IT Hardware Manager

1.4 Network Services

LARS WIEGERS & JESSE DE WILD

TABLE OF CONTENTS

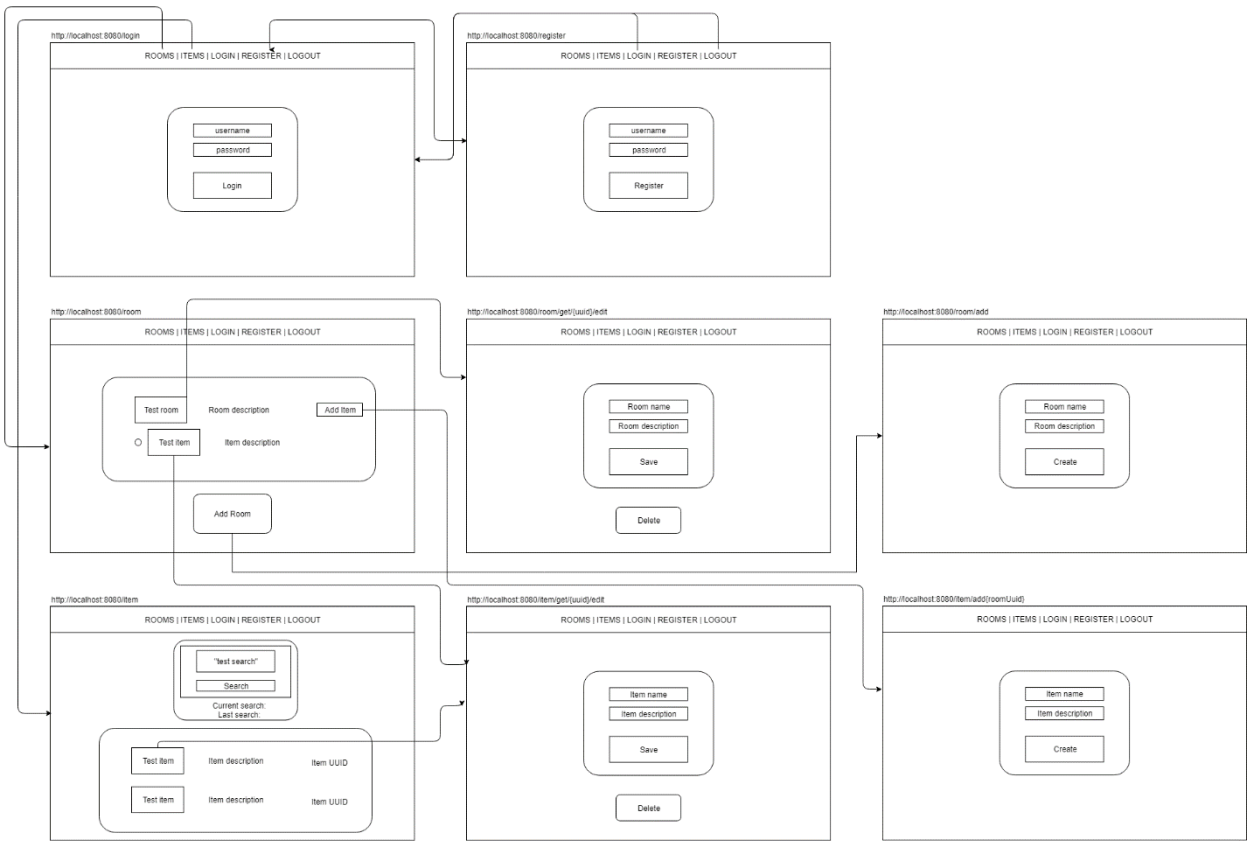
APP DESCRIPTION

The IT Hardware Manager website can be used to store Rooms with Items. All the Rooms are stored in one list, and inside the Rooms, a list of Items can be stored. For example, the first Room with the name ‘Living Room’ contains three items; a television with the name ‘tv’, a computer with the name ‘pc’ and a laptop with the same name as the object itself. We tried to make the app as user friendly as it could be. With working buttons, searching forms and cookies.

WIREFRAMES

In the diagrams below, you will find a complete drawing of how the website itself looks, all the pages and behaviors of the buttons.

Every top bar is the same in every page, with the Rooms, Items, Login, Register and Logout. (If the diagram is unreadable, the .png file is inside the /documents folder in the projects folder)



USE OF THE APPLICATION

The way to run the web application is to open and start the project in IntelliJ. Go to http://localhost/8080/register to go to the register page. After that, you will be redirected http://localhost/8080/login to login.

There are two hardcoded accounts already:

Username	**Password**
jesse@saxion.nl	jessePassword
lars@saxion.nl	larsPassword

There is already one dummy Room and Item added to the system, so you can already test out the application itself. If you want to remove the hardcoded data, it is stored inside the **Storage.java** class inside the main source folder.

USE CASE DIAGRAMS

We have only one actor in this system, namely the 'User'. Everyone is able to register, login, logout and see the rooms and items.

LOGIN

Use case	Login into the website
Summary	The user can use the website when you have logged in.
Actor(s)	User
Precondition	The user has an account registered.
Scenario	- User has an account - User fills in username - User fills in password - User clicks login - User is redirected to the room page
Postcondition	The user is logged in and is redirected to the room page.

REGISTER

Use case	Register an account
Summary	The user can use the website after registering.
Actor(s)	User
Precondition	The user does not have an account.
Scenario	- User clicks 'Register' - User fills in username - User fills in password - User clicks login - User is redirected to the room page
Postcondition	The user is logged in and is redirected to the room page.

LOGOUT

Use case	Logout of the application
Summary	The user is logging out in the current session of the website.
Actor(s)	User
Precondition	The user is logged in.
Scenario	- User clicks 'Logout' - User is redirected to the login page
Postcondition	The user is logged out and is redirected to the login page.

ROOMS

Use case	See all the rooms and items of the room
Summary	The user can see a list of all the rooms with every item inside that room.
Actor(s)	User
Precondition	The user is logged in.
Scenario	- User clicks 'Rooms' - User can see a list of rooms and items
Postcondition	The user can see a list of all the rooms and items.

Use case	Edit and save a room
Summary	The user can edit and save a room.
Actor(s)	User
Precondition	The user is logged in and has clicked on a room's name.
Scenario	- User clicks 'Rooms' - User clicks on a room's name the user wants to edit and save - User enters a new room name and/or description - User clicks 'Save' - User is redirected to the rooms page
Postcondition	The user can see a list of all the rooms and items with the edited room name and/or description.

Use case	Delete a room
Summary	The user can delete a room.
Actor(s)	User
Precondition	The user is logged in and has clicked on a room's name.
Scenario	- User clicks 'Rooms' - User clicks on a room's name the user wants to delete - User clicks 'Delete' - User is redirected to the rooms page
Postcondition	The user can see a list of all the rooms and items without the just deleted room (items included).

ITEMS

Use case	See all the items
Summary	The user can see a list of all the items.
Actor(s)	User
Precondition	The user is logged in.
Scenario	- User clicks 'Items' - User can see a list of items
Postcondition	The user can see a list of all the items.

Use case	Edit and save an item
Summary	The user can edit and save an item.
Actor(s)	User
Precondition	The user is logged in and has clicked on an item's name.
Scenario	- User clicks 'Items' - User clicks on an item's name the user wants to edit and save - User enters a new item name and/or description - User clicks 'Save' - User is redirected to the items page
Postcondition	The user can see a list of all the items with the edited item name and/or description.

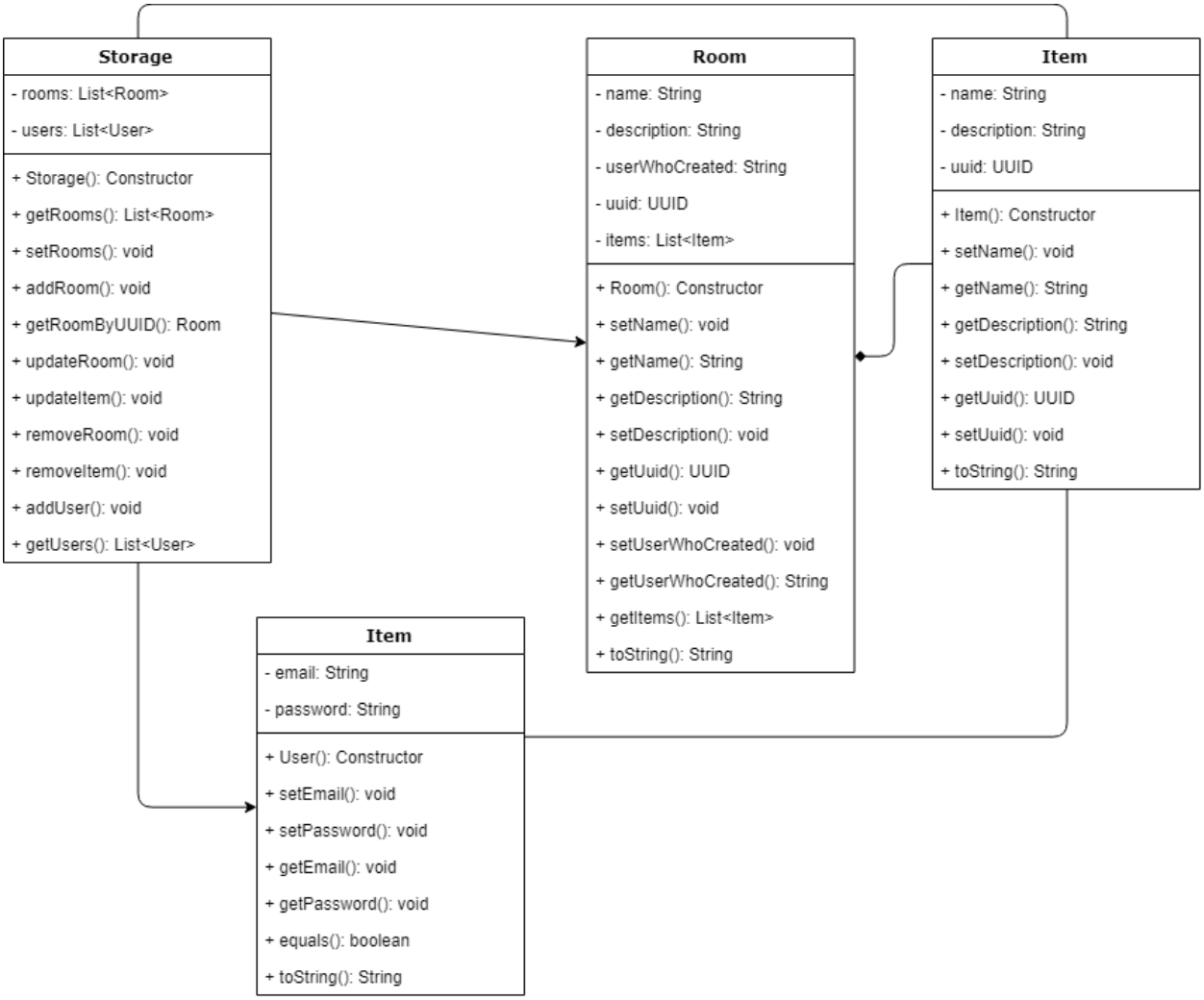
Use case	Delete an item
Summary	The user can delete an item.
Actor(s)	User
Precondition	The user is logged in and has clicked on an item's name.
Scenario	- User clicks 'Rooms' - User clicks on an item's name the user wants to delete - User clicks 'Delete' - User is redirected to the items page
Postcondition	The user can see a list of all the items without the just deleted item.

SEARCH

Use case	Search an item
Summary	The user can search an item/a list of items.
Actor(s)	User is logged in.
Precondition	The user is logged in and has clicked on an item's name.
Scenario	- User clicks 'Items' - User fills in a search term inside the open field above 'Search' - User clicks 'Search' - User can see a list of items with the search term
Postcondition	The user can see a list of all the items with the searched term.

CLASS DIAGRAM

Below you will find the class diagram. The Controller's are not included. The Storage class is used to maintain the all the user's to login into the website. It also stores all the rooms in the system.



API DESCRIPTION

This page describes all the URL's, if they are GET or POST mappings and their input/output.

REQUEST MAPPINGS

- http://localhost:8080/login
  Already have an account? Login via this URL.
- http://localhost:8080/register
  If you want to register, use this URL.
- http://localhost:8080/logout
  This URL is used to logout of the current session.
- http://localhost:8080/room
  This is the main URL for functions for rooms.
- http://localhost:8080/item
  This is the main URL for functions for items.

GET MAPPINGS

- http://localhost:8080/login
  When a user is not logged in, this mapping will bring you to the login-index.html. If the session does exist, it will redirect you to the /room.
- http://localhost:8080/register
  When a user is not logged in, the user can create a new account. When created, the user is redirected to /room.
- http://localhost:8080/logout
  This mapping brings you to the login-index.html, namely /login.
- http://localhost:8080/room
  Adds full list of rooms with items to the model, which sends it to the room-index.html. It will show a list of rooms with items.
- http://localhost:8080/room/add
  Brings the user to the edit-room.html, for creating a new room.
- http://localhost:8080/room/get/{uuid}/edit
  It adds the room's UUID to the {uuid} to edit the clicked room, it will bring you to the edit-room.html.
- http://localhost:8080/item
  Adds full list of items to the model, which sends it to the item-index.html. It will show a list of rooms with items.

  This list can differ on the added search value, which will change the URL to http://localhost:8080/item?search={search} with the added search param.

- http://localhost:8080/item/add/{roomUuid}
  Brings the user to the edit-item.html, for creating a new item to the room by the room UUID.
- http://localhost:8080/item/get/{uuid}/edit
  It adds the item's UUID to the {uuid} to edit the clicked item, it will bring you to the edit-item.html.

POST MAPPINGS

- http://localhost:8080/login [CREATE]
  Requests a username and password. If you entered a correct combination, it will redirect you to the /room and start a session. If not, it will redirect to you the /login again.
- http://localhost:8080/register [CREATE]
  Requests a username and password and adds the new created user to the Storage and session before redirecting you to /room.
- http://localhost:8080/room/add [CREATE]
  Requests a name and description param, which will add a room to the Storage. It will redirect you back to /room.
- http://localhost:8080/room/get/{uuid}/edit [UPDATE]
  Requests a room UUID, name and description to edit the room. It will redirect you back to /room.
- http://localhost:8080/room/remove/{uuid} [DELETE]
  It requests the room UUID to delete the room. It will redirect you back to the /room page.
- http://localhost:8080/item/add/{roomUuid} [CREATE]
  Requests a name and description param and room UUID, which will add an item to the Storage via the room's UUID. It will redirect you back to /item.
- http://localhost:8080/item/get/{uuid}/edit [UPDATE]
  Requests a room UUID, name and description to edit the item. It will redirect you back to /item.
- http://localhost:8080/item/remove/{uuid} [DELETE]
- It requests the item UUID to delete the item. It will redirect you back to the /item page.