FE2: Systemtestplan

- **Test Plan**: **a management planning document** that shows:
  - How the testing will be done (including SUT (system under test) configurations).
  - Who will do it
  - What will be tested
  - How long it will take (although this may vary, depending upon resource availability).
  - What the test coverage will be, i.e. what quality level is required

Login - TestId: LoginT

Precondition:

A connection to the server is already established.

Dependencies:

Working TCP connection, client side and server side. Working database is recommended, but not required to perform the test.

Objective:

Authenticate users on the system. The system should take user name and password as input, and produce appropriate feedback, whether the authentication is successful or not. Some additional feedback is handled locally in the client, like missing input and/or invalid input, such errors are handled locally to not bother the system.

Expected results

If the login is successfully, the system should respond by changing to the users personal view. For testing purposes, the system should also give some feedback in form of a message, in order for the developers to test the part independently.

Add Appointment - TestId: AddApp

Precondition:

An authenticated connection to the server is established.

Dependencies:

LoginT,

Objective:

Add appointments to the system. An appointment contains date, start and end time, description, and place. The system should gather this input from the user. Some test should test this data locally, before adding the appointment to the calender system. These test may contain

validation tests on the input; invalid input, missing data, etc. If such error occur the user should be informed before the data is sent.

Expected results:

If the operation is successful, the system should give feedback inform of changes in the view, and additional messages informing, the user of success. The latter messages are included to make the testing independently, e.g do not require GUI.

If the operation yields unsuccessful, the system gives feedback in form of a message telling the user that the action failed. The message should include why the operation failed. Possible failures, the appointment already exists, the room is already booked, and so on.

Remove Appointment - TestId: RmvApp

Precondition:

An connection is already established. The user is presented with the correct view.

Dependencies:

LoginT, AddApp

Objective:

Remove a registered appointment from the user's personal calender. Note: The appointment as no participants.

Expected results:

The information of the appointment to be deleted, is sent to the server. The information is required by the server in order to delete appropriate appointment. The server respond in form of a message that yield; successful or unsuccessful.

If the operation is successful, the local system perform the appropriate updates to the view.

If the operation is unsuccessful, the message should indicate why, and the system must apply the appropriate steps to recover.

Edit Appointment - TestID: EdApp

Precondition:

An connection is already established. The user is presented with the correct view.

Dependencies:

LoginT, AddApp

Objective:

Edit information in an already existing appointment in the users personal calendar. The local system should send the updated information to the server, some validation is performed on the data to make sure the information is correct before it's sent. The server should respond to the update request successful, or unsuccessful.

Expected results:

If the server respond to the operation successful, the local system should respond by updating the view, and report that the changes are made.

If the server respond to the operation unsuccessful, the local system should give the appropriate feedback, and recover. The feedback should include why the operation failed.

Call a meeting - TestId: CallM

Precondition:

An connection is already established.

Dependencies:

LoginT

Objective:

Create a new meeting, setting the person calling the meeting as the meeting leader. The meeting leader should be able to invite other users from the company. The data is encapsulated in an appropriate manner, forming a request to the server. The data should be validated locally before it's sent. This may involving checking if the participants as well. The server should request the operation, yielding successful or unsuccessful.

Expected results:

If the operation is successful, the view should be updated accordingly, and status information about the operation should be printed.

If the operation is unsuccessful, a message indicating the error should be displayed, and the system must recover. This may involve interaction by the user.

Receive notification of meeting - TestId: RcvNoM

Precondition:

An connection is already established. The user is in the same company as the meeting leader.

Dependencies:

LoginT, CallM

Objective:

User shall receive the notification, and then be prompted to accept or refuse it. Should he accept it, the meeting should be added as an appointment to his or her personal calendar. If he refuses it, it should not be added, and a message is sent to the meeting leader that the meeting was not accepted by this user. The meeting Leader should then be prompted if he wants to set a new meeting date, cancel the meeting, or remove the user from the meeting list.

Expected results:

The participants should receive a notification object from the server. The object contains all relevant information about the meeting. On client side, the user should be prompted with the notification. The user can either accept or decline.

If the user accept, the view should reflect the changes by adding the meeting to his personal calender.

If the user declines, a message notification object should be sent to the leader. The server is responsible for receiving and re-sending the notification.

If the operation is unsuccessful, the operation should prompt the user where the error occurred, and recover. For easier debugging, the sever should in any case generate a report of the operations carried out.

Change notification of meeting - TestId: ChngNoM

Precondition:

An connection is already established. The user must be the leader of the meeting in question. The user is presented with the correct view.

Dependencies:

LoginT, CallM

Objective:

The meeting leader should be able to edit the date of a meeting. There should then be sent out message to everyone invited that states the new date, and the users can accept or refuse the new date. Should a user refuse the new date, a message should be sent to all other meeting participants, and the meeting leader should be able to either set another date or cancel the meeting.

Expected results:

The leader of a meeting should have the ability to change the date of a meeting through the

view from his calendar. Once the date is changed and committed, the server should send out a notification to all users participating in the meeting. The users should then be prompted to either accept the new date, or reject it. Should a user reject the new date, the leader should get a notification back, and should get an notification to either cancel the meeting or set a new date.

Cancel meeting - TestiD: CMeet

Precondition:

An connection is already established. The user is the leader of the meeting in question. The user is presented with the correct view.

Dependencies:

LoginT, CallM

Objective:

The meeting leader should be able to cancel a meeting, there should then be sent a message to everyone involved, and the meeting should be cancelled in their personal calendars. Information about the meeting, including its participants, is encapsulated and sent to the server. Some local validation of the data may be carried out before its sent. The server is required to detect and perform the operation, this include updating the database, and send a “cancel” notification to all participants in the list. When completed, or if an error occurs, the server should either way produce a report containing the operations performed.

Expected results:

If the request is carried out out successful, the view will be updated accordingly in all the clients. For debugging purposes, the result should also contain a report of the actions performed by the system. This report should only be sent to the client which performed the operation

Cancel appointment to meeting - TestId: CApptMeet

Precondition:

An connection is already established. The user is presented with the correct view. The user must have received a notification to a meeting.

Dependencies:

LoginT, CallM, RcVNoM

Objective:

A user should be able to cancel his appointment to a meeting by deleting the meeting in his or her personal calendar. A message should then be sent to all other participants to the meeting, and the meeting leader should be given a choice to change the meeting date, or cancel the meeting. The information about the appointment, including information about the participants, is encapsulated in an appropriate manner, and sent to the server.

The server is responsible for sending out notifications to the other participants, and generate a proper message in the form of a report, indicating success or unsuccessful. The report should be generated when the server have the proper responses from the clients. The server may also keep track of “pending” messages so the users are aware of any changes made, during offline time.

Expected results:

If the operation is carried out successful, the view should be updated accordingly. Either way a report is generated. The report is used for debugging.

Reserve a meeting room - TestId: RsrvMRoom

Precondition:

An connection is already established.The user must have called a meeting or have set up an appointment. The user is presented with the correct view.

Dependencies:

LoginT, CallM or AddApp

Objective:

When creating a meeting or an appointment, a user should be able to, instead of writing a place where the meeting/appointment should take place, reserve a meeting room from a list of available rooms. The application should create a list with available rooms in the time period when the meeting is scheduled, the user should then be able to choose a room from this list. Should a meeting or an appointment with a booked room be canceled, the application should delete the room reservation from the calendar.

Expected results:

The user should receive a message if the reservation process went through successfully, including the room number and the allocated time frame. If an error occurred, the program should return an error message and file a report to be used for debugging.

Show all - TestId: ShowAll

Precondition:

An connection is already established. The user is presented with the correct view.

Dependencies:

Login, CallM or AddApp

Objective:

The application shall show a weekly calendar in which all the users appointments and meetings are inserted. It should be easy to flip between different weeks.

Expected results:

If the operation is successful, the user should be given a view of a weekly calendar with all his appointments and meetings inserted. If the user has no appointments or meetings, the calendar should be empty, possibly displaying a message that there are no plans for the coming weeks.

Track meeting notifications - TestId: TrckMNot

Precondition:

An connection is already established. The user is presented with the correct view.

Dependencies:

Login, CallM, RcvNoM

Objective:

The calender server should track the following notifications for a meeting; accepted, cancelled and pending. The server should produce a report whenever an update is made on a specific meeting. The report should be sent to the leader of the meeting.

Expected results:

The report from the server is analyzed and the notifications are updated accordingly.

Show other calendars - TestId: ShowOtherCal

Precondition:

An connection is already established. The user is presented with the calendar view.

Dependencies:

Login, CallM or AddApp

Objective:

The application should be able to show other users appointments side by side the current users appointments in his calendar.

Expected results:

The user should, upon choosing another user from the “other calendars” button, be presented

with a calendar view including that users meetings and appointments and meetings, as well as the other users appointments and meetings.