

Підвищення ефективності сервісу благодійної допомоги

Магістерська кваліфікаційна робота
студента групи АКІТ-19м
Антоненко О. С.
керівник: к.т.н., доцент Маслій Р.В.

Актуальність

1. У цивілізованих країнах благодійність є нормою життя. Люди вважають абсолютно природним 10% свого часу або грошей спрямувати на суспільно добрі справи: на допомогу або підтримку перспективних проектів.
2. На даний момент існують сайти, які виконують благодійну діяльність, шляхом безпосереднього збору коштів та передання їх особам, що потребують благодійної допомоги.
3. Дана робота реалізує **новий підхід до благодійності**, який полягає в тому, що організація робить замовлення на отримання певних матеріальних цінностей – меблів, одягу, іграшок і т.д., які знаходяться в асортименті сайту, а відвідувачі сайту оплачують дане замовлення.

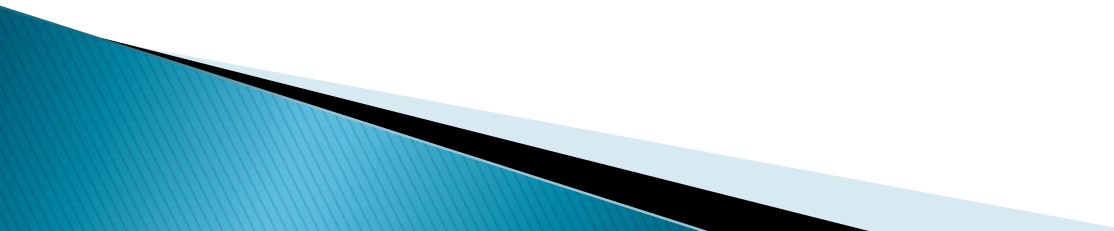
Мета і задачі дослідження

Метою дослідження є підвищення ефективності сервісу послуг благодійної допомоги.

Для досягнення мети потрібно розв'язати низку *задач*:

- ▶ розглянути основні напрями благодійної діяльності;
- ▶ проаналізувати вже існуючі сервіси благодійної допомоги;
- ▶ дослідити технологію створення серверної частини сайту благодійної допомоги на основі платформи Node.js;
- ▶ проаналізувати взаємодію з базою даних MySQL:
- ▶ на основі розглянутих підходів та концепцій створити ефективне програмне забезпечення для сайту благодійної допомоги.

Об'єкт та предмет дослідження

- ▶ *Об'єктом* дослідження є процес підвищення ефективності забезпечення сервісу благодійної допомоги.
 - ▶ *Предметом* дослідження є технології розробки односторінкових веб-застосунків, JavaScript, Node.js та MySQL.
- 

Науково-практична цінність

- ▶ *Науково-практичний результат* полягає у розробці алгоритмічних та програмних засобів для реалізації веб-застосунку благодійної організації у вигляді нативного додатку, який, на відміну від традиційного веб-сайту, відрізняється більшою ефективністю роботи.
- ▶ *Апробація результатів роботи.* Результати досліджень апробовано на XLVIII науково-технічній конференції професорсько-викладацького складу, співробітників та студентів ВНТУ (березень 2019 року).

Концепція односторінкового веб-додатку

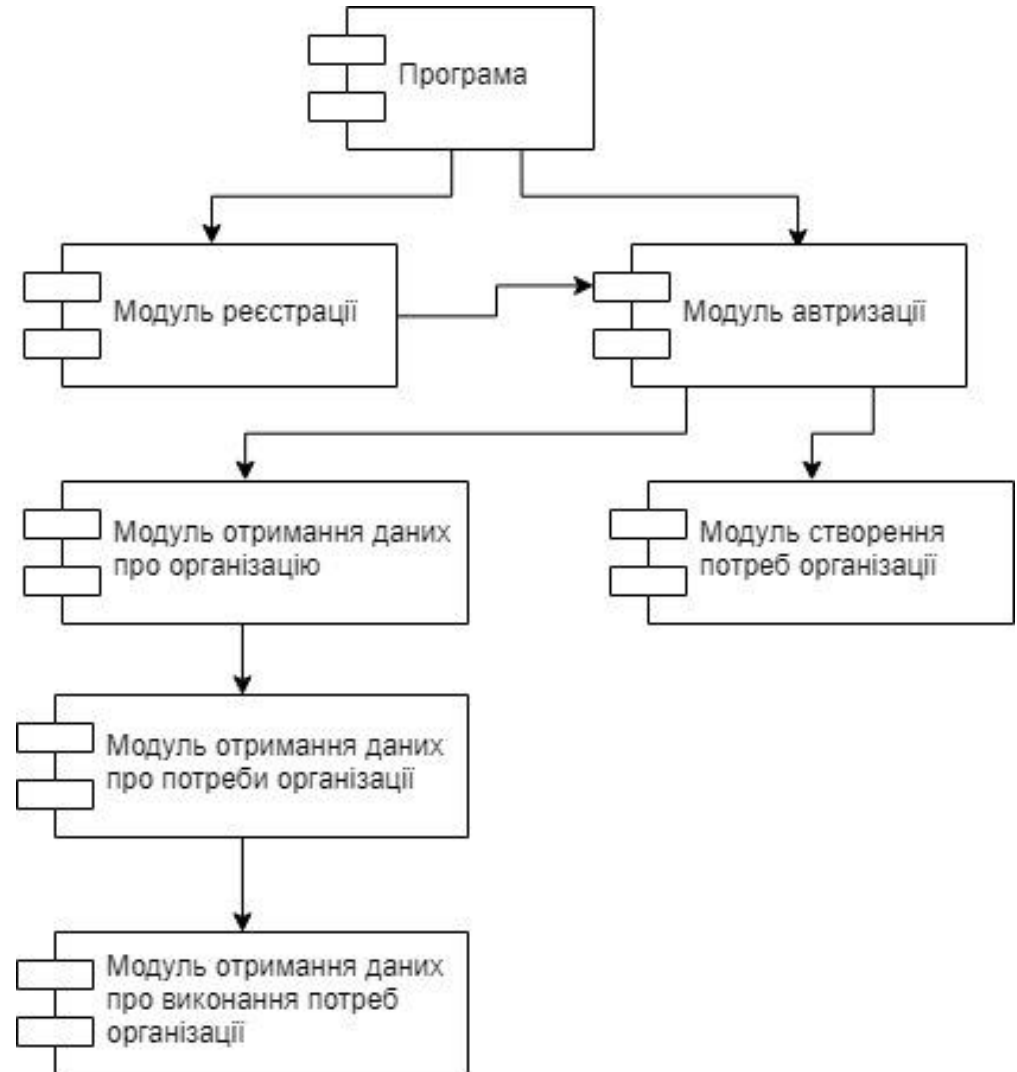
Три основні підходи до розробки веб-застосунків:

1. односторінкові (SPA – від англ. *Single Page Application*);
2. багатосторінкові (MPA – від англ. *Multi Page Application*);
3. прогресивні (PWA – від англ. *Progressive Web Apps*).

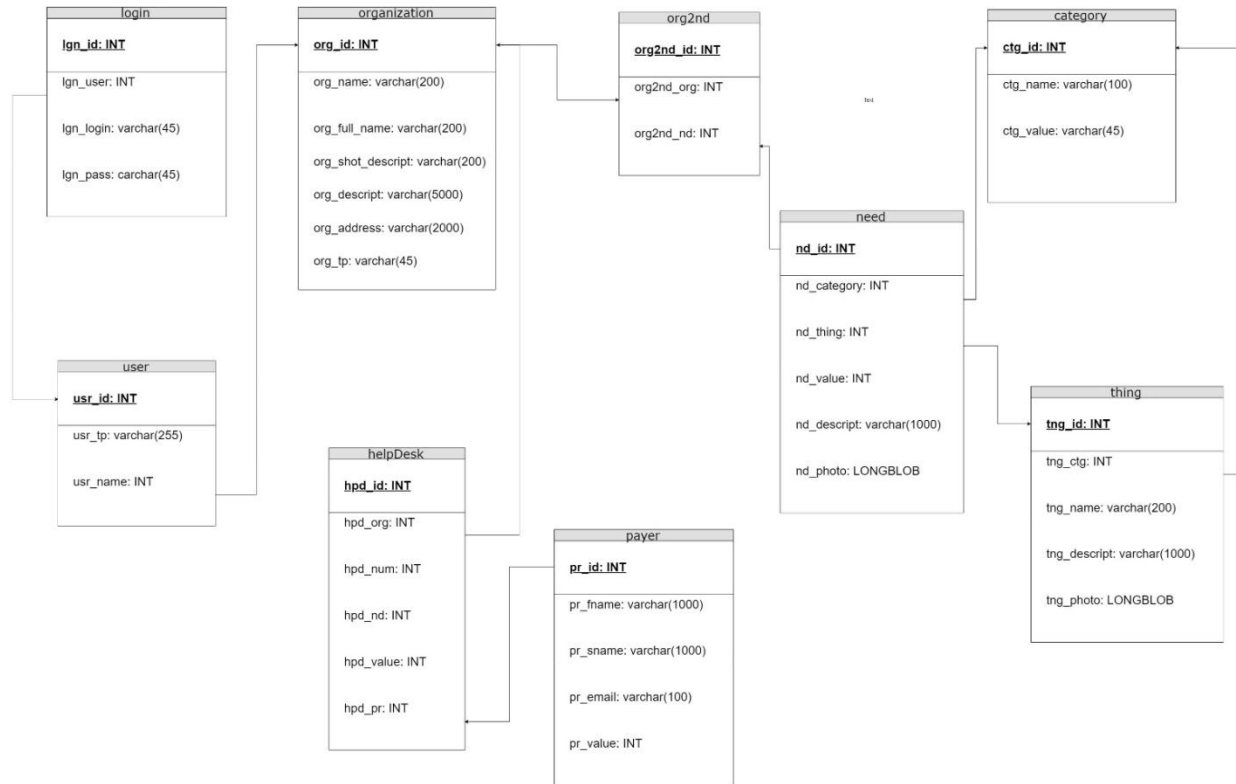
Переваги SPA

- ▶ висока швидкодія – усі ресурси завантажуються за одну сесію, а під час дій на сторінці дані просто міняються, що дуже економить час;
- ▶ гнучкість призначеного для користувача інтерфейсу – за рахунок того, що веб-сторінка всього одна, простіше побудувати насичений інтерфейс, зберігати відомості про сеанс, управляти станами представлень і анімацією;
- ▶ спрощена розробка – код можна починати писати з файлу *file://URL*, не використовуючи сервер, не потрібний окремий код для рендера сторінки на стороні сервера;
- ▶ кешування даних – додаток відправляє всього один запит, збирає дані, а після цього може функціонувати в *offline*-режимі.

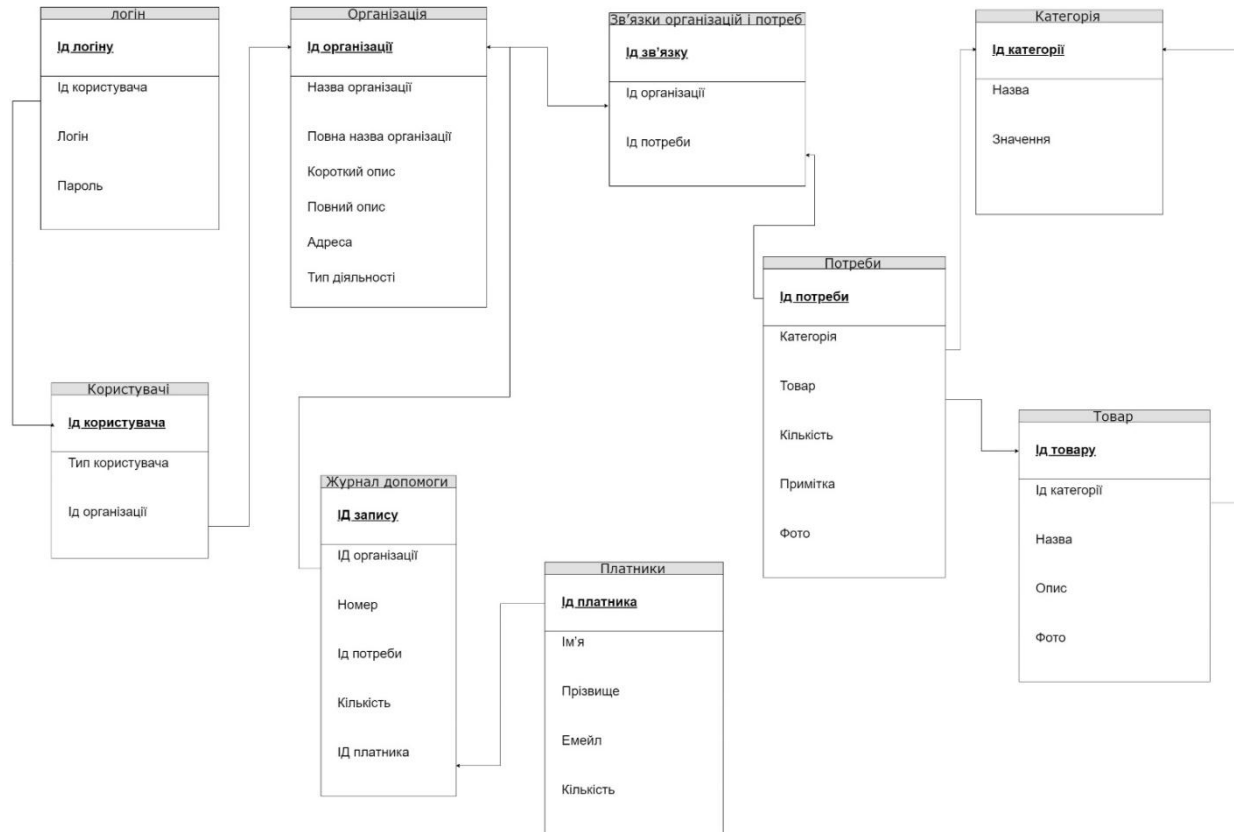
UML діаграма компонентів



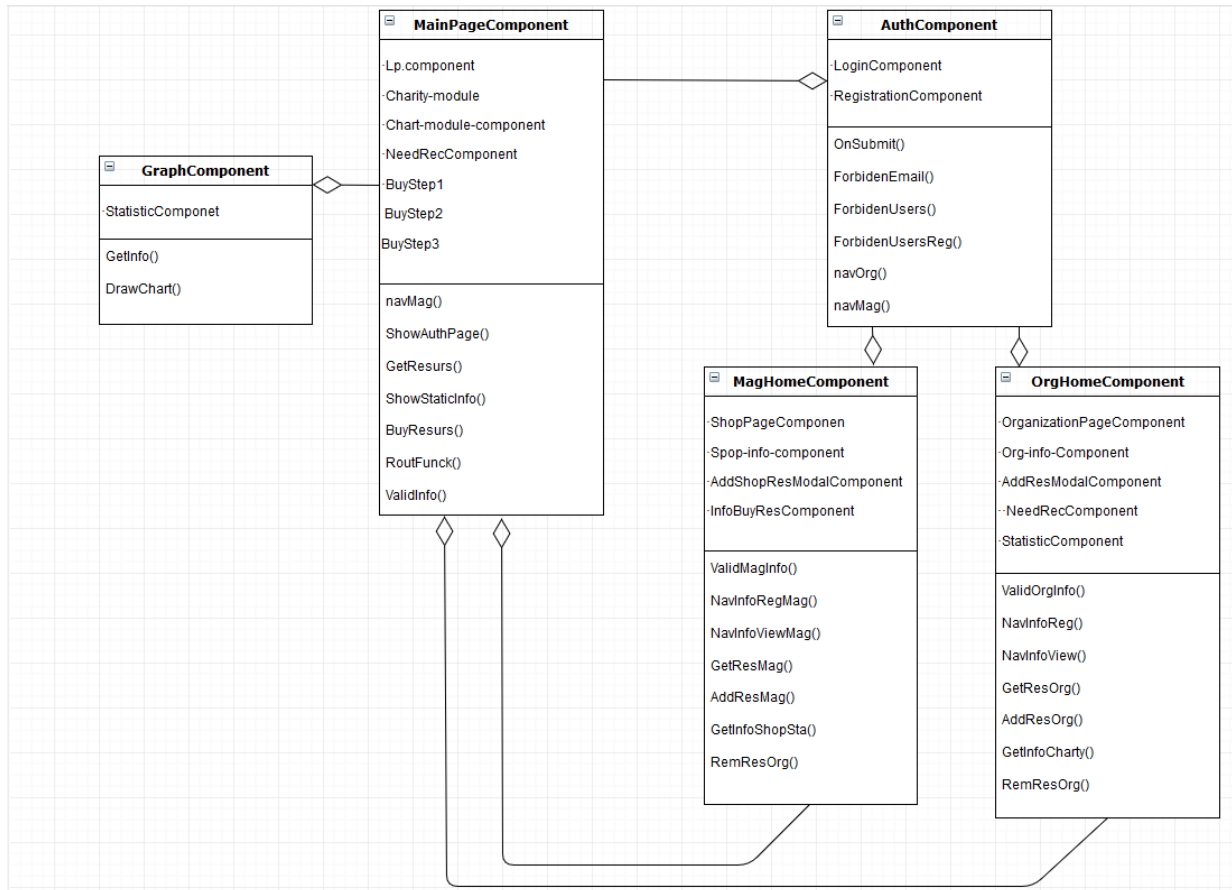
ER модель бази даних (фіз. рівень)



ER модель бази даних (лог. рівень)




UML діаграма класів веб-додатку



Головна сторінка додатку (екран потреб організації)

Наші потреби


Ліжко односпальне
Кількість: 1



Потрібно одне односпальне ліжко для дитячого табору

[Змінити](#) [Видалити](#)


Канцелярські товари
Кількість: 20



Потрібно 20 наборів канцелярських товарів для школи інтернату

[Змінити](#) [Видалити](#)

Покривало
Кількість: 5



У дитячий табір необхідно 5 покривал

[Змінити](#) [Видалити](#)

[Додати](#)

Фрагмент лістингу серверної частини

КОДУ

```
▶ const express = require("express");
▶ const mysql = require("mysql2");
▶ const app = express();
▶ const connection = mysql.createConnection({
▶   host: "localhost",
▶   user: "root",
▶   database: "charity",
▶   password: "Silver_Horn"
▶ });
▶ connection.connect(function(err) {
▶   if (err) {
▶     return console.error("Ошибка: " + err.message);
▶   }
▶   else{
▶     console.log("Подключение к серверу MySQL успешно установлено");
▶   }
▶ });
```

Фрагмент лістингу клієнтської частини

КОДУ

```
> <form [formGroup]="form" (ngSubmit)="onSubmit()">
>   <div class="form-group" [ngClass]="{'hasError':form.get('email').status ==
'INVALID' && form.get('email').touched}">
>     <!-- <label for="email">Email</label>
>     <input
>       type="text"
>       class="form-control underlined"
>       id="email"
>       placeholder="Введіть ваш email"
>       formControlName="email">
>     <span *ngIf="form.get('email').status == 'INVALID' &&
form.get('email').touched">
>       <span *ngIf="form.get('email')['errors']['required']">Поле Email є
обов'язковим</span>
>       <span *ngIf="form.get('email')['errors']['email']"> Email не
коректний</span>
>     </span> -->
>     <mat-form-field class="full-width">
>       <input id="email" type="text" matInput placeholder="Email"
formControlName="email">
>       <mat-error *ngIf="form.get('email').status == 'INVALID' &&
form.get('email').touched && form.get('email')['errors']['required']">
>         Поле Email є обов'язковим
>     </mat-error>
```

Висновки

- ▶ Проведений огляд основних понять, розглянуто організацію роботи з базою даних, проаналізовано можливі альтернативи для створення серверного програмного забезпечення.
- ▶ Проаналізовано переваги технології Node.js, приведені приклади використання готових методів. Досліджена можливість взаємодії з базою даних за допомогою СУБД – MySQL.
- ▶ Створено архітектуру серверної частини сайту благодійної допомоги та представлено UML діаграму системи. Розроблено фізичну та логічну ER модель бази даних. Розроблено програмне забезпечення для сервера сайту благодійної допомоги та веб-додаток.
- ▶ Підвищено ефективність сервісу благодійної допомоги за рахунок використання нового підходу до організації збору коштів та використання переваг SPA (*Single Page Application*) веб-застосувань.

Дякую за увагу