System Requirements Specification

Theater Ticket System

Version 2.0

October 29, 2014

Prepared By Team D

Kesterson, Andy

Knight, Samantha

Kore, Sumeet

Madyun, Rashad

McComas, Angela

Morse, Anthony

UAH / CS650 / Software Engineering Process

This page intentionally left blank.

Table of Contents

[**Part I** 2](#_Toc402294054)

[1.0 Introduction 2](#_Toc402294055)

[1.1 Purpose of This Document 2](#_Toc402294056)

[1.2 Glossary of Terminology 2](#_Toc402294057)

[2.0 Overall Description 4](#_Toc402294058)

[2.1 Scope 4](#_Toc402294059)

[2.2 System Objective 4](#_Toc402294060)

[2.3 System Goals 4](#_Toc402294061)

[2.3.1 Event Reporting 4](#_Toc402294062)

[2.3.2 Captures All Patron Information 4](#_Toc402294063)

[2.3.3 Multiple Venue Support 4](#_Toc402294064)

[2.3.4 Multiple Event Support 4](#_Toc402294065)

[2.3.5 Event Season Support 4](#_Toc402294066)

[2.4 System Assumptions 4](#_Toc402294067)

[2.5 System Constraints 5](#_Toc402294068)

[3.0 Requirements 6](#_Toc402294069)

[3.1 CSA Requirements 6](#_Toc402294070)

[3.2 TTS Requirements 6](#_Toc402294071)

[4.0 Models and Diagrams 9](#_Toc402294072)

[4.1 Use Case Diagram – TTS Level 0 9](#_Toc402294073)

[4.1.1 Reserve Ticket Use Case 9](#_Toc402294074)

[4.1.2 Purchase Ticket 13](#_Toc402294075)

[4.1.3 Pick Up Ticket 14](#_Toc402294076)

[4.1.4 Select Unpaid Reservation 15](#_Toc402294077)

[4.1.5 Validate Patron Payment 16](#_Toc402294078)

[4.1.6 Search Event by Venue 17](#_Toc402294079)

[4.1.7 Patron Request Refund 19](#_Toc402294080)

[4.1.8 Refund All Tickets 21](#_Toc402294081)

[4.1.9 Create New Patron 23](#_Toc402294082)

[4.1.10 Exchange Ticket 24](#_Toc402294083)

[4.2 Function Point Estimates 25](#_Toc402294084)

[**Part II** 30](#_Toc402294085)

[1.0 How We Operate 30](#_Toc402294086)

[1.1 Roles 30](#_Toc402294087)

[1.2 Tools Used 30](#_Toc402294088)

[1.3 Team Communication Strategy 30](#_Toc402294089)

[1.4 Team Quality Assurance Strategy 30](#_Toc402294090)

[1.4.1 Preliminary Quality Assurance Audit 31](#_Toc402294091)

[1.4.2 Preliminary CM Audit 32](#_Toc402294092)

[1.5 Project Schedule 32](#_Toc402294093)

[1.6 Meeting Minutes 34](#_Toc402294094)

[**Appendix A** 36](#_Toc402294095)

**Index of Tables**

[Table 1.2‑1 Glossary 2](#_Toc402294036)

[Table 3.1‑1 CSA Requirements 6](#_Toc402294037)

[Table 3.2‑1 TTS Requirements 6](#_Toc402294038)

[Table 3.2‑2 TTS Interface Requirements 7](#_Toc402294039)

[Table 4.1‑1 Reserve Ticket Use Case 9](#_Toc402294040)

[Table 4.1‑2 Reserve Ticket Function Point Estimation 12](#_Toc402294041)

[Table 4.1‑3 Purchase Ticket Use Case 13](#_Toc402294042)

[Table 4.1‑4 Pick Up Ticket Use Case 14](#_Toc402294043)

[Table 4.1‑5 Select Unpaid Reservation Ticket Use Case 15](#_Toc402294044)

[Table 4.1‑6 Validate Patron Payment Ticket Use Case 16](#_Toc402294045)

[Table 4.1‑7 Search Event by Venue Ticket Use Case 17](#_Toc402294046)

[Table 4.1‑8 Patron Request Refund Ticket Use Case 19](#_Toc402294047)

[Table 4.1‑9 Refund All Tickets Ticket Use Case 21](#_Toc402294048)

[Table 4.1‑10 Create New Patron Ticket Use Case 23](#_Toc402294049)

[Table 4.1‑11 Exchange Ticket Use Case 24](#_Toc402294050)

[Table 1.1‑1 Team Member Roles 30](#_Toc402294051)

[Table 1.5‑1 Theater Ticket System Project Schedule 32](#_Toc402294052)

[Table 1.6‑1 Team Meeting Minutes 34](#_Toc402294053)

**Index of Figures**

[Figure 4.1‑1 TTS Level 0 Use Case 9](#_Toc402294015)

[Figure 4.1‑2 Reserve Ticket Sequence Diagram 11](#_Toc402294016)

[Figure 4.1‑3 Reserve Ticket Function Point Diagram 12](#_Toc402294017)

[Figure 4.1‑4 Purchase Ticket Sequence Diagram - Cash 13](#_Toc402294018)

[Figure 4.1‑5 Purchase Ticket Sequence Diagram - Credit 14](#_Toc402294019)

[Figure 4.1‑6 Select Unpaid Reservation Sequence Diagram 16](#_Toc402294020)

[Figure 4.1‑7 Search Event by Venue Activity Diagram 18](#_Toc402294021)

[Figure 4.1‑8 Search Event by Venue Sequence Diagram 19](#_Toc402294022)

[Figure 4.1‑9 Patron Request Refund Activity Diagram 20](#_Toc402294023)

[Figure 4.1‑10 Patron Request Refund Sequence Diagram 21](#_Toc402294024)

[Figure 4.1‑11 Refund All Tickets Activity Diagram 22](#_Toc402294025)

[Figure 4.1‑12 Refund All Tickets Sequence Diagram 23](#_Toc402294026)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision | Change Description | Name |
| 2014/10/08 | 1.0 | Initial document creation | A. Kesterson |
| 2014/10/29 | 2.0 | The second deliverable of this document includes additional use cases with activity diagrams and sequence diagrams. New requirements were added for the new use cases. New function point diagrams and associated function point estimations were added. A QA audit of progress was performed and the audit report is included. The tasks needed for this deliverable and the final deliverable are included in the project schedule. | A. McComas |
|  |  |  |  |
|  |  |  |  |

**Part I**

# Introduction

## Purpose of This Document

The purpose of this document is to detail the requirement specifications for the Theater Ticket System (TTS). This document is intended to outline the requirements of the system as a whole and display the desired functionality. The document itself is composed of two parts which are further divided into specific areas. Part one will outline the system requirements and estimations. Part two will outline the planned process operations, collaboration, and quality assurance strategies.

## Glossary of Terminology

Table 1.2‑1 Glossary

|  |  |
| --- | --- |
| Term | Definition |
| Book | Payment has been made for a reservation. |
| Booked Seat | A particular seat at an event which is associated with a patron and has been paid for. |
| Customer Service Agent (CSA) | A person who reserves a ticket for the patron and enters information in the machine. |
| Deactivate | Set a record to a state where it will not be returned by a query, but still exists for record keeping. |
| Delete | Permanently remove a record from the database. |
| Event | A purposed meeting for which a venue is used. |
| Event Series | A group of sequenced events. Such as a symphony season, or a summer concert series. |
| Locked Seat | A particular seat for an event which is in the process of being reserved but which the system has not yet been able to associate a patron. |
| Patron | A person who contacts the CSA to book a ticket. |
| Payment Gateway | An e-commerce application service provider that authorizes credit card payments for e-businesses. |
| Reserved Seat | A particular seat at an event which is associated with a patron but has not yet been paid for. |
| Theater Ticket Database | An application which stores information regarding ticket purchase and patron details. |
| Unlocked Seat | A particular seat at an event which has not been purchased by a patron and is not in the process of being reserved. |
| Venue | A location where an event takes place. |

## Referenced Documents

Table 1.3‑1 Referenced Documents

|  |  |  |
| --- | --- | --- |
| Date | Document Name | Source |
| 2014/10/28 | Reuse in Use Case Models | http://agilemodeling.com/essays/useCaseReuse.htm |
|  |  |  |

# Overall Description

## Scope

The TTS manages the different tasks involved with handling ticket sales for an event at a specified venue. A customer service agent (CSA) interacts with the TTS to handle the ticket sales, ticket reservation, or ticket exchanges for a patron either by telephone, in person prior to an event, or at the event’s venue. The TTS will also allow the CSA to sell a patron season tickets or to track patron information for customized ticket purchases. The TTS will also track ticket purchase information, reserved seating, custom patron data, and season ticket status in a remote database. The database should be available for access at both the CSA’s office site and any event venue ticket location.

## System Objective

The objective of the TTS is to consolidate the management of event ticket sales for a variety of different events and venues. The system will meet this objective by providing the user of the software various ways to query ticket and event-based information, input reservations, and perform various ticket sales-related functionality.

## System Goals

### Event Reporting

The TTS will support the capability for CSAs to run various types of reporting metrics about ticket sales. This will allow them to determine answers to questions such as "What percentage of seats were sold for Event X?”

### Captures All Patron Information

The TTS will store customer information. This will allow a CSA to more easily assist repeat patrons without requiring them to provide all required information every time they wish to purchase tickets.

### Multiple Venue Support

The TTS will support multiple types of venues where events will take place. Different venues have different physical locations and seating capabilities. The system will use this information to properly allow seat reservation by the CSA for the patron.

### Multiple Event Support

The TTS will support multiple events. Each event occurs at a particular venue, date, and time. The system will support querying for events so the CSA can easily place reservations for the patrons.

### Event Season Support

The TTS will support multiple related events in what is called an event season. Grouping a series of events into seasons allows the CSA to more easily reserve tickets on behalf of a patron for all related events.

## System Assumptions

Several assumptions about the system were identified:

1. The system is up and running.
2. Tickets are available for the desired event.
3. There is a working telephone line at the organization’s office.
4. There is proper connectivity to all third party systems.
5. There are CSAs available to work telephone lines and any event ticket booth.
6. The patron can provide a valid form of payment.
7. There is a third party payment system available to handle non cash payments.
8. Operating system versions are compatible.
9. A third party database system will be utilized to handle persistent data for the TTS.
10. A ticket booth at an event will have a data connection to the database.
11. The third party system will have an Application Programming Interface (API) to allow the capability to process patron charges through the third parties system.

## System Constraints

1. The TTS software must operate on a typical Windows machine that can be used at both the organization’s office and at any ticket booth of an event.
2. An external system will handle the management of events in the database. Items such as creating an event, cancelling an event, setting the maximum event tickets, etc. would all be handled by this external system.

# Requirements

## CSA Requirements

The importance of defining CSA requirements is to identify the minimum functionality to be provided by the system for the benefit of the CSA.

Table 3.1‑1 CSA Requirements

|  |  |
| --- | --- |
| CSA  Requirement  Number | Requirements Description |
|  | The CSA shall have the capability to enter customer data. |
|  | The CSA shall have access to theater venues. |
|  | The CSA shall have access to theater events. |
|  | The CSA shall have access to seat availability. |
|  | The CSA shall be able to book reserved seating. |
|  | The CSA shall be able to book general admission seating. |
|  | The CSA shall be able to accept patron payments. |
|  | The CSA shall be able to exchange tickets. |
|  | The CSA shall be able to refund tickets. |
|  | The CSA shall be able to book season tickets. |
|  | The CSA shall be able to book VIP seating. |
|  | The CSA shall be able to book special accommodation seating. |

## TTS Requirements

The TTS requirements refinement process examines each CSA requirement to see if it meets the characteristics of a good requirement. Each CSA requirement will be decomposed into a refined set of requirements. Newly derived requirements are created from this process, which continues until all requirements are defined, analyzed, and the final project architecture is defined.

Table 3.2‑1 TTS Requirements

|  |  |
| --- | --- |
| TTS  Requirement  Number | Requirements Description |
| REQ100 | The system shall have the capability to search for events. |
| REQ101 | The system shall have the capability to search for event seats. |
| REQ102 | The system shall have the capability to verify seat availability. |
| REQ103 | The system shall have the capability to lock a seat. |
| REQ104 | The system shall have the capability to search for a patron. |
| REQ105 | The system shall have the capability to verify the selected patron. |
| REQ106 | The system shall have the capability to associate a patron with a seat. |
| REQ107 | The system shall have the capability to select reserved seat. |
| REQ108 | The system shall have the capability to select general seat. |
| REQ109 | The system shall have the capability to search for venues. |
| REQ110 | The system shall have the capability to search for events by venue. |
| REQ111 | The system shall have the capability to filter events for a venue by a date range. |
| REQ112 | The system shall have the capability to search for event bookings |
| REQ113 | The system shall have the capability to verify an event booking can be refunded. |
| REQ114 | The system shall have the capability to unlock a seat. |
| REQ115 | The system shall have the capability to request a refund from the third-party payment gateway. |
| REQ117 | The system shall have the capability to search for event seats. |
| REQ121 | The system shall have the capability to add a new patron. |
| REQ122 | The system shall have the capability to update information of a patron. |
| REQ123 | The system shall have the capability to delete a patron. |
| REQ127 | The system shall have the capability to search for a reservation by number. |
| REQ128 | The system shall have the capability to search for a reservation by patron. |
| REQ129 | The system shall have the capability to select an existing reservation. |
| REQ130 | The system shall have the capability to deactivate an existing reservation. |

Table 3.2‑2 TTS Interface Requirements

|  |  |
| --- | --- |
| TTS Interface  Requirement  Number | Requirements Description |
| RINT100 | The system shall have an interface to search for events. |
| RINT101 | The system shall have an interface to search for event seats. |
| RINT102 | The system shall have an interface to display search results (events, event seats). |
| RINT103 | The system shall have an interface to select an event seat. |
| RINT104 | The system shall have an interface to display that a seat is locked. |
| RINT105 | The system shall have an interface to request patron info. |
| RINT106 | The system shall have an interface to display patron search results. |
| RINT107 | The system shall have an interface to allow the CSA to select a patron. |
| RINT108 | The system shall have an interface to allow the CSA to submit the selected patron. |
| RINT109 | The system shall have an interface to display reserved status. |
| RINT110 | The system shall have an interface to search for venues. |
| RINT111 | The system shall have an interface to display venue results. |
| RINT112 | The system shall have an interface to select a venue (??? “from venue results” ???) |
| RINT113 | The system shall have an interface to display event results by venue. |
| RINT114 | The system shall have an interface to input a date range used to filter event results by venue. |
| RINT115 | The system shall have an interface to search for event bookings. |
| RINT116 | The system shall have an interface to display event bookings. |
| RINT117 | The system shall have an interface to select an event booking (?? “from event booking results” ???) |
| RINT118 | The system shall have an interface to display a refund confirmation. |
| RINT119 | The system shall have an interface to select all event bookings for an event. |
| RINT122 | The system shall have the interface to verify seat availability. |
| RINT124 | The system shall have the interface to search for a patron. |
| RINT125 | The system shall have the interface to add a new patron. |
| RINT126 | The system shall have the interface to update information of a patron. |
| RINT127 | The system shall have the interface to delete a patron. |
| RINT132 | The system shall have an interface to search for a reservation by number. |
| RINT133 | The system shall have an interface to search for a reservation by patron. |
| RINT134 | The system shall have an interface to display reservation search results by number. |
| RINT135 | The system shall have an interface to display reservation search results by patron. |
| RINT136 | The system shall have an interface to allow the CSA to select an existing reservation. |
|  | The system shall have an interface to allow the CSA to select an existing by patron. |
| RINT137 | The system shall have an interface to allow the CSA to deactivate an existing reservation. |

# Models and Diagrams

## E:\UAH_CS650\uah\cs650\figures\TTS_UC_Diagram_Lvl_0.pngUse Case Diagram – TTS Level 0

Figure 4.1‑1 TTS Level 0 Use Case

### Reserve Ticket Use Case

#### Use Case Description

Table 4.1‑1 Reserve Ticket Use Case

|  |  |  |
| --- | --- | --- |
| UC Name | Reserve Ticket | |
| Description | The Reserve Ticket Use Case describes the process which will allow a CSA to reserve a ticket for a patron. | |
| Actors | Customer Service Agent (CSA) | |
| Pre-Conditions | A CSA is available.  The TTS is operational.  The CSA is logged in to TTS. | |
| Post-Conditions | A ticket is reserved for a patron.  The selected tickets are no longer available for other patrons. | |
| Triggers | A patron wishes to reserve a ticket. | |
| Flow | | |
|  | Actor | System |
|  | Query seats for a specific event |  |
|  |  | Search for a specific Event. |
|  |  | Return results available seats. |
|  | Select a seat |  |
|  |  | Verify seat is available, and lock selected seat. |
|  | Query for the patron’s information. |  |
|  |  | Search for the patron. |
|  |  | Return results for the searched patron. |
|  | Select the correct patron. |  |
|  |  | Verify selected patron. |
|  |  | Associate locked seat with selected patron. |
| Exceptions | The ticket(s) for the desired event are sold out.  The patron does not exist within the system.  A selected seat is not available when selected.  The desired event is not available or does not exist within the Event database. | |
| Extension Points | TBD | |

#### Activity Diagram

TBD

#### Sequence Diagram

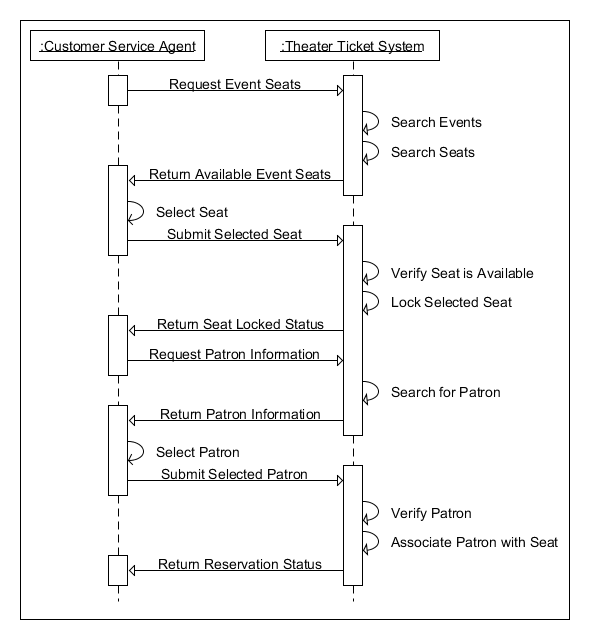


Figure 4.1‑2 Reserve Ticket Sequence Diagram

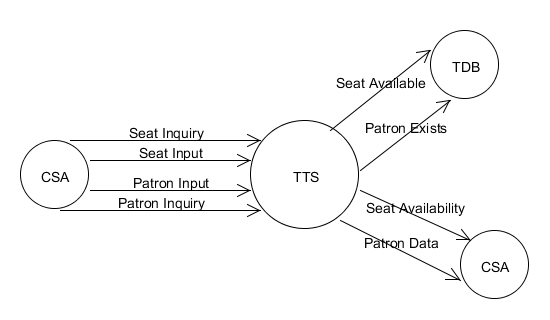


Figure 4.1‑3 Reserve Ticket Function Point Diagram

Table 4.1‑2 Reserve Ticket Function Point Estimation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Simple | Average | Complex | Count |
| External Inputs | 2 | 3 | 4 | 6 | 8 |
| External Outputs | 2 | 4 | 5 | 7 | 10 |
| External Inquiry | 2 | 3 | 4 | 6 | 8 |
| Internal Logical Files | 0 | 7 | 10 | 15 | 0 |
| External Interface Files | 2 | 5 | 7 | 10 | 14 |
|  | | | **Count Sub Total** | | 40 |
| **∑F** | | 38 |
| **FP Total** | | 42 |

### Purchase Ticket

#### Use Case Description

Table 4.1‑3 Purchase Ticket Use Case

|  |  |  |
| --- | --- | --- |
| UC Name | Purchase Ticket | |
| Description | The Purchase Ticket Use Case describes the process which will allow a CSA to purchase a ticket for a patron. | |
| Actors | Customer Service Agent (CSA) | |
| Pre-Conditions | A CSA is available.  The TTS is operational.  The CSA is logged in to TTS.  A reservation is selected.  The patron has valid payment.  The patron has full payment. | |
| Post-Conditions | A ticket is fully paid for by patron. | |
| Triggers | A patron wishes to pay for a ticket. | |
| Flow | | |
|  | Actor | System |
|  | Selects payment type. |  |
|  | Submit patron payment. |  |
|  |  | Accept cash payment. Or Validate Patron Payment. |
|  |  | Return payment accepted. |
| Exceptions | | |
| Extension Points | Validate Patron Payment | |

#### Activity Diagram

TBD

#### Sequence Diagram

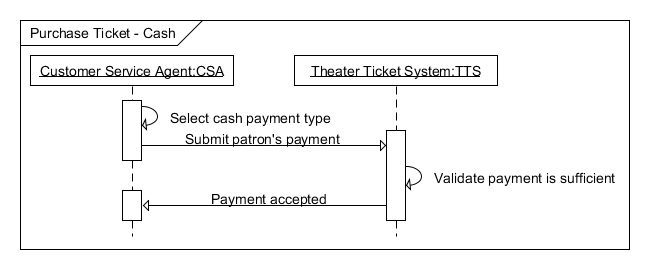


Figure 4.1‑4 Purchase Ticket Sequence Diagram - Cash

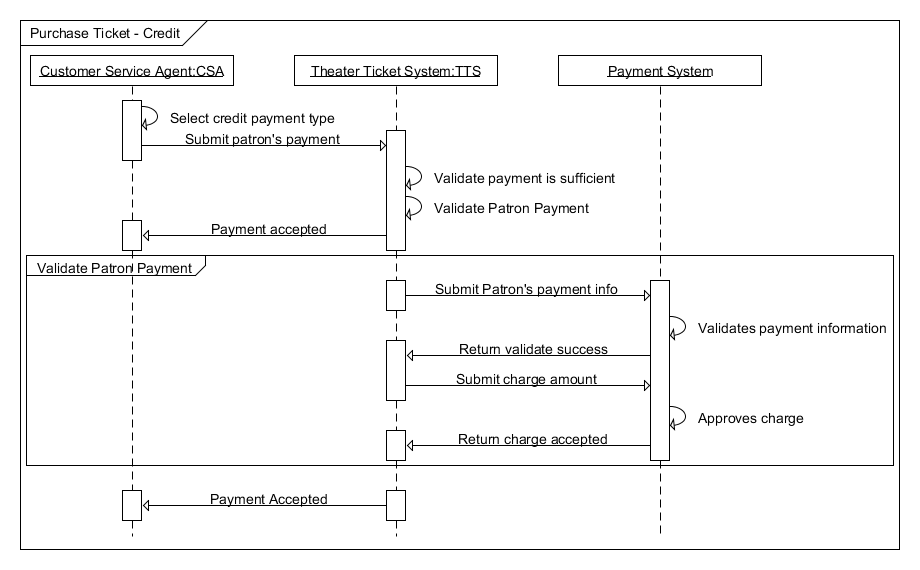


Figure 4.1‑5 Purchase Ticket Sequence Diagram - Credit

### Pick Up Ticket

#### Use Case Description

Table 4.1‑4 Pick Up Ticket Use Case

|  |  |  |
| --- | --- | --- |
| UC Name | Pick Up Ticket | |
| Description | The Pickup Ticket Use Case describes the process which will allow a patron to pick up a ticket. | |
| Actors | Customer Service Agent (CSA) | |
| Pre-Conditions | A CSA is available.  The TTS is operational.  The CSA is logged in to TTS.  A ticket is booked for a patron. | |
| Post-Conditions | A ticket is printed for a patron. | |
| Triggers | A patron wishes to pick up a ticket. | |
| Flow | | |
|  | Actor | System |
|  | Agent searches for patron’s booking |  |
|  |  | Looks up the booking for the patron |
|  |  | Returns available tickets. |
|  | Selects the ticket(s) to print. |  |
|  |  | Prints the selected ticket(s), and stores the number of tickets that have been printed. |
| Exceptions | No tickets found.  No reservations found.  Reservation not paid. | |
| Extension Points | TBD | |

#### Activity Diagram

TBD

#### Sequence Diagram

TBD

### Select Unpaid Reservation

#### Use Case Description

Table 4.1‑5 Select Unpaid Reservation Use Case

|  |  |  |
| --- | --- | --- |
| UC Name | Select Unpaid Reservation | |
| Description | The Select Unpaid Reservation Ticket Use Case describes the process which will allow a CSA to select an unpaid ticket for a patron. | |
| Actors | Customer Service Agent (CSA) | |
| Pre-Conditions | A CSA is available.  The TTS is operational.  The CSA is logged in to TTS.  The ticket has been reserved. | |
| Post-Conditions | Reservation is selected for payment. | |
| Triggers | A patron wishes to select an unpaid ticket. | |
| Flow | | |
|  | Actor | System |
|  | The CSA searches for a patron. |  |
|  |  | Finds the patron and returns unpaid reservations. |
|  | Selects an unpaid reservation. |  |
| Exceptions | The patron cannot be found within the system.  The reservation cannot be found within the system. | |
| Extension Points | TBD | |

#### Activity Diagram

TBD

#### Sequence Diagram

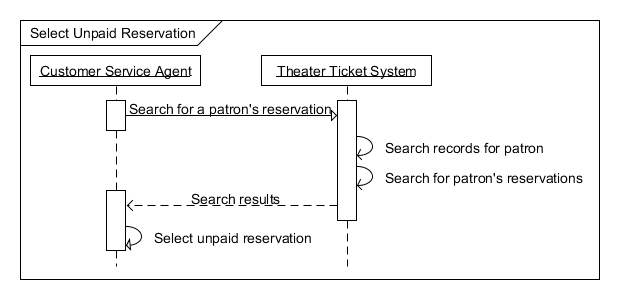


Figure 4.1‑6 Select Unpaid Reservation Sequence Diagram

### Validate Patron Payment

#### Use Case Description

Table 4.1‑6 Validate Patron Payment Use Case

|  |  |  |
| --- | --- | --- |
| UC Name | Validate Patron Payment | |
| Description | The Validate Patron Payment Ticket Use Case describes the process which will allow the TTS to validate the patron’s payment with the payment system. | |
| Actors | Theater Ticket System (TTS) | |
| Pre-Conditions | The TTS is operational.  Patron has submitted payment information.  Payment system is available. | |
| Post-Conditions | Payment method validated. | |
| Triggers | Patron wishes to purchase a ticket with a non-cash payment. | |
| Flow | | |
|  | Actor | Payment System |
|  | Submit patron’s payment information. |  |
|  |  | Validate payment information and return information accepted. |
|  | Submit the charge amount. |  |
|  |  | Accept charge and return payment accepted. |
| Exceptions | The patron information is not valid.  The charge amount is not accepted. | |
| Extension Points | TBD | |

#### Activity Diagram

TBD

#### Sequence Diagram

TBD

### Search Event by Venue

#### Use Case Description

Table 4.1‑7 Search Event by Venue Use Case

|  |  |  |
| --- | --- | --- |
| UC Name | Search Event by Venue | |
| Description | The Search for Event by Venue Use Case describes the process which will allow a CSA to search for events for a particular venue during a specified range of dates for a Patron. | |
| Actors | Customer Service Agent (CSA) | |
| Pre-Conditions | A CSA is available.  The TTS is operational.  The CSA is logged in to TTS. | |
| Post-Conditions | The patron’s query is answered. | |
| Triggers | A patron wants to know what events are scheduled for a particular Venue during a particular date range. | |
| Flow | | |
|  | Actor | System |
|  | Query for a specific venue |  |
|  |  | Search for a specific venue |
|  |  | Return venue results |
|  | Select a venue |  |
|  |  | Search for events for selected venue |
|  |  | Return event results for selected venue |
|  | Query for date range |  |
|  |  | Filter events for selected venue by date range |
|  |  | Return event results for selected venue and requested date range |
| Exceptions |  | |
| Extension Points | TBD | |

#### Activity Diagram

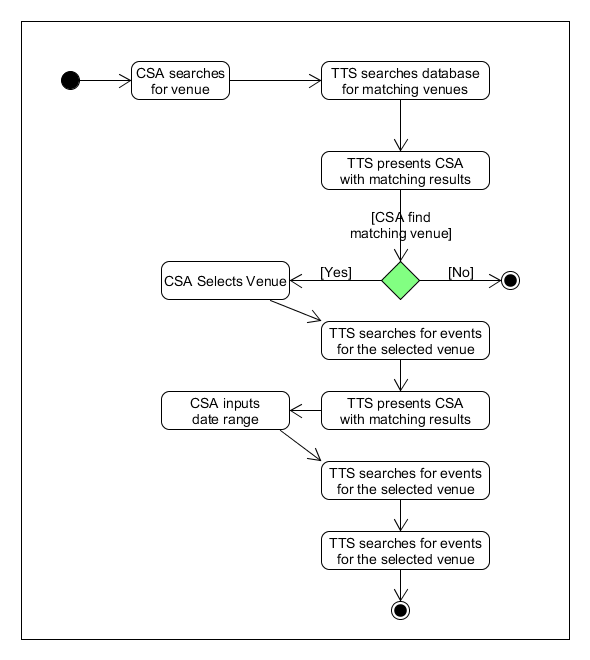


Figure 4.1‑7 Search Event by Venue Activity Diagram

#### Sequence Diagram

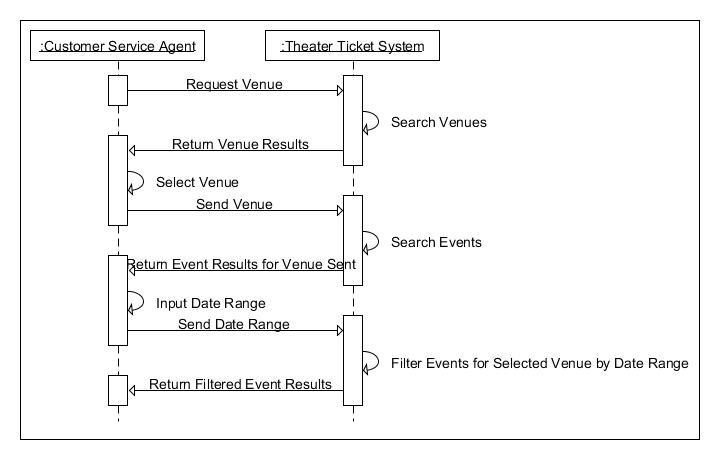


Figure 4.1‑8 Search Event by Venue Sequence Diagram

### Patron Request Refund

#### Use Case Description

Table 4.1‑8 Patron Request Refund Ticket Use Case

|  |  |  |
| --- | --- | --- |
| UC Name | Patron Request Refund Ticket | |
| Description | The Patron Requested Refund Ticket User Case describes the process which will allow a CSA to refund a purchased ticket for a patron, which will also unlock the booked seat associated with the purchased ticket for later reservations. | |
| Actors | Customer Service Agent (CSA) | |
| Pre-Conditions | A CSA is available.  The TTS is operational.  The CSA is logged in to TTS.  Patron has already purchased a ticket for a seat at an event. | |
| Post-Conditions | The third-party payment gateway returns a response to requests for refunds in real-time. | |
| Triggers | The booked seat is unlocked. | |
| Flow | | |
|  | Actor | System |
|  | Query for a specific event booking |  |
|  |  | Search for a specific event booking |
|  |  | Return booking results |
|  | Select booking to refund |  |
|  |  | Verify booking |
|  |  | Return seat to an unlocked state |
|  |  | Send request to third-party payment gateway for refund of purchase amount |
|  |  | Await third-party payment gateway response |
|  |  | Return refund confirmation results |
| Exceptions |  | |
| Extension Points | TBD | |

#### Activity Diagram

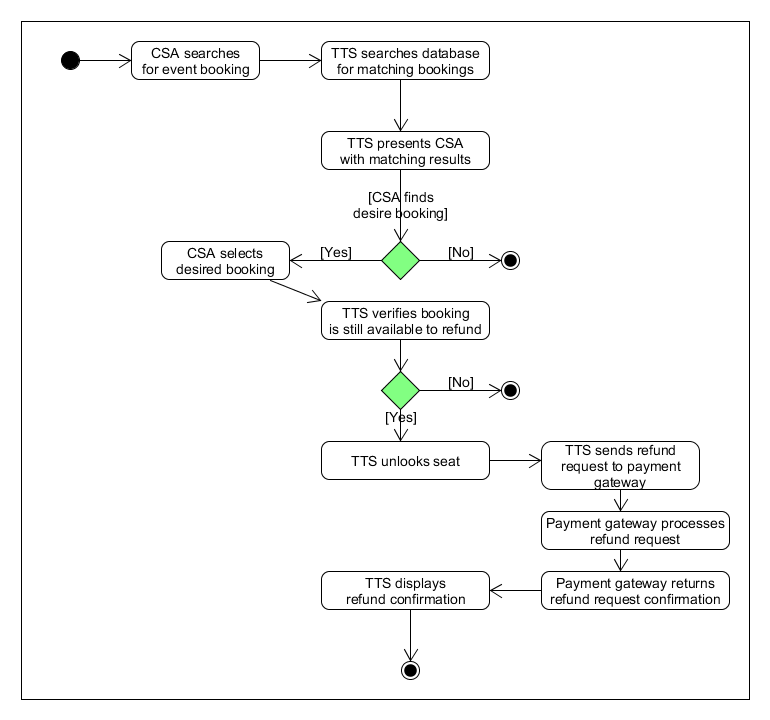


Figure 4.1‑9 Patron Request Refund Activity Diagram

#### Sequence Diagram

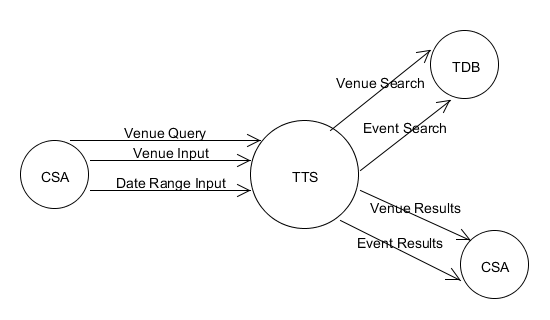


Figure 4.1‑10 Patron Request Refund Sequence Diagram

### Refund All Tickets

#### Use Case Description

Table 4.1‑9 Refund All Tickets Ticket Use Case

|  |  |  |
| --- | --- | --- |
| UC Name | Refund All Tickets Ticket | |
| Description | The Refund All Tickets for an Event User Case describes the process which will allow a CSA to refund all purchased tickets for an event, which will also unlock all the booked seats associated with the event. | |
| Actors | Customer Service Agent (CSA) | |
| Pre-Conditions | A CSA is available.  The TTS is operational.  The CSA is logged in to TTS.  Patron has already purchased a ticket for a seat at an event. | |
| Post-Conditions | The third-party payment gateway returns a response to requests for refunds in real-time. | |
| Triggers | All seats for the event are unlocked | |
| Flow | | |
|  | Actor | System |
|  | Query for a specific event booking |  |
|  |  | Search for a specific event booking |
|  |  | Return booking results |
|  | Select all bookings for an event |  |
|  |  | Verify all bookings for the selected event |
|  |  | Return all seats for the selected event to an unlocked state |
|  |  | Send all requests to third-party payment gateway for refund of the purchase amount for each individual booking |
|  |  | Await third-party payment gateway response |
|  |  | Return all confirmation results |
| Exceptions |  | |
| Extension Points | TBD | |

#### Activity Diagram

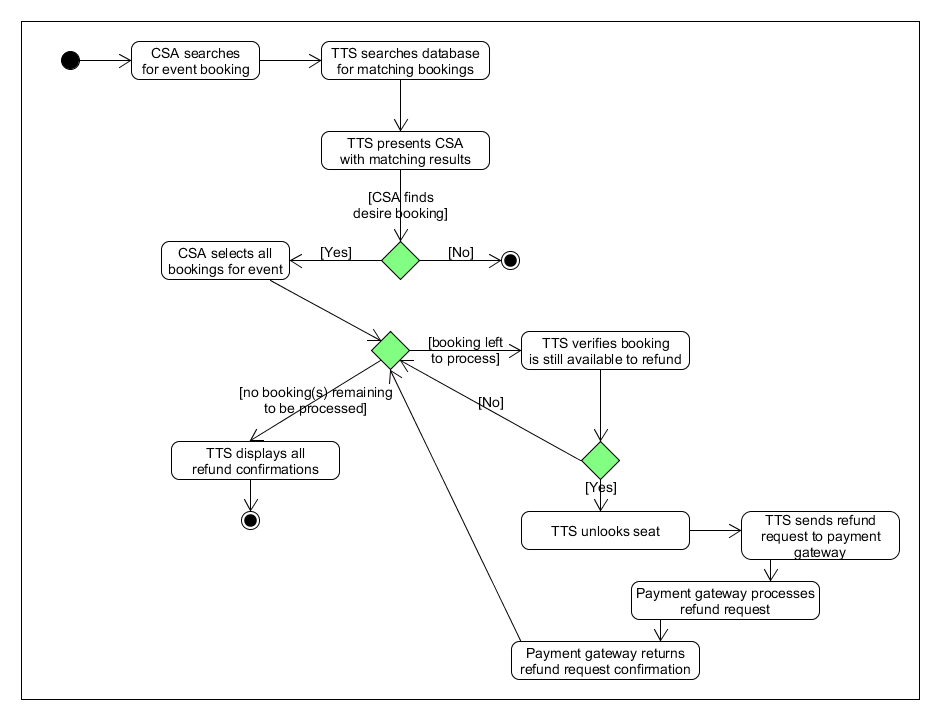


Figure 4.1‑11 Refund All Tickets Activity Diagram

#### Sequence Diagram

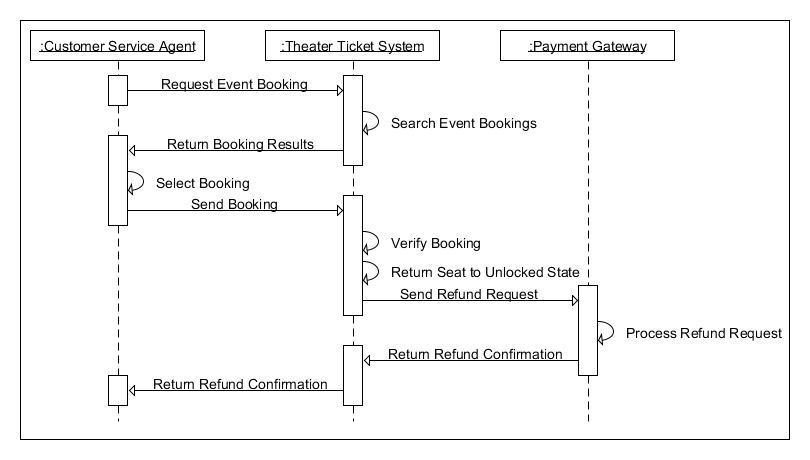


Figure 4.1‑12 Refund All Tickets Sequence Diagram

### Create New Patron

#### Use Case Description

Table 4.1‑10 Create Patron Use Case

|  |  |  |
| --- | --- | --- |
| UC Name | Create Patron | |
| Description | Creating a new patron. | |
| Actors | Customer Service Agent (CSA) | |
| Pre-Conditions | A CSA is available.  The TTS is operational.  The CSA is logged in to TTS.  A patron does not exists. | |
| Post-Conditions | A new patron is created. | |
| Triggers | A patron needs to be added in the system. | |
| Flow | | |
|  | Actor | System |
|  | CSA enters information of new patron. |  |
|  | CSA needs to submit information to the TTS. |  |
|  |  | Verify information of the patron and save. |
|  |  | Notify that patron has saved successfully. |
| Exceptions | Information of two patrons match. | |
| Extension Points | TBD | |

#### Activity Diagram

TBD

#### Sequence Diagram

TBD

### Exchange Ticket

#### Use Case Description

Table 4.1‑11 Exchange Ticket Use Case

|  |  |  |
| --- | --- | --- |
| UC Name | Exchange Ticket | |
| Description | The Exchange Ticket Use Case describes the process which will allow a CSA to exchange a ticket for a patron who has an existing reservation. | |
| Actors | Customer Service Agent (CSA) | |
| Pre-Conditions | A CSA is available.  The TTS is operational.  The CSA is logged in to TTS.  A valid reservation exists for the patron.  Patron has already been verified. | |
| Post-Conditions | A ticket is exchanged for a patron.  A ticket is not exchanged and patron keeps existing reservation. | |
| Triggers | A patron wishes to exchange a ticket. | |
| Flow | | |
|  | Actor | System |
|  | Query for existing reservation number. |  |
|  |  | Search for a reservation by number. |
|  |  | Return reservation and patron info. |
|  | Select correct reservation/patron. |  |
|  |  | Search for event. |
|  |  | Return results available seats. |
|  | Select seat. |  |
|  | Submit exchange selection. |  |
|  |  | Verify seat is available, and lock selected seat. |
|  |  | Associate locked seat with selected patron. |
|  | Remove selected reservation. |  |
|  |  | Send confirmation of reservation deletion. |
| Exceptions | The reservation does not exist or cannot be found.  The event does not exist, or cannot be found.  Tickets for the event are sold out.  The selected seat is unavailable. | |
| Extension Points | TBD | |

#### Activity Diagram

TBD

#### Sequence Diagram

TBD

## Function Point Estimates

Purchase Ticket

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Simple | Average | Complex | Count |
| External Inputs | 2 | 3 | 4 | 6 | 8 |
| External Outputs | 1 | 4 | 5 | 7 | 5 |
| External Inquiry | 0 | 3 | 4 | 6 | 0 |
| Internal Logic Files | 0 | 7 | 10 | 15 | 0 |
| External Interface Files | 3 | 5 | 7 | 10 | 21 |
|  | | | **Count Sub Total** | | 34 |
| **∑F** | | 38 |
| **FP Total** | | 36 |

Pickup Ticket

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Simple | Average | Complex | Count |
| External Inputs | 2 | 3 | 4 | 6 | 8 |
| External Outputs | 1 | 4 | 5 | 7 | 5 |
| External Inquiry | 1 | 3 | 4 | 6 | 4 |
| Internal Logic Files | 0 | 7 | 10 | 15 | 0 |
| External Interface Files | 1 | 5 | 7 | 10 | 7 |
|  | | | **Count Sub Total** | | 24 |
| **∑F** | | 34 |
| **FP Total** | | 24 |

Select Unpaid Reservation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Simple | Average | Complex | Count |
| External Inputs | 3 | 3 | 4 | 6 | 12 |
| External Outputs | 3 | 4 | 5 | 7 | 15 |
| External Inquiry | 3 | 3 | 4 | 6 | 12 |
| Internal Logic Files | 0 | 7 | 10 | 15 | 0 |
| External Interface Files | 3 | 5 | 7 | 10 | 21 |
|  | | | **Count Sub Total** | | 60 |
| **∑F** | | 37 |
| **FP Total** | | 62 |

Validate Patron Payment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Simple | Average | Complex | Count |
| External Inputs | 2 | 3 | 4 | 6 | 8 |
| External Outputs | 2 | 4 | 5 | 7 | 10 |
| External Inquiry | 0 | 3 | 4 | 6 | 0 |
| Internal Logic Files | 0 | 7 | 10 | 15 | 0 |
| External Interface Files | 3 | 5 | 7 | 10 | 21 |
|  | | | **Count Sub Total** | | 39 |
| **∑F** | | 37 |
| **FP Total** | | 41 |

Search for Event by Venue

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Simple | Average | Complex | Count |
| External Inputs | 2 | 3 | 4 | 6 | 8 |
| External Outputs | 2 | 4 | 5 | 7 | 10 |
| External Inquiry | 1 | 3 | 4 | 6 | 40 |
| Internal Logic Files | 0 | 7 | 10 | 15 | 0 |
| External Interface Files | 2 | 5 | 7 | 10 | 14 |
|  | | | **Count Sub Total** | | 36 |
| **∑F** | | 31 |
| **FP Total** | | 35 |

Patron Requested Refund Ticket

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Simple | Average | Complex | Count |
| External Inputs | 1 | 3 | 4 | 6 | 4 |
| External Outputs | 2 | 4 | 5 | 7 | 10 |
| External Inquiry | 3 | 3 | 4 | 6 | 8 |
| Internal Logic Files | 0 | 7 | 10 | 15 | 0 |
| External Interface Files | 2 | 5 | 7 | 10 | 14 |
|  | | | **Count Sub Total** | | 36 |
| **∑F** | | 36 |
| **FP Total** | | 37 |

Refund All Tickets for an Event

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Simple | Average | Complex | Count |
| External Inputs | 1 | 3 | 4 | 6 | 4 |
| External Outputs | 2 | 4 | 5 | 7 | 10 |
| External Inquiry | 3 | 3 | 4 | 6 | 8 |
| Internal Logic Files | 0 | 7 | 10 | 15 | 0 |
| External Interface Files | 2 | 5 | 7 | 10 | 14 |
|  | | | **Count Sub Total** | | 36 |
| **∑F** | | 36 |
| **FP Total** | | 37 |

Create New Patron

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Simple | Average | Complex | Count |
| External Inputs | 4 | 3 | 4 | 6 | 16 |
| External Outputs | 2 | 4 | 5 | 7 | 10 |
| External Inquiry | 2 | 3 | 4 | 6 | 8 |
| Internal Logic Files | 0 | 7 | 10 | 15 | 0 |
| External Interface Files | 0 | 5 | 7 | 10 | 0 |
|  | | | **Count Sub Total** | | 34 |
| **∑F** | | 34 |
| **FP Total** | | 34 |

Exchange Ticket

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Simple | Average | Complex | Count |
| External Inputs | 3 | 3 | 4 | 6 | 12 |
| External Outputs | 3 | 4 | 5 | 7 | 15 |
| External Inquiry | 2 | 3 | 4 | 6 | 8 |
| Internal Logic Files | 0 | 7 | 10 | 15 | 0 |
| External Interface Files | 3 | 5 | 7 | 10 | 21 |
|  | | | **Count Sub Total** | | 56 |
| **∑F** | | 46 |
| **FP Total** | | 63 |

Theater Ticket Function Point Total Analysis Estimate

|  |  |
| --- | --- |
| Use Case | FP Total |
| Reserve Ticket | 42 |
| Purchase Ticket | 36 |
| Pickup Ticket | 24 |
| Select Unpaid Reservation | 62 |
| Validate Patron Payment | 41 |
| Search for Event by Venue | 35 |
| Patron Requested Refund Ticket | 37 |
| Refund All Tickets for an Event | 37 |
| Create New Patron | 34 |
| Exchange Ticket | 63 |
| Total Count | 411 |

This page intentionally left blank.

**Part II**

# How We Operate

## Roles

Table 1.1‑1 Team Member Roles

|  |  |
| --- | --- |
| Role | Team Member |
| Scribe | Samantha, Anthony (alt) |
| Submitter | Samantha |
| Meeting Scheduler | Angela |
| UML Modeler | Andy, Sumeet |
| Task Coordinator | Angela |
| Configuration Manager | Andy |
| Quality Assurance Manager | Rashad |
| Reviewer | All |
| Tester | All |
| Use Case Development | All |

## Tools Used

The team will utilize git for documentation version control, and UMLet to develop UML diagrams.

The team plans to utilize the services of GitHub for hosting the change tracking and version control repository.

## Team Communication Strategy

The team has decided to meet weekly on Tuesdays and Wednesdays at 7:00 pm at the university. Telephone conferences may be scheduled as necessary. The team will also use Angel and UAH email for general communication during the semester.

## Team Quality Assurance Strategy

The Quality Assurance (QA) plan describes how the team will implement a procedure to ensure that all products are delivered with the highest quality possible. Each team member will review all work products before delivery to the customer. Informal peer reviews of all work products will be conducted multiple times a week. The review type will be round robin with comments and suggestions from each team member provided. During the review, several types of work products will be reviewed for overall quality and correctness. These include use cases, activity diagrams, sequence diagrams, and any written documentation that is created. After the review is performed, the Quality Assurance lead is responsible for verifying that all of the accepted comments and input from each team member has been addressed, and that the overall product has all of the required components. After the Quality Assurance manager verifies that all of the comments from the review have been addressed, each team member will perform a final check of the product and report to the Quality Assurance manager that they accept it before the product is delivered.

### Preliminary Quality Assurance Audit

The QA Audit was performed using a defect-based checklist focusing in three specific areas. The first area of focus is the defect of omission. The following questions were used as a basis to review the requirements for defects of omission:

* Is this concept precisely defined somewhere?
* Is this acronym defined?
* Are these definitions summarized in the glossary of terms?
* Is this objective operationalized through specific requirements?
* Are those requirements sufficient to ensure this objective?
* Is the rationale for this requirement(or assumption) made explicit somewhere?
* If this requirement or assumption relates to another, is the latter specified somewhere?

The second area of focus is the defect of contradiction. The following questions were used as a basis to review the requirements for defects of contradiction:

* Is this statement consistent with the system objectives and constraints?
* Is this statement consistent with other related statement?

The third area of focus is the defect of ambiguity. The following questions were used as a basis to review the requirements for defects of ambiguity:

* Can this statement be interpreted differently in different relevant contexts or by readers from different background?
* What are the possible interpretations?
* Are the other statements using this term with a different meaning?

QA Audit Defect-Based Checklist for Requirements

|  |  |  |
| --- | --- | --- |
| **Defect** | **Checklist** | **Results** |
| Omission | Is this concept precisely defined somewhere? |  |
|  | Is this acronym defined? |  |
|  | Are these definitions summarized in the glossary of terms? |  |
|  | Is this objective operationalized through specific requirements? |  |
|  | Are those requirements sufficient to ensure this objective? |  |
|  | Is the rationale for this requirement(or assumption) made explicit somewhere? |  |
|  | If this requirement or assumption relates to another, is the latter specified somewhere? |  |
| Contradiction | Is this statement consistent with the system objectives and constraints? |  |
|  | Is this statement consistent with other related statement? |  |
| Ambiguity | Can this statement be interpreted differently in different relevant contexts or by readers from different background?  What are the possible interpretations? |  |
|  | Are the other statements using this term with a different meaning? |  |

### Preliminary CM Audit

The second deliverable of this document includes additional use cases with activity diagrams and sequence diagrams. New requirements were added for the new use cases. New function point diagrams and associated function point estimations were added. A QA audit of progress was performed and the audit report is included. The tasks needed for this deliverable and the final deliverable are included in the project schedule. Changes made to existing requirements are documented in table XXX.

CM Audit

|  |  |  |
| --- | --- | --- |
| **Requirement Number** | **Original Requirement Text** | **New Requirement Text** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Project Schedule

For our team project, a centralized team structure will be used. All team members will work together on each part of the project with specific tasks for each phase of the project. Assignment of tasking will be based on each team member’s roles based on varied skills and talents. For each deliverable, there will be a specific set of tasks that will be added to the schedule with resources allocated as appropriate for the tasks.

Table 1.5‑1 Theater Ticket System Project Schedule

| Delivery | Task | Person Responsible | Due Date |
| --- | --- | --- | --- |
| Delivery 1 Part 1 | Create document cover page, revision history page, table of contents, table of figures and table of tables | Samantha | Oct 7 |
|  | Revise Scope and project description | Andy | Oct 7 |
|  | Create Glossary | Sumeet | Oct 7 |
|  | Describe features of the system | Anthony | Oct 7 |
|  | Create CSA requirements table | Angela | Oct 7 |
|  | Create future software requirements table | Angela | Oct 7 |
|  | Identify software requirements | TBD | TBD |
|  | Create top level use case diagram | Andy | Oct 7 |
|  | Create Reserve Ticket Use Case description | Andy | Oct 7 |
|  | Create Reserve Ticket Activity Diagram | TBD |  |
|  | Create Reserve Ticket Function Point estimate artifacts | Andy | Oct 7 |
| Delivery 1 Part 2 | Describe how we will operate, roles, tools to be used, common strategy | Samantha | Oct 7 |
|  | Describe QA strategy | Rashad | Oct 7 |
|  | Create Project Schedule | Angela | Oct 7 |
|  | Write up meeting minutes | Samantha | Oct 7 |
|  | Integrate tasks for Delivery 1 | All contribute with one person creating the master | Oct 7 |
|  | Peer review Delivery 1 | All | Oct 8 |
|  | Submit Delivery 1 | Samantha | Oct 8 |
| Delivery 2 Part 1 | Goals and Objectives | TBD | Oct 29 |
|  | Refine scope | Sumeet | Oct 29 |
|  | Function Point Estimate | All | Oct 29 |
|  | Set of Use Cases | All | Oct 29 |
|  | Activity Diagrams | All | Oct 29 |
|  | Sequence Diagrams | All | Oct 29 |
|  | Update Requirements tables | Andy | Oct 29 |
|  | Update Traceability Matrix | Anthony | Oct 29 |
|  | Update Glossary | Rashad | Oct 29 |
| Delivery 2 Part 2 | Preliminary QA Audit Report | All | Oct 29 |
|  | CM Audit | Angela | Oct 29 |
|  | Add to meeting minutes | Samantha | Oct 29 |
|  | Add to Project Schedule | Angela | Oct 29 |
|  | Peer Review Delivery 2 | All | Oct 29 |
| Delivery 3 Part 1 | TBD | TBD | Nov 17 |
|  | Refine requirements | TBD | Nov 17 |
|  | Refine Activity Diagrams | TBD | Nov 17 |
|  | Refine Sequence Diagrams | TBD | Nov 17 |
|  | Update all requirements tables | TBD | Nov 17 |
| Delivery 3 Part 2 | QA Audit Report | TBD | Nov 17 |
|  | CM Audit | TBD | Nov 17 |
|  | Update meeting minutes | TBD | Nov 17 |
|  | Update Lessons Learned | TBD | Nov 17 |
|  | Peer Review Delivery 3 | TBD | Nov 17 |
| Presentation | TBD | TBD | Nov 24, Dec 01 |

## Meeting Minutes

Table 1.6‑1 Team Meeting Minutes

| Date | Description | Attendees |
| --- | --- | --- |
| 9/20/14  2pm | Team meeting dates fixed. Project topic was chosen. | Andy, Samantha, Sumeet, Rashad (via phone), Angela, Anthony |
| 9/23/14  7pm | Team members reviewed the proposal and signed off. | Andy, Samantha, Sumeet, Rashad, Angela, Anthony |
| 10/4/14  10am | Completed a white board discussion of Delivery 1 Part 1 and most of Part 2. | Andy, Samantha, Sumeet, Rashad, Angela, Anthony |
| 10/5/14  2pm | Completed a white board discussion of Delivery 1 Part 2. Assigned tasks to team members and laid out the document template and project schedule. Computed function points. | Andy, Samantha, Sumeet, Rashad, Angela, Anthony |
| 10/7/14  7pm | Checked progress on assigned tasks. | Andy, Samantha, Sumeet, Rashad, Angela, Anthony |
| 10/8/14  7pm | Completed an informal peer review for Delivery 1. Submitted Delivery 1. | Andy, Samantha, Sumeet, Rashad, Angela, Anthony |

This page intentionally left blank.

**Appendix A**

**Requirements Traceability Matrices**

CSA to TTS Requirements Traceability Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| CSA  Requirement  Number | Requirement Description | TTS  Requirement  Number | Use Case |
|  | The CSA shall have the capability to enter customer data. | TBD |  |
|  | The CSA shall have access to theater venues. | TBD |  |
|  | The CSA shall have access to theater events. | TBD |  |
|  | The CSA shall have access to seat availability. | TBD |  |
|  | The CSA shall be able to book reserved seating. | TBD |  |
|  | The CSA shall be able to book general admission seating. | TBD |  |
|  | The CSA shall be able to accept patron payments. | TBD |  |
|  | The CSA shall be able to exchange tickets. | TBD |  |
|  | The CSA shall be able to refund tickets. | TBD |  |
|  | The CSA shall be able to book season tickets. | TBD |  |
|  | The CSA shall be able to book VIP seating. | TBD |  |
|  | The CSA shall be able to book special accommodation seating. | TBD |  |

TTS to CSA Requirements Traceability Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| TTS  Requirement  Number | Requirement Description | CSA  Requirement  Number | Use Case |
| REQ100 | TBD | TBD |  |