# Introduction

Website where you can choose the location of the workplace/office, and see and mark, which equipment item is currently used by someone and which is free to use.

# Database

## Requirement analysis for the database

### Introduction

Website where you can choose the location of a workplace, the city of it for example, and see and mark, which equipment item is currently used by someone and which is free to use.

### Actions

* N1: You can add and remove locations of the workplaces
* N2: Locations consist of a name
* N3: You can add and remove equipment items to the location
* N4: Items consist of a name and ID
* N5: Items can be marked “used” or “unused” for everyone to see

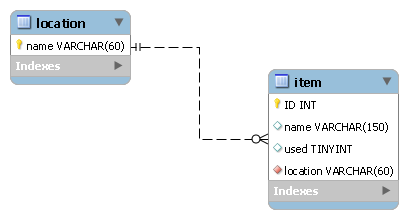
### Connections

User interface – in the first page you choose the location from a list of locations, you can also add a new location with a button that brings up a modal with input for it. Then you can see information of the items of the location presented in a list in the second page. Each item has a “Yes” or “No” button in the list that signifies if the item is in use. When these buttons are clicked, they change from “Yes” to “No” and vice versa. You can add a new item with a button that brings up a modal with input for it. You can enable “Remove Items Mode” with a button – in this mode “Remove” buttons will be shown for the items in the list, these buttons remove the items. You can disable “Remove Items Mode” with a button. You can remove the current location with a button in the same page.

### Other features

If user tries to add a location that already exists, an error message will be displayed. Prepared statements are used to protect against SQL injections, htmlspecialchars() is used to protect against cross-site scripting.

## ER diagram



## Field definitions and description of the foreign keys

Location name – VARCHAR(60).

Item: ID – INT, name – VARCHAR(150), used – BOOLEAN, location – VARCHAR(60).

Referential Integrity

In "location"-table name is the foreign key to "item"-table.

If we update or delete the name in "location" table, we will cascade the updates or deletes to "item"-table.

## A SQL code for creating the tables

CREATE TABLE location(

name VARCHAR(60) PRIMARY KEY

) Engine=InnoDB;

CREATE TABLE item(

ID int AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(150),

used BOOLEAN,

location VARCHAR(60),

FOREIGN KEY (location) REFERENCES location(name)

ON UPDATE CASCADE

ON DELETE CASCADE

) Engine=InnoDB;