

Національний лісотехнічний університет України  
(повне найменування вищого навчального закладу)

Навчально-науковий інститут деревообробних та  
комп'ютерних технологій і дизайну  
(повне найменування інституту, назва факультету (відділення))

Кафедра інформаційних технологій  
(повна назва кафедри (предметної, циклової комісії))

## **Пояснювальна записка**

до дипломної роботи

другий (магістерський)

(рівень вищої освіти)

на тему: Розроблення інформаційної системи “Агенти крові” для донорів  
крові та волонтерів

---

Виконала: студентка 6 курсу групи КН-61м  
спеціальності

122 “Комп'ютерні науки”

(шифр і назва напрямку підготовки, спеціальності)

Гайда Анна-Марія Сергіївна

(прізвище та ініціали)

Керівник

Процах Н. П.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Національний лісотехнічний університет України  
(повне найменування вищого навчального закладу)

ННІ деревообробних та комп'ютерних технологій і дизайну  
Кафедра інформаційних технологій  
Рівень вищої освіти другий (магістерський)  
Спеціальність 122 "Комп'ютерні науки"  
(шифр і назва)

**ЗАТВЕРДЖУЮ**  
**Завідувач кафедри**

\_\_\_\_\_ Крошній І. М.  
" \_\_\_\_ " \_\_\_\_\_ 20\_\_ року

**З А В Д А Н Н Я**  
**НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Гайда Анна-Марія Сергіївна  
(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення інформаційної системи "Агенти крові" для донорів крові та волонтерів

керівник роботи: Процах Наталія Петрівна, доктор фізико-математичних наук

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "13" грудня 2021 року № С-617

2. Термін подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи \_\_\_\_\_

- Аналіз стану проблемної області \_\_\_\_\_

- Визначення інформаційного та математичного забезпечення \_\_\_\_\_

- Обґрунтування вибору обраних засобів розробки програмного забезпечення \_\_\_\_\_

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

Розділ 1. Стан проблемної області

Розділ 2. Інформаційне та математичне забезпечення

Розділ 3. Програмне та технічне забезпечення

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_

Додаток А.

6. Дата видачі завдання 20.12.2021

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Затвердження теми дипломної роботи	11.11.2021	
2	Обґрунтування технічної доцільності розробки інформаційної системи “Агенти крові”	15.01.2022	
3	Розробка програмного забезпечення для інформаційної системи “Агенти крові”	22.03.2022	
4	Аналіз стану проблемної області на етапі розвитку технологій та розгляд можливих способів вирішення.	1.04.2022	
5	Розбір конструкції та порівням до вже існуючого продукту	15.04.2022	
6	Визначення математичного та інформаційного забезпечення	23.06.2022	
7	Проектування та аналіз бази даних для інформаційної системи “Агенти крові”	07.08.2022	
8	Проведення паралелі між самостійною розробкою та у великій компанії з залученням спектру працівників різних сфер.	10.09.2022	
9	Обґрунтування засобів розробки програмного забезпечення	12.10.2022	
10	Опис інтерфейсу та програмної реалізації	15.11.2022	

Студент

\_\_\_\_\_ ( підпис )

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ ( підпис )

\_\_\_\_\_ (прізвище та ініціали)

## РЕФЕРАТ

Дипломна робота містить 43 сторінки пояснювальної записки, 26 рисунків, 1 додаток, 30 джерел.

В даній роботі розроблено програмне та технічне забезпечення для інформаційної системи “Агенти крові” для донорів крові та волонтерів. Веб-орієнтована система, що створена на базі фреймворку Angular, може використовуватися як особами, що хочуть стати донорами крові, так і жертводавцями чи волонтерами для підтримки проекту.

**Ключові слова:**

JavaScript, HTML, CSS, Java, PHP, MySQL

## ABSTRACT

Thesis contains 43 pages of explanatory note, 26 figures, 1 appendix, 30 sources.

In this thesis, the software and technical support of the "Blood Agents" information system for blood donors and volunteers was developed. The web-oriented system, which is based on the Angular framework, can be used both by people who want to become blood donors, and by victims created or volunteers to support the project.

**Keywords:**

JavaScript, HTML, CSS, Java, PHP, MySQL

## ТЕХНІЧНЕ ЗАВДАННЯ

Тема роботи: **Розроблення інформаційної системи “Агенти крові” для донорів крові та волонтерів.**

Вихідні дані до роботи:

- Аналіз поточного стану технічної проблемної області, включаючи стадію розвитку технології та можливі шляхи вирішення. Також передбачає порівняння структури інформаційної системи з існуючими продуктами.

- Визначення інформаційно-математичного забезпечення інформаційної системи, включаючи проектування та аналіз бази даних і побудову об’єктно-орієнтованої моделі бази даних. Також включає математичне та алгоритмічне забезпечення системи.

- У тексті описується проведення паралелі між самостійною розробкою і великою компанією з різноманітним колом співробітників, які працюють у різних сферах.

- Обґрунтування використання засобів розробки програмного забезпечення та вимоги до технічної підтримки інформаційної системи «Агенти крові». Також наявний опис інтерфейсу та програмної реалізації системи.

## ЗМІСТ

<b>ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ</b> .....	6
<b>ВСТУП</b> .....	7
<b>РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ</b> .....	8
1.1 Стан комп'ютерної інформаційної технології.....	8
1.2 Алгоритм побудови інформаційної системи та порівняння з існуючим продуктом.....	8
1.2.1.1 Алгоритм побудови інформаційної системи .....	8
1.2.1.2 Розгляд існуючої інформаційної системи.....	10
<b>РОЗДІЛ 2. МАТЕМАТИЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ</b> .....	12
2.1. Алгоритм проектування бази даних.....	12
2.2. Побудова об'єктно-орієнтованої моделі .....	15
2.3. Алгоритмічне та математичне забезпечення .....	19
<b>РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ</b> .....	24
3.1. Засоби розробки .....	24
3.2. Вимоги до технічного та програмного забезпечення .....	28
3.2.1. Вимоги до технічного забезпечення .....	28
3.2.2. Вимоги до апаратного забезпечення .....	29
3.2.3. Вимоги до програмного забезпечення .....	29
3.3. Опис програмної реалізації та інтерфейс .....	29
<b>РОЗДІЛ 4. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ</b> .....	38
4.1. Опис ідеї проекту .....	38
4.2. Розроблення ринкової стратегії .....	39
4.3. Вимоги до технічного забезпечення .....	40

<b>ВИСНОВКИ .....</b>	<b>41</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>42</b>
<b>ДОДАТКИ .....</b>	<b>43</b>

## ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

1. DEFAULT 0 raquo - повідомляє серверу, що поле за замовчуванням буде нульовим.
2. unsigned - без від'ємних чисел.
3. tinyint - тип даних, який вимагає 1 байт пам'яті і може зберігати числа від 0 до 255.
4. CRM - програмне забезпечення, яке автоматизує взаємодію з клієнтами для покращення обслуговування клієнтів і збільшення витрат.
5. NET Framework - програмна технологія, розроблена Microsoft для створення веб-додатків і програм.
6. IP - унікальний числовий ідентифікатор пристрою в комп'ютерній мережі.
7. DNS-сервер - спеціалізований комп'ютер або група комп'ютерів, які зберігають IP-адреси веб-сайтів.
8. Chromium – веб-браузер із відкритим кодом, розроблений Google.
9. Blink - безкоштовний механізм візуалізації для відображення веб-сторінок, розроблений Opera Software і Google на основі вихідного коду WebCore з WebKit для браузера Chromium.

## ВСТУП

**Актуальність теми.** Інформаційна система “Агенти крові” для донорів крові та волонтерів є важливою темою так як тисячі людей щодня потребують кров у різних випадках життя. Особливо в Україні кількість потребуючих збільшується на фоні воєнного стану та все більше крові потрібно нашим воєнним та волонтерам. Значним плюсом є те, що будь-хто може допомогти при бажанні, що не потрібно довго шукати куди можна здати кров чи пожертвувати гроші на донорство, адже за допомогою сайту можна знайти хто більше потребує кров та в якому пункті прийому її можна здати. Важливим є те, що пожертви здійснюється через веб-сайт та не потрібно тратити багато часу, щоб зробити це фізично. Конкуренція в даній сфері не є висока тому треба зробити продукт зручним та доступним, щоб людям хотілось надіслати гроші чи допомогти потребуючим, бо люди часто вибирають інший веб-сайт, якщо незрозуміло все побудовано.

**Об’єкт дослідження.** Інформаційна система “Агенти крові” для донорів крові та волонтерів.

**Предмет дослідження.** Інформаційні технології, способи реалізації та програмні рішення для створення інформаційної системи для веб-сайту.

**Мета роботи.** Розробка інформаційної системи за допомогою фреймворку Angular для подальшого використання донорами крові, жертводавцями та волонтерами.

### **Завдання дослідження.**

- Аналіз ключових моментів для розробки інформаційної системи.
- Проектування дизайну інтерфейсу сайту.
- Проектування деревовидної структури сайту.
- Розробка схеми меню програмного забезпечення.
- Алгоритм розробки сторінки для грошових пожертв.
- Аналіз вже існуючих схожих продуктів.

**Практичне значення.** Інформаційна система «Агенти крові» завдяки своїй унікальності та зручності має значне практичне значення, оскільки використовується в благодійних цілях.

## **РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ**

### **1.1 Стан комп'ютерної інформаційної технології**

Інформаційна система — це сукупність організаційних і технічних засобів зберігання й обробки інформації для задоволення інформаційних потреб користувачів, оптимізації й автоматизації процесів. Маркетинг і жива торгівля або надання послуг у режимі реального часу — це поєднання того, за що відповідає значна частина веб-сайтів. Використання веб-сайту на відміну від фізичного походу для реєстрації та запису на прийом - відрізняється тим, що кожен користувач аналізується програмним забезпеченням та має свою власну історію. Однак у живому спілкуванні неможливо тримати в пам'яті інформацію про кожну людину та знати її вподобання, тому може виникнути необхідність фіксувати конкретні моменти в базі даних. Це використовується не часто або вкрай рідко. Інформаційна система також має свої недоліки, такі як неможливість фізично оцінити та приміряти продукт або записатися на прийом до лікаря, якщо немає відповідного. З іншого боку, веб-сайт може надати більше інформації, ніж консультант, оскільки він не завжди може знати про продукт або послугу.

### **1.2 Алгоритм побудови інформаційної системи та порівняння з існуючим продуктом**

#### **1.2.1 Алгоритм побудови інформаційної системи**

Звичайний веб-сайт можна перетворити на потужний і ефективний інструмент лише за допомогою кількох змін. Важливо відзначити, що між веб-сайтами існує велика конкуренція за надання послуг, оскільки користувачів щодня заохочують великою кількістю реклами та різноманітної інформації. Мета веб-сайтів — привернути увагу багатьох користувачів і утримувати їх на певних сторінках,

заохочуючи їх робити пожертви або купувати шляхом обробки їхніх уподобань. Важливо також пам'ятати, що з добре розробленим сайтом, зрозумілими пропозиціями та відповідним дизайном веб-сайт матиме великий успіх і користувачі будуть рекомендувати його своїм друзям та знайомим.

Коли користувач вперше заходить на сайт, він побачить меню зі списком пропозицій і статей, які є в інформаційній системі. Меню на веб-сайті представлено у вигляді деревовидної структури з різними категоріями як елементами. Структура меню дозволяє користувачам швидко орієнтуватися на сайті та знаходити потрібну інформацію. Веб-сайт також повинен мати чіткий і стислий макет, а також просту у використанні навігацію. Це допоможе користувачам швидко знайти те, що вони шукають, і зацікавить їх веб-сайтом (рис. 1.1.).

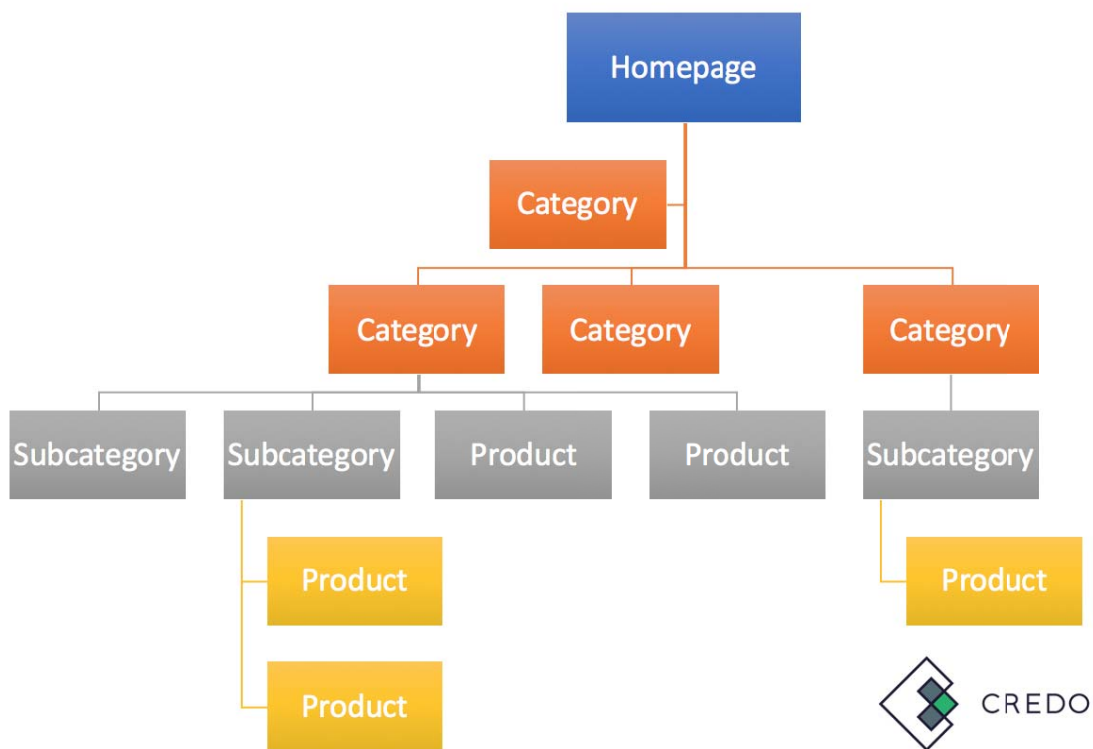


Рисунок 1.1 – Деревоподібна структура

Меню має різні групи. Наприклад при якщо користувач хоче підтримати фонд він це може зробити різними доступними на сайті способами. При першій спробі надіслати кошти чи зареєструватись на здачу крові - потрібно буде зареєструватись, щоб не робити це щоразу.

## 1.2.2 Розгляд існуючої інформаційної системи

Як приклад взято інформаційну систему «donor.ua», який є подібним до програмного забезпечення розробленого в цій роботі та використані такі ж мови програмування (рис. 1.2.).

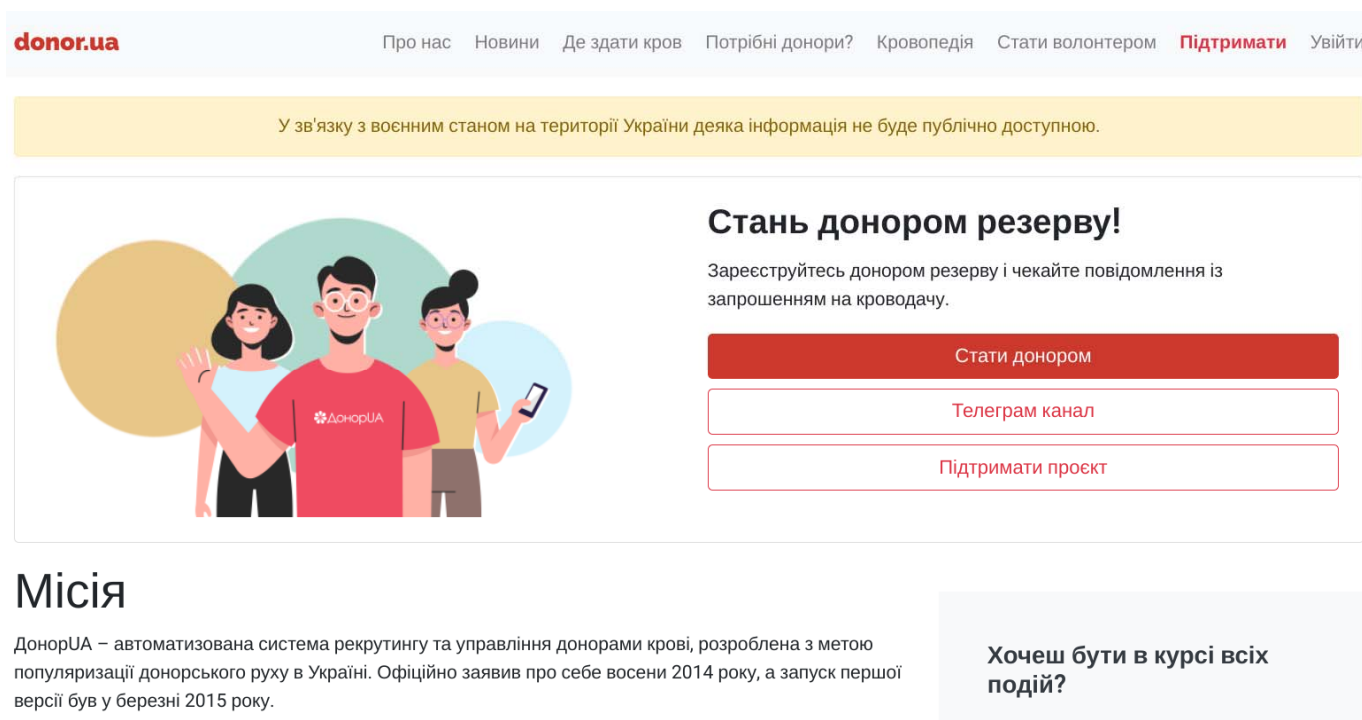


Рисунок 1.2 – Web-сайт «donor.ua»

*Меню.* У меню даного web-сайту є всі потрібні категорії. А саме містять: реєстрація пацієнта, якому потрібна кров, можливість стати волонтером, детальна інформація про web-сайт, можливість підтримати фінансово та записатись на здачу крові.

*Підтримати фінансово.* Коли користувач хоче пожертвувати гроші на благодійність, то його перенаправляє на сторінку з різними реквізитами та для початку необхідно пройти реєстрацію. Після перевірки введених даних потрібно натиснути кнопку “Підтвердити переказ”.

## Підтримати проєкт з картки будь-якого банку

**Назва банку:** АТ КБ "ПРИВАТБАНК"

**IBAN:** UA583052990000026006036215167

**ЄДРПОУ:** 34934620

**Отримувач:** ГО «Всеукраїнська асоціація донорів України»

**Призначення платежу:** Безповоротна фінансова допомога.

або за допомогою [Way for Pay](#), [Liqpay](#), [Monobank](#).

Також ви можете оформити регулярний платіж на [Patreon](#).

## PayPal

support@donor.ua

## ETH

0x14CD339CBD2AabDca1FC8647c7b9D1B50286fFd5

## SWIFT (USD)

**Company Name:** ВАДУ ГО

**IBAN Code:** UA023052990000026000016211875

**Name of the bank:** JSC CB "PRIVATBANK", 1D HRUSHEVSKOHO STR., KYIV, 01001, UKRAINE

**Bank SWIFT-Code:** PBANUA2X

Рисунок 1.3 – Варіанти переказу коштів для підтримки фонду

## РОЗДІЛ 2. МАТЕМАТИЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1. Алгоритм проектування бази даних

Потрібну функціонуючу структуру, що використовується в даному програмному забезпеченні, представлено за такою схемою (рис. 2.1.):

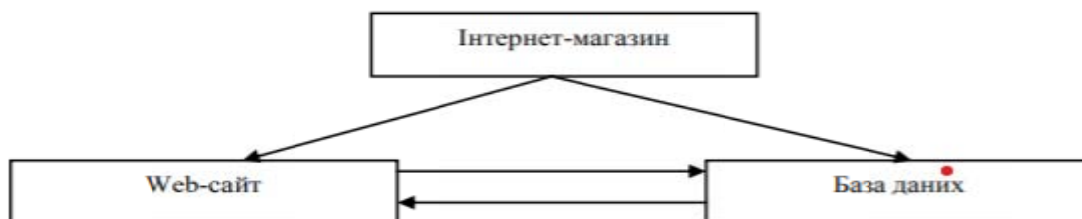


Рисунок 2.1 – Схема інформаційної системи

З часом кількість сторінок на веб-сайті збільшується, а зміст продовжує зростати. Бази даних використовуються для ефективного управління великою кількістю інформації. Вони також оновлюються новими даними. MySQL містить внутрішню серверну бібліотеку.

Даною метою є реалізація правильних типів моделей даних та аналіз предметної області для бази даних, опанування основних етапів проектування бази даних. База даних повинна бути розроблена для обробки різних типів запитів реалізації.

Враховуючи всю вище викладену інформацію про електронну комерцію, створено діючу базу даних користувачів інформаційної системи «Агенти крові» з використанням системи управління реляційними базами даних MySQL та мови програмування Java.

Вся база даних інформаційної системи «Агенти крові» складається з 9 таблиць. Ось список усіх із них:

- Послуги;
- Способи оплати;

- Донори;
- Потребуючі кров;
- Групи крові;
- Міста;
- Адреси лікарень;
- Користувачі;
- Статті;

У структурі бази даних усі таблиці з'єднані зв'язками «один до багатьох». Ця схема з'єднання забезпечує цілісність бази даних, оскільки вона гарантує, що дані правильно організовані та пов'язані один з одним. Крім того, використання зв'язків «один-до-багатьох» у структурі бази даних дозволяє спростити керування даними та оновлення, оскільки зміни, внесені до однієї таблиці, можуть автоматично відображатися у пов'язаних таблицях. Це може допомогти зменшити ризик помилок і невідповідностей у базі даних.(рис. 2.2.).

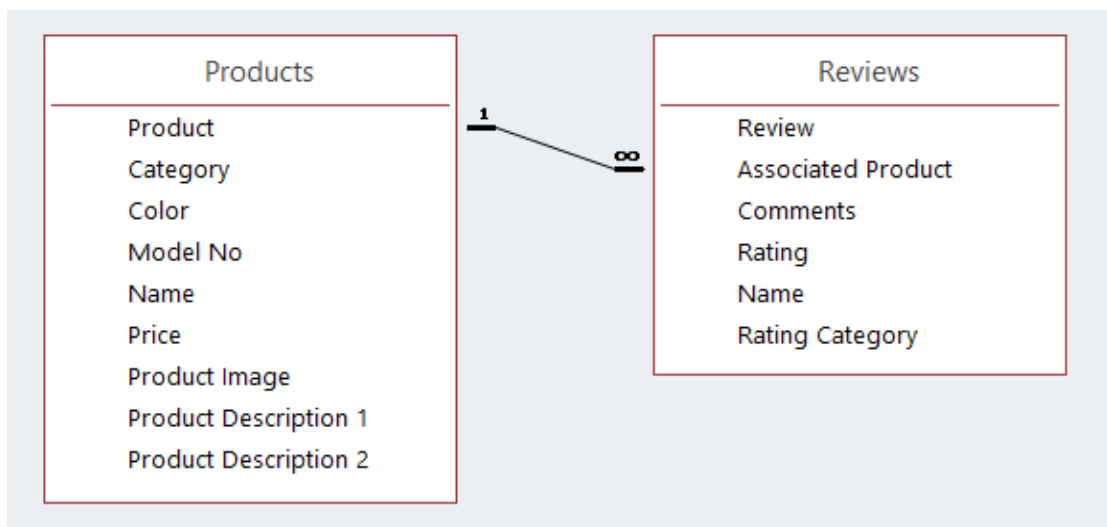


Рисунок 2.2 – Один-до-багатьох

Для цього проекту була обрана мова програмування Java та завдяки відкритому стандарту комунікаційного інтерфейсу баз даних можливе підключення до будь-яких баз даних, які мають встановлений драйвер.

Дизайн бази даних для категорій товарів виглядає наступним чином.

Ось приклад таблиці:

```
CREATE TABLE blood_donors (  
    donor_id INTEGER PRIMARY KEY,  
    name TEXT NOT NULL,  
    gender CHAR(1) NOT NULL,  
    date_of_birth DATE NOT NULL,  
    blood_type CHAR(3) NOT NULL,  
    email VARCHAR(255),  
    phone VARCHAR(20),  
    location VARCHAR(255),  
    availability CHAR(1) NOT NULL,  
    donations INTEGER  
);
```

Цей код створює таблицю під назвою "blood\_donors" із такими полями:

"donor\_id" - це ціле числове поле, яке служить первинним ключем для таблиці.

Первинний ключ — це поле, яке однозначно ідентифікує кожен запис у таблиці.

"name" - це текстове поле, яке не можна залишати порожнім.

"gender" – це поле символів, яке може містити лише один символ і не може залишатися порожнім.

"date\_of\_birth" - це поле дати, яке не можна залишати порожнім.

"blood\_type" - це символічне поле, яке може містити лише три символи та не може залишатися порожнім.

"email" – це символічне поле змінної довжини, яке може містити до 255 символів.

"phone" - це поле символів змінної довжини, яке може містити до 20 символів.

"location" - це символічне поле змінної довжини, яке може містити до 255 символів.

"availability" — це поле символів, яке може містити лише один символ і не може залишатися порожнім.

"donations" є цілим полем.

Основна мета веб-сайту – взаємодія з базою даних. Це необхідно для належного функціонування веб-сайту, оскільки всі послуги, користувачі, меню, способи оплати тощо вимагають використання бази даних і пов'язаних з нею можливостей обробки. Без цього веб-сайт не міг би нормально функціонувати.

## 2.2. Побудова об'єктно-орієнтованої моделі

Щоб побудувати об'єктно-орієнтовану модель, потрібно виконати такі дії:

- Визначити об'єкти: подумати про об'єкти реального світу, які ви хочете відобразити у своїй моделі. Це можуть бути фізичні об'єкти, як-от машини чи книги, або абстрактні поняття, як-от користувачі чи замовлення.

- Визначити атрибути кожного об'єкта: визначити характеристики або властивості кожного об'єкта. Ці атрибути можуть включати такі речі, як колір, розмір, ціна або кількість.

- Визначити поведінку кожного об'єкта: розглянути дії або функції, які може виконувати кожен об'єкт. Ця поведінка може включати такі речі, як водіння, читання або покупки.

- Визначити зв'язки між об'єктами - як об'єкти у вашій моделі пов'язані один з одним. Наприклад, об'єкт автомобіля може мати зв'язок із об'єктом водія, або об'єкт книги може мати зв'язок із об'єктом автора.

- Реалізувати об'єктно-орієнтовану модель: використовуючи мову програмування для представлення об'єктів, атрибутів, поведінки та зв'язків у вашій моделі.

В об'єктно-орієнтованому програмуванні є кілька принципів, які можна дотримуватися, щоб спроектувати та створити модель:

- Інкапсуляція відноситься до ідеї об'єднання даних і методів, які працюють з цими даними в одному блоку або об'єкті. Це допомагає приховати деталі реалізації класу від інших об'єктів, полегшуючи зміну реалізації, не впливаючи на решту системи.

- Спадкування — це механізм, який дозволяє створювати нові класи на основі існуючих класів. Нові класи, які називаються похідними класами,

успадковують атрибути та поведінку від базового класу, який потім можна змінювати або розширювати за потреби.

- Поліморфізм дозволяє використовувати той самий інтерфейс для кількох різних типів об'єктів. Це може зробити ваш код більш гнучким і простим у підтримці, оскільки ви можете написати єдиний метод, який працює з кількома типами об'єктів.

- Абстрагування – це процес зосередження на основних характеристиках об'єкта та ігнорування несуттєвих деталей. Це дозволяє спростити складні системи та зробити їх легшими для розуміння та роботи.

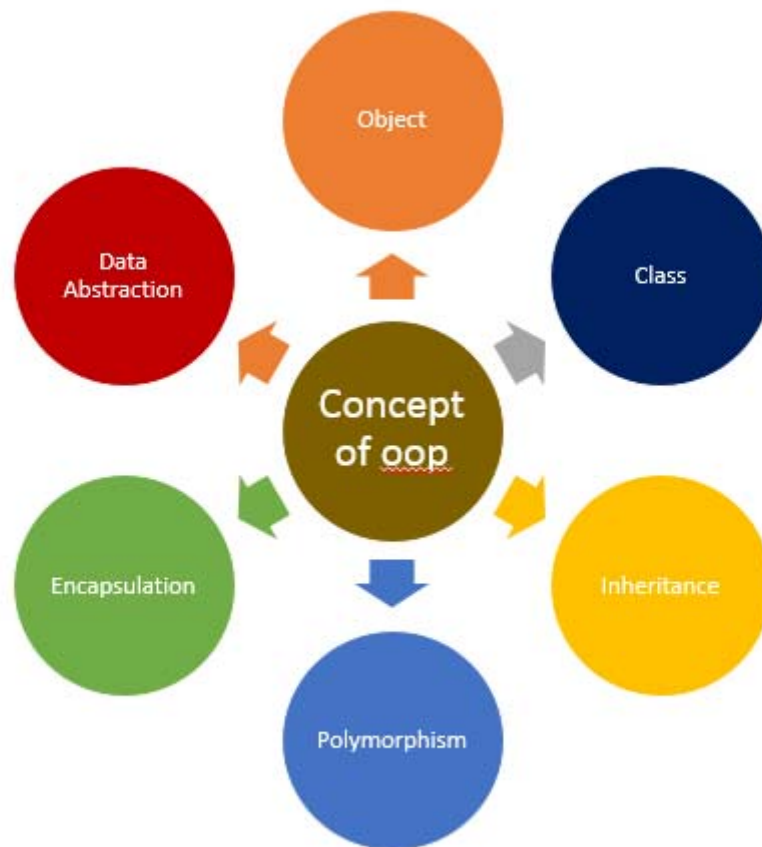


Рисунок 2.3 – Концепції ООП

Моделювання має вирішальне значення при проектуванні систем, а об'єктно-орієнтоване проектування пропонує широкий спектр моделей на вибір. Ці моделі не тільки відтворюють ієрархію об'єктів, а й ієрархію класів. Використання об'єктно-орієнтованого підходу має багато переваг, найважливіша з яких полягає в тому, що

він дозволяє створювати системи, які демонструють п'ять характеристик правильно структурованих систем.

Одним із ключових аспектів об'єктно-орієнтованого проектування є методи розподілення пам'яті під час виконання, такі як:

- визначення типів змінних;
- обґрунтування визначених користувачем змінних;
- створення нових елементів керування або екземплярів форми та використання структур даних, такі як масиви, черги, стеки, списки та дерева;
- використання різних типів списків, наприклад двосторонні черги чи стеки;

Стек - це саме набір даних об'єктів, з якого спочатку здійснюється доступ до останнього записаного об'єкта, і може бути реалізований за допомогою масивів, об'єднань, скалярних змінних та інших структур даних. Операції зі стеком включають перевірку, чи стек порожній, додавання та видалення елементів із кінця стека та доступ до останнього елемента в стеку, а черга працює протилежним чином

Є тільки два види об'єктно-орієнтованих баз даних:

- мультимедійна база даних, яка зберігає мультимедійні дані, такі як зображення, які не можуть бути в реляційній базі даних.
- гіпертекстова база даних, яка дозволяє зв'язувати будь-які об'єкти, допомагаючи в організації різних даних.

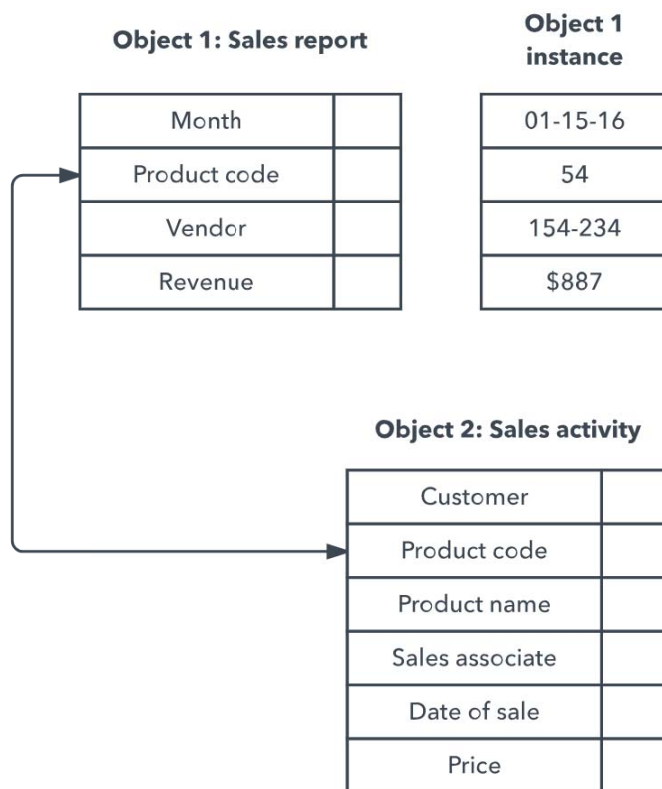


Рисунок 2.4 – Об’єктно-орієнтована модель бази даних

Об’єктно-орієнтована модель бази даних — це модель бази даних, яка представляє дані як об’єкти, а не як прості записи. В об’єктно-орієнтованій базі даних кожен об’єкт є самодостатньою сутністю, яка містить як дані, так і методи, які працюють з цими даними.

Цей підхід до проектування баз даних базується на принципах об’єктно-орієнтованого програмування, яке є парадигмою програмування, зосередженою навколо поняття «об’єкти». В об’єктно-орієнтованому програмуванні об’єкт - це структура даних, яка містить як дані, так і методи, які працюють з цими даними.

В об’єктно-орієнтованій базі даних об’єкти зберігаються в таблицях у базі даних, ними можна керувати та запитувати їх за допомогою методів об’єктно-орієнтованого програмування, таких як успадкування та поліморфізм.

Об'єктно-орієнтовані бази даних безпосередньо обробляють дані як цілі об'єкти, представляючи всю інформацію в одному, миттєво доступному пакеті об'єктів замість кількох таблиць. Навпаки, реляційні бази даних, такі як MySQL, використовують таблиці як основні будівельні блоки, а дії базуються на логічних зв'язках між даними в таблицях.

### 2.3. Алгоритмічне та математичне забезпечення

Інтернет-маркетинг, який передбачає аналіз всього ринку, є ефективним і важливим аспектом управління діяльністю бізнесу. Він спрямований на прогнозування та вивчення фінансування, попиту, ціни, реклами та інших факторів. Електронна комерція поєднує в собі широкий спектр бізнес-операцій і відома своєю універсальністю. (рис. 2.5.).

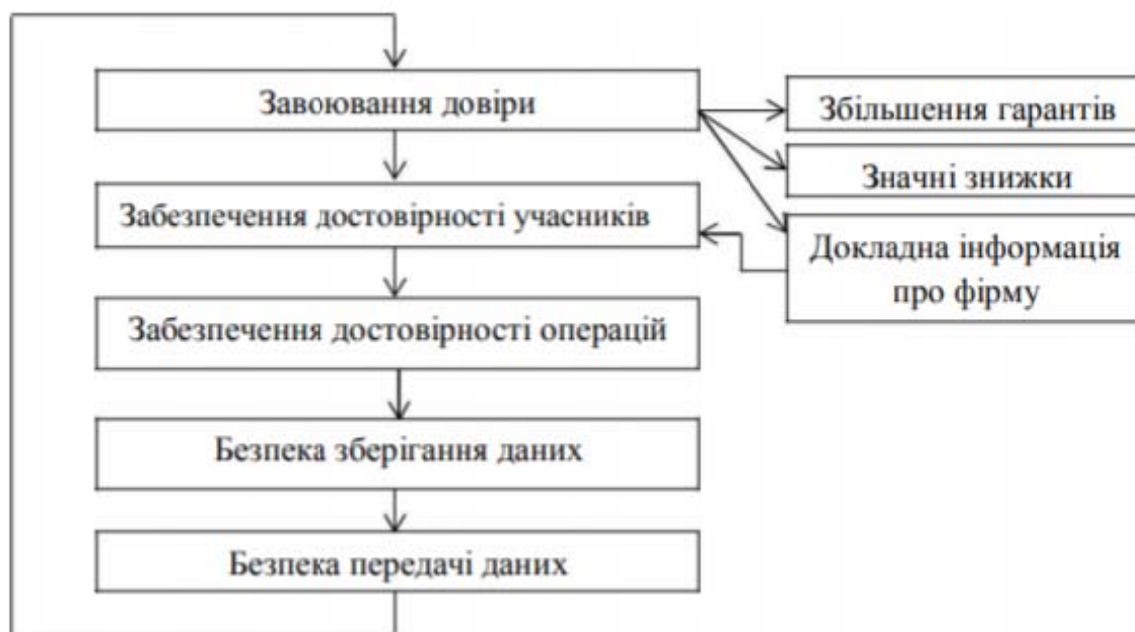


Рисунок 2.5 – Підхід створення інформаційної системи

Веб-сайт, що пропонує послуги, має багато переваг. Це ефективний спосіб заощадити час і вибрати послуги онлайн. Розробка високоякісного програмного забезпечення є вирішальним аспектом цього, оскільки передбачає не лише написання коду, але й оптимізацію модулів у компоненти для легкого додавання

послуг чи товарів у меню, надсилання коштів на благодійність через різні платіжні системи та надання користувачам доступні, прості та зручні у використанні послуги. Основна мета веб-сайту – підвищити залученість користувачів та участь у благодійній діяльності (рис. 2.6.).



Рисунок 2.6 – Алгоритм функціонування web-сайту

Щоб підвищити довіру до компанії та її послуг, необхідно створити розділи, що забезпечують зворотній зв'язок щодо послуг та достовірності коштів, перерахованих на благодійність, а також функцію детального пошуку.

Інформаційна система містить такі аспекти:

- складна пошукова система;
- фінансова підтримка;
- реєстрація для користувачів;
- реєстрація пацієнтів, які потребують кров;
- реєстрація донорів;
- реєстрація волонтерів;

*Пошукова система.* Пошукова система — це комплексна структура даних, яка містить багато розділів і підрозділів. Це надзвичайно важливо, адже якщо користувач не зможе знайти в меню те, що шукає, він покине сайт. Тому надзвичайно важливо якомога точніше структурувати пошукову систему, щоб уникнути подібних випадків.

*Фінансова підтримка.* Якщо користувач хоче підтримати фонд фінансово, то обов'язково потрібно прив'язати різні можливості надсилання коштів, щоб користувач вибрав той, який йому найбільш зручно використовувати.

*Реєстраційна форма.* Для запису даних користувача використовується форма. Після збереження цих даних вони будуть автоматично витягнуті в майбутньому, тож клієнту не доведеться щоразу вводити свою інформацію. Дані можуть зберігатися як на стороні клієнта, так і на стороні сервера.

*Реєстрація пацієнтів, які потребують кров.* Дана реєстрація дозволяє зареєструватися пацієнтам чи контактним особам, яким потрібна допомога, де вони вказують місто, групу крові, хворобу та кількість потрібної крові.

*Реєстрація донорів.* За допомогою чат боту в телеграмі можна зареєструватися на здачу крові і буде вказане зручне для вас місце, де знаходяться наші волонтери яким ви здаєте кров.

*Реєстрація волонтерів.* Також за допомогою чат боту в телеграмі можна стати волонтером у різних сферах (програміст, лікар, медсестра, маркетинговий спеціаліст, дизайнер).

Алгоритм створення web-сайту (рис. 2.7.):

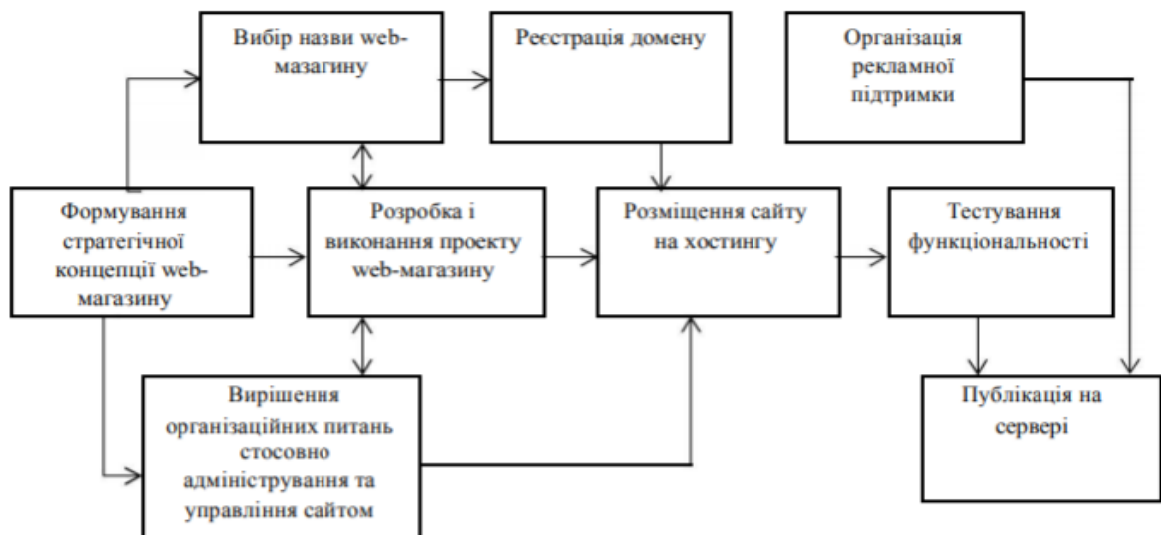


Рисунок 2.7 – Схема створення веб-сайту

Спочатку створення сайту з сервісами є формування стратегічної концепції, яка передбачає визначення цільової аудиторії та ринку. Це має містити повний опис послуг, зображення та дизайн сайту.

Наступним кроком буде вибір імені сайту, яке легко запам'ятати, і реєстрація домену. Існує широкий спектр доступних доменів, тому важливо враховувати різні фактори, наприклад бюджет веб-сайту, оскільки підтримка доменів має різні цінові категорії. Також потрібно врахувати обмеження території обслуговування та відповідно вибрати домен.

Третій етап - розробка технічної документації на обрану концепцію з урахуванням усіх необхідних вимог до сайту з сервісами.

Четвертий етап - це управління сайтом, яке передбачає:

- можливість додавати та редагувати сервіси;
- бухгалтерський, логістичний та економічний аналіз веб-сайту;
- аналіз переходу користувачів та обробку даних;
- реєстрація доменного імені;
- адміністрування;
- публікація сайту на сервері ;
- програмного забезпечення;

- хостинг сайту;
- тестування розробленого сайту;
- забезпечення рекламною підтримкою.

П'ятий етап - вибір місця розташування сайту, який можна здійснити шляхом створення віртуального сервера з використанням хостингу на обладнанні провайдера або шляхом підписки на сервер у власній комп'ютерній мережі.

Наступний етап - тестування всіх етапів розробки на наявність помилок і забезпечення зручності використання інтерфейсу. Тестування проводиться як розробниками, так і клієнтами, щоб уникнути будь-яких проблем.

Останній етап - це розміщення сайту на фізичний сервер або віртуальний сервер у хмарі. Окрім розміщення сайту, важливо мати спосіб керування ним, що часто здійснюється за допомогою системи керування вмістом або системи керування взаємовідносинами з клієнтами. Ці інструменти дозволяють користувачам легко додавати, редагувати та видаляти вміст на сайті або керувати взаємодією донорів і даними. Останнім кроком у процесі розробки веб-сайту є просування сайту та заохочення людей до його використання. Це можна зробити за допомогою різноманітних методів, включаючи пошукову оптимізацію, маркетинг у соціальних мережах або традиційні форми реклами, такі як друкована чи телевізійна реклама. Одним із нових способів просування веб-сайту донорів крові, який набув популярності в останні роки, є маркетинг електронною поштою, за допомогою якого організація надсилає інформаційні бюлетені або сповіщення своїм передплатникам про можливості та події здачі крові.

На основі аналізу ми дійшли висновку, що мови програмування Java та JavaScript є найкращим вибором для цього сайту. Ці мови зазвичай використовуються у веб-додатках через їх практичність, ефективність і гнучкість.

## РОЗДІЛ 3. ТЕХНІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

### 3.1. Опис засобів розробки програмного забезпечення

Є два способи створення програмного забезпечення:

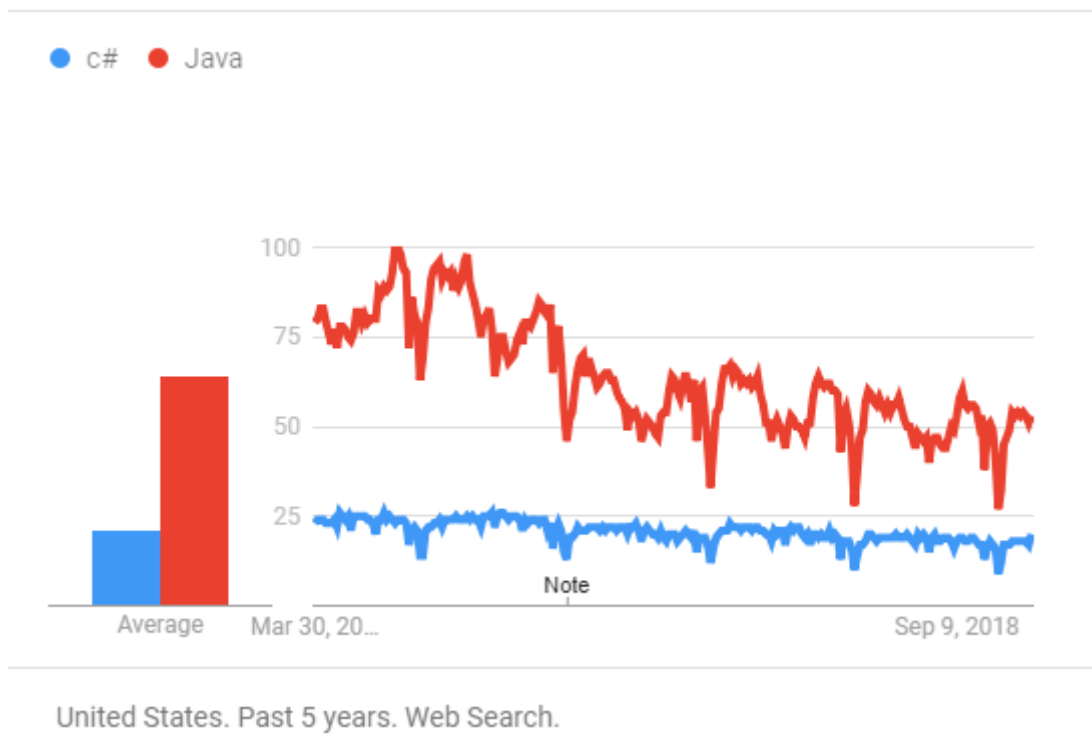
- розробка інформаційної системи самостійно;
- використання різноманітних сайтів для допомоги.

Розробка програмного забезпечення для надання послуг, таких як пожертвування та благодійність, не є складною та не має великої функціональності, як інтернет-магазин, і не обов'язково мати підтримку CRM для адміністрування. Досить надати адміністратору доступ до сайту.

Найпопулярнішою мовою серед веб-сайтів для розробки програмного забезпечення на стороні сервера є Java. Більше половини веб-сайтів електронної комерції написані цією мовою, а приблизно 30-40% європейських магазинів написані на Java, оскільки вона проста та безкоштовна.

Решта сайтів реалізовано за технологіями Microsoft, зокрема .NET Framework. Код програми написаний на мовах C# або VB.NET. PHP також широко використовується, оскільки це безкоштовна мова сценаріїв, яка працює на веб-сервері. Він підходить для розробки невеликих веб-сайтів і програм.

Після вибору мови необхідно визначитися з дизайном бази даних і структурою сайту. База даних може бути як реляційною, так і не реляційною. Реляційна база даних використовує таблиці для зберігання даних і встановлення зв'язків між ними, тоді як нереляційна база даних використовує інші структури даних, такі як документи, пари ключ-значення або графіки (рис. 3.1.).



United States. Past 5 years. Web Search.

Рисунок 3.1 – Розробка на мовах C# та Java

Сайт «Агенти крові» розроблений на Java з підтримкою JavaScript, HTML5 і CSS.

База даних необхідна для належного функціонування веб-сайту. У більшості випадків MySQL використовується з Java для створення, додавання, видалення та редагування даних. Продукти Microsoft можна безкоштовно використовувати для веб-сайтів, але вони мають обмеження щодо розміру бази даних.

Клієнтська частина сайту, яка є інтерфейсом для користувача, створюється за допомогою HTML для структури, JavaScript для функціональності та CSS для стилю. Ці технології працюють разом, щоб забезпечити зручний і візуально привабливий досвід для веб-сайту (рис. 3.2.).



Рисунок 3.2 – Мови програмування для розробки web-сайтів

На додаток до використання чистого JavaScript, веб-сайт використовує Angular Framework для полегшення взаємодії між HTML і JavaScript. Angular полегшує обробку даних і робить їх більш доступними. Мультимедійну платформу Flash також можна використовувати для банерів і візуальних ефектів.

Наведена нижче діаграма ілюструє взаємодію між стороною клієнта, базою даних і кодом сервера (рис. 3.3.).

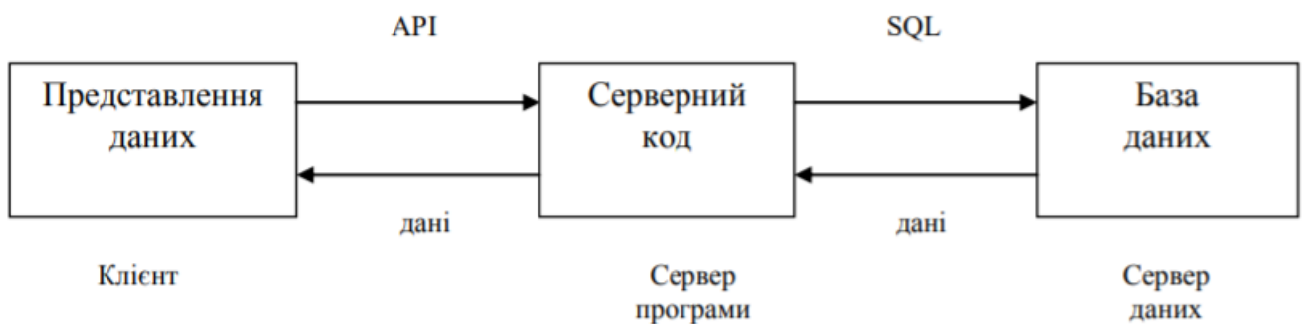


Рисунок 3.3 – Зв'язок між клієнтом та сервером даних.

Розробка програмного продукту для веб-сайту, який надає такі послуги, як пожертвування та благодійність, — складний процес, який потребує команди спеціалістів. Сюди входять розробники FrontEnd, які працюють на стороні клієнта, розробники BackEnd, які обробляють код сервера та базу даних, менеджер проекту

для контролю за бюджетом і процесами, бізнес-аналітики для спілкування з клієнтами та окреслення завдань для розробників, а також тестувальники, які гарантують, що завдання виконані розробниками, функціонують належним чином.

Процес розробки починається з розробки дизайну сайту, обговорення з замовником бюджету та функціоналу, поділу сайту на основні розділи.

Потім створюється база даних для зберігання даних про продукти, клієнтів та іншої відповідної інформації. Для обробки даних база даних підключена до мови Java. Паралельно проект реалізується на стороні клієнта. Після розробки серверної та клієнтської сторони їх об'єднують і перевіряють, щоб переконатися, що вони функціонують належним чином. Для запуску сайту потрібен сервер і зареєстроване доменне ім'я. Веб-сайт розміщується на обладнанні провайдера або на орендованому сервері в центрі обробки даних, залежно від розміру сайту та обсягу трафіку, який він отримує.

Існує багато типів програмного забезпечення, яке можна використовувати для розробки та керування веб-сайтами. Деякі з найпоширеніших типів програмного забезпечення, що використовується для цієї мети, включають:

- Веб-браузери: веб-браузери – це програми, які дозволяють користувачам отримувати доступ до веб-сайтів і переглядати їх. Приклади популярних веб-браузерів включають Google Chrome, Mozilla Firefox і Microsoft Edge.
- Веб-сервери: веб-сервери — це програми, які зберігають і надають користувачам файли веб-сайтів за запитом. Прикладами популярних веб-серверів є Apache і Microsoft IIS.
- Системи керування вмістом (CMS): CMS — це програмне забезпечення, яке дозволяє користувачам легко створювати, редагувати та публікувати вміст на веб-сайті. Приклади популярних CMS включають WordPress, Joomla та Drupal.
- Системи управління взаємовідносинами з клієнтами (CRM): системи CRM – це програмне забезпечення, яке допомагає підприємствам керувати взаємодією з клієнтами та даними. Приклади популярних систем CRM включають Salesforce, Zoho CRM і HubSpot.

- Інструменти розробки: Інструменти розробки – це програми, які використовуються розробниками для створення та підтримки веб-сайтів. Приклади популярних інструментів розробки включають редактори коду, такі як Sublime Text, і інтегровані середовища розробки (IDE), такі як Visual Studio.
- Програмне забезпечення для графіки та дизайну: програмне забезпечення для графіки та дизайну – це програми, які використовуються для створення та редагування графіки, зображень та інших візуальних елементів для веб-сайтів. Прикладами популярних графічних і дизайнерських програм є Adobe Photoshop, Illustrator і InDesign.

## **3.2. Вимоги до технічного та програмного забезпечення**

### **3.2.1. Вимоги до технічного забезпечення**

Етапи розробки веб-сайту для надання послуг зазвичай включають наступні кроки:

- Планування: на цьому етапі вам потрібно буде визначити мету та цілі вашого веб-сайту, а також вашу цільову аудиторію та послуги, які ви будете пропонувати.
- Дизайн: на цьому етапі вам потрібно буде створити дизайн для свого веб-сайту, включаючи макет, колірну схему та загальний вигляд.
- Розробка: на цьому етапі вам потрібно буде створити фактичний веб-сайт за допомогою різноманітних інструментів і технологій, таких як HTML, CSS і JavaScript.
- Тестування: на цьому етапі вам потрібно протестувати веб-сайт, щоб переконатися, що він працює належним чином і відповідає вашим вимогам.
- Розгортання: на цьому етапі вам потрібно буде розмістити веб-сайт на сервері, щоб до нього могли отримати доступ користувачі в Інтернеті.
- Технічне обслуговування: на цьому етапі вам потрібно буде підтримувати веб-сайт в актуальному стані, виправляти будь-які проблеми, що виникають, і переконатися, що веб-сайт працює оптимально.

- Маркетинг: на цьому етапі вам потрібно буде просувати веб-сайт і залучати до нього відвідувачів, використовуючи такі методи, як пошукова оптимізація (SEO), реклама з оплатою за клік (PPC), маркетинг у соціальних мережах або традиційні форми реклами.

### **3.2.2 Вимоги до апаратного забезпечення**

Програмне забезпечення розробляється на мові програмування Java. Ця мова була обрана в результаті ретельного пошуку та аналізу. Ця мова має ряд переваг, таких як:

- об'єктно-орієнтований підхід до програмування;
- висока швидкість і ефективність веб-сайту;
- доступність і простота використання веб-сайту.

### **3.2.3 Вимоги до обраної інформаційної системи**

Щоб створити веб-сайт донорів крові, знадобляться різноманітні програмні засоби та технології. Деякі вимоги до програмного забезпечення для створення веб-сайту донорів крові можуть включати: веб-браузер, веб-сервер, система керування вмістом, система управління взаємовідносинами з клієнтами, інструменти розробки, програмне забезпечення для графіки та дизайну.

Важливо також враховувати технічні вимоги до розміщення та роботи веб-сайту, наприклад необхідне обладнання, операційну систему та програмне забезпечення для баз даних.

## **3.3 Реалізація та інтерфейс інформаційної системи**

На головній сторінці представлено меню, зупинимо кров (збір грошей на турнікети і захисти, щоб запобігти появі кровотечі), новини, проекти, команду,

вказівки як стати донором, корпоративне донорство, збір коштів та магазин. Внизу є новини, статті та досягнення (рис. 3.5).

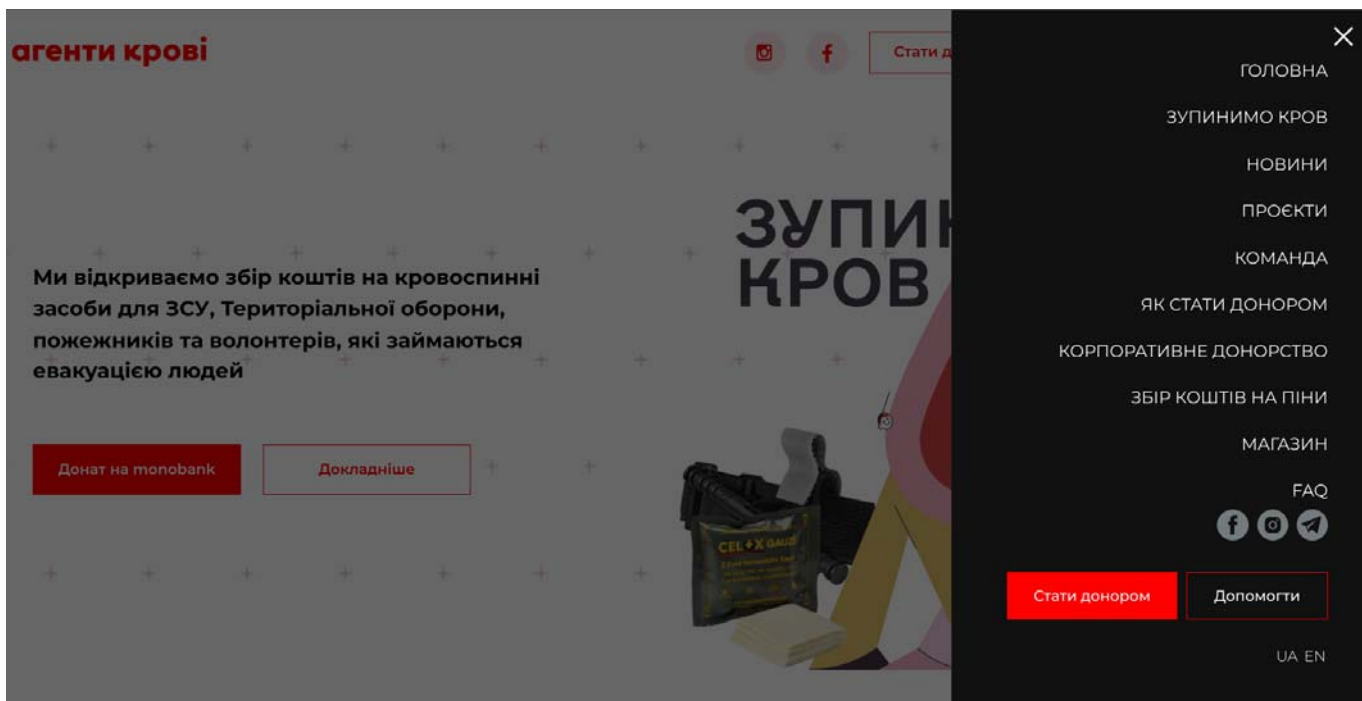
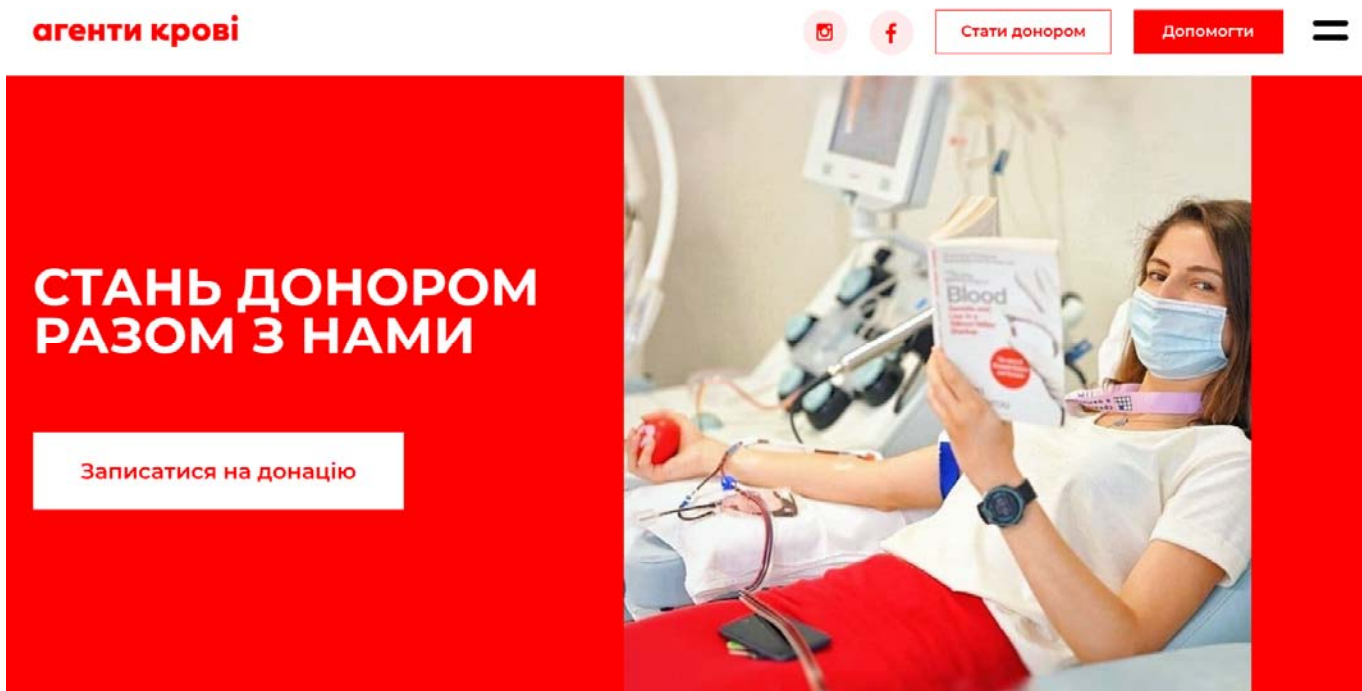


Рисунок 3.5 – Головна сторінка інформаційної системи «Агенти крові»

Розроблений сайт має розділ Стати донором, у котрому є перехід у чат бот телеграму, який зареєструє вас у зручне для вас найближче місце примому крові та запише на певну годину (рис. 3.8.):



### Рисунок 3.8 – Розділ “Стати донором”

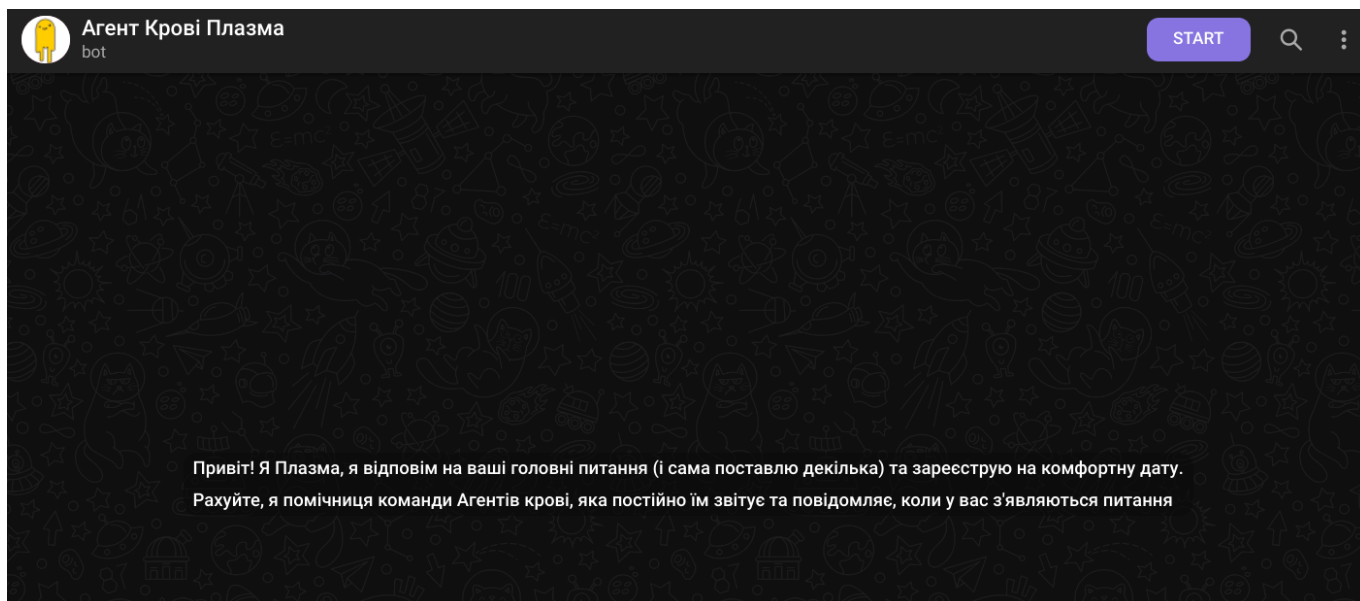


Рисунок 3.9 – Чат бот телеграму

На web-сайті можна підтримати фонд через піни (), разову допомогу (виріб суми з запропонованих) та за реквізитами (представлені різні банки та зручні для користувачів онлайн банкінги). Ось представлено сторінку з варіаціями підтримки фонду (рис. 3.10.).

# Підтримати фонд

Разом ми завжди можемо більше

Збираємо на піни      Разова допомога      Реквізити

**Разова допомога**

50 грн    100 грн    200 грн    500 грн    інша сума

Рисунок 3.10 – Приклад варіацій підтримки фонду

Якщо є потреба у кровоспинних засобах, то можна залишити заявку, щоб отримати запитувану продукцію. Якщо є змога оплатити товар, то він прийде до швидко, якщо є можливість лиш подати запит, щоб отримати товар без оплати, то потрібно буде почекати певний час, щоб жертводавці оплатили його за вас. (рис. 3.11.).

Зупинимо кров - форма запиту

anya.gaida@gmail.com [Switch account](#)

\* Required

Email \*

Your email

Повне ім'я (ПІБ)

Your answer

Номер телефону

Your answer

Профіль в телеграм

Your answer

Група захисників, для якої збираєте:

- волонтери
- ТРО
- пожежники

Рисунок 3.12 – Форма запиту, щоб отримати потрібні товари

Якщо користувач зайшов на web-сайт, як реципієт чи людина відповідальна за нього, то є можливість ввести свої дані, зображення при потребі, хворобу, центр крові, місце лікування, кількість донорів та групу крові. Та вказати які компоненти крові потрібні (рис. 3.12.).

## Інформація про реципієнта

Виберіть зображення

Фото не є обов'язковим, однак значно збільшує кількість відгуків донорів.

Прізвище	Виберіть групу крові
Ім'я	Вкажіть хворобу
По батькові	Вкажіть центр крові
Дата народження	Місце лікування
Вкажіть, які компоненти крові вам потрібні.	
<input type="checkbox"/> Гранулоцити <input type="checkbox"/> Кров (еритроцитна маса) <input type="checkbox"/> Плазма	
<input type="checkbox"/> Тромбоцити	
Необхідна кількість донорів	

## Контактна особа

ПІБ, хто ви для реципієнта	
Телефон	Електронна адреса

Рисунок 3.12 – Інформація про реципієнта

Також на сайті представлений магазин з товарами, при покупці яких, значна частина коштів підує на підтримку фонду (рис. 3.13.).

**агенти крові**



## Магазин

Весь прибуток з продажу мерчу йде на реалізацію проєктів з розвитку донорства крові

Не хочеш купувати? Просто підтримай Агентав!

Допомогти



Рисунок 3.13 – Магазин для підтримки фонду

Ось зображені приклади товарів, які можна посортувати по таких категоріях:

- всі;
- постери;
- футболки;
- популярні товари;

ALL ПОСТЕРИ ФУТБОЛКИ ПОПУЛЯРНІ ТОВАРИ



Каталог Artdonation 2021  
Каталог постерів-учасників конкурсу  
присвяченого Всесвітньому дню донора  
2021

**450 грн.**



Футболка "Краплина" (Іл Сергія Майдукова)

**790 грн.**



Футболка "Дракула на шляху Будди" (Іл Анни  
Богданової)

**790 грн.**

Рисунок 3.14 – Приклади товарів з меню

На web-сайті також представлено список волонтерів та можливість долучитись до команди через чат бот телеграму (рис. 3.15.).

### Волонтери проєкту



**Олена Балбек**

Засновниця проєкту,  
амбасадорка донорства  
крові



**Свєнєія  
В'язовченко**

ІТ-фахівчиня



**Іванка Коваль**

Арбітражна керуюча,  
юристка



**Дмитро  
Охрименко**

Логіст

Рисунок 3.15 – Волонтери проєкту

Також волонтери організують День донора у компанії до яких може долучитись велика кількість працівників та волонтери приїдуть до вас в офіс (рис. 3.16.).

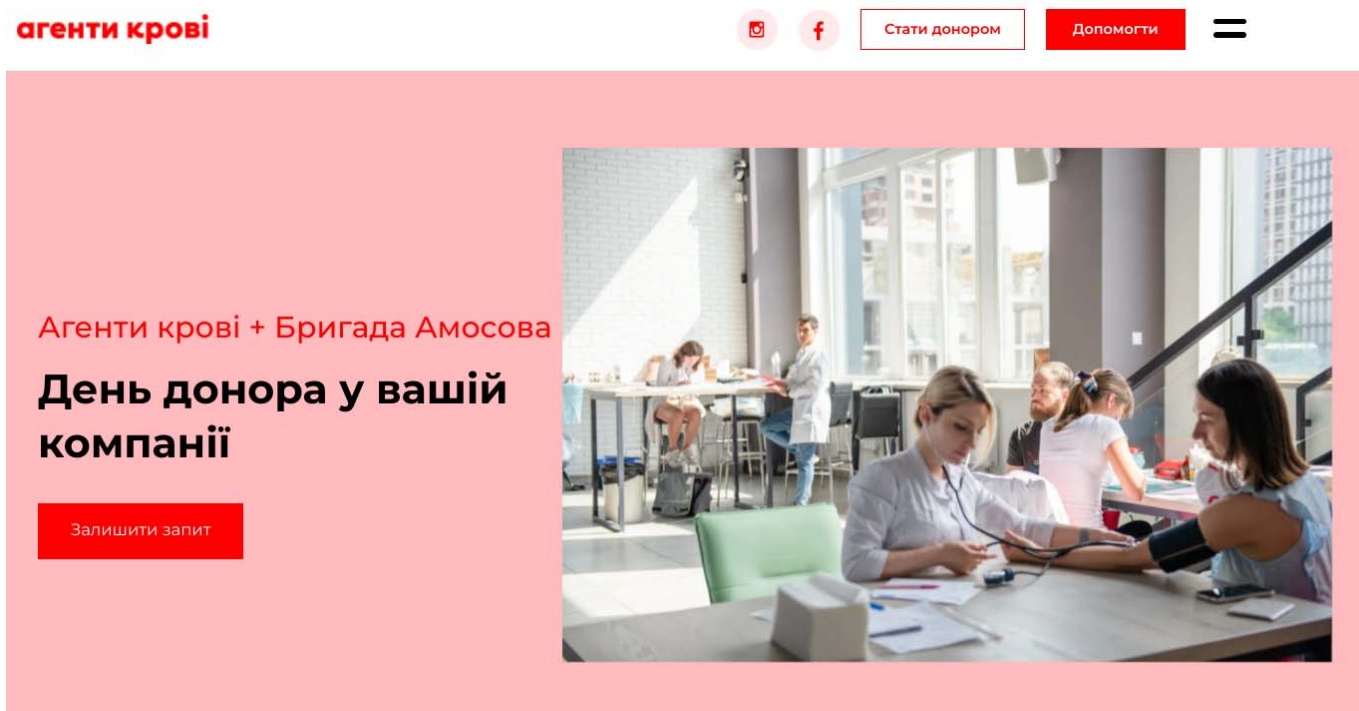


Рисунок 3.16 – День донора в компанії

Ось що потрібно для організації дня донора (рис. 3.17.).

### Що потрібно для організації дня донора?

- ✓ Від 20 бажуючих донорів
- ✓ Відповідальний координатор від компанії
- ✓ Реєстрація компанії у гугл формі
- ✓ Заповнення донорами анкети
- ✓ Приміщення для реєстрації (20м2) та під "операційну" (50+м2)
- ✓ Забезпечення донорів чаєм, печивом та водою
- ✓ Затверджена дата і час
- ✓ Сплачений або виставлений рахунок

Рисунок 3.17 – Що потрібно для організації дня донора

Також є різні пакети для організації дня донора (рис. 3.18.).

ПАКЕТИ	Базофіл 5000 ₴	Еритроцит 10000 ₴	Лімфоцит 15000 ₴	Плазма 30000 ₴
Інформаційна підтримка координатора компанії	✓	✓	✓	✓
Віезд медичної бригади Амосова (6 чоловік)	✓	✓	✓	✓
Координація і супровід Дня донора командою	✓	✓	✓	✓
Лекція лікаря трансфузіолога про донорство		✓	✓	✓
Значок кожному успішному донору (20 шт.)	✓	✓	✓	✓
Фотосесія	✗	✗	✓	✓
Відео з Дня донора	✗	✗	✗	✓
Солодкий презент від нашого партнера @bakehouse	✗	✗	✓	✓
Диплом із подякою соціально-відповідальної компанії	✓	✓	✓	✓



Якщо донорів більше 20 осіб, до основної вартості пакету додається сума у 3000 грн. за кожні 20 донорів.

Наприклад:

Якщо пакет за 5 000 грн, а донорів 60 осіб, то разом буде: 5 000 + 3 000 + 3 000 = 11 000 грн.

Рисунок 3.18 – Пакети до дня донора

Волонтери організують різні проекти та на сайті представленні статті з детальним описом (рис. 3.19.).



## ARTDONATION 2022

Запрошуємо дизайнерів, художників та ілюстраторів взяти участь у 3му Щорічному конкурсі постерів присвячених Всесвітньому дню донора «Artdonation».

**Донорство крові - червона лінія фронту**

[Детальніше](#)



## Кров Чарівна 2022

У Києві та Одесі до кінця зими 2022 року триває акція "Кров чарівна", яку організували Агенти крові. Завдяки ній, ви можете стати донором та отримати за це портрет від іменитих фотографів України.

[Детальніше](#)

[Приєднатися до акції](#)

Рисунок 3.19 – Проекти організовані волонтерами

## 4.1. Опис ідеї проекту

Ідея веб-сайту для донорів крові полягає в тому, щоб створити онлайн-платформу, на якій люди зможуть зареєструватися, щоб стати донорами крові, і призначити зустріч для здачі крові. Веб-сайт також надає інформацію про процес та важливість донорства крові. Крім того, веб-сайт можна використовувати для зв'язку донорів крові з лікарнями та банками крові, які потребують донорства, що забезпечить ефективний і спрощений процес збору та розповсюдження крові. Це потенційно може допомогти врятувати життя, забезпечуючи постійне постачання крові для тих, хто її потребує.

Ось кілька ідей, які включає веб-сайт для донорів крові:

- Зручний інтерфейс, який полегшує людям реєстрацію як донорів крові та планування зустрічей.
- Функція карти та каталогу, яка дозволяє донорам легко знаходити центри донорства крові та лікарні у своєму районі.
- Освітні ресурси та інформація про різні групи крові та процес здачі крові, щоб допомогти потенційним донорам зрозуміти чому це важливо.
- Спосіб для зареєстрованих донорів відстежувати свою історію донорства, включаючи дату останнього донорства та кількість зданої крові.
- Система сповіщень, яка може нагадати зареєстрованим донорам, коли прийде час призначити наступну зустріч для пожертвування.
- Функція, яка дозволяє лікарням і банкам крові швидко й легко знаходити доступних донорів у своєму регіоні.
- Спосіб для донорів поділитися своїм досвідом і мотивувати інших стати донорами крові.

Загалом веб-сайт для донорів крові має на меті зробити процес донорства крові більш доступним, ефективним і корисним як для донорів, так і для тих, хто потребує переливання крові.

## 4.2. Розроблення ринкової стратегії

Щоб розробити ринкову стратегію для проекту створення веб-сайту для донорів крові, потрібно кілька ключових кроків:

- Провести дослідження ринку, щоб зрозуміти потенційну аудиторію веб-сайту. Це може включати опитування потенційних донорів, щоб дізнатися про їхні мотиви, проблеми та потреби, а також дослідження ширшого ринку послуг донорства крові та онлайн-ресурсів охорони здоров'я.
- Визначити унікальну цінність веб-сайту та використати це, щоб відрізнити його від інших ресурсів донорства крові. Це може включати підкреслення зручності та доступності веб-сайту, комплексного характеру його освітніх ресурсів або його здатності зв'язувати донорів із лікарнями та банками крові, які цього потребують.
- Розробити план цільового маркетингу та охоплення, щоб охопити потенційних донорів і залучити їх до веб-сайту. Це може включати цільову цифрову рекламу, партнерство з лікарнями та банками крові, а також роботу з громадськими організаціями та іншими групами, які можуть бути зацікавлені в просуванні донорства крові.
- Створити зручний і привабливий веб-сайт, який ефективно передає ціннісні пропозиції та заохочує користувачів реєструватися як донори крові та призначати зустрічі. Це може включати зручний дизайн, чіткий і привабливий вміст та інтерактивні функції, які роблять веб-сайт приємним і інформативним у використанні.
- Відстежити та виміряти ефективність маркетингових і інформаційних зусиль і використовувати ці дані для постійного вдосконалення веб-сайту та його здатності зв'язувати донорів із тими, хто цього потребує. Це може включати відстеження трафіку веб-сайту та показників залучення користувачів, проведення опитувань користувачів та аналіз інших даних, щоб зрозуміти, що працює добре, а що можна покращити.

Загалом, метою ринкової стратегії проекту розробки веб-сайту для донорів крові буде ефективне просування веб-сайту та його ціннісної пропозиції, а також заохочення якомога більшої кількості людей зареєструватися та стати донорами крові.

### **4.3 Вимоги до технічного забезпечення**

Існує кілька основних вимог до технічної підтримки проекту розробки сайту:

- Надійна та безпечна платформа хостингу, яка може підтримувати веб-сайт і його базу користувачів, з достатньою пропускну здатністю, сховищем і обчислювальною потужністю для обробки очікуваного трафіку та використання.
- Зручна та інтуїтивно зрозуміла система керування вмістом (CMS), яка дозволяє легко оновлювати та підтримувати вміст веб-сайту, включаючи текст, зображення, відео та інші медіафайли.
- Зручний для мобільних пристроїв дизайн, який гарантує, що веб-сайт доступний і простий у використанні на різноманітних пристроях, включаючи смартфони та планшети.
- Інтеграція з іншими відповідними системами та службами, такими як інструменти планування зустрічей, картографічні служби та служби визначення місцезнаходження, а також платформи соціальних мереж.
- Заходи безпеки для захисту даних користувачів і запобігання несанкціонованому доступу до конфіденційної інформації, такої як записи донорів і розклад зустрічей.

## **ВИСНОВКИ**

Інформаційна система на даний момент є актуальною темою, адже відіграє важливу роль серед споживачів, оскільки реєстрація, фінансова підтримка та вибір місця на здачу крові є більш зручними та доступними на web-сайті, аніж фізичний похід в пункти реєстрації.

У даній роботі визначено доцільність розробки програмного забезпечення для web-сайту “Агенти крові”. З отриманого результату можна зробити висновок, що в провадженні розробки гарантоване одержання економічного та технічного ефекту.

Під час виконання даної дипломної роботи по створенню web-сайту “Агенти крові”, отримано такі результати:

- проаналізовано стан проблемної області, а саме: стан технічної області на етапі розвитку технологій, можливі способи вирішення проблеми та розбір конструкції побудови магазину з порівнянням до вже існуючого продукту;
- визначено інформаційне та математичне забезпечення, а саме: проектування та аналіз бази даних для інформаційної системи, побудову об’єктно-орієнтованої моделі бази даних, математичне та алгоритмічне забезпечення;
- проведено паралель між самостійною розробкою та великою компанією з залученням великого спектру працівників різних сфер діяльності;
- обгрунтовано засоби розробки програмного забезпечення, вимоги до технічного забезпечення, та опис інтерфейсу і програмної реалізації web-сайту “Агенти крові”.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://www.bibliofond.ru/view.aspx?id=601048>
2. <https://ukrbukva.net/page,4,100812-Razrobotka-bazy-dannyh-dlya-CMS-internet-magazina.html>
3. <https://ukrbukva.net/41541-Razrobotka-bazy-dannyh-dlya-internet-magazina.html>
4. <https://www.oreilly.com/library/view/hands-on-big-data/9781788620901/b2b76205-d932-4afa-957f-7013376c553e.xhtml>
5. <https://www.bestreferat.ru/referat-141220.html>
6. <https://phoenixnap.com/kb/object-oriented-database>
7. [https://essuir.sumdu.edu.ua/bitstream-download/123456789/69573/1/Chernenko\\_Masters\\_thesis.pdf;jsessionid=4B9C3A38E731765CA7DAC32A145E3ED5](https://essuir.sumdu.edu.ua/bitstream-download/123456789/69573/1/Chernenko_Masters_thesis.pdf;jsessionid=4B9C3A38E731765CA7DAC32A145E3ED5)
8. [https://ela.kpi.ua/bitstream/123456789/31044/1/Brovdi\\_bakalavr.pdf](https://ela.kpi.ua/bitstream/123456789/31044/1/Brovdi_bakalavr.pdf)
9. <http://inmad.vntu.edu.ua/portal/static/C7464950-961A-4E5E-8B98-4C7DBA978508.pdf>
10. <https://www.bibliofond.ru/view.aspx?id=601048>
11. <https://ukrbukva.net/page,4,100812-Razrobotka-bazy-dannyh-dlya-CMS-internet-magazina.html>
12. <https://ukrbukva.net/41541-Razrobotka-bazy-dannyh-dlya-internet-magazina.html>
13. <https://www.oreilly.com/library/view/hands-on-big-data/9781788620901/b2b76205-d932-4afa-957f-7013376c553e.xhtml>
14. <https://www.bestreferat.ru/referat-141220.html>
15. <https://phoenixnap.com/kb/object-oriented-database>
16. [https://essuir.sumdu.edu.ua/bitstream-download/123456789/69573/1/Chernenko\\_Masters\\_thesis.pdf;jsessionid=4B9C3A38E731765CA7DAC32A145E3ED5](https://essuir.sumdu.edu.ua/bitstream-download/123456789/69573/1/Chernenko_Masters_thesis.pdf;jsessionid=4B9C3A38E731765CA7DAC32A145E3ED5)
17. [https://ela.kpi.ua/bitstream/123456789/31044/1/Brovdi\\_bakalavr.pdf](https://ela.kpi.ua/bitstream/123456789/31044/1/Brovdi_bakalavr.pdf)

18. <http://inmad.vntu.edu.ua/portal/static/C7464950-961A-4E5E-8B98-4C7DBA978508.pdf>
19. <https://www.bibliofond.ru/view.aspx?id=601048>
20. <https://ukrbukva.net/page,4,100812-Razrobotka-bazy-dannyh-dlya-CMS-internet-magazina.html>
21. <https://ukrbukva.net/41541-Razrobotka-bazy-dannyh-dlya-internet-magazina.html>
22. <https://www.oreilly.com/library/view/hands-on-big-data/9781788620901/b2b76205-d932-4afa-957f-7013376c553e.xhtml>
23. <https://www.bestreferat.ru/referat-141220.html>
24. <https://phoenixnap.com/kb/object-oriented-database>
25. [https://essuir.sumdu.edu.ua/bitstream-download/123456789/69573/1/Chernenko\\_Masters\\_thesis.pdf;jsessionid=4B9C3A38E731765CA7DAC32A145E3ED5](https://essuir.sumdu.edu.ua/bitstream-download/123456789/69573/1/Chernenko_Masters_thesis.pdf;jsessionid=4B9C3A38E731765CA7DAC32A145E3ED5)
26. [https://ela.kpi.ua/bitstream/123456789/31044/1/Brovdiu\\_bakalavr.pdf](https://ela.kpi.ua/bitstream/123456789/31044/1/Brovdiu_bakalavr.pdf)
27. <http://inmad.vntu.edu.ua/portal/static/C7464950-961A-4E5E-8B98-4C7DBA978508.pdf>
28. <https://www.oreilly.com/library/view/hands-on-big-data/9781788620901/b2b76205-d932-4afa-957f-7013376c553e.xhtml>
29. <https://www.bestreferat.ru/referat-141220.html>
30. <https://phoenixnap.com/kb/object-oriented-database>

## ДОДАТОК

### Model.php

```
<?php
class NewsModel extends Model {
public function getNews() {
    $sql = "SELECT * FROM news";
    $stmt = $this->db->prepare($sql);
    $stmt->execute();

    $result = array();
    while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
        $result[$row['id']] = $row;
    }
    return $result;
}

public function getNewsById($id) {
    $sql = "SELECT * FROM news WHERE id= :id";
    $stmt = $this->db->prepare($sql);
    $stmt->bindValue(":id", $id, PDO::PARAM_INT);
    $stmt->execute();

    $result = array();

    $result = $stmt->fetch(PDO::FETCH_ASSOC);
    if(empty($result)) {
        return false;
    } else {
        return $result;
    }
}

public function saveNews($id, $title, $text) {
    $sql = "UPDATE news
            SET title = :title, description = :text
            WHERE id =:id
            ";
    $stmt = $this->db->prepare($sql);
    $stmt->bindValue(":title", $title, PDO::PARAM_STR);
    $stmt->bindValue(":text", $text, PDO::PARAM_STR);
    $stmt->bindValue(":id", $id, PDO::PARAM_INT);
```

```

$stmt->execute();

}

}

```

## Controller.php

```
<?php
```

```
class NewsController extends Controller {
```

```
private $pageTpl = '/views/news.tpl.php';
```

```
public function __construct() {
    $this->model = new NewsModel();
    $this->view = new View();
}
```

```
public function index() {
    if(!$_SESSION['user']) {
        header("Location: /");
    }

    $this->pageData['title'] = "Новости";
```

```

    $news = $this->model->getNews();
    $this->pageData['news'] = $news;

```

```

    $this->view->render($this->pageTpl, $this->pageData);

```

```
}
```

```
public function getNewsById() {
    if(!$_SESSION['user']) {
        header("Location: /");
    }
}
```

```

    if(isset($_POST['id'])) {
        $newsId = $_POST['id'];
        if($newsInfo = $this->model->getNewsById($newsId)) {
            echo json_encode($newsInfo);
        } else {
            echo json_encode(array("success" => false, "text" => "Новость
не найдена"));
        }
    }
}

```

```

    } else {
        echo json_encode(array("success" => false, "text" => "Ошибка"));
    }
}

public function save() {
    if(!$_SESSION['user']) {
        header("Location: /");
    }

    if(!empty($_POST)      &&      !empty($_POST['title'])      &&
!empty($_POST['text'])) {
        $newsId = $_POST['id'];
        $newsTitle = $_POST['title'];
        $newsText = $_POST['text'];
        $this->model->saveNews($newsId, $newsTitle, $newsText);
        echo json_encode(array("success" => true, "text" => "Saved"));
    } else {
        echo json_encode(array("success" => false, "text" => "Error"));
    }
}
}
}

```

### News.php

```

var news = angular.module('news', ['ui.bootstrap']);

news.controller('newsController', function($scope, $http, $uibModal) {

$scope.open = function(id) {

    $http({
        method: "POST",
        url: "http://cabinet.kamil-abzalov.ru/cabinet/news/getNewsById",
        data: $.param({id: id}),
        headers: {"Content-Type": 'application/x-www-form-urlencoded'}
    }).then(function(result) {
        console.log(result);
        if(result.data.success != false) {
            $scope.newsData = result.data;

            var modalWindow = $uibModal.open({
                animation: true,
                controller: "modalWindowController",

```

```

        templateUrl: '/views/modal.tpl.php',
        resolve: {
            newsData: function() {
                return $scope.newsData;
            }
        }
    });
}

});

news.controller('modalWindowController', function($scope, $http, $window,
$uibModalInstance, newsData) {
    $scope.newsId = newsData.id;
    $scope.newsTitle = newsData.title;
    $scope.newsDescription = newsData.description;
    $scope.save = function() {

    }

    $scope.close = function() {
        $uibModalInstance.dismiss('cancel');
    }

    $scope.save = function() {
        $http({
            method: "POST",
            url: "http://cabinet.kamil-abzalov.ru/cabinet/news/save",
            data: $.param({id: $scope.newsId, title: $scope.newsTitle, text:
$scope.newsDescription}),
            headers: {"Content-Type": 'application/x-www-form-urlencoded'}
        }).then(function(result){
            alert(result.data.text);
            $uibModalInstance.close();
            $window.location.href = 'cabinet/news';
        })
    }

});

```

### **Modal.tpl.php**

```

<div class="modal-content">
  <div class="modal-header">
    <h4 class="modal-title">Edit</h4>
  </div>
  <div class="modal-body">
    <form>
      <input type="hidden" data-ng-model="newsId" name="newsId">
      <div class="form-group">
        <input type="text" data-ng-model="newsTitle" name="newsTitle"
class="form-control">
      </div>
      <div class="form-group">
        <textarea data-ng-model="newsDescription" name="newsDescription"
class="form-control"></textarea>
      </div>
    </form>
  </div>
  <div class="modal-footer">
    <button type="button" class="btn btn-default" data-dismiss="modal" data-ng-
click="close();">Close</button>
    <button type="button" class="btn btn-primary" data-ng-
click="save()">Save</button>
  </div>
</div>

```

```
<!DOCTYPE html>
```

```
<html lang="ru" data-ng-app="news">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<title><?php echo $pageData['title']; ?></title>
```

```
<!-- Bootstrap Core CSS -->
```

```
<link href="/css/bootstrap.min.css" rel="stylesheet">
```

```
<!-- MetisMenu CSS -->
```

```
<link href="/css/admin/metisMenu.min.css" rel="stylesheet">
```

```
<!-- Custom CSS -->
```

```
<link href="/css/admin/sb-admin-2.css" rel="stylesheet">
```

```

<!-- Morris Charts CSS -->
<link href="/css/admin/morris.css" rel="stylesheet">

<!-- Custom Fonts -->
<link href="/css/font-awesome.min.css" rel="stylesheet" type="text/css">

<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if lt IE 9]>
  <script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
  <script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->

</head>

<body>

  <div id="wrapper">

    <!-- Navigation -->
    <nav class="navbar navbar-default navbar-static-top" role="navigation"
style="margin-bottom: 0">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target=".navbar-collapse">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="index.html">Кабинет</a>
      </div>
      <!-- /.navbar-header -->

      <ul class="nav navbar-top-links navbar-right">
        <!-- /.dropdown -->
        <li class="dropdown">
          <a class="dropdown-toggle" data-toggle="dropdown" href="#">
            <i class="fa fa-user fa-fw"></i> <i class="fa fa-caret-down"></i>
          </a>
          <ul class="dropdown-menu dropdown-user">

```

```

        <li><a href="/cabinet/profile"><i class="fa fa-user fa-fw"></i>
Профиль</a>
        </li>
        <li class="divider"></li>
        <li><a href="/cabinet/logout"><i class="fa fa-sign-out fa-fw"></i>
Выйти</a>
        </li>
    </ul>
    <!-- /.dropdown-user -->
</li>
<!-- /.dropdown -->
</ul>
<!-- /.navbar-top-links -->

<div class="navbar-default sidebar" role="navigation">
    <div class="sidebar-nav navbar-collapse">
        <ul class="nav" id="side-menu">

            <li>
                <a href="/cabinet"><i class="fa fa-area-chart"></i>
Статистика</a>
            </li>
            <li>
                <a href="/cabinet/products"><i class="fa fa-cart-plus"></i>
Товары</a>
            </li>
            <li>
                <a href="/cabinet/users"><i class="fa fa-user-o"></i>
Пользователи</a>
            </li>
            <li>
                <a href="/cabinet/news"><i class="fa fa-newspaper-o"></i>
Новости</a>
            </li>
        </ul>
    </div>
<!-- /.sidebar-collapse -->
</div>
<!-- /.navbar-static-side -->
</nav>

<div id="page-wrapper" data-ng-controller="newsController">
    <div class="row">
        <div class="col-lg-12">

```

```

        <h1 class="page-header"><?php echo $pageData['title']; ?></h1>
    </div>
    <!-- /.col-lg-12 -->
</div>
<!-- /.row -->
<div class="row">
    <div class="col-lg-12">
        <!-- /.panel -->
        <div class="panel panel-default">
            <div class="panel-heading">
                <i class="fa fa-bar-chart-o fa-fw"></i> Новости
            </div>
            <!-- /.panel-heading -->
            <div class="panel-body">
                <div class="row">
                    <div class="col-lg-12">
                        <?php foreach($pageData['news'] as $news) { ?>
                            <div class="panel-group" id="accordion" role="tablist" aria-
multiselectable="true">
                                <div class="panel panel-default">
                                    <div
                                        class="panel-heading"
                                        role="tab"
id="headingOne">
                                        <h4 class="panel-title">
                                            <a data-ng-click="open(<?php echo $news['id'];
?>)" role="button" aria-expanded="true" aria-controls="collapseOne">
                                                <?php echo $news['title']; ?>
                                            </a>
                                        </h4>
                                    </div>
                                    <div id="collapseOne" class="panel-collapse collapse
in" role="tabpanel" aria-labelledby="headingOne">
                                        <div class="panel-body">
                                            <?php echo $news['description']; ?>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        <?php } ?>
                    </div>
                <!-- /.col-lg-4 (nested) -->
                <!-- /.col-lg-8 (nested) -->
            </div>
            <!-- /.row -->
        </div>
    </div>

```

```

        <!-- /.panel-body -->
    </div>
    <!-- /.panel -->
    <!-- /.panel -->
</div>
<!-- /.col-lg-8 -->
</div>
<!-- /.row -->
</div>
<!-- /#page-wrapper -->

```

```

</div>
<!-- /#wrapper -->

```

```

<!-- jQuery -->
<script src="/js/jquery.js"></script>

```

```

<!-- Angular -->
<script src="/js/angular.min.js"></script>
<script
                src="https://angular-ui.github.io/bootstrap/ui-bootstrap-tpls-
2.5.0.min.js"></script>
<script src="/js/admin/news.js"></script>

```

```

<!-- Bootstrap Core JavaScript -->
<script src="/js/bootstrap.min.js"></script>

```

```

<!-- Metis Menu Plugin JavaScript -->
<script src="/js/admin/metisMenu.js"></script>

```

```

<!-- Custom Theme JavaScript -->
<script src="/js/admin/sb-admin-2.js"></script>

```

```

</body>

```

```

</html>

```

### **Animation.js**

```

(function () {
    "use strict";

    if(typeof webPacker === "undefined")
    {

```

```

        return;
    }

    if (!window.BX)
    {
        window.BX = {};
    }
    else if (window.BX.SiteButton)
    {
        return;
    }

    var Classes = webPacker.classes;
    var Browser = webPacker.browser;
    var Type = webPacker.type;

    var Shadow = {
        clickHandler: null,
        shadowNode: null,
        displayed: false,
        init: function(params)
        {
            this.shadowNode = params.shadowNode;

            webPacker.addEventListener(this.shadowNode,
this.onClick.bind(this));
            webPacker.addEventListener(document, 'keyup', function (e) {
                if ((e || window.e).keyCode === 27) this.onClick();
            }).bind(this));
        },
        onClick: function()
        {
            if (!this.displayed)
            {
                return;
            }

            WidgetManager.hide();
            ButtonManager.hide();

            if (!this.clickHandler)
            {
                return;
            }

```

```

        this.clickHandler.apply(this, []);
        this.clickHandler = null;
    },
    show: function(clickHandler)
    {
        this.clickHandler = clickHandler;
        Classes.add(this.shadowNode, 'b24-widget-button-show');
        Classes.remove(this.shadowNode, 'b24-widget-button-hide');

        Hacks.saveScrollPos();
        Classes.add(document.documentElement, 'crm-widget-button-mobile',
true);
        this.displayed = true;
    },
    hide: function()
    {
        if (this.displayed)
        {
            Classes.add(this.shadowNode, 'b24-widget-button-hide');
        }
        Classes.remove(this.shadowNode, 'b24-widget-button-show');

        Classes.remove(document.documentElement, 'crm-widget-button-
mobile');
        Hacks.restoreScrollPos();
        this.displayed = false;
    }
};

```

```

var ButtonManager = {
    isShown: false,
    isInit: false,
    wasOnceShown: false,
    wasOnceClick: false,
    blankButtonNode: null,
    list: [],
    frozen: false,
    init: function(params)
    {
        this.container = params.container;
        this.blankButtonNode = params.blankButtonNode;
        this.openerButtonNode = params.openerButtonNode;
    }
};

```

```

        /* location magic */
        this.openerClassName = Manager.config.location > 3 ? 'b24-widget-
button-bottom' : 'b24-widget-button-top';

        webPacker.addEventListener(this.openerButtonNode, 'click', function
0 {
            if (this.frozen)
            {
                this.unfreeze();
            }
            else
            {
                if (this.list.length === 1 && this.list[0].onclick &&
!this.list[0].href)
                {
                    this.list[0].onclick.apply(this, []);
                }
                else
                {
                    this.toggle();
                }
            }
        }.bind(this));

        this.isInit = true;

        this.list.forEach(function (button) {
            if (!button.node) this.insert(button);
        }, this);

        Animation.restart();
    },
    getByType: function(type)
    {
        var buttons = this.list.filter(function (button) {
            return type === button.type;
        }, this);

        return (buttons.length > 0 ? buttons[0] : null);
    },
    toggle: function()
    {
        this.isShown ? this.hide() : this.show();
    },

```

```

show: function()
{
    if(Browser.isIOS())
    {
        Classes.add(document.documentElement, 'bx-ios-fix-frame-
focus');
    }

    //if (Browser.isMobile())
    {
        Shadow.show();
    }

    this.isShown = true;
    this.wasOnceShown = true;
    Classes.add(Manager.container, this.openerClassName);
    Classes.add(this.container, 'b24-widget-button-show');
    Classes.remove(this.container, 'b24-widget-button-hide');

    Hello.hide();
},
hide: function()
{
    if(Browser.isIOS())
    {
        Classes.remove(document.documentElement, 'bx-ios-fix-frame-
focus');
    }

    this.isShown = false;

    Classes.add(this.container, 'b24-widget-button-hide');
    Classes.remove(this.container, 'b24-widget-button-show');
    Classes.remove(Manager.container, this.openerClassName);

    Hello.hide();
    Shadow.hide();
},
freeze: function(type)
{
    this.hide();
    if (type)
    {
        Animation.freeze(type);
    }
}

```

```

    }

    this.frozen = true;
},
unfreeze: function()
{
    Animation.start();
    WidgetManager.hide();
    this.hide();
    this.frozen = false;
},
displayButton: function (id, display)
{
    this.list.forEach(function (button) {
        if (button.id !== id) return;
        if (!button.node) return;
        button.node.style.display = display ? "" : 'none';
    });
},
sortOut: function ()
{
    this.list.sort(function(buttonA, buttonB){
        return buttonA.sort > buttonB.sort ? 1 : -1;
    });

    this.list.forEach(function(button){
        if (!button.node) return;
        button.node.parentNode.appendChild(button.node);
    });
},
add: function (params)
{
    this.list.push(params);
    return this.insert(params);
},
insert: function (params)
{
    if (!this.isInit)
    {
        params.node = null;
        return null;
    }

    var buttonNode = this.blankButtonNode.cloneNode(true);

```

```

params.node = buttonNode;
params.sort = params.sort || 100;

buttonNode.setAttribute('data-b24-crm-button-widget', params.id);
buttonNode.setAttribute('data-b24-widget-sort', params.sort);

if (params.classList && params.classList.length > 0)
{
    params.classList.forEach(function (className) {
        Classes.add(buttonNode, className);
    }, this);
}

if (params.title)
{
    var tooltipNode = buttonNode.querySelector('[data-b24-crm-
button-tooltip]');
    if (tooltipNode)
    {
        tooltipNode.innerText = params.title;
    }
    else
    {
        buttonNode.title = params.title;
    }
}

if (params.icon)
{
    buttonNode.style['background-image'] = 'url(' + params.icon +
');
}
else
{
    if (params.iconColor)
    {
        setTimeout(function () {
            var styleName = 'background-image';
            if(!window.getComputedStyle)
            {
                return;
            }
        }
    }
}

```

```

        var styleValue =
window.getComputedStyle(buttonNode, null).getPropertyValue(styleName);
        buttonNode.style[styleName] = (
            styleValue || ""
        ).replace('FFF', params.iconColor.substring(1));
    }, 1000);
    }

    if (params.bgColor)
    {
        buttonNode.style['background-color'] = params.bgColor;
    }
}

if (params.href)
{
    buttonNode.href = params.href;
    buttonNode.target = params.target ? params.target : '_blank';
}

if (params.onclick)
{
    webPacker.addEventListener(buttonNode, 'click', function () {
        this.wasOnceClick = true;
        params.onclick.apply(this, []);
    }.bind(this));
}

this.container.appendChild(buttonNode);
this.sortOut();
Animation.restart();

return buttonNode;
}
};

```

```

var Animation = {
    isInit: false,
    timer: null,
    timerPeriod: 1500,
    icons: [],
    pulsar: null,
    stop: function()
    {

```

```

        this.rotate(false).pulse(false);
    },
    freeze: function(type)
    {
        this.rotate(type).pulse(false);
    },
    start: function()
    {
        this.rotate().pulse(true);
    },
    rotate: function(type)
    {
        this.init();
        if (this.timer) clearTimeout(this.timer);
        if (type === false)
        {
            return this;
        }

        var className = 'b24-widget-button-icon-animation';
        var current = 0;

        var icons = this.icons.filter(function (icon) { return !icon.hidden; });
        icons.forEach(function (icon, index) {
            if (Classes.has(icon.node, className)) current = index;
            Classes.remove(icon.node, className);
        }, this);

        var icon;
        if (type && !(icon = icons.filter(function (icon) { return icon.type ===
type; })[0]))
        {
            throw new Error('Animation.rotate: Unknown type `` + type +
``');
        }

        if (!icon && !(icon = icons.concat(this.icons).slice(current+1)[0]))
        {
            return this;
        }

        Classes.add(icon.node, className);
        if (!type && icons.length > 1)
        {

```

```

        this.timer = setTimeout(this.rotate.bind(this), this.timerPeriod);
    }

    return this;
},
pulse: function(state)
{
    Classes.change(this.pulsar, 'b24-widget-button-pulse-animate', state);
    return this;
},
restart: function()
{
    this.isInit = false;
    this.start();
},
init: function()
{
    if (this.isInit)
    {
        return this;
    }

    var attributeName = 'data-b24-crm-button-icon';
    this.icons = Type.toArray(
        Manager.context.querySelectorAll('[ + attributeName + ]')
    ).map(function (node) {
        var type = node.getAttribute(attributeName);
        var hidden = !ButtonManager.getByType(type);
        node.style.display = hidden ? 'none' : '';
        return {node: node, type: type, hidden: hidden};
    }, this).filter(function (icon) {
        return !icon.hidden;
    }, this);

    this.pulsar = Manager.context.querySelector('[data-b24-crm-button-
pulse]');

    this.isInit = true;

    return this;
}
};

var WidgetManager = { /* Widget Manager */
    showedWidget: null,

```

```

loadedCount: 0,
getList: function()
{
    return Manager.config.widgets.filter(function (widget) {
        return widget.isLoaded;
    }, this);
},
getById: function(id)
{
    var widgets = Manager.config.widgets.filter(function (widget) {
        return (id === widget.id && widget.isLoaded);
    }, this);

    return (widgets.length > 0 ? widgets[0] : null);
},
hide: function()
{
    if (!this.showedWidget)
    {
        return;
    }

    var showedWidget = this.showedWidget;
    this.showedWidget = null;
    if(showedWidget.hide)
    {
        Utils.evalGlobal(showedWidget.hide);
    }

    Manager.show();
    Shadow.hide();
},
show: function(widget)
{
    this.storeTrace(widget);

    if(!widget.show || !Type.isString(widget.show))
    {
        return;
    }

    this.showedWidget = widget;
    if (!widget.freeze)
    {

```

```

        Shadow.show();
    }

    Utils.evalGlobal(widget.show);
    if (widget.freeze)
    {
        Manager.freeze(widget.type);
    }
    else
    {
        Manager.hide();
    }
},
storeTrace: function(widget)
{
    if (!widget || !widget.tracking || !widget.tracking.detecting)
    {
        return;
    }
    widget.tracking.detecting = false;

    var trace = Manager.getTrace({channels: [widget.tracking.channel]});
    Manager.b24Tracker.guest.storeTrace(trace);
},
showById: function(id)
{
    var selectorId = this.getById(id);
    if (selectorId)
    {
        this.show(selectorId);
    }
},
checkAll: function()
{
    return Manager.config.widgets.some(this.check, this);
},
check: function(widget)
{
    return this.checkPages(widget) && this.checkWorkTime(widget);
},
checkPagesAll: function()
{
    return Manager.config.widgets.some(this.checkPages, this);
},

```

```

checkPages: function(widget)
{
    var isPageFound = Utils.isCurPageInList(widget.pages.list);
    if(widget.pages.mode === 'EXCLUDE')
    {
        return !isPageFound;
    }
    else
    {
        return isPageFound;
    }
},
checkWorkTimeAll: function()
{
    return Manager.config.widgets.some(this.checkWorkTime, this);
},
checkWorkTime: function(widget)
{
    if (!widget.workTime)
    {
        widget.isWorkTimeNow = true;
        widget.isWorkTimeChecked = true;
    }
    if (widget.isWorkTimeChecked)
    {
        return widget.isWorkTimeNow;
    }

    var workTime = widget.workTime;

    // get date with timezone
    var date = new Date();
    if (Manager.config.serverTimeStamp)
    {
        date = new Date(Manager.config.serverTimeStamp);
    }
    var    timeZoneOffset    =    workTime.timeZoneOffset    +
date.getTimezoneOffset();
    date = new Date(date.valueOf() + timeZoneOffset * 60000);
    var minutes = date.getMinutes();
    minutes = minutes >= 10 ? minutes : '0' + minutes;
    var currentTime = parseFloat(date.getHours() + '.' + minutes);

    var isSuccess = true;

```

```

if (workTime.dayOff) // check day off
{
    var day = date.getDay();
    if (workTime.dayOff.some(function (item) { return item ===
day; })))
        {
            isSuccess = false;
        }
}

if (isSuccess && workTime.holidays) // check holidays
{
    var currentDay = (date.getMonth() + 1).toString();
    currentDay = (currentDay.length === 1 ? '0' : '') + currentDay;
    currentDay = date.getDate() + '.' + currentDay;
    if (workTime.holidays.some(function (item) { return item ===
currentDay; })))
        {
            isSuccess = false;
        }
}

if (isSuccess) // check time
{
    var isNightMode = workTime.timeTo < workTime.timeFrom;
    if (isNightMode)
    {
        // ex: 22:00 - 08:00
        if (currentTime > workTime.timeTo && currentTime <
workTime.timeFrom)
            {
                isSuccess = false;
            }
    }
    else
    {
        // ex: 09:00 - 18:00
        if (currentTime < workTime.timeFrom || currentTime >
workTime.timeTo)
            {
                isSuccess = false;
            }
    }
}
}

```

```

        widget.isWorkTimeChecked = true;
        widget.isWorkTimeActionRule = false;
        if (!isSuccess && !!workTime.actionRule)
        {
            isSuccess = true;
            widget.isWorkTimeActionRule = true;
        }
        widget.isWorkTimeNow = isSuccess;
        return isSuccess;
    },
    loadAll: function()
    {
        Manager.config.widgets.forEach(this.load, this);
    },
    load: function(widget)
    {
        widget.isLoaded = false;

        Manager.execEventHandler('load-widget-' + widget.id, [widget]);

        if(!this.check(widget))
        {
            return;
        }

        if (widget.workTime && widget.isWorkTimeActionRule)
        {
            switch (widget.workTime.actionRule)
            {
                case 'text':
                    if (widget.type === 'callback')
                    {
                        Manager.addEventHandler('form-init',
function (form) {
                            if (!form.isCallbackForm) return;

                            window.Bitrix24FormLoader.addEventHandler(
                                form, 'init-frame-params',
function (form, frameParameters) {
                                    frameParameters.resultSuccessText = widget.workTime.actionText;

                                    frameParameters.stopCallBack = true;

```

```

    }
    });
    }
    break;
}
}

widget.buttonNode = ButtonManager.add({
    'id': widget.id,
    'type': widget.type,
    'href': this.getButtonUrl(widget),
    'sort': widget.sort,
    'classList': (typeof widget.classList !== "undefined" ?
widget.classList : null),
    'title': (typeof widget.title !== "undefined" ? widget.title : null),
    'onclick': this.show.bind(this, widget),
    'bgColor': widget.useColors ? Manager.config.bgColor : null,
    'iconColor': widget.useColors ? Manager.config.iconColor : null
});

this.loadScript(widget);
widget.isLoaded = true;
this.loadedCount++;
},
getButtonUrl: function(widget)
{
    if (widget.script || !widget.show)
    {
        return null;
    }

    if (Type.isString(widget.show) || !widget.show.url)
    {
        return null;
    }

    var url = null;
    if (Browser.isMobile() && widget.show.url.mobile)
    {
        url = widget.show.url.mobile;
    }
    else if (!Browser.isMobile() && widget.show.url.desktop)
    {

```

```

        url = widget.show.url.desktop;
    }
    else if (Type.isString(widget.show.url))
    {
        url = widget.show.url;
    }

    return url;
},
loadScript: function(widget)
{
    if (!widget.script)
    {
        return;
    }

    var scriptText = "";
    var isAddInHead = false;
    var                                parsedScript
widget.script.match(/<script\b[^\>]*>(.*?)<\script>/i);
    if(parsedScript && parsedScript[1])
    {
        scriptText = parsedScript[1];
        isAddInHead = true;
    }
    else if(!widget.freeze)
    {
        widget.node = Utils.getNodeFromText(widget.script);
        if(!widget.node)
        {
            return;
        }
        isAddInHead = false;

        if (typeof widget.caption !== "undefined")
        {
            var                                widgetCaptionNode
widget.node.querySelector("[data-bx-crm-widget-caption]");
            if (widgetCaptionNode)
            {
                widgetCaptionNode.innerText = widget.caption;
            }
        }
    }
}

```

```

        else
        {
            scriptText = widget.script;
            isAddInHead = true;
        }

        if (isAddInHead)
        {
            widget.node = document.createElement("script");
            try {

                widget.node.appendChild(document.createTextNode(scriptText));
            } catch(e) {
                widget.node.text = scriptText;
            }
            document.head.appendChild(widget.node);
        }
        else
        {
            document.body.insertBefore(widget.node,
document.body.firstChild);
        }
    }
};

var Hello = {
    isInit: false,
    wasOnceShown: false,
    condition: null,
    cookieName: 'b24_sitebutton_hello',
    init: function (params)
    {
        if (this.isInit)
        {
            return;
        }

        this.context = params.context;
        this.showClassName = 'b24-widget-button-popup-show';
        this.config = Manager.config.hello || {};
        this.delay = this.config.delay;

        this.buttonHideNode = this.context.querySelector('[data-b24-hello-btn-
hide]');

```

```

this.iconNode = this.context.querySelector('[data-b24-hello-icon]');
this.nameNode = this.context.querySelector('[data-b24-hello-name]');
this.textNode = this.context.querySelector('[data-b24-hello-text]');

this.initHandlers();
this.isInit = true;

if (webPacker.cookie.get(this.cookieName) === 'y')
{
    return;
}

if (!this.config || !this.config.conditions || this.config.conditions.length
=== 0)
{
    return;
}

if (!this.condition)
{
    this.setConditions(this.config.conditions, true);
}
Manager.addHandler('first-show', this.initCondition.bind(this));
},
setConditions: function (conditions, setOnly)
{
    this.condition = this.findCondition(conditions);
    if (!setOnly)
    {
        this.initCondition();
    }
},
initCondition: function ()
{
    if (!this.condition)
    {
        return;
    }

    if (!this.isInit)
    {
        return;
    }
}

```

```

        if (this.condition.icon)
        {
            this.iconNode.style['background-image'] = 'url(' +
this.condition.icon + ')';
        }
        if (this.condition.name)
        {
            this.nameNode.innerText = this.condition.name;
        }
        if (this.condition.text)
        {
            this.textNode.innerText = this.condition.text;
        }
        if (this.condition.delay)
        {
            this.delay = this.condition.delay;
        }

        this.planShowing();
    },
    initHandlers: function ()
    {
        webPacker.addEventListener(this.buttonHideNode, 'click', function (e)
        {
            this.hide();

            if(!e) e = window.event;
            if(e.stopPropagation){e.preventDefault();e.stopPropagation();}
            else {e.cancelBubble = true;e.returnValue = false;}
        }).bind(this);
        webPacker.addEventListener(this.context, 'click',
this.showWidget.bind(this));
    },
    planShowing: function ()
    {
        if (this.wasOnceShown || ButtonManager.wasOnceClick)
        {
            return;
        }

        setTimeout(this.show.bind(this), (this.delay || 10) * 1000);
    },
    findCondition: function (conditions)
    {

```

```

if (!conditions)
{
    return;
}

var filtered;
// find first suitable condition with mode 'include'
filtered = conditions.filter(function (condition) {
    if (!condition.pages || condition.pages.MODE === 'EXCLUDE'
|| condition.pages.LIST.length === 0)
    {
        return false;
    }

    return Utils.isCurPageInList(condition.pages.LIST);
}, this);
if (filtered.length > 0)
{
    return filtered[0];
}

// find first suitable condition with mode 'exclude'
filtered = conditions.filter(function (condition) {
    if (!condition.pages || condition.pages.MODE === 'INCLUDE')
    {
        return false;
    }

    return !Utils.isCurPageInList(condition.pages.LIST);
}, this);
if (filtered.length > 0)
{
    return filtered[0];
}

// find first condition with empty pages
filtered = conditions.filter(function (condition) {
    return !condition.pages;
}, this);
if (filtered.length > 0)
{
    return filtered[0];
}

```

```

        // nothing found
        return null;
    },
    showWidget: function ()
    {
        this.hide();

        var widget = null;
        if (this.condition && this.condition.showWidgetId)
        {
            widget
WidgetManager.getById(this.condition.showWidgetId);
        }

        if (!widget)
        {
            widget = WidgetManager.getById(this.config.showWidgetId);
        }

        if (!widget)
        {
            var list = WidgetManager.getList();
            if (list.length > 0)
            {
                widget = list[0];
            }
        }

        if (widget)
        {
            WidgetManager.show(widget);
        }
    },
    showImmediately: function (options)
    {
        options = options || null;
        if (options)
        {
            this.setConditions([ {
                'icon': options.icon,
                'name': options.name,
                'text': options.text,
                'page': '',
                'delay': 0
            }

```

```

        })
    }

    this.show(true);
},
show: function (forceShowing)
{
    if (!this.condition)
    {
        return;
    }

    forceShowing = forceShowing || false;
    if (!forceShowing && ButtonManager.isShown) //maybe use
wasOnceShown
    {
        this.planShowing();
        return;
    }

    this.wasOnceShown = true;
    Classes.add(this.context, this.showClassName);
},
hide: function ()
{
    Classes.remove(this.context, this.showClassName);
    webPacker.cookie.set(this.cookieName, 'y', 60*60*6);
}
};

var Utils = {
    getNodeFromText: function(text)
    {
        var node = document.createElement('div');
        node.innerHTML = text;
        return node.children[0];
    },
    evalGlobal: function(text)
    {
        webPacker.resource.loadJs(text, false, true);
    },
    isCurPageInList: function(list)
    {
        var filtered = list.filter(function (page) {

```

```

        page = encodeURI(page);
        var pattern =
this.prepareUrl(page).split('*').map(function(chunk) {
    return chunk.replace(/[\-\[\]\/\{\}\(\)\*\+\?\.\\\^\$\|]/g,
"\$&");
}).join('*');
    pattern = '^' + pattern + '$';
    return (new
RegExp(pattern)).test(this.prepareUrl(window.location.href));
}, this);

    return filtered.length > 0;
},
prepareUrl: function(url)
{
    var result;
    if (url.substring(0, 5) === 'http:')
    {
        result = url.substring(7);
    }
    else if (url.substring(0, 6) === 'https:')
    {
        result = url.substring(8);
    }
    else
    {
        result = url;
    }

    return result;
}
};

var Hacks = {
    scrollPos: 0,
    saveScrollPos: function()
    {
        this.scrollPos = window.pageYOffset;
    },
    restoreScrollPos: function()
    {
        if (!Browser.isMobile())
        {
            return;

```

```

        }
        window.scrollTo(0,this.scrollPos);
    }
};

```

```

var Manager = window.BX.SiteButton = {
    buttons: ButtonManager,
    animation: Animation,
    shadow: Shadow,
    wm: WidgetManager,
    hello: Hello,
    util: Utils,
    classes: Classes,
    hacks: Hacks,

```

```

    isShown: false,

```

```

    init: function(config)
    {

```

```

        this.b24Tracker = window.b24Tracker || {};

```

```

        this.userParams = window.Bitrix24WidgetObject || {};

```

```

        this.config = config;

```

```

        this.handlers = this.userParams.handlers || {};

```

```

        this.eventHandlers = [];

```

```

        this.execEventHandler('init', [this]);

```

```

        if(!this.check())

```

```

        {

```

```

            return;

```

```

        }

```

```

        this.load();

```

```

        if(this.config.delay)

```

```

        {

```

```

            window.setTimeout(this.show.bind(this), 1000

```

```

            *

```

```

this.config.delay);

```

```

        }

```

```

        else

```

```

        {

```

```

            this.show();

```

```

        }

```

```

    },

```

```

    check: function()
    {

```

```

    {

```

```

        if(!this.config.isActivated)
        {
            return false;
        }

        if(this.config.widgets.length === 0)
        {
            return false;
        }

        if(this.config.disableOnMobile && Browser.isMobile())
        {
            return false;
        }

        return WidgetManager.checkAll();
    },
    loadResources: function()
    {
        //throw Error('loadResources unavailable');
    },
    load: function()
    {
        this.execEventHandler('load', [this]);

        // set common classes
        Browser.isIOS() ? Classes.add(document.documentElement, 'bx-ios') :
null;
        Browser.isMobile() ? Classes.add(document.documentElement, 'bx-
touch') : null;

        // load resources
        this.loadResources();

        this.container = document.body.querySelector('[data-b24-crm-button-
cont]');
        this.context = this.container.parentNode;

        // init components
        this.shadow.init({
            'shadowNode':      this.context.querySelector('[data-b24-crm-
button-shadow]')
        });
    }

```

```

        this.buttons.init({
            'container': this.container.querySelector('[data-b24-crm-button-
block]'),
            'blankButtonNode': this.context.querySelector('[data-b24-crm-
button-widget-blank]'),
            'openerButtonNode': this.context.querySelector('[data-b24-crm-
button-block-button]')
        });
        this.hello.init({
            context: this.container.querySelector('[data-b24-crm-hello-
cont]')
        });

        // load widgets
        this.wm.loadAll();

        this.execEventHandler('loaded', [this]);
    },
    show: function()
    {
        Classes.remove(this.container, 'b24-widget-button-disable');
        Classes.add(this.container, 'b24-widget-button-visible');

        this.execEventHandler('show', [this]);
        if (!this.isShown)
        {
            this.execEventHandler('first-show', [this]);
        }
        this.isShown = true;
    },
    hide: function()
    {
        Classes.add(this.container, 'b24-widget-button-disable');
        this.execEventHandler('hide', [this]);
    },
    freeze: function(type)
    {
        setTimeout(function () {
            ButtonManager.freeze(type);
            this.show();
        }.bind(this));
    },
    addEventHandler: function(eventName, handler)
    {

```

```

        if (!eventName || !handler)
        {
            return;
        }

        this.eventHandlers.push({
            'eventName': eventName,
            'handler': handler
        });
    },
    execEventHandler: function(eventName, params)
    {
        params = params || [];
        if (!eventName)
        {
            return;
        }

        this.eventHandlers.forEach(function (eventHandler) {
            if (eventHandler.eventName === eventName)
            {
                eventHandler.handler.apply(this, params);
            }
        }, this);

        if(this.handlers[eventName])
        {
            this.handlers[eventName].apply(this, params);
        }

        var externalEventName = 'b24-sitebutton-' + eventName;
        if (window.BX.onCustomEvent)
        {
            window.BX.onCustomEvent(document, externalEventName,
params);
        }
        if (window.jQuery && typeof(window.jQuery) === 'function')
        {
            var obj = window.jQuery( document );
            if (obj && obj.trigger) obj.trigger(externalEventName, params);
        }
    },
    onWidgetFormInit: function(form)
    {

```

```

        this.execEventHandler('form-init', [form]);
    },
    onWidgetClose: function()
    {
        ButtonManager.unfreeze();
        this.show();
    },
    getTrace: function(options)
    {
        if (!this.b24Tracker.guest)
        {
            return null;
        }

        options = options || {};
        options.channels = options.channels || [];
        options.channels
        [this.config.tracking.channel].concat(options.channels);
        return this.b24Tracker.guest.getTrace(options);
    }
};
})();

```

=