**Document Name:** System Proposal

**Project Name:** Univents

**Project Number:** 524

The system proposal encapsulates information gathered during the analysis of the Univents system. It includes the system request, detailed requirements definition and structural, functional, and behavioural models that document the system analysis.

Keywords: “event management system”, “requirements”, “system proposal”, “initial design”, “system modelling”

**Document Number:** 100-20-25

**Document Status:** Under Development

**Author:**

**Company:** UniUnion

**Classification:** For internal use only

**Based on Template:** 10-100-101 (01/01/2007)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Author** | **Remarks** |
| 1 | 26/7/2011 | Matthew Sladescu | Initial System Request and Detailed Requirements documented. |
| 2 | 02/09/2011 | Minh Tuan Nguyen | Editing all Use Cases |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1. Systems Request – Event Management System 3](#_Toc299815815)

[1.1 Business Needs 3](#_Toc299815816)

[1.2 Business Requirements 3](#_Toc299815817)

[1.3 Business Value 3](#_Toc299815818)

[1.4 Special Issues or Constraints 3](#_Toc299815819)

[2. Detailed Requirements Definition 4](#_Toc299815820)

[3. Functional Models 6](#_Toc299815821)

[3.1 Package Diagram 6](#_Toc299815822)

[3.2 Use case Diagrams 7](#_Toc299815823)

[3.4 Activity Diagrams 22](#_Toc299815824)

[4. Structural Models 23](#_Toc299815825)

[4.1 Class Diagram 23](#_Toc299815826)

[5. Behavioural Models 24](#_Toc299815827)

[5.1 Object Interaction Modelling 24](#_Toc299815828)

[5.2 Object State Models 24](#_Toc299815829)

# 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

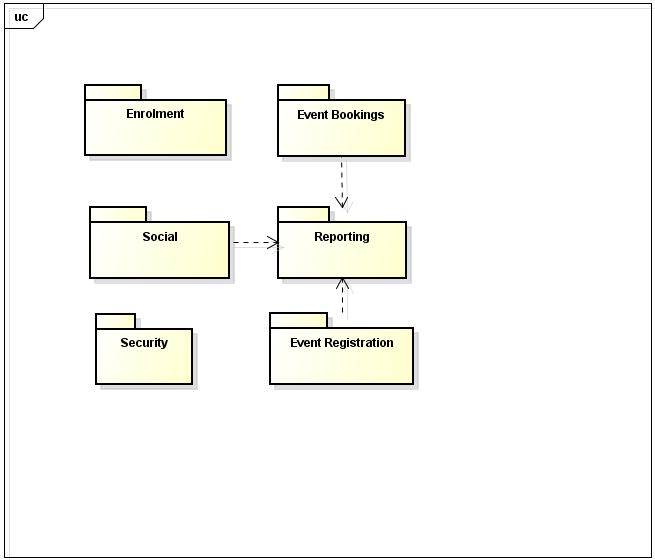
1. The Analysis Phase for this project must be completed by week 6.
2. The CTO and stakeholders will only be available in limited ways (email and questions through a dedicated thread on the eLearning discussion board).

# 2. Detailed Requirements Definition

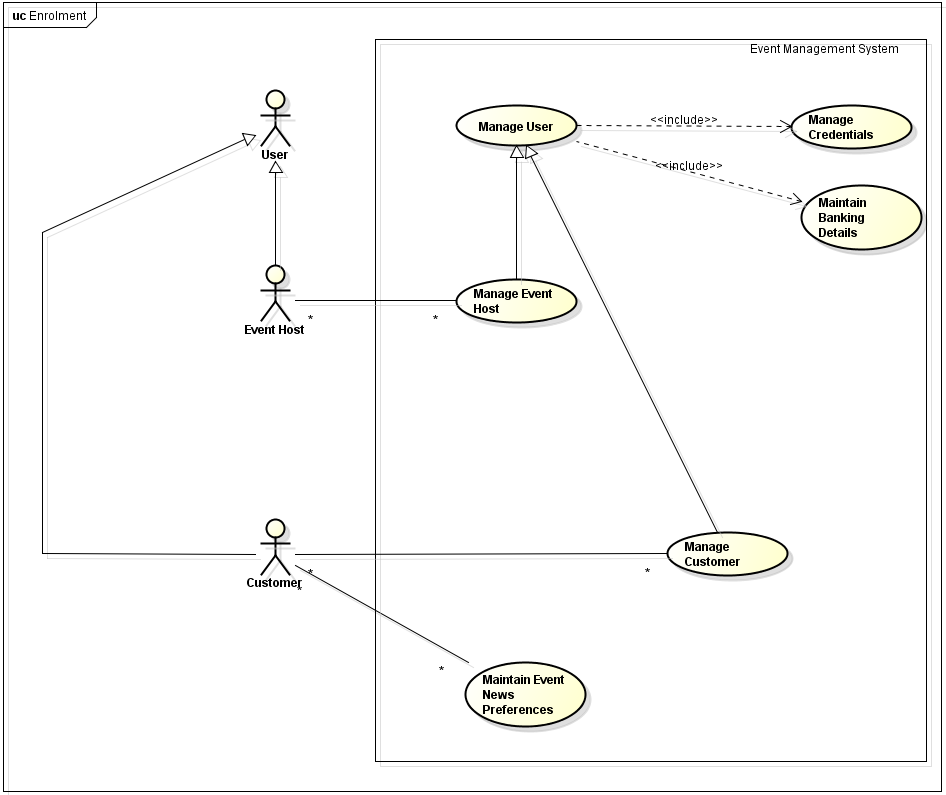
|  |  |  |  |
| --- | --- | --- | --- |
| **Priority** | **Business**  **Req. ID** | **Functional Requirement** | |
| **H**  **H**  **H**  **L**  **H**  **H**  **H**  **H**  **L** | **2**  **2**  **1**  **1**  **1, 2**  **1, 2,3**  **1**  **1, 2,3**  **1** | 1. Event Host Enrolment    1. The system will allow event hosts like union clubs and societiesto enrol into the system.    2. Event hosts will be able to enter at least the following details at enrolment: event host name, ABN, and email.    3. Event hosts will be able to update the details mentioned in II at any time.    4. 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.    5. The event host should be able to change their nominated account details (see 1II) at any time.    6. Event hosts shall be able to make changes to any enrolment details at any time. 2. Event Registration    1. Event hosts will be able to register new events.    2. Events will be linked to the event host making the event registration.    3. Event hosts should be able to query for suitable venues based on event needs such as: expected turnout, required equipment, event time and duration.    4. Event hosts should be able to select a suitable venue from the results in 2 III    5. Event hosts will be able to specify at least the following details: event name, description, venue, time, duration, and cost.    6. Event hosts will be able to specify if booking cancellations are allowed for this event, and the deadline for such booking cancellations.    7. Event hosts will be able to specify if payment for this event can be collected at the venue. 3. Customer Enrolment    1. Customers that wish to make bookings for events hosted by clubs and societies should be able to enrol into the system.    2. Customers will need to nominate an account that can be used to pay for event bookings, and into which any refunds can be deposited.    3. Customers shall be able to make changes to their enrolment details at any time. 4. Event News    1. 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.       1. Customers shall be able to modify their preferences for Functional Requirement 4 I at any time. 5. Event Queries    1. All users shall be able to query for events by event name, description, event host name, venue, time, duration, or cost.    2. 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. 6. Event Details    1. All users shall be able to see the details for a listed event, including the event name, description, venue, time, duration and cost.    2. 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:       1. A booking is open if the booking event occurs in the future.       2. A booking is flagged if payment has not been received and the event has finished.       3. A booking is closed if the event has been cancelled by the customer or event host; or if the event has finished and payment has been received.    3. Event hosts will be able to query their event booking lists by booking name, booking customer contact details, and booking state.    4. Event hosts shall be able to change the state of a booking from flagged to closed, for booking linked to their events. 7. Event Bookings    1. Enrolled customers shall be able to make bookings for any listed event.    2. Customers will be asked to pay immediately by credit card, or at the venue (if permitted by the event host (see Functional Requirement 2VII)).    3. If customers choose to pay immediately, payment is deducted from their nominated account (Functional Requirement 3II)    4. 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. 8. Event Updates    1. Event hosts shall be able to update event information.    2. 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.    3. 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. 9. Event Recommendations    1. Enrolled customers shall be able to recommend an event to friends. | |
|  | **Non – Functional Requirement** | |
| **H**  **H**  **H**  **H** | 1. Interoperability    1. 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. 2. Operational    1. Event hosts and Customers should be able to access the system from any desktop web browser.    2. Customers should be able to access the system from the default iPhone, android, and Windows Phone web browsers. 3. Security    1. 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.    2. 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,.    3. 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. 4. Performance    1. Response times for any type of request should be less than 2 seconds. | |

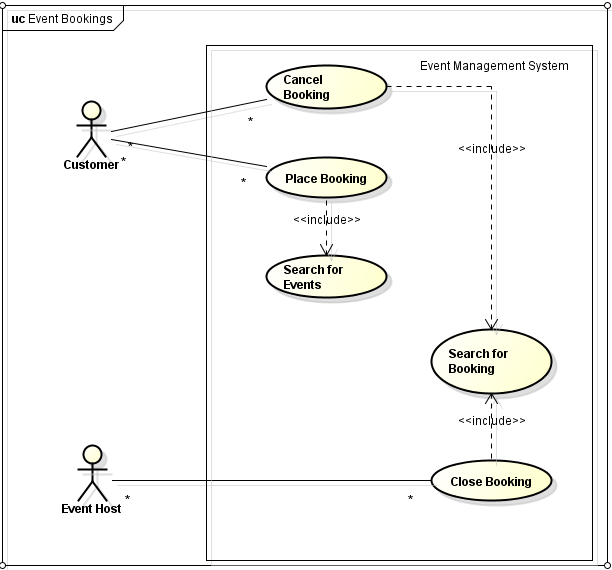
# 3. Functional Models

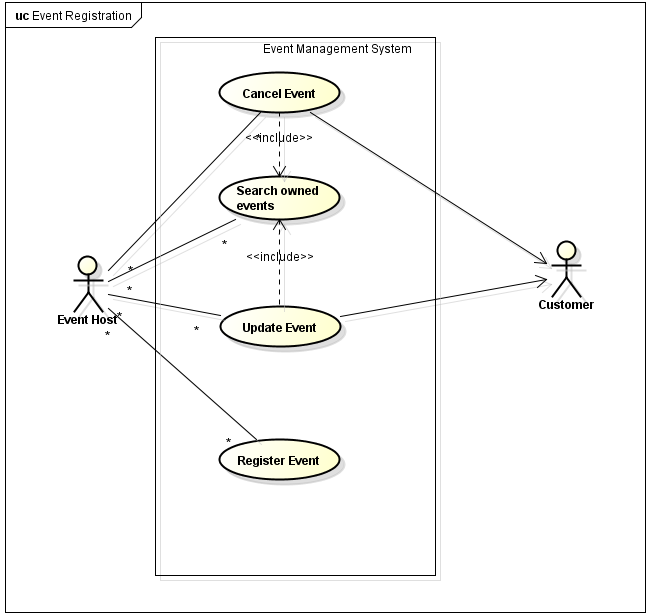
## 3.1 Package Diagram

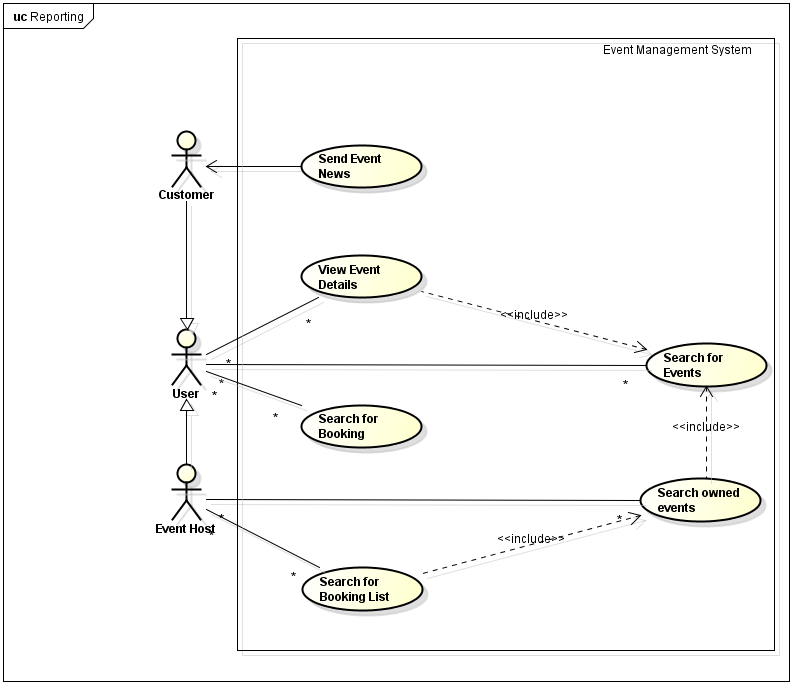


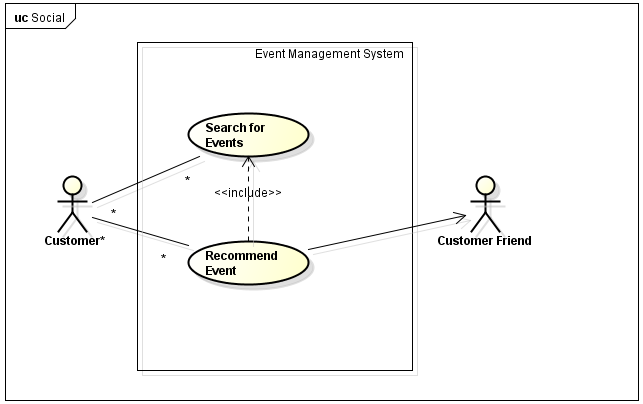
## 3.2 Use case Diagrams

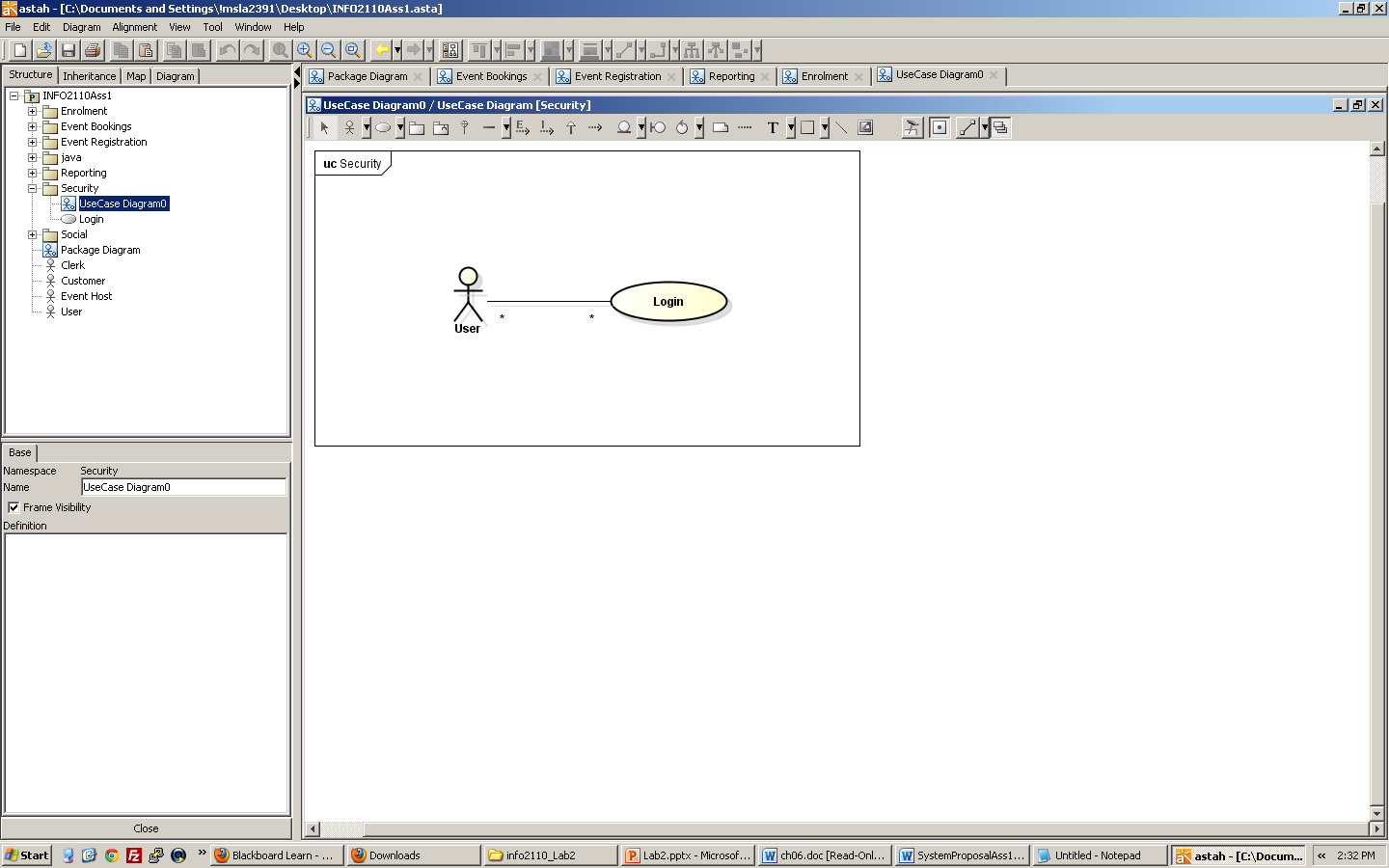












**3.3 Use Case Descriptions**

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   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   1. User executes manage credentials use case. 2. User executes maintain banking details use case 3. System notifies user that his details have been recorded. | | | |
| **Sub-flows:**  S-1 User Enrolment   1. System prompts user to enter details 2. User enters enrolment details. 3. System sends confirmation of enrolment.   S-2 User Details Update   1. System prompts user to enter details. 2. User changes details. 3. System sends confirmation message that changes have been made. | | | |
| **Alternate/Exceptional Flows:**  3a. [User provides invalid details] System refuses to update.  3a.1 Return to step 1.  5a. System displays a message that the changes have not been accepted.  5a1. System terminates  S-1, 3a: [User provides invalid details] System displays a message that enrolment changes were not successful. 3a.1 Return to S-1 step 1  S-2, 3a: [User provides invalid details] System displays a message that changes to user details were not successful.  3a.1 Return to S-2 step 1 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   1. User requests credential management 2. If the user already has credentials:   If the user desires a change in credentials:  Subflow S-1 Credential Assignment is followed  If the user does not already have credentials:  Subflow S-2 Credential Creation is followed   1. If the user’s credentials were updated   The system confirms a successful credential update | | | |
| **Sub-flows:**  S-1 Credential Assignment:   1. System prompts user to enter credentials. 2. User enters credentials. 3. System sends confirmation that credentials have been changed.   S-2 Credential Creation:   1. System prompts user to enter details. 2. User enters credential details. 3. System sends confirmation that credentials have been changed. | | | |
| **Alternate/Exceptional Flows:**  3a. [User provides invalid credentials] System refuses to update.  3a.1 System displays a message that no changes have been made.  3a.2 System terminates.  S-1, 3a [User provides invalid credentials] System refuses to accept credentials.  3a.1 System displays a message that the credentials have not been accepted  3a.2 Return to S-1 step 1  S-2, 3a [User provides invalid credentials] System refuses to create credentials.  3a.1 System displays a message that the credentials have not been created  3a.2 Return to S-1 step 2 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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 enrols into the system, or requests an update to their banking details  **Type:** External | | | |
| **Relationships:**  **Association:** User | | | |
| **Normal Flow of Events:**   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 Change Banking Details is followed  If the User does not already have banking details:  Subflow S-1 Set Banking Details is followed   1. If banking details were set   The system notifies the User of a successful update to their banking details. | | | |
| **Sub-flows:**  S-1 Change Banking Details:   1. System prompts user to enter bank details. 2. User enters bank details. 3. System sends confirmation that bank details have been changed.   S-2 Set Banking Details:   1. System prompts user to enter details. 2. User enters credential details. 3. System sends confirmation that credentials have been changed. | | | |
| **Alternate/Exceptional Flows:**  3a. [User provides invalid bank details] System refuses to update.  3a.1 System displays a message that no changes have been made.  3a.2 Return to S-1 step 1.  S-1, 3a [User provides invalid bank details] System refuses to accept bank details.  3a.1 System displays a message that the bank details have not been accepted  3a.2 Return to S-1 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:**   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. 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:**[No Sub-flows] | | | |
| **Alternate/Exceptional Flows:**  2a. [Event Host provides invalid details]. System notifies Event Host that the details provided are not valid.   1. If Event Host decides to input valid details:   Return to step 2.   1. If Event Host decides to quit, or otherwise:   System terminates.  3a. [No Venues matching entered details found]. System notifies Event Host that no Venue has been found.   1. If Event Host decides to enter different details:   Return to step 2.   1. If Event Host decides to quit, or otherwise:   System cancels request and terminates.  5a. [Event Host makes invalid choice/no choice]. System displays message that no valid choice has been made.   1. If Event Host decides to input valid choice:   Return to step 3.   1. If Event Host decides to quit, or otherwise:   System cancels request and terminates.  6a. [Event Host specifies invalid deadline]. System displays message that invalid deadline was entered.   1. If Event Host decides to input valid deadline:   Return to step 6.   1. If Event Host decides to quit, or otherwise:   System cancels request and terminates.  7a. [Event Host specifies invalid location]. System displays message that the location is invalid.   1. If Event Host decides to input new location:   Return to step 7.   1. If Event Host decides to quit, or otherwise:   System cancels request and terminates. | | | |
| **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:**   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:** [No Sub-flows] | | | |
| **Alternate/Exceptional Flows:**  3a. [No Event Found]. System notifies Event Host that no Event was found.   1. If Event Host wants to Search again:   Return to step 2.   1. If Even Host wants to quit, or otherwise:   System cancels request and terminates.  4a. [Customers have not been notified] System notifies that some customers have not been notified. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** Maintain Event News Preferences | | **ID:** 6 | **Importance Level: H** |
| **Primary Actor:** Customer | **Use Case Type:** Detail, Essential | | |
| **Stakeholders and Interests:**  Customer – wants to receive notifications about upcoming events which they might find interesting. | | | |
| **Brief Description:** This use case describes the steps necessary for customers to register for event news. | | | |
| **Trigger:** Customer  **Type:** External | | | |
| **Relationships:**  **Association:** Customer | | | |
| **Normal Flow of Events:**   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 | | | |
| **Sub-flows:** [No Sub-flows] | | | |
| **Alternate/Exceptional Flows:**  1a. [Customer makes invalid selection]. System notifies the Customer that an invalid selection has been made.   1. If Customer wants to re-select:   Return to step 1.   1. If Customer wants to quit or otherwise:   System cancels request and terminates. | | | |
|  | | | |
| **Use Case Name:** Send Event News | | **ID:** 7 | **Importance Level: H** |
| **Primary Actor:** Triggered By System | **Use Case Type:** Detail, Essential | | |
| **Stakeholders and Interests:**  Customer – wants to receive notifications about upcoming events which they might find interesting. | | | |
| **Brief Description:** This use case describes the steps necessary for sending weekly event news to customers. | | | |
| **Trigger:** Beginning of Week (Every week the event management system sends out news to customers about upcoming events which customers might be interested in).  **Type:** Temporal | | | |
| **Relationships:**  **Association:** Customer | | | |
| **Normal Flow of Events:**   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)   1. System repeats steps 4 and 5 for all event hosts that Customer X is registered to. 2. System sends email to Customer X regarding upcoming events for the event hosts that they are subscribed to. 3. System repeats steps 2 to 7 for each customer that has registered for weekly emails. | | | |
| **Sub-flows:** [No Sub-flows] | | | |
| **Alternate/Exceptional Flows:**  2a. [No customer is found on list]. System terminates.  5a. [No events for Event Host Y]. System goes to step 6.  7a. [No email to send]. System does not send any email to Customer X. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   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:   Sub-flow S-1 Payment options is performed  If the event does not have a pay at venue option:  Sub-flow S-2 Process Direct Debit Payment is performed   1. System reserves seating for customer, and updates available seating for the event. 2. System provides customer with a unique booking number, and the booking is confirmed with the customer. | | | |
| **Sub-flows:**  Sub-flow S-1 Payment Options:   1. System asks Customer to choose paying option, either Pay at Venue or Pay Now 2. If Customer chooses Pay at Venue:   System records that Customer will Pay at Venue.   1. If Customer chooses Pay Now:   Sub-flow S-2 option is performed.  Sub-flow S-2 Process Direct Debit Payment:   1. System retrieves Banking Details from Customer 2. System transfers money from Customer Account to Event Host Account. 3. System produces receipt for Customer. | | | |
| **Alternate/Exceptional Flows:**  3a. [No Event found] System notifies Customer that no Event found.   1. If Customer wants to search again:   Return to step 1.   1. If Customer wants to quit or otherwise:   System cancels request and terminates.  4a. [No available Seats] System notifies Customer that there are no available seats left.   1. If Customer wants to search again:   Return to step 1.   1. If Customer wants to quit or otherwise:   System cancels request and terminates.  7a. [Customer does not confirm seating and/or booking price.   1. If Customer wants to search again:   Return to step 1.   1. If Customer wants to quit or otherwise:   System cancels request and terminates. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   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:** [No Sub-flows] | | | |
| **Alternate/Exceptional flows:**  3a. [No booking found] System notifies the Customer that no booking matching the details have been found.   1. If Customer wants to enter details again:   Return to step 2.   1. If Customer wants to quit or otherwise:   System cancels booking request and terminates.  4a. [Event Host does not allow for booking cancellation] System notifies the user that the booking cancellation has not been accepted.   1. If Customer wants to enter details again:   Return to step 2.   1. If Customer wants to quit or otherwise:   System cancels booking request and terminates.  5a. [Customer choose to Pay at Venue] System skips step 5 if Customer Payment details is set to Pay at Venue. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   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   1. Customer lists the addresses of friends he wishes to forward the recommendations to 2. Customer enters a recommendation message 3. Customer requests that the recommendation is sent 4. System sends recommendation including recommendation message and events selected in step 3 to the list of addresses from step 5. 5. System gives confirmation to customer that the recommendation was sent. | | | |
| **Sub-flows:** [No Sub-flows] | | | |
| **Alternate/Exceptional flows**  3a. [No Event chosen] System notifies Customer that no Event has been chosen.   1. If Customer wants to search again:   Return to step 2.   1. If Customer wants to quit or otherwise:   System cancels request and terminates.  5a. [Invalid addresses] System notifies user that some addresses were invalid and that messages have not been sent.  5a1. System terminates.  6a. [Customer did not enter message] System notifies Customer that they did not enter a message.   1. If Customer wants to enter a message:   Return to step 6   1. If Customer wants to quit or otherwise:   System cancels request and terminates. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   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:** [No Sub-flows] | | | |
| **Alternate/Exceptional flows**  2a. [No Event found] System notifies User that no Event was found.   1. If User wants to search again:   Return to step 6   1. If User wants to quit or otherwise:   System cancels request and terminates. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   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:** [No Sub-flows] | | | |
| **Alternate/Exceptional flows:**  2a. [No Event found] System notifies Event Host that no matching Event has been found.   1. If Event Host wants to search again:   Return to step 2   1. If Event Host wants to quit or otherwise:   System cancels request and terminates. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   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:** [No Sub-flows] | | | |
| **Alternate/Exceptional flows:**  2a. [No Event found] System notifies User that no matching Event has been found.   1. If User wants to search again:   Return to step 2   1. If User wants to quit or otherwise:   System cancels request and terminates. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | | | |
| **Relationships:**  **Association:** Event Host  **Includes:** Search for Owned Events | | | |
| **Normal Flow of Events:**   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. | | | |
| **Sub-flows:** [No Sub-flows] | | | |
| **Alternate/Exceptional flows:**  2a. [No Event found] System notifies Event Host that no matching Event has been found.   1. If Event Host wants to search again:   Return to step 2   1. If Event Host wants to quit or otherwise:   System cancels request and terminates.  4a. [No bookings found] System notifies Event Host that no bookings have been found.   1. If Event Host wants to search again:   Return to step 2   1. If Event Host wants to quit or otherwise:   System cancels request and terminates. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   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,   The event hosts requests a change of booking state to closed  The system returns confirmation of booking change of state. | | | |
| **Sub-flows:** [No Sub-flows] | | | |
| **Alternate/Exceptional flows:**  2a. [No Event found] System notifies Event Host that no matching Event has been found.   1. If Event Host wants to search again:   Return to step 2   1. If Event Host wants to quit or otherwise:   System cancels request and terminates.  4a. If the Booking is closed or open:   1. System notifies the Event Host that the booking can not be closed. 2. If Event Host wants to cancel another booking:   Return to step 2   1. If Event Host wants to quit or otherwise:   System cancels request and terminates. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   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:** [No Sub-flows] | | | |
| **Alternate/Exceptional flows:**  2a. [User does not have permission] System informs user that they do not have permission.  2a.1 System terminates.  5a. [No bookings found]. System informs user that no bookings have been found.   1. If Event Host wants to search again:   Return to step 2   1. If Event Host wants to quit or otherwise:   System cancels request and terminates. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   1. The event host requests an event cancellation 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 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. | | | |
| **Sub-flows:** [No Sub-flows] | | | |
| **Alternate/Exceptional flows:**  3a. [No Event found]. System notifies Event Host that no Event has been found.   1. If Event Host wants to search again:   Return to step 2   1. If Event Host wants to quit or otherwise:   System cancels request and terminates.  4a. [No customers found for that booking] System notifies Event Host that no customers have been found.   1. If Event Host wants to search for another booking :   Return to step 2   1. If Event Host wants to quit or otherwise:   System skips step 4 and terminates.  6a. [No customers found for that booking]. System notifies Event Host that no customers have been found.   1. If Event Host wants to search for another booking:   Return to step 2   1. If Event Host wants to quit or otherwise:   System does not send any email and terminates. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   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   1. Event Host executes the normal flow of events in Manage User. 2. System notifies event host of any successful updates to their details. | | | |
| **Sub-flows:** [No Sub-flows] | | | |
| **Alternate/Exceptional Flows:**  2a. [Event Host entered Invalid Details] System notifies Event Host about invalid details   1. If Event Host wants to enter details again:   Return to step 2   1. If Event Host wants to quit or otherwise:   System cancels request and terminates.  2b. [Event Host entered same details] System notifies Event Host that the details are the same.   1. If Event Host wants to enter details again:   Return to step 2   1. If Event Host wants to quit or otherwise:   System cancels request and terminates. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   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:** [No Sub-flows] | | | |
| **Alternate/Exceptional Flows:**  3a. [Not successful] System notifies Customer that the request was not successful   1. If Customer wants to enter details again:   Return to step 2   1. If Customer wants to quit or otherwise:   System cancels request and terminates. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   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:** [No Sub-flows] | | | |
| **Alternate/Exceptional Flows:**  4a. [User entered invalid Credentials]. System notifies User that credentials are not valid.   1. If User wants to enter Credentials again:   Return to step 2   1. If User wants to quit or otherwise:   System cancels request and terminates. | | | |

## 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 Univents domain*

# 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.*

## 5.2 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.*