Meeting Engagement and Tracking System (MEAT)

- Software Requirements Document (Group 7) -

(Draft 1 – Sep 18, 2016)

Members : Peng Cheng, Mengjiao Zeng, Sunghun Park

1. Introduction

a. Purpose

The aim of system is to allow users in a company to book and manage meetings and vacation time and keep track of their schedules for mitigating their burden to organize and remember meetings.

b. Scope

This Requirements Document will outline the external, functional, and other requirements identified by the customer for the Meeting Engagement and Tracking System (MEAT). In addition, Expected features, Use Case diagram, and Use Cases are also presented in this document.

c. Definitions & Acronyms

i. MEAT : The Meeting Engagement and Tracking System

ii. User : The users are company employees who will use the system

iii. Meeting organizer : An employee who organize a meetings with coordinate conference rooms and assemble attendees.

iv. Attendees : Employees who are supposed to attend a meeting.

d. References

i. The first requirement document that the customer prepared.

ii. Online session of requirements elicitation with the customer in the moodle.

2. Overall Description

a. Product Perspective

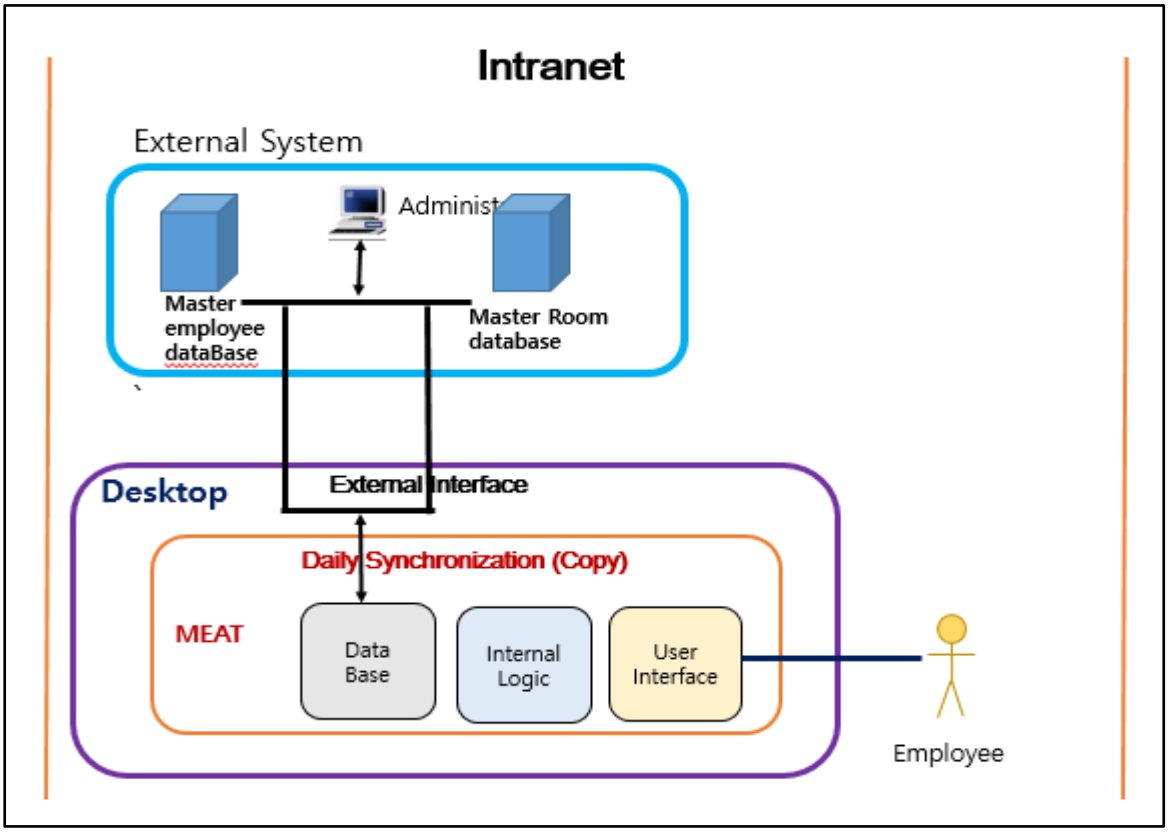
• MEAT system provides

- Complete required development functionality.

- Enable to communicate with external master employee database and room database.

- Runnable on the desktop environment of the customer's company.

[Schematic diagram]



b. Product Functions

i. The system provides users to organize and book meeting schedules.

ii. The users enable to modify and cancel the existing meeting schedules.

iii. The system allows users to look up open rooms and attendees' schedules at the given time.

iv. The users are able to change times, room, attendees' information in the existing meeting schedules.

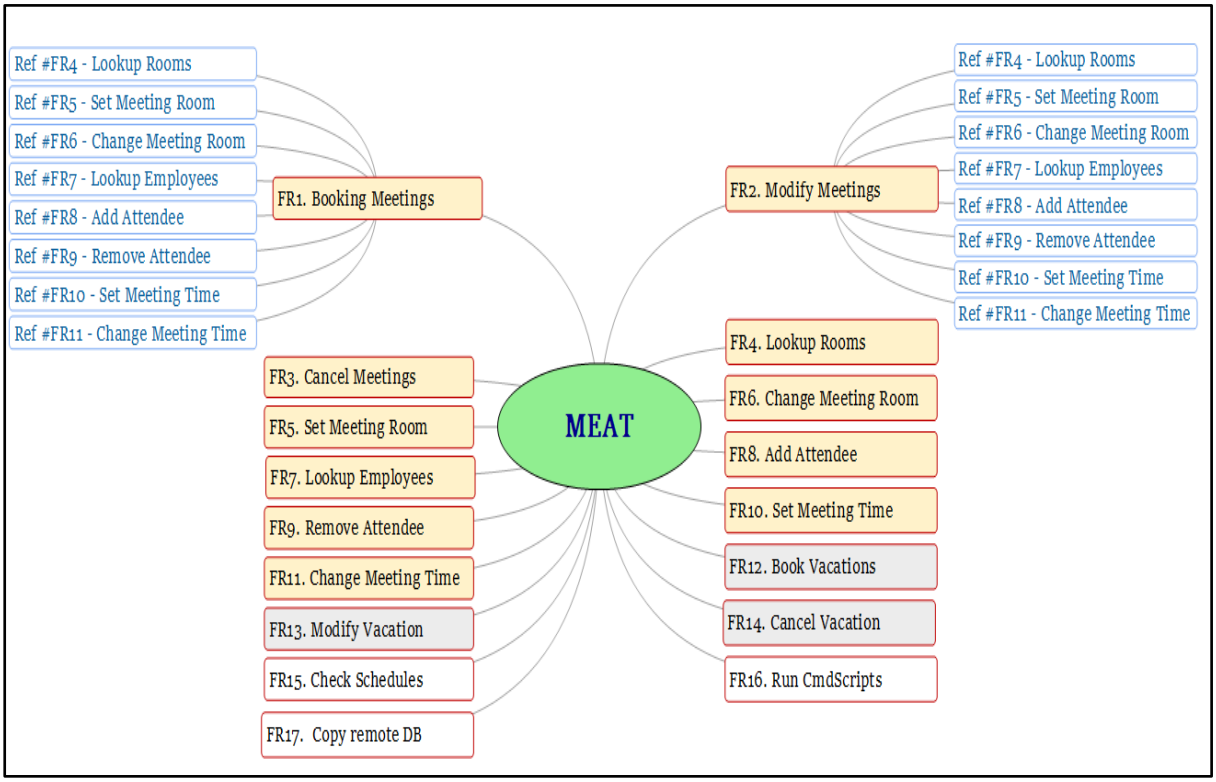
v. The system provide users to book their vacation time or change their vacation time.

vi. The users enable to check all existing schedules for specific employee, or room, or universal all rooms at given time.

vii. The system supports automation by providing the functionality of accepting the action scripts file which contains a list of action paired with necessary input.

viii. The system provides the functionality of copying and storing data on local machine daily from remote master employees and rooms databases.

[Main Features Of MEAT]



c. User characteristics

i. There is no different user role or role among the users.

ii. All users in the employee database have equal privileges. The system allows all users to look up, book, and modify all company schedules.

3. Requirements

a. External Interface Requirements

i. User interfaces

➢ Requirement ID : ER1

➢ Requirement Description : Intuitive user interface

- The system shall have an intuitive and easy to use interface.

[Specification]

1. Every user interface with feature have a brief instruction how to use it.
2. A new recruiters should be able to organize a meeting and booking a vacation within 3 hours learning.

➢ Rationale

- The customer required the intuitive interface and easiness of using the system.

➢ Test Cases

- After learning how to use this system to an inexperienced person of this system for 3 hours, and then check if the person is able to book a meeting and a vacation.

ii. Communication interfaces

➢ Requirement ID : ER2

➢ Requirement Description : Communication interface for external system

- The system shall have communication interface with user and room master database.

[Specification]

1. The system shall transfer data from external user and room database.
2. The system shall use the JDBC or API which the external system provided as a communication interface.

➢ Rationale

- The customer address daily synchronization of external database on the online session of the requirements elicitation. Developers decided to use JDBC or API.

➢ Test Cases

- TBD (to be detailed)

b. Functional Requirements (Organized by Feature)

■ Book a Meetings

➢ Requirement ID : FR1

➢ Requirement Description

- The user shall organize a meeting information and save it to the MEAT database using the MEAT system.

[Specification]

- Booking a meeting shall be allowed if,
  1. Information about room ID, date, start time, end time, description, and a list of people who will attend the meeting is ready.
  2. Target room ID is not booked by another meetings at the same time. (Req. FR5)
  3. All attendees' vacation time is not overlapped with the meeting time. (Req. FR8)
- If booking is allowed, the information on meeting shall be stored in the MEAT database.
- If booking is not allowed, schedule booking error message pops up.

➢ Rationale

- The customers want to book a meeting on new system to avoid the complexity of organizing meetings and keep track of schedules for their employees.

➢ Inputs

- room ID, date, start time, end time, description, attendees

➢ Outputs

- Booking status (Success or Fail) and physically stored data in the database.

➢ Persistent Changes

- The booking information will be stored in the database of MEAT.

➢ Related Requirements and Use Cases

- Related requirements : FR4, FR5, FR6, FR7, FR8, FR9, FR10, FR11

- Related use cases : UC1, UC7, UC8, UC9

➢ Test Cases

- TBD (To be detailed)

■ Modify a Meeting

➢ Requirement ID : FR2

➢ Requirement Description

- The user shall be able to edit and change any of the fields in the existing meeting schedule data.

[Specification]

- Changing a meeting schedule shall be allowed if,

1. More than one of the fields about room ID, date, start time, end time, description, and a list of people who will attend the meeting are changed.

2. Changed room ID is never booked by another meetings at the same time. (Req. FR5)

3. All changed attendees' vacation time is not overlapped with the meeting time. (Req. FR8)

- If changing is allowed, the changed information on meeting shall be stored in the MEAT database.

- If changing is not allowed, schedule changing error message pops up.

➢ Rationale

- The customer addressed the necessity of changing meeting schedules on any of the fields on the online session of the requirements elicitation.

➢ Inputs

- More than one of room ID, date, start time, end time, description, attendees

➢ Outputs

- Modify status (Success or Fail), and the changed data in the database.

➢ Persistent Changes

- The modified information will be stored in the database of MEAT.

➢ Related Requirements and Use Cases

- Related requirements : FR4, FR5, FR6, FR7, FR8, FR9, FR10, FR11

- Related use cases : UC2, UC7, UC8, UC9

➢ Test Cases

- TBD (To be detailed)

■ Cancel a Meetings

➢ Requirement ID : FR3

➢ Requirement Description

- The user shall be able to delete an existing meeting schedules.

[Specification]

- Deleting a meeting schedule shall be allowed if,

  1. Meeting schedule ID is present in the MEAT's database.

- If the deleting is allowed, the meeting schedule data shall be deleted in the MEAT database.

- If the deleting is not allowed, schedule deleting message pops up.

➢ Rationale

- The customer addressed the necessity of canceling meeting schedules on the online session of the requirements elicitation.

➢ Inputs

- Meeting ID

➢ Outputs

- Deleting status (Success or Fail), and physically delete data in the database.

➢ Persistent Changes

- The target meeting schedule data will be deleted in the database of MEAT.

➢ Related Requirements and Use Cases

- Related use cases : UC3

➢ Test Cases

- TBD (To be detailed)

■ Lookup Rooms

➢ Requirement ID : FR4

➢ Requirement Description

- The user shall be able to look up open rooms in the database with the given time.

[Specification]

• If user request with specific time is issued,

1. Check schedules of all rooms with the same time in the MEAT's database. (Req. FR15)

2. Show a list of the open rooms which are not booked at input meeting time.

• If not specify time, time requiring message pops up.

➢ Rationale

- The customer addressed the feature of searching open room in the initial request.

➢ Inputs

- Start Time (YYYYMMDDHH24MI), End Time (YYYYMMDDHH24MI)

➢ Outputs

- List of open rooms.

➢ Persistent Changes

- Users can select a room of the list which they want to book.

➢ Related Requirements and Use Cases

- Related requirements : FR15, FR1, FR2

- Related use cases : UC1, UC2, UC8

➢ Test Cases

- TBD (To be detailed)

■ Set a Meeting Room

➢ Requirement ID : FR5

➢ Requirement Description

- The system shall allow user to select a room of list and save the room ID into the MEAT's database.

[Specification]

• After user request with specific time is issued,

1. Show a list of the open rooms which are not booked at input meeting time. (Req. FR5)
2. Select a room ID from the list.
3. Save the room ID into the database.

➢ Rationale

- The customer addressed the feature of looking up and saving the open room for meeting schedule in the initial request.

➢ Inputs

- Meeting ID, Room ID

➢ Outputs

- Setting room status (Success or Fail), and physically save room ID data in the database.

➢ Persistent Changes

- Selected room ID will be stored into the database.

➢ Related Requirements and Use Cases

- Related requirements : FR5, FR1, FR2

- Related use cases : UC1, UC2

➢ Test Cases

- TBD (To be detailed)

■ Change a Meeting Room

➢ Requirement ID : FR6

➢ Requirement Description

- The system shall allow user to change an existing meeting room into another.

[Specification]

- If meeting ID, old room ID, and new room ID are provided,

  1. Update the old room ID of existing meeting information with new room ID.

- If any of those information is not provided,

  1. Show pop-up message about fail message of changing room information.

➢ Rationale

- The customer addressed the necessity of changing meeting schedules on any of the fields on the online session of the requirements elicitation.

➢ Inputs

- Meeting ID, old room ID, new room ID

➢ Outputs

- Changing room status (Success or Fail), and physically update room ID into new one in the database.

➢ Persistent Changes

- New room ID will be updated into new one physically.

➢ Related Requirements and Use Cases

- Related requirements : FR1, FR2

- Related use cases : UC2

➢ Test Cases

- TBD (To be detailed)

■ Lookup Employees

➢ Requirement ID : FR7

➢ Requirement Description

- The user shall be able to look up employees not having overlapped vacation time with the given meeting time.

[Specification]

- If user request with specific time is issued,
  1. Check schedules of all people with the same vacation time in the MEAT's database. (Req. FR15)
  2. Show a list of the employees who don't overlapped with their vacation time.
- If user request with specific time and additional name keyword is issued,
  1. Check schedules of name-matched people with the same vacation time in the MEAT's database. (Req. FR15)
  2. Show a list of the employees who don't overlapped with their vacation time.
- If not specify time, time requiring message pops up.

➢ Rationale

- The customer addressed the feature of searching attendees in the initial request. Developers decided to offer additional functionality to look up employees by name conveniently.

➢ Inputs

- Start Time, End Time (YYYYMMDDHH24MI), employee's name (optional)

➢ Outputs

- List of employees who are able to attend the meeting time.

➢ Persistent Changes

- Users can select a person of the list which they want to add into attendees list.

➢ Related Requirements and Use Cases

- Related requirements : FR15, FR1, FR2

- Related use cases : UC1, UC2, UC7

➢ Test Cases

- TBD (To be detailed)

■ Add an Attendee

➢ Requirement ID : FR8

➢ Requirement Description : Add an Attendee

- The user shall be able to select attendees and save a list of attendees into the MEAT's database.

[Specification]

- After user request with specific time is issued,

  1. Show a list of the employees (not overlapped time). (Req. FR7)
  2. Select an employee ID from the list.
  3. Save the employee ID into the database.

➢ Rationale

- The customer addressed the feature of looking up and saving the attendee for meeting schedule in the initial request document.

➢ Inputs

- Meeting ID, employee ID

➢ Outputs

- Adding attendee status (Success or Fail), and physically save attendee ID data in the database.

➢ Persistent Changes

- The employee ID will be stored into the database as an attendee.

➢ Related Requirements and Use Cases

- Related requirements : FR7, FR1, FR2

- Related use cases : UC1, UC2

➢ Test Cases

- TBD (To be detailed)

■ Remove Attendees

➢ Requirement ID : FR9

➢ Requirement Description

- The user shall be able to delete attendees from the attendee list of a meeting.

[Specification]

- After user request with specific time and meeting ID is provided,
  1. Show a list of the attendees from the database.
  2. Select an attendee ID from the list.
  3. Delete the attendee ID from the database.
- If not provided, deleting attendee error message pop up.

➢ Rationale

- The customer addressed the necessity of changing meeting attendees from the online session of the requirements elicitation.

➢ Inputs

- Meeting ID, attendee ID

➢ Outputs

- Deleting attendee status (Success or Fail), and physically delete attendee ID data in the database.

➢ Persistent Changes

- The employee ID will be deleted from the database.

➢ Related Requirements and Use Cases

- Related requirements : FR1, FR2

- Related use cases : UC2

➢ Test Cases

- TBD (To be detailed)

■ Set a Meeting Time

➢ Requirement ID : FR10

➢ Requirement Description

- The system shall allow user to input meeting time and save it into the MEAT's database.

[Specification]

- Setting a time shall be allowed if,
  1. Meeting start time and end time should be provided, and
  2. Their time format is YYYYMMDDHH24MI
  3. Save the meeting time into the database.
- If the time inputs are not provided or not an exact time format,
  1. Meeting Time error message pop up.

➢ Rationale

- The customer addressed the feature of storing meeting time in the initial request.

➢ Inputs

- Meeting ID, Meeting start time, end time (YYMMDDHH24MI)

➢ Outputs

- Setting meeting time status (Success or Fail), and physically save the meeting time data in the database.

➢ Persistent Changes

- The meeting start time and end time will be stored into the database.

➢ Related Requirements and Use Cases

- Related requirements : FR1, FR2

- Related use cases : UC1, UC2

➢ Test Cases

- TBD (To be detailed)

■ Change a Meeting Time

➢ Requirement ID : FR11

➢ Requirement Description

- The system shall allow user to change existing meeting time into new time.

[Specification]

- Changing a time shall be allowed if,

  1. New meeting start time and end time should be provided, and

  2. Their time format is YYYYMMDDHH24MI (Req. FR10)

  3. Update existing meeting time into the new meeting time in the database.

- If the new time inputs are not provided or not an exact time format,

  1. Meeting Time error message pop up.

➢ Rationale

- The customer addressed the necessity of changing meeting time on the online session of the requirements elicitation.

➢ Inputs

- Meeting ID, New meeting start time, end time (YYMMDDHH24MI)

➢ Outputs

- Changing meeting time status (Success or Fail), and physically save the meeting time data in the database.

➢ Persistent Changes

- New meeting start time and end time will change old ones of an existing meeting schedule information in the database.

➢ Related Requirements and Use Cases

- Related requirements : FR10, FR1, FR2

- Related use cases : UC2

➢ Test Cases

- TBD (To be detailed)

■ Book a Vacation

➢ Requirement ID : FR12

➢ Requirement Description

- The users shall be able to book their vacation dates into the MEAT's database.

[Specification]

- Booking a vacation shall be allowed if,

  1. Vacation start date and end date should be provided, and

  2. Their time format is YYYYMMDD.

  3. Save the dates into the database.

- If the new dates inputs are not provided or not an exact time format,

  1. Booking vacation time error message pop up.

- If vacation dates are overlapped with existing meeting time as an attendee, the user should remove his or her name from existing meeting schedules in advance. (Req. FR9, FReq.15)

➢ Rationale

- The customer addressed the feature of booking vacation time in the initial request document. In addition, developers decided that users should remove his or her in the attendee list when vacation and existing meeting time conflicts.

➢ Inputs

- Employee ID, Vacation start date, end date (YYMMDD)

➢ Outputs

- Booking vacation status (Success or Fail) and physically save the vacation time data in the database.

➢ Persistent Changes

- The employee ID and vacation time will be stored into the database.

➢ Related Requirements and Use Cases

- Related requirements : FR9, FR15

- Related use cases : UC4

➢ Test Cases : TBD (To be detailed)

■ Modify a Vacation

➢ Requirement ID : FR13

➢ Requirement Description

- The users shall be able to change their existing vacation dates in the MEAT's database.

[Specification]

- Changing a new vacation time shall be allowed if,

  1. New vacation start date and end date should be provided, and

  2. Their time format is YYYYMMDD.

  3. Change the old existing vacation dates into new dates.

- If the new dates inputs are not provided or not an exact time format,

  1. Changing vacation time error message pop up.

- If new vacation dates are overlapped with existing meeting time as an attendee, the user should remove his or her name from existing meeting schedules in advance. (Req. FR9, Req.FR15)

➢ Rationale

- The customer addressed the feature of booking vacation time in the initial request document. In addition, developers decided that users should remove his or her in the attendee list when vacation and existing meeting time conflicts.

➢ Inputs

- Employee ID, vacation ID, new vacation start date, end date (YYMMDD)

➢ Outputs

- Changing vacation status (Success or Fail), and physically swap the vacation time data in the database.

➢ Persistent Changes

- The new vacation time of existing vacation will be stored into the database.

➢ Related Requirements and Use Cases

- Related requirements : FR9, FR15

- Related use cases : UC5

➢ Test Cases : TBD (To be detailed)

■ Cancel a Vacation

➢ Requirement ID : FR14

➢ Requirement Description

- The users shall be able to cancel their existing vacation dates in the MEAT's database.

[Specification]

- Cancel an existing vacation time shall be allowed if,

  1. Vacation ID is provided

  2. Delete the vacation date in the database.

- If not accepted,

  1. Deleting vacation error message pop up.

➢ Rationale

- Developers decided to offer the functionality of cancel the booked vacation considering the case of vacation's cancellation.

➢ Inputs

- Vacation ID

➢ Outputs

- Deleting vacation status (Success or Fail), and physically delete the vacation data in the database.

➢ Persistent Changes

- The existing vacation date will be deleted from the database.

➢ Related Requirements and Use Cases

- Related use cases : UC6

➢ Test Cases

- TBD (To be detailed)

■ Check Schedules

➢ Requirement ID : FR15

➢ Requirement Description

- The system shall allow users to look up the existing schedules of an employee, or a room, or universal schedules at given period in the MEAT's database.

[Specification]

• There should be three sub-functionalities of this feature.

1. Check schedule of an employee

- An employee ID or Employ name with specific period should be provided

- If the checking is allowed, it should show the list of the employee's meeting, vacation, and non-vacation date information.

2. Check schedule of a room

- A room ID with specific period should be provided

- If the checking is allowed, it should show the list of all scheduled meetings of the room at given period.

3. Check universal schedule

- Specific period should be provided

- If the checking is allowed, it should show the list of all scheduled meetings of all room at given period.

• If not accepted,

1. Check schedule error message pop up.

➢ Rationale

- The customer addressed the feature of checking schedules in the initial request document.

➢ Inputs

- For person : An employee ID or Employee name, search period

- For room : A room ID, search period

- For universal : search period

➢ Outputs

- For person : a list of the employee's meeting, vacation, non-vacation date

- For room : a list of all scheduled meetings of the room

- For universal : a list of all scheduled meetings of all room

➢ Persistent Changes

- The list is showed up on the user interface.

➢ Related Requirements and Use Cases

- Related use cases : UC7, UC8, UC9

➢ Test Cases

- TBD (To be detailed)

■ Run Scripts

➢ Requirement ID : FR16

➢ Requirement Description

- The system shall also be able to accept “scripts” through the command line. A script shall be a list of actions, each action paired with necessary input.

[Specification]

• It should be a file that includes a list of action commands having proper inputs.

- It should allow us to perform same action that we can do in normal interface.

• The action commands should be both readable human and machine.

• If the scripts contains incorrect commands,

- After scripts are completed, then display error list for uncompleted commands.

➢ Rationale

- The customer required the feature of running with commands script for automation tasks in the initial request document.

➢ Inputs

- A file contains commands

➢ Outputs

- Error list for uncompleted commands, each action matched each command

➢ Persistent Changes

- Each action described in the script file performed.

➢ Related Requirements and Use Cases

- Related use cases : UC10

➢ Test Cases

- TBD (To be detailed)

■ Copy Remote Database

➢ Requirement ID : FR17

➢ Requirement Description
- The system shall be able to copy data from external master employees and rooms database and store on local machine.

[Specification]

1. All changes of external employees and rooms database should be updated to the local MEAT database in daily basis.
2. To support this functionality, an automated process should exist.
3. If not exist,
   - Synchronization error message should be present to user on launch .

➢ Rationale
- The customer required the feature of copying and storing remote database into local machine in daily basis on online session of requirements elicitation.

➢ Inputs
- None

➢ Outputs
- Synchronization status (Success or Fail)

➢ Persistent Changes
- All of the changes in remote database are updated on local MEAT database .

➢ Related Requirements and Use Cases
- Related use cases : UC11

➢ Test Cases
- TBD (To be detailed)

c. Other Requirements

➢ Requirement ID : OR1

➢ Requirement Description : Executable on desktop machine

- The system shall be executed on desktop machine regardless of operating system.

➢ Rationale

- The customer addressed runnable environment for desktop machine on the online session of the requirements elicitation

➢ Requirement ID : OR2

➢ Requirement Description : Use Java language for development

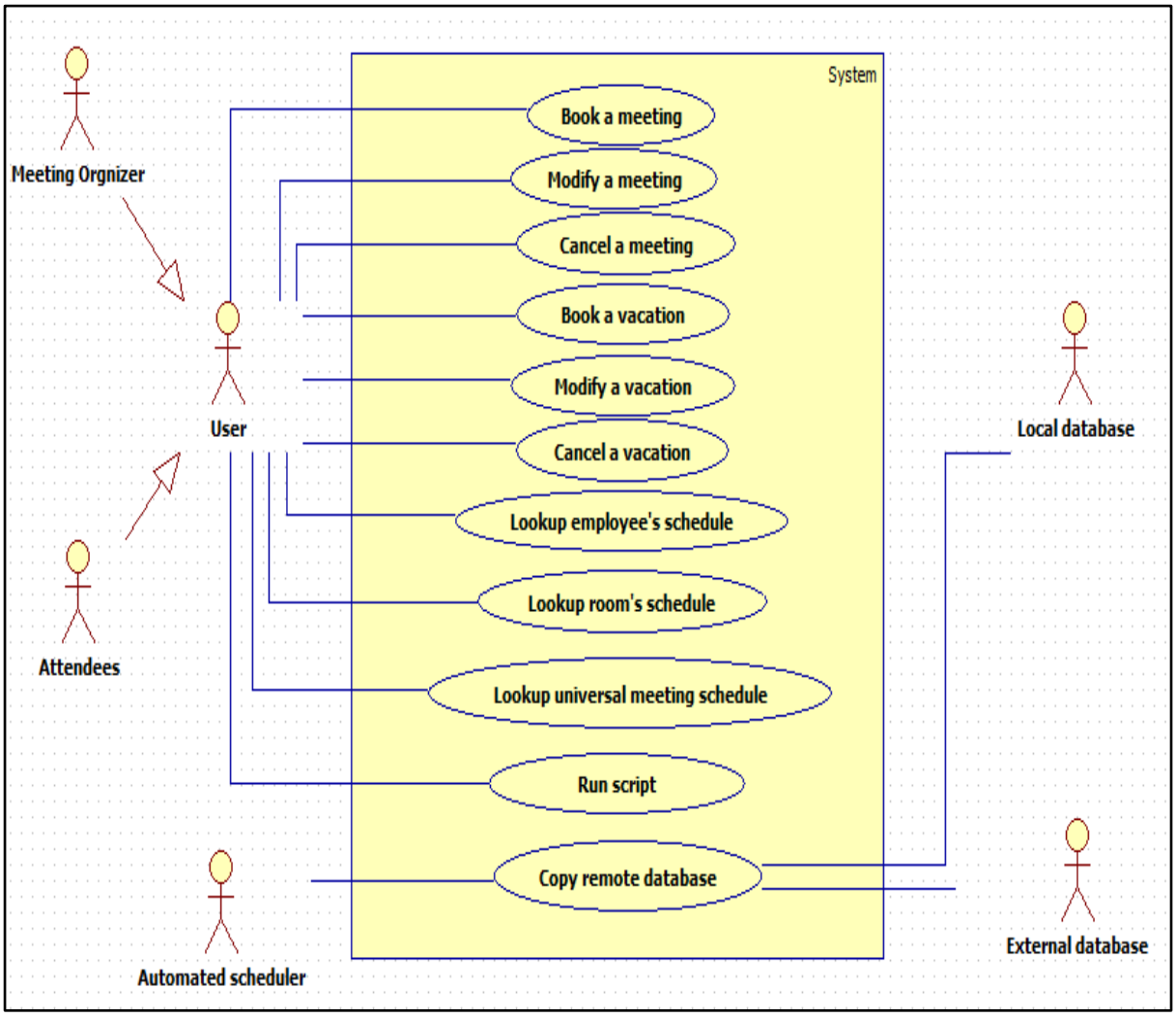
- The system shall be developed by the Java language.

➢ Rationale

- The customer wants the system to be developed by Java language on the online session of the requirements elicitation

4. Use Cases

a. Use Case Diagram



b. Use Cases

- **Use Case ID :** UC1

- **Use Case Name:** Book a Meeting

- **Iteration:** Filled.

- **Summary:** The user shall organize a meeting information and save it to the MEAT database using the MEAT system.

- **Basic Course of Events:**

1. The user enter into MEAT system.
2. The user enter into the user interface.
3. The MEAT system asks the user to enter date, start time, end time, description of the meeting.
4. The MEAT system validates the input from command line.
5. MEAT show the list of available room which is ready for selecting.
6. User choose one available room.
7. MEAT show the list of available attendee that is ready for selecting.
8. User choose all of the attendees for this meeting.
9. The MEAT ask the user to confirm the input date, meeting time, description of meeting, selecting room and attendees.
10. The user confirms the meeting information.
11. The MEAT system complete setting up the meeting.
12. MEAT system save the meeting information into the database.
13. A message about successfully setting up the meeting is popout.
14. The status of selected room in selected meeting time is changed to unavailable.
15. The status of selected attendees in this time is changed to unavailable.
16. MEAT pop out a "Thank you for using, have a great day" message.

- **Alternative Paths:** In step 10, if the user is can not agree with the meeting information, the user can cancel the setting up the meeting and go directly to step 18.

- **Exception path:** In step 4, if the input has errors, the MEAT system will refuse to continue the process and pop out an error message and indicate the error part, and the system will proceed to step 16. In step 4, if no available room in selecting time, the MEAT system will refuse to continue the process and pop out error message and indicate the error part, and the system will proceed to step 16.

- **Extension point:** None.

- **Trigger:** The user wants to book a meeting with MEAT system.

- **Assumptions:** The user has authority to enter into the MEAT system, and willing to set up a meeting that all of attendees are inside this MEAT system.

- **Precondition**: The MEAT system have already save room data and employee data in the database.

- **Postcondition:** The status of selected room and attendees will be changed to unavailable in certain meeting time range.

- **Author:** Sunghun Park; Peng Cheng; Mengjiao Zeng

- **Date:**09/16/2016

- **Use Case ID :** UC2

- **Use Case Name:** Modify a meeting

- **Iteration:** Filled.

- **Summary:** The user shall be able to edit and change any of fields in the existing meeting schedule data.

- **Basic Course of Events:**

1. The user enter into the MEAT system.
2. The user enter into the user interface, and select existing meeting schedule page.
3. A list of current meeting schedule is showed.
4. User choose one of the existing schedule.
5. This meeting's information (meeting date, meeting start time, meeting end time, meeting room, and meeting attendees list) is showed to the user.
6. User choose to change one part of the meeting information about the current meeting.
7. If user choose to change meeting room, a list of available room in that same time will be showed. User choose a new meeting room by room ID.
8. If user choose to change attendees for the meeting, the user will choose between add attendees or remove attendees.
9. If user choose to add attendees, a list of users in the database will be showed to the user.
10. User choose the adding attendees by name.
11. If user choose to remove attendees, a list of meeting attendees for that meeting will be showed to the user.
12. User choose to remove attendees by name.
13. The MEAT ask the user to confirm the meeting date, meeting time, description of meeting, selecting room and attendees.
14. The user confirms the meeting information.
15. The MEAT system complete modifying the meeting and save information to database.
16. A message about successfully setting up the meeting is popout.
17. The status of selected room in selected meeting time is changed to unavailable.
18. The status of selected attendees in this time is changed to unavailable.
19. MEAT pop out a "Thank you for using, have a great day" message.

- **Alternative Paths**: In step 6, if user choose to change meeting date, meeting start time or meeting end time, MEAT will check the availability of the old meeting room. If it is open in the new selecting time, the time modification will be allowed, and the system will proceed to In step 4,5,6,7,8,9,10,11,12,13, the user can cancel the modification of the current schedule, the modification page is closed and go directly to step 19. If user does not confirm the modifying information in step 14, proceed directly to step 19.

- **Exception path:** In step 6, If the old meeting room is not open for the new selecting time, a list of available room in that time will be showed. User choose a new room by room ID.If some attendees are not available in the new meeting time, a message about the list of attendees who will not be able to come to the new meeting time will pop out,and the system will proceed to step 16.

- **Extension point:** None.

- **Trigger:** The user wants to modify an existing meeting in MEAT system.

- **Assumptions:** The user has authority to enter into the MEAT system, and willing to modify a meeting which is already settled in MEAT system.

- **Precondition**: The MEAT system have already save room data and employee data in the database. An old meeting information is stored in the database.

- **Postcondition:** The status of selected room and attendees will be changed to unavailable new meeting time range. The status of old meeting time and room will be changed to be available.

- **Author:** Sunghun Park; Peng Cheng; Mengjiao Zeng

- **Date:**09/16/2016

- **Use Case ID :** UC3

- **Use Case Name:** Cancel a Meeting

- **Iteration:** Filled.

- **Summary:** The user shall be able to delete an existing meeting schedule in MEAT database.

- **Basic Course of Events:**

1. The user enter into the MEAT system.
2. The user enter into the user interface, and select existing meeting schedule.
3. The user choose to cancel one meeting by meeting ID.
4. The MEAT ask the user to confirm the meeting cancellation information.
5. The user confirms to cancel the meeting.
6. The MEAT system complete user's modifying meeting process and save information to database.
7. A message about successfully cancel a meeting popout.
8. The status of selected meeting room and attendees in selected meeting time is changed to available.
9. MEAT pop out a "Thank you for using, have a great day" message.

- **Alternative Paths**: In step 2,3,4, the user can cancel the modify meeting process, the cancel meeting page is closed and go directly to step 9. If user does not confirm the cancel meeting information in step 5, proceed directly to step 9.

- **Exception path:** If user does not confirm the cancel meeting information in step 5, proceed directly to step 9. If user is not allowed to cancel the selected meeting in step 4, proceed directly to step 9.

- **Extension point:** None.

- **Trigger:** The user wants to cancel an existing meeting schedule in MEAT system.

- **Assumptions:** The user has authority to enter into the MEAT system.

- **Precondition**: MEAT system stored a list of meeting schedule, and the user is willing to cancel some meetings.

- **Postcondition:** The status of selected meeting room and attendees in that meeting schedule will be changed to available.

- **Author:** Sunghun Park; Peng Cheng; Mengjiao Zeng

- **Date:**09/16/2016

- **Use Case ID :** UC4

- **Use Case Name:** Book a Vacation

- **Iteration:** Filled.

- **Summary:** The user shall be able to book their vacation dates into the MEAT database.

- **Basic Course of Events:**

1. The user enter into the MEAT system.
2. The user enter into the user interface, and select personal schedule page.
3. The user choose to book a vacation.
4. Two calendar pop out. First calendar ask user to choose vacation start date, and the second calendar ask user to choose vacation end date.
5. The MEAT ask the user to confirm the vacation start date and end date.
6. The user confirms the vacation information.
7. The MEAT system complete user's booking vacation process and save information to database.
8. A message about successfully book a vacation is popout.
9. The status of selected user in selected vacation time is changed to unavailable.
10. MEAT pop out a "Thank you for using, have a great day" message.

- **Alternative Paths**: In step 3,4,5, the user can cancel the book vacation process, the book vacation page is closed and go directly to step 10. If user does not confirm the vacation information in step 6, proceed directly to step 10.

- **Exception path:** In step 4, if any existing meeting is time conflict with the vacation time, the book vacation process will stop. A message about the error will pop out, and the system will proceeds to step 10.

- **Trigger:** The user wants to book some weekdays off for his personal vacation in MEAT system.

- **Assumptions:** The user has authority to enter into the MEAT system.

- **Precondition**: MEAT system stored this user's personal meeting schedule.

- **Postcondition:** The status of selected user under vacation time will be changed to unavailable.

- **Author:** Sunghun Park; Peng Cheng; Mengjiao Zeng

- **Date:**09/16/2016

- **Use Case ID :** UC5

- **Use Case Name:** Modify a vacation

- **Iteration:** Filled.

- **Summary:** The user shall be able to change their existing vacation information in MEAT database.

- **Basic Course of Events:**

1. The user enter into the MEAT system.
2. The user enter into the user interface, and select personal schedule page.
3. The user choose to modify a vacation.
4. Two calendar pop out. First calendar ask user to choose vacation start date, and the second calendar ask user to choose vacation end date.
5. The MEAT ask the user to confirm the vacation start date and end date.
6. The user confirms the vacation information.
7. The MEAT system complete user's modifying vacation process and save information to database.
8. A message about successfully book a vacation is popout.
9. The status of selected user in selected vacation time is changed to unavailable.
10. The status of old vacation time is changed to available.
11. MEAT pop out a "Thank you for using, have a great day" message.

- **Alternative Paths**: In step 3,4, the user can cancel the modify vacation process, the modify vacation page is closed and go directly to step 11. If user does not confirm the new vacation information in step 7, proceed directly to step 11.

- **Exception path:** In step 4, if any existing meeting is time conflict with the new vacation time, the book vacation process will stop. A message about the error will pop out.

- **Trigger:** The user wants to modify his or her personal vacation schedule in MEAT system.

- **Assumptions:** The user has authority to enter into the MEAT system.

- **Precondition**: MEAT system stored this user's personal meeting schedule.

- **Postcondition:** The status of selected user under new vacation time will be changed to unavailable. The status of selected user under old vacation time will be changed to available.

- **Author:** Sunghun Park; Peng Cheng; Mengjiao Zeng

- **Date:**09/16/2016

- **Use Case ID :** UC6

- **Use Case Name:** Cancel a Vacation

- **Iteration:** Filled.

- **Summary:** The user shall be able to delete an existing vacation schedule in MEAT database.

- **Basic Course of Events:**

1. The user enter into the MEAT system.
2. The user enter into the user interface, and select personal schedule page.
3. The user select one vacation and choose to cancel that vacation.
4. The MEAT ask the user to confirm the cancel vacation information.
5. The user confirms the deleting vacation information.
6. The MEAT system complete user's modifying vacation process and save information to database.
7. A message about successfully cancel a vacation pops out.
8. The status of selected user in selected vacation time is changed to available.
9. MEAT pops out a "Thank you for using, have a great day" message.

- **Alternative Paths**: In step 2,3, user can stop cancel vacation process and the system will directly proceed to step 9.

- **Exception path:** None.

- **Extension point:** None.

- **Trigger:** The user wants to cancel an existing vacation schedule in MEAT system.

- **Assumptions:** The user has authority to enter into the MEAT system.

- **Precondition**: MEAT system stored employee's personal vacation schedule, and the user is willing to cancel some vacations.

- **Postcondition:** The status of selected employee in that vacation time will be changed to available.

- **Author:** Sunghun Park; Peng Cheng; Mengjiao Zeng

- **Date:**09/16/2016

- **Use Case ID :** UC7

➢ **Use Case Name:** Lookup Employee's Meeting Schedule

➢ **Iteration:** Filled.

➢ **Summary:** The user shall be able to lookup his or her own meeting schedule and other employees' meeting schedule in MEAT database.

➢ **Basic Course of Events:**

1. The user enter into the MEAT system.
2. The user enter into the user interface, and select lookup employee's meeting schedule.
3. A list of employees is showed.
4. User select employee by employee name.
5. Selected employee's meeting schedule is showed.

➢ **Alternative Paths**: After step 5, if the user wants to see another employee's meeting schedule, the system will proceed back to step 4.

➢ **Exception path:** None.

➢ **Extension point:** None.

➢ **Trigger:** The user wants to lookup employee's existing meeting schedule in MEAT system.

➢ **Assumptions:** The user has authority to enter into the MEAT system.

➢ **Precondition**: MEAT system stored a list of employees' meeting schedule.

➢ **Postcondition:** None.

➢ **Author:** Sunghun Park; Peng Cheng; Mengjiao Zeng

➢ **Date:**09/16/2016

■ **Use Case ID :** UC8

➢ **Use Case Name:** Lookup Room Meeting Schedule

➢ **Iteration:** Filled.

➢ **Summary:** The user shall be able to lookup room's meeting schedule in MEAT database.

➢ **Basic Course of Events:**

1. The user enter into the MEAT system.
2. The user enter into the user interface, and select look up room meeting schedule.
3. A list of meeting room is showed.
4. User select meeting room by room ID.
5. Selected room's meeting schedule is showed.

➢ **Alternative Paths**: After step 5, if the user wants to see another room's meeting schedule, the system will proceed back to step 4.

➢ **Exception path:** None.

➢ **Extension point:** None.

➢ **Trigger:** The user wants to cancel an existing meeting schedule in MEAT system.

➢ **Assumptions:** The user has authority to enter into the MEAT system.

➢ **Precondition**: MEAT system stored a list of meeting schedule, and the user is willing to cancel some meetings.

➢ **Postcondition:** The status of selected meeting room and attendees in that meeting schedule will be changed to available.

➢ **Author:** Sunghun Park; Peng Cheng; Mengjiao Zeng

➢ **Date:**09/16/2016

- **Use Case ID :** UC9

- **Use Case Name:** Lookup Universal Meeting Schedule

- **Iteration:** Filled.

- **Summary:** The user shall be able to lookup all of meeting schedule in selected time period in MEAT database.

- **Basic Course of Events:**

1.The user enter into the MEAT system.

2.The user enter into the user interface, and select look up universal meeting schedule.

3.The user choose a time period start in date and start time, end in date and end time.

4.Meeting schedule in selected time period is showed.

- **Alternative Paths**: After step 4, if the user wants to see another time period's meeting schedule, the system will proceed back to step 3.

- **Exception path:** None.

- **Extension point:** None.

- **Trigger:** The user wants to lookup all meetings' schedule in selected time period in MEAT system.

- **Assumptions:** The user has authority to enter into the MEAT system.

- **Precondition**: MEAT system stored a list of meeting schedule, and the user is willing to lookup some meetings' schedule by choose time period.

- **Postcondition:** None.

- **Author:** Sunghun Park; Peng Cheng; Mengjiao Zeng

- **Date:**09/16/2016

- **Use Case ID :** UC10

- **Use Case Name:** Run Script

- **Iteration:** Filled.

- **Summary:** The user shall be able to use scripts to complete a task. A script contains a list of commands, each command includes necessary inputs.

- **Basic Course of Events:**

1. The user enter into the MEAT system.
2. The user enter into the user interface, and select script page.
3. The user enter a list of command(String) into the command box.
4. The user click “run” button.
5. The script will be running automatically.
6. The MEAT system complete “script running” process and save information to database.
7. A message about successful actions and other warning information pops out.
8. Finish your job, go back to the main menu.

- **Alternative Paths**: In step 3, user can stop use “script” function, go back to main menu.

- **Exception path:** in step 7, if the system sends out error message, you can enter a new command or go back to the main menu..

- **Extension point:** None.

- **Trigger:** The user wants to use script to accomplish a list of actions.

- **Assumptions:** The user has authority to enter into the MEAT system.

- **Precondition**: MEAT system can support “script” feature.

- **Postcondition:** According to the result of “script”, the database must be update the information.

- **Author:** Sunghun Park; Peng Cheng; Mengjiao Zeng

- **Date:**09/16/2016

■ **Use Case ID :** UC11

➢ **Use Case Name:** Copy Remote Database

➢ **Iteration:** Filled.

➢ **Summary:** The system shall be able to copy from remote employees and rooms database to local MEAT database. An automated scheduling process performs the synchronization on daily basis.

➢ **Basic Course of Events:**

1. The scheduled process is executed at the designated time a day automatically.
2. The process connects the remote database system.
3. The process copy into local database from all the data in master employees and rooms database.
4. The process writes the result status into log file(db).
5. The process stops running and waits for next day execution.

➢ **Alternative Paths**: None

➢ **Exception path:** None.

➢ **Extension point:** None.

➢ **Trigger:** Scheduled time.

➢ **Assumptions:** None.

➢ **Precondition**: The remote database is running within connectable network.

➢ **Postcondition:** None.

➢ **Author:** Sunghun Park; Peng Cheng; Mengjiao Zeng

➢ **Date:**09/17/2016