

TEST RESULTS DOCUMENT SAMS

Student Auditorium Management Software

GROUP 59

PRAVEEN KUMAR

MITIKIRI (14CS30018)

SRIKANTH REDDY

PALEM (14CS30020)

Contents

I.	Black Box Testing.....	2
I.I	BBT for Login.....	2
I.2	BBT for Adding a Sales Person.....	3
I.3	BBT for Removing a Sales Person.....	3
I.4	BBT for Changing Password.....	4
I.5	BBT for Adding a show details.....	4
I.6	BBT for Adding a date and time for a show.....	5
I.7	BBT for Displaying Balance Sheets.....	6
I.8	BBT for Selecting a show.....	7
I.9	BBT for Booking a Ticket.....	7
I.I0	BBT for Cancelling a Ticket.....	8
I.II	BBT for Entering Expenditures.....	8
I.I2	BBT for Querying Seats.....	9
2.	White Box Testing.....	9
2.I.	WBT for Login.....	9
2.2.	WBT for Balance Sheets.....	11
2.3.	WBT for Ticket Booking.....	12
3.	User Interface Testing.....	15
3.I.	Start Window.....	15
3.2.	Login Window.....	16
3.3.	Show Manager Window.....	17
3.4.	Show Details Window.....	18
3.5.	Date and Time Window.....	19
3.6.	Show Select Window.....	20
3.7.	Balance Sheet Window.....	21
3.8.	Remove Sales Person Window.....	22
3.9.	Add Sales Person Window.....	22
3.I0.	Change Password Window.....	23
3.II.	Status of Sales Person Window.....	24
3.I2.	Accounts Clerk Window.....	24
3.I3.	Expenditures Window.....	25
4.	Entry and Exit criteria.....	26
4.I	Unit Testing.....	26
4.I.I.	Black Box Phase.....	26
4.I.2.	White Box Phase.....	26
4.2	Integration Testing.....	27
4.2.I.	Integration Test Entry Criteria.....	27
4.2.2.	Integration Test Exit Criteria.....	27
4.3.	System Testing.....	28

4.4.	Shipping or Live Release	28
4.4.I.	Shipping/ Live Release Criteria	28
5.	Deliverables	28
6.	Environmental Needs	28

I. Black Box Testing

All the black box tests are done assuming that the database is connected. Else an SQL Exception is thrown and stack trace is printed.

I.I BBT for Login

This method prompts for the input of UserID and Password of the employee. It verifies the same with the database.

Type	Case	Expected Output	Actual Output	Result
Invalid	Any field is left empty	Unable to log in	A pop-up appears with the text "Please Re-enter credentials"	Pass
Invalid	An invalid UserID is entered	Unable to log in	A pop-up appears with the text "Please Re-enter credentials"	Pass
Invalid	UserID valid, but password incorrect	Unable to log in	A pop-up appears with the text "Please Re-enter credentials"	Pass
Valid	Appropriate Log in details of Show Manager are given	The Show Manager should be able to login.	The Show Manager GUI is displayed	Pass
Valid	Appropriate Log in details of Accounts Clerk are given	The Accounts Clerk should be able to login.	The Accounts Clerk GUI is displayed	Pass
Valid	Appropriate Log in details of Sales Person are given	The Sales Person should be able to login.	The Sales Person GUI is displayed	Pass

1.2 BBT for Adding a Sales Person

This method prompts to enter the Name and Contact number of the sales person to be added. It then creates a random UserID and a password for the sales person which can be altered later. It updates the database accordingly.

Type	Case	Expected Output	Actual Output	Result
Invalid	Contact Number is alphanumeric or empty	Invalid Contact Number	A pop-up appears with the text "Please enter a valid Contact Number"	Pass
Invalid	Contact Number does not contain 10 digits	Invalid Contact Number	A pop-up appears with the text "Please enter a valid Contact Number"	Pass
Invalid	Name Field is empty	Enter the name	A pop-up appears displaying "Enter a valid name".	Pass
Valid	Name and appropriate Contact Number are entered	Sales Person is Created	A pop-up appears with the Name, Contact, Username and Password of the newly added sales person.	Pass

1.3 BBT for Removing a Sales Person

It displays the list of currently employed sales person and prompts to select one person from the list. Accordingly, database is updated.

Type	Case	Expected Output	Actual Output	Result
Valid	A sales Person is selected from the list	Sales Person is removed	A pop-up appears with the text "Successfully fired !"	Pass
Invalid	When no sales person is employed	No one is removed	A pop up appears with text "No sales person removed"	Pass

I.4 BBT for Changing Password

This prompts the user to enter a new string to update/change the password. Accordingly, database is updated.

Type	Case	Expected Output	Actual Output	Result
Invalid	An empty is password is said to be updated	Do not change the password	A pop-up appears with the text "Enter a non-empty password."	Pass
Valid	Any non-empty string is entered	Password is updated	A pop-up appears with the text "Password Updated Successfully."	Pass

I.5 BBT for Adding a show details

It prompts the user to enter the Name of the show, the prices for balcony and ordinary seats and also the balcony and ordinary seats available for booking. The it is directed to enter further details like show dates and timings.

Type	Case	Expected Output	Actual Output	Result
Invalid	Any of the fields is empty	Error Message	A pop-up appears with the text "Kindly enter the previous details before proceeding to adding shows on a particular date."	Pass
Invalid	Strings are entered in any of the integer filed like ticket prices and seats available.	Error Message to enter valid details	A popup appears with the text "One or more fields are invalid! Please re-enter."	Pass
Invalid	Seats prices are floating point numbers.	Do not update the database	A popup appears with the text "One or more fields are invalid! Please re-enter."	Pass
Invalid	Balcony seat price is less than ordinary seat price	Error Message should be shown to correct the seat prices.	A pop-up appears with the text showing that "Balcony seats must be higher than Ordinary Seats"	Pass
Valid	Balcony and Ordinary Seat	Consider the values and the seats as free.	Accepts and proceeds to the	Pass

	prices are set to zero.		addition of date and time.	
Invalid	Every filed is appropriate but dates and times are not chosen.	No changes in database.	Database is not updated as it doesn't mean anything to add a show without show timings.	Pass
Invalid	Every field is empty and is saved.	No changes in database.	Database is not updated as it doesn't mean anything to add a show without show timings.	Pass
Valid	All fields are entered according to the formats.	Show is listed on the database.	Show details are updated into the database with the details if and only if the dates and times entered are valid.	Pass

1.6 BBT for Adding a date and time for a show

User is prompted is enter the date for a show and select timings (max 5.) for that show and on that day. Database is updated accordingly.

Type	Case	Expected Output	Actual Output	Result
Invalid	Date is selected but Time is not selected	No change in database.	The show details are not updated in the database as it doesn't mean anything to add a show without any show timings.	Pass
Invalid	Date selected is already over.	Invalid Date	A pop-up with the text "Date time is already over. Enter a valid date." appears	Pass
Invalid	Two timings clash for the same show on the same date.	Invalid timings.	A pop-up appears with the text "Clash of Times"	Pass
Invalid	Start time is later than end time	Invalid timings.	A pop-up appears "Please select valid show timings" in that corresponding row.	Pass

Invalid	Show stretches over two days i.e. starts on a particular day and extends through the mid-night	Accept timings	A pop-up appears "Please select valid show timings" in that corresponding row.	Fail
Invalid	Timings do not clash with each other but clash with show timings on another show on that particular	Invalid Timings	A pop-up appears "The show timings entered clash with other show times screening on the same day. Kindly try again!"	Pass
Valid	Date is proper and timings do not clash with any other show.	Accept the show details.	The date & time frame disposes and the database is updated with the show dates and timings.	Pass.

1.7 BBT for Displaying Balance Sheets

User is prompted is select either a show or a year to view the balance sheets.

Type	Case	Expected Output	Actual Output	Result
Invalid	Check balance sheets for a year when there are no shows.	No balance sheets displayed.	A NullPointerException is thrown and stack trace is printed	Pass
Invalid	Check balance for a show when there are no shows	No balance sheets are displayed.	The show selection frame doesn't show ant shows and thus no balance sheet is displayed.	Pass
Valid	A valid year is chosen	Balance sheet should be shown.	All the expenditures and sales incomes of that year are displayed.	Pass
Valid	A valid show is selected	Balance sheet is displayed.	All the expenditures including logistics, artist payments and sales income are displayed.	Pass

I.8 BBT for Selecting a show

User is prompted to first select a show name and then show date and finally show time. Show time can't be selected without selecting show date.

Type	Case	Expected Output	Actual Output	Result
Valid	A show is selected from the list of displayed shows.	Show is selected	That show is selected and proceeded for further usage.	Pass

I.9 BBT for Booking a Ticket

User is prompted to first select a show name and then show date and finally show time. Show time can't be selected without selecting show date. Then the required number of balcony and ordinary seats are to be entered. Ticket will be booked and database is updated.

Type	Case	Expected Output	Actual Output	Result
Invalid	After selecting the show, if either balcony seats or ordinary seats is kept empty	Consider the empty field to be zero.	The corresponding seats required will be set to zero and thus ticket is generated.	Pass
Invalid	Both balcony and ordinary seats requested are zero.	Do not update database.	A pop-up appears with the text "Please enter a valid number of seats to book."	Pass
Invalid	Either of balcony or ordinary seats requested seats are more than seats available.	Do not book.	A pop-up with the text "Requested number of corresponding type are not available. Kindly select fewer seats."	Pass
Invalid	Both balcony and ordinary requested seats are more than available seats.	Do not book.	A pop-up appears with text "Requested no of Balcony seats and Ordinary seats aren't available kindly select fewer seats"	Pass
Invalid	Strings are entered in the balcony and ordinary seats required	Show error message to enter valid seats.	A pop-up is displayed with the text "Enter valid number of seats required"	Pass
Valid	Proper number of seats are selected for balcony and ordinary.	Book the tickets.	Ticket is generated and is shown on the screen. Database	Pass

			is updated. The transaction details are updated for sales person also.	
--	--	--	---	--

I.IO BBT for Cancelling a Ticket

User is prompted for Ticket-ID and then on verification tickets are cancelled. Database is updated with changes I seats available, commission for the sales person and also cancellation charges.

Type	Case	Expected Output	Actual Output	Result
Invalid	Invalid ticket transaction ID	Ticket is not cancelled.	A pop-up appears with text "Invalid Ticket ID. Please try again."	Pass
Valid	Valid Transaction ID is entered.	Ticket is cancelled.	Ticket is cancelled and pop-up shows the amount to be refunded.	Pass
Valid	Ticket is cancelled by a sales person different from the sales person who booked.	Properly database is updated.	The commission is cut in the sales person who booked it and no changes for the sales person who cancelled it.	Pass

I.II BBT for Entering Expenditures

Accounts clerk is prompted to select a show and enter the expenditures for that show like payments to artists, logistics, etc. Database is updated accordingly.

Type	Case	Expected Output	Actual Output	Result
Invalid	All fields are empty or any field is empty.	No expenditure is added.	A pop-up with the text "One or more fields are invalid" appears.	Pass
Invalid	String is entered in any of the fields.	Error Message.	A pop-up with the text "One or more fields are invalid" appears.	Pass
Valid	All proper expenditures are entered.	Expenditures are noted.	Expenditures are added for that particular show and database is updated.	Pass

1.I.2 BBT for Querying Seats

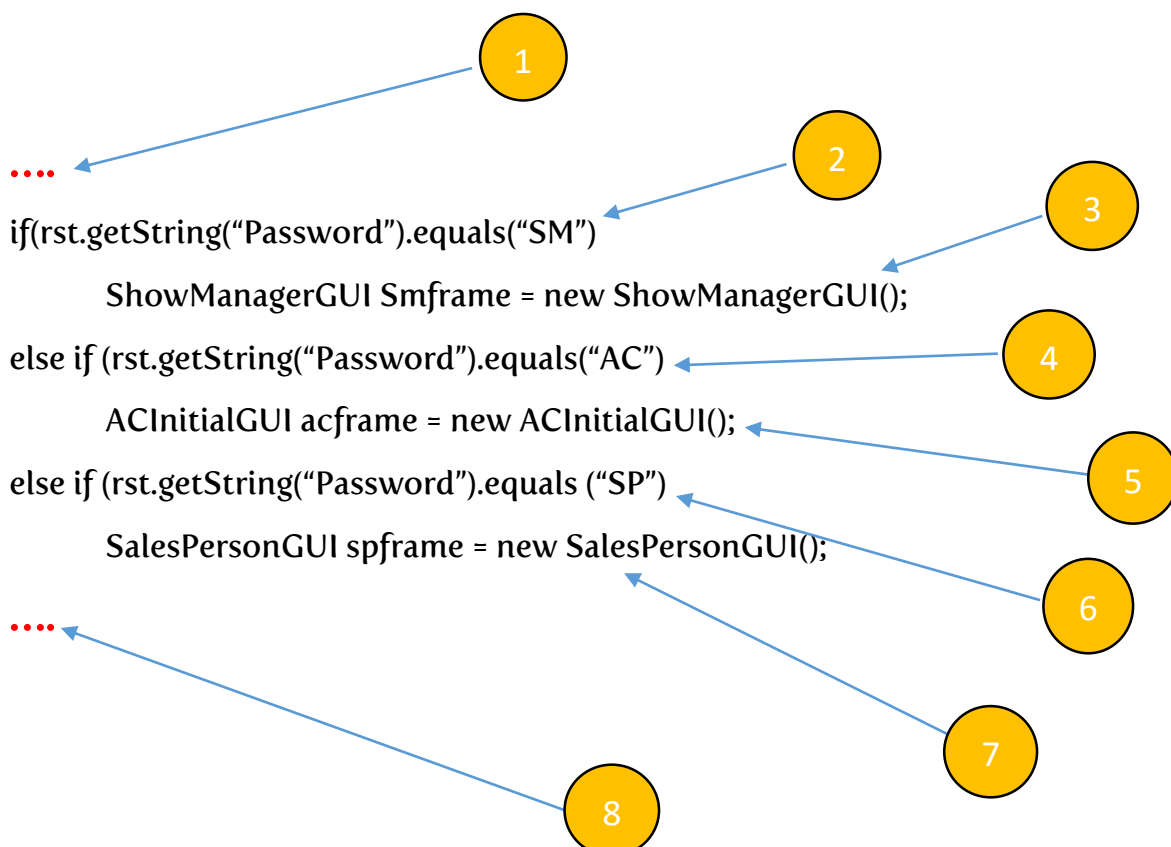
The user is prompted to select a show to see seat availability.

Type	Case	Expected Output	Actual Output	Result
Valid	Show is selected properly.	See the seats available.	A pop-up containing the show details selected and also the balcony, ordinary seats available still is displayed.	Pass

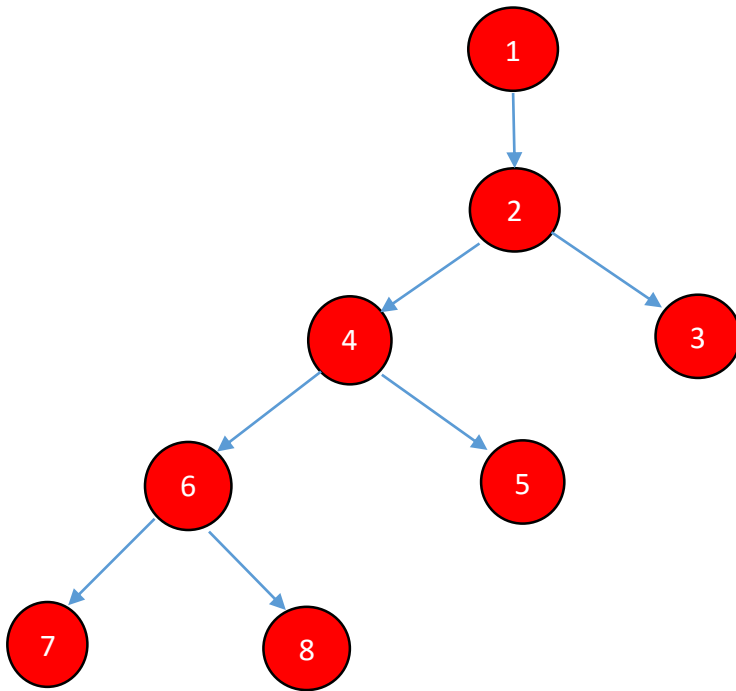
2. White Box Testing

2.I. WBT for Login

Code Snippet:



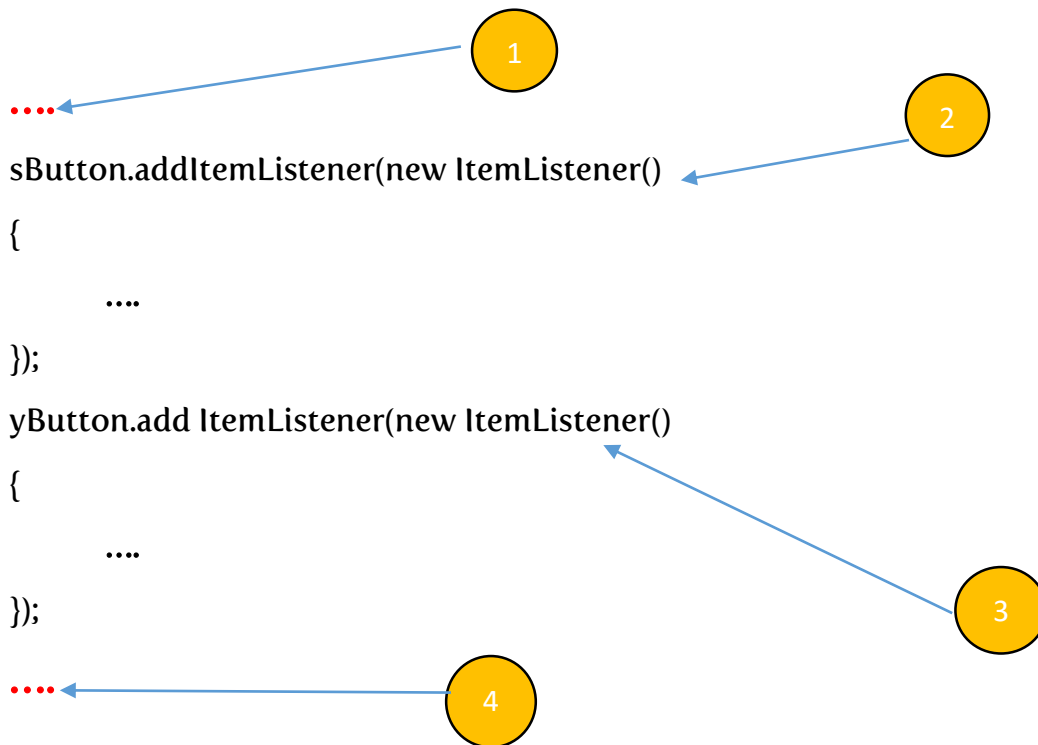
Path Diagram



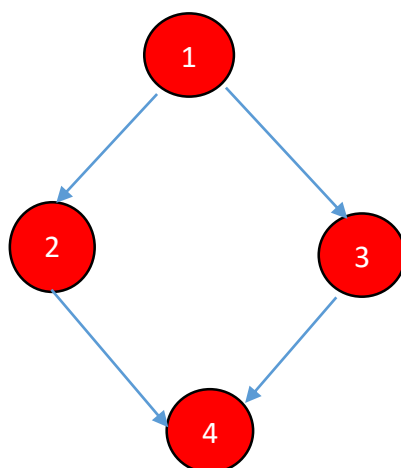
Path - I	I-2-3
Path - 2	I-2-4-5
Path - 3	I-2-4-6-8
Path - 4	I-2-4-6-7
Path - I Expected Result	Show Manager logins
Path - 2 Expected Result	Accounts Clerk logins
Path - 3 Expected Result	Sales Person logins
Path - 4 Expected Result	Invalid Credentials

2.2. WBT for Balance Sheets

Code Snippet



Path Diagram



Path - I	I-2-4
Path - 2	I-3-4
Path - I Expected Result	Show Selection is visible
Path - 2 Expected Result	Year Selection is visible

2.3. WBT for Ticket Booking

Code Snippet



```
....
if(boo <= (rst.getInt("BSale") - rst.getInt("BFilled"))
    && ooo <= (rst.getInt("OSale") - rst.getInt("OFilled")))
{
    int bef = 0;
    Statement st9 = db.getConnection().createStatement();
    ResultSet rt9 = st9.executeQuery("SELECT * FROM loginid WHERE UserID =
    '"+SPusername+"'");
    while(rt9.next())
    {
        bef = rt9.getInt("Amount");
    }
    int amount = rst.getInt("BPrice") * boo
rst.getInt("OPrice") * ooo;
    int newb = rst.getInt("BFilled") + boo;
    int newo = rst.getInt("OFilled") + ooo;
    Statement smtI = db.getConnection().createStatement();
    ResultSet rtI = smtI.executeQuery("select * from ticket where AutoInc >= 0");
    int id = 0;
    while(rtI.next())
    {
        id = rtI.getInt("AutoInc");
    }
    ++id;
    String transID = "Ticket-" + Integer.toString(id);
    String q = "INSERT INTO ticket (TransID, SPID, ShowName, ShowDate, ShowTime, BSeats,
    OSeats, Amount)" + "VALUES ('"+transID+"', '"+SPusername+"',
    '"+showBox.getSelectedItem().toString()+"', '"+dateBox.getSelectedItem().toString()+"',
    '"+timeBox.getSelectedItem().toString()+"', '"+boo+"', '"+ooo+"',
    + '"+amount+'");
    +
```

```
String u = "UPDATE showdetails SET BFilled = '"+newb+"' , OFilled = '"+newo+"' WHERE "+
"Name = '"+showBox.getSelectedItemAt().toString()+"' "+ "AND ShowDate =
 '"+dateBox.getSelectedItemAt().toString()+"' "+ "AND STime =
 '"+timeBox.getSelectedItemAt().toString()+"'";
```

```
amount += bef;
```

```
double com = amount * 0.05;
```

```
String uI = "UPDATE loginid SET Amount = '"+amount+"', Commission = '"+com+"' WHERE "+
"UserID = '"+SPusername+"'";
```

```
Statement st3 = db.getConnection().createStatement();
```

```
Statement st = db.getConnection().createStatement();
```

```
Statement st2 = db.getConnection().createStatement();
```

```
st.executeUpdate(q);
```

```
st3.executeUpdate(uI);
```

```
st2.executeUpdate(u);
```

```
}
```

```
else
```

```
{
```

```
if(Integer.parseInt(bBook.getText()) > (rst.getInt("BSale") - rst.getInt("BFilled"))&&
Integer.parseInt(oBook.getText()) > (rst.getInt("OSale") - rst.getInt("OFilled")))
```

```
{
```

```
...
```

```
}
```

```
else if(Integer.parseInt(bBook.getText()) > (rst.getInt("BSale") - rst.getInt("BFilled")))
```

```
{
```

```
....
```

```
}
```

```
else
```

```
{
```

```
....
```

```
}
```

```
}
```

```
....
```

```
7
```

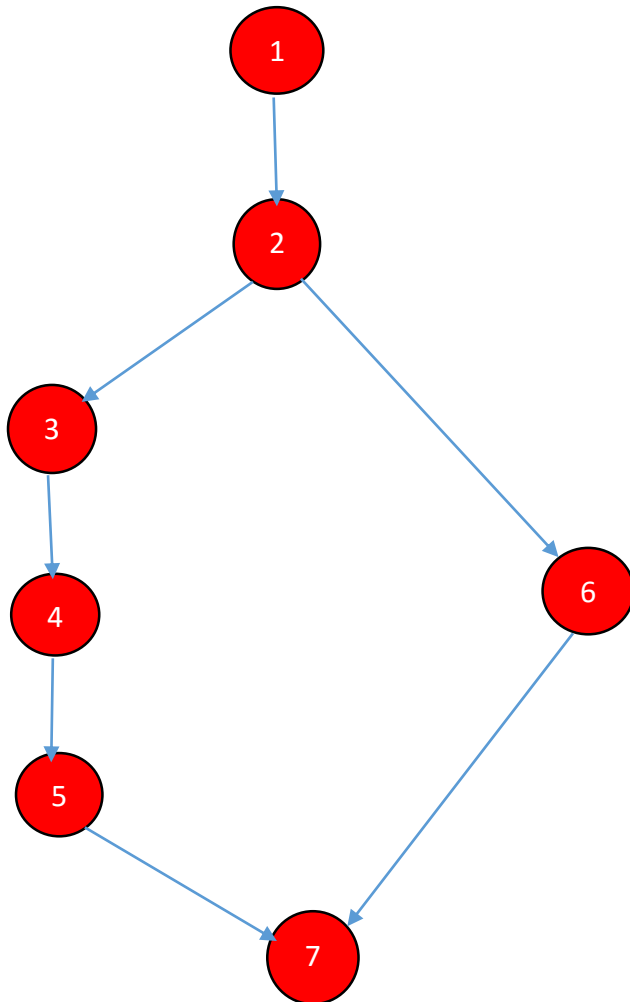
3

4

5

6

Path Diagram

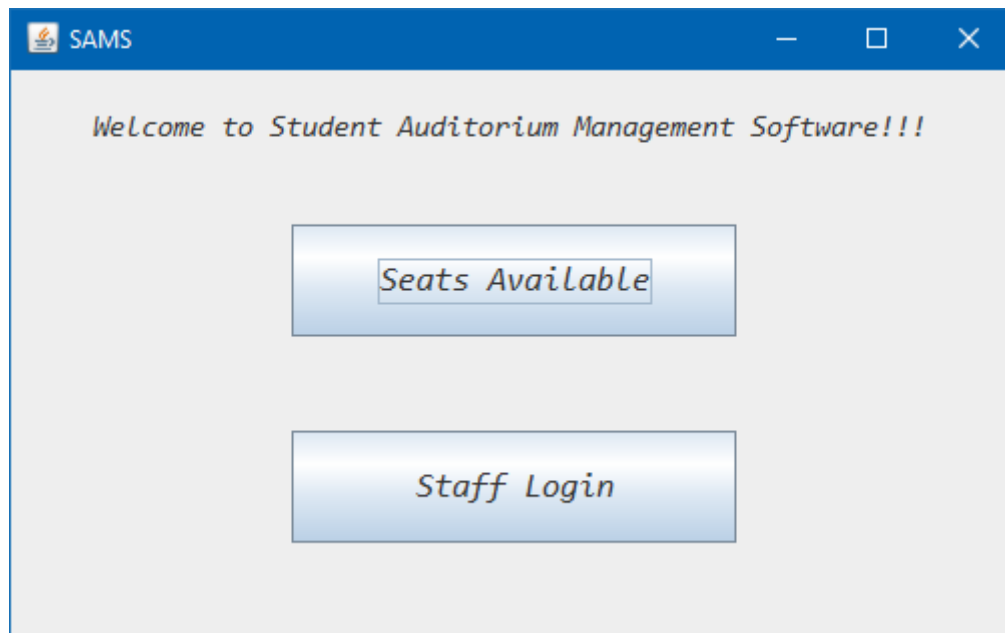


Path - I	I-2-3-4-5-7
Path - 2	I-6-7
Path - I Expected Result	Required seats are available, so sales person books the ticket, and showdetails, loginid, ticket tables in Data Base are updated.
Path - 2 Expected Result	Requires seats are available aren't available, prompts the user to enter a fewer seats.

3. User Interface Testing

To test the user interface, each GUI was tested manually. We describe here the techniques.

3.I. Start Window



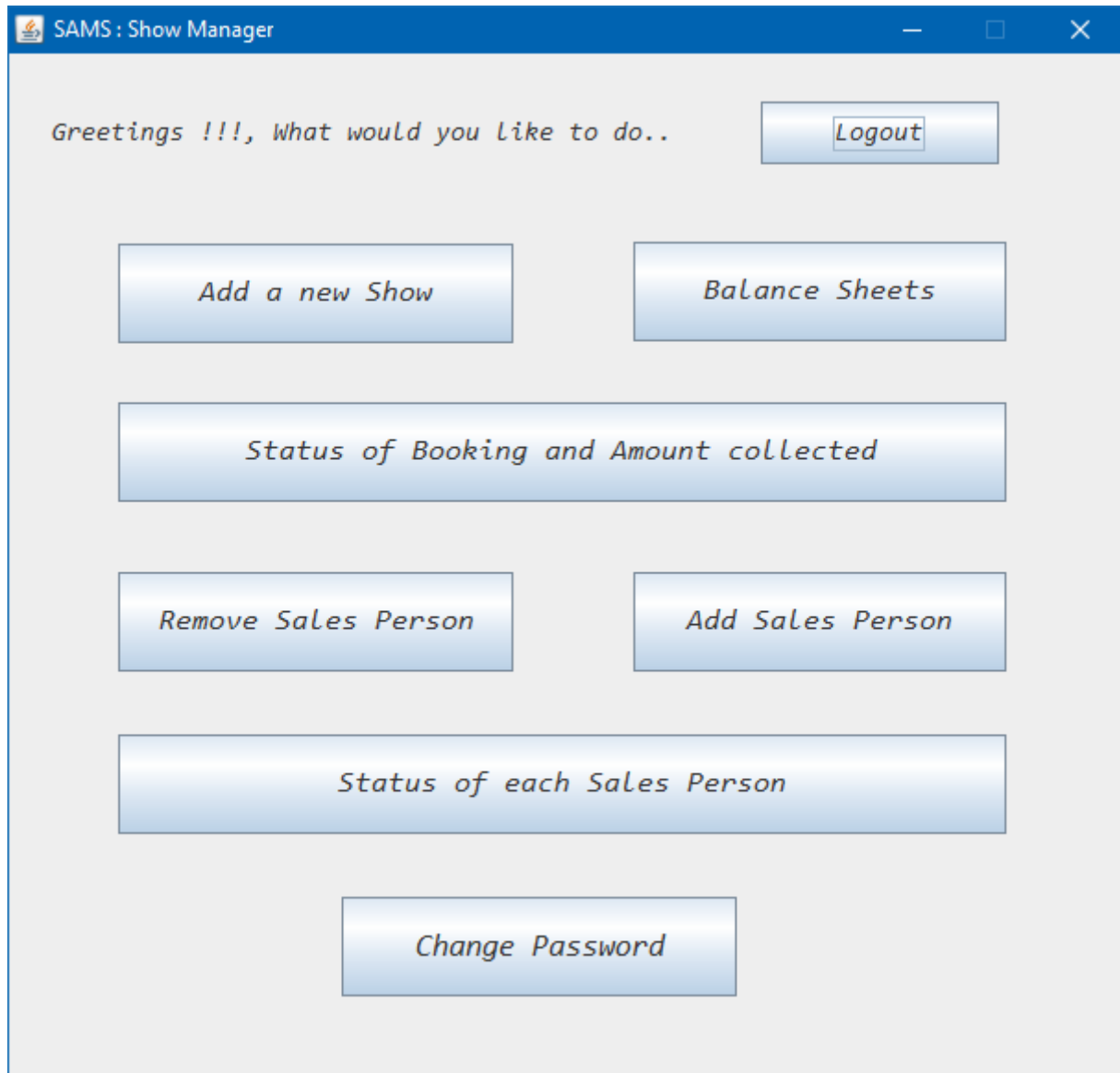
What is tested?	All the components which interact with the user.
Inputs	<ul style="list-style-type: none">• User press Staff Login Button• User presses Seats Available Button
Expected Result(s)	Directed to the corresponding windows
Effective Result(s)	<ul style="list-style-type: none">• If Staff Login is pressed, Login Window is displayed• If Seats Available Button is pressed, Show Select Window Is opened.

3.2. Login Window

The image shows a screenshot of a login window titled "SAMS : Login". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is light gray. It contains two input fields: the first is labeled "UserID" and the second is labeled "Password". Below these fields is a blue button with the text "Verify".

What is tested?	All the components which interact with the user.
Inputs	<ul style="list-style-type: none">• User enters UserID and password• User presses Verify Button
Expected Result(s)	If entered details are correct, then the corresponding employee's windows should appear, else an error message will be displayed
Effective Result(s)	All results match the expected results

3.3. Show Manager Window



What is tested?	All the components which interact with the user.
Inputs	<ul style="list-style-type: none">• Add New Show button is pressed.• Balance sheets button is pressed.• Status of booking and Amount collected button is pressed.• Add sales person button is pressed.• Remove Sales Person button is pressed.• Status of each Sales Person button is pressed.• Change password button is pressed.• Log out button is pressed.
Expected Result(s)	<ul style="list-style-type: none">• Show Details window will be shown to enter show details.• A GUI to select the show or a year will be shown and then the corresponding balance sheets will be displayed.• A Show Select is opened and then the show status for that show is displayed.

	<ul style="list-style-type: none"> • New GUI for the details of Sales Person is displayed and he is added. • The Sales Person is selected and fired. • Shows all the sales persons with the amount payable to them. • Password Change GUI is opened. • Return to the Login Window.
Effective Result(s)	All results match the expected results

3.4. Show Details Window

The screenshot shows a window titled "SAMS: Show Details". It contains the following elements:

- Name:** A single text input field.
- Prices....**
 - Balcony:** A text input field.
 - Ordinary:** A text input field.
- Seats for Sale...**
 - Balcony:** A text input field.
 - Ordinary:** A text input field.
- Buttons:** "Add shows on a Date" and "Save".

What is tested?	All the components which interact with the user.
Inputs	<ul style="list-style-type: none"> • Show Name is entered • Prices of balcony and ordinary seats are entered • Balcony and ordinary seats available for booking are entered. • Add shows on a date button is pressed. • Save button is pressed.
Expected Results	<ul style="list-style-type: none"> • After all the fields are entered Date and Time Selection Windows will be opened. • Database will be updated.
Effective Results	All results match the expected results

3.5. Date and Time Window

Select a Date

19 Apr, 2016

Start Time End Time

- 07:10:17 08:10:17

- 09:10:17 10:10:17

- 11:10:17 12:10:17

+

+

Save

What is tested?	All the components which interact with the user.
Inputs	<ul style="list-style-type: none">• Date is selected• End time and Start time are selected• Save button is pressed.
Expected Results	<ul style="list-style-type: none">• Any time clashes present will be reported.• The show timings and date will be updated for that particular show in the database
Effective Results	All results match the expected results

3.6. Show Select Window

Kindly , select Show, then Date, then Show time ..

Buttons: Show, Show Date, Show Time

Selections: Suicide Squad, 22-Aug-2016, 07:09:57

CONTINUE

What is tested?	All the components which interact with the user.
Inputs	<ul style="list-style-type: none">• Show button is pressed.• Show Name is selected.• Show Date button is pressed.• Show Date is selected.• Show Time button is pressed.• Show Time is selected.• Continue button is pressed.
Expected Results	<ul style="list-style-type: none">• All the details of the show are stored and passed on.
Effective Results	All results match the expected results

3.7. Balance Sheet Window

SAMS : Balance Sheets

☐ Show ☒ Year

Select Year

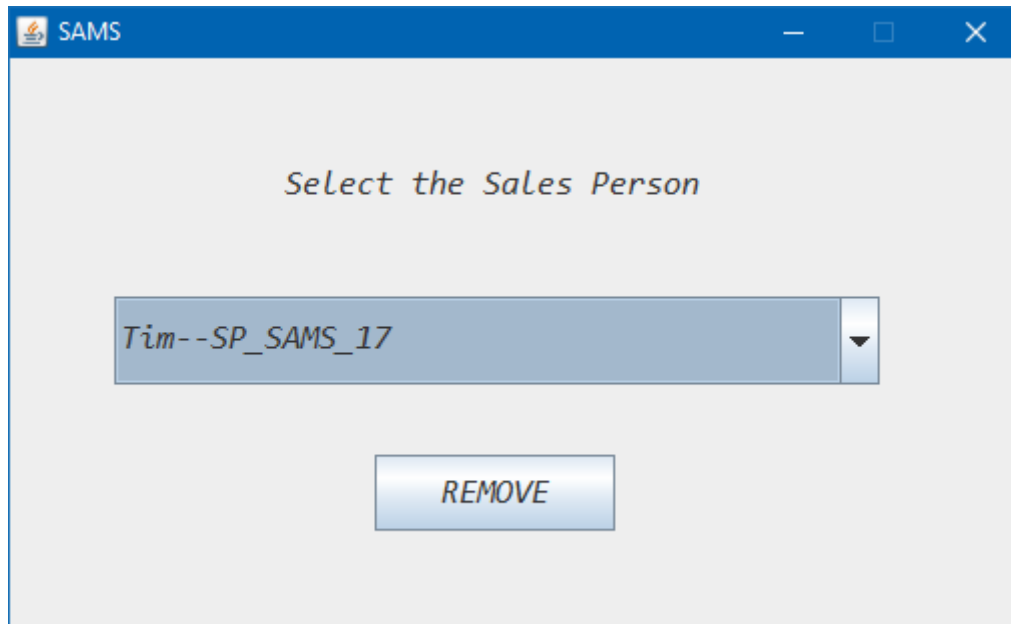
2016

2016

Get Balance Sheets

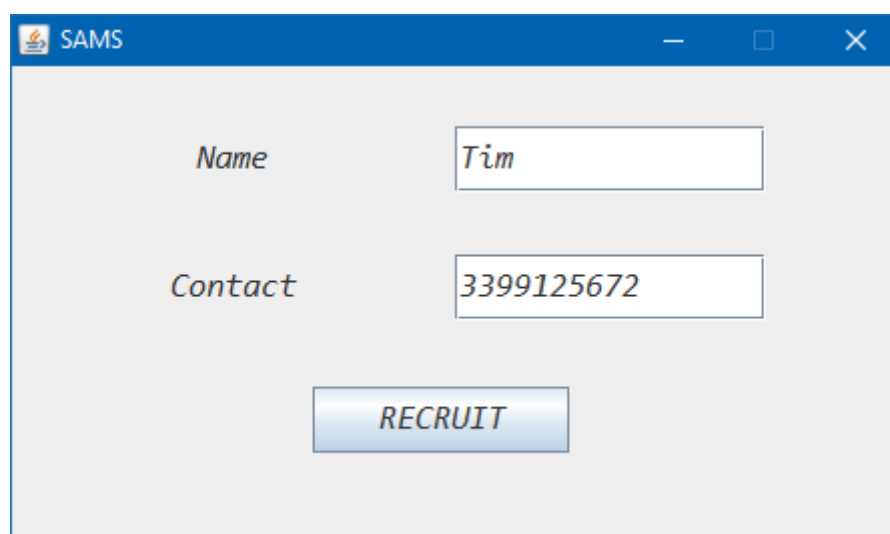
What is tested?	All the components which interact with the user.
Inputs	<ul style="list-style-type: none">• Show radio button is selected.• Year Radio Button is selected.• Year is selected.• Get Balance Sheets button is pressed.
Expected Results	The concerned balance sheet, either for a year or for a particular show will be displayed.
Effective Results	All results match the expected results

3.8. Remove Sales Person Window



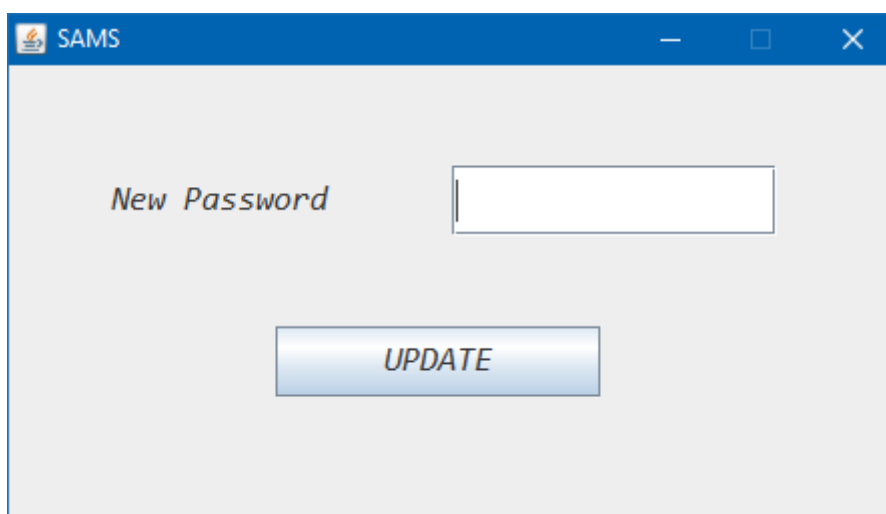
What is tested?	All the components which interact with the user.
Inputs	<ul style="list-style-type: none">• The Sales Person is selected.• Remove Button is pressed.
Expected Results	<ul style="list-style-type: none">• The sales person will be fired and removed from the database.
Effective Results	All results match the expected results

3.9. Add Sales Person Window



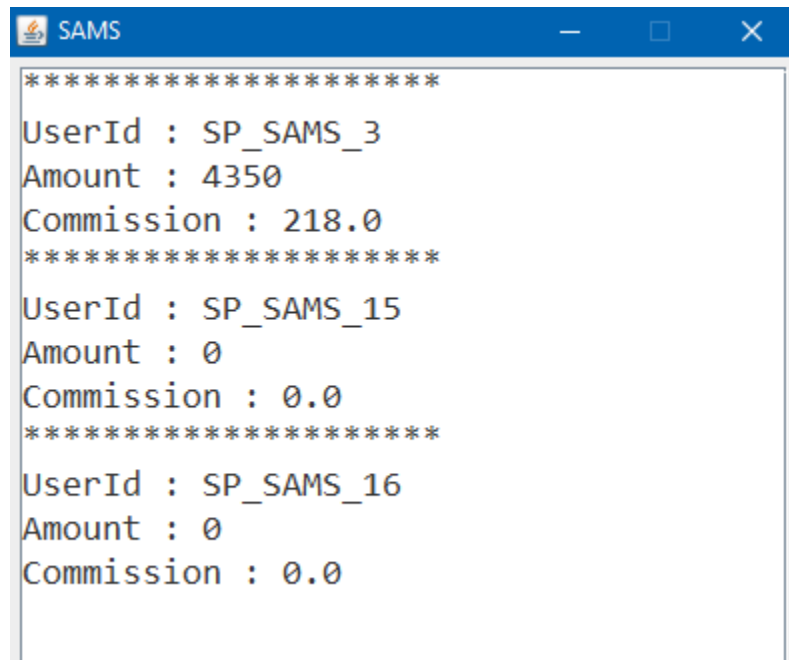
What is tested?	All the components which interact with the user.
Inputs	<ul style="list-style-type: none"> • Name is entered • Valid Contact Number is entered • Recruit button is pressed.
Expected Results	<ul style="list-style-type: none"> • A message will be shown confirming the addition of new sales person.
Effective Results	All results match the expected results

3.IO. Change Password Window



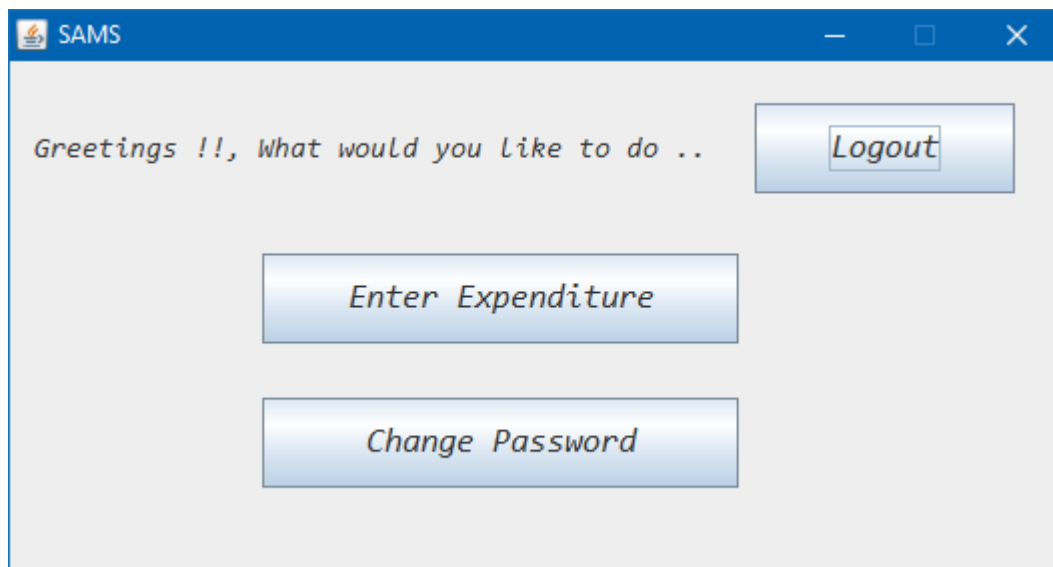
What is tested?	All the components which interact with the user.
Inputs	<ul style="list-style-type: none"> • New password is entered. • Update button is pressed.
Expected Results	Password will get updated in database and the confirmation is shown through a message box.
Effective Results	All results match the expected results

3.II. Status of Sales Person Window



What is tested?	All the components which interact with the user.
Inputs	None
Expected Results	Shows the amount collected by a sales person and also the commission payable to him.
Effective Results	All results match the expected results

3.I2. Accounts Clerk Window



What is tested?	All the components which interact with the user.
Inputs	<ul style="list-style-type: none">• Enter Expenditure button is pressed• Change Password button is pressed• Logout button is pressed
Expected Results	<ul style="list-style-type: none">• Expenditures Windows will be opened after selecting show

	<ul style="list-style-type: none"> • Change Password Windows will be opened • Returns to Login window.
Effective Results	All results match the expected results

3.I3. Expenditures Window

SAMS : Accounts Clerk

Greetings !!, What would you like to do ..

Various Expenditures ..

Logistics

Payment to Artists

Others

What is tested?	All the components which interact with the user.
Inputs	<ul style="list-style-type: none"> • Logistics expenditure is entered. • Payments to artists expenditure is entered. • Other expenditure is entered • Save button is pressed • Logout button is pressed
Expected Results	<ul style="list-style-type: none"> • The values are updated for that show in database. • Returns to main Login window.
Effective Results	All results match the expected results

4. Entry and Exit criteria

This section describes the general criteria by which testing commences, temporarily stopped, resumed and completed within each testing phase. Different features/components may have slight variation of their criteria, in which case, those should be mentioned in the feature test plan. The testing phase also maps to the impact level definition when a defect is entered in the bug-tracking phase.

4.I Unit Testing

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

4.I.I. Black Box Phase

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We will use Error Guessing and Boundary Value Analysis complexity metrics in order to quantifiably determine how many test cases needed to achieve maximum code coverage.

4.I.I.I. Black Box Entry Criteria

The Black Box Entry Criteria will rely on the component specification, and user interface requirements. Things that must be done on entry to the Black Box stage:

- ✚ All functions like ticket booking, ticket cancellation, sales person management, viewing balance sheets, show management, etc. must either be coded or stubs written.
- ✚ The type of Black Box Testing Methods will be determined upon entry. We will use Error Guessing, and Boundary Value Analysis.
- ✚ Error Guessing included entering garbage strings in search field, strings in numeric fields, closing the server and starting application, entering invalid inputs for different fields.
- ✚ Boundary Value Analysis included adding show timings which collide by a second, adding show on a day that has already completed.

4.I.I.2. Black Box Exit Criteria

The Black Box Exit Criteria listed below explains what needs to be completed in-order to exit Black Box phase. To exit the Black Box phase 100% success rate must be achieved. Things that must be done upon exiting the Black Box stage:

- ✚ The application showed no results in case of garbage string.
- ✚ If a show timing entered already existed in the database, an error message shown.
- ✚ For a very high stress on the database, the response time of the server was increased.
- ✚ All code bugs that had been discovered were corrected whenever possible.

4.I.2. White Box Phase

The White Box criteria apply for purposes of focusing on internal program structure, and discover all internal program errors. Defects were categorized and the quality of the software was assessed.

4.1.2.1. White Box Entry Criteria

The White Box Entry Criteria relied on verifying that the major features work separately but not necessarily in combination. The design and human interface were stable. Things that were done on entry to the White Box stage:

- ✚ Unit tests were written for as many functions as possible.
- ✚ The type of White Box Testing Methods that were used were determined upon entry. We used unit testing and test for memory leaks.
- ✚ Black Box Testing was in its late stages.

4.1.2.2. White Box Exit Criteria

The Student Auditorium Management Software in the White Box stage generally had a stable feel to it. White Box testing continued until the Black Box or next milestone criteria were met. To exit the White Box phase 100% success rate was achieved. The following describes the state of the product upon exit from the White Box Stage:

- ✚ All functions like ticket booking, ticket cancellation, sales person management, viewing balance sheets, show management, etc. were implemented, operational and tested.
- ✚ All test cases were generated. The test cases were generated from the Control Flow diagrams of all functions.
- ✚ The graphical interface was reviewed and found to satisfactory and stable, that is, no further changes to dialog boxes or other interface elements were planned. Minor changes were acceptable, but must be arranged with the Development and Test Engineers.
- ✚ All code bugs that were exposed were corrected.

4.2 Integration Testing

There are two modules that will be integrated for Integration Testing. The two modules are The Graphic User Interface module and the Controller (back-end). The two components consist of a mixture of stubs, driver, and fully functional code. The following describes the entry and exit criteria for Integration testing.

4.2.1. Integration Test Entry Criteria

The Integration Test Entry Criteria relies on both modules to be operational. The controller and human interfaces were stable. Things that were done on entry to the Integration Test stage:

- ✚ All functions like ticket booking, ticket cancellation, sales person management, viewing balance sheets, show management, etc. must either be coded or stubs written.
- ✚ The Graphical User Interface was either be coded and/or a driver and stubs were created. The driver was implemented to facilitate test case input and output values.
- ✚ Interfaces and interactions between the Controller and the Graphical User Interface was operational.

4.2.2. Integration Test Exit Criteria

The Integration Test Exit Criteria relied on both modules to be operational. The controller and human interface was stable. To exit the Integration Testing phase 100% success rate was achieved. Things that were done on exit from the Integration Test stage:

- ✚ All code bugs that had been discovered were corrected.
- ✚ The parser and Graphical User Interface Module interacted together with complete accuracy, according to the System Specification Design. All discrepancies were corrected.

- + Both Modules were ready for System Testing. Stubs and drivers were replaced with fully functional code.
- + Black Box Testing was completed.

4.3. System Testing

The System Test criteria apply for purposes of categorizing defects and the assessing the quality level of the product. All elements of the Controller and Graphical User Interface were meshed together and tested as a whole. System test focuses on functions and performance, reliability, instillation, behaviour during special conditions, and stress testing.

4.4. Shipping or Live Release

The Controller and server testing was scaled down and combined all phases of testing into two phases – Function Complete and Regression testing – and follows the release criteria.

4.4.I. Shipping/ Live Release Criteria

The criteria for entering the final stages are as follows:

- + All open product defects, regardless of fixed defects, documented, deferred, or otherwise addressed were identified.
- + Regression testing on all product defects and the entire product was completed and verified.

5. Deliverables

- + Program function specifications
- + Program source code
- + Test plan document – this document addresses testing objectives, criteria, standards, schedule and assignments, and testing tools.
 - o Unit Testing Plan
 - o Integration Plan
 - o System Testing Plan

6. Environmental Needs

As the project has been developed in Java, the software works in both Windows and Linux platforms.

MySQL is also required to function to be able to connect to the database.