**Log-in use case:**

The user runs the application and is required to enter his username(ID) and password. Then, after the system analyzes his details, it decides whether the user is a board member, or just a resident and redirects him to the relevant page.

Detailed use case:

**Scope:** ØstManSys

**Level:** Sub-function

**Primary Actor:** Resident/Board Member

**Stakeholders and Interests:**

**Resident** and **Board Member** want to have a secure log-in system, which will keep their details safe and make sure they are redirected to the respective to their role page.

**Company** wants to avoid security flaws and have a properly working log-in system.

**Preconditions:** Resident/Board Member runs the application.

**Success Guarantee(or Post-conditions):** The resident/board member is correctly identified and redirected to the respective page.

**Main Success Scenario:**

1. Resident/Board Member runs the application.
2. Fills out Username and Password fields.
3. Is redirected to Resident/Board Member view.

**Extensions:**

1. Incorrect password, or username.  
   - The user introduces a wrong password, or username.  
   - An error message is displayed.

**Special requirements:**

A working device, supporting UWF(PC, Tablet, Phone, etc.).

**Technology and Data Variations List:**

**Frequency of Occurrence:**

Once. When the application is being run.

**Miscellaneous:**

What happens if a user writes a wrong password more than three times?   
 What happens if a user forgets his/her password?

**Showing information about residents and apartments use case:**

The Board Member, after entering his/her username and password, is redirected to the respective view, where a list of apartments is shown and, when clicked, the board member can see detailed description of the selected apartment, including all the people residing in that apartment. Also, the board member can press on the name of any of the residents and see full details about that person. More than that, he/she can sort the list of the apartments, based on last inspection date, number of residents, ID. At the same time, he/she can see all the details about his own account.

Detailed use case:

**Scope:** ØstManSys

**Level:** User-goal

**Primary Actor:** Board Member

**Stakeholders and Interests:**

**Board Member** wants to have a good overview of all the apartments and their residents, to be able to add and delete users and to get information about the previous and upcoming maintaining processes. And, also, have an easier access to the contact information of all the residents. Finally, a board member wants an extension, which is going to be a connecting wire between him/her and residents – to see the problem reports.

**Company** wants ensure a better communication between its board members and residents, so that the administration process is more effective and technical evidence can be kept, in order to prevent the eventual conflicts.

**Preconditions:** Board Member runs the application. Fills out Username and Password fields. Is redirected to the Board Member View page.

**Success Guarantee(or Postconditions):** Board member gets the respective view containing a list of apartments with a short description and, also, his full details. Selecting an apartment, a new page opens, containing the full description about the selected apartment(address, people living there and full information about downpipes, windows and so on).

**Main Success Scenario:**

1. Board Member runs the application.
2. Fills out Username and Password fields.
3. Is redirected to Board Member View.
4. Gets his/her details displayed.
5. Gets a list containing all the apartments, with a short description.
6. Presses on an apartments.
7. Gets the detailed description of each apartment, including the names of its residents.
8. Presses on a resident’s name.
9. Gets full details about the selected resident.

**Extensions:**

1. **Problem notification**a. The board member gets a list containing all the apartments.  
   b. The system checks if there are any problems reported.  
   c. The system finds an apartment, whose residents reported a problem.

d. The respective apartment is market with a right sign.  
e. The board member sees the notification and can start solving the reported problem.

1. **Planned inspection**a. The board member gets a list containing all the apartments.  
   b. The system checks the date of the last inspection.  
   c. Adds approximately 2 years and compares this date to the current date.  
   d. In case the remaining period is shorter than 2 months, a yellow sign is displayed.  
   e. A list containing all the soon-to-be planned inspections is generated automatically.
2. **Sorting**a. The board member gets a list containing all the apartments.  
   b. A sorting option is chosen from the available drop-down menu.  
   c. All the apartments are sorted, based on the chosen preference.  
   d. The arranged list is displayed.
3. **Searching**a. The board member gets a list containing all the apartments.  
   b. A searching field can be found in the right upper corner.  
   c. The board member formulates his/her request and presses the “Search” button.  
   d. A list of items(both apartments and residents), matching the searching criteria, is displayed.

**Special requirements:**

A working device, supporting UWF(PC, Tablet, Phone, etc.).

**Technology and Data Variations List:**

**Frequency of Occurrence:**

Multiple times, whenever a board member needs some information about a specific apartment or resident.

Miscellaneous:

What happens in case no items matching the searching criteria are found?

**Registering a resident account use case:**

On the main page for board members, the user can click the “Add a resident” button, which will redirect it to a new page, containing all the necessary forms, which should be filled in and the user should be attached to an existing apartment.

Detailed use case:

**Scope:** ØstManSys

**Level:** Sub-function

**Primary Actor:** Board Member

**Stakeholders and Interests:**

**Board Member** wants to be able to easily add new residents and attach them to the existing apartments, without having any technical flaws and preventing legal errors.

**Resident** wants to be quickly registered in the system and get access to all the facilities in a short period of time.

**Company** wants to make sure the system runs as smoothly as possible and all the parts are satisfied.

**Preconditions:** Board Member fills out the log-in fields and is correctly identified. Fills out all the compulsory fields, required when registering a new resident.

**Success Guarantee(or Post-conditions):** Board Members fills out all the forms for creating a new user, presses the “Add” button and a new resident is registered and attached to an existing apartment.

**Main Success Scenario:**

1. Board Member runs the application.
2. Fills out “Username” and “Password” fields.
3. Is redirected to the Board Member view.
4. Clicks “Add a New Resident” button.
5. Fills out all the necessary forms.
6. Presses “Add”.
7. A new resident is added.
8. Board Member is redirected to Board Member’s main view.

**Extensions:**

1. Too many residents are added to one apartment.  
   a. A board member presses “Add a New Resident” button, fills out all the details and presses the “Add” button.  
   b. The system checks the approximate number of people who can live legally in that apartment(based on the number of square meters of the specific apartment – approximately, 17 sq.m./resident).  
   c. An error message is thrown.
2. Two residents with the same details  
   a. A board member presses “Add a New Resident” button, fills out all the details and presses the “Add” button.  
   b. The system checks if there are any interferences between the details of the user being registered and other users in the system.  
   c. If there are any, an error message is thrown.
3. Not all the fields are filled out  
   a. Board member presses the “Add a New Resident” button, fills out some of the details about the respective user and presses the “Add” button.  
   b. The system checks if all the forms were filled.  
   c. If not, an error message is thrown.

**Special requirements:**

A working device, supporting UWF(PC, Tablet, Phone, etc.).

**Technology and Data Variations List:**

**Frequency of Occurrence:**

Multiple times, whenever a board member to add a new resident in the system.

**Miscellaneous:**

How is a resident converted into a board member?

**Attach residents to an apartment(by the main resident) use case:**

On the main page for resident, the user can click the “Add a resident” button, which will redirect it to a new page, containing all the necessary forms, which should be filled in and the user should be attached to an existing apartment.

Detailed use case:

**Scope:** ØstManSys

**Level:** Sub-function

**Primary Actor:** Resident

**Stakeholders and Interests:**

**Resident** wants to be able to easily attach the other residents to his/her apartment and add their full details.

**Board Member** wants to make sure he/she has access to all residents’ details and have everything organized, so that he/she knows precisely how many people live in each apartment and in the entire complex.

**Company** wants to make sure the system runs as smoothly as possible and all the parts are satisfied.

**Preconditions:** Resident fills out the log-in fields and is correctly identified. Fills out all the compulsory fields, required when registering a new resident.

**Success Guarantee(or Post-conditions):** Residentfills out all the forms for adding a new person, presses the “Add” button and a new resident is attached to an existing apartment.

**Main Success Scenario:**

1. Resident runs the application.
2. Fills out “Username” and “Password” fields.
3. Is redirected to the Resident view.
4. Clicks “Attach a Resident” button.
5. Fills out all the necessary forms.
6. Presses “Attach”.
7. A new resident is attached to the apartment.
8. Resident is redirected to Resident’s main view.

**Extensions:**

1. Two residents with the same details  
   a. A resident presses “Attach a Resident” button, fills out all the details and presses the “Attach” button.  
   b. The system checks if there are any interferences between the details of the user being registered and other users in the system.  
   c. If there are any, an error message is thrown.
2. Not all the fields are filled out  
   a. Resident presses the “Attach a Resident” button, fills out some of the details about the respective user and presses the “Attach” button.  
   b. The system checks if all the forms were filled.  
   c. If not, an error message is thrown.

**Special requirements:**

A working device, supporting UWF(PC, Tablet, Phone, etc.).

**Technology and Data Variations List:**

**Frequency of Occurrence:**

Multiple times, whenever a resident wants to attach a person to the existing apartment.

**Miscellaneous:**

How is a resident converted into a board member?

**See the list of past residents:**

A board member runs the application, fills out “Username” and “Password” fields. Then, after the system identifies his/her role, the relevant page is displayed, containing a list of apartments, his details and a few buttons, among which is “See all residents”, which, once pressed, opens a new page, containing a list of all residents. On this page, there is a drop-down menu, containing more sorting criteria. From these, the “See previous resident” will display only previous residents. Also, there is a possibility to see previous residents for each apartment in part. Going back to the relevant page displayed, the board member choses an apartment – whose previous residents he/she wants to see – and presses the “See previous residents” button, which will open a new page, containing a list of previous residents only for this apartment.

Detailed use case:

**Scope:** ØstManSys

**Level:** Sub-function

**Primary Actor:** Board Member

**Stakeholders and Interests:**

**Board Member** wants to be able to easily see a list of past residents, both for the entire complex of buildings and each apartment, in particular.

**Resident** wants to be legally registered in the system and avoid any breaches of the contract.

**Company** wants to make sure all its residents keep on contract’s statements and have access to the previous resident’s contact details, in order to come with new proposals.

**Preconditions:** Board Member fills out the log-in fields and is correctly identified.

**Success Guarantee(or Post-conditions):** Board Member fills out Username and Password fields. Is correctly identified. Is redirected to the respective page. Presses the “See all residents” button. Gets a list of current residents. Has the option to sort them, where he/she can choose “Previous residents”. A list of previous residents is displayed.

**Main Success Scenario:**

1. Board Member runs the application.
2. Fills out “Username” and “Password” fields.
3. Is redirected to Board Member view.
4. Presses the “See all residents” button.
5. Gets a list containing all the residents.
6. From the drop-down menu, he/she chooses to see only the previous residents.
7. A list of previous residents is displayed.
8. Board Member runs the application.
9. Fills out “Username” and “Password” fields.
10. Is redirected to Board Member view.
11. Gets a list containing all apartments.
12. Choses one apartment.
13. A page containing detailed information about the selected apartment is displayed.
14. Presses the “See previous residents” button.
15. A page, containing a list of previous residents, is displayed.

**Extensions:**

1. **Cost-saving**a. The system checks if there are residents who moved out more than 5 years ago.

b. The found residents are deleted from the data-base, in order to save storage given by data-base hosting services and, therefore, reduce costs. Also, those previous residents will not be sent any advertisements.

**Special requirements:**

A working device, supporting UWF(PC, Tablet, Phone, etc.).

**Technology and Data Variations List:**

**Frequency of Occurrence:**

Multiple times, whenever a board member wants to see the history of previous residents, or to get their previous contact details.

**Miscellaneous:**

What happens with the discharged residents?  
 How can a previous resident be made current, again?

**Reporting a problem use case:**

The user sees all the details about his apartment and presses the “Report a problem” button, which will redirect him to a new page, where he can enter a subject and a description of the problem and he, also, can attach an image and then click the “Send” button, which will send all the entered data to all the board members.

Detailed use case:

**Scope:** ØstManSys

**Level:** Sub-function

**Primary Actor:** Resident

**Stakeholders and Interests:**

**Board Member** wants to be able to react, in a short period of time, to and solve the problems residents face with their living conditions.

**Resident** wants to be able to report a problem, in case something breaks in the apartment and get is solved as soon as possible, without causing more damage to his and other apartments.

**Company** wants to ensure a good communication between board members and residents and react fast to the occurring problems.

**Preconditions:** Residentfills out the log-in fields and is correctly identified.

**Success Guarantee(or Post-conditions):** Resident fills out Username and Password fields. Is correctly identified. Is redirected to the respective page. Presses the “Report a Problem” button. Is redirected to a new page, containing a form for reporting a problem(header, detailed description, a photography of the reported defect). After completing all the fields, the “Report” button is pressed and a confirmation message is displayed.

**Main Success Scenario:**

1. Resident runs the application.
2. Fills out “Username” and “Password” fields.
3. Is redirected to resident view.
4. Presses the “Report a Problem” button.
5. Is redirected to another page, containing a form for reporting a problem.
6. Fills out all the forms.
7. Presses “Report”.
8. Gets a confirmation message.

**Extensions:**

1. **No-image report**
   1. Resident fills out only the “Header” and “Detailed description” fields, without attaching an image to the reported case.
   2. The system processes the information and throws a confirmation notice.

**Special requirements:**

A working device, supporting UWF(PC, Tablet, Phone, etc.).

**Technology and Data Variations List:**

**Frequency of Occurrence:**

Multiple times, whenever a residents wants to report a defect in his/her apartment.

**Miscellaneous:**

What happens if no board member sees the notification?  
 Can a resident report two cases, uploading the same image for both of them?

**Adding an apartment use case:**

The board member sees a list containing all the apartments from his complex, presses the “Add apartment button”. Then, he/she is redirected to a new page, containing all the necessary fields for adding an apartment(size, the number of apartment, etc.). The system checks whether all the fields have been filled up, or not and, in case they have, a confirmation message is thrown. Otherwise, an error message is thrown.

Detailed use case:

**Scope:** ØstManSys

**Level:** Sub-function

**Primary Actor:** Board Member

**Stakeholders and Interests:**

**Board Member** wants to be able to easily add an apartment to the existing list of apartmets.

**Company** wants to make sure its system is up-to-date, containing all the information in real-time.

**Preconditions:** Residentfills out the log-in fields and is correctly identified.

**Success Guarantee(or Post-conditions):** Resident fills out Username and Password fields. Is correctly identified. Is redirected to the respective page. Presses the “Add an apartment” button. Is redirected to a new page, containing a form for adding an apartment(size, the number of apartment, etc.). After completing all the fields, the “Add” button is pressed and a confirmation message is displayed.

**Main Success Scenario:**

1. Board Member runs the application.
2. Fills out “Username” and “Password” fields.
3. Is redirected to Board Member view.
4. Presses the “Add an apartment” button.
5. Is redirected to another page, containing a form for adding a new apartment.
6. Fills out all the forms.
7. Presses “Add”.
8. Gets a confirmation message.

**Extensions:**

1. **Not all the details entered**
   1. Board Member fills out only the some of the fields from the form for adding an apartment, without giving full details.
   2. The system processes the information and throws a confirmation notice.

**Special requirements:**

A working device, supporting UWF(PC, Tablet, Phone, etc.).

**Technology and Data Variations List:**

**Frequency of Occurrence:**

Multiple times, whenever a board member wants to add an apartment to the complex.

**Miscellaneous:**