

ПРОЕКТУВАННЯ ТА РОЗРОБКА ВЕБ-ЗАСТОСУНКУ “DIM SIRKA”

Слободян Марія, Падалка Іван,
IV курс ОР бакалавр,
факультет математики та інформатики.
Науковий керівник – Іщеряков С.М.,
кандидат технічних наук, доцент.

На сьогоднішній момент у світі дедалі більшає кількість безпритульних тварин. Кількість волонтерів та організацій, що займаються порятунком безпритульних теж поступово зростає. Важко уявити будь-яку сферу життя людини, де непотрібно використовувати жодних медійних ресурсів чи поширення інформації через інтернет. Саме тому надзвичайно важливо мати широко доступний сервіс, який буде здатен виконувати універсальну функцію, що дозволить зекономити час та зусилля як для волонтерів, так і для людей, що бажають усиновити тварину чи зробити пожертву.

Програмне забезпечення “Dim-Sirka” має стати інформаційним ресурсом, щоб: зручно менеджити кількість тварин в організації за їхніми статусами та характеристиками; мати систему донату та інтерфейс для вибору тварини за бажаними параметрами, характеристиками; проводити швидку комунікацію; поширювати новини про тварин і організацію в цілому.

Для розробки цього програмного забезпечення було поставлено наступні завдання:

- проведення аналізу предметної області;
- аналіз бізнес-вимог та вимог користувача;
- визначення функціональних та нефункціональних вимог, складання специфікації вимог до програмного забезпечення;
- проектування архітектури бази даних та сервісу;
- імплементація та тестування програмного забезпечення.

В ході дослідження аналогічних сервісів, аналізу їх сильних та слабких сторін стало можливим оцінити, які саме проблеми необхідно подолати у нашій системі. Можна чітко сказати, що питання створення унікального програмного забезпечення для організації “Dim Sirka” є актуальним, тому що наявні ресурси не мають відповідного функціоналу або мають ряд недоліків, пов’язаних із функціоналом, який є надзвичайно важливим, щоб полегшити менеджмент тварини як моделі та інтерфейс пошуку.

Аналіз вимог був критичним для успішної поетапної розробки проекту. Спершу було визначено зацікавлених осіб та їх потреби. Визначено набір завдань для користувача, які повинен вирішувати даний веб-застосунок, а також можливі алгоритми їхнього вирішення у програмі. Як результат було визначено основне призначення ПЗ та межі проекту: необхідно було розробити програмне забезпечення, що перейняло би переваги уже наявних веб-застосунків і в той же час надавало користувачу більший функціонал для пошуку тварини, специфічні статуси та систему створення замовлень.

За допомогою “Brainstorming” і “Prototyping” технік [1] ми провели збір вимог до даного програмного забезпечення. Для перевірки цілісності та закінченості вимог було використано “Requirements Reviews” [2] техніку. Функ-

ціональні вимоги, визначені у процесі аналізу, представлено у вигляді наступних документів: SRS (Software Requirements Specification) та діаграма прецедентів (Use Case diagram), що описує набір базових сценаріїв користування сервісом. Також було визначено нефункціональні вимоги, що дали певні обмеження щодо безпеки та надійності ПЗ. Завершальним етапом було проведено тестування вимог таким чином, щоб вони були простими та зрозумілими як для звичайних користувачів, так і для розробників.

Згідно із визначеними функціональними та нефункціональними вимогами було встановлено комплекс завдань, що має вирішити розроблений програмний модуль. Провівши проектування бази даних та програмного забезпечення, було вибрано багаторівневу клієнт - сервер архітектуру. Було визначено три логічні архітектурні шари: рівень представлення, рівень бізнес-логіки та рівень доступу до даних. Направлення залежностей між ними іде від рівня представлення до рівня доступу до даних. Тобто рівень доступу до даних не залежить від інших рівнів, але від нього залежить рівень, що інкапсулює бізнес – логіку. Самі модулі є легкозв'язними (loose coupling), а взаємозв'язки всередині модулів мають високе зчеплення (high cohesion) [3].

Одним із головних завдань при створенні програмного забезпечення був вибір технологій та інструментів для реалізації даного завдання. Для створення серверної частини було обрано мову Java [4] та Spring Boot фреймворк, а саме його модулі: Core Container, Data Access/Integration та Web. Як генератор шаблонів обрано Freemarker, що допомагає створювати динамічні Web-сторінки без дублікації коду. Для підключення всіх модулів та бібліотек до проекту використано Gradle інструмент. Базою даних було обрано PostgreSQL так як вона є гнучкою та надійною.

На етапі реалізації програмного забезпечення було проведено розподіл завдань та оцінку часу на їхнє виконання [5]. Згідно з методологією SCRUM, проект був поділений на ітерації, які називаються спринтами. Перед кожним спринтом ми визначали функціонал, який ми були повинні реалізувати на даному етапі. Основна наша ціль була, щоб протягом спринта вимоги залишались незмінними і завдання були вчасно виконаними. Двічі на тиждень проводились “status” сесії.

Було обрано Git як основну систему контролю версій, яка дозволяє зберігати попередні версії файлів та завантажувати їх за потреби. Для спільної розробки програмного забезпечення та зберігання вихідного коду обрано GitHub, який є одним з найпопулярніших, найбільших веб-сервісів. Для неперервного тестування програмного забезпечення на GitHub було додано Travis CI. Для розгортання проекту було обрано безплатну платформу Heroku. Також було вирішено інтегрувати наш проект з SonarQube для аналізу статичного коду на наявність помилок, запахів коду за їхніми типами та вразливості безпеки [6].

У ході виконання даної роботи було розроблено сервіс для менеджменту тваринами та замовлення. Програму можна використовувати як систему для журналювання тварин у притулку, яка надає функції вибору тварини за різними параметрами та подальший процес комунікації. Публічна частина веб-застосунку доступна усім користувачам з можливістю перегляду списку тварин, пошук за різними параметрами та сортуванням, створення замовлення на тварину,

пожертвування, перегляду архіву тварин, які знайшли сім'ю, зворотна комунікація за допомогою коментарів. Користувач з роллю адміністратор додатково має можливість додавати нові тварини до списку чи редагувати існуючі записи, бачити перелік замовлень, підтверджувати їх або відхиляти.

1. Requirements Elicitation Techniques. URL: <https://www.softwaretestinghelp.com/requirements-elicitation-techniques>.
2. Requirements Validation Techniques. URL: <https://www.geeksforgeeks.org/software-engineering-requirements-validation-techniques>.
3. Loose coupling and high cohesion in multiplayer architecture. URL: <https://medium.com/clarityhub/low-coupling-high-cohesion-3610e35ac4a6>.
4. Іщеряков С. М., Яновський Ю. М., Козленко М. І. Програмування множинного наслідування реалізації на основі анонімних внутрішніх класів Java. *Proceedings of the 2019 Scientific Seminar on Innovative Solutions in Software Engineering* (м. Івано-Франківськ, 10 грудня 2019 р.). Івано-Франківськ, 2019. С. 26–27. URL: <https://doi.org/10.5281/zenodo.4263418>.
5. Agile estimation techniques. URL: <https://www.pmi.org/learning/library/agile-project-estimation-techniques-6110>.
6. Code Smells That Are Found The Most. URL: <https://apiumhub.com/tech-blog-barcelona/code-smells>.

МЕТОДИКИ ВИВЧЕННЯ МОДУЛЯ У ШКІЛЬНОМУ КУРСІ МАТЕМАТИКИ

Тугай Ольга,
І курс ОР магістр,
факультет математики та інформатики.
Науковий керівник – Кравців В.В.,
кандидат фізико-математичних наук, доцент.

Оскільки діти вчать математику, то найчастіше результат нашого навчання потрапляє в когнітивну групу за класифікацією педагога-психолога Бенджаміна Блума.

І ми часто формулюємо мету нашого навчання в контексті “знати...”. Не знав означення модуля, а тепер знає, не знав як розв’язувати рівняння з модулем, а тепер знає. Але знання має різну глибину. Дійсно, є дві дитини які знають означення модуля, але на питання звідки це взялось, одна дитина скаже “так вивчали в школі”, а друга яким чином почали використовувати модуль. Тому ціль навчання залежить від ієрархії розумових процесів. Чи дитина просто запам’ятала чи вона розуміє, чи вона може створити щось нове. Тому хочу продемонструвати цю систему детальніше і показати дії вчителя на кожному з цих рівнів.

Перший рівень: запам’ятовування. На цьому етапі метою є те, щоб учень запам’ятав та зміг відтворити навчальну інформацію. Цей етап ідеальний для ознайомлення з чимось новим. При цьому наша роль тут: спрямовувати, розповідати, показувати. Можна показати модуль як душеву кабінку, в які входять