SmartBase

Software Requirements Specification

Version 1.0

20/06/2025

Bulhakova Oleksandra

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 20/06/25 | Version 1 | Bulhakova Oleksandra | First Revision |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Bulhakova Oleksandra | Software Eng. | 20/06/25 |
|  |  |  |  |
|  |  |  |  |

**ЗМІСТ**

[Вступ 4](#_Toc201491580)

[Огляд продукту 4](#_Toc201491581)

[Мета 4](#_Toc201491582)

[Межі 4](#_Toc201491583)

[1 Загальний опис 5](#_Toc201491584)

[1.1 Перспектива продукту 5](#_Toc201491585)

[1.2 Функціональність продукту 5](#_Toc201491586)

[1.3 Характеристики користувачів 6](#_Toc201491587)

[1.4 Загальні обмеження 6](#_Toc201491588)

[1.5 Припущення та залежності 6](#_Toc201491589)

[2 Конкретні вимоги 7](#_Toc201491590)

[2.1 Функціональні вимоги 7](#_Toc201491591)

[2.2 Нефункціональні вимоги 8](#_Toc201491592)

[2.3 Вимоги до інтерфейсів 8](#_Toc201491593)

[2.4 Вимоги до бази даних 8](#_Toc201491594)

[3 Моделі аналізу 10](#_Toc201491595)

[3.1 Діаграма варіантів використання 10](#_Toc201491596)

[3.2 Загальна діаграма класів 11](#_Toc201491597)

[4 Управління змінами 13](#_Toc201491598)

ВСТУП

Огляд продукту

Цей документ містить специфікацію вимог до програмного забезпечення SmartBase – вебзастосунку для автоматизації процесів рекрутингу в IT-компаніях.

Мета

Встановити єдине уявлення між замовником і розробником щодо функціональності системи.

Межі

Програмне забезпечення є автономним веб-застосунком без інтеграції з фінансовими або сторонніми CRM.

1 ЗАГАЛЬНИЙ ОПИС

1.1 Перспектива продукту

Система SmartBase – це автономний веб застосунок для автоматизації рекрутингових процесів у межах ІТ-компаній. Система не є частиною більшого ПЗ, але може розгортатися як окремий модуль у корпоративній інфраструктурі.

Система складається з фронтенд-частини (React), бекенд-сервера (Spring Boot) і бази даних (PostgreSQL). Для зберігання файлів використовується Azure Blob Storage. SmartBase підтримує багаторольову модель користувачів: керівник відділу рекрутменту, рекрутери, наймаючі менеджери.

1.2 Функціональність продукту

Основні можливості системи:

* управління користувачами та ролями (реєстрація, авторизація, підтвердження поштою);
* додавання та редагування кандидатів і перегляд їхньої інформації;
* коментування кандидатів усіма учасниками рекрутингового процесу;
* управління вакансіями та їх атрибутами;
* прив’язка кандидатів до вакансій та перегляд історії активностей;
* призначення кроків рекрутингового процесу (інтерв'ю, перевірка англійської тощо);
* формування звіту для керівництва.

1.3 Характеристики користувачів

Керівник відділу рекрутменту планує та координує процеси, створює облікові записи, формує звіт. Рекрутери здійснюють пошук кандидатів, додають інформацію до системи, призначають інтерв’ю, супроводжують кандидата та фіксують коментарі. Наймаючі менеджери переглядають кандидатів, залишають фідбек, працюють із пулом кандидатів.

1.4 Загальні обмеження

Система повинна працювати у сучасних браузерах: Chrome, Firefox, Edge. Використовується архітектура REST для взаємодії між клієнтом і сервером. Аутентифікація користувачів виконується через JWT.

1.5 Припущення та залежності

Стабільне інтернет-з’єднання. Передбачається, що користувачі мають базові цифрові навички. Дані зберігаються в централізованій PostgreSQL-базі даних. Для коректної роботи з файлами необхідна активна інтеграція з Azure Blob Storage.

2 КОНКРЕТНІ ВИМОГИ

2.1 Функціональні вимоги

US-1: Як керівник відділу рекрутменту, я хочу мати змогу створювати користувачів з відповідними ролями і висилати їм на імейл посилання для завершення реєстрації.

US-2: Як користувач, я хочу авторизуватись у системі, щоб отримати доступ до функціоналу згідно моєї ролі.

US-3: Як рекрутер, я хочу створювати нового кандидата, щоб додати його до бази.

US-4: Як рекрутер, я хочу оновити профіль кандидата, щоб зберігати актуальну інформацію.

US-5: Як рекрутер, я хочу залишити коментар до кандидата, щоб фіксувати важливі нотатки з процесу спілкування.

US-6: Як наймаючий менеджер, я хочу переглядати профілі кандидатів, щоб оцінити відповідність ролі.

US-7: Як користувач, я хочу переглядати список вакансій.

US-8: Як рекрутер, я хочу створювати вакансію, вказуючи основні параметри (позиція, вимоги, технології).

US-9: Як користувач, я хочу фільтрувати кандидатів за технологією.

US-10: Як керівник відділу рекрутменту, я хочу формувати щотижневий звіт про діяльність команди, щоб контролювати процеси.

US-11: Як рекрутер, я хочу призначити інтерв’ю для кандидата та мати змогу вказати дату і його статус.

US-12: Як наймаючий менеджер, я хочу мати змогу написати коментар щодо кандидата за результатами перегляду резюме або інтерв'ю.

2.2 Нефункціональні вимоги

Завантаження списку кандидатів не повинно перевищувати 3 секунд; фільтрація має виконуватись не довше ніж за 1 секунду.

Система повинна працювати стабільно впродовж робочого дня без збоїв, з інформативною обробкою помилок.

Контроль доступу до функцій відповідно до ролей, використання JWT для авторизації.

Додавання нових типів користувачів чи полів не повинно вимагати суттєвих змін архітектури; код має бути підтримуваним і зрозумілим для нових учасників команди.

2.3 Вимоги до інтерфейсів

Інтерфейс користувача – веб інтерфейс з формами, таблицями, фільтрами. Мова інтерфейсу – українська. Апаратний інтерфейс відсутній. Обмеження пам’яті не встановлюється, система масштабована.

2.4 Вимоги до бази даних

У якості системи управління базами даних має використовуватись PostgreSQL.

Структура бази даних повинна відповідати принципам третьої нормальної форми (3НФ) для уникнення дублювання даних.

Усі сутності мають бути пов’язані за допомогою зовнішніх ключів відповідно до логіки предметної області.

Для зберігання файлів повинна використовуватись зовнішня служба (Azure Blob Storage), а в базі зберігаються лише посилання.

Створення та оновлення структури бази даних має виконуватись через систему міграцій Liquibase.

На рисунку 1 наведено ER-діаграму бази даних.

Зображення, що містить текст, схема, План, знімок екрана

Вміст на основі ШІ може бути неправильним.

Рисунок 1 – ER-діаграма бази даних

3 МОДЕЛІ АНАЛІЗУ

3.1 Діаграма варіантів використання

На діаграмі, наведеній на рисунку 2, зображено три основні ролі: керівник відділу, рекрутер і наймаючий менеджер. Кожна з ролей має доступ до певного набору функцій.

**Зображення, що містить текст, схема, ряд, знімок екрана

Вміст на основі ШІ може бути неправильним.**

Рисунок 2 – Діаграма варіантів використання

Керівник відділу може реєструвати працівників у системі, переглядати звіти та управляти усіма сутностями. Рекрутер керує кандидатами, вакансіями, клієнтами, може додавати коментарі та бачити звітність. Наймаючий менеджер може переглядати профілі кандидатів і залишати зворотній зв'язок.

3.2 Загальна діаграма класів

Діаграма класів, наведена на рисунку 3, описує основні сутності системи SmartBase та їх взаємозв’язки.

Зображення, що містить текст, схема, План, знімок екрана

Вміст на основі ШІ може бути неправильним.

Рисунок 3 – Загальна діаграма класів

Ключові класи:

* User – зберігає облікові дані, пов'язаний з компанією та може залишати коментарі;
* Company – представляє профіль компанії, зареєстрованої у системі, до якої відносяться користувачі, кандидати та клієнти;
* Candidate – містить інформацію про кандидата, пов'язаний із вакансіями, клієнтами та компанією;
* Vacancy – створюється користувачем для клієнта, містить вимоги і зв’язана з кандидатами;
* Appointment – репрезентує події, пов’язані з кандидатом (інтерв’ю, перевірка англійської тощо);
* Client – представляє клієнта, для якого ведеться пошук кандидата;
* Comment, Contact, Address, Technology, Attachment – допоміжні сутності для зберігання відповідної інформації.

Між об'єктами встановлено зв'язки типу один-до-одного, один-до-багатьох або багато-до-багатьох відповідно до логіки предметної області.

4 УПРАВЛІННЯ ЗМІНАМИ

Усі зміни до вимог повинні бути погоджені з керівником проєкту або замовником та внесені у специфікацію. Кожна зміна має проходити наступні етапи:

* ініціація: опис зміни та причина, автор та час;
* обговорення: оцінка впливу на функціонал, трудомісткість, можливі ризики;
* затвердження: отримання погодження від відповідальних осіб;
* реалізація: внесення змін у специфікацію, реалізація в коді;
* документування: оновлення відповідних артефактів (діаграми, документація, тести).

Для прозорості змін має використовуватись система контролю версій Git.