

Table of Contents

1. Systems Request – Event Management System	4
1.1 Business Needs	4
1.2 Business Requirements	4
1.3 Business Value	4
1.4 Special Issues or Constraints.....	4
2. Detailed Requirements Definition	5
3. Functional Models	7
3.1 Package Diagram	7
3.2 Use case Diagrams	8
3.4 Activity Diagrams.....	26
4. Structural Models	28
4.1 Class Diagram.....	28
5. Behavioural Models	29
5.1 Object Interaction Modelling	29
5.2 Object State Models.....	30

1. Systems Request – Event Management System

Project Sponsor: Alan Fekete (UniUnion CEO)

Project Owner: Matthew Sladescu (Business Champion, Head of Business Unit)

Business Subject Matter Expert: Matthew Sladescu (Business Liaison)

1.1 Business Needs

- 1) Increase general student participation for events hosted by university clubs and societies.
- 2) Allow all university clubs and societies the opportunity to promote and manage their events.
- 3) Increase the efficiency and efficacy of the event management and booking process.
- 4) Reduce the costs associated with manual booking and event management processes.
- 5) Reduce costs associated with the provision of funds for development and maintenance of separate event management systems for individual clubs or societies.

1.2 Business Requirements

- 1) Customers should be able make bookings for events that they are interested in.
- 2) Event hosts should be able to register, promote and manage their events.
- 3) Event hosts should be able to manage customer bookings for their events.

1.3 Business Value

- 1) Increase participation at events hosted by clubs and societies by 50%.
- 2) All registered clubs, societies and other event hosts are able to promote and manage their events at university based venues.
- 3) The cost associated with building a new event management system every time one is requested by an event host is eliminated.
- 4) The cost associated with manually keeping track of event bookings and associated booking payments are reduced by 80%.
- 5) An increase in efficiency and efficacy which will boost the image of UniUnion and streamline operations.

1.4 Special Issues or Constraints

- 1) The Analysis Phase for this project must be completed by week 6.
- 2) The CTO and stakeholders will only be available in limited ways (email and questions through a dedicated thread on the eLearning discussion board).

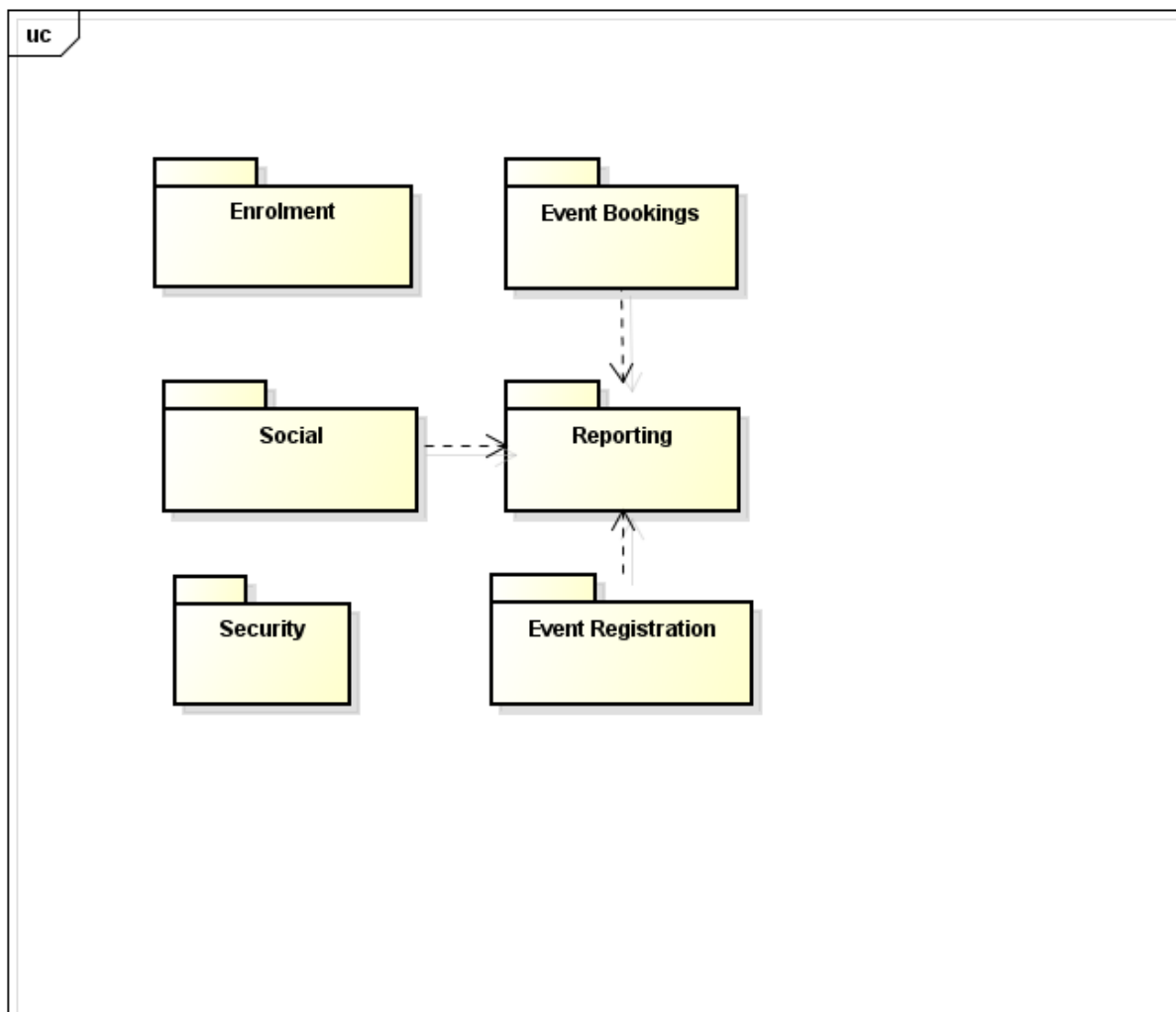
2. Detailed Requirements Definition

Priority	Business Req. ID	Functional Requirement
H	2	<ol style="list-style-type: none"> 1. Event Host Enrolment <ol style="list-style-type: none"> I. The system will allow event hosts like union clubs and societies to enrol into the system. II. Event hosts will be able to enter at least the following details at enrolment: event host name, ABN, and email. III. Event hosts will be able to update the details mentioned in II at any time. IV. During enrolment, the event host will be prompted to enter a nominated account into which booking revenues can be deposited and from which refunds can be issued. V. The event host should be able to change their nominated account details (see 1II) at any time. VI. Event hosts shall be able to make changes to any enrolment details at any time.
H	2	<ol style="list-style-type: none"> 2. Event Registration <ol style="list-style-type: none"> I. Event hosts will be able to register new events. II. Events will be linked to the event host making the event registration. III. Event hosts should be able to query for suitable venues based on event needs such as: expected turnout, required equipment, event time and duration. IV. Event hosts should be able to select a suitable venue from the results in 2 III V. Event hosts will be able to specify at least the following details: event name, description, venue, time, duration, and cost. VI. Event hosts will be able to specify if booking cancellations are allowed for this event, and the deadline for such booking cancellations. VII. Event hosts will be able to specify if payment for this event can be collected at the venue.
H	1	<ol style="list-style-type: none"> 3. Customer Enrolment <ol style="list-style-type: none"> I. Customers that wish to make bookings for events hosted by clubs and societies should be able to enrol into the system. II. Customers will need to nominate an account that can be used to pay for event bookings, and into which any refunds can be deposited. III. Customers shall be able to make changes to their enrolment details at any time.
L	1	<ol style="list-style-type: none"> 4. Event News <ol style="list-style-type: none"> I. Customers shall be able to select a list of event hosts that they are interested in, and elect to receive weekly news about associated events. <ol style="list-style-type: none"> i. Customers shall be able to modify their preferences for Functional Requirement 4 I at any time.
H	1, 2	<ol style="list-style-type: none"> 5. Event Queries <ol style="list-style-type: none"> I. All users shall be able to query for events by event name, description, event host name, venue, time, duration, or cost. II. Event hosts shall be able to see a list of events that they have registered, and query them by name, description, venue, time, duration and cost.
H	1, 2, 3	<ol style="list-style-type: none"> 6. Event Details <ol style="list-style-type: none"> I. All users shall be able to see the details for a listed event, including the event name, description, venue, time, duration and cost. II. Event hosts will be able to see booking lists for each event they have created. Each booking in the booking list will display the booking name, customer contact details, and booking state. The booking state may be open, flagged, or closed: <ol style="list-style-type: none"> i. A booking is open if the booking event occurs in the future. ii. A booking is flagged if payment has not been received and the event has finished. iii. A booking is closed if the event has been cancelled by the customer

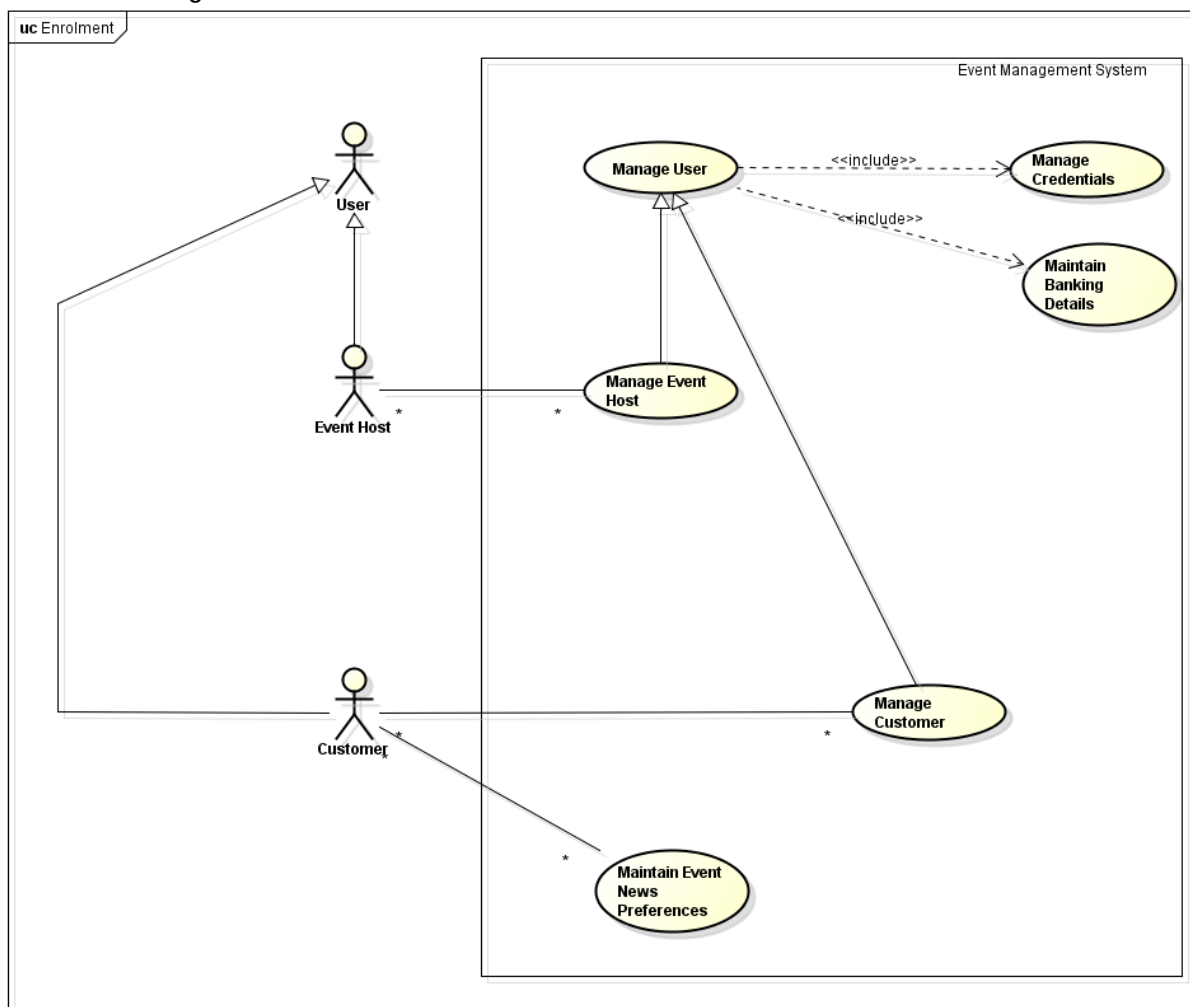
		<p>or event host; or if the event has finished and payment has been received.</p> <ul style="list-style-type: none"> III. Event hosts will be able to query their event booking lists by booking name, booking customer contact details, and booking state. IV. Event hosts shall be able to change the state of a booking from flagged to closed, for booking linked to their events.
H	1	<p>7. Event Bookings</p> <ul style="list-style-type: none"> I. Enrolled customers shall be able to make bookings for any listed event. II. Customers will be asked to pay immediately by credit card, or at the venue (if permitted by the event host (see Functional Requirement 2VII)). III. If customers choose to pay immediately, payment is deducted from their nominated account (Functional Requirement 3II) IV. Enrolled customers shall be able to cancel their bookings if the event host permits booking cancellations, and the booking cancellation deadline defined in functional requirement 2 VI has not passed.
H	1, 2,3	<p>8. Event Updates</p> <ul style="list-style-type: none"> I. Event hosts shall be able to update event information. II. Event hosts shall be able to cancel events that they have registered. When this occurs, customers who have bookings for these events shall have their booking costs refunded, and be notified of the event cancellation by email. III. Customers with bookings to any events that have updates will be notified via email. Customers shall be eligible to cancel their bookings when there is an update to event details, regardless of event host cancellation preferences.
L	1	<p>9. Event Recommendations</p> <ul style="list-style-type: none"> I. Enrolled customers shall be able to recommend an event to friends.
Non – Functional Requirement		
H	1.	<p>Interoperability</p> <ul style="list-style-type: none"> I. The system will allow enrolled event hosts to import and export event and booking information using a public data exchange API, after logging in with appropriate event host credentials.
H	2.	<p>Operational</p> <ul style="list-style-type: none"> I. Event hosts and Customers should be able to access the system from any desktop web browser. II. Customers should be able to access the system from the default iPhone, android, and Windows Phone web browsers.
H	3.	<p>Security</p> <ul style="list-style-type: none"> I. Only customers with appropriate login credentials shall be able to see and update their personal details and their personal booking details with the exception of 3III. II. Only event hosts with appropriate login credentials shall be able to see and update their details, view booking lists associated with their events, update details for events that they have registered,. III. Event hosts shall be able to view customer postal address, name, and email address for customers that have made bookings for their events. Event hosts shall also be able to see details for all bookings related to their events.
H	4.	<p>Performance</p> <ul style="list-style-type: none"> I. Response times for any type of request should be less than 2 seconds.

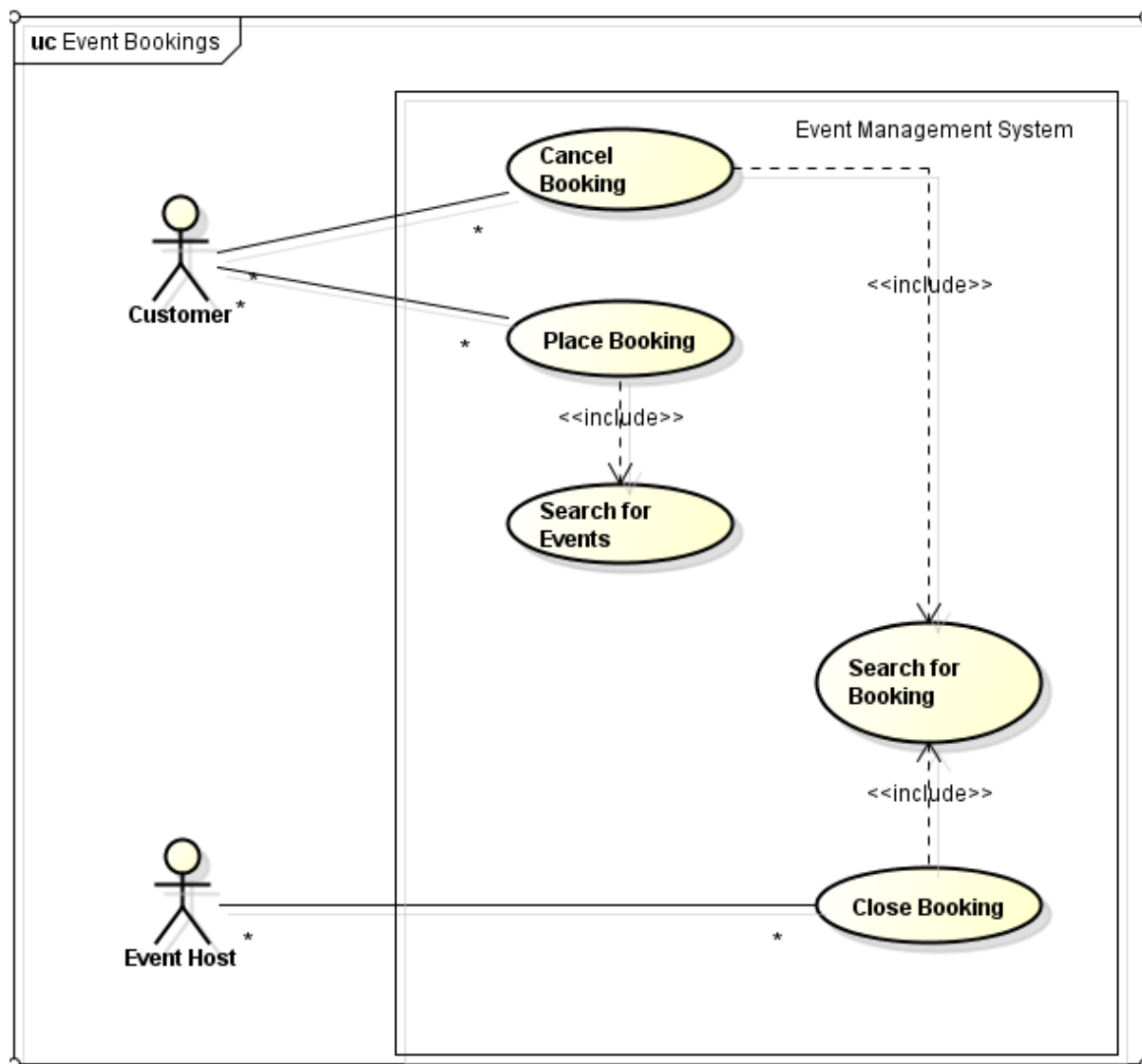
3. Functional Models

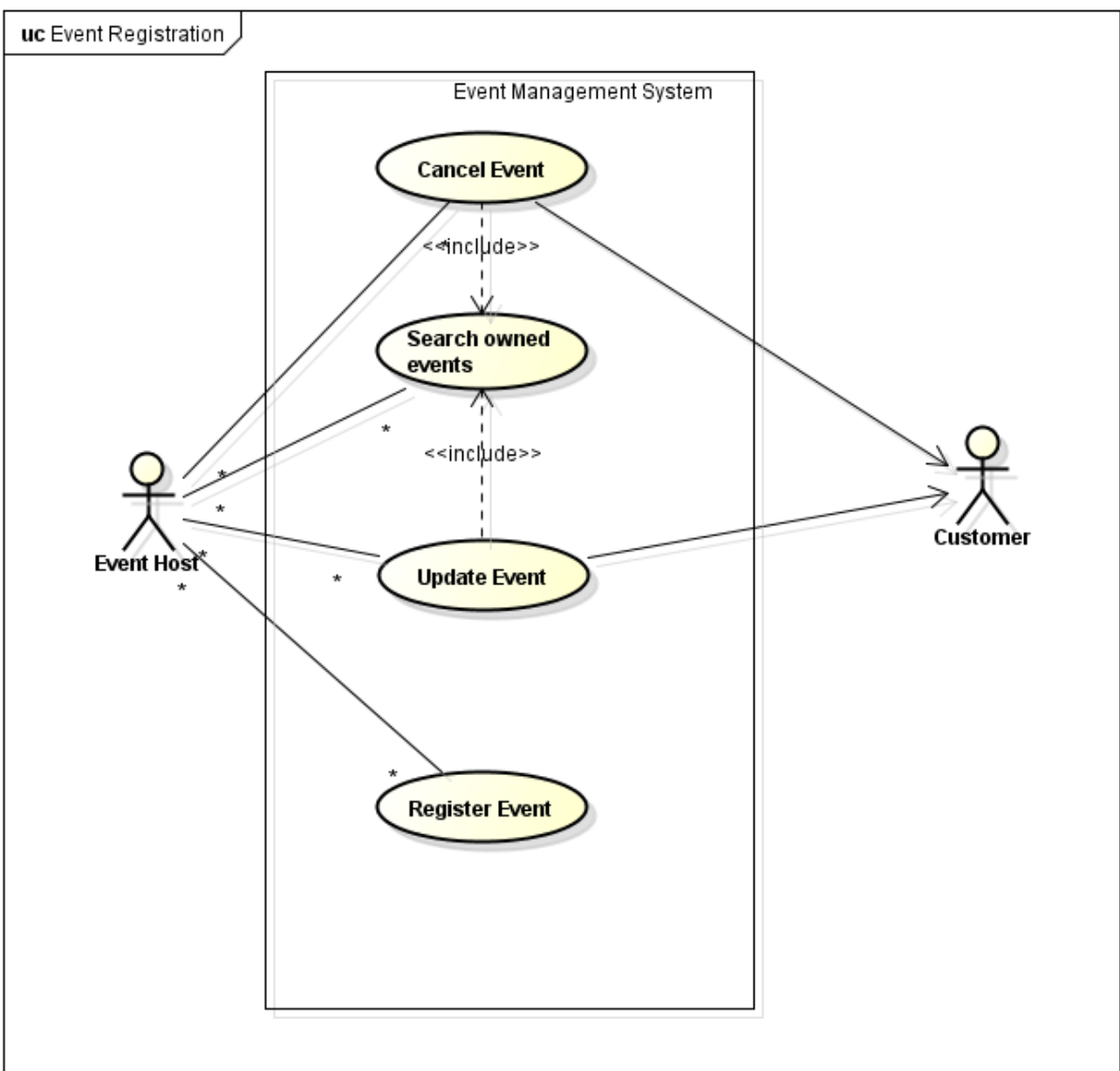
3.1 Package Diagram

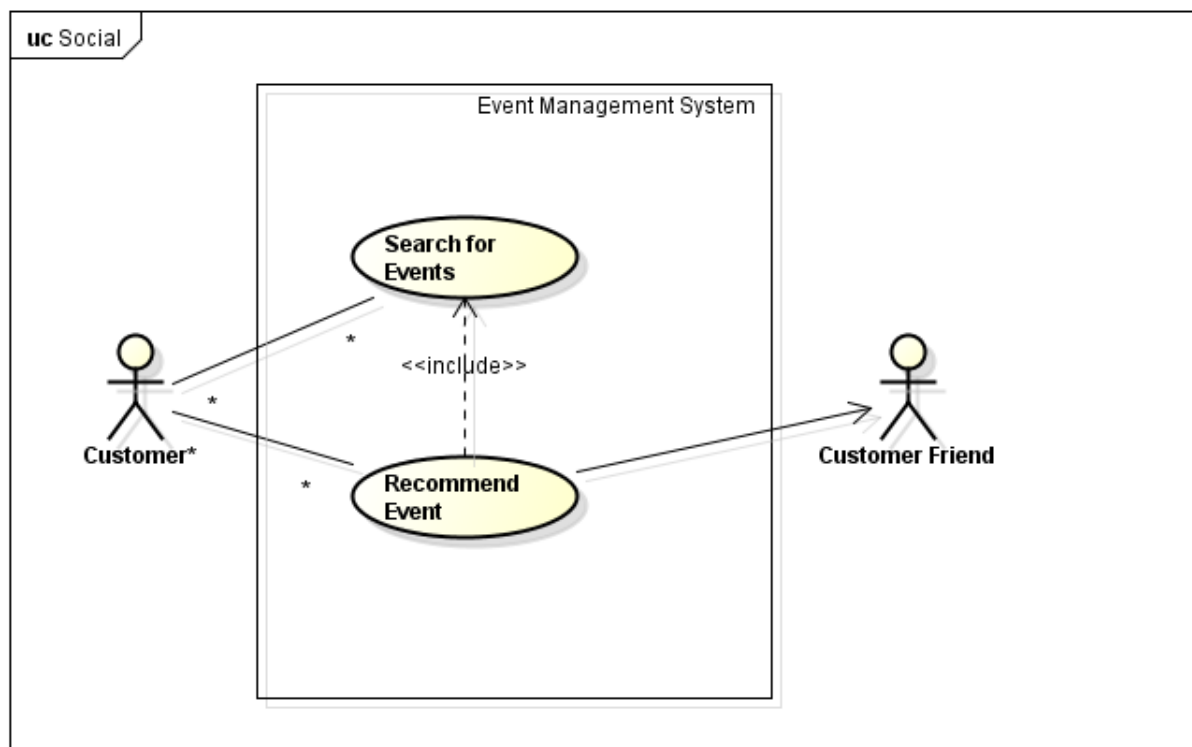
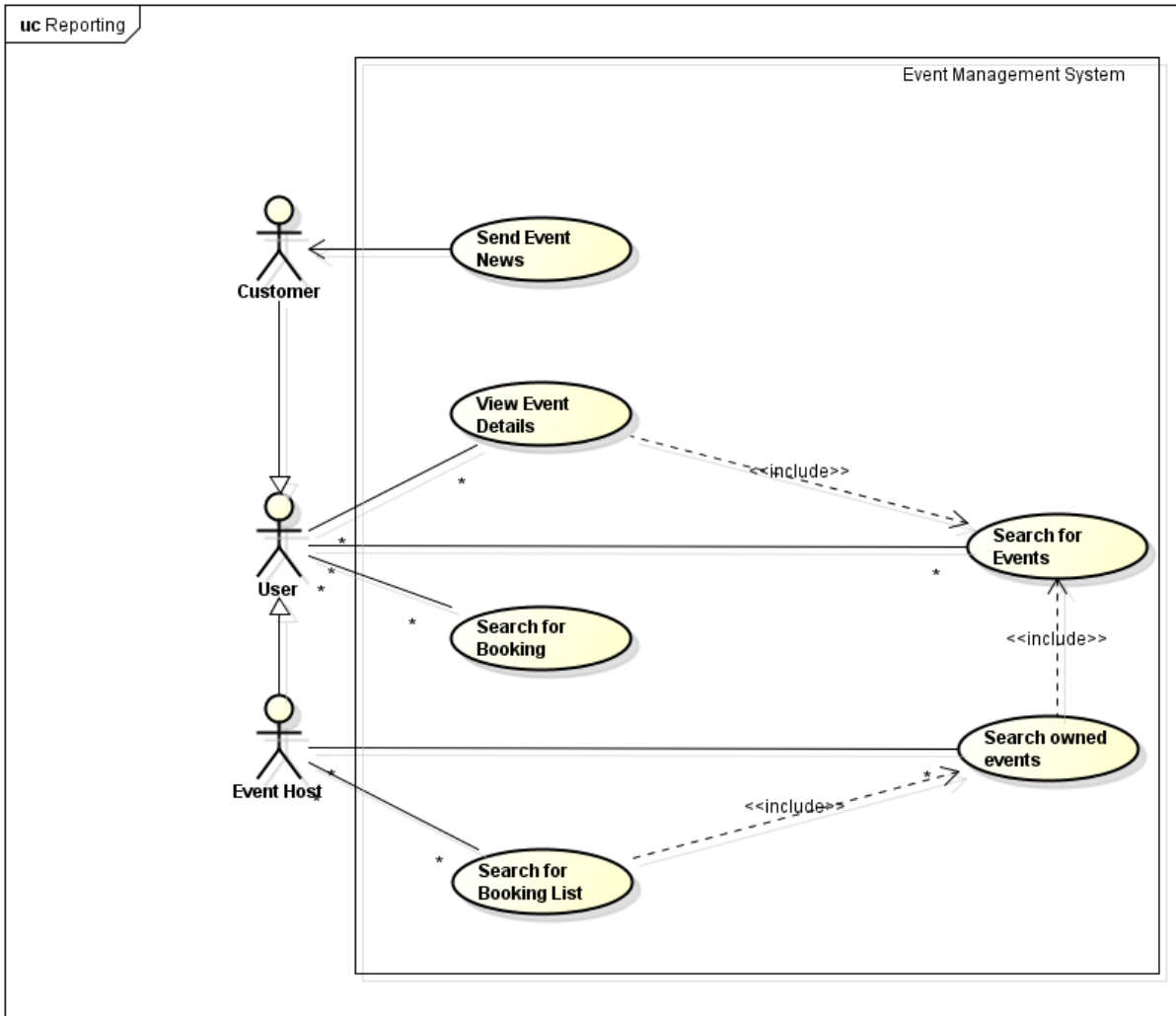


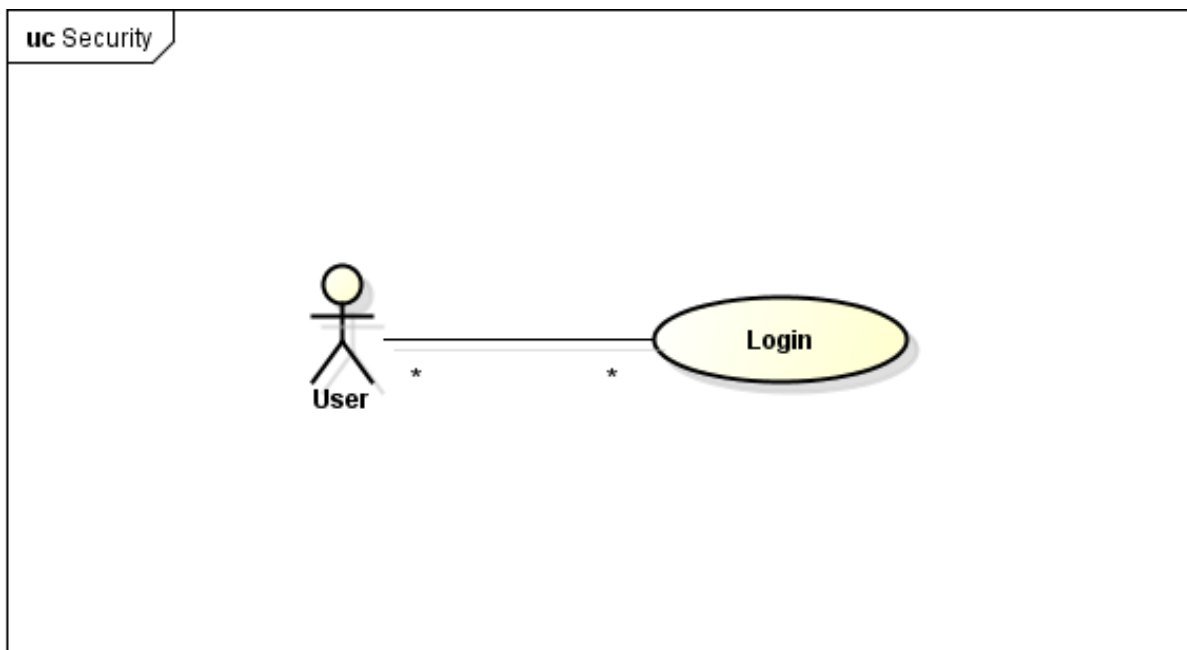
3.2 Use case Diagrams











3.3 Use Case Descriptions

Use Case Name: Manage User		ID: 1	Importance Level: H
Primary Actor: User		Use Case Type: Detail, Essential	
Stakeholders and Interests: User – wants to enrol or update their details in the event management system.			
Brief Description: This use case describes the steps necessary for managing a user in the system.			
Trigger: User requests enrolment or update of their details Type: External			
Relationships:			
Association:			
Include: Manage Credentials, Maintain Banking Details			
Normal Flow of Events:			
<ol style="list-style-type: none"> 1. User requests updates to their details 2. If the user requests enrolment, Subflow S-1 User Enrolment is followed If the user requests updates, Subflow S-2 User Details Update is followed 3. User executes manage credentials use case. 4. User executes maintain banking details use case 5. System notifies user that his details have been recorded. 			
Sub-flows:			
S-1 User Enrolment:			
<ol style="list-style-type: none"> 1. The system asks the user to fill in user information in order to enrol in the system 			

<p>2. The user confirms once he is done entering and changing details</p> <p>S-2 User Details Update:</p> <ol style="list-style-type: none"> 1. The user provides his/her username and password to log in the system. 2. The system validates that the user exists in the database. 3. The user updates the information details 4. The user confirms the changes once he is done
<p>Alternate/Exceptional Flows:</p> <p>3a-1. The system notifies the user the system failure.</p> <p>3a-2. The user requires an explanation.</p> <p>3a-3. The system gives the explanation.</p> <p>3a-4. The user modifies the information and return back in step 2.</p> <p>5a-1. The user aborts the enrolment/updates.</p> <p>5a-2. The system notifies the user that the user information has not been changed/created</p>

Use Case Name: Manage Credentials	ID: 2	Importance Level: H
Primary Actor: User	Use Case Type: Detail, Essential	
Stakeholders and Interests: User – wants credentials to be able to authenticate / authorize with the system		
Brief Description: This use case describes the steps necessary for providing users with the necessary credentials for authenticating / authorizing with the system.		
Trigger: User requests credential management Type: External		
Relationships: Association: User		
Normal Flow of Events: <ol style="list-style-type: none"> 1. User requests credential management 2. If the user already has credentials: <ul style="list-style-type: none"> If the user desires a change in credentials: <ul style="list-style-type: none"> Subflow S-1 Credential Assignment is followed If the user does not already have credentials: <ul style="list-style-type: none"> Subflow S-1 Credential Assignment is followed 3. If the user's credentials were updated <ul style="list-style-type: none"> The system confirms a successful credential update 		
Sub-flows: S-1 Credential Assignment: <ol style="list-style-type: none"> 1. The user log in the system. 2. The system validates the user in the database. 3. The user updates credential in their accounts. 		
Alternate/Exceptional Flows: <p>2.S-1. 2a user credential validation fails, back to step S-1.1</p> <p>3a-1. The user aborts the change in credentials.</p> <p>3a-2. The system notifies the user that the credential has not been changed.</p>		

Use Case Name: Maintain banking details	ID: 3	Importance Level: H
Primary Actor: User	Use Case Type: Detail, Essential	
Stakeholders and Interests: User – wants to register or update their banking details.		
Brief Description: This use case describes the steps necessary for registering new banking details or updating the banking details for a customer or event host.		
Trigger: User enrolls into the system, or requests an update to their banking details Type: External		
Relationships: Association: User		
Normal Flow of Events: <ol style="list-style-type: none"> 1. User requests first time entry of their banking details (at enrolment) or an update to their banking details. 2. If the User already has banking details: If the User desires a change in banking details: Subflow S-1 Set banking details is followed If the User does not already have banking details: Subflow S-1 Set banking details is followed 3. If banking details were set The system notifies the User of a successful update to their banking details. 		
Sub-flows: S-1 Set Banking Details: <ol style="list-style-type: none"> 1. The user log in the system. 2. The system validates the user in the database. 3. The user updates banking details in their accounts. 		
Alternate/Exceptional Flows: 2.S-1 2a. user credential validation fails, back to step S-1.1 3a-1. The system notifies the user of the banking details update failure. 3a-2. The user requires an explanation. 3a-3. The system gives the explanation. 3a-4. The user modifies the information and return back to step 2		

Use Case Name: Register Event	ID: 4	Importance Level: H
Primary Actor: Event Host	Use Case Type: Detail, Essential	
Stakeholders and Interests: Event Host – wants to register a new event for which customers can make bookings.		
Brief Description: This use case describes the steps necessary for registering a new event.		
Trigger: Event Host requests registration of a new event Type: External		
Relationships: Association: Event Host		
Normal Flow of Events: <ol style="list-style-type: none"> 1. The event host requests registration for a new event. 2. The event host enters details about the event: name, description, time, duration, cost, and required equipment. 		

<ol style="list-style-type: none"> 3. The event host queries the system about suitable venues based on the details entered in step 2. 4. System returns a list of suitable venues to the event host. 5. The event host chooses from a selection of venues listed in step 4. 6. The event host specifies if customers are able to cancel their bookings for this event, and specifies the deadline for such booking cancellations. 7. The event host specifies if payment for this event can be collected at the venue, and specifies the location for such payment. 8. System notifies event host of successful event creation.
Sub-flows:
Alternate/Exceptional Flows:
4a. System returns no suitable venues, back to step 3
5a. event host resubmit a new search query, back to step 2
8a The event host is notified the error if they didn't fill in the compulsory information(name, time, duration, cost)
8b The event host is notified the error if the time and venues are conflict with existing event.

Use Case Name: Update Event	ID: 5	Importance Level: H
Primary Actor: Event Host	Use Case Type: Detail, Essential	
Stakeholders and Interests:		
Event Host – wants to update the details for an event which it has created Customer – wants to know whenever there are updates for events to which they have bookings for.		
Brief Description: This use case describes the steps necessary for updating event details		
Trigger: Event Host requests update for an event which it created Type: External		
Relationships:		
Association: Event Host, Customer Includes: Search owned events		
Normal Flow of Events:		
<ol style="list-style-type: none"> 1. The event host requests an update to one of the events that they created 2. The event host executes Search owned events with event name, description, venue, time, duration and/or cost. 3. The event host selects the desired event from step 2, and makes the desired changes. 4. The system notifies all Customers who hold bookings for this event of the changes that have been made. 5. System notifies the event host of a successful update to the event. 		
Sub-flows:		
Alternate/Exceptional Flows:		
3a. System notify host the system could not find the queried event, back to step 2		
4a. System notify host a update failure, back to step3		
4b. The event host aborts updates.		

Use Case Name: Maintain Event News Preferences	ID: 6	Importance Level: H
Primary Actor: Customer	Use Case Type: Detail, Essential	

<p>Stakeholders and Interests:</p> <p>Customer – wants to receive notifications about upcoming events which they might find interesting.</p>
<p>Brief Description: This use case describes the steps necessary for customers to register for event news.</p>
<p>Trigger: Customer Type: External</p>
<p>Relationships: Association: Customer</p>
<p>Normal Flow of Events:</p> <ol style="list-style-type: none"> 1. The customer requests to register or un-register for weekly emails about events. If the customer chooses to register for weekly emails: The customer selects a collection of event hosts, for which they would like to receive news about their associated events. System notifies customer of registration success If the customer chooses to un-register for weekly emails: Customer selects event-hosts from which to unregister for weekly emails System notifies customer of un-registration success
<p>Sub-flows:</p>
<p>Alternate/Exceptional Flows:</p> <ol style="list-style-type: none"> 1a. Customer is notified that he/she is not registered to any event, can not un-register. 1b. Customer is notified that subscribe is failure due to system error, ask customer to do it again or abort

Use Case Name: Send Event News	ID: 7	Importance Level: H
Primary Actor: Triggered By System	Use Case Type: Detail, Essential	
<p>Stakeholders and Interests:</p> <p>Customer – wants to receive notifications about upcoming events which they might find interesting.</p>		
<p>Brief Description: This use case describes the steps necessary for sending weekly event news to customers.</p>		
<p>Trigger: Beginning of Week (Every week the event management system sends out news to customers about upcoming events which customers might be interested in).</p>		
<p>Type: Temporal</p>		
<p>Relationships: Association: Customer</p>		
<p>Normal Flow of Events:</p> <ol style="list-style-type: none"> 1. System retrieves a list of all the customers that have registered to receive weekly emails about upcoming events. 2. System brings up one of the customer records from step 1 (Let this customer be called Customer X). 3. The system brings up a list of the event hosts Customer X has registered to receive notifications about. 4. The system brings up the list of events for one of the event hosts from step 3 (Let this event host be called Event Host Y). 5. If Event Host Y has events occurring in the next month 		

- i) System creates a new email if one has not already been created, which is addressed to Customer X.
- ii) System records Event Host Y's upcoming events on email that was created in step 5 i)
- 6. System repeats steps 4 and 5 for all event hosts that Customer X is registered to.
- 7. System sends email to Customer X regarding upcoming events for the event hosts that they are subscribed to.
- 8. System repeats steps 2 to 7 for each customer that has registered for weekly emails.

Sub-flows:
 ii. If the event host has no events occurring in the next month, no events will be noted down on an email.

Alternate/Exceptional Flows:
 1a. system displays that no customer registered for event news.
 1b. system notifies user a retrieve failure, ask user either to try again, or abort.
 5iia. System notifies user that an email creation failed.
 7a. system notifies user sending email to customer X has failed, back to step 2.

Use Case Name: Place Booking	ID: 10	Importance Level: H
Primary Actor: Customer	Use Case Type: Detail, Essential	
Stakeholders and Interests: Customer – wants to make a booking for an event.		
Brief Description: This use case describes the steps that Customers need to follow to make a booking for an event.		
Trigger: Customer requests a booking for an event.		
Type: External		
Relationships: Association: Customer Include: Search for Event		
Normal Flow of Events: <ol style="list-style-type: none"> 1. Customer requests booking. 2. Customer executes the Search for Event use case with event key words that they are interested in. 3. Customer selects an event that was returned in step 2. 4. System displays available seating for event selected in step 3. 5. Customer selects from available seating displayed in step 4. 6. System displays booking price, and selected seating 7. Customer confirms seating and booking price 8. If the event has a pay at venue option: Subflow S-1 Payment options is performed If the event does not have a pay at venue option: Subflow S-2 Process Direct Debit Payment is performed 9. System reserves seating for customer, and updates available seating for the event. 10. System provides customer with a unique booking number, and the booking is confirmed with the customer. 		

<p>Sub-flows:</p> <p>Subflow S-1 Payment Options:</p> <ol style="list-style-type: none"> 1. The customer asks to pay at venue. 2. The system confirms with a receipt marked with pay at venue. <p>Subflow S-2 Process Direct Debit Payment:</p> <ol style="list-style-type: none"> 1. Payment is deducted from his/her nominated account. 2. The system confirms with a receipt marked with paid amount.
<p>Alternate/Exceptional Flows:</p> <p>3a-1. No result match the customer requirement.</p> <p>3a-2. System suggests searching for Event with alternative key words that they are interested in back in step 2.</p> <p>3a-3. Customer selects an event or they abort it.</p> <p>4a. No available/not enough seating, system suggests customer back to step 2.</p> <p>5a. Customer gives up booking.</p> <p>7a. customer either wants to change seating (back to step 4), or abort booking</p> <p>8.S-2 1a. System notifies user failure of payment (e.g. not enough funds, failed transaction), back to step 6.</p> <p>9a. reservation of seating is failed, ask user to try again (back to step 6, omitting payment step 8)</p>

Use Case Name: Cancel Booking	ID: 11	Importance Level: H
Primary Actor : Customer	Use Case Type: Detail, Essential	
Stakeholders and Interests:		
Customer – wants to cancel a booking for an event.		
Brief Description: This use case describes the steps that Customers need to follow to cancel a booking for an event.		
Trigger: Customer requests cancellation for an event booking.		
Type: External		
Relationships:		
Association: Customer		
Includes: Search for Booking		
Normal Flow of Events:		
<ol style="list-style-type: none"> 1. Customer requests cancellation for a booking. 2. Customer executes Search for booking use case using the event name or booking number. 3. Customer selects the event that they wish to cancel from results returned in step 2. 4. System cancels the booking. 5. System refunds the booking fee to the Customer's account. 6. System makes Seats under the cancelled booking available again. 7. System notifies customer of successful booking cancellation. 		
Sub-flows:		
Alternate/Exceptional flows:		
3a-1. System notifies the search failure.		
3a-2. Customer requests the failure explanation.		

3a-3. System gives explanation.
 3a-4. Customer modifies the search information and back to Step 2.

4a-1. Cancellation failure.
 4a-2. Customer requests explanation.
 4a-3. System gives the explanation that the cancellation deadline has passed (or no cancellation option for this event) and return back to step 2.

Use Case Name: Recommend an Event	ID: 12	Importance Level: H
Primary Actor: Customer	Use Case Type: Detail, Essential	
Stakeholders and Interests:		
Customer – wants to recommend an event to friends		
Brief Description: This use case describes the steps that Customers need to follow in order to recommend an event to their friends.		
Trigger: Customer requests send recommendation for event function.		
Type: External		
Relationships:		
Association: Customer, Customer Friend		
Includes: Search for Events		
Normal Flow of Events:		
<ol style="list-style-type: none"> 1. Customer requests to send a recommendation for an event to friends 2. Customer executes search for events use case 3. Customer selects the events he wants to recommend from the results returned in 2. 4. If Customer wishes to recommend more events: System Returns customer to step 2 5. Customer lists the addresses of friends he wishes to forward the recommendations to 6. Customer enters a recommendation message 7. Customer requests that the recommendation is sent 8. System sends recommendation including recommendation message and events selected in step 3 to the list of addresses from step 5. 9. System gives confirmation to customer that the recommendation was sent. 		
Sub-flows:		
Alternate/Exceptional flows		
<p>2a-1. System requires customer log in the system for recommending events to friends. 2a-2. Customer goes back to step 2 and log in the system.</p> <p>2b-1. Customer did not register to the system. 2b-2. System requires customer enrol to the system for recommending events to friends. 2a-3 Customer enrolment use case executed.</p>		

2a-4. Recommend an event user cased executed.

3a-1. No result match the customer requirement.
 3a-2. System suggests searching for Event with alternative key words that they are interested in back in step 2.

5a. system returns an empty friend list, system suggests adding friends.

9a. system returns an email sending failure, back to step 7.

Use Case Name: Search for Events		ID: 13	Importance Level: H
Primary Actor: User		Use Case Type: Detail, Essential	
Stakeholders and Interests:			
User- wants to search for events			
Brief Description: This use case describes the steps that Users need to follow in order to search for an event			
Trigger: User requests an event search.			
Type: External			
Relationships:			
Association: User			
Normal Flow of Events:			
<ol style="list-style-type: none"> 1. User requests an event search 2. User provides search keywords to find events with. 3. The system searches the all events using the keywords given in 2. 4. The system returns a list of all matching events, where the name, venue, time, duration, and cost are shown for each match. 			
Sub-flows:			
Alternate/Exceptional flows			
4a-1. No suitable result match the requirements. 4a-2. Back to step 2, ask customer to modify the searching information, or customer aborts it.			

Use Case Name: Search Owned Events		ID: 14	Importance Level: H
Primary Actor: Event Host		Use Case Type: Detail, Essential	
Stakeholders and Interests:			
Event Host- wants to search for own events			
Brief Description: This use case describes the steps that Event Hosts need to follow in order to search within their own events			
Trigger: Event requests an owned event search.			
Type: External			
Relationships:			
Association: Event Host			

Includes: Search for Events
Normal Flow of Events: <ol style="list-style-type: none"> 1. Event Host requests a search within the range of events that were created by them. 2. Event Host provides search keywords to find events with. 3. Event host executes Search for events use case with keywords specified in step 2, as well as keywords to specify that only events with the current event host name should be considered. 4. The system returns the results from search for events use case (executed in step 3) to the user.
Sub-flows:
Alternate/Exceptional flows: 4a-1. No suitable result matches. 4a-2. Ask event host back to step 2, to modify the search keywords or abort

Use Case Name: View Event Details	ID: 16	Importance Level: H
Primary Actor: User	Use Case Type: Detail, Essential	
Stakeholders and Interests: User- wants to view the details for an event		
Brief Description: This use case describes the steps that a user shall follow to view event details		
Trigger: User requests details for an event Type: External		
Relationships: Association: User Includes: Search for Events		
Normal Flow of Events: <ol style="list-style-type: none"> 1. User requests the details for an event. 2. User executes search for events with the keywords for the event he wishes to view details for 3. The user selects the event he wishes to view the details for from the results returned in step 2. 4. The system presents the user with details for the selected event, including name, description, venue, time, duration and cost. 		
Sub-flows:		
Alternate/Exceptional flows: 3a-1. No result matches the searching requirements. 3a-2. System asks to go to step 2 where user modifies the searching information or aborts it.		

Use Case Name: Search for Booking List	ID: 17	Importance Level: H
Primary Actor: Event Host	Use Case Type: Detail, Essential	
Stakeholders and Interests: Event Host- wants to view bookings for an event which they have created		
Brief Description: This use case describes the steps that an event host shall follow to view a list of bookings for an event they have created.		
Trigger: Event Hosts requests booking list for an event Type: External		

<p>Relationships: Association: Event Host Includes: Search for Owned Events</p>
<p>Normal Flow of Events:</p> <ol style="list-style-type: none"> 1. Event host requests booking list for an event. 2. Event Host executes search for owned events with the keywords for the event he wishes to view the booking list for 3. The event host selects the event he wishes to view the booking list for from the results returned in step 2. 4. The system presents the event host with a booking list for the selected event. For each booking in this list, the booking name, customer contact details, and booking state are shown.
<p>Sub-flows:</p> <p>Alternate/Exceptional flows: 3a-1. No result matches the searching requirements. 3a-2. System ask user to go back to step 2, Event Host modifies the searching keywords.</p> <p>4a-1. System requires Event Host log in the system before viewing the details. 4a-2. Event Host login the system with an existing account. 4a-3. Credential is identified and back to step 4.</p> <p>4b. system returns no booking for the event.</p>

Use Case Name: Close Booking	ID: 18	Importance Level: H
Primary Actor: Event Host	Use Case Type: Detail, Essential	
Stakeholders and Interests:		
Event Host– wants to change the state of a booking from flagged to closed		
Brief Description: This use case describes the steps that an event host shall follow to change the state of a booking from flagged to closed.		
Trigger: Event Hosts requests change of state for a booking		
Type: External		
Relationships:		
Association: Event Host		
Includes: Search for Booking		
Normal Flow of Events:		
<ol style="list-style-type: none"> 1. Event host requests change of state for a booking. 2. Event host executes search for booking using a booking number, event name, and/or booking name. 3. The event hosts selects the desired booking from the bookings returned in step 2. 4. If the booking is in the flagged state, <ol style="list-style-type: none"> i. The event hosts requests a change of booking state to closed ii. The system returns confirmation of booking change of state. 		
Sub-flows:		
Alternate/Exceptional flows:		
3a-1. No desired booking is found.		

3a-2. System ask user to go back to step 2. Event host modifies the searching information.

4iia. System notifies booking state change failure, asks user back to step 3.

Use Case Name: Search for Booking	ID: 19	Importance Level: H
Primary Actor: User	Use Case Type: Detail, Essential	
Stakeholders and Interests: User – wants to search for a booking for an event which it has permissions to view.		
Brief Description: This use case describes the steps that a user shall follow to search for a booking for an event which it has permission to view		
Trigger: User requests a booking search Type: External		
Relationships: Association: Event Host		
Normal Flow of Events: <ol style="list-style-type: none"> 1. User requests a booking search 2. System identifies User permissions for viewing bookings from caller identity. 3. User provides search keywords to find bookings with. (This may include event name, booking name, or booking number). 4. The system searches the all bookings that the user has permissions to view using the keywords given in 1. 5. The system returns a list of all matching bookings, where the booking name, event name, and booking number, booking state, and booking contact details are shown. 		
Sub-flows:		
Alternate/Exceptional flows: 3a. user permission is denied. 5a-1. System notifies the user no result match. 5a-2. System suggests user back to step 3, or abort.		

Use Case Name: Cancel an Event	ID: 20	Importance Level: H
Primary Actor: Event Host	Use Case Type: Detail, Essential	
Stakeholders and Interests: Event Host– wants to cancel an event which it has created		
Brief Description: This use case describes the steps that an event host shall follow to cancel an event which it has created		
Trigger: Event Hosts requests event cancellation Type: External		
Relationships: Association: Event Host Includes: Search owned events		
Normal Flow of Events: <ol style="list-style-type: none"> 1. The event host requests an event cancellation 2. The event host executes Search owned events with event name, description, venue, time, duration 		

<p>and/or cost.</p> <ol style="list-style-type: none"> 3. The event host selects the desired event from step 2, and proceeds with cancellation. 4. System refunds all customers with bookings for the cancelled event with their booking fees. 5. System notifies event host of successful event cancellation. 6. The system notifies all Customers who hold bookings for this event of the changes that have been made.
<p>Sub-flows:</p> <p>Alternate/Exceptional flows:</p> <p>3a. No desired event is found and return back to step 2.</p> <p>4a-1. Cancellation failure.</p> <p>4a-2. The event host request explanation.</p> <p>4a-3. The event host is given explanation that they need login before cancelling the event.</p> <p>4a-4. The event host login the system with an existing account.</p> <p>4a-5. System validates the user ID and return back to step 3.</p> <p>5a. system notifies host that refund is not successful for some customers with failure details, suggest host to contact system help desk.</p>

Use Case Name: Manage Event Host	ID: 21	Importance Level: H
Primary Actor: Event Host	Use Case Type: Detail, Essential	
Stakeholders and Interests: Event host – wants to enrol or update their details in the event management system.		
Brief Description: This use case describes the steps necessary for managing an event host in the system.		
Trigger: Event Host requests enrolment or update of their details Type: External		
Relationships: Association: Event Host Inherits: Manage User use case		
Normal Flow of Events:		
<ol style="list-style-type: none"> 1. Event host requests enrolment or updates to their details 2. If the event host requests enrolment, Event host enters their ABN System records the Event Host's ABN If the Event Host requests updates, Event host updates their ABN if desired 3. Event Host executes the normal flow of events in Manage User. 4. System notifies event host of any successful updates to their details. 		
Sub-flows:		
Alternate/Exceptional Flows:		
<p>4a-1. Enrolment or Update failure.</p> <p>4a-2. Event host is told that credentials are not correct.</p> <p>4a-3. System ask to go back to step 3, Event host modifies the information, or aborts it.</p>		

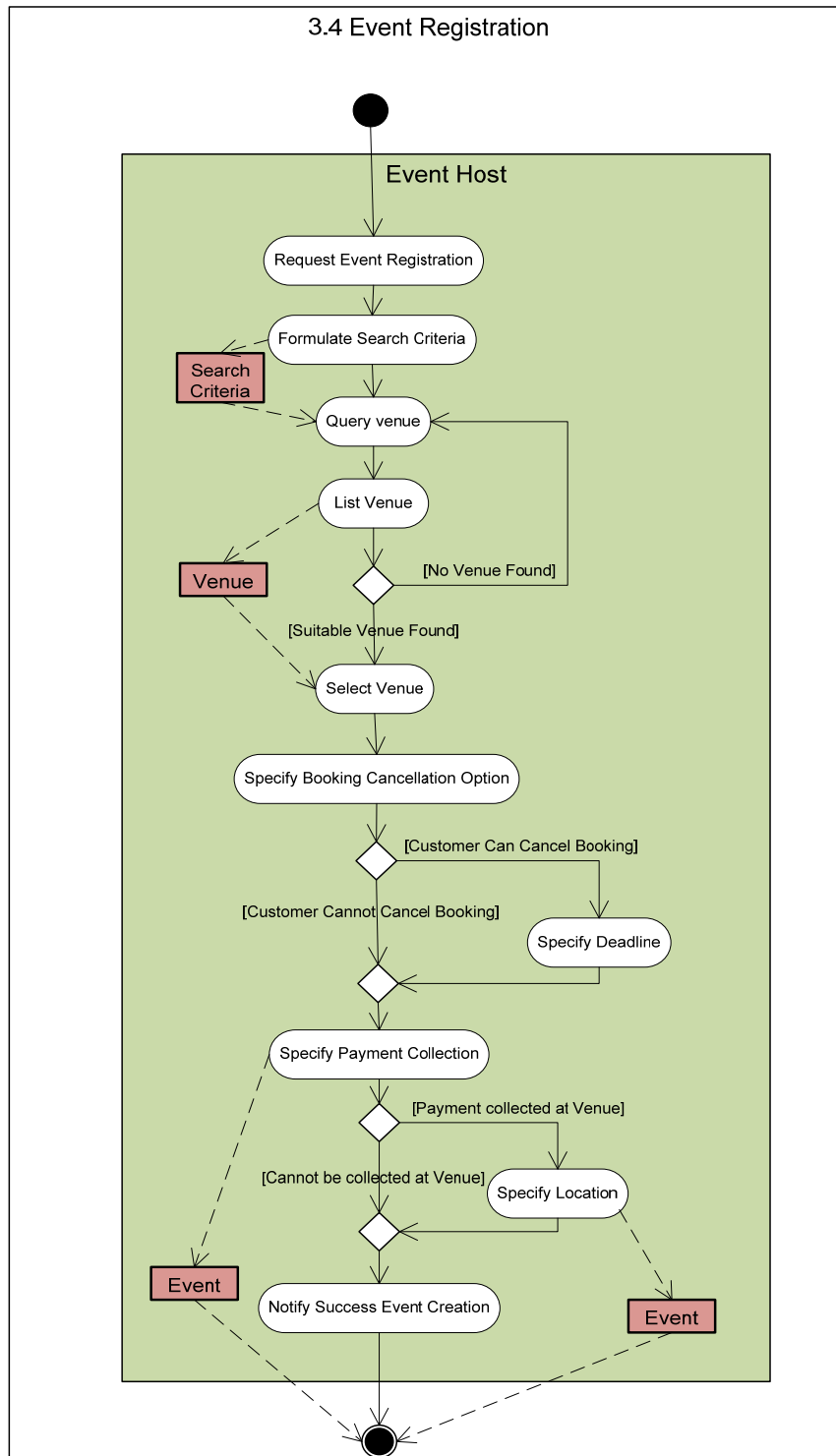
Use Case Name: Manage Customer	ID: 22	Importance Level: H
---------------------------------------	---------------	----------------------------

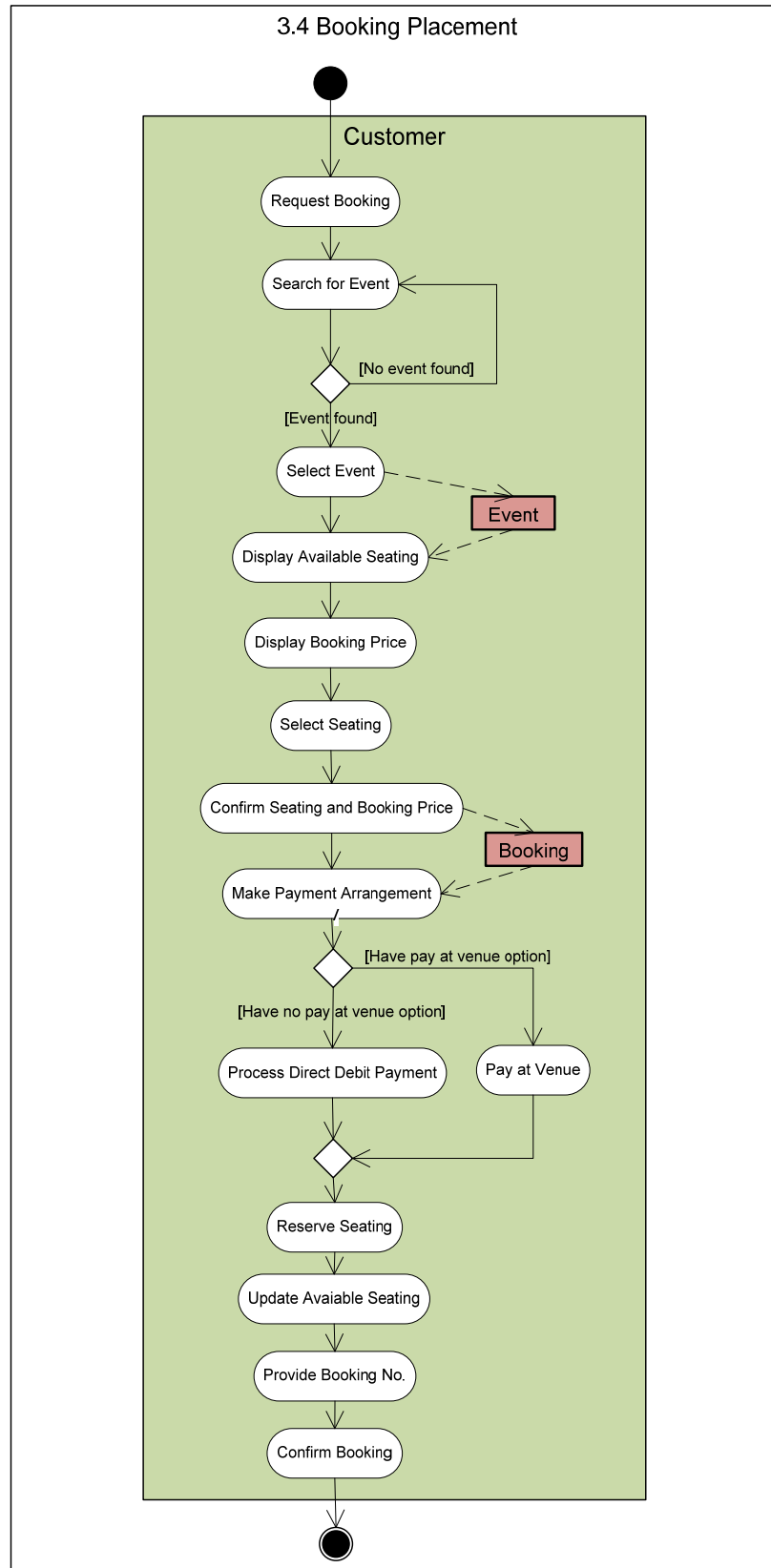
Primary Actor: Customer	Use Case Type: Detail, Essential
Stakeholders and Interests: Customer – wants to enrol or update their details in the event management system.	
Brief Description: This use case describes the steps necessary for managing a Customer in the system.	
Trigger: Customer requests enrolment or update of their details Type: External	
Relationships: Association: Customer Inherits: Manage User use case	
Normal Flow of Events: <ol style="list-style-type: none"> 1. Customer requests enrolment or updates to their details 2. Customer executes the normal flow of events in the manage user use case. 3. System notifies customer of any successful updates to their details 	
Sub-flows:	
Alternate/Exceptional Flows: 3a-1. Enrolment or Update failure. 3a-2. Customer is told that credentials are not correct. 3a-3. System ask user to go back in step 2, Customer modifies the information, or aborts it.	

Use Case Name: Login	ID: 23	Importance Level: H
Primary Actor: User	Use Case Type: Detail, Essential	
Stakeholders and Interests: User – wants to login to the system		
Brief Description: This use case describes the steps necessary for logging in to the system		
Trigger: User requests login Type: External		
Relationships: Association: User		
Normal Flow of Events: <ol style="list-style-type: none"> 1. The user issues a request to log into the system 2. The user is prompted for credentials 3. The user provides username and password credentials to the system 4. The system authenticates the user based on their credentials. 5. The system authorizes the user for access rights. 6. The user is logged into the system. 7. The user is notified of successful logon. 		
Sub-flows:		
Alternate/Exceptional Flows: 3a-1. The user is not an existing user in the database or misses spelled username or password. 3a-2. The user request to enrol (the enrolment use case is executed) or try to log in again, back to step 1. 5a-1. The system notifies the user that their credentials are not match in the database. 5a-2. Back to step 2, The user modifies the username and password credentials.		

3.4 Activity Diagrams

This section describes the workflow for the main business processes in the Univentssystem. This should include at least the event registration and booking placement processes.



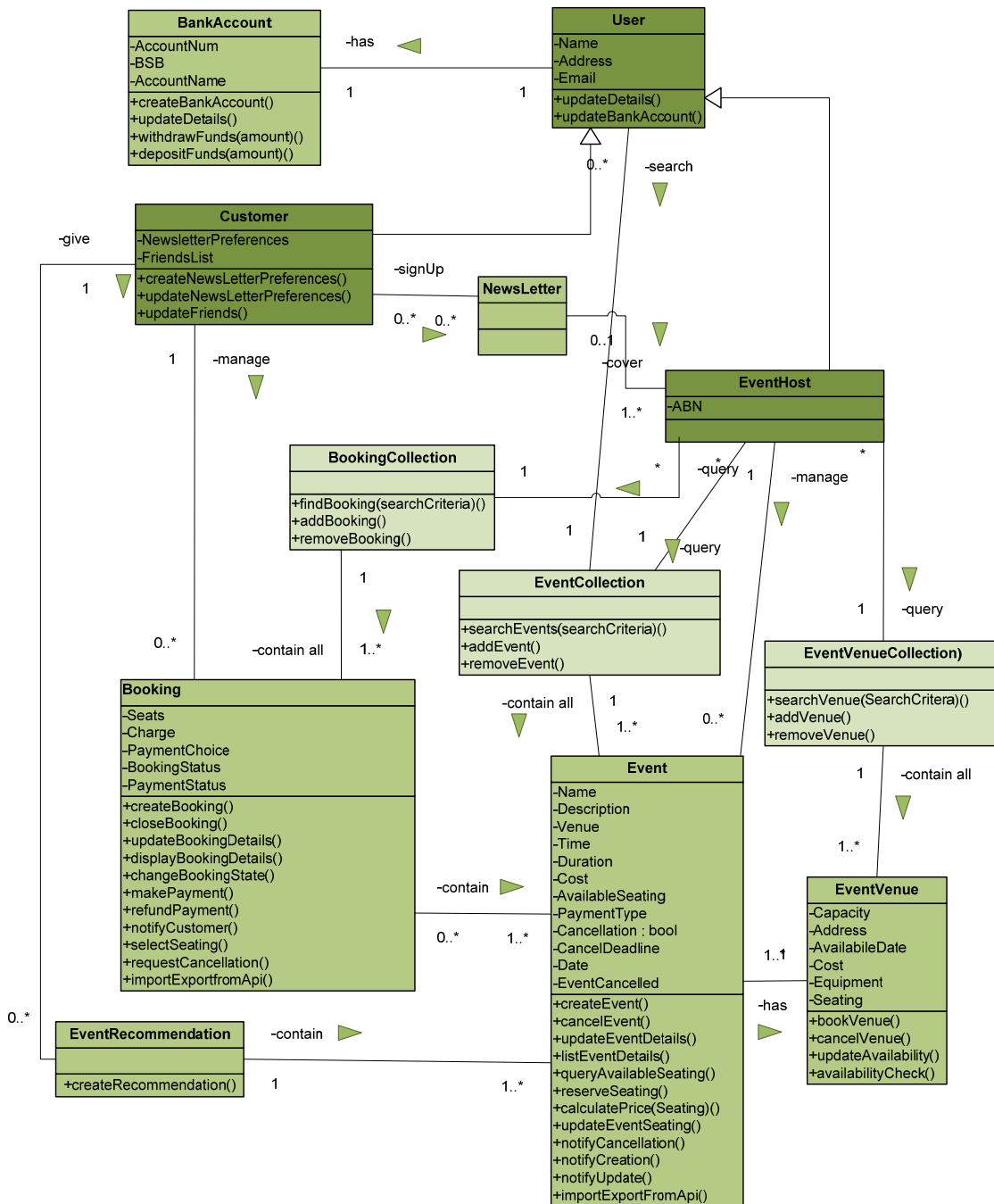


4. Structural Models

4.1 Class Diagram

This section describes the properties, behaviour, and relationships of entities in the Universts domain

4.1 Class Diagram

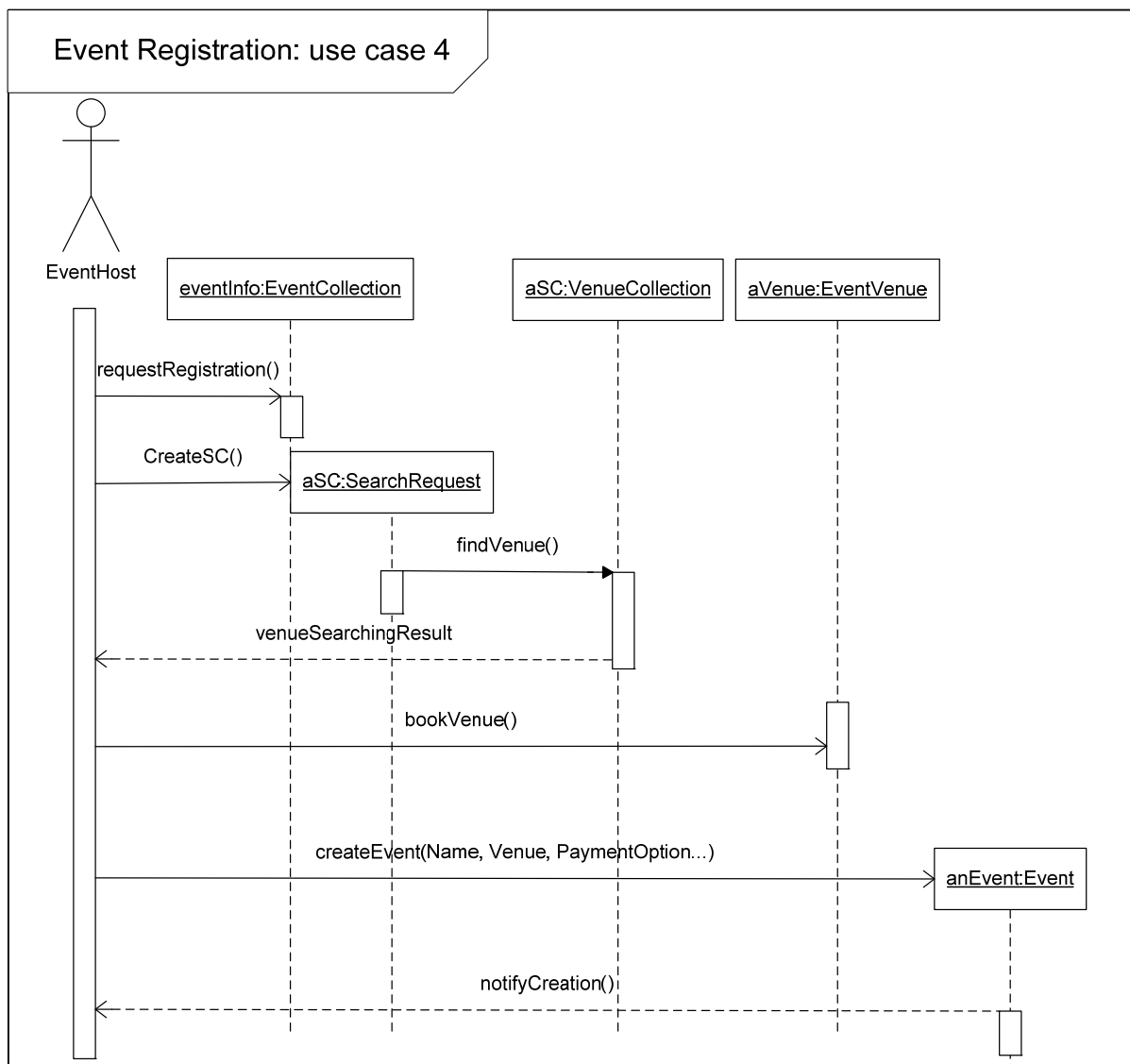


5. Behavioural Models

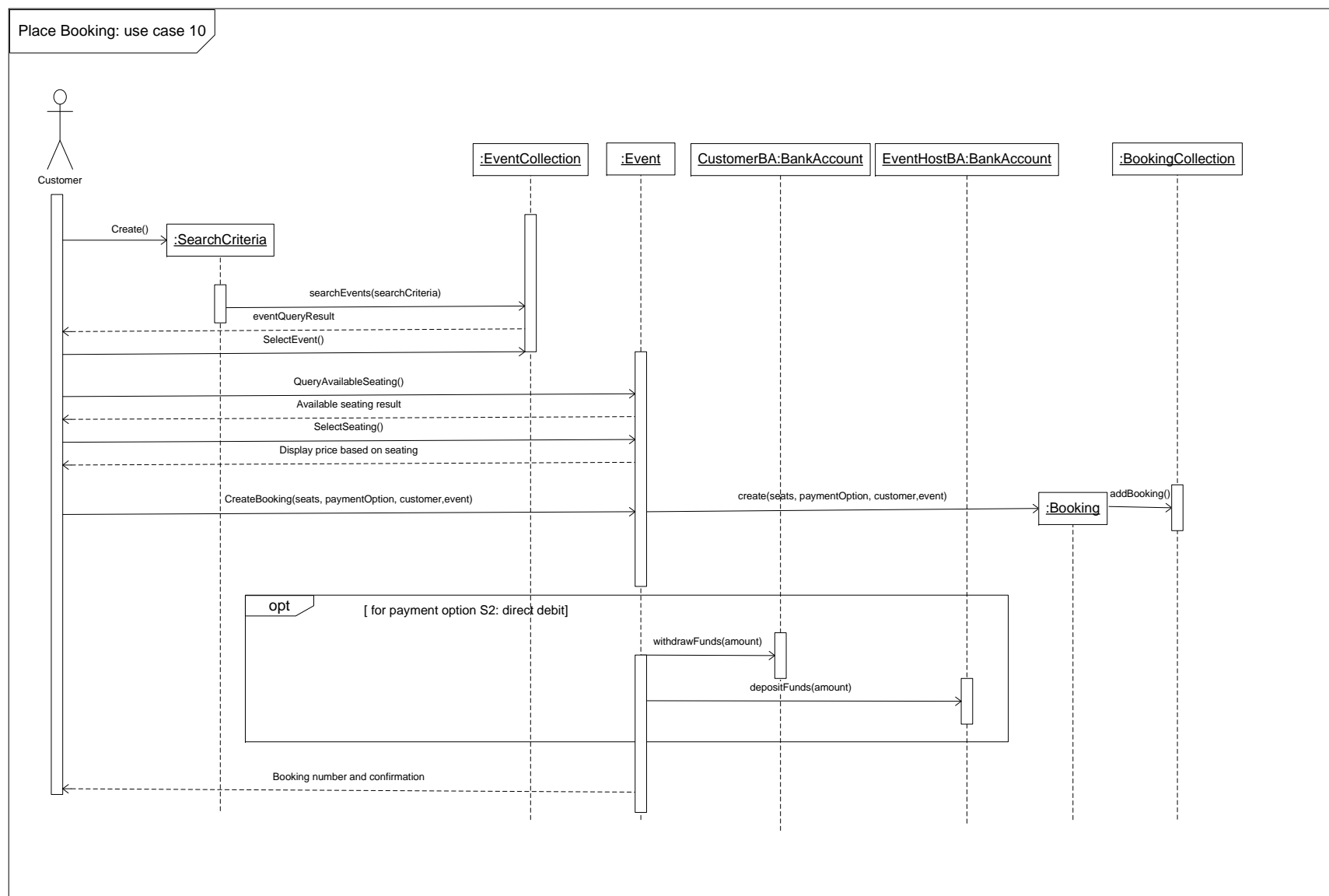
5.1 Object Interaction Modelling

In this section you describe the object interactions for the main Univents business processes. This should include at least the event registration and booking placement processes. You may use either sequence or communication diagrams to represent these processes.

5.1 Sequence Diagram: Event Registration



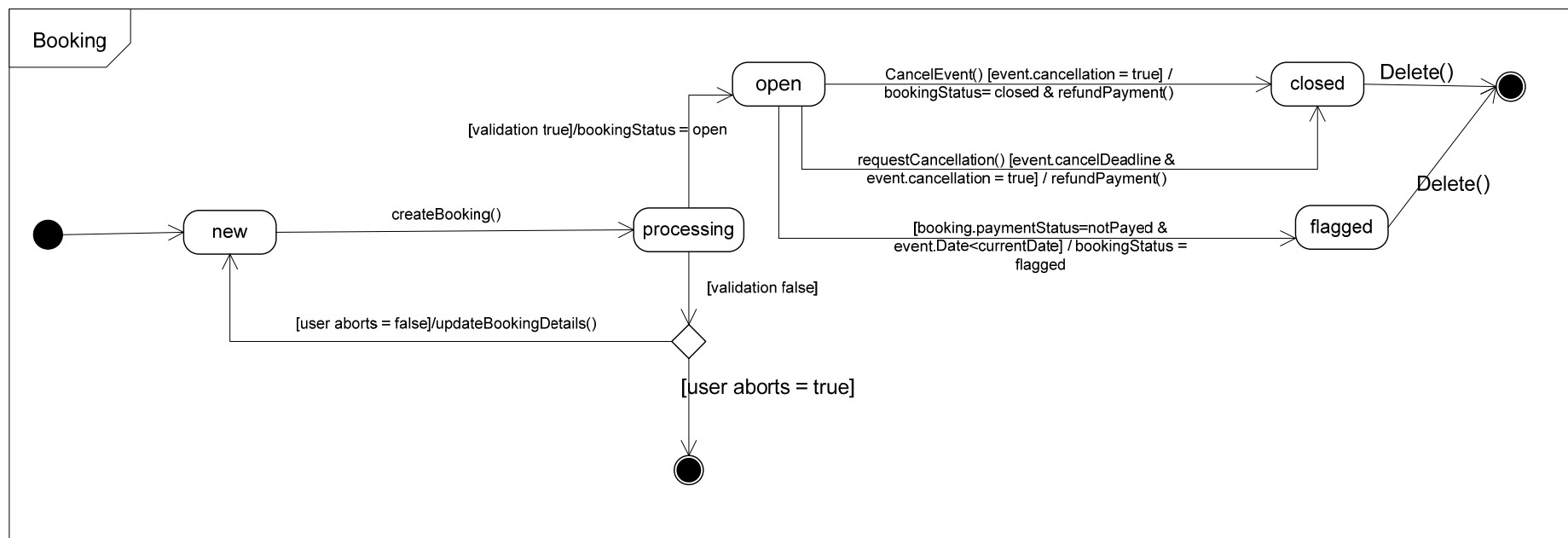
5.1 Sequence Diagram: Booking



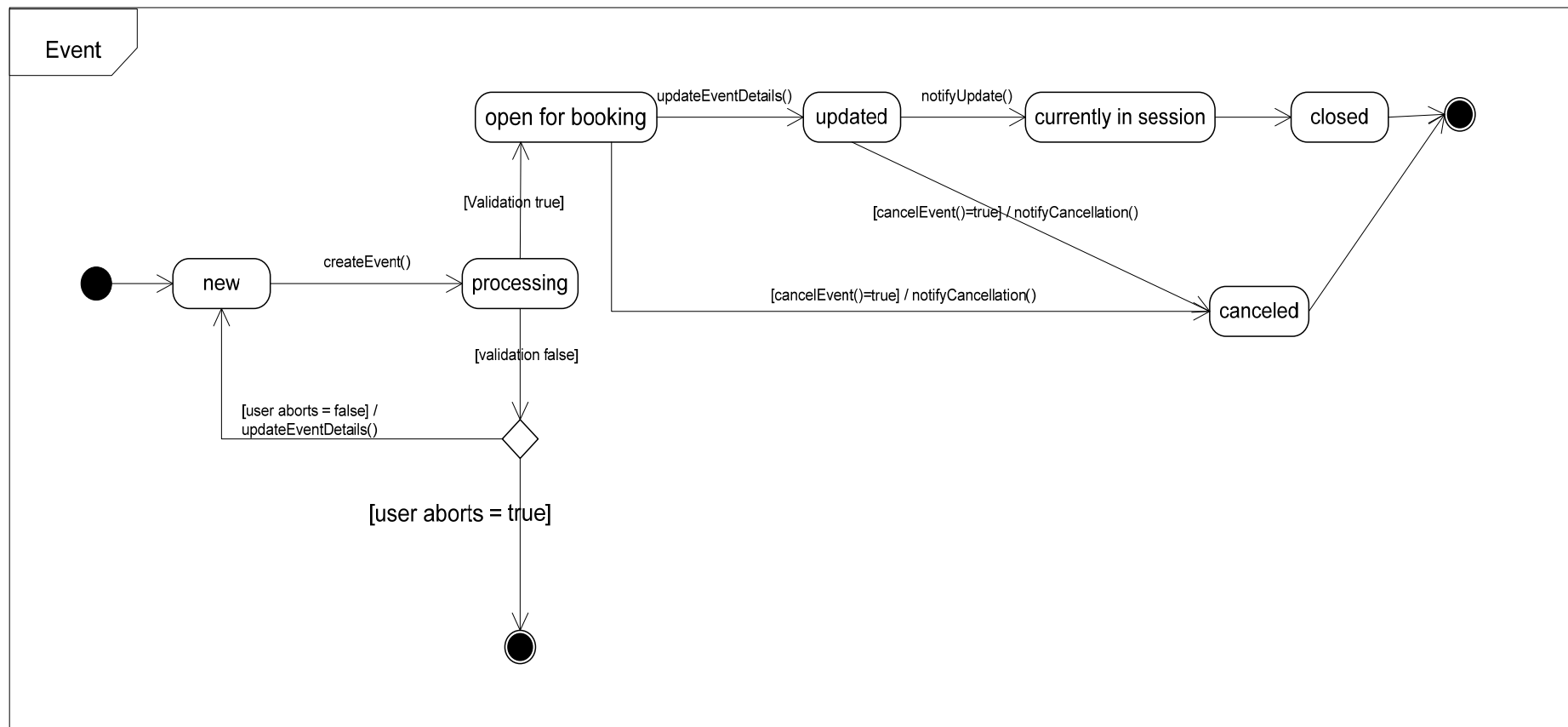
5.2 Object State Models

In this section you describe the object lifetime for the main domain objects in the Univents system. This should include at least the event, booking, and venue object lifecycles.

5.2 State Diagram: Booking



5.2 State Diagram: Event



5.2 State Diagram: EventVenue

