

Inf 43 – Spring Quarter, 2015 – Homework 1

Student Name: Kevin Anteater
Student Number: 12345678

Awarded Points	Maximum Points	Document Aspect
	15	Clarity of writing (spelling, grammar, sentence construction) and Clarity of expression (flow, structure, making logical arguments). Roughly 7.5 each.
	15	Introduction / Executive Summary (can be different sections or combined into one)
	7.5	Application Context / Environmental Constraints (can be different sections or combined into one)
	35	Functional Requirements, including use-case diagram and each use case (following a use-case template).
	7.5	Software Qualities and Non-functional Requirements
	5 (+5)	Other Requirements and Other Items. At least a Glossary of Terms. You can earn up to 5 points Extra Credit if you go beyond Glossary
	7.5	Assumptions / Risks (can be different sections or combined into one)
	7.5	Priorities / Implementation Phases; Future Directions and Expected Changes
	100	TOTAL

xxx System Requirements

July 10, 2013

Kevin Anteater

1. Introduction

BeachBurn Manager (BBM) is a software project commissioned by the ABC event management firm for their annual *BeachBurn* music festival. BBM is being developed by Kevin Anteater, LLC. BBM will assist ABC with ticket management, schedule management, and resource management before and during the festival. BBM will be used to automate communication about these resources between their respective festival managers. The project will allow real time updating between different areas of management to help ABC maximize the number of tickets sold for each day of the event and maximize the number of tickets sold for each different sector during the event (VIP, General Admission, etc).

This document details the requirements for BBM and will be used as a template for later implementing the project and as a reference for all persons involved in the development and purchase of BBM. Each section of the document contains a header with a short description of its contents (*Use Cases*, *Systems Requirements Specification*, etc). The headers are as follows:

1. Introduction [This Section] – Contains an introduction to the BBM project and this specification document.
2. Overview/Executive Summary – A brief introduction to the project covering the major features/concerns of the system.
3. Application Context/Environmental Constrains – Provides further details of the specific context in which the application will run. Includes information about the systems the project will run on.
4. Functional Requirements – Covers the most important ways each type of user will interact with the system. Includes a use case diagram. Broken down into subsections organized by user:
 - 4.1 All Users
 - 4.2 Ticket Vendors
 - 4.3 Ticket Manager
 - 4.4 Stadium Manager
 - 4.5 Schedule Manager
 - 4.6 Web Administrator
 - 4.7 System Administrator
5. Software Qualities – Covers the nonfunctional requirements of the project. Organized into the following specific subsections.
 - a. Availability
 - b. Usability
 - c. Scalability
 - d. Reliability
6. Other Requirements – Contains a glossary of all major terms used in this specification.
7. Assumptions/Risks –
8. Priorities/Implementation Phases – Covers what the most important implementation priorities are. Organized into three levels.

9. Future Directions/Expected Changes – A brief explanation of how the project is expected to change.

2. Overview / Executive Summary

BeachBurn is an annual five-day music festival held in the *XYZ club* at Laguna Beach, California. The organizer of the festival is the *ABC event management firm*. BeachBurn currently has a capacity of about 1500 seats across five arenas at the date of writing of this document. The event has been growing in popularity and overall attendance has been on the rise.

ABC currently handles the management and coordination of ticket sales and scheduling manually. ABC has experienced problems with their current system, which is not able to maximize ticket sales and profits. Notably, ABC has previously operated at a loss for periods of time during the event due to last minute band cancellations and the subsequent ticket refunds they have endured due to their lack of a single system to coordinate band/stadium schedules. Additionally, selling close to the maximum available number of seats in the VIP and other sectors at the festival has been extremely difficult for ABC as they have no system in place for arena management to communicate their changing seat composition/availability in real time to the event's ticket vendors. ABC would like to put in place one automated web system to minimize financial losses due to problems like these.

ABC management has commissioned Kevin Anteater, LLC to design and implement this software system, which will be called BeachBurn Manager (BBM). BBM's primary purpose is to ensure effective communication between the BeachBurn arena management, ticket management, schedule management, and the ticket vendors. In addition to the three aforementioned user groups, ABC's user administrator will be taking on the responsibilities of allocating and maintaining access amongst the various users of BBM.

3. Application Context / Environmental Constraints

The BBM project is expected to run on Windows machines running Windows 98 or later. The software is intended to run as a web based application. It will run in all major web browsers, including Mozilla Firefox, Google Chrome, and Microsoft Internet Explorer. There are currently no plans for a mobile version of the application. The customer has requested that the UI be "simple." While there are no specified restrictions on choice of programming language, the application must be capable of outputting information to a Microsoft Excel spreadsheet and also must be capable of performing in all browsers. It is necessary to reiterate these restrictions two application contexts as they will be relevant in the choice of programming language(s) used.

4. Functional Requirements

This section introduces the basic ways in which each unique type of user of the system will interact with the system. It identifies the main use cases, or interactions with the system, which will be addressed in the final product. The section starts with general-purpose use cases for all system users and is then organized into subsections addressing the use cases of each of the individual users of the BBM system. An additional section relating to the user interface (UI) is located at the end of this section. See the glossary of terms for further definitions of these components as they are referenced in this document.

Figure 1 shows the use case diagram for BBM. The remainder of this section describes each use case in further detail.

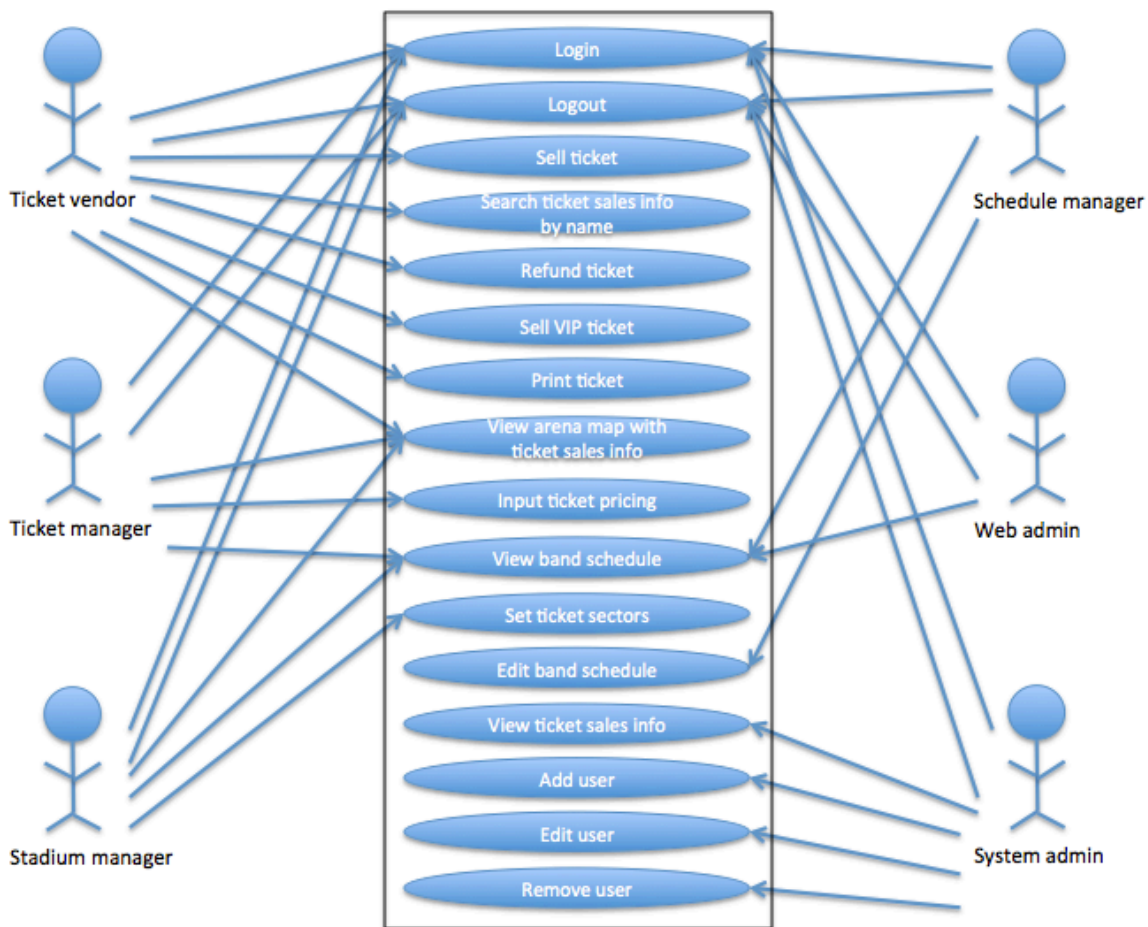


Figure 1: Use Case Diagram for BBM.

4.1a Functionality Included for All Users

4.1a.1 – Logging In:

Login information is requested from each user every time they want to start a new session in the BBM system.

- Preconditions: User must already have login credentials, ie username and password.

Basic Flow of Events:

1. User opens a compatible web browser and points it to the BBM application
2. BBM application prompts the user to login with their username and password
3. User fills in their login credentials
4. System confirms the user has logged in and directs them to the main user interface for their position.

4.1a.2 – Logout Function

Users are required to logout every time they are finished using the BBM system. This insures no one can use their session after they've left the computer.

- Preconditions: User is logged in

Basic Flow of Events:

1. User clicks the link that takes them to the logout page
2. System requests confirmation that the user wants to logout.
3. System terminates the users session
4. System displays a confirmation that the user has logged out and redirects the browser to the login page.

Alternative Flow:

- 3a. User indicates they do not want to log out.
 1. System returns the user to the main user interface for their position.

4.1b – Functionality Included for Multiple Users

4.1b.1 – View Arena Map with Ticket Sales Information

Shows the user a map of the arena with all of the color-coded ticket sales information (i.e., tickets sold/tickets available, names/locations of ticket sectors, etc.). The arena map will be different based on the day of the event (as selected by the user).

4.1b.2 – View Band Schedule

Shows the user a calendar of events from opening to closing for a day of the event (as specified by the user). Should show all time slots scheduled for playing times with the names of the bands. Additionally, all empty time slots should also be displayed.

4.2 – Ticket Vendor Functionality

4.2.1 – Sell a Ticket

Allows the vendor to sell individual seats in the arena for a single day. The vendor can sell multiple tickets in one transaction.

- Preconditions: Ticket vendor is logged in and located at the main ticket vendor UI.

Basic Flow of Events:

1. Ticket vendor clicks the link to sell tickets and then selects then selects the choice of event day for which to sell tickets.
2. System displays the arena map with ticket sales information.
3. Customer looks through the available seats for the day and selects the seats(s) they'd like.
4. System flags the chosen seat(s) as unavailable.
5. System presents current pricing information to the vendor.
6. Vendor presents the price per seat and total price to the customer
7. Ticket vendor confirms the customer would like to buy the indicated seat(s) at the given price.
8. Vendor accepts cash payment and enters the amount collected into the system.
9. Vendor checks the customer's ID and enters their name.
10. Vendor enters the name(s) of person(s) intending to use extra seat(s).
11. System prints paper tickets for the customer and updates the ticket sales excel file with their information.

Alternative Flow:

- 3b. Customer selects a ticket in the VIP sector.
 1. System prompts the vendor for the customer's general admissions ticket number(s).
 2. Vendor enters the customers existing ticket number to confirm that a ticket has already been purchased.
 3. The system displays the cost of the VIP seat(s) and then returns to the basic flow at step 7.
- 7a. Customer decides to not purchase the seats at the given price
 1. Vendor confirms customer no longer wants the seats in question.
 2. System reflags the seats as available for sale.

4.2.2 – Search for Ticket Sales Information by Name

Perform a search of the ticket sales log to find a specific customer name. Used for printing out lost tickets, initiating a ticket refund, etc.

- Preconditions: User has specific first and last name to search the ticket sales Excel spreadsheet.

Basic Flow of Events:

1. Vendor enters the system menu used to search the ticket sales log by name.
2. System prompts the user to enter the search name, both first and last.
3. System continues to search the ticket sales log until it locates all instances of the search name.
4. System displays a list of tickets purchased for use by the customer entered. This list is ordered by day of the festival.
5. Vendor selects the specific ticket day for which they want to see details.
6. System displays the ticket information on file for the day selected, including purchase price and seat/sector location.

Alternative Flow:

- 3a. System exhausts sales log without finding a ticket matching the search data.
 1. System displays message that no matching ticket has been found and returns to the main ticket vendor UI.

4.2.3 – Refund a Ticket

Allows the vendor to process a refund for a ticket that has not yet been used. Admissions tickets are not refunded upon entrance to the arena. Tickets can only be refunded for their original purchase price as saved in the ticket sales Excel file.

- Preconditions: Ticket vendor has searched for the specific ticket number in question using the ***Search for Ticket Sales Information By Name*** function.

Basic Flow of Events:

1. Ticket vendor enters the ticket number to refund.
2. System searches for the ticket number in the Excel file.
3. System displays the ticket sales information necessary from the ticket sales Excel file, including the price paid.
4. Vendor confirms they have checked the customers ID and refunds the customers ticket price.
5. System reflags the seat as available and then removes the sale from the ticket sales Excel file.

OPEN ISSUE: ABC Events has not confirmed what they want done with the refunded ticket information. The BBM development team suggests moving the ticket sale entry to another Excel file if it's needed.

6. System returns to the main ticket vendor UI screen.

Alternative Flow:

- 3a. Vendor enters a ticket sales number not matching a valid ticket number sold.
 1. System displays a message indicating the ticket number entered does not match anything on file, returns to basic flow at step 6.
- 4a. Ticket number matches a ticket that has already been used for admission

1. System displays a message indicating that the ticket in question has already been used and cannot be refunded, returns to basic flow at step 6.

4.2.4 – Print Ticket

Ticket vendors can reprint lost tickets for customers who lose their tickets if and only if the ticket has not yet been processed at the admissions gate.

- Preconditions: Ticket vendor has searched for the specific ticket number in question using the ***Search for Ticket Sales Information By Name*** function.

Basic Flow of Events:

1. Vendor selects the print ticket option from the ticket vendor main UI screen.
2. System prompts the vendor for the ticket number to reprint.
3. Customer enters the ticket number.
4. System displays the ticket sales information necessary from the ticket sales Excel file, including customer name.
5. Vendor confirms they have checked the customers ID.
6. System prints a copy of ticket.
7. System returns to the ticket.
8. System goes back to main ticket vendor UI screen.

Exception Flow:

- 3a. Vendor enters a ticket sales number not matching a valid ticket number sold.
 2. System displays a message indicating the ticket number entered does not match anything on file, returns to basic flow at step 8.
- 4a. Ticket number matches a ticket that has already been used for admission.
 2. System displays a message indicating that the ticket in question has already been used and cannot be reprinted, returns to basic flow at step 8.

4.2.5 – Upgrade Ticket

To upgrade a customer's ticket to a VIP ticket, the ticket vendor uses the ***Sell a Ticket*** menu option and selects a seat in the VIP sector. For further detail, see the ***Sell a Ticket*** specification.

4.3 – Ticket Manager Functionality

4.3.1 – Input Ticket Prices

Allows the ticket manager to input a new price for seats that have yet to be sold. Ticket manager can select any number of unsold seats in a single sector to update with new pricing information.

- Preconditions: Ticket on the arena map has not been already been sold.

Basic Flow of Events:

1. Ticket manager selects the menu option to input ticket prices and then enters the event day to be affected.
2. System displays the arena map with ticket sales information.
3. Ticket manager selects the seats to be changed.
4. System displays current sales price of seats selected and prompts for new sales price.
5. Ticket manager enters the new sales price.
6. System updates the arena map with ticket sales information to reflect the new ticket price for the selected seats and flags the seat(s) as available.

4.4 – Stadium Manager Functionality

4.4.1 – *Set Ticket Sectors*

Lets the stadium manager create ticket sectors. Also allows the stadium manager to change the size, name, or color code of a ticket sector.

- Preconditions: Seats to be made into new sectors or moved from one sector to another cannot have been sold. It is not expected that seats that have been sold will need to be re-sectored often. Any seats that do need to be re-sectored are best processed manually to insure all necessary refunds are processed and that any updated print tickets are given to the customer.

Basic Flow:

1. User follows the link to ***Set Ticket Sectors*** and then enters the event day(s) to be affected.
2. System displays the stadium map with ticket sales information.
3. Stadium manager selects the seats he wants to form into a new sector or add to an existing sector.
4. System requests confirmation that the seats selected are the correct ones.
5. Stadium manager confirms he wants to edit the seats selected and enters the name/color code of the new sector to be constructed.
6. System updates the stadium map with ticket sales information to show the newly updated sector as unavailable as they are not yet priced.
7. System sends an e-mail notification to the ticket manager requesting he price the new sector ASAP.

Alternative Flow:

- 4b. Stadium manager enters the name/color code of an existing sector.
 1. System alerts stadium manager that he has entered an existing sector's information and sets the selected seats to be part of the sector entered, then returns to the basic flow at step 5.

4.4.2 – Mark Seats as Available/Unavailable

Allows the user to toggle a seat from available to unavailable/from unavailable to available. Marking seats unavailable in this manner is typically used for seats which are broken or inoperable and thus the menu updates the arena map for all even days. In the rare event that the stadium manager wants to mark seats as unavailable for only part of the festival, the sales manager must follow the current manual procedure or create a single day sector named “Unavailable.” The choice is up to ABC staff.

- Preconditions: Assumes that the seat has been previously marked as available or unavailable. All seats are assumed available until stated otherwise.

Basic Flow:

1. System displays the stadium map.
2. Manager selects the seats to toggle availability.
3. System displays the stadium information for the seats entered (ie available/unavailable).
4. The manager confirms his choice.
5. The system toggles the availability of the selected seats.

4.5 – Schedule Manager Functionality

4.5.1 –Input Band Schedule

Allows the schedule manager to input a band’s schedule for the show for the first time.

- Preconditions: Assumes that the schedule manager accurately knows the band lineup.

Basic Flow:

1. Schedule Manager follows the link for inputting the band schedule.
2. System prompts for the name of the band to input into the schedule.
3. Schedule Manager inputs the name of the band and then enters their schedule using an interactive calendar function.
4. System updates the band calendar and displays a confirmation message.

Alternative Flow:

- 3a. Schedule manager inputs the name of a band already entered into the system.
 1. System displays a message to the user indicating that the band is already scheduled to play at a certain time.
 2. System asks if user would like to modify the band’s schedule and then routes the user to the ***Modify Band Schedule*** link if the user indicates so.
- 3b. Schedule Manager tries to input the band into a time slot that is already

occupied by another band.

1. System displays an error message to the user regarding the conflicting time schedule.
2. System asks if the user would like to modify the schedule of the band currently in the time slot and routes user to the ***Modify Band Schedule*** page if the user indicates to do so.

4.5.2 – Modify Band Schedule

Allows the user to modify the schedule for an existing band in the system. User can input new playing times or remove times to play. If the user wants to extend/reduce a current playing time in the system, the user must separately remove the existing time schedule and add a new one reflecting the updates.

- Preconditions: The band to be modified in the schedule must already have been input into the system using the ***Input Band Schedule*** screen.

Basic Flow:

1. Schedule Manager follows the link to modify a band schedule.
2. System prompts the user to pick a band from a listing of all bands in the system to modify.
3. Schedule Manager selects a band to modify.
4. System shows a listing of all times that the band selected is scheduled to play and the calendar for all days of the event and prompts the user to either add or remove a schedule time.
 - a. If the user selects to “remove” a scheduled time, the user must also select a time from the list presented to remove.
 - b. If the user selects to “add” a scheduled time, the user must also select an available time slot in the schedule to add for the band.
5. After user makes their choices, system displays the changes to be made to the schedule and requests confirmation.
6. System outputs a message confirming the changes to the schedule.

4.5.3 – Remove Band from Schedule

Allows the user to remove a band and all of its scheduled performances from the system. To reintroduce this band to the event after removal, the input band schedule process must be used like any other “new” band.

- Preconditions: Assumes that the band has been input into the system using the ***Input Band Schedule*** function.

Basic Flow:

1. User points the browser to the ***Remove Band from Schedule*** screen.
2. System displays a listing of all of the bands input into the system.
3. User chooses a band to remove and confirms their choice.

4. System clears all performances of the removed band from the system and displays a confirmation message.

4.6 – Web Administrator Functionality

The website administrator's only use case in the BeachBurn Manager system is to check the band schedule. For a detailed use case specification for '*View Band Lineup*', consult the section on functionality included for more than one user.

4.7 – System Administrator Functionality

4.7.1 – Add User to System

Allows the system administrator to add a new user to the BBM system. Prompts the sys admin for the full name of the new user to be added. After the sys admin enters the new users name, the system prompts the sys admin for a user name and password. Once, the user name and password are established, the system prompts the sys admin for the position of the new user so that the new account can be given the proper access permissions.

4.7.2 – Remove User from System

Allows the system administrator to remove an existing user from the BBM system. The system prompts the sys admin for the username to be removed. Upon entering a current username, the system requests confirmation to remove said user. If the sys admin confirms that this is the user to be removed, the system removes the username and corresponding password from the system.

To add a user back into the system, the system administrator must reprocess the user being added as though he's a new user with the *Add User to System* menu option.

4.9 – User Interface (UI) Specification

UI links per position:

- I. Ticket Vendor:
 - a. Sell Ticket
 - b. Refund Ticket
 - c. Search for Ticket Sales Information by Name
- II. Ticket Manager:
 - a. Input Ticket Pricing
 - b. View Band Schedule
- III. Stadium Manager
 - a. Set Ticket Sectors
 - b. Mark Seats as Available/Unavailable
- IV. Schedule Manager

- a. Check Band Schedule
 - b. Modify Band Schedule
- V. Web Administrator
 - a. Check Band Schedule
- VI. System Administrator
 - a. View Ticket Sales Information
 - b. Add User
 - c. Remove User

5. Software Qualities and Non-functional Requirements

Availability – The system must be always available for use whenever needed in the three months before the festival. The system may not go down or be otherwise inaccessible as the sale of tickets is dependent on the real-time updates to the stadium arena map.

Usability – The system and its user interfaces must be simple enough for people with every day computer literacy (ie web browsing, using keyboard/mouse, etc.) to pick up. It is not expected that ABC Management will have to train its employees in using the application outside of explaining the core processes. The interface must be largely self-explanatory.

Scalability – The system as it is currently being designed will only handle one of the five stages used for the BeachBurn event. It is known that in later iterations BBM will need to handle all of these additional stages. It is also expected that the event is going to expand further as its popularity has been on the rise. The software must be designed with great consideration to scalability as this is known to be a future issue.

Reliability – The system is being implemented so as to minimize financial losses to the ABC Management firm. It will be the only system in place to sell tickets. The implementation must reliably update the systems data with all changes made by all users in real time.

6. Other Requirements/Glossary

Glossary –

- **BeachBurn (BB)** – Annual music festival for which software project has been commissioned
- **BeachBurn Manager (BBM)** – Software project described in this specification document. Used to manage the resources of both ticket sales and band schedules for BeachBurn.
- **ABC Event Management Firm** – Management firm who puts on BeachBurn
- **XYZ club** – Club in Laguna Beach, CA where festival is held
- **Kevin Anteater, LLC** – Company commissioned to develop BBM.
- **Ticket Vendor(s)** – Customer service employee(s) processing sales trans

- **Ticket Manager – Manager who sets the prices for tickets to the event.**
- **Ticket Sales Excel File – Microsoft Excel file containing all ticket sales information. Used as a database of ticket sales information.**
- **Stadium Manager – Responsible for managing and maintaining seats for the**
- **Schedule Manager – Responsible for maintaining the schedule for all bands playing during BeachBurn**
- **Web Administrator – Manages the consumer website for BeachBurn**
- **System Administrator – Manages the BBM system**
- **Arena Map with Ticket Sales – A colored, visual map updated in real time that shows the color code/layout of various sectors as well as which seats are available/unavailable for each day of the festival.**
- **Excel – Software program made by Microsoft**

7. Assumptions / Risks

It is assumed that the users of the software system will have continued access to an internet service provider, in addition to working Windows machines, whenever they need to access the BBM system. It is assumed ABC Management has the capital necessary to maintain constant connections to the network.

The proposed process for selling tickets to customers assumes the availability of display monitors to show the available seats on the arena map to the customer. If this is not possible, that section of the specification will need to be revised.

The system as specified depends on the continued availability/continuity of the Microsoft Excel program for maintaining a log of ticket sales information. This is not expected to be a major concern as it is unlikely that Microsoft will change the Excel software so drastically as to inhibit the needs of BeachBurn Manager.

8. Priorities / Implementation Phases

This section should be used to guide what sections of the project should be worked on first, by order of importance. Order of importance is determined by considering both the demands of the customer and the current timeline of events for BeachBurn.

Must Haves (First):

- Real Time Updating of Stadium Seating Map
- Input functionality for ticket sectors and ticket pricing

Should Have (Second):

- An Excel log of all the ticket sales information.
- Ability to sell and refund tickets.

- Tickets go on sale at a period of time later than the launch of the BeachBurn Manager (currently scheduled for two months later at the time of writing for this specification).

Nice to Have (Third):

- Exception handling for tickets that have already been processed or sold.
 - It is expected that the ticket vendors can help in handling this task. They should be instructed to not sell any tickets which are already marked as sold. In the future there will need to be automated checks against this, especially as the number of seats managed by BBM increases.

9. Future Directions and Expected Changes

In the future, the client has expressed interest in being able to add QR codes or barcodes to the tickets to prevent counterfeit ticket creation. Additionally, expanded capabilities for all possible arenas and stages will be added in future iterations when necessary. As the event is growing in size, this should be expected to occur ASAP.