Product Specifications

Tenant/Superintendent mobile based interface

# Table of contents

[Table of contents](#h.1keoe6m5bxvx)

[Executive summary](#h.weh20xyrov95)

[Entities/Relationships/Attributes definition](#h.h46ekrzci4gc)

[Users and Roles](#h.oe1eoe3q3u7v)

[Tenants](#h.39myusqj7lsz)

[Superintendents](#h.k492776au80u)

[V2. Owners](#h.1hxcg6f2vu9)

[Buildings & Apartments](#h.npljfs9r7og5)

[Work requests](#h.78nl3mtzl8pa)

[V2. User Identification process](#h.a5idii1axg2z)

[Functions](#h.c16rkrvzz5k7)

[For the tenant](#h.lqasu8ibc341)

[Submit a work request](#h.t8u68crldxrs)

[View on-going requests](#h.pjqld8fsymlm)

[View details of any on-going request](#h.p0x9zr8p16zw)

[V2. View announcements notifications from superintendent](#h.j9utnob0g5c4)

[For the superintendent (V2. and the owner)](#h.wuemwt2ngqyu)

[View on-going requests](#h.ygqq0eov6vrr)

[View/Update details of any on-going request](#h.i7f2wjy6qkyc)

[V2. Issue an announcement to the tenant](#h.vnct2hms2mh5)

[V2. Administration rights](#h.kja6ahtgh9ki)

[Work requests](#h.oi0239fisqq3)

[List of work request types](#h.ugb8fnohmf6d)

[Detailed process of work requests](#h.uaobpz1plok1)

[V2. Specific features per work requests](#h.cke0njcpab01)

[V2. Customizable work requests](#h.e0bnkx5bm09y)

[Notifications](#h.8yiyqifb26i5)

[Work request notifications](#h.fyzbkiexbjpt)

[Notification of work request submission](#h.xd1idovwtx6w)

[Notification of response from superintendent](#h.t5mju4620kq)

[Notification of work request resolution](#h.5aqdtu1mzgc)

[V2. Announcements](#h.w62gxnvsj60e)

[Interface](#h.7e6f5ul0bvdm)

# Executive summary

The mobile application would be a tool to connect tenants with their superintendent and vice versa quickly and easily.

**Functionality**:

Tenants:

The main functionality would allow the tenants to notify the superintendent the following:

- service requests - Items that need to be repaired

- potentially dangerous/ damaging situations

- bug/insect problems

- required clean ups or snow removal

- Urgent messages: locked out, gas leak (app will ensure tenant has already called 911), etc.

The app would contain a comprehensive list of all typical service and other requests.

Additionally, functionality would allow for tenants to communicate with all other tenants about building related events or social activities.

Superintendent:

The main functionality would also allow the superintendent to communicate the following:

- receipt of service request

- approximate time of resolution

- building notifications (turning water off at certain times, heat broken for west building, etc.)

The superintendent will be able to see all pending requests and confirm completion of service requests when completed. (This will send a notification to the tenant as well).

**Other Requirements**:

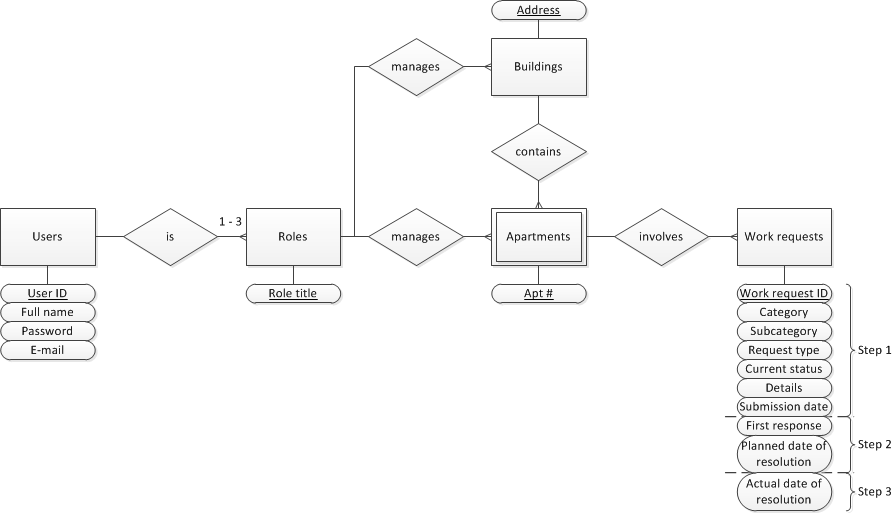
Each "tenant" may have multiple users associated with that unit (i.e. husband, wife, teenager, etc)

Anyone added as a tenant would need to be verified by superintendent.

Superintendent will be able to toggle between buildings if intending to multiple buildings. This feature will also enable business owners to see maintenance statuses at their different buildings.

# Entities/Relationships/Attributes definition

Entities are related as follows in the application:



## Users and Roles

Users are characterized by:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Possible value(s)\*** | **Unicity requirements** |
| User ID | Integer | - | Unique |
| Full name | Text | - | - |
| Password | Text | - | - |
| E-mail | Text | - | Unique |

A **user table**\*\* on our server will contain all existing accounts, and their associations to the fields listed above.

*\*In V2, robust type checks will need to be implemented at user creations.*

*\*\*The user table will need to be specifically secured in a commercial version.*

*\*\*\*Those affiliations should be accomplished in the Building/Apartment table*

Roles are characterized by:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Possible value(s)** | **Unicity requirements** |
| Role title | Text | Tenant  Superintendent  Owner | - |

### Tenants

Tenant profiles are created and associated to apartments by the superintendents.

Tenants create work requests.

### Superintendents

Superintendent profiles are created when subscribing to the application via our company\*.

Superintendents create tenant profiles, buildings and apartments (they are automatically associated in the user table to the buildings and apartments they create).

They administrate the associations between tenant profiles and buildings/apartments.

(cf. §V.B.5. Administration rights)

### V2. Owners

Owner profiles are created when subscribing to the application via our company\*.

Owners have identical rights to superintendents, but are associated to buildings and not to apartments.

***\*We will need to refine our business model here: are we selling the app to owners or superintendents?***

## Buildings & Apartments

Buildings are characterized by:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Possible value(s)** | **Unicity requirements** |
| Address | Text | - | Unique |

Apartments are characterized by:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Possible value(s)** | **Unicity requirements** |
| Apt # | Text | - | Unique within a same building |

Buildings and apartments are created and named (text) by the superintendent or the owner via the administration interface (cf §V.B.5. Administration rights).

Buildings must always contain at least one apartment.

## Work requests

**Creation:**

A work request is created when the tenant selects a work request, completes the associated form and submits it (cf. Step 1 of §VI.B) Detailed process)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Type** | **Determination** | **Input** | **Default value** |
| Category | Text | cf. VI.A) List of functions | Touch selection | - |
| Subcategory | Text | cf. VI.A) List of functions | Touch selection | - |
| Request type | Text | cf. VI.A) List of functions | Touch selection | - |
| Current step | Integer | At each step completion | Auto | - |
| Building | Text | Completed by tenant in step 1 | Drop-down menu  (must only list building(s) of the user) | 1st building associated with user in user table |
| Apartment | Text | Completed by tenant in step 1 | Drop-down menu  (must only list apartment(s) of the user) | 1st apartment associated with user in user table |
| Details | Text (long) | Completed by tenant in step 1 | typed in | - |
| Work request ID | Integer | At submission (end of step 1) | Auto | - |
| Date of submission | Date & Time | At submission (end of step 1) | Auto | - |
| First response | Text (long) | Completed by superintendent in step 2 | typed in | - |
| Planned date of resolution | Date & Time | Completed by superintendent in step 2 | typed in | - |
| Date of resolution | Date & Time | At resolution (end of step 3) | Auto | - |

# V2. User Identification process

Identification is done through an e-mail/password portal\*, which constitutes the very first screen of the application.

When a user submits a user name/password, the application checks in the user table:

* The existence of the user name
* The match between password and user name

(If any of the two above fails, the process stops and returns the same error message)

* The role matched with this user name

The app then grants access to:

* if the user only has one role only: the portal associated to his/her role,
* if the user has several roles: to a selection portal where the user selects which portal he wishes to access (obviously, he is only offered the roles he has).

*\*For the mobile application, the option to remain identified should be implemented.*

*To consider for next versions:*

* *A “Password forgotten?” function*
* *A “Change password” function*

# Functions

Immediately after logging in, the users,accordingly with their respective roles, are offered the following functions:

## For the tenant

### Submit a work request

Cf. §V.B. Detailed process of work requests, step 1.

### View on-going requests

When selecting this option, the user is provided with a list of all work request which status is different of 4 (Closed).

V2. The list displays information sum up about the work requests in a table.

By touching any of the on-going requests, the user initiates the function described next §: **View details of any on-going request** for the touched work request.

### View details of any on-going request

The viewed work request is presented in a table listing all the fields of the work requests and their current values (graphical improvements to be specified in V2).

1. **V2. View closed requests (and their details)**

### V2. View announcements notifications from superintendent

## For the superintendent (V2. and the owner)

### View on-going requests

Identical page to §IV.A.2. “View on-going requests”.

### View/Update details of any on-going request

Identical page to §IV.A.3. “View on-going requests”, however the superintendent has access to an additional button: **UPDATE/CLOSE**, which effect is described in §V.B.Steps 2 and 3.

1. **V2. View closed requests (and their details)**

### V2. Issue an announcement to the tenant

### V2. Administration rights

* Create buildings
* Create apartments and associate them to buildings
* Create tenant profiles
* Enable/Disable work request types

# Work requests

## List of work request types

Requests are sorted by categories and subcategories as follows:

|  |  |  |
| --- | --- | --- |
| **Category** | **Sub-category** | **Request type** |
| C1. Fix me | S1. Plumbing | R1. Broken faucet |
|  |  | R2. Leaking faucet |
|  |  | R3. Toilet overflow |
|  |  | R4. Roof leak |
|  |  | R5. Flooding |
|  | S2. Electrical/Appliances | R1. Burnout bulb |
|  |  | R2. Power outage |
|  |  | R3. Oven not working |
|  |  | R4. Fridge not working |
|  |  | R5. Other electrical issue |
|  |  | R6. Other appliance issue |
|  | S3. Tempertature’s not right | R1. Heating not working |
|  |  | R2. Central air not working |
|  |  | R3. AC not working |
|  | S4. Other repairs | R1. Broken window |
|  |  | R2. Broken lock |
|  |  | R3. Tiles broken |
| C2. Infestation | S1. Bugs | R1. Roaches |
|  |  | R2. Bees |
|  |  | R3. Bed bugs |
|  |  | R4. Others |
|  | S2. Others | R1. Mice/Rats |
|  |  | R2. Others |
| C3. Outdoor maintenance | S1. Seasonal maintenance | R1. Snow removal needed |
|  |  | R2. Leaf removal needed |
|  |  | R3. Grass cutting needed |
|  |  | R4. Bush pruning needed |
|  |  | R5. Other clean up |
|  | S2. Outdoor repair | R1. Broken steps |
|  |  | R2. Broken tiles |
|  |  | R3. Door |
|  |  | R4. Porch |
| C4. Doorman duties | S1. Notifications | R1. Visitor coming |
|  |  | R2. Let in my apartment |
|  |  | R3. Delivery |

## 

## Detailed process of work requests

**Step 1:**

The tenant:

* Starts the app
* Clicks on “Issue a work request”
* Selects a Category, then a Subcategory, then a Work request
* Fills in the work request submission form (which, in V1 contains the field “**details**”, “**building**”, “**apartment**”. Those fields are mandatory (hitting submit returns an error message if “details” is left empty). Building and apartment are drop down menus which, by default, contain the first apartment associated to the tenant in the user table.
* Hits submit

At submission:

* A new work request is created. Its work request ID is generated (an incremental number is attributed to each new work request)
* The work request is updated with:
  + Category, Subcategory and Request type depending on the work request type selected initially by the user,
  + Building, apartment, details,
  + Date of submission (= Current date)
  + Current step of the new work request = 2 - “Submitted”
* A **notification of work request submission** (cf. §B.1) is sent to the superintendent

**Step 2:**

The superintendent (after receiving the notification of new work request):

* Accesses the concerned on-going work request detailed view (cf. §4.B.2),
* Clicks on UPDATE/CLOSE
* Fills in the work request response form (which, in V1, only contains the fields “**planned date of resolution**” and “**First response**”. Only “planned date of resolution” is mandatory (hitting submit returns an error message if “planned date of resolution” is left empty).
* Hits submit

At submission:

* The work request is updated with:
  + Planned date of resolution and First response
  + Current step = 3 - “Planned for resolution”
* A **notification of response from superintendent** (cf. §V.B.2) is sent to the tenant.

**Step 3:**

The superintendent (after having achieved the requested work):

* Accesses the concerned on-going work request detailed view (cf. §4.B.2),
* Clicks on UPDATE/CLOSE
* Fills in the work request closing form (which, in V1 contains no field)
* Hits submit

At submission:

* The work request is updated with:
  + Date of resolution = Current date
  + Current step = 4 - “Closed”
* A **notification of work request resolution** (cf. §V.B.3) is sent to the tenant.

## V2. Specific features per work requests

## V2. Customizable work requests

# Notifications

Notifications are e-mails\* addressed to either superintendents or tenants. Those e-mails include different type of information depending on the kind of notification. We distinguish the following types of notifications, for which we detail the information that they carry.

*\*in the mobile application version, those notifications will be actual phone notifications.*

## Work request notifications

Work request notifications are all based on the recap table, which includes the following information:

Tenant information:

|  |  |
| --- | --- |
| **User name** | *- value -* |
| **Building** | *- value -* |
| **Apartment** | *- value -* |

Work request:

|  |  |
| --- | --- |
| **Work request ID** | *- value -* |
| **Work request type** | *- value -* |
| **Date of submission** | *- value -* |
| **Details** | *- value -* |
| **Planned date of resolution** | *- value - (empty when finishing step 1)* |
| **First response** | *- value - (empty when finishing step 1)* |
| **Date of resolution** | *- value - (empty when finishing step 2)* |

Only e-mail objects and recipients differ depending on the notification, at least in V1:

### Notification of work request submission

To: superintendent associated to the apartment in the building/apartment table

Object: Work request submission

### Notification of response from superintendent

To: tenant

Object: Response to work request

### Notification of work request resolution

To: tenant

Object: Work request resolution

## V2. Announcements

# Interface

See prototype here:

<http://people.stern.nyu.edu/ahg299/websys/mobile_proto.html>