

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ПОЛТАВСЬКИЙ ПОЛІТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Циклова комісія дисциплін програмної інженерії

ЗВІТ

з технологічної практики

«Розробка і супроводження програмного продукту»

на тему Компаньйон водіїв підприємства у відрядженні

Виконав: здобувач освіти 4 курсу,
групи 45

спеціальності 121

Інженерія програмного забезпечення

Любченка О.В.

(прізвище та ініціали)

Керівник

Кривцова О. П.

(підпис)

(прізвище та ініціали)

Полтава – 2024

ЗМІСТ

| | |
|---|----|
| ВСТУП | 3 |
| 1. ПОСТАНОВКА ЗАВДАННЯ | 5 |
| 1.1. Основні вимоги до продукту | 5 |
| 1.2. Вимоги до інтерфейсу..... | 6 |
| 2. ПЛАНУВАННЯ СИСТЕМИ | 7 |
| 2.1. Архітектура системи..... | 7 |
| 2.2. Тестування | 8 |
| 2.3. Інструкція з використання системи..... | 8 |
| ВИСНОВКИ..... | 10 |
| СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ | 11 |
| ДОДАТОК А. ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПРОГРАМНИХ ПРОДУКТІВ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ | 12 |
| ДОДАТОК Б. UML ДІАГРАМА ПРЕЦЕДЕНТІВ | 14 |
| ДОДАТОК В. ПРОТОТИП ІНТЕРФЕЙСУ | 15 |
| ДОДАТОК Г. UML ДІАГРАМА КЛАСІВ (АБО ІНШІ)..... | 16 |
| ДОДАТОК Д. СТРУКТУРА САЙТУ (Цей додаток буде лише тоді коли ви розробляєте сайт) | 17 |
| ДОДАТОК Е. ER ДІАГРАМА (Цей додаток буде лише тоді коли ваш продукт використовує базу даних)..... | 18 |
| ДОДАТОК Д. ВИХІДНІ КОДИ..... | 19 |
| ДОДАТОК Ж. UML ДІАГРАМА РОЗГОРТАННЯ..... | 20 |
| ДОДАТОК И. РЕЗУЛЬТАТИ ТЕСТУВАННЯ | 21 |
| ДОДАТОК Е. ЗНІМКИ ЕКРАНУ | 22 |

ВСТУП

Місцем моєї технологічної практики була компанія Акціонерне товариство «Укртранснафта».

Компанія є єдиним оператором системи магістральних нафтопроводів України з 2001 року, що надає послуги з транспортування нафти трубопровідним транспортом на нафтопереробні підприємства (НПЗ) України та транзитом до країн Східної та Центральної Європи, а також нафтопродуктів з використанням української ділянки магістрального нафтопродуктопроводу «Самара-Західний напрям» як в прямому, так і реверсному напрямку.

Основним напрямком її діяльності є оперування нафтотранспортної системи України, яка включає 18 нафтопроводів загальною протяжністю 4569 км, 51 нафтоперекачувальну станцію (НПС), 11 резервуарних парків загальною місткістю 1010 тисяч кубометрів. Роботу НПС забезпечують 176 насосних агрегатів загальною потужністю електродвигунів 356,9 тисяч кВт.

В ході практики я ознайомився з роботою відділу аналітики комп'ютерних систем цеху електрозв'язку «Глинсько-Розбишівська» ЛВДС «Глинсько-Розбишівська», де переді мною було поставлено завдання розробити мобільний застосунок для допомоги водіям підприємства у пошуку інфраструктури для них та автомобілів під час відрядження.

Для виконання поставленого завдання мені стали в нагоді знання, вміння й навички з дисциплін «Автоматизація інформаційних систем» та «Бази даних».

Під час практики я працював із низкою сучасних інструментів та технологій, які суттєво розширили мої знання та навички в галузі розробки програмного забезпечення.

Серед використаних інструментів була потужна і зручна IDE Microsoft Visual Studio Code. Цей інструмент забезпечує ефективне середовище

розробки завдяки своїй гнучкості, підтримці багатьох мов програмування та широкому вибору плагінів, які оптимізують робочий процес.

Також я освоїв Google Firebase Command Line Interface (CLI), що дозволило мені працювати з хмарними сервісами Firebase безпосередньо через командний рядок. Це значно пришвидшує процес розробки та тестування додатків.

Для управління ресурсами в хмарному середовищі я використовував Google Cloud Console, яка надає інтуїтивно зрозумілий інтерфейс для адміністрування проєктів та відстеження використання ресурсів. Google Maps Platform допомогла мені інтегрувати інтерактивні карти та геолокаційні сервіси у додаток, що значно підвищило його функціональність.

Під час практики я також працював із Figma – сучасним інструментом для дизайну інтерфейсів. Завдяки Figma я створив прототипи додатків, які дозволили наочно уявити кінцевий продукт ще до початку кодування.

Особливу увагу я приділив вивченню мови програмування Dart, яка є основою для розробки на Flutter. Ця мова вирізняється простотою та високою продуктивністю, що дозволяє створювати швидкі та ефективні додатки.

Фреймворк Flutter став ключовим елементом моєї практики. Завдяки Flutter я зміг розробляти кросплатформенні додатки з багатим функціоналом і сучасним дизайном. Цей фреймворк забезпечує високу швидкість розробки та дозволяє створювати додатки для iOS та Android з єдиною базою коду.

Я також вивчив роботу з базою даних Firebase та її компонентами, такими як Firebase Firestore, Firebase Authentication та Firebase Realtime Database. Firestore забезпечує зручне зберігання даних із можливістю масштабування, Firebase Authentication – простий спосіб реалізації автентифікації користувачів, а Realtime Database дозволяє обмінюватися даними в реальному часі, що є важливим для створення інтерактивних додатків.

1. ПОСТАНОВКА ЗАВДАННЯ

Опишіть коротко яке завдання Ви отримали...

Які вимоги до програмного продукту поставив замовник...

Терміни розробки....

Функціональність програми....

Середовище і мова розробки...

Вимоги стосовно супроводу та реалізації нових функцій...

Порівняйте вашу майбутню програму з вже існуючими на ринку продуктами аналогічного призначення (порівняльну характеристику у вигляді таблиці - в додаток!)

1.1. Основні вимоги до продукту

Наведіть перелік детальних вимог замовника – окремо функціональних (які можливості має надавати Ваш продукт) і нефункціональних (безпека, зручність використання, масштабованість, відмово стійкість тощо).

Візуалізуйте функціональні вимоги з допомогою діаграми прецедентів (в додатки!)

ФУНКЦІОНАЛЬНІ ВИМОГИ:

—

—

НЕФУНКЦІОНАЛЬНІ ВИМОГИ:

—

—

Архітектуру якого типу має Ваш продукт?

З яких підсистем він складається?

Як його побудовано?

Які технології/ мови програмування/ середовища розробки/ бібліотеки/компоненти/фреймворки було використано під час створення продукту?

Чи мають місце якісь залежності? Встановлення яких додаткових компонентів (і яких версій) вимагає Ваша програма?

Зобразіть архітектуру вашого продукту з допомогою діаграм класів, компонентів та розгортання (в додатки!)

Яку функціональність надає програма?

ПЕРЕЛІК РЕАЛІЗОВАНИХ ЗА ЧАС ПРАКТИКИ ФУНКЦІОНАЛЬНИХ ВИМОГ

—

— ...

ПЕРЕЛІК НЕРЕАЛІЗОВАНОЇ ФУНКЦІОНАЛЬНОСТІ

—

— ...

вказіть причини з яких її не було реалізовано.

1.2. Вимоги до інтерфейсу

Наведіть перелік вимог користувача стосовно зовнішнього вигляду програми.

Візуалізуйте їх за допомогою прототипу інтерфейсу, створеного з застосуванням відповідних сервісів та ПЗ, наприклад, draw.io чи Pencil GUI

Вкажіть який саме GUI-фреймворк було використано в ході створення Вашого продукту, якщо було використано візуальний конструктор, то який саме.

Перелічіть і обґрунтуйте прийняті Вами рішення в плані поліпшення зручності користування продуктом і загального покращення користувацького досвіду.

Опишіть зовнішній вигляд програми, її вікна, елементи керування, надайте посилання на додаток з скріншотами. Знімки екрану наводимо в додатках, а в тексті розділу пишемо посилання (Додаток В)!

2. ПЛАНУВАННЯ СИСТЕМИ

Згадайте «Основи програмної інженерії» та практику з ТРПЗ:

Опишіть життєвий цикл Вашого проекту.

Яку модель ЖЦ Ви обрали?

У відповідності до якої методології розробки ПЗ працювали?

Які етапи ЖЦ Ви пройшли? Що і у якій послідовності робили?

Які програмні засоби використовували для планування/керування проектом, контролю версій (Gantt Project, Trello, Azure DevOps, GitHub...)

2.1. Архітектура системи

Детально опишіть архітектуру Вашого продукту.

Які патерни (шаблони проектування) чи стандартні (референсні) архітектури Ви використали?

З яких підсистем і компонентів складається Ваша система?

Надайте посилання на додатки з пакетом структурних UML-діаграм (діаграми класів, діяльностей і послідовностей – обов'язкові). Якщо Ваша система – не об'єктно-орієнтована (малоймовірно, однак, може бути) – наведіть детальні блок-схеми реалізованих алгоритмів (теж в додатках!).

Діаграми наводимо в додатку, а в тексті розділу пишемо посилання (Додаток Г-???)

В цьому розділі ви також оприсуєте особливості коду. Та надаєте посилання на додаток – вихідні коди програми. В додатку ви наводите код одного компоненту чи класу, а решту кодів розміщуєте у публічному репозиторії на GitHub, і вказуєте посилання для доступу до нього.

Веб-застосунки мають бути розгорнуті та доступні для перегляду (не треба шукати підозрілі хостинги - використовуйте Azure, AWS, GCP тощо, клієнтські веб-застосунки хостимо в GitHub Pages). Ігри мають бути розміщені в одному з магазинів – наприклад, в itch.io, для десктопних

застосунків інсталяційний пакет має бути доступний для завантаження з репозиторію, мобільні застосунки мають бути розміщені в одному з безкоштовних магазинів (SlideMe, F-Droid, Aptoide тощо).

УВАГА! Ваш програмний продукт має обов’язково містити ваші копірайти (інформацію про те, що ви є автором програми – у вікні «Про програму», рядку статусу тощо).

2.2. Тестування

Опишіть використані Вами види тестування (юніт-тести – обов’язкові, інші види – інтеграційне, навантажувальне, сумісності, доступності, стрес-тестування тощо – можуть різнитись).

Опишіть використані вами фреймворки для юніт-тестування (наприклад, Junit, NUnit тощо), засоби для автоматизації тестів інших видів (TestComplete, AutoIt, Selenium тощо)

Опишіть процес тестування програми та знайдені помилки, наведіть посилання на відповідні скріншоти (в додатку)

Які зміни Ви внесли до проекту у результаті тестування чи, можливо, його пілотного впровадження?

Які зауваження стосовно продукту Ви отримали від замовника?

Знімки екрану наводимо в відповідному додатку, а в тексті розділу пишемо посилання (Додаток-???)

2.3. Інструкція з використання системи

Опишіть послідовність дій, необхідних для встановлення і запуску Вашого продукту.

Розкажіть, як працювати з програмою – надайте детальний і зрозумілий покроковий опис дій користувача з посиланнями на скріншоти (в додатку).

Найкращий і найпростіший спосіб оформлення цього розділу – згадати про прецеденти, зображені Вами на діаграмі прецедентів і розписати сценарії, які відповідають кожному прецеденту для кожного виду користувачів системи.

Увага! В цьому випадку знімки екрану не виносимо в окремий додаток, а додаємо рисунками під текстом (Рисунок ??.)

ВИСНОВКИ

Коротко опишіть, яке завдання було перед Вами поставлено, що саме було зроблено, який продукт Ви створили, наведіть приклади його застосування, зазначте чого Ви навчилися, які навички вдосконалили, розкажіть чи плануєте Ви продовжувати роботу над проектом. Обсяг висновків – близько 1 сторінки.

Висновки мають перекликатись зі вступом, тож тут теж слід сказати (але коротше) де ви проходили практику, яке завдання виконували, що в результаті було зроблено, а що ні, чи використовує замовник Ваш продукт. Якщо у Вас є відгук чи акт впровадження про це теж слід сказати у висновках (а копію самого документа – в додатки!). Згадайте про навчальні дисципліни, знання з яких Вам допомогли, які навички ви отримали, вкажіть чи створений Вами продукт є частиною дипломного проекту, чи маєте Ви домовленості з замовником про роботу над проектом/його супровід у приватному порядку тощо...

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

Список літератури має бути оформленим у відповідності до стандарту ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні вимоги та правила складання» і створений за допомогою відповідного сервісу – <https://4ref.onaft.edu.ua/>

1.

ДОДАТОК А. ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПРОГРАМНИХ ПРОДУКТІВ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ

Зведіть в таблицю порівняння Вашого продукту з існуючими на ринку продуктами аналогічного призначення (підберіть 2 аналоги). Стовпці таблиці можуть відповідати програмним продуктам, рядки – критеріям, за якими Ви порівнюєте продукти (наприклад, адаптивний інтерфейс, наявність української мови, можливість імпорту/експорту даних, вартість ліцензії тощо). Ви можете ставити у комірках таблиці плюси чи мінуси щоб показати наявність чи відсутність реалізації того чи іншого критерія) або оцінювати програми за 5-бальною шкалою.

| | Програмний продукт 1 | Програмний продукт 2 | Створений програмний продукт |
|--|-------------------------|-------------------------|------------------------------------|
| Адаптивність | | | |
| Багатокористувацький режим | | | |
| Інтуїтивно-зрозумілий інтерфейс користувача | | | |
| Наявність української мови | | | |
| Можливість імпорту/експорту даних | | | |
| Крос-платформність (наявність версій для Linux/macOS) | | | |
| Автентифікація з використанням соціальних облікових записів (Google/Facebook/Twitter тощо) | | | |
| Наявність довідкових матеріалів та документації | | | |
| Можливості з налагодження та розширення функціональності (Rest API/SDK) | | | |

| | | | |
|-------------------|--|--|--|
| Вартість ліцензії | | | |
|-------------------|--|--|--|

ДОДАТОК Б. UML ДІАГРАМА ПРЕЦЕДЕНТІВ

Діаграма має бути побудована з використанням відповідного інструменту (draw.io, Visual Paradigm Online, Lucid Charts, StarUML тощо), експортована звідти й вставлена сюди. Не треба робити скріншоти й додавати зображення які містять сітку, направляючі лінії, маркери тощо!

ДОДАТОК В. ПРОТОТИП ІНТЕРФЕЙСУ

Прототип інтерфейсу має бути побудований за допомогою відповідного ПЗ (наприклад, draw.io чи Pencil GUI), експортований звідти й вставлений сюди.

Не треба робити скріншоти й додавати зображення які містять сітку, направляючі лінії, маркери тощо!

Згадайте, що прототип – це не готовий вигляд інтерфейсу й не скріншоти, а просто макет, на якому показано які елементи де і як розташовані! До речі, у вас скоріш за все буде кілька зображень прототипу – для різних вікон програми.

ДОДАТОК Г. UML ДІАГРАМА КЛАСІВ (АБО ІНШІ)

Діаграми мають бути побудовані з використанням відповідного інструменту (draw.io, Visual Paradigm Online, Lucid Charts, StarUML тощо), експортовані звідти й вставлені сюди. Не треба робити скріншоти й додавати зображення які містять сітку, направляючі лінії, маркери тощо!

Нагадуємо, що діаграми класів, діяльностей і послідовностей – обов’язкові. Якщо Ваша система – не об’єктно-орієнтована (малоймовірно, однак, може бути) – наведіть детальні блок-схеми реалізованих алгоритмів (так, їх скоріш за все буде кілька!)

ДОДАТОК Д. СТРУКТУРА САЙТУ (Цей додаток буде лише тоді коли ви розробляєте сайт)

Додаємо цей додаток, якщо ви працюєте над створенням сайту. Намалювати її можна, наприклад, з допомогою draw.io – там навіть є для цього набір позначень. Така схема демонструє основні сторінки сайту та їх ієрархію.

УВАГА! Не існує таких понять як «діаграма сайту», «алгоритм роботи сайту» тощо!

ДОДАТОК Е. ER ДІАГРАМА (Цей додаток буде лише тоді коли ваш продукт використовує базу даних)

Це діаграма, яка ілюструє схему БД. Намалювати її можна з допомогою draw.io. Додаємо цей додаток, якщо ПП використовує базу даних (що буде у переважній більшості проєктів).

ДОДАТОК **Д**. ВИХІДНІ КОДИ

Додаємо цей додаток, ВІН Є ОБОВ'ЯЗКОВИМ!

Використовуйте підсвітку синтаксису!

Один з додатків – вихідні коди програми. Тут ви наводите код одного компоненту чи класу, а решту кодів розміщуєте у публічному репозиторії на GitHub, і вказуєте посилання для доступу до нього.

Веб-застосунки мають бути розгорнуті та доступні для перегляду (не треба шукати підозрілі хостинги - використовуйте Azure, AWS, GCP тощо, клієнтські веб-застосунки хостимо в GitHub Pages). Ігри мають бути розміщені в одному з магазинів – наприклад, в itch.io, для десктопних застосунків інсталяційний пакет має бути доступний для завантаження з репозиторію, мобільні застосунки мають бути розміщені в одному з безкоштовних магазинів (SlideMe, F-Droid, Aptoide тощо).

Якщо обсяг вихідних кодів значний – наводите посилання на публічний репозиторій з кодами, а тут – лише вихідні коди певного компоненту (класу, підсистеми) Вашого продукту.

Коди наводяться 12-м шрифтом і у форматованому вигляді (з підсвіткою синтаксису) – скористайтесь tohtml.com або highlight.hohli

ДОДАТОК **Ж**. UML ДІАГРАМА РОЗГОРТАННЯ

Додаємо цей додаток, якщо варто показати як саме (на які обчислювальні вузли – сервери, клієнтські машини тощо) відбувається розгортання вашого ПП (бажано зробити цей додаток)

ДОДАТОК **II**. РЕЗУЛЬТАТИ ТЕСТУВАННЯ

Цей додаток є підсумком розділу 2.2.!

Використовуйте відповідні сервіси та фреймворки для проведення тестів
(дивіться вище – опис розділу, присвяченому тестуванню)

ДОДАТОК **Е.** ЗНІМКИ ЕКРАНУ

Наведіть достатню кількість знімків екрану аби якнайкраще показати Вашу роботу!

Буде чудово, якщо це будуть знімки тих вікон, прототипи яких Ви навели в Додатку В – це дозволить побачити як змінився зовнішній вигляд продукту при переході від етапу проектування до розробки.

Враховуємо, що звіт з практики друкуємо в відтінках сірого (НЕ кольорові)