

УДК 004.4

DOI: 10.31471/1993-9981-2021-1(46)-132-137

ВЕБ-ОРІЄНТОВАНА СИСТЕМА МОНІТОРИНГУ ТА КЕРУВАННЯ ПРОЄКТАМИ ЗА МЕТОДОЛОГІЄЮ AGILE

Т. В. Кривцун, М. О. Слабінога, Я. І. Заячук

Інститут інформаційних технологій, Івано-Франківський національний технічний університет нафти і газу, 76019, м. Івано-Франківськ, вул. Карпатська, 15, e-mail: mslabinoha@gmail.com

Робота присвячена розробці веб-орієнтованої системи підтримки моніторингу виконання та управління невеликими проєктами, що виконуються за методологією agile (зокрема, в галузі інформаційних технологій). Моніторинг виконання проєктів є важливим фактором їх успіху, що в подальшому позитивно впливає на прибутки компанії та економіки країни в цілому. Були проаналізовані існуючі рішення в галузі керування проєктами: розглянуто основні моделі ведення проєктів, а також основні програмні засоби їх реалізації, їх переваги та недоліки. Було сформовано задачу щодо розробки веб-орієнтованої системи моніторингу та керування проєктами за методологією Agile. Було вибрано програмні засоби реалізації системи та спроектовано структуру бази даних. Програмне забезпечення серверної частини було реалізовано з використанням архітектури "модель - вигляд - контролер", для чого було розроблено UML-діаграми відповідних компонентів системи. Розроблене серверне програмне забезпечення використовує програмний інтерфейс з відповіддю на запити у форматі JSON, що дозволяє в подальшому розробити крім веб-інтерфейсу ще і мобільні застосунки. Веб-інтерфейс, реалізований засобами фреймворка Angular реалізовує основні функції для користувачів та адміністраторів системи, такі як керування сутностями, авторизація та реєстрація, а також містить в собі аналітичний інструмент для моніторингу виконання проєкту, а саме діаграму згорання. Розроблене програмне забезпечення буде корисним для невеликих проєктів, що використовують Agile-методології як частково, так і повністю, та у яких немає потреби у складних комплексних інструментах, що потребують тривалого процесу налаштування. Реалізація даної системи як веб-орієнтованого додатку дає змогу використовувати її як зі стаціонарних комп'ютерів, так і з мобільних пристроїв.

Ключові слова: керування проєктами, agile-методологія, веб-орієнтована система, діаграма згорання, клієнт-серверна архітектура.

Работа посвящена разработке веб-ориентированной системы поддержки мониторинга выполнения и управления небольшими проектами, которые ведутся по методологии agile (в частности, в области информационных технологий). Мониторинг выполнения проектов является важным фактором их успеха, что в дальнейшем положительно влияет на доходы компании и экономики страны в целом. Были проанализированы существующие решения в области управления проектами: рассмотрены основные модели ведения проектов, а также основные программные средства их реализации, их преимущества и недостатки. Было сформирована задача по разработке веб-ориентированной системы мониторинга и управления проектами по методологии Agile. Были выбраны программные средства реализации системы и спроектирована структура базы данных. Программное обеспечение серверной части было реализовано с использованием архитектуры "модель - вид - контроллер", для чего были разработаны UML-диаграммы соответствующих компонентов системы. Разработанное серверное программное обеспечение использует программный интерфейс с ответом на запросы в формате JSON, что позволяет в дальнейшем разработать кроме веб-интерфейса еще и мобильные приложения. Веб-интерфейс, реализованный средствами фреймворка Angular реализует основные функции для пользователей и администраторов системы, такие как управление сутностями, авторизация и регистрация, а также содержит в себе аналитический инструмент для мониторинга выполнения проекта, а именно диаграмму сгорания. Разработанное программное обеспечение будет полезным для небольших проектов, использующих Agile-методологии как частично, так и полностью, и в которых нет необходимости в сложных комплексных инструментах, требующих длительного процесса настройки. Реализация данной системы как веб-ориентированного приложения позволяет использовать ее как со стационарных компьютеров, так и с мобильных устройств.

Ключевые слова: управление проектами, agile-методология, веб-ориентированная система, диаграмма сгорания, клиент-серверная архитектура.

The work is devoted to the development of a web-based support system for monitoring the implementation and management of small projects implemented according to the agile methodology (in particular, in the field of information technologies). Monitoring the implementation of projects is an important factor in their success, which in turn has a positive impact on the company's profits and the economy as a whole. Existing solutions in the field of project management were analyzed: the main models of project management were considered, as well as the main software tools for their implementation, their advantages and disadvantages. The task of developing a web-based system for monitoring and project management according to the Agile methodology was formed. The software tools for the system implementation were selected and the database structure was designed. The software of the server part was implemented using the architecture "model - view - controller", for which UML-diagrams of the relevant components of the system were developed. The developed server software uses a software interface with response to requests in JSON format, which allows you to further develop in addition to the web interface and mobile applications. The web interface implemented by the Angular framework implements basic functions for users and system administrators, such as entity management, authorization and registration, and includes an analytical tool for monitoring project performance, namely the burn-out chart. The developed software will be useful for small projects that use Agile methodologies, both partially and completely, and which do not require complex tools that require a long setup process. Implementing this system as a web-based application allows you to use it from both desktop computers and mobile devices.

Keywords: project management, agile-methodology, web-oriented system, burn-out chart, client-server architecture.

Моніторинг та керування проектами (зокрема, в галузі інформаційних технологій), є важливим фактором успішного розвитку бізнесу та економіки країни в цілому. Так, згідно звіту Standish Group[1], середня частка успішних проектів протягом різних років коливалася в межах 30-35%, проектів, у яких виникали труднощі - 45-50%, повністю провалених проектів - 15-20%. Високий відсоток проектів, які не є успішними, тягне за собою збільшення витрат на утримання проектів, зниження потенційних прибутків та відтік інвестицій. Саме тому задача моніторингу та керування процесу виконання проекту є дуже важливою.

На сьогоднішній день, існує велика кількість методологій та моделей виконання проектів: інкрементна модель, V-модель, b-модель, каскадна модель, тощо [2]. Кожна з моделей має свої переваги та недоліки, однак будь-яке виконання проекту згідно таких моделей має під собою одне і те ж підґрунтя - розділення великих задач на дрібні завдання, конверсія їх в технічні вимоги та подальша реалізація і апробація. Найпопулярнішими, при цьому, для проектів в галузі інформаційних технологій, залишаються інкрементні моделі, зокрема методологія Scrum, що відноситься до agile-методологій. Так, за даними Forbes[3], згідно проведеного опитування, близько 50% респондентів з різних компаній зазначили, що використовують Scrum у більш ніж половині проектів. Згідно того ж опитування, частка успішних проектів, що велися за допомогою

Scrum, складає 62%, що пояснюється тим, що Scrum дозволяє відносно легко моніторити основні показники ефективності команди при реалізації проекту [4-5].

Для програмної підтримки методологій Agile реалізована велика кількість програмних продуктів, зокрема Jira та Redmine. Ці програмні продукти забезпечують комплексний підхід до ведення великих проектів та моніторингу роботи команди. При цьому, слід зазначити, що обидва ці продукти є доволі складними у налаштуванні, мають багато надбудов та часто є надто складними для ведення малих проектів, що виконуються за 2-3 спринти. Крім того, малі проекти часто застосовують технологію Scrum фрагментарно, користуючись тільки окремими її артефактами (спринтами, командними ролями, беклогом), а також мають незначні відхилення по термінах виконання. Тому розробка простого та інтуїтивно зрозумілого програмного забезпечення, що підтримує моніторинг та керування проектом за методологією Scrum, є важливим завданням, яке і лягло в основу технічних вимог до програмного продукту, що розроблявся.

Розробку програмного забезпечення було розділено на дві частини. Для написання серверної частини було вибрано фреймворк Laravel, написаний на мові серверного програмування PHP. Для написання клієнтської частини було вибрано популярний фронтенд фреймворк Angular, реалізований на динамічній мові Javascript. Обидві частини реалізації було

виконано з використанням принципів побудови архітектури MVC (модель - вигляд - контроллер), що дозволяло структуровано та зрозуміло подати структуру системи та взаємодію її основних компонентів. Структура бази даних подана на рис. 1. Для роботи з базою даних було реалізовано наступні моделі:

- Client – модель, клієнтів;

- Lead – модель замовлень;
- Task – модель завдань;
- User – модель користувачів;
- Role – модель ролей;
- Permission – модель прав;
- Vote - модель голосування;
- Sprint - модель спринтів.

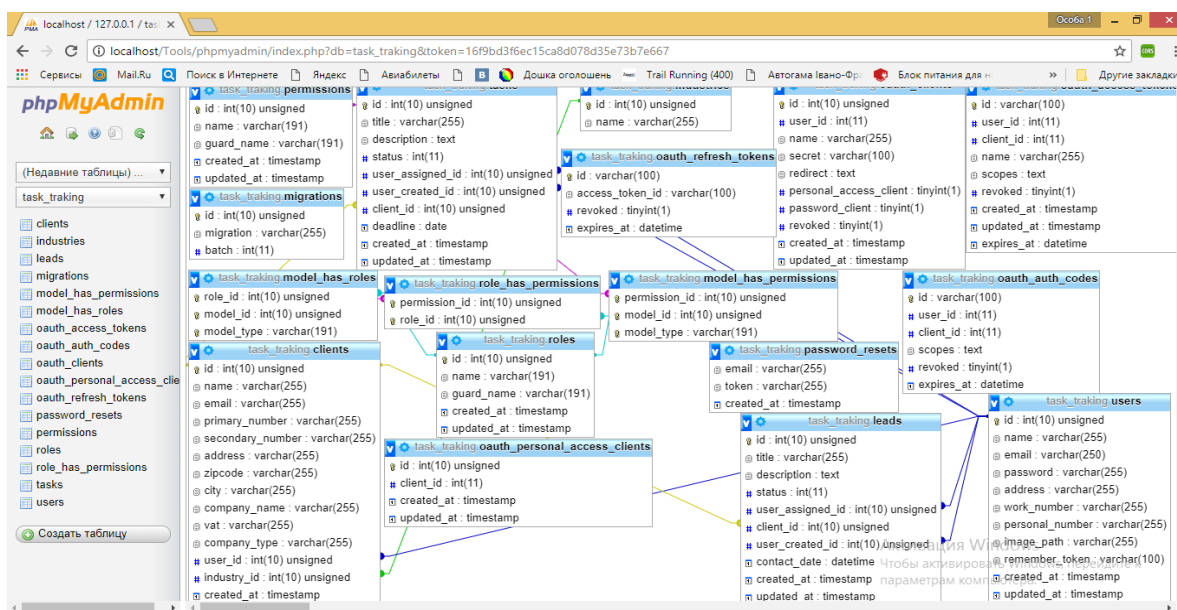


Рисунок 1 — Загальна схема бази даних системи

Для керування моделями було розроблено ряд класів-контролерів, а саме:

- ClientsController - використовується для роботи з даними клієнтів. Він працює з моделлю Client та використовується для реалізації вибірки, створення, модернізації та видалення клієнтів з бази даних;

- LeadsController - використовується для роботи з замовленнями. Він призначений для роботи з замовленнями. Працює з моделями: Lead (використовується для роботи з замовленнями), Client (для роботи з клієнтами), User (для роботи з користувачами, Sprint (для роботи зі спринтами);

- RolesController використовується для роботи з рольовою системою. Він працює з моделями: Role (працює з ролями користувачів в системі) та Permission (призначена для роботи з правами користувачів конкретних ролей щодо дій у системі);

- SprintsController використовується для роботи з блоками задач (спринтами). Він працює з моделями: Lead, Vote, User, Sprint;

- TasksController використовується для роботи з завданнями. Працює з моделями: Task, User, Sprint, Client, Vote;

- UsersController використовується для роботи з користувачами. Працює з моделями User і Role.

Приклад UML-діаграми класів для одного з контролерів подано на рис. 2.

Для створення проекту на Laravel framework, було міграції для необхідних таблиць в базі даних, відповідно до розробленої структури майбутньої системи. Механізм міграцій дозволяє за допомогою засобів фреймворка написати структуру таблиці, яку потім можна, за допомогою консолі, занести в базу даних.

Після цього були реалізовані CRUD (create, read, update, delete) функції для всіх основних таблиць з даними, а саме для таблиць: tasks, clients, sprints, leads, users.

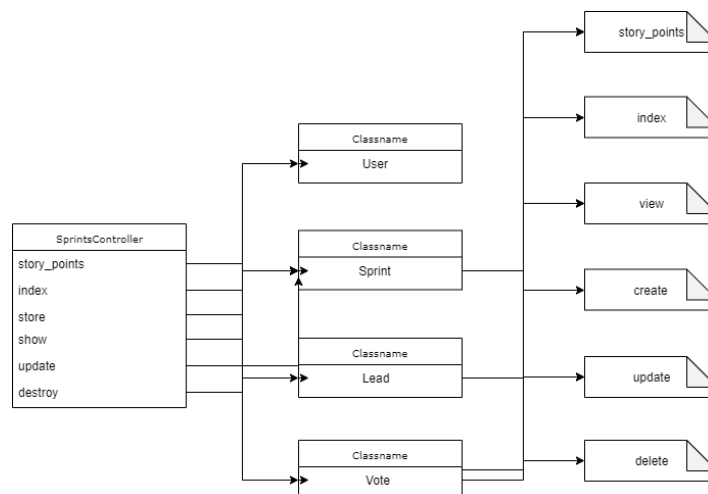


Рисунок 2 — Реалізація контролера системи на прикладі контролера спринтів

Крім того, була реалізована система авторизації та реєстрації користувача. Авторизація відбувалася стандартними механізмами Laravel, а можливість додавати нового користувача було відкрито тільки для адміністратора системи. Весь функціонал авторизації та реєстрації було реалізовано методами та засобами Laravel-модуля Laravel-passport authentication. Реалізацію було здійснено в методах класу UsersController.

Серверне програмне забезпечення реалізує взаємодію із клієнтськими інтерфейсами з допомогою API (програмного інтерфейсу додатку) та виводить відповіді в форматі JSON. Це дозволяє використовувати дане серверне програмне забезпечення для опрацювання даних та забезпечення роботи користувачів через широкий спектр інтерфейсів - як веб-орієнтованих, так і нативних додатків для операційних систем мобільних та настільних пристроїв. Це дозволяє спростити доступ користувачів до інформації, що знаходиться в базі даних [6].

Користувацький інтерфейс, що початково розглядався для вирішення даної задачі, реалізований засобами Angular. UML-діаграма подій подана на рис. 3. Приклад реалізації сторінок клієнтського інтерфейсу поданий на рис. 4.

Клієнтський інтерфейс забезпечує такі функції як:

- додавання, редагування та видалення клієнтів;
- додавання, редагування та видалення користувачів;
- авторизація користувачів;
- додавання, редагування та видалення ролей;
- оцінювання складності завдань учасниками команди;
- додавання, редагування та видалення завдань;
- додавання, редагування та видалення спринтів.

Додатково, як основний інструмент оперативного аналізу успішності виконання спринтів, була реалізована діаграма згоряння з відображенням кількості завдань, що залишилися для виконання у спринтів, лінії “ідеального” виконання спринта та прогнозованого часу виконання, базуючись на даних поточного дня.

Приклад діаграми згоряння подано на рис. 5.

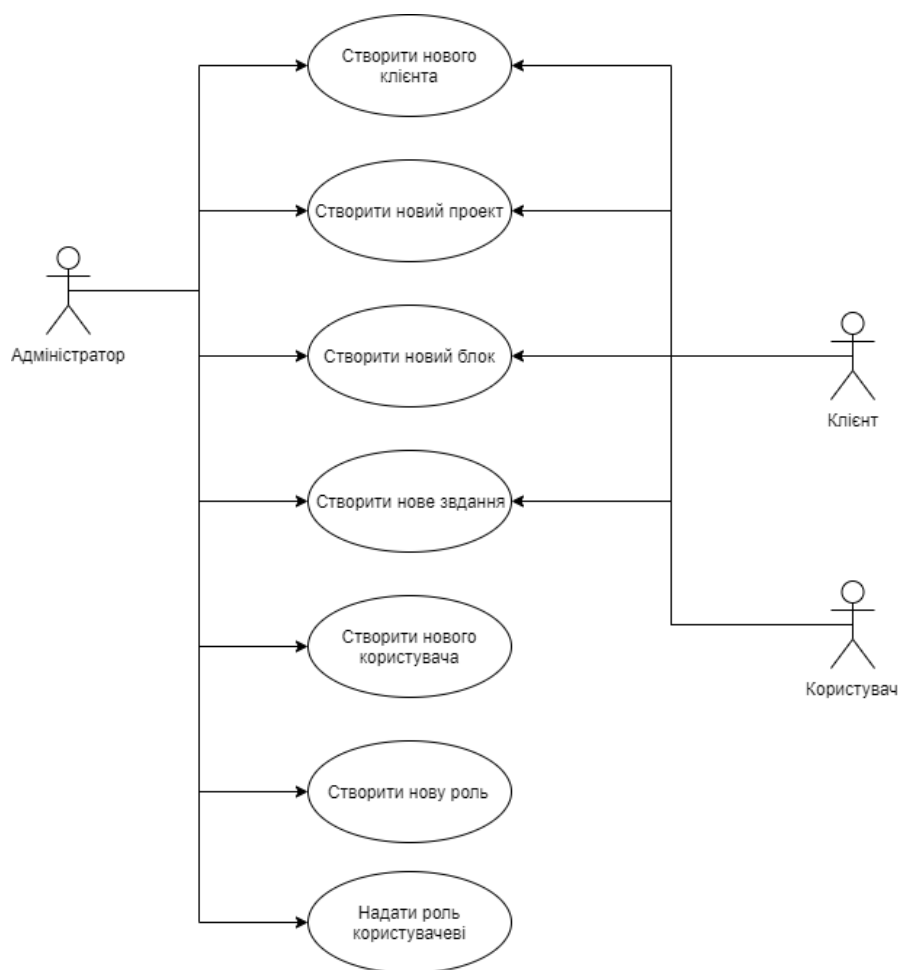


Рисунок 3 — UML-діаграма подій в розробленій системі

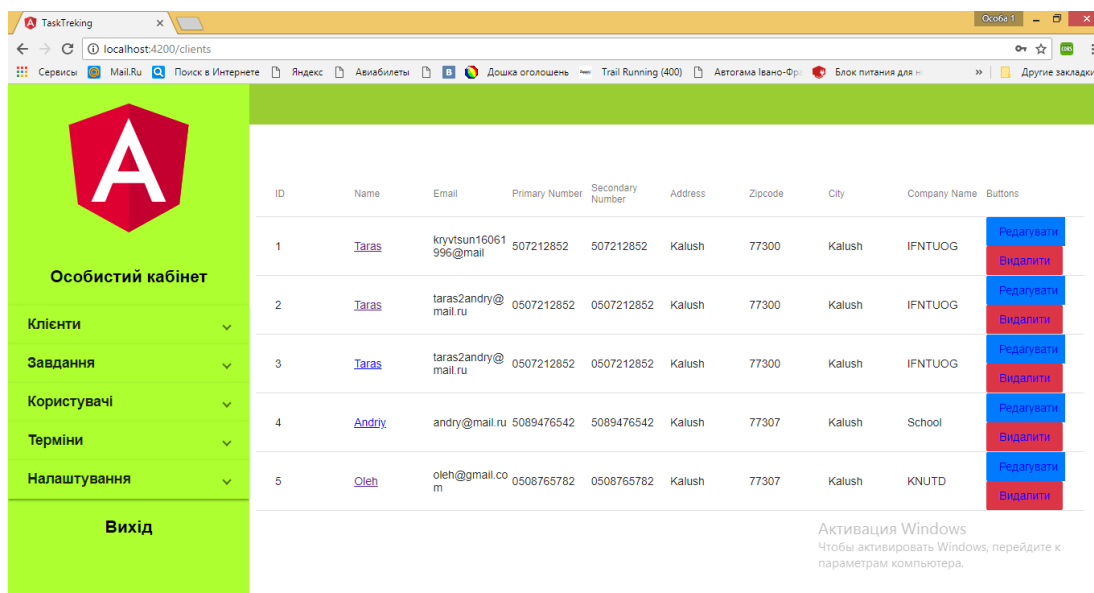


Рисунок 4 — Приклад роботи інтерфейсу користувача

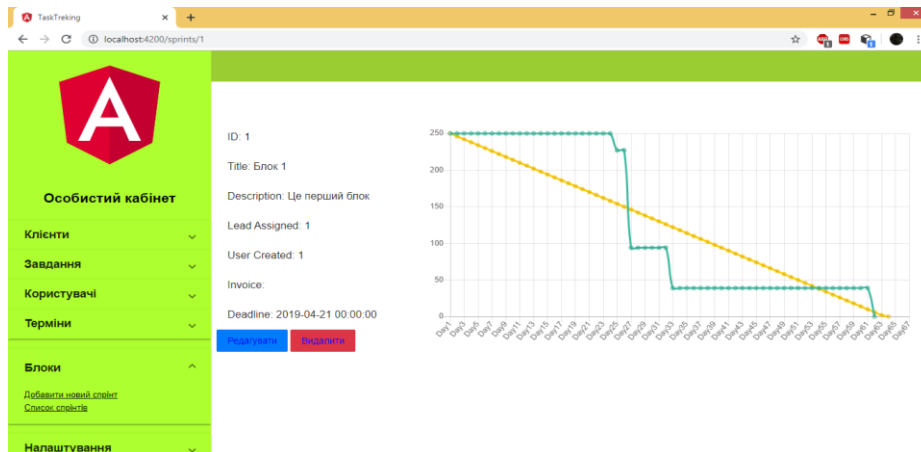


Рисунок 5 — Реалізація діаграми згоряння у веб-інтерфейсі користувача

Розроблений інструмент аналізу дозволяє керівнику проекту слідкувати за ходом виконання завдань спринта, та при потребі, коригувати графік, згідно прогнозованої дати завершення всіх завдань спринта.

Розроблена система виконана з доволі мінімалістичним набором інструментів, порівняно програмним забезпеченням Jira або Redmine, однак забезпечує всі необхідні інструменти для підтримки ведення проектів за інкрементними моделями. Архітектура MVC може бути легко масштабована та доповнена новими модулями та розширеннями, а програмний інтерфейс серверного забезпечення дозволяє окрім веб-інтерфейсу, в перспективі підключити інші клієнтські інтерфейси, як-то чат-боти чи мобільні додатки. Розроблене програмне забезпечення буде корисним для виконання невеликих проектів, що тривають декілька спринтів, та в яких частка роботи адміністративного персоналу є невеликою порівняно із технічними працівниками та інженерами з розробки програмного забезпечення.

Література

1. Rao BV, Seetharamaiah P. Organizational strategies and social interaction influence in software development effort estimation. J Comput Eng 2014;16(2) p. 29-40.
2. James Cadle. Project management for Information systems. - Pearson, 2008. - 465 p.
3. Agile: The World's Most Popular Innovation Engine [Electronic Resource]. Access mode: <https://www.forbes.com/sites/stevedenning/2015/07/23/the-worlds-most-popular-innovation-engine/?sh=3b8864ea7c76>

07/23/the-worlds-most-popular-innovation-engine/?sh=3b8864ea7c76

4. Показники ефективності KPI для працівників галузі IT технологій. [Електронний ресурс] Режим доступу: <https://habr.com/ru/post/22536/>

5. Які KPI можна вимірювати, якщо бюджет проекту і розклад не дуже важливі? [Електронний ресурс] Режим доступу: <https://habr.com/ru/post/135652/>

6. Chris Lee. Working with Web APIs. - Launch School, 2006. - 563 p.

References

1. Rao BV, Seetharamaiah P. Organizational strategies and social interaction influence in software development effort estimation. J Comput Eng 2014;16(2):29-40.
2. James Cadle. Project management for Information systems. - Pearson, 2008. - 465 p.
3. Agile: The World's Most Popular Innovation Engine [Electronic Resource]. Access mode: <https://www.forbes.com/sites/stevedenning/2015/07/23/the-worlds-most-popular-innovation-engine/?sh=3b8864ea7c76>
4. Pokazniki effektivnosti KPI dlya pratsivnikov galuzi IT tehnologiy. [Elektronniy resurs] Rezhim dostupu: <https://habr.com/ru/post/22536/>
5. 2. Yaki KRI mozha vimiryuvati, yakscho byudzheth proektu i rozklad ne duzhe vazhlyvi? [Elektronniy resurs] Rezhim dostupu: <https://habr.com/ru/post/135652/>
6. Chris Lee. Working with Web APIs. - Launch School. 2006. - 563 p.