HeroSPOTTER

Technical Design Document

Karl Marx

TAble of  
Contents

OverVIEW 2

Terms and Definitions 2

User Stories 4

Application Architecture 6

Test Plan 12

## Overview

The Hero Education and Relationship Organization (HERO) has recognized a need to allow users to report Superhero sightings and view other user-reported sightings. HeroSpotter will accomplish just this in an feature-rich, interactive web application that will both satisfy consumers’ SuperHero curiosity and provide invaluable consumer data to various business units at HERO.

## Terms and definitions

2.1 Technical Terms

* MySQL: a database system that efficiently stores data.
* Java: object-oriented language, upon which this program will be built.
* Python: a scripting language that can be incorporated easily into a Java program.
* JavaScript: a programming language that enables live functionality in web browsers.

2.2 User Roles

* Admin: Users with administration privileges will be allowed to view, create, update, and delete any data stored in the database. They will also be tasked with approving user-submitted images before they can be displayed.
* End User: End users will be able to view and create data (validated to remove obscenities). The text will appear immediately, but the pictures will only appear after approved by an administrator.

2.3 Data Entities

* Supers: Superheroes and supervillains created (data-wise) by admins or end users.
* Sightings: User reported instances of superheroes or supervillains appearing in public at a specific location.
* Locations: User input locations that have been validated as actual places.
* Powers: User-created/described powers that can belong to one or more supers.
* Organizations: user-described groups to which one or more supers belong.

2.4 Processes and Formulae

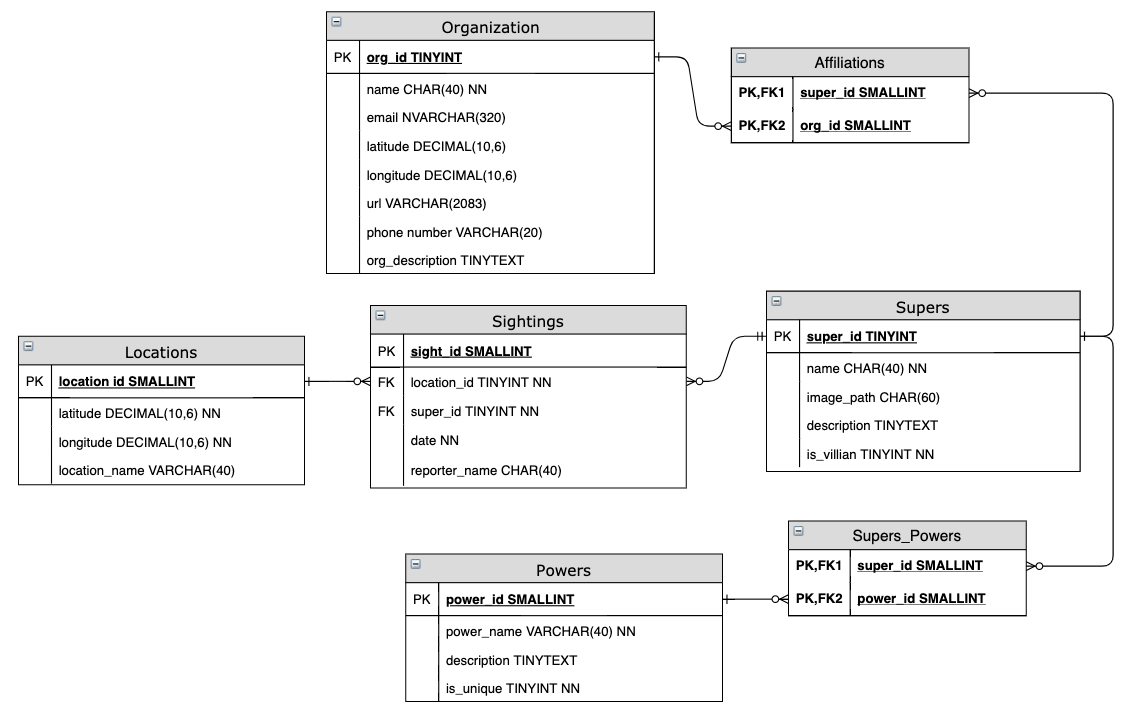
* Image moderation: All user submitted images will be moderated by an admin before being visible by the end users. Administrators will have an extra option in the Sightings view for this.
* Profanity validation: will be completed using a Profanity-Filter, a very accurate Python-based library for filtering out profanity. Other validation will occur in the browser, using JavaScript
* Data storage: All information will be stored in a MySQL database allowing for simple access for real-time analysis by other business units (e.g. Marketing and Sales)
* Location storage: Locations will be stored as single sets of coordinates to allow for simple analysis of geographic business trends by other units.

## User stories

|  |  |  |
| --- | --- | --- |
|  | USER STORY | WEIGHT |
| 1 | As a user, I would like to know more about this page. | 3 |
| 2 | As a user/admin, I would like to view recent sightings in a map and a news feed. | 8 |
| 3 | As a user/admin, I would like to report a new sighting. | 8 |
| 4 | As a user/admin, I would like to see all sightings or those for a specific date. | 8 |
| 5 | As a user/admin, I would like to display and/or add a new superhero/supervillain. | 5 |
| 6 | As a user/admin, I would like to display and/or add a new location. | 5 |
| 7 | As a user/admin, I would like to display and/or add a new organization. | 5 |
| 8 | As a user/admin, I would like to display and/or add a new superpower. | 5 |
| 9 | As an admin, I would like to approve pending photos. | 8 |
| 10 | As an admin, I would like to delete supers, locations, organizations, or powers. | 8 |
| 11 | As an admin, I would like to update supers, locations, organizations, or powers. | 8 |

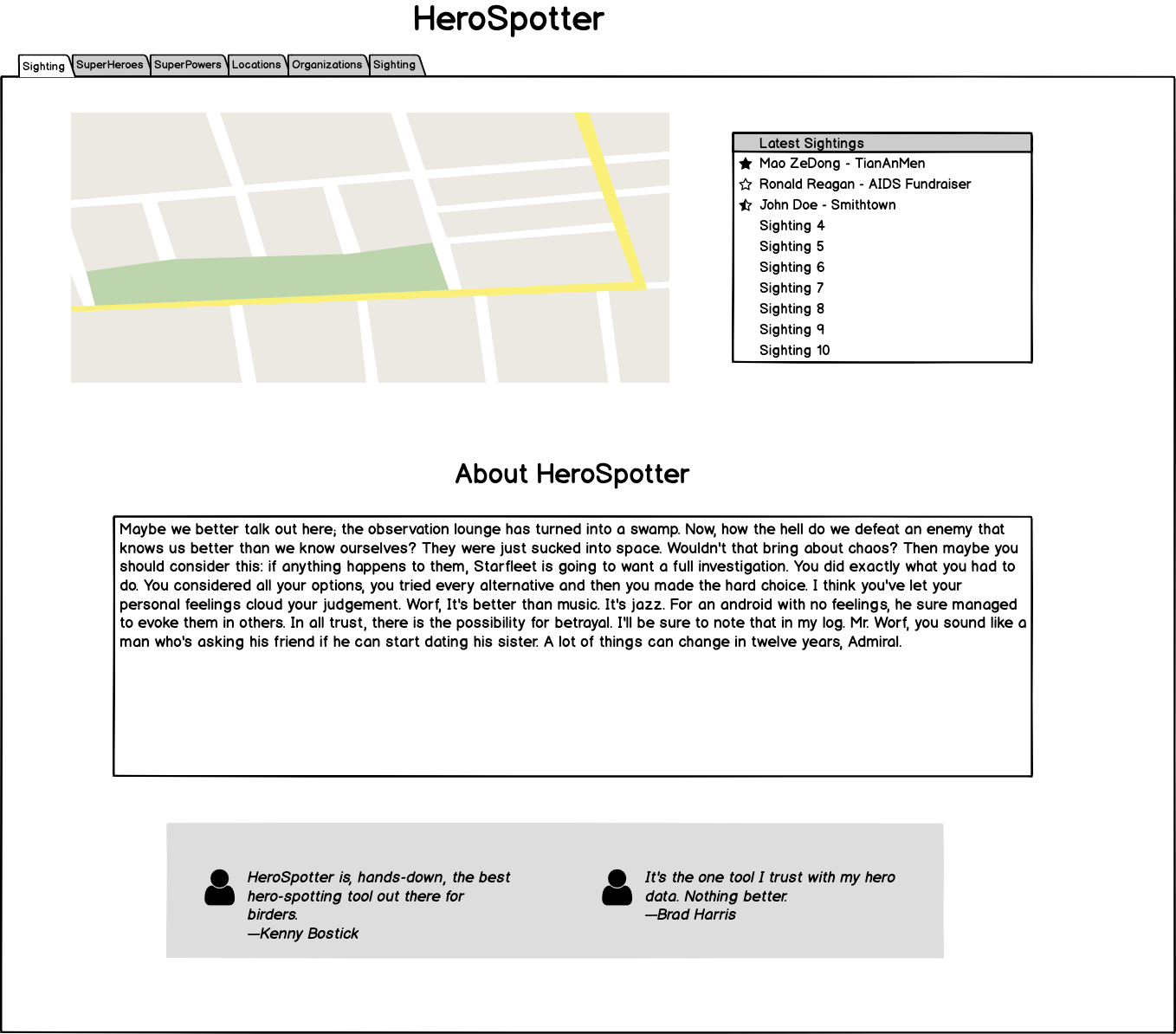
## application architecture

4.1 Entity Relationship Diagrams

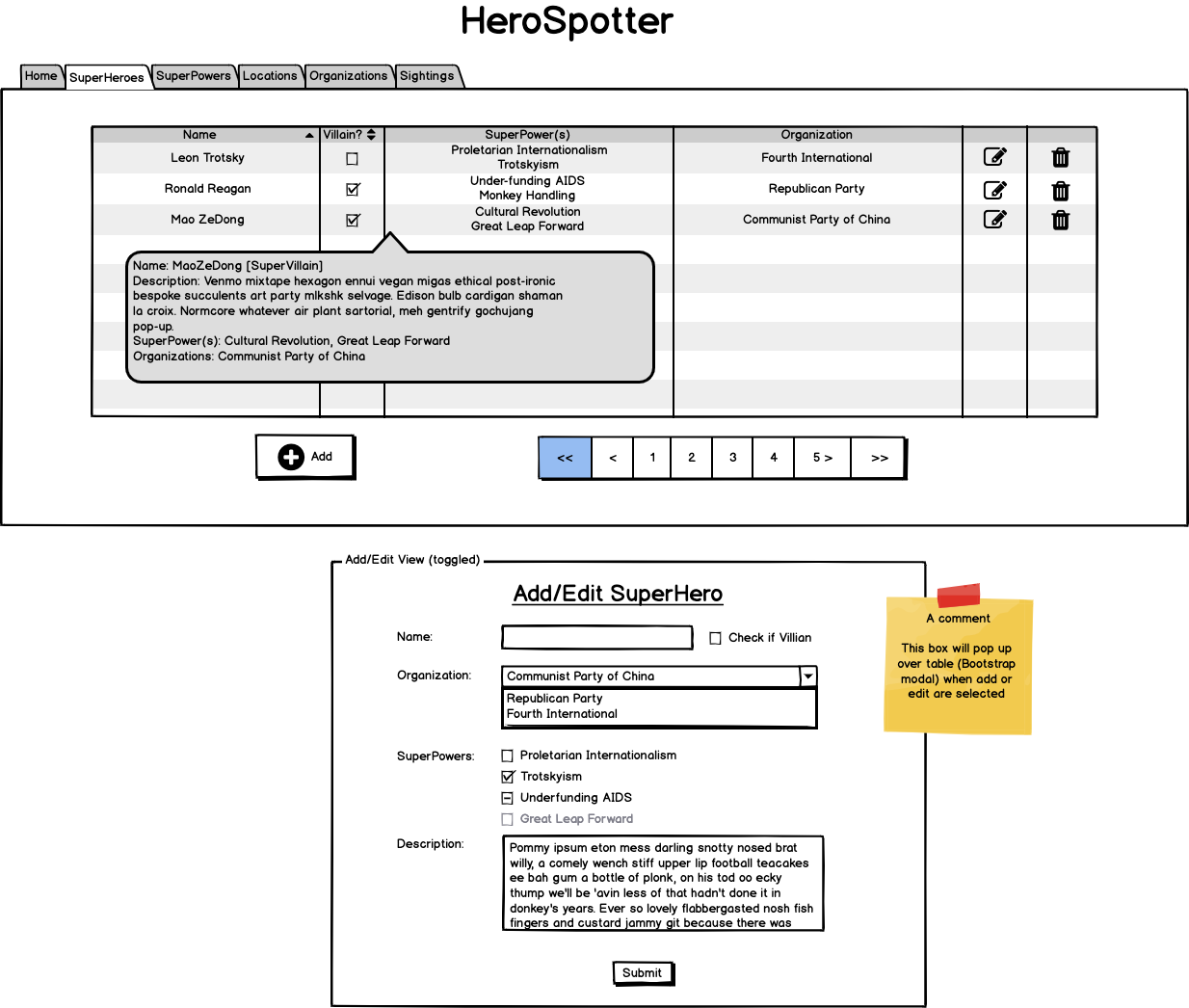


4.2 Entity Relationship Diagrams

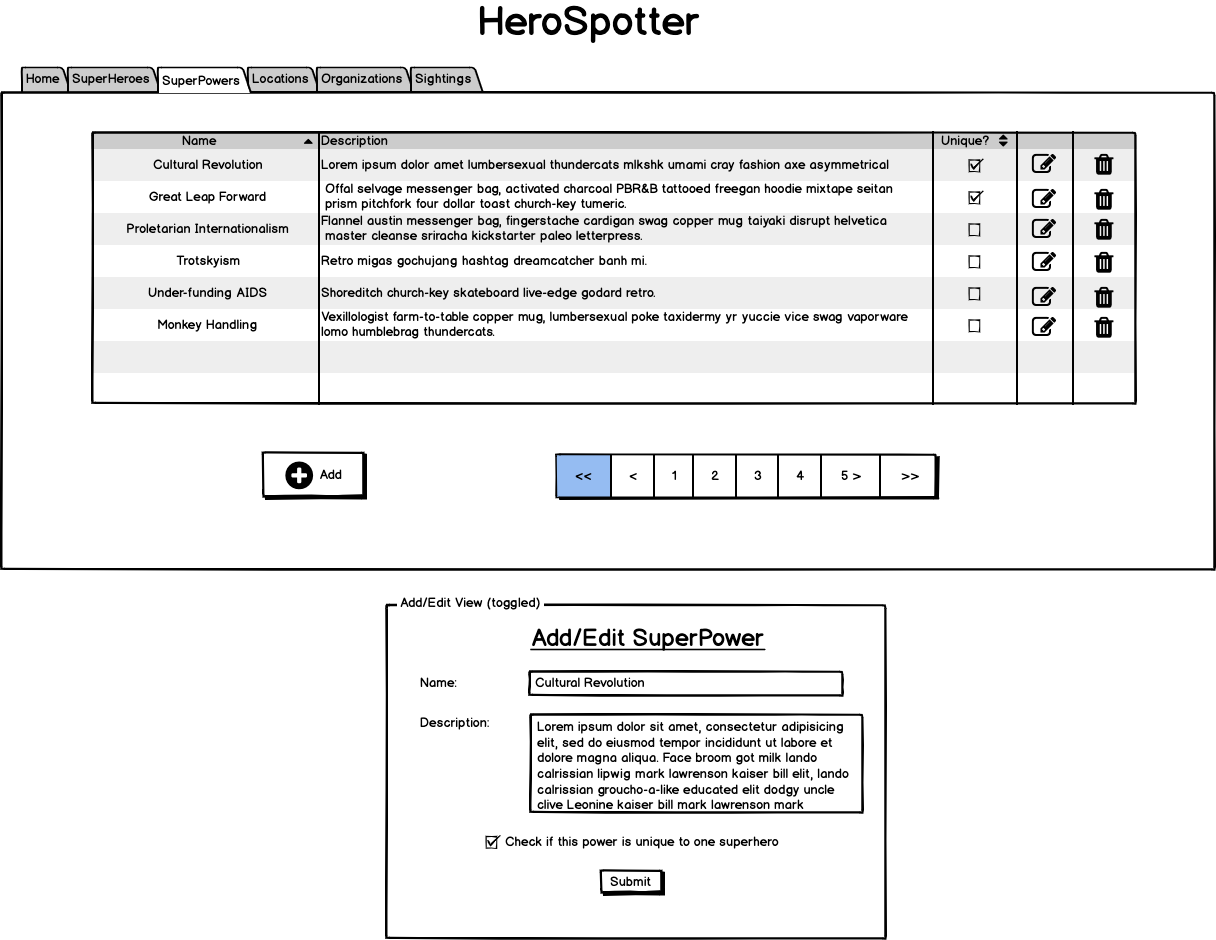
4.2.1 - Home



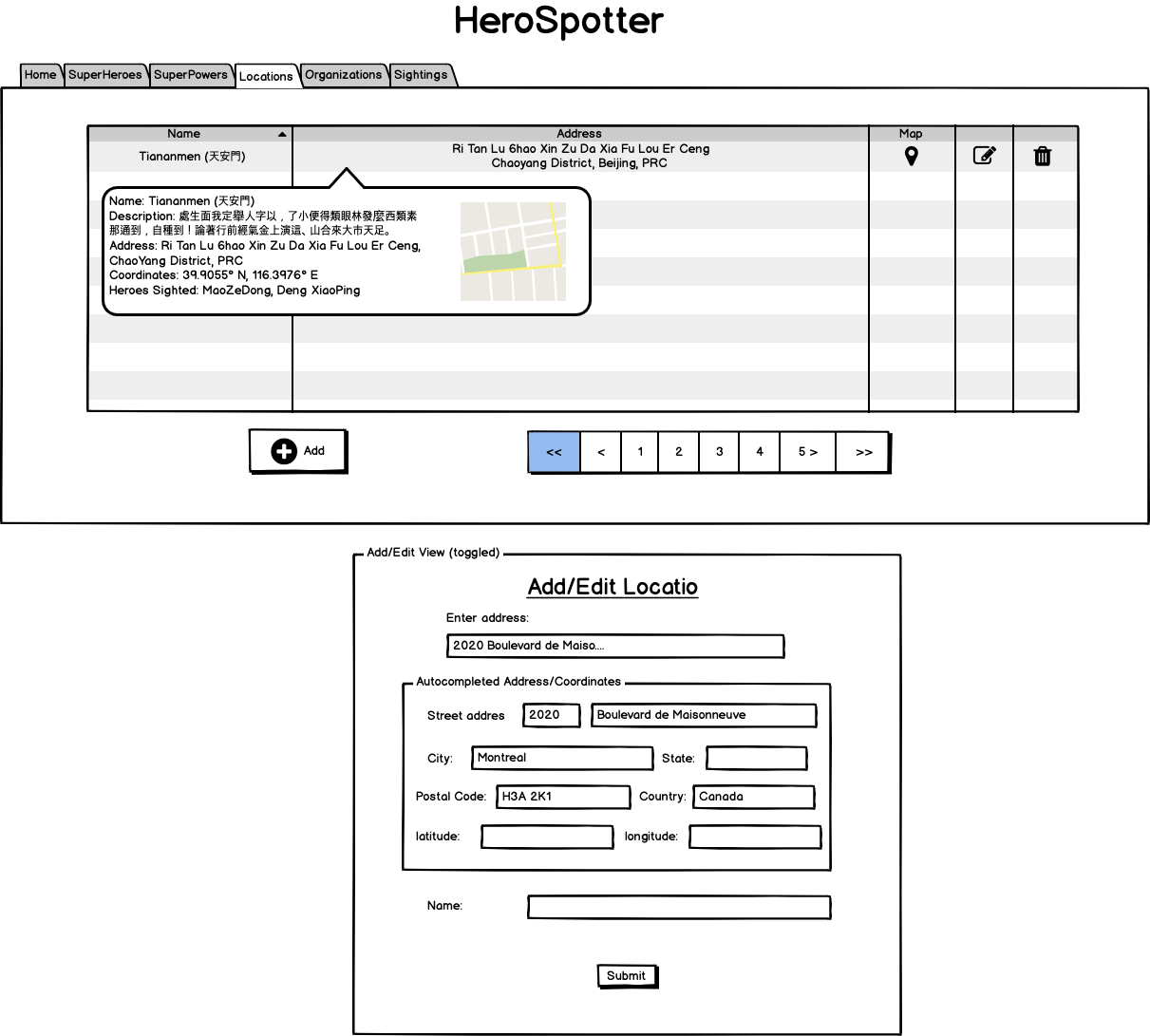
4.2.2 - Supers



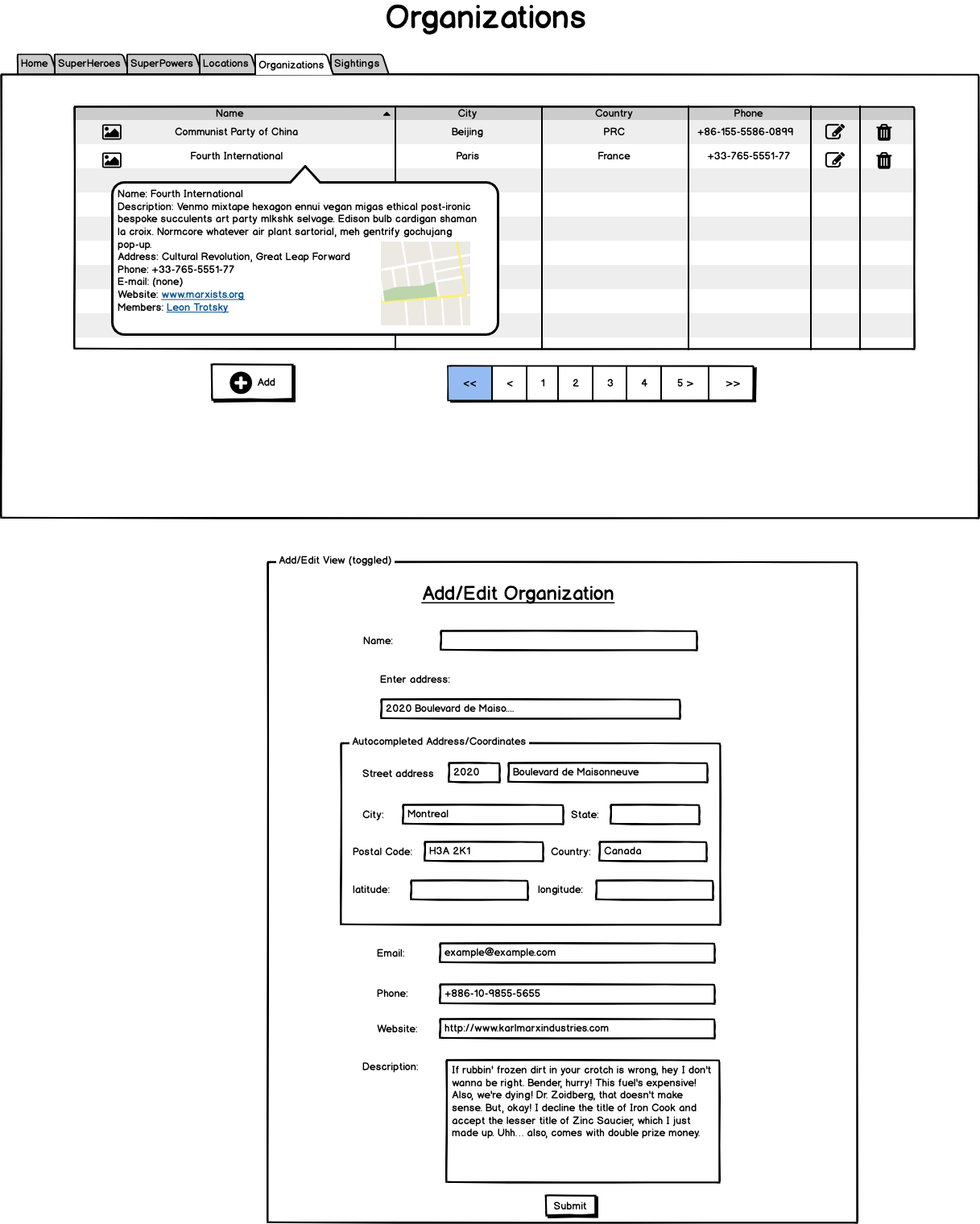
4.2.3 – SuperPowers



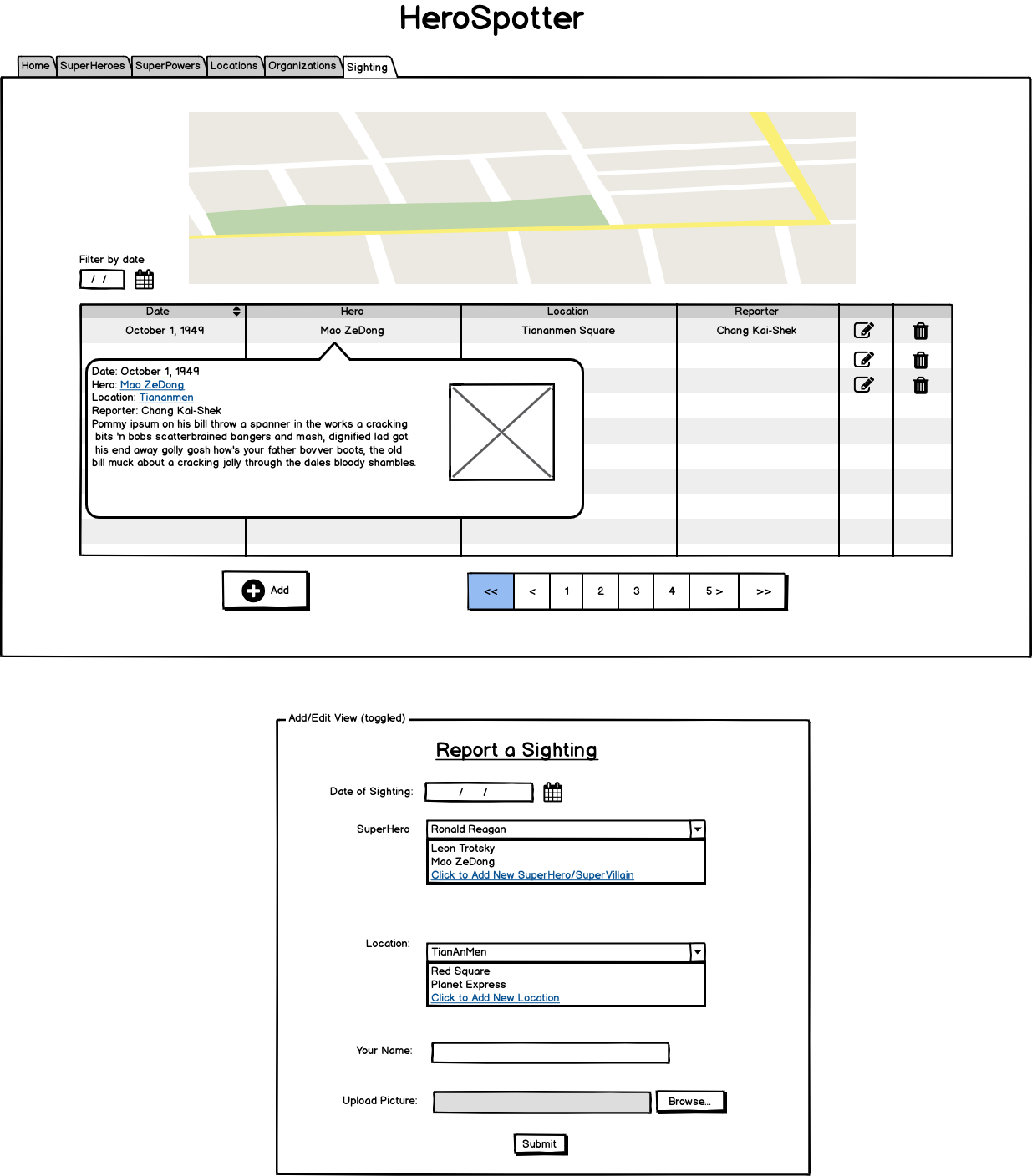
4.2.4 – Locations



4.2.5 – Organizations



* + 1. – Sightings



## Test Plan

All create/read/update/delete (CRUD) functions between the application and the database will be subjected to a battery of tests. The profanity filter will be thoroughly tested to ensure it filters subtle profanity but that it is not so strict that it rejects acceptable language. Censoring and displaying images before and after moderation respectively will be tested as well. All functions that affect rendering of the user interface will be tested under a wide variety of scenarios.