

УДК 004.42

РОЗРОБЛЕННЯ ДОДАТКУ ПО НАДАННЮ ПОБУТОВИХ ПОСЛУГ

О.М. Павлюк, А-О.А. Стронціцька

*Національний університет „Львівська політехніка”
вул. С. Бандери, 12, Львів 79013, Україна*

Розроблено веб-додаток для побутових послуг. Проаналізовано принципи роботи, переваги та недоліки програмних продуктів-аналогів. Побудовано дерево цілей. Був наданий огляд використовуваних технологій, і підтверджено вибір конкретного набору інструментів та фреймворків для створення продукту.

Досліджено процес та підходи розробки веб-додатку за допомогою популярних фреймворків, таких як Spring та Angular. Вивчено на практиці інструменти для створення бекенд сервера та дизайну сайту, а саме - SpringBoot, Angular та Bootstrap, оскільки вони є найпопулярнішими і найбільш швидкісними, особливо з Spring і Angular в своїй основі. Також було проведено аналіз з перевагами та недоліками уже існуючих програмних продуктів. Зокрема було розглянуто такі платформи як Linkedin та Elevated carriers – одні з найбільших гігантів у сфері веб-додатків з пошуку роботи та наданням послуг.

Ключові слова: веб-додаток, Spring Boot, Angular, SDLC, CodePen, підхід Model-View-Controller, формат JSON, фронтенд-клієнт, сервер, побутові послуги.

Постановка проблеми. На сьогоднішній день все більш і більш популярним стає пошук роботи в Інтернеті. І така статистика стосується не лише сфери ІТ. 93% усіх рекрутерів використовують різні соціальні мережі для пошуку кандидатів. Але не в усіх сферах людської діяльності є потреба посередника для організації співпраці між роботодавцем та працівником. Зокрема до таких сфер можна віднести і побутові послуги. Особливо поширеною є ситуація, коли роботодавець не планує вести бізнес на постійній основі, а потребує допомоги лише в якійсь одиничній ситуації. Хорошою ідеєю буде скомбінувати усталений вже на сьогоднішній день метод Інтернет-відбору кандидатів на певну позицію з підходом, коли роботодавець може звертатись за послугами працівника на непостійній основі за потреби і обов'язково з використанням новітніх технологій. Таким чином обидві сторони бізнесу зекономлять чимало часу, не кажучи вже про гроші. З іншого боку працівники замість того, щоб надсилати своє резюме кожному роботодавцю не залежно від вакансії зможуть знайти того, якому підійдуть за досвідом та специфікою роботи.

Дуже актуальним прикладом послуг на непостійній основі залишаються побутові послуги. Ідея власноруч займатися благоустроєм власного будинку багатством може видатись досить привабливою. Адже дозволить заощадити певну суму грошей та насолоджуватися роботою, яка зроблена своїми руками. Але в реальності деякі речі краще довірити експертам. Професійні робітники вирішать навіть складні та трудомісткі завдання. Незалежно від того, чи планується ре-

монт усього житла чи якоїсь його частини- часто доводиться робити розумний вибір між тим, чи робити все самостійно, чи наймати когось та платити певну грошову суму. Дуже часто вибір на користь першого варіанту робиться через те, що клієнти бояться бути ошуканими недобросовісними працівниками, не готові платити певну «круглу» суму, яку запропонував за свої послуги перший випадковим чином вибраний працівник або ж взагалі не знають, де знайти людину, яка знає, як допомогти з конкретною проблемою по господарству.

Саме через описані вище проблеми і виникла потреба у розробці додатку, який зможе допомогти людям із вирішенням їхніх побутових проблем і водночас буде генератором нових вакансій для кваліфікованих працівників, які шукають роботу. Щоб додаток був швидкісним і надійним – він має реалізовувати добре продуману бізнес-логіку і використовувати новітні технології на рівні програмного забезпечення.

У даній роботі розглянуто основні кроки процесу розробки веб-сайту з надання побутових послуг за допомогою популярних фреймворків, таких як Spring та Angular. Це веб-сайт для клієнтів, які стикаються з деякими побутовими проблемами та потребують когось компетентного для вирішення цих питань. З іншого боку, кваліфіковані ремонтники, які зараз шукають роботу, також можуть знайти тут роботодавця. Щоб полегшити процес пошуку роботодавця або працівника, додаток забезпечує різні види пошуку по даних.

Аналіз останніх досліджень та публікацій. Перед тим, як почати розробку власного програмного продукту доцільно проаналізувати уже існуючі програмні продукти. Найбільше уваги заслуговують два дуже популярні проекти, які також спеціалізуються на наданні послуг. Одним з нових і перспективних представників у галузі пошуку роботи є платформа «Elevated careers». Elevated Careers допомагає співставляти працівників з роботодавцями на основі сумісності. Це, безумовно, унікальне налаштування яке має свої переваги і недоліки.

«Elevated careers» відрізняється від усіх інших програмних продуктів з пошуку роботи, оскільки використовує алгоритми прогнозування для визначення рівня відповідності працівників та пропозицій замовників завдань на основі трьох різних типів сумісності – навиків, досвід, характер. Це унікальний підхід який враховує набагато більше моментів, ніж типові елементи пошуку роботи, такі як: місце розташування, зарплата та посада. Як результат, в людини буде набагато більше шансів здобути роботу яка буде її задовольняти і водночас приносити користь.

З іншого боку, «Elevated careers» вимагає великого вкладу з боку користувача. Завантаживши своє резюме, потрібно опісля перейти до розділу «Створення профілю», і заповнити опитувальник який складається з великої кількості питань. Для того, хто не має багато часу, цей тривалий процес може бути не надто приємним.

Наступним, не менш важливим програмним продуктом, який було взято за основу і проаналізовано під час розробки – LinkedIn. По своїй суті, LinkedIn – це професійний інструмент для полегшення спілкування між працівниками,

представниками бізнесу та клієнтами. І хоча LinkedIn - це набагато більше, ніж просто сайт з переліком вакансій, одним з його ключових аспектів все-таки є функціональність пошуку роботи.

Беззаперечною перевагою LinkedIn є його величезна популярність. Реєструючись на LinkedIn, можна бути впевненим, що будь-який бізнес, з яким є бажання працювати, уже є там і готовий спілкуватись. Але з різних причин великі компанії не завжди бажають розміщувати вакансії на LinkedIn, боячись, що це зашкодить їх репутації. Тут справа більше в тому, що LinkedIn позиціонує себе не тільки як система для пошуку роботи, а й як дуже популярна соціальна мережа – тому тут потрібно бути обережним, щоб не сплутати бізнес-процеси з повсякденним спілкуванням.

Мета статті. Розробити компоненту веб-орієнтованої інформаційної технології для надання побутових послуг для збору опрацювання та відображення інформації, оглянути відповідну інформацію пов'язану із розробкою схожих програмних продуктів, дослідити інфраструктурну можливість скомбінувати дві найбільш популярні технології Java розробки для створення унікального продукту, який приносить прибутки

Виклад основного матеріалу дослідження. Розроблювана система використовує досвід програмних продуктів-попередників і застосує його на практиці в контексті сфери побутових послуг. Щоб досягти успіху у розробці варто правильно розставити пріоритети. У цьому неабияк допомагає створення дерева цілей.

Вибір методології життєвого циклу розробки програмного забезпечення (SDLC) є складним завданням для багатьох інженерів програмного забезпечення. Перш ніж ставити собі якісь рамки для вибору заданої методології SDLC, програмний інженер має визначити різні типи та проілюструвати для себе переваги та недоліки цих моделей.

На рисунку 1 зображено дерево цілей. В ньому описані підходи які мають найбільший пріоритет для реалізації поставленого завдання. Об'єктивно розроблене дерево цілей важливе для розробників дизайну проекту та відслідковування успіху на шляху до фінальної стадії проекту. Також при створенні дерева цілей чим конкретнішими є цілі-тим кориснішим буде розроблюване дерево. Добре додавати нові цілі, які спадають на думку під час розробки програмного продукту. Адже таким чином можна так само і вбачати нові проблеми – які можна в перспективі доєднати до дерева проблем. Таким чином можна отримати повноцінну картину усіх процесів, що відбуваються під час життєвого циклу розробки програмного продукту.

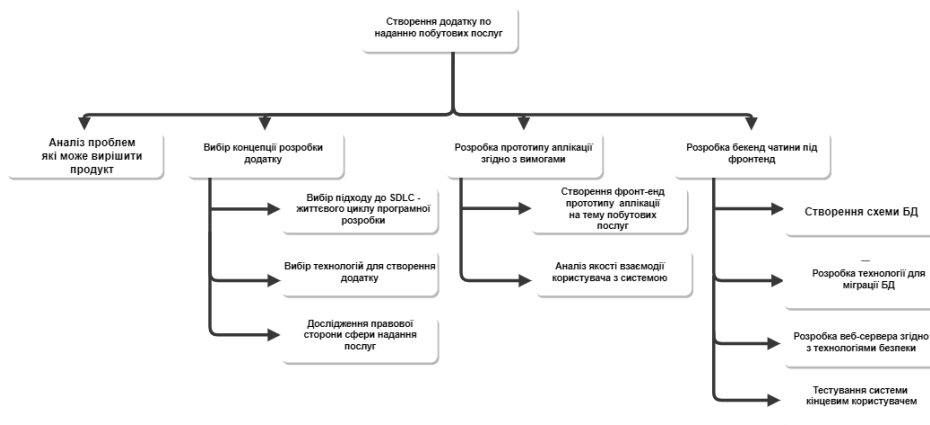


Рис.1 Дерево цілей

Вибравши правильний підхід до життєвого циклу розробки – варто перейти до вибору технологій. Щоб зробити цей вибір максимально правильним потрібно уточнити і проговорити декілька наступних моментів. Перш за все треба добре зрозуміти – чи розроблюваний додаток вирішує якусь унікальну проблему? Адже більшість програм (навіть з унікальною бізнес ідеєю) можуть бути реалізовані за допомогою сталих комбінацій технологій. До таких відносяться і розроблюваний додаток. Попри новаторство в концепції – програмний продукт не потребує якогось особливого підходу з точки зору підтримки та розробки програмного забезпечення. Коли якась технологія добре спрацювала для схожого програмного продукту, то вона не завжди добре відпрацює для розроблюваного. Саме тому попри усталений набір програмних засобів для розробки – їхнє використання було чітко регламентовано. Адже рідко коли корисно копіювати наосліп технологічні рішення інших програмних інженерів.

Після правильного вибору технологій варто приступити безпосередньо до розробки кістяка фронтенд частини аплікації. Для цього було використано відомий на сьогодні фреймворк від компанії Google – Angular 7. Даний фронтенд-клієнт повністю сумісний з Spring Boot, який був використаний для розробки бекенд частини. Він містить усю логіку підходу Model-View-Controller, тобто розділення на рівень перегляду, рівень контролера та рівень моделі. Після початкового запуску програми лише даними у форматі JSON будуть обмінюватись між собою фронтенд-клієнт і сервер.

Spring Boot та Angular у поєднанні дійсно пропонують абсолютно новий спосіб створення веб-додатків. Зокрема спостерігається значне підвищення продуктивності, яке забезпечує цей підхід. Також підтримується концепція Stateless архітектури - тобто відсутності обміну даними про стан сервера між запитами (окрім файлів cookie чи автентифікації) – що дозволяє уникати цілого ряду помилок [5].

Майже все в Angular є компонентами. Навіть якщо говорити про загальний дизайн розроблюваної single-page програми-це набір компонентів. Є component tree, який дозволяє їм комунікувати між собою. Є й інші аспекти Angular, які

не мають компонентної структури, зокрема: дані з форм, наприклад, надходять через input елементи, а відображаються на сторінці через output елементи. Кожен компонент має життєвий цикл, яким керує сам фреймворк Angular [8]. Навіть при їх оголошенні та створенні автоматично застосовується даний життєвий цикл для ініціалізації програми та для знищення об'єктів.

Для розробки бекенд-частини використовувався дуже популярний серед Java розробників фреймворк - Spring Boot. Загалом це проект, побудований на основі фреймворку Spring [3]. Він пропонує розробнику більш простий і швидкий спосіб налаштування та запуску як простих, так і складних веб-програм. Адже у базовому фреймворку Spring потрібно налаштовувати практично все самостійно. Отже, у розробника багато часу може піти на написання файлів конфігурації, таких як дескриптори XML [1]. Це одна з основних проблем, яку вирішує Spring Boot. Він розумним чином (через Maven-repository наприклад) вибирає певні необхідні для аплікації залежності, автоматично налаштовує весь функціонал, який розробник планує використовувати і опісля можна легко запустити готовий бекенд-сервер. Крім того, використання Spring Boot-ом вбудованого сервера (Tomcat, Jetty, тощо) також спрощує процес розгортання програми на сервісах Amazon чи Google Cloud Platform [4].

Але і це ще не всі переваги Spring Boot, які активно використовувались у процесі розробки. Дуже важливою і корисною властивістю є інтелектуальна автоматична конфігурація, яка намагається налаштувати програму на основі залежностей, які були додані. Можна цю властивість пояснити на прикладі Бази Даних. Якщо додати певну залежність, що стосується бази даних, до pom.xml (конфігураційний файл, якщо використовується Maven), Spring Boot передбачить, що, ймовірно, планується використовуватись якась БД.[2] Після цього фреймворк автоматично налаштовує конфігураційні файли доступу до Бази Даних [6]. До того ж забезпечити таку автоматизованість надзвичайно просто. Потрібно лише додати анотацію `@EnableAutoConfiguration` до програми Spring Boot.

Як вже було описано вище - у розробці фронтенд частини використовувався фреймворк Angular. Крім того використовувався уже доступний кістяк, який був кастомізований під потреби замовника. До цього процесу залучався CodePen - це інтернет-спільнота для тестування та демонстрації фрагментів коду HTML, CSS та JavaScript, створених користувачем. Головна сторінка створювалась за допомогою саме ресурсів сайту CodePen.

Домашня сторінка - це своєрідний початок знайомства клієнта з веб-додатком. Тому варто зробити її максимально простою. Оскільки домашня сторінка часто є "відправною точкою" клієнта, було зроблено все, щоб навігація була чіткою та зрозумілою. Користувач має доступ до всіх важливих розділів веб-сайту по наданню побутових послуг з домашньої сторінки. Зокрема, якщо прогорнути сторінку нижче - то можна буде побачити список активних постачальників послуг веб-сайту. Також залогований користувач матиме змогу переглянути непрочитані повідомлення від інших користувачів у правому верхньому куті. До того ж, якщо до Користувача хтось звернеться з пропозицією роботи, то нагадування прийде також. (Рис.2 а)

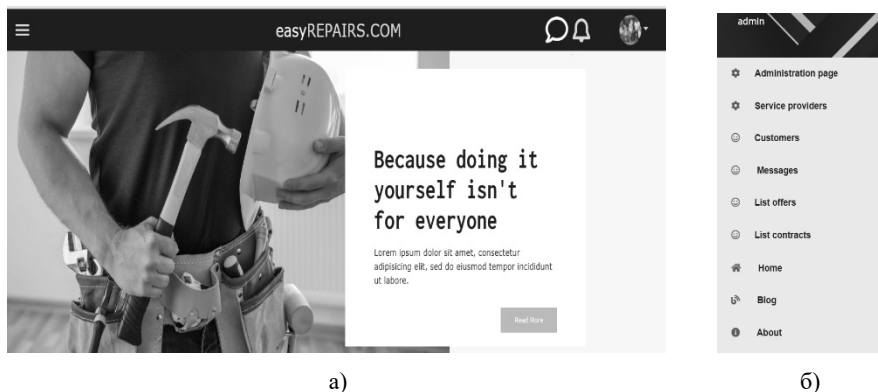


Рис.2 а) Головна сторінка сайту, б) Burger menu сайту

Для того, щоб зекономити трохи місця на головній сторінці, було прийнято рішення використати burger menu (див. рис. 2 б). У цьому меню користувач зможе швидким чином перейти на ті частини сторінки, які є йому потрібними. Зокрема це чат з іншими користувачами, а також посортовані списки профілів постачальників послуг та замовників. Також можна буде перейти на персональні портфоліо замовників, в яких вони презентуватимуть попередньо зроблені замовлення.

Модулі допомагають організувати додаток у згуртовані блоків функцій, обгортаючи компоненти, директиви та послуги. Angular заохочує принцип singleresponsibility, контролюючи кожен патч сторінки за допомогою власного компонента [7]. Саме за допомогою технології модульності і було розроблено профіль сервіс провайдерів на сайті. (Рис.3)

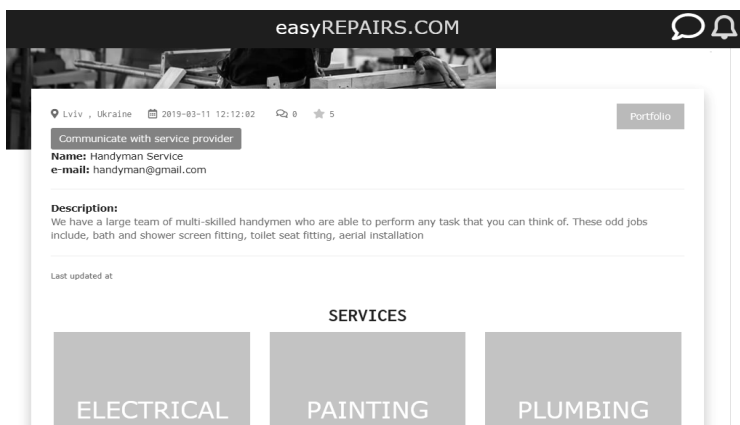


Рис.3 Профіль сервіс провайдерів

Також було розроблено спеціальну сторінку виключно для адміністратора веб-сайту. Коли адміністратор на своїй сторінці захоче видалити певні дані, робитиметься автоматизований бекап бази даних. Створення бекапу обґрунтоване забезпеченням попереднього збереження даних, які можуть бути втрачені. Адже

можливі такі випадки, коли адміністратор випадково зробить якесь оновлення чи видалення і захоче деактивувати операцію. Також він матиме можливість заблокувати активність певних користувачів на визначений час, оновити певні поля, змінити пароль, завантажити нову профільну картинку користувачеві тощо (див. рис. 4).

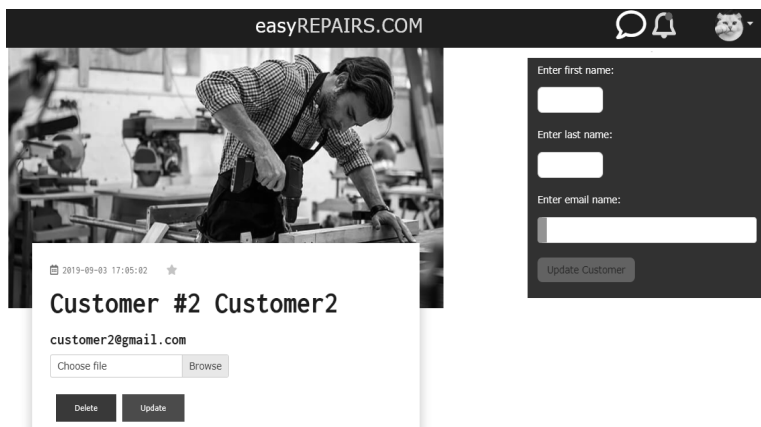


Рис. 4 Адміністраторська сторінка

На формі оновлення полів даних про користувача було реалізовано спеціальну валідацію емейлу а також даних про ім'я та прізвище на наявність характерних символів. На цій же сторінці по аналогії з списком постачальників послуг головної сторінки – адміністратор матиме можливість посортувати користувачів на бічній панелі.

Висновки. У даній роботі було розглянуто методи роботи з різними технологіями для розробки веб-додатків. На початкових стадіях розробки було розроблено дерево цілей для обрання правильних підходів розробки, проведено аналіз задачі проекту та проведено дослідження технологій, які застосовуватимуться для розробки. Наступним кроком було створення програми за допомогою різних фреймворків, зроблено їх короткий огляд. Саме за допомогою технологій Angular, Spring та її надбудови SpringBoot було розроблено веб-сайт по наданню побутових послуг для клієнтів, які стикаються з деякими побутовими проблемами та потребують компетентного вирішення цих питань, а з іншого боку, кваліфіковані ремонтники, які зараз шукають роботу, також можуть знайти на веб-сайті роботодавця. Процес розробки такого проекту дає можливість правильно вибирати та успішно використовувати методи розробки повноцінних веб додатків. Також було досліджено базові підходи до організації процесів розробки програмного продукту – зокрема побудову дерева цілей та визначення правильного типу життєвого циклу програмного продукту.

Список використаних джерел

1. Craig Walls Spring in Action. - London, 2010. - 726 с.
2. Документація по Spring. [Електронний ресурс]–Режим доступу: <https://spring.io/guides/gs/rest-service/>
3. Iuliana Cosmina Spring Pro 5. - London, 2017 - 849 с.

4. Codepen. [Електронний ресурс] – Режим доступу: <https://codepen.io/>
5. Vaskaran Sarcar Java Design Patterns. – London, 2016 – 171с.
6. Ben Forta SQL in 10 Minutes a Day. –USA, 2016 – 306 с.
7. Vijay Nair Practical Domain-Driven Design in Enterprise Java. –USA, 2016 – 396 с.
8. Shyam Seshadri Angular: Up and Running. –USA, 2018 – 341 с.

REFERENCES

1. CraigWalls (2010). Spring in Action. - London - 726 p. (in English)
2. Documentation. [Resource]–Retrieved from: <https://spring.io/guides/gs/rest-service/> (in English)
3. IulianaCosmina (2017). Spring Pro 5. - London - 849 p. (in English)
4. Codepen. [Resource] – Accessed: <https://codepen.io/> (in English)
5. Vaskaran Sarcar (2016). Java Design Patterns– London – 171p. (in English)
6. Ben Forta (2016). SQL in 10 Minutes a Day. –USA – 306 p. (in English)
7. Vijay Nair (2016). Practical Domain-Driven Design in Enterprise Java. –USA– 396 p. (in English)

DOI 10.32403/2411-9210-2021-2-46-55-62

DEVELOPMENT OF A WEB-APPLICATION FOR THE PROVISION OF HOUSEHOLD SERVICES

O.M. Pavliuk, A-O.A. Strontsitska

Lviv Polytechnic National University

12, Bandera St., Lviv, 79013, Ukraine

olena.m.pavliuk@lpnu.ua

A web application for the household services has been developed. The principles of operation, advantages and disadvantages of similar software products are analyzed. Objective Tree is constructed. An overview of the technologies used is provided, and the choice of a specific set of tools and frameworks for product creation is proved.

The process and approaches of web application development using popular frameworks such as Spring and Angular are studied. Tools for creating a backend server and site design, namely - SpringBoot, Angular and Bootstrap are researched, as they are the most popular and fast, especially with Spring and Angular in its core. The analysis with advantages and disadvantages of already existing software products is carried out. In particular, the platforms such as Linkedin and Elevated carriers are studied – one of the largest giants in the field of web applications for job search and service provision.

Keywords: *web application, Spring Boot, Angular, SDLC, CodePen, Model-View-Controller approach, JSON format, frontend client, server, household services.*

Стаття надійшла до редакції 07.06.2021

Received 07.06.2021