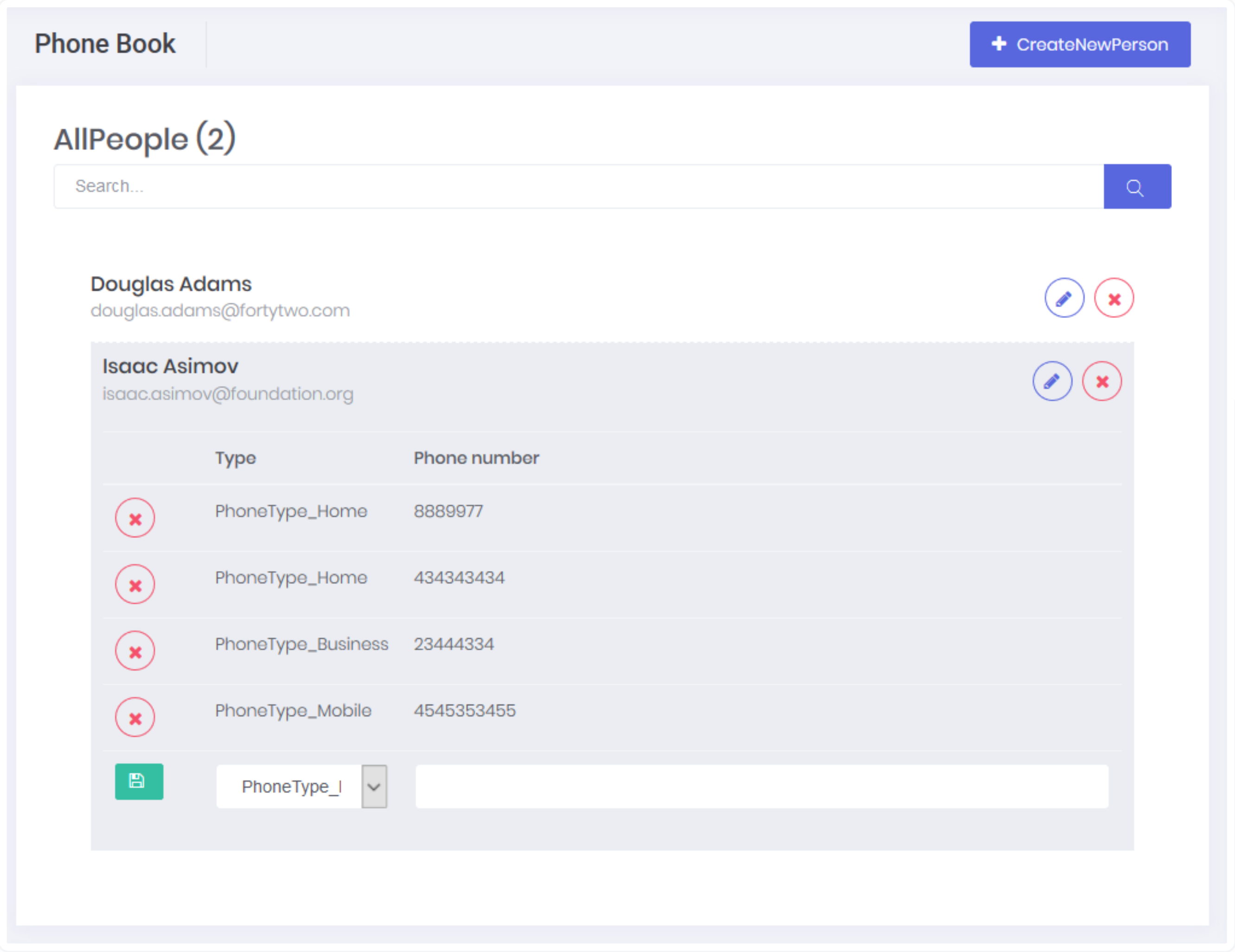
Edit Mode For Phone Numbers

Final UI is shown below:



Phone Book

AllPeople (2)

Douglas Adams
douglas.adams@fortytwo.com

Isaac Asimov
isaac.asimov@foundation.org

	Type	Phone number
✖	PhoneType_Home	8889977
✖	PhoneType_Home	434343434
✖	PhoneType_Business	23444334
✖	PhoneType_Mobile	4545353455
💾	PhoneType_I	

Feedback

When we click the **edit icon** for a person, its row is expanded and all phone numbers are shown. Then we can delete any phone by clicking the icon at left. We can add a new phone from the inputs at last line.

View

```
<div *ngFor="let person of people" [ngClass]="{'bg-secondary kt-padding-15': person
    <div class="row kt-row--no-padding align-items-center">
        <div class="col">
            <h4>{{person.name + ' ' + person.surname}}</h4>
            <span>{{person.emailAddress}}</span>
        </div>
        <div class="col kt-align-right">
            <button (click)="editPerson(person)" title="{{'Edit' | localize}}" clas
                <i class="fa fa-pencil"></i>
            </button>
            <button id="deletePerson" (click)="deletePerson(person)" title="{{'Dele
                <i class="fa fa-times"></i>
            </button>
        </div>
    </div>
    <div class="row">
        <div class="col-sm-12 kt-margin-t-20" *ngIf="person===editingPerson">
            <table class="table table-hover">
                <thead>
                    <tr>
                        <th style="width:10%"></th>
                        <th style="width:15%">{{"Type" | localize}}</th>
                        <th style="width:75%">{{"PhoneNumber" | localize}}</th>
                    </tr>
                </thead>
                <tbody>
                    <tr *ngFor="let phone of person.phones">
                        <td>
                            <button *ngIf="'Pages.Tenant.PhoneBook.EditPerson' | pe
                                <i class="fa fa-times"></i>
                            </button>
                        </td>
                        <td>{{getPhoneTypeAsString(phone.type)}}</td>
                        <td>{{phone.number}}</td>
                    </tr>
                    <tr *ngIf="'Pages.Tenant.PhoneBook.EditPerson' | permission">
                        <td>
                            <button (click)="savePhone()" class="btn btn-sm btn-suc
                                <i class="fa fa-floppy-o"></i>
                            </button>
                        </td>
                        <td>
                            <select name="Type" [(ngModel)]="newPhone.type"class="f
                            </select>
                        </td>
                        <td><input type="text" name="number" [(ngModel)]="newPhone.
                    </tr>
                </tbody>
            </table>
        </div>
    </div>
</div>
```

We added an edit button for each person. Then added a table for each person that shows phones of the related person and allows adding a new phone. Phones table is only shown if we click the edit button.

PhoneBook Component Class

Before changing PhoneBookComponent, we should re-generate service-proxies using nswag as did above. And finally we can change PhoneBookComponent as shown below:

```
import { Component, Injector, OnInit } from '@angular/core';
import { AppComponentBase } from '@shared/common/app-component-base';
import { appModuleAnimation } from '@shared/animations/routerTransition';
import { PersonServiceProxy, PersonListDto, ListResultDtoOfPersonListDto, PhoneInPe

import { remove as _remove } from 'lodash-es';

@Component({
    templateUrl: './phonebook.component.html',
    styleUrls: ['./phonebook.component.less'],
    animations: [appModuleAnimation()]
})
export class PhoneBookComponent extends AppComponentBase implements OnInit {

    people: PersonListDto[] = [];
    filter: string = '';

    editingPerson: PersonListDto = null;
    newPhone: AddPhoneInput = null;

    constructor(
    super(injector);
}

ngOnInit(): void {
    this.getPeople();
}

getPeople(): void {
    this._personService.getPeople(this.filter).subscribe((result) => {
        this.people = result.items;
    });
}

deletePerson(person: PersonListDto): void {
    this.message.confirm(
        this.l('AreYouSureToDeleteThePerson' | localize: person.name),
        isConfirmed => {
            if (isConfirmed) {
                this._personService.deletePerson(person.id).subscribe(() => {
                    this.notify.info(this.l('SuccessfullyDeleted'));
                    _remove(this.people, person);
                });
            }
        }
    );
}

editPerson(person: PersonListDto): void {
    if (person === this.editingPerson) {
        this.editingPerson = null;
    } else {
        this.editingPerson = person;

        this.newPhone = new AddPhoneInput();
        this.newPhone.type = PhoneType.Mobile;
        this.newPhone.personId = person.id;
    }
};

getPhoneTypeAsString(phoneType: PhoneInPersonListDtoType): string {
    switch (phoneType) {
        case PhoneType.Mobile:
            return this.l('PhoneType_Mobile');
```

```
                    return this.l('PhoneType_Business');
                default:
                    return '?';
            }
        };

        deletePhone(phone, person): void {
            this._personService.deletePhone(phone.id).subscribe(() => {
                this.notify.success(this.l('SuccessfullyDeleted'));
                _remove(person.phones, phone);
            });
        };

        savePhone(): void {
            if (!this.newPhone.number) {
                this.message.warn('Please enter a number!');
                return;
            }

            this._personService.addPhone(this.newPhone).subscribe(result => {
                this.editingPerson.phones.push(result);
                this.newPhone.number = '';

                this.notify.success(this.l('SavedSuccessfully'));
            });
        };
    }
```

Next

- Edit Mode For People