogs and volunteers at Norfolk animal sanctuary. She has been working in the marketing and events field for four years and enjoys her job. She is very ambitious and wants to get promoted to get a higher pay in order to move out from her parent's house. During her free time, she likes to listen and watch music performances on YouTube which also enables her to discover new artists.

Rachel Walker

Demographics

Age: 28
Location: Fairfield, Liverpool
Education: Nursing school
Occupation: Nurse
Family: Husband, infant son

"I want to book tickets to a local gig for my husband and I while my mother looks after our son."

Goals

• To locate the gig's website
• To make a booking in as quick and pain-free a process as possible
• To be aware of any options or categories relevant to the purchased ticket

Frustrations

- I currently have to book tickets over the phone, rather than online
- The booking process is tedious and I have to repeat my details over the phone whenever I want to make a booking
- The tickets must be sent to me in the mail rather than electronically, which means I may lose them

Bio

Rachel Walker is an up-and-coming nurse who is pulling long hours to get ahead in her career. Her husband works in construction, which means he also has an unpredictable schedule. Their three year old son is often left in the care of Rachel's mother. Both her and her husband have sacrificed a good portion of free time in order to work towards a better future for the family, but this means that they have had very little time to spend together as a couple. Music brought them together—they used to attend gigs frequently when they were younger—and recently Rachel has been looking around for the bands that they used to follow in order to re-discover them together and re-ignite their relationship. She is fairly familiar with technology, although she wouldn't describe herself as technical. She spends a decent amount of time on her phone, especially on breaks at work, but barely uses her laptop at home.

Simon Perez

<u>Demographics</u>

Age: 46
Location: Croydon, London
Occupation: Accountant
Family: Wife,   1 daughter and 1 son

<u>Interests</u>

- Attend gigs with his son
- To learn Spanish before turning 50
- Football fan

<u>Frustrations</u>

- When the gigs are already sold out without even knowing tickets were on sales
- I don't like giving my personal details when booking tickets on the phone

<u>Motivations</u>

- Professional development
- Family

<u>Bio</u>

Simon Perez attended St Peter's primary and secondary school in Croydon. In sixth form, he became interested in becoming an accountant after a summer internship and decided to complete an accounting course at Queen Mary University of London. Simon often has to pull long hours and make sacrifices in order to make ends meet due to the cost of living in London. He spends most of his limited free time with his family and enjoys playing football with his teenage son. Simon used to play the guitar for his secondary school's musical band and also shares the same passion for electric guitar with his son. Both of them like to attend musical performances and are looking forward to listening live to Simon's favourite band.

Lauren Smith

Demographics

Age: 36
Location: Chorlton, Manchester
Job: Graphic Designer
Family: Husband, son, daughter.
Languages: English, French, Portuguese

Interests
- Gaming
- Foreign Languages
- Traveling
- Listening to music

Frustrations
- I prefer using a website to book a gig, rather than booking it on the phone, as I believe the data being sent is more secure that way.
- No discounts or offers are given for those who've booked multiple tickets previously.
- I'd like to live in Brazil in the near future, as all my family members live there.

Technology

- Before booking an event or buying tickets, she checks with friends first on WhatsApp.
- Discovers new music and listens to music on Spotify.
- Checks Pinterest for ideas on creating new paintings.

Bio
Lauren Smith studied her BA in Graphic Design in Brighton University and currently works in a Corporate Branding company. She has two daughters, ages 12 and 14. Both of her daughters are considered to be great singers and like attending musical events on the weekends. They have been in many talent shows held by the school and Lauren attends and supports them. Her husband works as a Web Developer in Curitiba, Brazil, and visits twice a month. He also supports his daughters by viewing the talent shows online, and he suggests they sing in Britain's Got Talent. Both of their daughter's goal is to become professional singers to share their talent to the world. Lauren spends most of her leisure time listening to Alternative and Classical music, as well as playing the piano. She also loves painting and reading Science Fiction books in her free time.

Joshua Wilson

Demographics

Age: 28 years old
Location: Beeston, Nottingham
Job: Business Analyst
Family: Divorced
Languages: English, German

Interests

- Listening to music
- Biking
- Sports

Frustrations

- I'd like to book tickets as efficient as possible.
- If there are discounts or offers, I should be notified by e-mail, rather than checking the website.

Technology

- Uses Facebook to announce any musical events.
- Also uses Facebook to post cover of famous songs using the drums.

Bio

Joshua Wilson studied in the University of Nottingham for a Bachelor's degree in Business Administration. He then earned a Master's Degree in Business Analysis and Consulting. He was married at the age of 22, however was then divorced by 28 years old. He currently hangs out with his friends every weekend to attend music events. Joshua has a passion in music and is a professional drummer. He learned to drum at the age of 14, and continued to pursue his talent. One of his goals was to join a band, however he decided to do that after his Master's degree, as he believed that was a better time to focus on music. He mainly listens to Hip Hop, Alternative Rap, and Rock music in his free time. His friends support him with his music career, by attending his events as a solo artist.

3. Object Oriented Analysis

3.1 Natural Language Modelling

NLM was used during the OOA process to identify candidate classes for the initial class diagram. The categories for each Actor were informed by the Use Cases in Section 2.1

3.1.1 Events Coordinator (EC)

Log into the system:
1. The EC types their details into the account login screen
2. The system processes their login
3. The system returns the EC 'dashboard' upon successful login

Add/edit existing events
1. From the 'dashboard':
2. The Events Coordinator clicks on 'Add new event' button.
3. The system presents the new event screen, providing options for adding start time/end time, date, ticket types, maximum number of tickets, prices, and sundries.
4. The Events Coordinator enters event details and clicks on 'Publish event'.

Design ticket
1. From the 'dashboard':
2. The Events Coordinator clicks on 'Edit event' button.
3. The system presents the event edit screen.
4. The Events Coordinator clicks on 'Upload ticket design' button.
5. The system allows them to select an image file from their computer, and uploads the image, attaching it to the ticket.
6. The Events Coordinator clicks on 'Apply'.

View revenue
1. From the 'dashboard':
2. The Events Coordinator navigates to the events list screen.
3. The system presents each event's total revenue (a function of ticket sales).

Notify customers of event changes
1. From the 'dashboard':
2. The Events Coordinator navigates to the events list screen.
3. The Events Coordinator clicks on the event they wish to edit.
4. The system presents the edit event screen.
5. The Events Coordinator makes any edits they wish.
6. The system generates an email to notify any customers who have tickets to that event.
7. The system sends the emails.

Create discount codes
1. From the 'dashboard':
2. The Events Coordinator navigates to the events list screen.
3. The Events Coordinator clicks on the event they wish to create a discount code for.
4. The system presents the edit event screen.

5. The Events Coordinator types the required discount code in the 'Discount code' field and chooses how much (in percent) a customer will save.
6. The Events Coordinator clicks on 'Apply'.

Generate reports
1. From the 'dashboard':
2. The Events Coordinator navigates to the Reports page.
3. The Events Coordinator selects the scope of the report from several drop-down boxes and clicks 'Generate'.
4. The system generates the report according to the specified scope, and presents it as a PDF.

Customise the look and feel of the customer-facing pages:
1. From the 'Edit Event' screen EC clicks 'Customise look'
2. EC specifies an image and colour scheme for the corresponding event.
3. EC clicks 'Accept', system changes the image and colour scheme for the corresponding ticket sales page.

Add sundry items to an existing event:
1. From the 'Edit Event' screen:
2. EC clicks the 'Add new items' button from the Sundry Items section
3. EC adds the details of the new items including name, description and price and sales limit
4. EC clicks accept, system adds tax and additional (pre-set) fees to price and adds it to the event.

Add different types of ticket to an existing event:
1. From the 'Edit Event' screen:
2. EC clicks the 'Add new ticket' button in the tickets box of the system
3. EC enters the new ticket details including name, price, ticket sales limit
4. EC confirms details and the system adds the ticket to the event for sale and updates the maximum number of tickets to be sold to include the new tickets

Change the maximum number of tickets to be sold:
1. From the 'Edit Event' screen:
2. EC clicks the 'Total Max tickets' field and enters the limit number, which cannot be lower than the amount of tickets sold so far
3. System changes the maximum number of tickets for sale and tickets remaining based off of this number

Generate and print tickets from the system:
1. From 'Dashboard', EC clicks the events button
2. EC finds and selects the event they wish to print tickets from
3. EC clicks the 'print tickets' button
4. EC enters how many tickets to print – system limits to remaining tickets available
5. System builds and sends request to specified printer
6. System deducts number of printed tickets from remaining ticket count

3.1.2 Agent

Logging into the system:
1. The agent types their details into the account login screen
2. The system processes their login
3. The system returns the agent 'dashboard' upon successful login

Making a booking:
1. Customer calls agent to make a booking or for enquiry.
2. From the 'dashboard':
3. Agent accesses details of event.
4. System shows event description; artists, start time, end time, ticket types, prices and sundries available.
5. Agent asks for customer details.
6. Agent makes a booking on the event.
7. System records customer details for the booking.
8. Agent asks customer for payment details.
9. System processes payment.
10. System generates receipt and emails ticket to customer.

Edit a booking:
1. Customer calls to make changes to their booking.
2. From the 'dashboard':
3. Agent accesses the customer panel to view details of customer's booking.
4. Agent makes changes to customer's booking.
5. System records the changes made.
6. System processes additional payment if required.
7. System generates new receipt and emails edited ticket to customer.

Send emails to advertise an event:
1. From the 'dashboard':
2. Agent clicks 'Events' button, system displays list of events.
3. Agent selects the event they wish to email about to view event's details.
4. System displays event information page.
5. Agent clicks 'advertise event' button.
6. The system accesses the list of previous customers that are happy to be emailed
7. System sends emails to advertise upcoming event.

3.1.3 Customer

Log into the system:
1. The customer phones the box office to speak to an agent
2. From the agent 'dashboard':
3. The agent clicks the 'find customer' button and requests the customers' name and postcode
4. The agent enters the above details into the search box on the 'find customer' screen
5. The system displays the customer's account information including full name, address, email and purchase history on the 'customer details' screen

Creating a new account:
1. The customer phones the box office to speak to an agent
2. From the agent 'dashboard':
3. The agent clicks the 'Add new user' button, the system displays the corresponding page
4. The agent asks the customer for their details including name, address and email for contact
5. The agent confirms with the customer that they are happy to receive emails from us
6. The agent confirms the above fields, the system adds a record for the new customer

Making a booking:
1. From the agent 'customer details' screen:
2. The agent clicks the 'new booking' button
3. The customer confirms which event they would like to book for, the agent clicks the booking as listed on the given screen
4. The customer gives the number of tickets and any extras they would like to order
5. The agent checks the tickets are bookable, confirms the customer's request and submits the details to the system
6. The system creates a new booking for that customer on the selected event and removes the number of tickets sold from the total available
7. The system generates a ticket and receipt which is emailed to the customer
8. The agent confirms the above to the customer

View a previous booking:
1. From the agent 'customer details' screen:
2. The agent finds the booking the customer has requested and clicks it
3. The agent relays the requested information to the customer

View the frequently asked questions (FAQs):
1. The customer phones the box office to speak to an agent
2. The customer relays the question to the agent
3. The agent gives the customer the answer to their question

3.2 Initial Class Diagram

4. Object Oriented Design

4.1 Completed Class Diagram

UML Class Diagram

View / Presenter classes

- LoginView.fxml — Passes gesture to / Selects → LoginViewPresenter
- DashboardView.fxml — Passes gesture to / Selects → DashboardViewPresenter
- CustomerListView — Passes gesture to / Selects → CustomerListPresenter
- CustomerView — Passes gesture to / Selects → CustomerViewPresenter
- EventListView — Passes gesture to / Selects → EventListPresenter
- EventView — Passes gesture to / Selects → EventViewPresenter
- BookingView — Passes gesture to / Selects → BookingViewPresenter
- ReportsView — Passes gesture to / Selects → ReportsPresenter

EventBookingSystem
+ main (String[] args)

(Initialises / Initialised by → LoginViewPresenter)

<<Enumeration>> AccountType
- -CUSTOMER
- -AGENT
- -EC

<<Abstract>> User
- #username: String
- #password: String
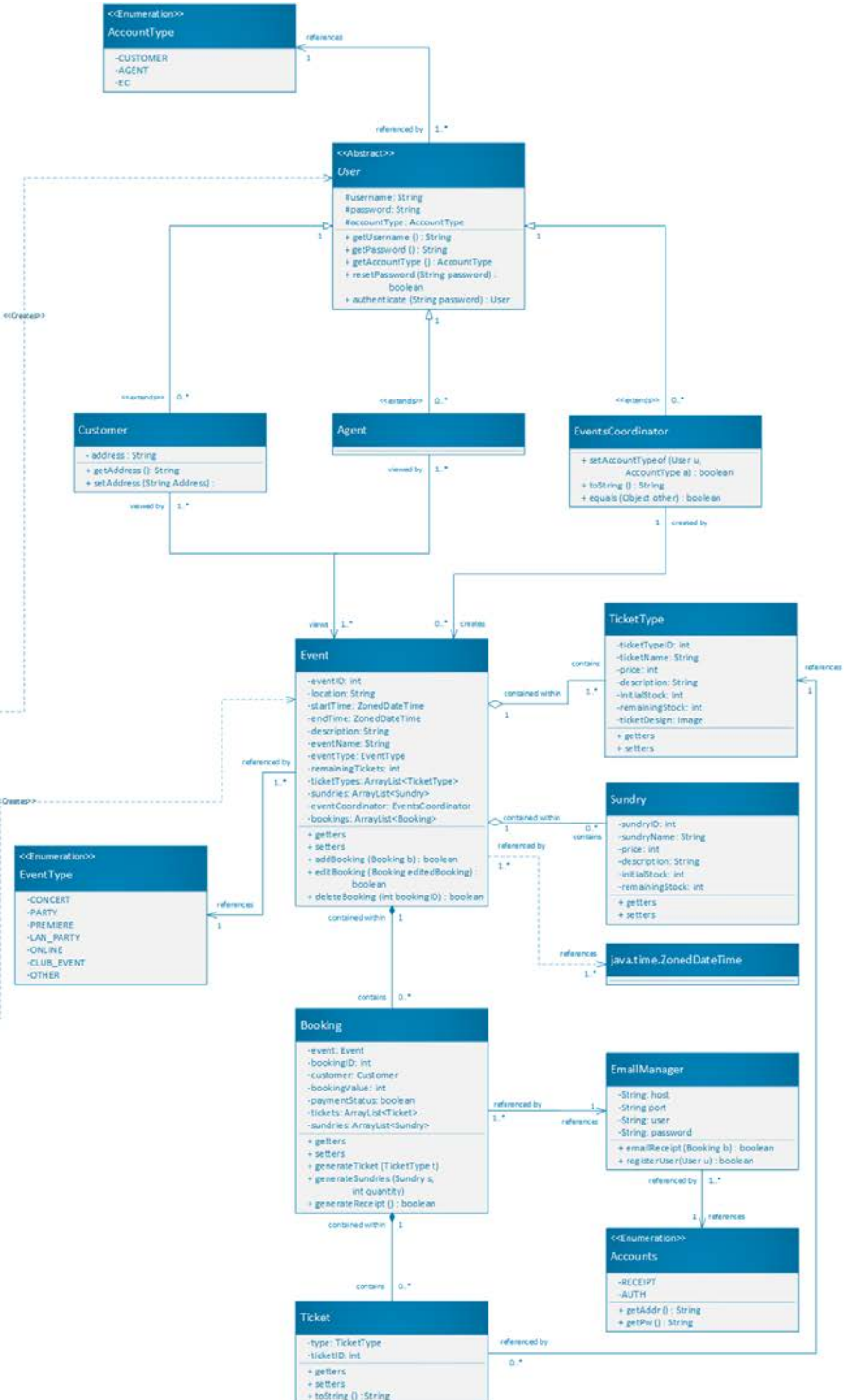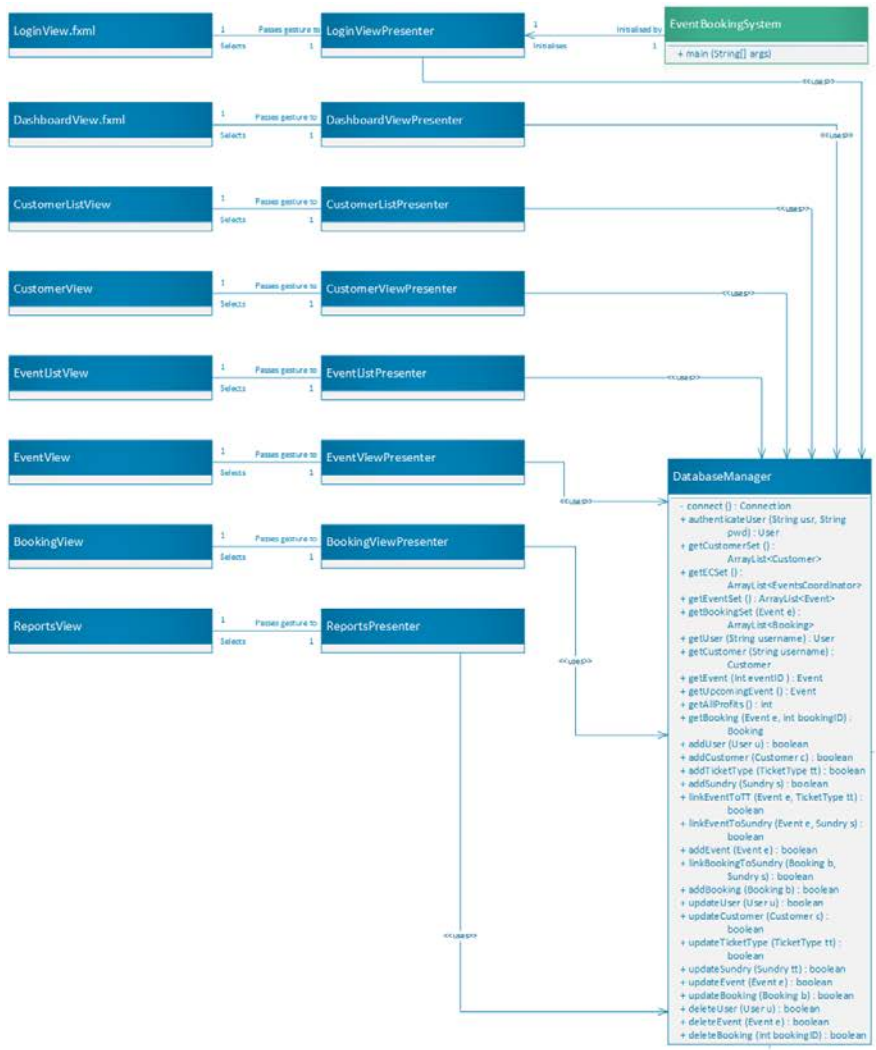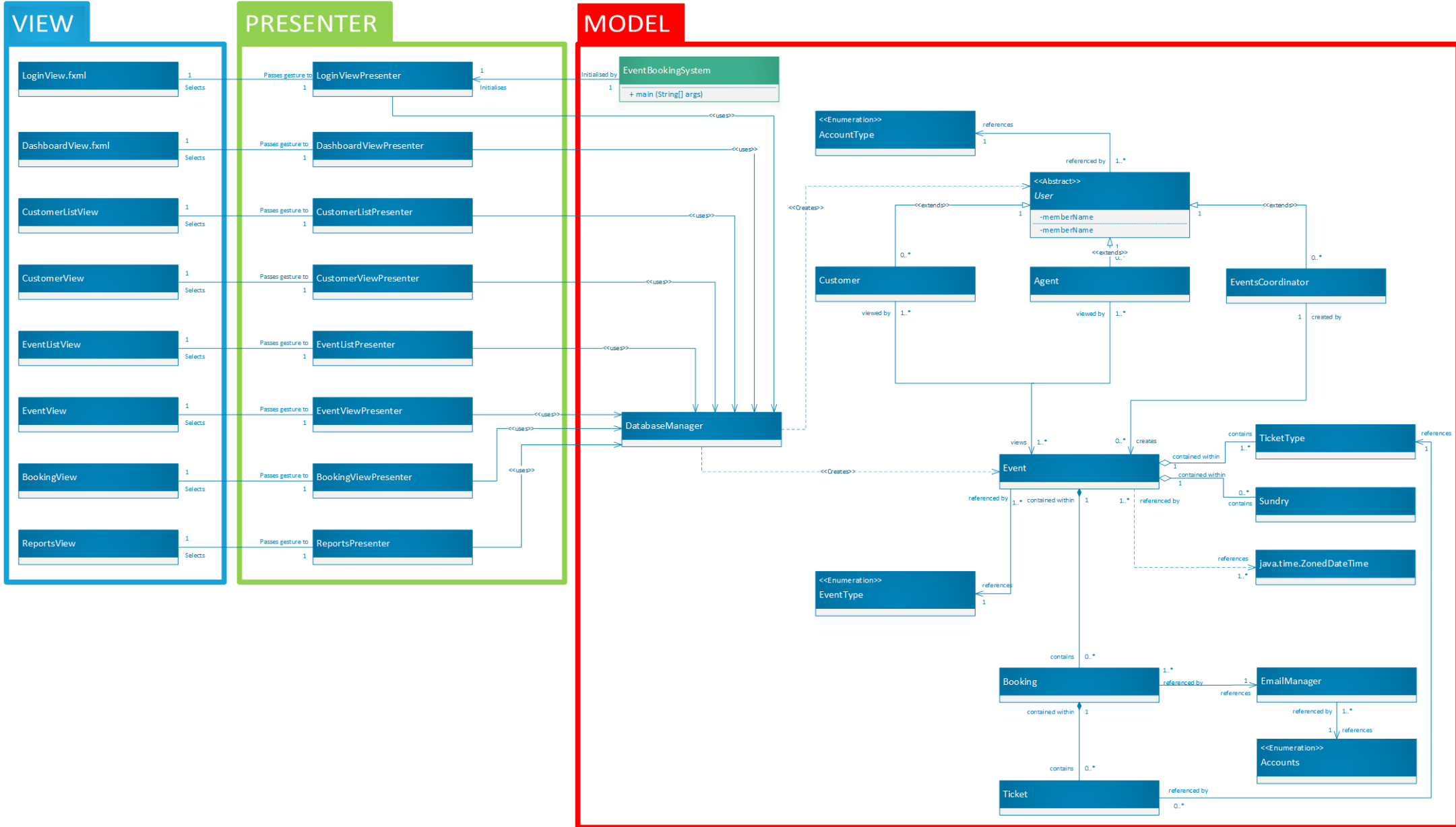- #accountType: AccountType
- + getUsername () : String
- + getPassword () : String
- + getAccountType () : AccountType
- + resetPassword (String password) : boolean
- + authenticate (String password) : User

Customer
- - address : String
- + getAddress () : String
- + setAddress (String Address) :

Agent

EventsCoordinator
- + setAccountTypeof (User u, AccountType at) : boolean
- + toString () : String
- + equals (Object other) : boolean

DatabaseManager
- - connect () : Connection
- + authenticateUser (String usr, String pwd) : User
- + getCustomerSet () : ArrayList<Customer>
- + getECSet () : ArrayList<EventsCoordinator>
- + getEventSet () : ArrayList<Event>
- + getBookingSet (Event e) : ArrayList<Booking>
- + getUser (String username) : User
- + getCustomer (String username) : Customer
- + getEvent (int eventID) : Event
- + getUpcomingEvent () : Event
- + getAllProfits () : int
- + getBooking (Event e, int bookingID) : Booking
- + addUser (User u) : boolean
- + addCustomer (Customer c) : boolean
- + addTicketType (TicketType tt) : boolean
- + addSundry (Sundry s) : boolean
- + linkEventToTT (Event e, TicketType tt) : boolean
- + linkEventToSundry (Event e, Sundry s) : boolean
- + addEvent (Event e) : boolean
- + linkBookingToSundry (Booking b, Sundry s) : boolean
- + addBooking (Booking b) : boolean
- + updateUser (User u) : boolean
- + updateCustomer (Customer c) : boolean
- + updateTicketType (TicketType tt) : boolean
- + updateSundry (Sundry tt) : boolean
- + updateEvent (Event e) : boolean
- + updateBooking (Booking b) : boolean
- + deleteUser (User u) : boolean
- + deleteEvent (Event e) : boolean
- + deleteBooking (int bookingID) : boolean

Event
- -eventID: int
- -location: String
- -startTime: ZonedDateTime
- -endTime: ZonedDateTime
- -description: String
- -eventName: String
- -eventType: EventType
- -remainingTickets: int
- -ticketTypes: ArrayList<TicketType>
- -sundries: ArrayList<Sundry>
- -eventCoordinator: EventsCoordinator
- -bookings: ArrayList<Booking>
- + getters
- + setters
- + addBooking (Booking b) : boolean
- + editBooking (Booking editedBooking) : boolean
- + deleteBooking (int bookingID) : boolean

TicketType
- -ticketTypeID: int
- -ticketName: String
- -price: int
- -description: String
- -initialStock: int
- -remainingStock: int
- -ticketDesign: Image
- + getters
- + setters

Sundry
- -sundryID: int
- -sundryName: String
- -price: int
- -description: String
- -initialStock: int
- -remainingStock: int
- + getters
- + setters

java.time.ZonedDateTime

<<Enumeration>> EventType
- -CONCERT
- -PARTY
- -PREMIERE
- -LAN_PARTY
- -ONLINE
- -CLUB_EVENT
- -OTHER

Booking
- -event: Event
- -bookingID: int
- -customer: Customer
- -bookingValue: int
- -paymentStatus: boolean
- -tickets: ArrayList<Ticket>
- -sundries: ArrayList<Sundry>
- + getters
- + setters
- + generateTicket (TicketType t)
- + generateSundries (Sundry s, int quantity)
- + generateReceipt () : boolean

EmailManager
- -String: host
- -String port
- -String: user
- -String: password
- + emailReceipt (Booking b) : boolean
- + registerUser (User u) : boolean

<<Enumeration>> Accounts
- -RECEIPT
- -AUTH
- + getAddr () : String
- + getPw () : String

Ticket
- -type: TicketType
- -ticketID: int
- + getters
- + setters
- + toString () : String

4.2 MVP Architectural Diagram

5. Initial Implementation

The initial implementation of the application was developed in the IntelliJ IDE using Java (specifically JavaFX) and SQLite. This was because it is the language that the development team is most familiar with, and has the added benefit of being platform agnostic. The IDE offers an extensive suite of quality of life features, including a database manager that is simple to set up. The Model View Presenter architecture ensures separation between Model and View which limited complexity and unnecessary coupling. It provides a logical framework upon which to build the design, and JavaFX lends itself readily to that arrangement.

5.1 Login Screen

This screen checks the input data against user data stored in the database and validates it. In addition, it sets the current user of the system which affects available functionality based upon the user's access level.
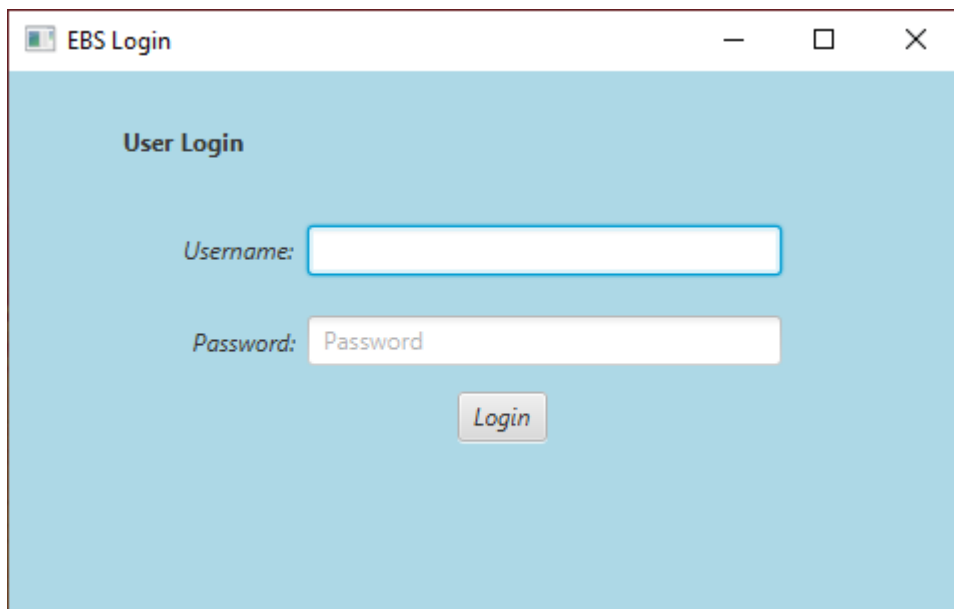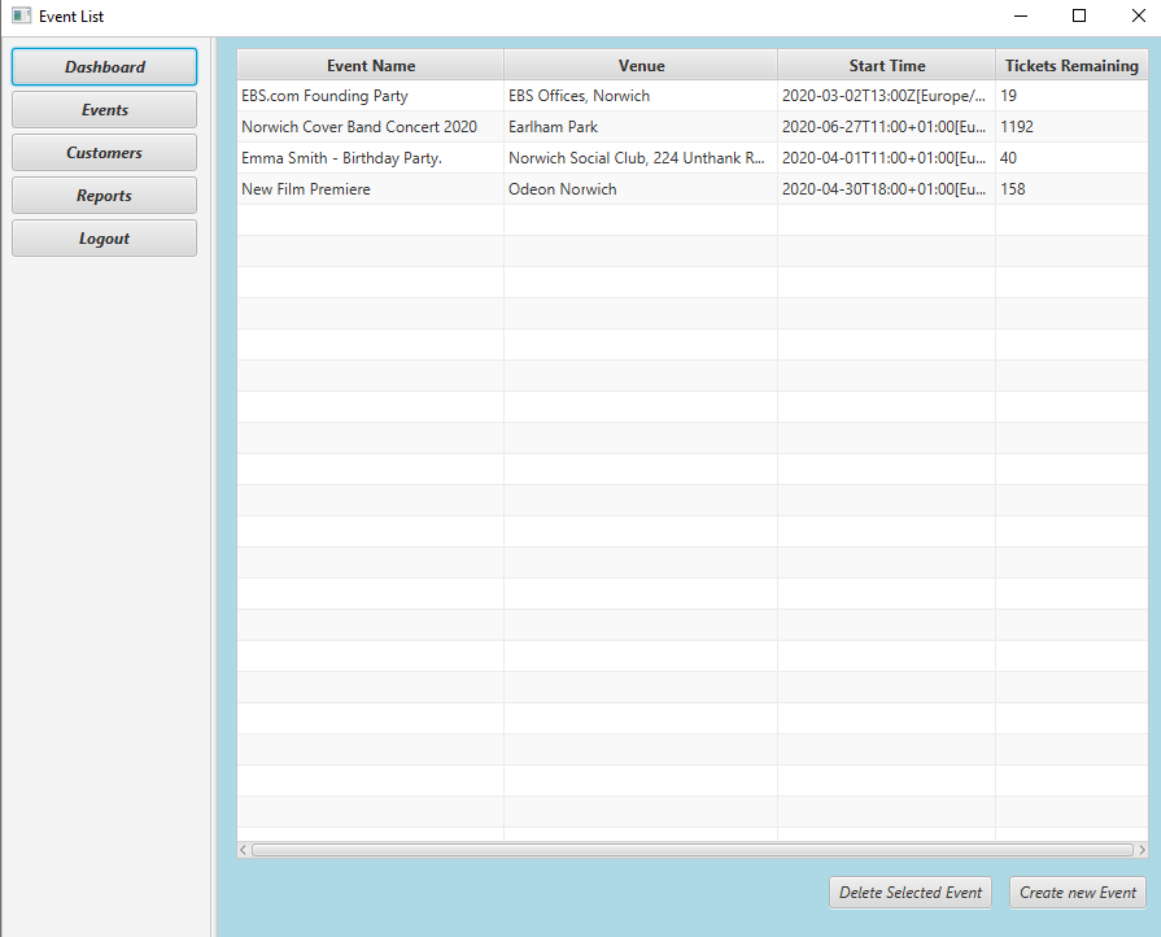
5.2 Dashboard

The dashboard page presents the soonest upcoming event (as this is likely the most relevant to the user), and offers a button to quickly make a booking on that event. It also presents the total profits (calculated from the value of all the bookings in the database), and a pie chart breakdown of profits by event.
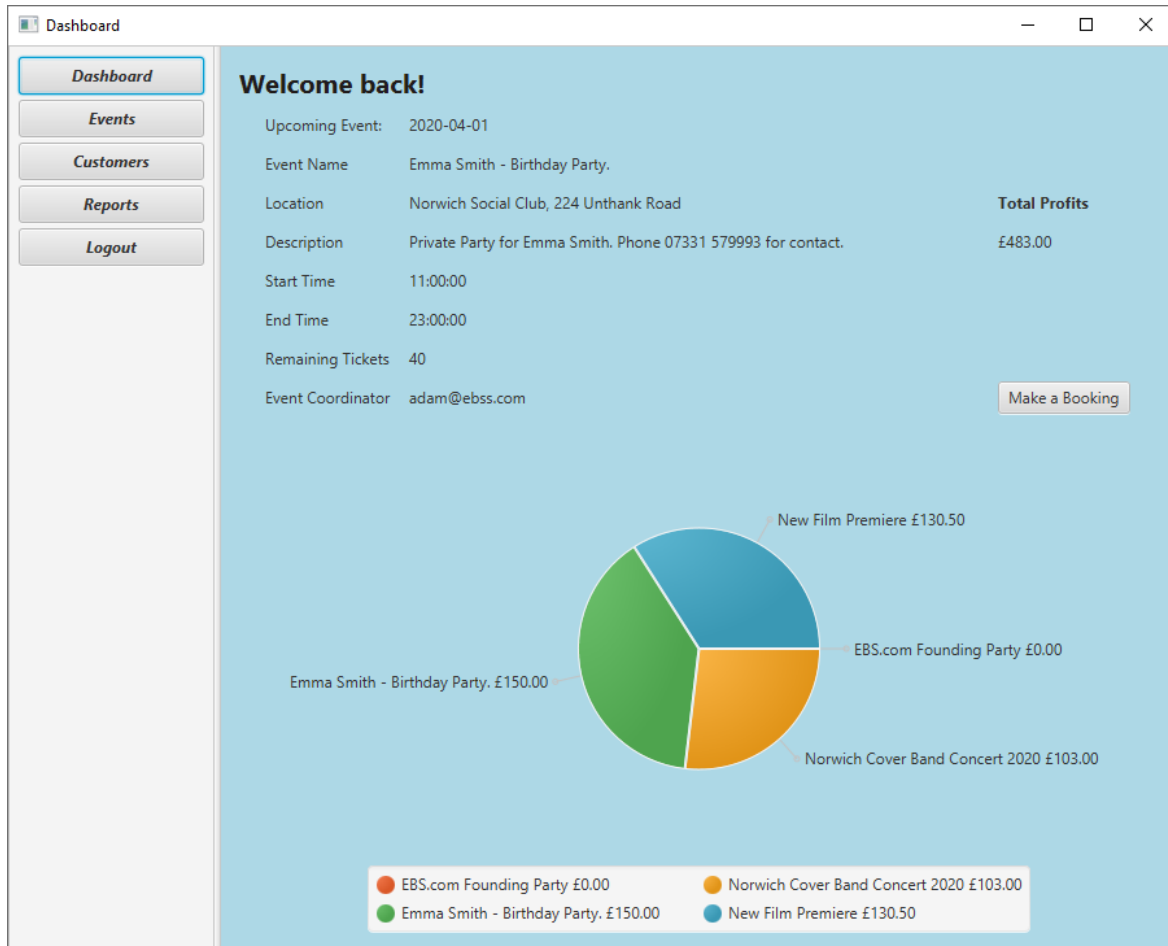
5.3 Events List Screen

This screen shows a table of all events stored in the database. The objects presented in this list have several null elements in order to limit calls to the database when populating the table. Double clicking on an event will lead to a more detailed view. Note that the 'Create new Event' and 'Delete Selected Event' buttons are not available for an Agent.

	Event List			— □ ✕
Dashboard	**Event Name**	**Venue**	**Start Time**	**Tickets Remaining**
Events	EBS.com Founding Party	EBS Offices, Norwich	2020-03-02T13:00Z[Europe/...	19
Customers	Norwich Cover Band Concert 2020	Earlham Park	2020-06-27T11:00+01:00[Eu...	1192
Reports	Emma Smith - Birthday Party.	Norwich Social Club, 224 Unthank R...	2020-04-01T11:00+01:00[Eu...	40
Logout	New Film Premiere	Odeon Norwich	2020-04-30T18:00+01:00[Eu...	158
			Delete Selected Event	Create new Event

5.4 Event Add/Edit/View screen

This screen offers a detailed view of an event. If creating a new event, the fields will be blank, otherwise they are populated with data from the selected event. The Event Type and Event Coordinator choice fields are likewise populated with data from the database. Note that an Agent has no ability to modify an event, and the buttons on this page are not available to them.

5.5 Bookings List Screen

This screen shows a list of all bookings attached to the selected event. Both Event Coordinator and Agent can create and delete bookings here.

5.6 New Booking Screen

This screen allows an Agent or Event Coordinator to take a customer's booking. They can select a pre-existing customer from the list if their data already exists in the database or add a new one. The 'quantity' field in the table is editable and items selected here will add to the Total Booking Value. When the 'Save Booking' button is pressed, the system generates an email and sends it to the selected customer. In addition, the 'Resend receipt' button will resend that email.

5.7 Customer List Screen

This screen presents a list of customers held in the database. Both Event Coordinators and Agents can create and delete new customers. Double clicking on a customer will lead to an editable view that includes the customer's password.

	Customer List	— □ ✕

	Username	Address
Dashboard	alice@hottmael1.com	123 Fake St, Norwich, NR1 1AA
Events	belinda@geemeail.com	43 Dereham Road, Norwich, NR6 7DS
Customers	catherine@aeyoel.co.uk	12 Unthank Road, Norwich, NR2 5RQ
Reports	dot@aeyoel.com	133 Tuddenham Rd, Ipswich, IP4 3SL
Logout	emily@fremal.ru	130 Lynn Road, King's Lynn, Norfolk, PE30 4PB
	m.siddons@uea.ac.uk	127 Earlham Green Lane, Norwich, Norfolk, NR5 8HF

Delete Selected Customer   Create new Customer

5.8 Customer Add/Edit/View Screen

This screen offers the ability to edit and save a customer. It also allows you to view their password.

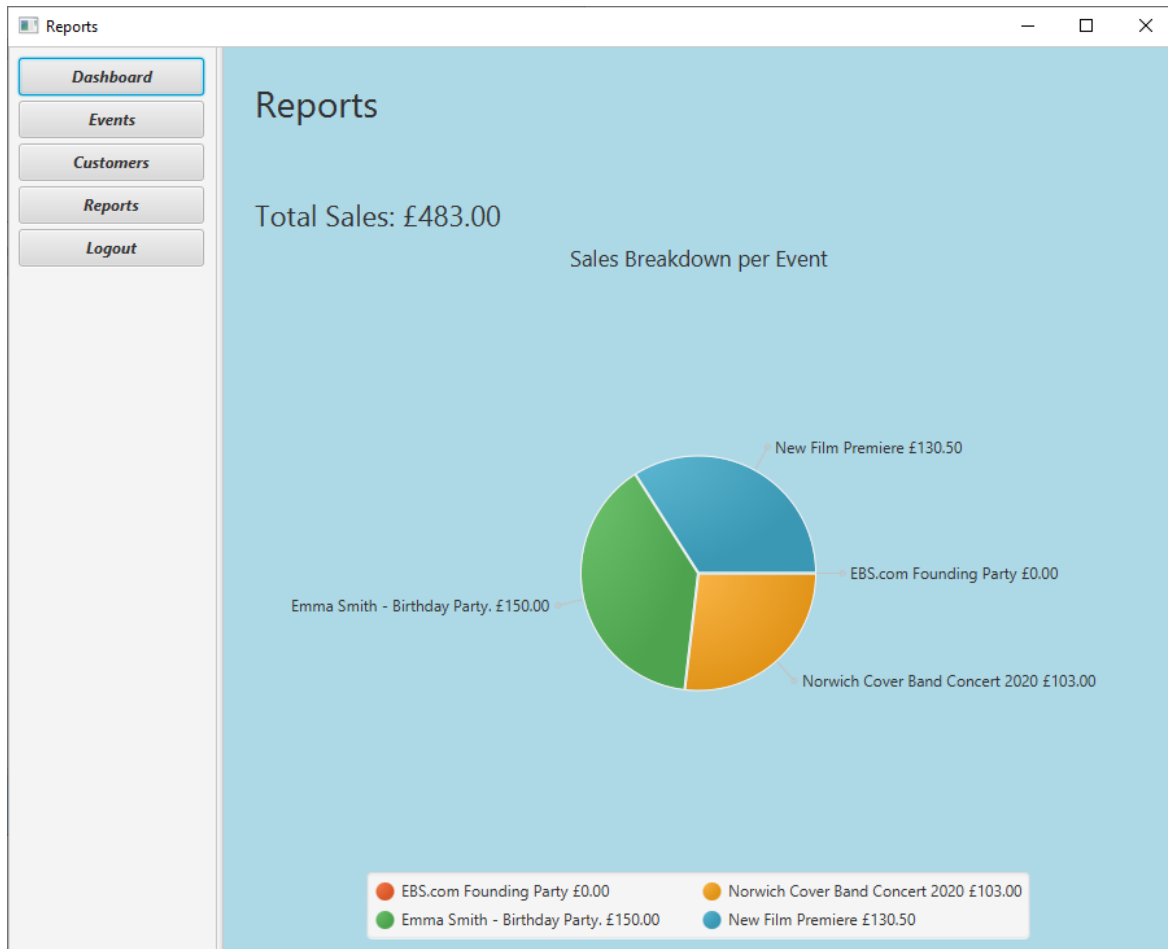## 5.9 Reports Screen

In this iteration of the system, the reports screen is very simple, and only offers a duplicate of the information available on the Dashboard. However, all this information is still pulled directly from the database and will update as bookings and events are made.

6. Stage 2 Preparation

Stage 2 of the project will be developed in PyCharm, using Python and the Django framework. This allows the MVP design of the current system to be easily ported over, as Django has a very similar built in functionality.

One of the major improvements that will be made in Stage 2 is the ability for customers to create their own accounts and make their own bookings through a web app. To this end, a registration page will need to be created. This will allow for a much more versatile system that can be integrated with marketing websites.

Stage 2 will include much more robust type checking for all input fields, and more informative error messages. In addition, data will be stored and transmitted in a much more secure way, especially sensitive data such as passwords (which will probably utilise an Encryption Manager class).

Stage 2 will offer some form of payment processing within the system, likely provided by a third-party API e.g. Sage, PayPal.

Currently, the system makes no distinction between upcoming events and those that have already happened. The Stage 2 system will have an 'archive' section into which events will be browsable once they have concluded. There will also be a search functionality available for events, bookings and customers.

Stage 2 will take into consideration ethical concerns such as what kind of customer information is being stored in the database, and for how long. The system will try to avoid handling payment information directly for the same reason.

There will be some design flexibility in the tickets in Stage 2. The Event Coordinator should be able to design their tickets to reflect the event and provide more information to customers. In addition, tickets will be printable so they can be handed out at the door or sent to the customer.

Look and feel will be improved all around. Elements will scale correctly, and all data will be formatted in a way that makes it easy to read. There will likely be changes to how some data is stored (such as addresses) and what is stored (such as people's names and date of last login).

Finally, with the experience gained from development of Stage 1, Stage 2 will have many logical and efficiency improvements in the backend regarding how the Presenters communicate with the Model, and how data is presented in the View.

Appendix

Electronic Booking System Database– ERD (Crows Foot Diagram)