
Testing Document

SWEN90007

Musical Event System
Team: SDA 1

In charge of:

Name	Student id	Email	Unimelb username	Github username
Chengyi Huang	1173994	chengyih@student.unimelb.edu.a	chengyih	KindOfblue
Ziqiang Li	1173898	ziqiangl1@student.unimelb.edu.au	ziqiangl1	johnnylild
Yuxin Liu	1408760	yuxliu20@student.unimelb.edu.au	YUXLIU20	YuxinLiu-unimelb
Hanming Mao	1257522	hanmingm@student.unimelb.edu.au	HanmingM	KkkellyM



SCHOOL OF
**COMPUTING &
INFORMATION
SYSTEMS**

Revision History

[illegible]

Contents

1. Introduction	5
1.1 Proposal.....	5
1.2 Target Users	5
1.3 Interface for Users (After login)	5
2. Covered Requirements	6
2.1 Functional or Product Requirements.....	7
3. Functional Test Cases	7
3.1 UC000: All types of user (admin, customer and event planner) can login to the system.....	8
3.2 UC001: Administrator views all user	8
3.2.1 TC001: Administrator views all user created successfully	8
3.3 UC002: Administrator authorize user as event planner	9
3.3.1 TC001: Administrator assigns events to the event planner	9
3.4 UC003: Administrator creates venue	9
3.4.1 TC001: Administrator creates venue	9
3.5 UC004: Administrator creates user account	11
3.5.1 TC001: Administrator creates user account	11
3.6 UC005: Administrator view events	11
3.6.1 TC001: Administrator view events.....	11
3.7 UC006: Administrator view bookings	12
3.7.1 TC001: Administrator view bookings	12
3.8 UC007: Event planner view events.....	13
3.8.1 TC001: Event planner view events.....	13
3.9 UC008: Event planner view booking.....	13
3.9.1 TC001: Event planner view booking.....	13
3.10 UC009: Event planner cancel booking.....	14
3.10.1 TC001: Event planner cancel booking	14
3.11 UC0010: Event planner creates musical events	15
3.11.1 TC001: Event planner creates musical events	15
3.12 UC0011: Event planner modify event (change time, date, ticket price)	16
3.12.1 TC001: Event planner modify event	16
3.13 UC0012: Event planner cancel event	17
3.13.1 TC001: Event planner cancel event.....	17
3.14 UC0013: Customer view events.....	18
3.14.1 TC001: Customer view events	18
3.15 UC0014: Customer view bookings	19
3.15.1 TC001: Customer view bookings	19
3.16 UC0015: Customer book event.....	19
3.16.1 TC001: Customer book event.....	19
3.17 UC0016: Customer cancel booking.....	20
3.17.1 TC001: Customer cancel booking	20

3.18	UC0017: Customer book event.....	21
3.18.1	TC001: Customer book event.....	21
4.	Entry Data.....	22
4.1	DATA...: <Administrator view events >	22
4.2	DATA...: <Administrator creates venue >.....	22
4.3	DATA...: <Administrator assigns event >	23
4.4	DATA...: <Event planner creates event>	23
4.5	DATA...: <Login for different types of user>	24

1. Introduction

1.1 Proposal

The purpose of this document is to define and present the test cases for **project Music Events System of team SDA 1**, covering the test cases for the system use cases.

1.2 Target Users

This document is mainly designed for those responsible for executing the test cases in this project **Music Events System of team SDA 1**.

1.3 Interface for Users (After login)

1.3.1.1 Admin

Admin:admin Navigation

Create and Authenticate Account

View User Information and Assign
Events

Create Venue

View Bookings

View All Events

1.3.1.2 User

User:client Navigation

View Events

Search For and Book Events

View and Cancel Own Bookings

1.3.2 Event Planner

Event Planner:eventplanner Navigation

View and Cancel Events and Bookings

Create Event

2. Covered Requirements

This section lists the system requirements covered in the test cases.

2.1 Functional or Product Requirements

Note: All tests should start from <https://nine0007-1b.onrender.com/musicsystem0918/login>

Requirement Identifier	Requirement Name
UC000	All types of user (admin, customer and event planner) can login to the system
UC001	Administrator views all user
UC002	Administrator authorize user as event planner
UC003	Administrator creates venue
UC004	Administrator creates user account
UC005	Administrator view events
UC006	Administrator view bookings
UC007	Event planner view events
UC008	Event planner view booking
UC009	Event planner cancel booking
UC010	Event planner creates musical events
UC011	Event planner modify event (change time, date, ticket price)
UC012	Event planner cancel event
UC0013	Customer view events
UC0014	Customer view bookings
UC0015	Customer book event
UC0016	Customer cancel booking
UC0017	Customer book event

3. Functional Test Cases

This section describes the test cases that cover the product requirements of the system.

3.1 UC000: All types of user (admin, customer and event planner) can login to the system

Data sample:

Username: “admin”, password: “admin”

Username: “client”, password: “client”

Username: “eventplanner”, password: “eventplanner”

3.2 UC001: Administrator views all user

3.2.1 TC001: Administrator views all user created successfully

Test Type: Functional	Execution Type: Manual
Objective: Verify if the administrator can view all users created	
Setup: <ul style="list-style-type: none"> • The system is running and connected with the database. • Some users have been created by the admin and stored in the database. 	
Pre-Conditions: <ul style="list-style-type: none"> • The system is up and running • The administrator successfully login to the system • There are users stored in the database 	
Notes: <ol style="list-style-type: none"> 1. The administrator needs to successfully login to the system. 2. Click ‘View User Information and Assign Events’ menu 3. The page should show users information <p><i>*The administrator can view all the user information</i></p> <p>Data sample: Username: “admin”, password: “admin”</p>	
Time constraint: Minimum: 5 min Maximum: 25 min	

3.3 UC002: Administrator authorize user as event planner

3.3.1 TC001: Administrator assigns events to the event planner

Test Type: Functional	Execution Type: Manual
Objective: Verify if the admin can assign events to the event planner	
Setup: <ul style="list-style-type: none"> • The events are stored in the database and are available to be assigned • The event planners are created and stored in the database. 	
Pre-Conditions: <ul style="list-style-type: none"> • The events are stored in the database and are available to be assigned • The event planners are created and stored in the database. • The system is running • Logged in as the administrator 	
Notes: <ol style="list-style-type: none"> 1. Admin user logs in 2. Click 'View User Information and Assign Events' menu 3. The page should show users information 4. Admin click 'Assigned Event' button after an event planner record 5. Choose a event with the Dropbox 6. Click 'submit' button <p><i>*The event table for the particular event planner gets updated.</i></p> <p>Data sample: Username: "admin", password: "admin" Selected User: "eventplanner", assigned event: "Event22"</p>	
Time constraint: Minimum: 10 min Maximum: 25 min	

3.4 UC003: Administrator creates venue

3.4.1 TC001: Administrator creates venue

Test Type: Functional	Execution Type: Manual
Objective: Verify if the administrator can creates venues	
Setup: <ul style="list-style-type: none"> The system is running and connected with the database. Logged in as administrator 	
Pre-Conditions: <ul style="list-style-type: none"> The system is up and running and database is online 	
Notes: <ol style="list-style-type: none"> Admin logs in Click 'Create Venue' menu Input data list below to create a venue <p>Other:</p> <ol style="list-style-type: none"> Try to create venue with some inputs missing (venue name, total capacity...) Venue name can only take letters, location can contains letters and digits with no special characters The sum of the vip, mosh, standing and seated capacity should not exceed the total capacity Try to create a duplicated venue with the same venue name <p><i>*The administrator should be redirected to the navigation page when the venue has been created successfully</i></p> <p><i>*A message saying: "Venue exists" should be displayed when a duplicated venue name is submitted.</i></p> <p><i>*A message saying: "Capacity is too big" should be displayed when the total capacity entered is greater than 50000 or less than 1.</i></p> <p><i>*"Missing input" should be popped up when there is at least one input is missing when the form is submitted.</i></p> <p>Data sample:</p> <p>Username: "admin", password: "admin"</p> <p>Venue Name: "Nona Football Court"</p> <p>Location: "Sydney"</p> <p>Total Capacity: 17000</p> <p>VIP capacity: 1000</p> <p>Mosh Capacity: 1000</p> <p>Seated Capacity: 5000</p> <p>Standing Capacity: 10000</p>	

Time constraint:

Minimum: 20 min

Maximum: 25 min

3.5 UC004: Administrator creates user account**3.5.1 TC001: Administrator creates user account**

Test Type: Functional	Execution Type: Manual
Objective: Verify if the administrator can creates venues	
Setup: <ul style="list-style-type: none"> The system is running and connected with the database. 	
Pre-Conditions: <ul style="list-style-type: none"> The system is up and running and database is online 	
Notes: <ol style="list-style-type: none"> Admin logs in Click 'Create and Authenticate Account' menu Input data list below to create a user Data sample: Username: "admin", password: "admin" NewUser: Username: Testuser01 Password: Testuser01 Role: User	

3.6 UC005: Administrator view events**3.6.1 TC001: Administrator view events**

Test Type: Functional	Execution Type: Manual
Objective:	

Verify if the administrator can creates venues
Setup: <ul style="list-style-type: none"> The system is running and connected with the database.
Pre-Conditions: <ul style="list-style-type: none"> The system is up and running and database is online
Notes: <ol style="list-style-type: none"> Admin logs in Click 'View All Events' menu All events added should show on page Data sample: Username: "admin", password: "admin"

3.7 UC006: Administrator view bookings

3.7.1 TC001: Administrator view bookings

Test Type: Functional	Execution Type: Manual
Objective: Verify if the administrator can creates venues	
Setup: <ul style="list-style-type: none"> The system is running and connected with the database. 	
Pre-Conditions: <ul style="list-style-type: none"> The system is up and running and database is online 	
Notes: <ol style="list-style-type: none"> Admin logs in Click 'View All Bookings menu All bookings added should show on page Data sample: Username: "admin", password: "admin"	

3.8 UC007: Event planner view events

3.8.1 TC001: Event planner view events

Test Type: Functional	Execution Type: Manual
Objective: Verify if the administrator can creates venues	
Setup: <ul style="list-style-type: none"> The system is running and connected with the database. 	
Pre-Conditions: <ul style="list-style-type: none"> The system is up and running and database is online 	
Notes: <ol style="list-style-type: none"> Event planner logs in Click 'View and Cancel Events and Bookings' menu All events assigned to this user should show on page <p>Data sample: Username: "eventplanner", password: "eventplanner"</p>	

3.9 UC008: Event planner view booking

3.9.1 TC001: Event planner view booking

Test Type: Functional	Execution Type: Manual
Objective: Verify if the administrator can creates venues	
Setup: <ul style="list-style-type: none"> The system is running and connected with the database. 	
Pre-Conditions: <ul style="list-style-type: none"> The system is up and running and database is online 	

Notes:

1. Event planner logs in
2. Click 'View and Cancel Events and Bookings' menu
3. All events assigned to this user should show on page
4. Click 'view all booking' button for event20 record(show below)

20	Event20	Sophie Y	2333	2077-08-20 09:12:11.0	2077-08-20 09:12:11.0 3
----	---------	----------	------	-----------------------	-------------------------

Delete	Update
View All Booking	

5. All bookings show on the page

Data sample:**Username: "eventplanner", password: "eventplanner"****3.10 UC009: Event planner cancel booking****3.10.1 TC001: Event planner cancel booking**

Test Type: Functional	Execution Type: Manual
Objective: Verify if the administrator can creates venues	
Setup: <ul style="list-style-type: none"> The system is running and connected with the database. 	
Pre-Conditions: <ul style="list-style-type: none"> The system is up and running and database is online 	

Notes:

1. Event planner logs in
2. Click 'View and Cancel Events and Bookings' menu
3. All events assigned to this user should show on page
4. Click 'view all booking' button for event20 record(show below)

20	Event20	Sophie Y	2333	2077-08-20 09:12:11.0	2077-08-20 09:12:11.0 3
----	---------	----------	------	-----------------------	-------------------------

Delete	Update
View All Booking	

5. All bookings show on the page
6. Choos one booking record and click 'cancel booking' button

Data sample:**Username: "eventplanner", password: "eventplanner"****3.11 UC0010: Event planner creates musical events****3.11.1 TC001: Event planner creates musical events**

Test Type:	Execution Type:
Functional	Manual
Objective:	
Verify if the event planner can create musical event	
Setup:	
<ul style="list-style-type: none"> The system is running and connected with the database. Logged in as event planner 	
Pre-Conditions:	
<ul style="list-style-type: none"> The event planner has logged into the system Database is connected with the system 	

Notes:

1. Event planner logs in
2. Click 'Create Event' button
3. Input data listed below
4. The event should be created and automatically assigned to the event planner
5. The event is created by each event planner and is automatically assigned to the event planner
- Other:
6. Event name can contain letters, digits or special characters
7. The event total capacity is integer value from 1 to 50000
8. Artist name contains only letters and digits and can be duplicated
9. Try to create duplicated event with the same event name
10. The time period of each event created should not be overlapped
11. Try to submit the form without venue assigned

**Event planner be redirected to the event table page if there is no venue to be assigned*

**Message "This event exists" should be popped up if a duplicated event name is submitted*

**Message "Time overlapped" should be displayed if the new event submitted has the time period which overlaps with other events.*

**Event planner should be redirected to the event table page once the event is successfully created*

Data sample:

Username: "event planner", password: "event planner"

Event Name: "EventTestForAss"

Artist Name: "Peter Lin"

Total Capacity: 20000

Start Time: 2023-10-27 21:20:00

End Time: 2023-10-27 21:20:10

Venue ID: 11

Time constraint:

Minimum: 20 min

Maximum: 25 min

3.12 UC0011: Event planner modify event (change time, date, ticket price)**3.12.1 TC001: Event planner modify event**

Test Type:	Execution Type:
Functional	Manual

Objective:

Verify if the administrator can create venues

Setup:

- The system is running and connected with the database.

Pre-Conditions:

- The system is up and running and database is online

Notes:

1. Event planner logs in
2. Click 'View and Cancel Events and Bookings' menu
3. All events assigned to this user should show on page
4. Click 'Update' button for event record created in UC0010 (shown below)

67	EventTestForAss	Peter Lin	20000	2023-10-27 21:20:10.0	2023-10-27 21:20:00.0	11
----	-----------------	-----------	-------	-----------------------	-----------------------	----

View All Booking

Delete

Update

View All Booking

5. A update even page opens with event information
6. Input data below and click save button
7. Event should be updated

Data sample:

Username: "eventplanner", password: "eventplanner"

Event Name: "EventTestForAss01"

Artist Name: "Peter Wang"

Total Capacity: 20000

Start Time: 2023-10-28 21:20:10

End Time: 2023-10-28 21:20:20

VIP Price: 100

Mosh Price: 80

Standing Price: 20

Seated Price: 50

3.13 UC0012: Event planner cancel event**3.13.1 TC001: Event planner cancel event**

Test Type:	Execution Type:
-------------------	------------------------

Functional	Manual
Objective: Verify if the administrator can creates venues	
Setup: <ul style="list-style-type: none"> The system is running and connected with the database. 	
Pre-Conditions: <ul style="list-style-type: none"> The system is up and running and database is online 	
Notes: <ol style="list-style-type: none"> Event planner logs in Click 'View and Cancel Events and Bookings' menu All events assigned to this user should show on page Click 'Delete' button for event record updated in UC0011(show below) <div> <div>67</div> <div>EventTestForAss01</div> <div>Peter Wang</div> <div>20000</div> <div>2023-10-28 21:20:20.0</div> <div>2023-10-28 21:20:10.0</div> <div>11</div> <div>Delete</div> <div>Update</div> <div>View All Booking</div> </div> <ol style="list-style-type: none"> The view event page should be refreshed and the event is deleted <p>Data sample: Username: "eventplanner", password: "eventplanner"</p>	

3.14 UC0013: Customer view events

3.14.1 TC001: Customer view events

Test Type:	Execution Type:
Functional	Manual
Objective: Verify if the administrator can creates venues	
Setup: <ul style="list-style-type: none"> The system is running and connected with the database. 	
Pre-Conditions: <ul style="list-style-type: none"> The system is up and running and database is online 	

Notes:

1. Customer logs in
2. Click 'View Events' menu
3. All events added should show on page

Data sample:**Username: "client", password: "client"****3.15 UC0014: Customer view bookings****3.15.1 TC001: Customer view bookings**

Test Type: Functional	Execution Type: Manual
Objective: Verify if the administrator can creates venues	
Setup: <ul style="list-style-type: none"> The system is running and connected with the database. 	
Pre-Conditions: <ul style="list-style-type: none"> The system is up and running and database is online 	
Notes: <ol style="list-style-type: none"> 1. Customer logs in 2. Click 'View and Cancel Own Bookings' menu 3. All bookings of this customer should show on page Data sample: Username: "client", password: "client"	

3.16 UC0015: Customer book event**3.16.1 TC001: Customer book event**

Test Type: Functional	Execution Type: Manual
Objective:	

Verify if the administrator can creates venues

Setup:

- The system is running and connected with the database.

Pre-Conditions:

- The system is up and running and database is online

Notes:

- Customer logs in
- Click 'Search For and Book Events' menu
- Click 'buy' for event Testing Concert3

Number	Event name	Name	Time	End Time	VIP(s)	Mosh(s)	Standing(s)	Seated(s)	Name	Location	Capacity	Capacity	Capacity	Capacity
1	Testing Concert3	John Doe	2024-03-13 00:00:00.0	2077-08-20 09:12:11.0	0	0	0	0	Vennue Test	Melbourne XXX	2	5	5	7

[Buy](#)

- Input at least one ticket and manually add total amount or data below
- Click 'submit'
- Should navigate to orders page
- A order with the amount as Total in step 4 and the newest date is listed

Data sample:

Username: "client", password: "client"

Order:

Event Name	Testing Concert12222
VIP Capacity	2
VIP Price(\$)	1
Purchase Quantity	1
Mosh Capacity	0
Mosh Price(\$)	2
Purchase Quantity	1
Standing Capacity	0
Standing Price	3
Purchase Quantity	0
Seated Capacity	1
Seated Price	4
Purchase Quantity	0
Total(\$)	3
<input type="button" value="Submit"/> Call Back	

3.17 UC0016: Customer cancel booking

3.17.1 TC001: Customer cancel booking

Test Type:	Execution Type:							
Functional	Manual							
Objective:								
Verify if the administrator can creates venues								
Setup:								
<ul style="list-style-type: none">The system is running and connected with the database.								
Pre-Conditions:								
<ul style="list-style-type: none">The system is up and running and database is online								
Notes:								
<div><div><div>1. Customer logs in</div><div>2. Click ‘View and Cancel Own Bookings’ menu</div><div>3. All bookings of this customer should show on page</div><div>4. Choose the order added in UC0015</div></div><table><tr><td>1</td><td>38</td><td>client</td><td>3</td><td>2023-09-18</td><td>Details</td><td>Cancel</td></tr></table><div><div>5. Click ‘Cancel</div><div>6. Page should be refreshed and the order is canceled</div></div></div>		1	38	client	3	2023-09-18	Details	Cancel
1	38	client	3	2023-09-18	Details	Cancel		
Data sample:								
Username: “client”, password: “client”								
Order:								
Note: error will occur if choose the order is for a event of which all the ticket price is 0.								

3.18 UC0017: Customer book event

3.18.1 TC001: Customer book event

Test Type: Functional	Execution Type: Manual
Objective: Verify if the administrator can creates venues	
Setup: <ul style="list-style-type: none"> The system is running and connected with the database. 	

Pre-Conditions:

- The system is up and running and database is online

Notes:**Search by input:**

1. Customer logs in
2. Click 'Search For and Book Events' menu
3. Enter 'event' in the search input box and click 'Go!'
4. Should can search out result

Search by data:

5. Customer logs in
6. Click 'View Events' menu
7. A button named 'Show Events in Next 6 Months' on this page
8. Click and will get Events in Next 6 Months

Data sample:

Username: "client", password: "client"

4. Entry Data

This section describes the entry data that will be used by more than one test case, avoiding data replication. These data are referenced by the test cases.

4.1 DATA...: <Administrator view events >

Description:

This data set is used to test if the administrator can view the event

Field	Value
1. username	"admin"
2. password	"admin"

4.2 DATA...: <Administrator creates venue >

Description:

This data set is used to test creating venue for the admin

Field	Value
-------	-------

1. username	“admin”
2. password	“admin”
3.venue name	“MCG”
4.location	“Melbourne CBD”
5.vipcapacity	1000
6. mosh capacity	1000
7. standing capacity	4000
8.seated capacity	10000

4.3 DATA...: <Administrator assigns event >

Description:

This data set is used to test assigning of the event to event planner for administrator

Field	Value
1. username	“admin”
2. password	“admin”
3.selected user	EventPlanner1
4.assign event name	“event6”

4.4 DATA...: <Event planner creates event>

Description:

This data set is used to test if the event planner can create event

Field	Value
1. username	“EventPlanner1”
2. password	“EventPlanner1”
3.Event name	“Happy World”
4.Artist name	“Joe Lee”

5.Total capacity	10000
6:Start Time	2023-09-18 20:30:00
7.End Time	2023-09-18 23:30:00
8.Venue Id	3

4.5 DATA...: <Login for different types of user>

Description:

This data set is used to test login of the system for different types of users

Field	Value
1. event planner username	"EventPlanner1"
2. event planner password	"EventPlanner1"
3.admin username	"admin"
4.admin password	"admin"
5.customer username	"client2"
6:customer password	"client2"