МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра «Програмної інженерії»

ЗВІТ

з лабораторної роботи №3

з дисципліни « Аналіз та рефакторінг коду »

на тему: «Розробка клієнтської/frontend частини»

Виконала:

Студентка групи ПЗПІ-18-4

Янковська Наталія Сергіївна

Перевірив:

Старший викладач кафедри програмної інженерії

Сокорчук Ігор Петрович

Харків 2020

3 РОЗРОБКА КЛІЄНТСЬКОЇ/FRONTEND ЧАСТИНИ ДЛЯ ПРОЕКТУ З ТЕМИ «ПРОГРАМНА СИСТЕМА ДЛЯ АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ПОВ’ЯЗАНИХ З УТИЛІЗАЦІЄЮ СМІТТЯ»

3.1 Мета роботи

Метою виконання даної лабораторної роботи є отримання навичок зі створення клієнтської/frontend частини для програмної системи для автоматизації процесів пов’язаних з утилізацією сміття, а саме розробка користувацького інтерфейсу, який повинен підтримувати локалізацію та інтернаціоналізацію, адміністрування системи, взаємодія з системою та розробка діаграм.

3.2 Хід роботи

3.2.1 UML діаграма прецедентів(див. рис. 3.1), яка відображає взаємодію з системою та описує її головні функції, а саме те, що після авторизації, користувач може передавати та змінювати дані, а також касир може створювати чеки, обслуговуючи покупців.

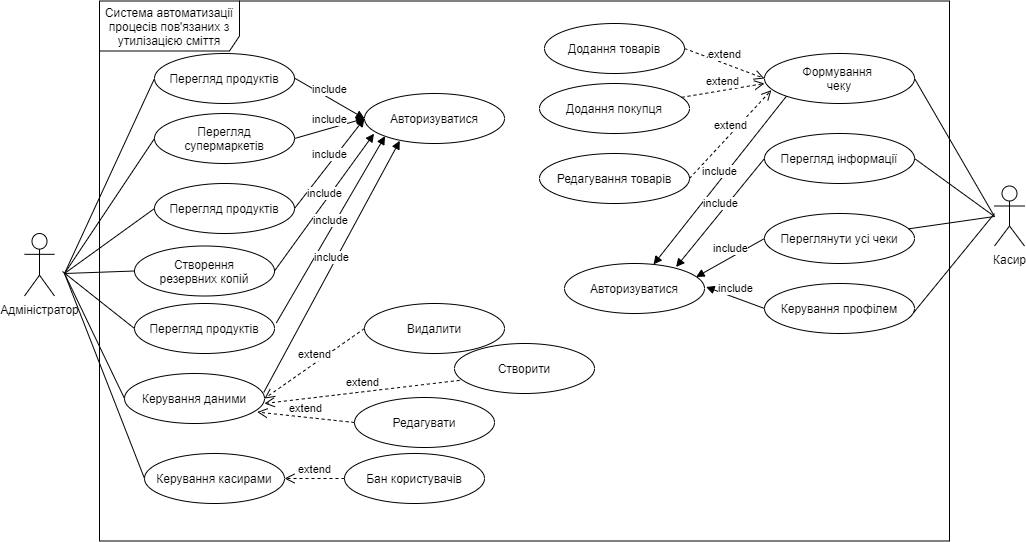


Рисунок 3.1 - UML діаграма прецеденті

3.2.2 Авторизація

Щоб користуватися усіма можливостями сайту, спочатку користувач повинен авторизуватися. Авторизація виконується за допомогою функції авторизаціїя, яка перевіряє введені дані на правильність, а також за допомогою бібліотек “Vuex” та “View-Router”, які дозволяють ставити блок на перехід сторінок, якщо користувач ще не авторизувався. Код наведено у Додатку А.

3.2.3 Використання API запитів

Згідно архітектурного підходу REST було реалізовано функціонал на клієнтській частині. Було використано основні методи HTTP: GET, POST, DELETE, з їх допомогою виводимо дані про продукти, супермаркети, чеки, касирів. Щоб мати можливість надсилати запити до серверу використовується JavaScript бібліотека Axios - HTTP-клієнт, заснований на промісах і призначений для браузерів і для Node.js. Код наведено у додатку Б.

3.2.4 Реалізація бізнес логіки

Реалізувала можливість створення касиром чеку, в якому міститься покупець, який створював покупку, продукти, які він купує, а також ціну за продукти та утилізацію пакування. Ціна за утилізацію пакування залежить від декількох показників, а саме від виду пакування та міста, де товар купується. Система аналізує усі потрібні дані та додає до вартості продукту вартість за утилізацію пакування, якщо воно існує. Код наведений у Додатку В.

3.2.6 Локалізація

Інтерфейс системи було локалізовано двома мовами: українською та англійською. Система забезпечує переклад основних термінів сторінок, одиниці вимірювання та підтримку різних кодових символів. Для реалізації було розроблено бібліотеку з локалізованими даними. Зверху на сторінці знаходяться два прапори - український та англійський, за допомогою яких можна переключати мову локалізації. Код наведено у Додатку Г.

3.2.5 UML діаграма компонентів

Було розроблено діаграму компонентів, яка відображає залежності між компонентами програмної системи(див. рис. 3.2). Веб-застунок було розроблено за допомогою Vue.js - JavaScript-фреймворк з відкритим вихідним кодом для створення користувацьких інтерфейсів. Vue.js легко інтегрується в проекти з використанням інших JavaScript-бібліотек(“vxios”, “vue-router”, “vuex”, “vue-cli”, “vue-flag-icon”). Функціональність екземплярів може бути розширена за допомогою компонентів, які зберігаються в деревовидному вигляді. Компоненти відрізняються від екземплярів тим, що вони не прив'язані до вузлів HTML, а містять власний шаблон HTML.

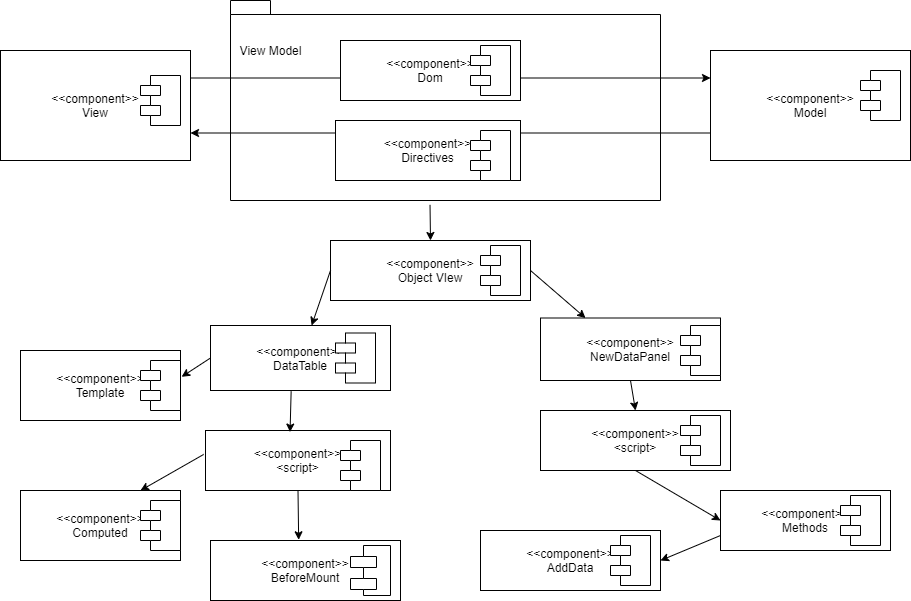


Рисунок 3.2 - UML діаграма компонентів

3.2.7 UML діаграма станів

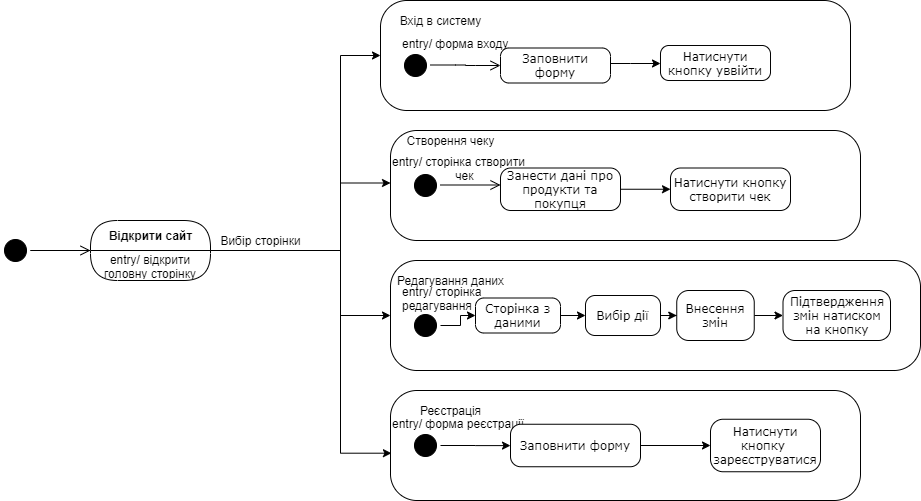


Рисунок 3.3 - UML діаграма станів

3.2.8 UML діаграма діяльності

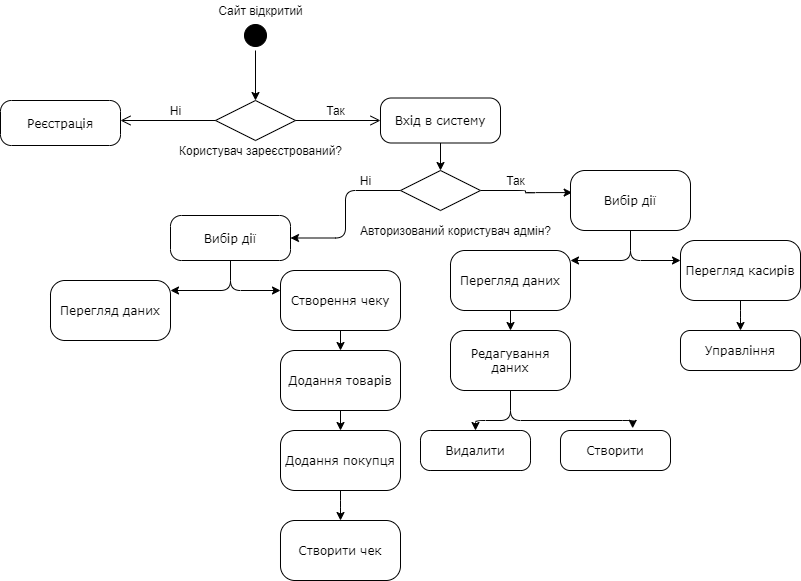


Рисунок 3.4 - UML діаграма діяльності

3.2.9 Створення контрольної суми за алгоритмом md5

Контрольна сума архіву з проектом : 81b18e2c80126c18e6ec5ab5d9b9cbf5

Посилання на файл з кодом на гугл-диску: https://drive.google.com/file/d/1GEa\_QFkLgfDfyMFwNN7coDixJpkBLvFr/view?usp=sharing

3.2.10 Висновки

У ході виконання лабораторної роботи будо отримано навичок зі створення клієнтської/frontend частини для програмної системи для автоматизації процесів пов’язаних з утилізацією сміття, а саме розробка користувацького інтерфейсу, який повинен підтримувати локалізацію та інтернаціоналізацію, адміністрування системи, взаємодія з системою та розробка діаграм.

ДОДАТОК А

Код авторизації користувача

1. const store = new Vuex.Store(
2. {
3. state: {
4. authenticated: false
5. },
6. mutations: {
7. setAuthentication(state, status) {
8. state.authenticated = status;
9. }
10. }
11. }
12. );
13. const routes = [
14. {
15. path: "/",
16. redirect: {name: "login"}
17. },
18. {
19. path: '/supermarkets',
20. name: 'supermarkets',
21. component: Supermarkets,
22. beforeEnter: (to, from, next) => {
23. if(store.state.authenticated == false) {
24. next(false);
25. } else {
26. next();
27. }
28. }
29. },
30. {
31. path: '/check',
32. component: Check,
33. name: "check",
34. params: true ,
35. beforeEnter: (to, from, next) => {
36. if(store.state.authenticated == false) {
37. next(false);
38. } else {
39. next();
40. }
41. }
42. },
43. {
44. path: '/products',
45. component: Products,
46. beforeEnter: (to, from, next) => {
47. if(store.state.authenticated == false) {
48. next(false);
49. } else {
50. next();
51. }
52. }
53. },
54. {
55. path: '/login',
56. name: "login",
57. component: LogIn
58. },
59. ];

ДОДАТОК Б

Код посилання запитів на сервер за допомогою Axios

1. import axios from 'axios';
2. export default () => {
3. return axios.create({
4. baseURL: '<http://localhost:3000>',
5. withCredentials: false,
6. headers: {
7. Accept: "application/json",
8. "Content-Type": "application/json"
9. }
10. });
11. }
12. import API from '@/services/api.js';
13. API().get('/products').then((response) =>{
14. this.products = response.data.data;
15. }),
16. API().get('/customers').then((response) =>{
17. this.customers = response.data.data;
18. }),
19. API().post('/checks', {cashier\_id: cashier\_id,
20. customer\_id: customer\_id}).then((response) =>{
21. check\_id = response.data.data.id;
22. }

ДОДАТОК В

Форма створення чеку

1. <template>
2. <div id="app">
3. <router-view></router-view>
4. <div class = "nextpers">
5. <ul class="rectangle">
6. <li v-for="product in products" v-bind:key="product.id">
7. <p>{{$t('name')}}: {{product.name}}</p>
8. <p>{{$t('price')}}: {{product.price}} {{$t('currency')}}</p>
9. <p><input :id="product.id" type="number" size="3" name="sock1"
10. min="0" max="10" value="0"></p>
11. </li>
12. </ul>
13. <p><label for="customer">{{$t('customer')}}</label></p>
14. <select id="customer">
15. <option value="customer">{{$t('chooseCustomer')}}</option>
16. <option :value="customer.id" v-for="customer in customers"
17. v-bind:key="customer.id">{{customer.name}} {{customer.surname}}
18. </option>
19. </select>
20. <p></p>
21. <button class="cute\_button" v-on:click="create\_check(9)" >
22. {{$t('createCheck')}}</button>
23. </div>
24. </div>
25. </template>
26. methods: {
27. create\_check: function (cashier\_id) {
28. var e = document.getElementById("customer");
29. var customer\_id = e.options[e.selectedIndex].value;
30. var check\_id = 0;
31. API().post('/checks', {cashier\_id: cashier\_id, customer\_id: customer\_id}).then((response) =>{
32. var products = [];
33. API().get('/products').then((response) =>{
34. products = response.data.data;
35. for (let i = 0; i< products.length; i++ ) {
36. var count = document.getElementById(products[i].id).value;
37. if(count > 0){
38. API().post('/check\_items', {check\_id: check\_id, product\_id: products[i].id,
39. count: count}).then((response) =>{
40. console.log(response.data.data);
41. })
42. }
43. }
44. this.$router.push({name: 'ShowCheck', params: { check\_id } });
45. });
46. });
47. }
48. },

ДОДАТОК Г

Код локалізації сторінок веб-застосунку

1. {
2. "menuProducts": "Продукти",
3. "menuSupermarkets": "Супермаркети",
4. "menuChecks": "Чеки",
5. "menuLogOut": "Вийти",
6. "name": "Назва",
7. "address": "Адреса",
8. "price": "Ціна",
9. "personName": "Ім`я",
10. "surname": "Прізвище",
11. "logIn": "Вхід",
12. "login": "Логін",
13. "password": "Пароль",
14. "addSupermarket": "Додати супермаркет",
15. "customer": "Покупець",
16. "chooseCustomer": "Оберіть покупця",
17. "currency": "₴",
18. "createCheck": "Створити чек",
19. "product": "Продуки",
20. "total": "Оплата",
21. "count": "Кількість",
22. "utilizationPrice": "Ціна за утилізацію",
23. "checkNumber": "Номер чеку"
24. }
25. {
26. "menuProducts": "Products",
27. "menuSupermarkets": "Supermarkets",
28. "menuChecks": "Checks",
29. "menuLogOut": "LogOut",
30. "name": "Name",
31. "address": "Address",
32. "price": "Price",
33. "personName": "Name",
34. "surname": "Surname",
35. "logIn": "Login",
36. "login": "Login",
37. "password": "Password",
38. "addSupermarket": "Add Supermarket",
39. "customer": "Customer",
40. "chooseCustomer": "Choose customer",
41. "currency": "$",
42. "createCheck": "Create check",
43. "product": "Product",
44. "total": "Total",
45. "count": "Count",
46. "utilizationPrice": "Utilization Price",
47. "checkNumber": "Сheck number"
48. }
49. export const i18n = new VueI18n({
50. locale: 'en',
51. fallbackLocale: 'ua',
52. messages:{
53. en
54. }
55. })
56. <div class = 'flags'>
57. <a href ="#" @click="setLocale('en')"><flag iso="us" size='big'/></a>
58. <a href ="#" @click="setLocale('ua')"><flag iso="ua" size='big'/></a>
59. </div>
60. import(`./langs/${locale}.json`).then((msgs) => {
61. this.$i18n.setLocaleMessage(locale, msgs)
62. this.$i18n.locale = locale
63. })
64. }