Requirements Based Test Cases for <MEAT>

<1.0>
<10/1/2016>

<Peng Cheng, Mengjiao Zeng, Sunghun Park>
<Group 7>

Document Revision History

Rev	Date	Author	Change Description

.

Table of Contents

1. Test Requirements

1.1 Objective

This document describes the collection of requirements-based test cases for the MEAT system. It also provides Traceability Matrix to show which test cases verify which requirements.

1.2 Definitions and Acronyms

- MEAT : The Meeting Engagement and Tracking System
- Meeting Organizer : An employee who organize a meetings with coordinate conference rooms and assemble attendees.
- Attendees : Employees who are supposed to attend a meeting.

1.3 Traceability Matrix

This Traceability Matrix is to show which test cases verify which requirements.

REQ \ TC	**ER1**	**ER2**	**FR1**	**FR2**	**FR3**	**FR4**	**FR5**	**FR6**	**FR7**	**FR8**	**FR9**	**FR10**	**FR11**	**FR12**	**FR13**	**FR14**	**FR15**	**FR16**	**FR17**
TC_E1.1	X																		
TC_E1.2		X																	
TC_F1.1			X				X	X		X	X	X							
TC_F1.2			X	X			X	X			X								
TC_F1.3			X	X			X	X			X								
TC_F1.4			X	X							X	X	X						
TC_F1.5			X	X							X	X	X						
TC_F1.6			X	X							X	X	X						
TC_F1.7			X	X							X	X	X						
TC_F1.8			X	X								X	X						
TC_F1.9			X	X								X	X						
TC_F1.10			X	X								X	X						
TC_F1.11			X	X															
TC_F1.12			X	X						X									
TC_F1.13			X	X						X									
TC_F1.14			X	X			X	X											
TC_F1.15			X	X						X									
TC_F1.16			X	X						X									
TC_F2.1				X				X											
TC_F2.2				X									X						
TC_F2.3				X															
TC_F2.4				X															
TC_F2.5				X															
TC_F3.1					X														

REQ \ TC	ER1	ER2	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9	FR10	FR11	FR12	FR13	FR14	FR15	FR16	FR17
TC_F3.2					X														
TC_F3.3					X														
TC_F12.1														X					
TC_F12.2														X	X				
TC_F12.3														X	X				
TC_F12.4														X	X				
TC_F12.5														X	X				
TC_F12.6														X	X				
TC_F12.7														X	X				
TC_F12.8														X	X				
TC_F12.9														X	X				
TC_F12.10														X	X				
TC_F13.1															X				
TC_F14.1																X			
TC_F14.2																X			
TC_F14.3																X			
TC_F15.1									X								X		
TC_F15.2									X								X		
TC_F15.3						X											X		
TC_F15.4																	X		
TC_F15.5						X											X		
TC_F15.6						X											X		
TC_F15.7						X											X		
TC_F15.8									X								X		
TC_F15.9									X								X		
TC_F16.1																		X	
TC_F16.2																		X	
TC_F16.3																		X	
TC_F16.4																		X	
TC_F17.1																			X

2. Test Cases

2.1 Description

This section describes detailed test cases scenarios based on requirements for the MEAT system. The naming rule of the test cases is concatenating "TC" and requirement ID with sequence.

2.2 Test Case

✦ TC_E1.1 Intuitive user interface

Description:	Checks that the system is easy to use intuitively
Test Inputs:	A new recruited employee
Expected Results:	The employee is able to organize meeting and booking vacation after 3 hours learning about the system.
Dependencies:	None
Initialization:	None
Test Steps:	1. A new employee learns by himself along with brief instructions in the system for 3 hours. 2. After that, verify that the employee can organize a meeting by oneself. 3. Verify that the employee can book a vacation by oneself

✦ TC_E1.2 Communication with external database.

Description:	Checks that the system can connect with external user and room master database.
Test Inputs:	Select query ("Select 'x' from tab")
Expected Results:	Java result set containing 'x' string records.
Dependencies:	None
Initialization:	Provide JDBC connection testing java module using query
Test Steps:	1. Put the select query into the connection testing module. 2. Run the module in JVM environment. 3. Verify the result record set contains 'x' string value.

✦ TC_F1.1 Book a meeting successfully.

Description:	Checks that the system provides functionality to schedule a meeting
Test Inputs:	RoomID : B203 StartTime : 201612011030 EndTime : 201612011130 Description : Brainstorming meeting Attendees' ID : E1000, E1111, E1112
Expected Results:	Return success status and Physically stored data in the database.
Dependencies:	None
Initialization:	No overlapped reservation time of rooms and attendees' vacation time conflicts

Test Steps:
1. Select book meeting menu.
2. Put the meeting time and select open room.
3. Description and attendees' ID add.
4. Select "book" menu to save the organized meeting data into database.
5. Verity success messages and the data existence in the database.

✦ TC_F1.2 Book a meeting with error (with null room ID)

Description: Checks that the system disallow to book a meeting with null room ID

Test Inputs: RoomID :
StartTime : 201612011030
EndTime : 201612011130
Description : Brainstorming meeting
Attendees'ID : E1000, E1111, E1112

Expected Results: Return fail status and
Booking error message ("empty room ID")

Dependencies: None

Initialization: None

Test Steps:
1. Select book meeting menu.
2. Put meeting time, but leave room information empty.
3. Description and attendees' ID add.
4. Select "book" menu to save the organized meeting data into database.
5. Verity fail messages with malformed room ID.

✦ TC_F1.3 Book a meeting with error (with room ID not in room database)

Description: Checks that the system disallow to book a meeting with null room ID

Test Inputs: RoomID : ZZ123
StartTime : 201612011030
EndTime : 201612011130
Description : Brainstorming meeting
Attendees' ID : E1000, E1111, E1112

Expected Results: Return fail status and
Booking error message ("Room ID is not in the database ").

Dependencies: None

Initialization: None

Test Steps:
1. Select book meeting menu.
2. Put the meeting time and incorrect room ID.
3. Description and attendees' ID add.
4. Select "book" menu to save the organized meeting data into database.
5. Verity fail messages with no such room ID in the database.

✦ TC_F1.4 Book a meeting with error (with empty StartTime)

Description: Checks that the system disallow to book a meeting with empty starting time

Test Inputs: RoomID : B203
StartTime :
EndTime : 201612011130
Description : Brainstorming meeting
Attendees' ID : E1000, E1111, E1112

Expected Return fail status and

Results: Booking error message (“Meeting starting time is null. ”).

Dependencies: None

Initialization: None

Test Steps:

1. Select book meeting menu.
2. Put the meeting time and room ID with leaving start time empty
3. Description and attendees’ID add.
4. Select “book” menu to save the organized meeting data into database.
5. Verity fail messages with starting time is empty.

✦ TC_F1.5 Book a meeting with error (with empty EndTime)

Description: Checks that the system disallow to book a meeting with empty ending time

Test Inputs: RoomID : B203
StartTime : 201612011030
EndTime :
Description : Brainstorming meeting
Attendees’ID : E1000, E1111, E1112

Expected Results: Return fail status and
Booking error message (“Meeting ending time is null. ”).

Dependencies: None

Initialization: None

Test Steps:

1. Select book meeting menu.
2. Put the meeting time and room ID with leaving end time empty
3. Description and attendees’ID add.
4. Select “book” menu to save the organized meeting data into database.
5. Verity fail messages with ending time is empty.

✦ TC_F1.6 Book a meeting with error (with out of time range)

Description: Checks that the system disallow to book a meeting out of time range

Test Inputs: RoomID : B203
StartTime : 201613011030
EndTime : 201612011130
Description : Brainstorming meeting
Attendees : E1000, E1111, E1112

Expected Results: Return fail status and
Booking error message (“Starting time is out of time range ”).

Dependencies: None

Initialization: None

Test Steps:

1. Select book meeting menu.
2. Put the meeting time with month value 13 in start time, and room ID
3. Description and attendees’ID add.
4. Select “book” menu to save the organized meeting data into database.
5. Verity fail messages with starting time is out of time range.

✦ TC_F1.7 Book a meeting with error (with out of time range)

Description: Checks that the system disallow to book a meeting out of time range

Test Inputs: RoomID : B203
StartTime : 201612011030
EndTime : 201612321130

	Description : Brainstorming meeting Attendees'ID : E1000, E1111, E1112
Expected Results:	Return fail status and Booking error message ("End time is out of time range ").
Dependencies:	None
Initialization:	None
Test Steps:	1. Select book meeting menu. 2. Put the meeting time with day value 32 in end time, and room ID 3. Description and attendees' ID add. 4. Select "book" menu to save the organized meeting data into database. 5. Verity fail messages with ending time is out of time range.

✦ **TC_F1.8** **Book a meeting with error (with reversed start time and end time)**

Description:	Checks that the system disallow to book a meeting with end time before start time
Test Inputs:	RoomID : B203 StartTime : 201612011130 EndTime : 201612011030 Description : Brainstorming meeting Attendees : E1000, E1111, E1112
Expected Results:	Return fail status and Booking error message ("End time precedes Start time.").
Dependencies:	None
Initialization:	None
Test Steps:	1. Select book meeting menu. 2. Put the meeting time with hour value 10 AM in end time and 11 AM in start time, and room ID 3. Description and attendees'ID add. 4. Select "book" menu to save the organized meeting data into database. 5. Verity fail messages with end time preceding start time.

✦ **TC_F1.9** **Book a meeting with error (with malformed format yyyymmddhh24mi)**

Description:	Checks that the system disallow to book a meeting with **start time** with "yymmddhh24mi" time format
Test Inputs:	RoomID : B203 StartTime : 1612011130 EndTime : 201612011030 Description : Brainstorming meeting Attendees : E1000, E1111, E1112
Expected Results:	Return fail status and Booking error message ("Start time format is malformed.").
Dependencies:	None
Initialization:	None
Test Steps:	1. Select book meeting menu. 2. Put the meeting time with yymmdd24mi-format in start time, and room ID 3. Description and attendees'ID add. 4. Select "book" menu to save the organized meeting data into database.

5. Verity fail messages with malformed start time format.

✦ TC_F1.10 Book a meeting with error (with malformed format yyyymmddhh24mi)

Description: Checks that the system disallow to book a meeting with **end time** with "yymmddhh24mi" time format

Test Inputs: RoomID : B203
StartTime : 201612011130
EndTime : 1612011030
Description : Brainstorming meeting
Attendees'ID : E1000, E1111, E1112

Expected Results: Return fail status and
Booking error message ("Start time format is malformed.").

Dependencies: None

Initialization: None

Test Steps:
1. Select book meeting menu.
2. Put the meeting time with yymmdd24mi-format in end time, and room ID
3. Description and attendees'ID add.
4. Select "book" menu to save the organized meeting data into database.
5. Verity fail messages with malformed end time format.

✦ TC_F1.11 Book a meeting with error (with empty description)

Description: Checks that the system disallow to book a meeting with empty description

Test Inputs: RoomID : B203
StartTime : 201612011030
EndTime : 201612011130
Description :
Attendees'ID : E1000, E1111, E1112

Expected Results: Return fail status and
Booking error message ("Meeting description is empty").

Dependencies: None

Initialization: None

Test Steps:
1. Select book meeting menu.
2. Put the meeting time and room ID correctly
3. Leave description empty and attendees'ID add.
4. Select "book" menu to save the organized meeting data into database.
5. Verity fail messages with empty description

✦ TC_F1.12 Book a meeting with error (with empty attendees)

Description: Checks that the system disallow to book a meeting with empty attendees

Test Inputs: RoomID : B203
StartTime : 201612011030
EndTime : 201612011130
Description : Brainstorming meeting
Attendees'ID :

Expected Results: Return fail status and
Booking error message ("Meeting attendees are empty").

Dependencies: None

Initialization: None

Test Steps:
1. Select book meeting menu.
2. Put the meeting time and room ID correctly
3. Put description and leave attendees'ID empty.
4. Select "book" menu to save the organized meeting data into database.
5. Verity fail messages with empty attendees.

✦ TC_F1.13 Book a meeting with error (with no such employees in the database)

Description: Checks that the system disallow to book a meeting with no such employee ID

Test Inputs:
RoomID : B203
StartTime : 201612011030
EndTime : 201612011130
Description : Brainstorming meeting
Attendees'ID : T1102

Expected Results: Return fail status and
Booking error message ("Meeting attendees are empty").

Dependencies: None

Initialization: T1102 is not stored in the database as an employee ID.

Test Steps:
1. Select book meeting menu.
2. Put the meeting time and room ID correctly
3. Put description and invalid attendees'ID add.
4. Select "book" menu to save the organized meeting data into database.
5. Verity fail messages with no such attendees.

✦ TC_F1.14 Book a meeting with error (room already occupied)

Description: Checks that the system disallow to book a meeting with a preoccupied room

Test Inputs:
RoomID : B203
StartTime : 201612011030
EndTime : 201612011130
Description : Brainstorming meeting
Attendees'ID : E1000, E1111, E1112

Expected Results: Return fail status and
Booking error message ("Meeting room is already occupied").

Dependencies: None

Initialization: The room (B203) is already scheduled by another meeting at same time.

Test Steps:
1. Select book meeting menu.
2. Put the meeting time and room ID
3. Description and attendees'ID add.
4. Select "book" menu to save the organized meeting data into database.
5. Verity fail messages with preoccupied room.

✦ TC_F1.15 Book a meeting with error (any of attendees' another meeting at same time)

Description: Checks that the system disallow to book a meeting with overlapped meeting time of attendees.

Test Inputs:
RoomID : B203
StartTime : 201612011030
EndTime : 201612011130

Description : Brainstorming meeting
Attendees' ID : E1000, E1111, E1112

Expected Results: Return fail status and
Booking error message ("Attendees' meeting time conflict").

Dependencies: None

Initialization: "Tom Alice" is already scheduled by another meeting of different rooms at the same time.

Test Steps:
1. Select book meeting menu.
2. Put the meeting time and room ID
3. Description and attendees'ID add.
4. Select "book" menu to save the organized meeting data into database.
5. Verity fail messages with attendees' scheduled time conflict.

✦ TC_F1.16 Book a meeting with error (any of attendees' vacation at same time)

Description: Checks that the system disallow to book a meeting with overlapped meeting time of attendees.

Test Inputs: RoomID : B203
StartTime : 201612011030
EndTime : 201612011130
Description : Brainstorming meeting
Attendees : E1000, E1111, E1112

Expected Results: Return fail status and
Booking error message ("Attendees' vacation time is overlapped").

Dependencies: None

Initialization: An employee having ID "E1000" already booked his vacation at the same time with the meeting.

Test Steps:
1. Select modify meeting menu.
2. Put the meeting time and room ID
3. Description and attendees'ID add.
4. Select "book" menu to save the organized meeting data into database.
5. Verity fail messages with attendees' vacation time conflict.

✦ TC_F2.1 Modify a meeting successfully (Changing a room)

Description: Checks that the system provides functionality to change room of the scheduled meeting. (B203→B204)

Test Inputs: Meeeting ID : M2016120110301201223
RoomID : B204

Expected Results: Return success status and
Physically store the changed data in the database.

Dependencies: None

Initialization: A meeting at B203 is already booked and stored in the database.

Test Steps:
1. Select modify meeting menu.
2. Check schedule of room (B203) with specific period
3. Select the time slot of list with the meeting ID and "modify" menu choose.
4. Put new Room ID and select "book" menu again.
5. Verity success messages and the changed data existence in the database.

✦ TC_F2.2 Modify a meeting successfully (Changing time)

Description:	Checks that the system provides functionality to change time of the scheduled meeting. (Dec 1→Dec 5)
Test Inputs:	Meeeting ID : M2016120110301201223 StartTime : 201612051030 EndTime : 201612051130
Expected Results:	Return success status and Physically store the changed data in the database.
Dependencies:	None
Initialization:	A meeting with the meeting ID is already booked and stored in the database.
Test Steps:	1. Select modify meeting menu. 2. Check schedule of a room (B204) with the specific period 3. Select the time slot of list with the meeting ID and “modify” menu choose. 4. Put new start and end time and select “book” menu again. 5. Verity success messages and the changed data existence in the database.

✦ TC_F2.3 Modify a meeting successfully (Changing description)

Description:	Checks that the system provides functionality to change description of the scheduled meeting.
Test Inputs:	Meeting ID : M2016120110301201223 Description: Job conference
Expected Results:	Return success status and Physically store the changed data in the database.
Dependencies:	None
Initialization:	A meeting with the meeting ID is already booked and stored in the database.
Test Steps:	1. Select modify meeting menu. 2. Check schedule of a room (B204) with the specific period 3. Select the time slot of list with the meeting ID and “modify” menu choose. 4. Put new description and select “book” menu again. 5. Verity success messages and the changed data existence in the database.

✦ TC_F2.4 Modify a meeting successfully (Changing attendees)

Description:	Checks that the system provides functionality to change attendees of the scheduled meeting.
Test Inputs:	Meeting ID : M2016120110301201223 Attendees’ID : E1000, E1111, E2112
Expected Results:	Return success status and Physically store the changed data in the database.
Dependencies:	None
Initialization:	A meeting with the meeting ID is already booked and stored in the database.
Test Steps:	1. Select modify meeting menu. 2. Check schedule of a room (B204) with the specific period 3. Select the time slot of list with the meeting ID and “modify” menu choose. 4. Put new employees’ name and remove registered attendees’ name, and

then select “book” menu again.
5. Verity success messages and the changed data existence in the database.

✦ TC_F2.5 Modify a meeting successfully (Changing all)

Description:	Checks that the system provides functionality to change room, start time, end time, description, and attendees of the scheduled meeting.
Test Inputs:	Meeeting ID : M2016120110301201223 RoomID : B205 StartTime : 201612061030 EndTime : 201612061130 Description: Junior board meeting Attendees’ID : E3000, E3001, E3002
Expected Results:	Return success status and Physically store the changed data in the database.
Dependencies:	None
Initialization:	A meeting with the meeting ID is already booked and stored in the database.
Test Steps:	1. Select modify meeting menu. 2. Check schedule of a room (B204) with the specific period 3. Select the time slot of list with the meeting ID and “modify” menu choose. 4. Put new room ID, new start and end time, description, and attendees, and finally select “book” menu again. 5. Verity success messages and the changed data existence in the database.

✦ TC_F3.1 Cancel a meeting successfully

Description:	Checks that the system provides functionality to cancel the scheduled meeting.
Test Inputs:	Meeting ID : M2016120110301201223
Expected Results:	Return success status and Physically deleted the data in the database.
Dependencies:	None
Initialization:	A meeting with the meeting ID is already booked and stored in the database.
Test Steps:	1. Select cancel meeting menu. 2. Check schedule of a room (B204) with the specific period 3. Select the time slot of list with the meeting ID and “cancel” menu choose. 4. Verity success messages and deletion of the data in the database.

✦ TC_F3.2 Cancel a meeting with error (empty meeting ID)

Description:	Checks that the system disallow to cancel a meeting with empty meeting ID.
Test Inputs:	Meeting ID :
Expected Results:	Return fail status and Cancel error message (“Meeting ID is empty.”).
Dependencies:	None
Initialization:	A meeting with the meeting ID is already booked and stored in the database.
Test Steps:	1. Select cancel meeting menu. 2. Check schedule of a room (B204) with the specific period 3. Select the time slot of list with the meeting ID, and then leave out the

meeting ID with empty, finally “cancel” menu choose.
4. Verity fail messages with empty meeting ID.

✦ TC_F3.3 Cancel a meeting with error (meeting ID not in the database)

Description: Checks that the system disallow to book a meeting with meeting ID not in the database

Test Inputs: Meeting ID : 9999999999999999999

Expected Results: Return fail status and
Cancel error message (“Meeting ID is not in database.”).

Dependencies: None

Initialization: A meeting with the meeting ID is already booked and stored in the database.

Test Steps:

1. Select cancel meeting menu.
2. Check schedule of a room (B204) with the specific period
3. Select the time slot of list with the meeting ID, and then change the meeting ID to malformed one, finally “cancel” menu choose..
4. Verity fail messages with empty meeting ID.

✦ TC_F12.1 Book a vacation successfully

Description: Checks that the system provides functionality to book an employee’s vacation successfully.

Test Inputs: Employee ID : E1001
StartDate : 20161220
EndDate : 20161225

Expected Results: Return success status and
Physically store the booked data in the database.

Dependencies: None

Initialization: None

Test Steps:

1. Select book vacation menu.
2. Put employee’s name and vacation starting date and end date.
3. Select “book” menu.
4. Verity success messages and the booked data existence in the database.

✦ TC_F12.2 Book a vacation with error (conflict with scheduled meeting time)

Description: Checks that the system disallow to book a vacation when vacation date conflicts with scheduled meeting time.

Test Inputs: Employee ID : E1001
StartDate : 20161205
EndDate : 20161212

Expected Results: Return fail status and
Booking vacation error message (“Time conflict with scheduled meeting time.”).

Dependencies: None

Initialization: The employee’s meeting is already scheduled between vacation time.

Test Steps:

1. Select book vacation menu.
2. Put employee’s ID and vacation starting date and end date.
3. Select “book” menu.
4. Verity fail messages with meeting time conflict.

✦ TC_F12.3 Book a vacation with error (name is empty)

Description: Checks that the system disallow to book a vacation with empty name.

Test Inputs: Employee ID :
StartDate : 20161205
EndDate : 20161212

Expected Results: Return fail status and
Booking vacation error message ("Employee's ID is empty").

Dependencies: None

Initialization: None

Test Steps:

1. Select book vacation menu.
2. Put vacation starting date and end date except employee's ID.
3. Select "book" menu.
4. Verity fail messages with empty employee's name.

✦ TC_F12.4 Book a vacation with error (start date is empty)

Description: Checks that the system disallow to book a vacation with empty starting date.

Test Inputs: Employee ID : E1001
StartDate :
EndDate : 20161212

Expected Results: Return fail status and
Booking vacation error message ("Vacation start date is empty").

Dependencies: None

Initialization: None

Test Steps:

1. Select book vacation menu.
2. Put employee's ID and end date except for start date.
3. Select "book" menu.
4. Verity fail messages with empty start date.

✦ TC_F12.5 Book a vacation with error (end date is empty)

Description: Checks that the system disallow to book a vacation with empty ending date.

Test Inputs: Employee ID : E1001
StartDate : 20161206
EndDate :

Expected Results: Return fail status and
Booking vacation error message ("Vacation end date is empty").

Dependencies: None

Initialization: None

Test Steps:

1. Select book vacation menu.
2. Put employee's ID and start date except for end date.
3. Select "book" menu.
4. Verity fail messages with empty end date.

✦ TC_F12.6 Book a vacation with error (end date before start date)

Description: Checks that the system disallow to book a vacation with ending date before starting date.

Test Inputs:	Employee ID : E1001 StartDate : 20161212 EndDate : 20161206
Expected Results:	Return fail status and Booking vacation error message ("Vacation end date precedes start date").
Dependencies:	None
Initialization:	None
Test Steps:	1. Select book vacation menu. 2. Put employee's ID and start date except for end date. 3. Select "book" menu. 4. Verity fail messages with ending date before starting date.

✦ TC_F12.7 Book a vacation with error (malformed start date)

Description:	Checks that the system disallow to book a vacation with malformed starting date
Test Inputs:	Employee ID : E1001 StartDate : 201612061201 EndDate : 20161212
Expected Results:	Return fail status and Booking vacation error message ("Vacation date format is malformed").
Dependencies:	None
Initialization:	None
Test Steps:	1. Select book vacation menu. 2. Put employee's ID, and start date and end date with YYYYMMDDHH24MI format. 3. Select "book" menu. 4. Verity fail messages with vacation start date format error.

✦ TC_F12.8 Book a vacation with error (malformed end date)

Description:	Checks that the system disallow to book a vacation with malformed ending date
Test Inputs:	Employee ID : E1001 StartDate : 20161206 EndDate : 201612121201
Expected Results:	Return fail status and Booking vacation error message ("Vacation date format is malformed").
Dependencies:	None
Initialization:	None
Test Steps:	1. Select book vacation menu 2. Put employee's ID, and start date and end date with YYYYMMDDHH24MI format. 3. Select "book" menu. 4. Verity fail messages with vacation end date format error.

✦ TC_F12.9 Book a vacation with error (invalid start date)

Description:	Checks that the system disallow to book a vacation with invalid stating date
Test Inputs:	Employee ID : E1001 StartDate : 20161306 EndDate : 20161212

Expected Results:	Return fail status and Booking vacation error message ("Vacation date is invalid").
Dependencies:	None
Initialization:	None
Test Steps:	1. Select book vacation menu. 2. Put employee's ID, and start date with month value "13" and end date correctly 3. Select "book" menu. 4. Verity fail messages with invalid vacation start date.

✦ TC_F12.10 Book a vacation with error (invalid end date)

Description:	Checks that the system disallow to book a vacation with invalid stating date
Test Inputs:	Employee ID : E1001 StartDate : 20161206 EndDate : 20161232
Expected Results:	Return fail status and Booking vacation error message ("Vacation date is invalid").
Dependencies:	None
Initialization:	None
Test Steps:	1. Select book vacation menu. 2. Put employee's ID, and start date correctly and end date with date value "32" 3. Select "book" menu. 4. Verity fail messages with invalid vacation start date.

✦ TC_F13.1 Modify a vacation successfully

Description:	Checks that the system provides functionality to change the employee's scheduled vacation time.
Test Inputs:	Vacation ID : V2016120110301201223 StartDate : 20161206 EndDate : 20161212
Expected Results:	Return success status and Physically store the changed data in the database.
Dependencies:	None
Initialization:	The employee's vacation information is already booked and stored in the database.
Test Steps:	1. Select modify vacation menu. 2. Check schedule of employee ("Tom Alice") with specific period 3. Select the time slot of list with vacation ID and "modify" menu choose. 4. Put start date and end date, and finally select "book" menu again. 5. Verity success messages and the changed data existence in the database.

✦ TC_F14.1 Cancel a vacation successfully

Description:	Checks that the system provides functionality to cancel the employee's scheduled vacation.
Test Inputs:	Vacation ID : V2016120110301201223
Expected	Return success status and

Results:	Physically delete the data in the database.
Dependencies:	None
Initialization:	The employee's vacation information is already booked and stored in the database.

Test Steps:

1. Check schedule of employee ("Tom Alice") with specific period
2. Select the time slot of list with vacation ID and "cancel" menu choose.
3. Verity success messages and the deleted data in the database.

✦ TC_F14.2 Cancel a vacation with error (empty vacation ID)

Description:	Checks that the system disallow to cancel a vacation with empty vacation ID.
Test Inputs:	Vacation ID :
Expected Results:	Return fail status and Canceling vacation error message ("Empty vacation ID")..
Dependencies:	None
Initialization:	The employee's vacation information is already booked and stored in the database.

Test Steps:

1. Select cancel vacation menu.
2. Check schedule of employee ("Dash John") with specific period
3. Select the time slot of list with vacation ID and leave out the selected ID with blank, finally "cancel" menu choose.
4. Verity fail messages with empty vacation ID.

✦ TC_F14.3 Cancel a vacation with error (vacation ID not in the database)

Description:	Checks that the system disallow to cancel a vacation with vacation ID not in the database.
Test Inputs:	Vacation ID : 9999999999999999
Expected Results:	Return fail status and Canceling vacation error message ("Vacation ID is not in the database.").
Dependencies:	None
Initialization:	The employee's vacation information is already booked and stored in the database.

Test Steps:

1. Select cancel vacation menu.
2. Check schedule of employee ("Dash John") with specific period
3. Select the time slot of list with vacation ID and change the selected ID with "999999999999999", finally "cancel" menu choose.
4. Verity fail messages with no such vacation ID in the database.

✦ TC_F15.1 Check schedules successfully (Employee's schedules by ID)

Description:	Checks that the system provides functionality to check an employee's schedules using ID.
Test Inputs:	Employee ID : E101 StartDate : 20160805 EndDate : 20161212
Expected Results:	List of the employee's meeting, vacation, and non-vacation date
Dependencies:	None

Initialization: The employee has two meeting schedules and vacation time within the start and end date and the schedules are stored in the database.

Test Steps:
1. Select check schedules menu and choose "by employee" option.
2. Put employee's ID and search starting date and end date.
3. Select "check" menu.
4. Verity List of the employee's meeting, vacation, and non-vacation date.

✦ TC_F15.2 Check schedules successfully (Employee's schedules by name)

Description: Checks that the system provides functionality to check an employee's schedules using name.

Test Inputs: Employee Name : Tom
StartDate : 20160805
EndDate : 20161212

Expected Results: List of the employees' meeting, vacation, and non-vacation date whose name contains Tom

Dependencies: None

Initialization: There are employees name "Tom catty" and "Tom Doggy" and they have a vacation time and meeting time respectively during the period.

Test Steps:
1. Select check schedules menu and choose "by employee" option.
2. Put employee's Name and search starting date and end date.
3. Select "check" menu.
4. Verity List of the employees' meeting, vacation, and non-vacation date having their name contains "Tom".

✦ TC_F15.3 Check schedules successfully (Room schedules)

Description: Checks that the system provides functionality to check a room's schedules.

Test Inputs: Room ID : B203
StartDate : 20160805
EndDate : 20161212

Expected Results: A list of all scheduled meetings of the room

Dependencies: None

Initialization: The room has 10 scheduled meetings within the start and end date and they are stored in the database.

Test Steps:
1. Select check schedules menu and choose "by room" option.
2. Put Room ID and search starting date and end date.
3. Select "check" menu.
4. Verity List of the scheduled meeting with date of the room.

✦ TC_F15.4 Check schedules successfully (Universal schedules)

Description: Checks that the system provides functionality to check company's universal schedules.

Test Inputs: StartDate : 20160805
EndDate : 20161212

Expected Results: A list of all scheduled meetings of all room

Dependencies: None

Initialization: The company has 20 scheduled meetings within the start and end date and the meeting information is stored in the database.

Test Steps:
1. Select check schedules menu and choose “by universal” option
2. Put search starting date and end date.
3. Select “check” menu.
4. Verity List of the scheduled meeting with date of all room.

✦ TC_F15.5 Check schedules with error (start date is empty)

Description: Checks that the system disallow to check employee’s schedules with empty start date.

Test Inputs: Employee ID : E101
StartDate :
EndDate : 20161212

Expected Results: Return fail status and
Check schedules error message (“Check start date is empty”).

Dependencies: None

Initialization: None

Test Steps:
1. Select check schedules menu and choose “by employee” option
2. Put employee’s ID and end date with leaving start date empty.
3. Select “check” menu.
4. Verity fail messages with empty start date

✦ TC_F15.6 Check schedules with error (end date is empty)

Description: Checks that the system disallow to check employee’s schedules with empty end date.

Test Inputs: Employee ID : E101
StartDate : 20161206
EndDate :

Expected Results: Return fail status and
Check schedules error message (“Check end date is empty”).

Dependencies: None

Initialization: None

Test Steps:
1. Select check schedules menu and choose “by employee” option
2. Put employee’s ID and start date with leaving end date empty.
3. Select “check” menu.
4. Verity fail messages with empty end date

✦ TC_F15.7 Check schedules with error (end date before start date)

Description: Checks that the system disallow to check employee’s schedules with end date before start date.

Test Inputs: Employee ID : E101
StartDate : 20161206
EndDate : 20161106

Expected Results: Return fail status and
Check schedules error message (“End date precedes start date.”).

Dependencies: None

Initialization: None

Test Steps:
1. Select check schedules menu and choose “by employee” option
2. Put employee’s ID and end date before start date value
3. Select “check” menu.
4. Verity fail messages with end before start date

✦ TC_F15.8 Check schedules with error (out of range time)

Description: Checks that the system disallow to check room’s schedules with end date having 13 month value.

Test Inputs: Room ID : B203
StartDate : 20161206
EndDate : 20161306

Expected Results: Return fail status and
Check schedules error message (“Out of time range”).

Dependencies: None

Initialization: None

Test Steps:
1. Select check schedules menu and choose “by room” option
2. Put room ID and start date correctly and end date with month value “13”
3. Select “check” menu.
4. Verity fail messages with out of time range

✦ TC_F15.9 Check schedules with error (out of range time)

Description: Checks that the system disallow to check room’s schedules with start date having 32 date value.

Test Inputs: Room ID : B203
StartDate : 20161132
EndDate : 20161206

Expected Results: Return fail status and
Check schedules error message (“Out of time range”).

Dependencies: None

Initialization: None

Test Steps:
1. Select check schedules menu and choose “by room” option
2. Put room ID and start date with date value “32” and end date correctly
3. Select “check” menu.
4. Verity fail messages with out of time range

✦ TC_F16.1 Run scripts successfully

Description: Checks that the system provides functionality to execute actions in the script files.

Test Inputs: Script file with content below

Contents ## (CMD, Room ID, StartTime, EndTime, Description,

AttendeeIDlist)

- Bookmeeting, B205, 201612051200, 201612051300, New business plan meeting, E1001y^E1002^E1003^E1004

Expected Results: Every actions result logs and
Physically stored, or changed, or deleted data in the database.

Dependencies: None

Initialization: None

Test Steps:
1. Select run script menu.
2. Select target script files and put file path.
3. Select "Run" menu.
4. Verity List of the logs of each actions and physically changed in the database.

✦ TC_F16.2 Run scripts with error (empty action list)

Description: Checks that the system disallow to run the script containing empty action list.

Test Inputs: Script file with no content

Expected Results: Return fail status and
Run script error message ("empty contents script file").

Dependencies: None

Initialization: None

Test Steps:
1. Select run script menu.
2. Select target script files and put file path.
3. Select "Run" menu.
4. Verity fail messages with empty content script file.

✦ TC_F16.3 Run scripts with partial error (containing unknown action commands)

Description: Checks that the system provides the functionality of dealing with unknown action commands.

Test Inputs: Script file with content below

-Modifymeeting, B205, 201612061200, 201612061300,,,
-Bookmeeting, B207, 201612051200, 201612051300, New business plan meeting, E1001^E1002^E1003
-Insertmeeting, B304, 201612071100, 201612071200, Junior conference, E2001^E2002^E2003 (wrong command)

Expected Results: Every actions result logs (Success and Fail logs for unknown command) and physically changed data on successful actions.

Dependencies: None

Initialization: None

Test Steps:
1. Select run script menu.
2. Select target script files and put file path.
3. Select "Run" menu.
4. Verity List of the logs of each actions and physically changed in the database.

✦ TC_F16.4 Run scripts (human readability)

Description: Checks that the system provides the readability and easy to create for new employees

Test Inputs: New employee with 1 hour formatting training.

Expected Results: Ability of the employee to book a meeting and modify the meeting using the script file method.

Dependencies:	None
Initialization:	None
Test Steps:	1. Teach to new employee about the format and commands sets for 1 hour. 2. The employee write commands in a script file. 3. Select run script menu and execute the script. 4. Verity List of the logs of each actions and physically changed in the database.

✦ TC_F17.1 Copy remote database

Description:	Checks that the system provides functionality to copy remote database into local database.
Test Inputs:	Connection information (sid, ip, port) and source table name.
Expected Results:	Copying result logs and Physically synchronized local database with remote database.
Dependencies:	None
Initialization:	Copying job is configured at specific time in the job queue.
Test Steps:	1. Configure the job scheduler time to 12:00AM 2. Run job scheduler. 3. Verity copying job logs and compare the local database with remote one.