Test script and protocols

Team: g1t2

Member 1: (Christina Brückl, 01219643)

Member 2: (Leonie Katharina Geyr, 11778679)

Member 3: (Thomas Krabichler, csaw4217)

Member 4: (Kevin Rimml, 12036261)

Member 5: (Ioannis Deligiannidis, 11913541)

Proseminar group: PS-703080-1, Team2

Date: 19.05.2022

Änderungshistorie

Version	Datum	Status	Autor	Änderung	Abnahme durch (Name)	Abnahmedatum
0.1	14.4.2022	Working On	Michael Breu	Initial Examples	-	-
0.2	15.05.2022	Working On	Christina Brueckl	First Draft	-	15.05.2022
0.3	19.05.2022	Working On	Ioannis Deligiannidis	Basic Tests Added	-	19.05.2022
0.4	24.05.2022	Working On	Christina Brueckl	Further Edits	-	24.05.2022
0.5	11.06.2022	Working On	Ioannis Deligiannidis	Further Testing	-	11.06.2022

Table of contents

1. Test preparation

1. Test data for Arduino and Rasberry Pi

Hardware preparation Rasberry Pi/Arduino:

1. Plug the Raspberry-Pi to a power source, connect with a keyboard and screen (if possible)
2. Add network - if keyboard and screen:

   sudo nano /etc/wpa_supplicant/wpa_supplicant.conf

   network={
   ssid="<name of your network>"
   psk="<network password>"
   }
   else: plug the sd card into the laptop and open the file wpa_supplicant.conf and add network = ….
3. Local new pi ip address via *"arp -a"* and connect via ssh
4. Change in pi>skeleton>api>SensorController IP_ADDRESSE of the current connected network (this guarantees a connection with the localhost for the rest-api)
5. Plug the Arduino to a power source
6. Log into the pi and run "./start.sh" in the terminal, keep in mind, that the web app has to be up and running (docker)

The integration tests can be started when:

1. The database has been initialized without any connection problems (it should initialize the database skeletondb if does not exists and the data automatically according the the application.properties file).
2. All JUnit5 tests have been run completely and successfully.
3. Starting the web application with the following commands:

   mvn spring-boot:run will start the web-application using the local MySQL database. MySQL connection with port :3306 must exist. The Database skeletondb will be initialized automatically, you don't have to initialize it by yourself. Application has to start on localhost:8080. Please make sure, that you set yours database connection password correctly under the application.properties file (spring.datasource.password=${MYSQL_PASSWORD: < your password>}.

   docker compose up - - build to create the image and to run the database and the web app containers. Application has to start on localhost:8081.

   In both cases above, the database then will automatically be initialized using the data.sql file under the package /src/main/resources/data.sql

2. Test data for the Web Application

The following data are already initialized by data.sql file

User / password	Role	Note
admin / passwd	Administrator	Department Administration

elvis / passwd	Administrator	Department Administration
spiderman / passwd	Facility Manager	Department Administration
hulk / passwd	Manager	Manager of IT-Department
borat / passwd	Manager	Manager of Mathematics Department
alan / passwd	Manager	Manager of Biology Department
From user1 up to user10 / passwd	Employee	Different Departments and offices assigned

3. Test data for the Web Application (Measurement Data)

To test the Measurement Data for each office and public room, we have constructed an Initializer Button in the Mask of the Dashboard. The problem with pre-initializing the data on run time was that, the initialization takes too long time, because it has to initialize too many measurement data for 1 month. So we decided to create a Button instead. By submitting the button, test data will be initialized for one month for the current logged in User. Test data will be initialized for each public room and office for the department where the user belongs. If the corresponding office does not visit the “Rules” for saving measurement data, then the data wont be initialized for that time.

The Rules are:

1) There exists at least one Sensor in that room
2) There are at the moment five or more present Users in the office or no Users at all, so the Data Privacy is not violated.

Note that the action to initialize test data may take some time, so please be patient 😊

2. Test protocol

This section should be completed to complete the test.

Test date: _____ (when have been tested? E.g., Period)

Tester: _____ (who tested?)

Tested Version: _____ (e.g. GIT Tag)

Test entry criteria fulfilled:
yes/no according to _____
(Verweis auf E-Mail der Entwickler, Maventestprotokoll, …)

Test environment:
_____
(e.g., Application locally on your own computer, database on server)

3. Test cases

Hinweis: Die nachfolgenden Testfälle sind nur Beispiele. Erweitern Sie die Testfälle entsprechend Ihrer Konzeptbeschreibung und insbesondere der darin angeführten Use Cases. Vergessen Sie nicht darauf auch allgemeine (Filterung, Sortierung, etc.) funktionale und nicht-funktionale (Antwortzeiten, Stabilität, etc.) Anforderungen mit Ihren Testfällen abzudecken.

The test cases described here fully cover the use cases listed in the concept description. Additional test cases were added to check general functional requirements. Deviations from the expected result statuses were documented as part of the tests carried out (see Chapter 2, test protocol) and classified according to the following classifications.

- **OK:** No deviations found.
- **Cosmetic Deviation:** Minor layout problems: e.g., clumsy line breaks in the text, texts for buttons too long, etc.
- **Middle Deviation:** The functionality is basically available, but can only be used to a limited extent, e.g., some expected entries in a drop-down list are missing, data changes are only visible after closing and reopening a dialog, etc.
- **Large Deviation:** The functionality cannot be used, e.g. action buttons do not show any reaction, data is not written correctly to the database, etc.
- **System unusable:** Running this test leaves the system in an unusable state, e.g. system crashes. Database becomes inconsistent, data is (unplanned) deleted.

3.1. Test Case Login

TC: Login User (Any User role)	
Use Case:	Login
Initial State:	The user (User1) is logged out.
Action	1. The user enters the URL localhost:8080. 2. The login mask appears 3. The user enters user1 and their user password passwd and submits the Login button.
expected outcome state:	The user is logged in. The user sees the dashboard and the settings page.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: Login User (Wrong Password) (Any User role)	
Use Case:	Login
Initial State:	The User is logged out.

Action	4. The user enters the URL localhost:8080. 5. The login mask appears 1. The user enters user1 and a wrong password (password123) and submits the Login button.
expected outcome state:	The user is not logged in. The error message "Wrong username or password!" appears.
observed deviation	

OK cosmetic variations medium variations large variations System unusable

3.2. Test Case Absence

TC: Register absence (Any User role)	
Use Case:	Register Absence
Initial State::	User is logged in and submits the “Settings” button:
Action	1. User settings mask and a calendar appears on the mask. 2. The user choose the day and a dialog window appears. 3. The user chooses “VACATION”, From:<today>, To:<two days from now>, submits “All Day” 4. Submitting Save the settings to save the absence
expected outcome state:	On the calendar a 2-day long bar is shown with the information “VACATION” The Event Details dialog is closed. The absence will be registered, and the user status (Status is an attribute only) will be changed, about 30 mins later. (Every 30 minutes the system checks if there are absences registered and if the time now is equal to it)
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: Un-register absence (Any User role)	
Use Case:	Unregister Absence
Initial State::	User is logged in and submits the “Settings”:
Action	1. User settings mask and a calendar appears on the mask. 2. The user submits the already registered absence. 3. The user submits the “DELETE” button
expected outcome state:	The absence is now removed from calendar.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

3.3 Test Case Email Notification

TC: Register for email - notifications (Any User role)	
Use Case:	Register for email - notifications
Initial State::	User is logged in and submits the “Settings”:
Action	1. The “Settings” mask appears 2. The User has initial Report frequency “OFF” 3. The User submits the daily / weekly / monthly radio button.

expected outcome state:	The Report frequency is updated, and the selected radio button shown as the chosen one. In the background, the system sets the date to send the data to date now plus the selected interval (1 day, 7 days or 30 days respectively) to send the email notification Every day at 12:00 PM the system checks for notifications to send. If the date setted is the day today and it is 12:00 PM, the user will receive an email nofitication with the average measurement of the selected interval. The email will be sent once and the next date to send again an email will be set to date plus the selected interval. To change the settings above and to test the email sending whenever you want, follow the steps below: 1) First, the measurement data should be initialized by the preferred interval from the button on the dashboard. Users email must be a real one in order to receive the email. Then select the mail interval under the settings menu on left bar side. 2) Search for the method below under the path: *webapp/src/main/java/at/qe/skeleton/Tasks/UserSettingsTask.java* and under the method: *doUserSendEmail()* and on line 55, change the *user.setNextEmailPost(LocalDateTime.now().plusDays(users.getMailSettings().getInterval()));* to *users.setNextEmailPost(LocalDateTime.now());* 3) Search method below under the path: *webapp/src/main/java/at/qe/skeleton/configs/SchedulerTaskConfig.java* and under the method: *startEmailTaskTimer( )*, change the HOUR_OF_DAY, MINUTE and SECOND to the time you want to run the test. 4) Finally Search method below under the path: *webapp/src/main/java/at/qe/skeleton/ui/controllers/UserSettingsController.java* and under the method: *selectMailIntervalListener(ValueChangeEvent event)* and on the line 162 replace the “plusDays(..)” method: *getCurrentUser().setNextEmailPost(LocalDateTime.now().plusDays(MailInterval.valueOf(interval).getInterval()));* to: *getCurrentUser().setNextEmailPost(LocalDateTime.now());*

observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: Unregister for email - notifications (Any User role)	
Use Case:	Unregister for email - notifications
Initial State::	User is logged in and submits the “Settings”:
Action	1. The “Settings” mask appears 2. The User has already selected any other frequency except “OFF”. 3. The User submits the “OFF” radio button.
expected outcome state:	The Report frequency is updated and the selected radio button “OFF” shown as the chosen one. In the background, the attribute set to get the next email notification will be set to null. No notifications will be received by the User.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

3.4 Test Case View Audit Log

TC: View Audit Log - Facility Manager, Administrator, Manager	
Use Case:	View Audit Log
Initial State::	User is logged in and has Manager, Facility Manager or Administrator role:
Action	1. The User submits the View Audit Log button

expected outcome state:	The User sees the Audit Log view. If the User is logged in as a Manager, then only logs from its Department will be shown. The data representing are the changes (editing/ adding/removing) from any entity, the Timestamp (when that event happened) and the User who cause that action. The Data in that view cannot be edited or removed from the view.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

3.5 View of climate measurements

TC: View climate data of public rooms (Any User role)	
Use Case:	View climate data of public rooms
Initial State::	User is logged in and the Dashboard appears on the start site:
Action	1. The User submits the Daily, Weekly or Monhtly view as well as the type of data. E.g. Average measurement, Diagram or Measured Data.
expected outcome state:	The user is getting the corresponding view according to the Tab selections. Logged in User as Admin will see a Global Dashboard, a Manager only for its Department and an Employee its own Office and the Public Rooms.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: View climate data of own office room (Any User role)	
Use Case:	View climate data of own office
Initial State::	User is logged in and the Dashboard appears on the start site:

Action	1. The User submits the Daily, Weekly or Monhtly view as well as the type of data. E.g. Average measurement, Diagram or Measured Data.
expected outcome state:	The user is getting the corresponding view according to the Tab selections. Logged in User as Admin will see a Global Dashboard, a Manager only for its Department and an Employee its own Office and the Public Rooms.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: View climate data of all departments - Facility Manager, Administrator	
Use Case:	View climate data of all departments
Initial State::	User is logged in and has Facility Manager or Administrator role:
Action	1. The User submits “Global Dashboard” button to view the global view.
expected outcome state:	The user is getting the corresponding view according to the Tab selections.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: View climate data from all rooms of the department - Manager	
Use Case:	View climate data of rooms from the department
Initial State::	User is logged in and has Manager role:

Action	1. The User submits “Departments Dashboard” button
expected outcome state:	The user is getting the corresponding view according to the Tab selections. Logged in User as a Manager getting views only for its Department.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

3.6 Sensor stations

TC: Add Sensor station to web app - Administrator, Facility Manager	
Use Case:	Add sensorstation
Initial State::	User has added the sensor to the pi, Is logged in as a Administrator or Facility Manager, Pi and Arduino are running:
Action	1. The User opens “Room Management” > “Sensors” 2. On the left upper corner of the sensor list, the user submits the rounded PLUS (+) button 3. The “Add Sensor” dialog appears. 4. Give the new Sensor the name “g1t2_001” and submit 5. The new Sensor is now in the List, but has now Limits
expected outcome state:	To the whole hour the sensor should have now limits set and displays them on the list
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

3.7 Sensor stations – Limits Configuration

TC: Sensor station - Limits configuration	
Use Case:	Configure Limits
Initial State::	User must have the access to the raberry pi code:
Action	1. The User accesses src/main/java/at/qe/skeleton/bleclient/Limits.java 2. The User changes the MIC_TO_LOUD to “155”
expected outcome state:	On the left window side on the web application view, under the menu “Room Management” > “Sensors” the sensors can be viewed. After an hour the limits should be set new and shown on the display
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

3.8 Arduino

TC Auth5.3: User - check limit noise - no LED	
Use Case:	Auth2
Initial State::	Arduino is connected to a powersource, Pi is setup as told above
Action	1. The user sees the Arduino 2. The user claps 5 times every loud very near the mic, pause between clap 4 and 5 30 seconds
expected outcome state:	The LED does not shine cyan
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC Auth5.3: User - check limit noise - LED	
Use Case:	Auth2
Initial State::	Arduino is connected to a powersource, Pi is setup as told above
Action	1. The user sees the Arduino 2. The user plays a song very near the mic for a longer time
expected outcome state:	The LED shines cyan
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

3.9 User Management

TC: Add New User Successfully - Administrator, Facility Manager	
Use Case:	Add New User
Initial State::	User is logged in and has Administrator or Facility Manager role:
Action	1. The User submits the “User Management” button from the menu. 2. The list with all users appears in the User Management mask. 3. On the left side above the User Table, the user submits the rounded PLUS (+) button. 4. The “Create User” dialog appears. 5. The user fills out the form in the dialog, with non-existing Username, Email and the selected Room of the Department must have a slot (Number of seats is not exceeded by the users). 6. The User submits the Add User button on the left corner of the dialog.
expected outcome state:	The dialog window hides and the New User is appended to the list of all users. An email with username and password will be sent to the user.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: Add New Manager Successfully - Administrator, Facility Manager	
Use Case:	Add New Manager
Initial State::	User is logged in and has Administrator or Facility Manager role:
Action	1. The User submits the “User Management” button from the menu. 2. The list with all users appears in the User Management mask. 3. On the left side above the User Table, the user submits the rounded PLUS (+) button. 4. The “Create User” dialog appears. 5. The user fills out the form in the dialog, with non-existing Username, Email and for the selected Department there should no Manager exists (if there is a Manager already, Remove him/her). 6. The User submits the Add User button on the left corner of the dialog.
expected outcome state:	The dialog window hides and the New User is appended to the list of all users. The User has the Manager role. In the Room menu on the left side, the new Manager is assigned to the selected Department An email with username and password will be sent to the user.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: Add New Manager Not Successfully - Administrator, Facility Manager	
Use Case:	Add New Manager
Initial State::	User is logged in and has Administrator or Facility Manager role:

Action	1. The User submits the “User Management” button from the menu. 2. The list with all users appears in the User Management mask. 3. On the left side above the User Table, the user submits the rounded PLUS (+) button. 4. The “Create User” dialog appears. 5. The user fills out the form in the dialog, with non-existing Username, Email and selects a Department where a Manager already exists. The User submits the Add User button on the left corner of the dialog.
expected outcome state:	The dialog window will not hide. A dialog with the proper information appears on the top right corner.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: Add New User Not Successfully - Administrator, Facility Manager	
Use Case:	Add New User
Initial State::	User is logged in and has Administrator or Facility Manager role:
Action	1. The User submits the “User Management” button from the menu. 2. The list with all users appears in the User Management mask. 3. On the left side above the User Table, the user submits the rounded PLUS (+) button. 4. The “Create User” dialog appears. 5. The user fills out the form in the dialog, with an already existing Username or an existing email or selects a Room of a Department which has no slot for new User.

expected outcome state:	If the Username or the email already exists or the room is full, then a dialog with the proper information appears on the right top corner. The dialog window is not hiding and the user is not appended to the list of users.
observed deviation	

OK cosmetic variations medium variations large variations System unusable

TC: Edit an existing User Successfully – Administrator, Facility Manager	
Use Case:	Edit User
Initial State::	User is logged in and has Administrator or Facility Manager role:
Action	1. The User submits the “User Management” button from the menu. 2. The list with all users appears in the User Management mask. 3. The user submits the “Edit” button from the row of the user which is about to be edited. 4. The “Edit User” dialog appears 5. The user then can be edited (Username or Password cannot be edited). 6. Changing the email of the user to a non-existing email or changing the Users room to a non-full room. 7. The User submits the Save button on the left corner of the dialog.
expected outcome state:	The dialog window is hiding and the User has been successfully edited.
observed deviation	

OK cosmetic variations medium variations large variations System unusable

TC: Edit an existing User Not Successfully - Administrator, Facility Manager	
Use Case:	Edit User
Initial State::	User is logged in and has Administrator or Facility Manager role:
Action	1. The User submits the “User Management” button from the menu. 2. The list with all users appears in the User Management mask. 3. The user submits the “Edit” button from the row of the user which is about to be edited. 4. The “Edit User” dialog appears 5. The user then can be edited (Username or Password cannot be edited). 6. Changing the email of the user to an existing email or changing the Users room to a room which capacity is full. 7. The User submits the Save button on the left corner of the dialog.
expected outcome state:	If the email already exists or the room is full, then a dialog with the proper information appears on the right top corner. The dialog window is not hiding and the user is
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: Edit Manager, change Managers Role to Employee - Administrator, Facility Manager	
Use Case:	Edit User
Initial State::	User is logged in and has Administrator or Facility Manager role:

Action	1. The User submits the “User Management” button from the menu. 2. The list with all users appears in the User Management mask. 3. The user submits the “Edit” button from the row of the user which is about to be edited. 4. The “Edit User” dialog appears 5. The user then changes the Managers role to Employee 6. The User submits the Save button on the left corner of the dialog.
expected outcome state:	The Dialog window will hide and from the edited Manager the Department has now no Manager shown. Checking from the Rooms menu on the left side, there is no Department Manager.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: Edit Manager, change Employee to Manager role successfully - Administrator, Facility Manager	
Use Case:	Edit User
Initial State::	User is logged in and has Administrator or Facility Manager role:

Action	1. The User submits the “User Management” button from the menu. 2. The list with all users appears in the User Management mask. 3. The user submits the “Edit” button from the row of the user which is about to be edited. 4. The “Edit User” dialog appears 5. The user then changes the Employee role to Manager 6. The user must select a Department with out Manager. 7. The User submits the Save button on the left corner of the dialog.
expected outcome state:	The Dialog window will hide and from the edited Manager the Department has now a new Manager. Checking from the Rooms menu on the left side, there is the edited User shown as Department Manager.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: Edit Manager, change Employee to Manager role Unsuccessfully - Administrator, Facility Manager	
Use Case:	Edit User
Initial State::	User is logged in and has Administrator or Facility Manager role:

Action	1. The User submits the “User Management” button from the menu. 2. The list with all users appears in the User Management mask. 3. The user submits the “Edit” button from the row of the user which is about to be edited. 4. The “Edit User” dialog appears 5. The user then changes the Employee role to Manager 6. The user must select a Department where a Manager already exists. 7. The User submits the Save button on the left corner of the dialog.
expected outcome state:	The Dialog Window is not hiding and a propper message is shown.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: Remove User – Administrator, Facility Manager	
Use Case:	Remove User
Initial State::	User is logged in and has Administrator of Facility Manager role:
Action	1. The User submits the “User Management” button from the menu. 2. The list with all users appears in the User Management mask. 3. The user submits the “Delete” button from the row of the 4. A confirmation dialog appears and the user submits the “YES” button.
expected outcome state:	The entity User has been removed from the user list, as well as from the database.
observed deviation	

OK cosmetic variations medium variations large variations System unusable

3.10 User Settings

TC: Change Current logged in user Email successfully	
Use Case:	Edit Settings
Initial State::	User is logged in and has any Role:
Action	1. The User submits the “Settings” button from the menu. 2. The Settings mask appears. 3. Under the Mail panel and in the input box of Email Address the user types an email that not exists 4. The user submits the Change Address button
expected outcome state:	The email has been changed and a dialog window with the corresponding message appears.
observed deviation	

OK cosmetic variations medium variations large variations System unusable

TC: Change Current logged in user Email Unsuccessfully	
Use Case:	Edit Settings
Initial State::	User is logged in and has any Role:
Action	1. The User submits the “Settings” button from the menu. 2. The Settings mask appears. 3. Under the Mail panel and in the input box of Email Address the user types an email that not exists 4. The user submits the Change Address button
expected outcome state:	The email has been not changed and a dialog window with the corresponding message appears.

observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: Change Current logged in user password successfully	
Use Case:	Edit Settings
Initial State::	User is logged in and has any Role:
Action	1. The User submits the “Settings” button from the menu. 2. The Settings mask appears. 3. Under the Reset Password panel and in the input box of Current Password the user types its password correctly 4. The user submits the Reset Password Button 5. Two further input boxes will then appear. 6. The user types in both the same password 7. The user submits the button Save new password
expected outcome state:	The password has been successfully changed to the new one and a dialog window with the corresponding message appears. The user can now logout and submit that the password has been changed.
observed deviation	
OK cosmetic variations medium variations large variations System unusable	

TC: Reset Current logged in user password Unsuccessfully	
Use Case:	Edit Settings
Initial State::	User is logged in and has any Role:

Action	1. The User submits the “Settings” button from the menu. 2. The Settings mask appears. 3. Under the Reset Password panel and in the input box of Current Password the user types its password not correctly 4. The user submits the Reset Password Button
expected outcome state:	The password has been not successfully reseted and a dialog window with the corresponding message will appear.
observed deviation	

OK cosmetic variations medium variations large variations System unusable

TC: Change Current logged in user password Unsuccessfully	
Use Case:	Edit Settings
Initial State::	User is logged in and has any Role:
Action	1. The User submits the “Settings” button from the menu. 2. The Settings mask appears. 3. Under the Reset Password panel and in the input box of Current Password the user types its password correctly 4. The user submits the Reset Password Button 5. Two further input boxes will then appear. 6. The user types in both not the same password 7. The user submits the Save new password button
expected outcome state:	The password has been not successfully changed and a dialog window with the corresponding message will appear.
observed deviation	

OK cosmetic variations medium variations large variations System unusable

1. Weitere nichtfunktionale Testfälle

Führen Sie in diesem Abschnitt weitere Testfälle zur nachvollziehbaren und reproduzierbaren Überprüfung relevanter nichtfunktionaler Anforderungen an. Dies sind z.B.:

- *Tests zu Antwortzeiten*
- *Konsistenz der Nutzeroberfläche*
- *Stabilitätstests*

3.3. Referenzierte Dokument

- Concept Description (Version, Datum)