Одеський національний політехнічний університет

Інститут комп'ютерних систем

Кафедра інформаційних систем

# КУРСОВА РОБОТА

з дисципліни “Об’єктно-орієнтоване програмування”

Тема “ Розробка десктопного застосунку для організації справ при командній роботі над проектами ”

Студента (ки) курсу групи

(прізвище та ініціали)

Керівник Рудніченко М.Д.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (підпис) (прізвище та ініціали

м. Одеса – 2018 рік

Одеський національний політехнічний університет

Інститут комп'ютерних систем Кафедра інформаційних систем

**ЗАВДАННЯ**

НА КУРСОВУ РОБОТУ

(прізвище, ім’я, по батькові) (група)

1. Тема роботи “ Розробка десктопного застосунку для організації справ при

командній роботі над проектами ”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Термін здачі студентом закінченої роботи 05.05.2018

1. Початкові дані до проекту (роботи): список користувачів та команд, в яких вони працюють; список всіх проектів команди, до якої входить користувач, додаткові дані про кожен з них: назва, адміністратор, дедлайн, опис; список справ користувача.

1. Зміст розрахунково-пояснювальної записки (перелік питань, які належить розробити): вступ, аналіз специфіки предметної області, проектування, опис

процесу роботи, висновки, перелік посилань, код програми

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Завдання видано \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Завдання прийнято до виконання \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_



ЗМІСТ

ВСТУП ....................................................................................................................... 4

1. АНАЛІЗ СПЕЦИФІКИ ПРЕДМЕТНОЇ ОБЛАСТІ ........................................... 5
   1. Особливості розробки десктопного застосунку .................................................... 5
   2. Аналіз аналогів на ринку ................................................................................ 5
   3. Використані засоби розробки ..................................................................................... 8
2. ПРОЕКТУВАННЯ .............................................................................................11
   1. Написання user stories та формування backlog`у продукту.............................. 11
   2. Розробка прототипів графічного інтерфейсу ....................................................... 19
   3. Виявлення класів і зв’язків за допомогою CRC-карток.................................... 19
   4. База даних .......................................................................................................20
   5. UML – діаграми ............................................................................................................ 24
      1. Діаграма класів ............................................................................................................. 25
      2. Діаграма послідовностей ........................................................................................... 25
3. ОПИС ПРОЦЕСУ РОБОТИ ..............................................................................29

ВИСНОВКИ .............................................................................................................42

ПЕРЕЛІК ПОСИЛАНЬ ............................................................................................44

ДОДАТОК А ...........................................................................................................45

ІТ КР 126 АД-171 ПЗ

Змн*.* Арк. № докум. Підпис Дата

ВиконавИванов И.И. Літ. Лист Листів

Перевірив *Рудніченко М.Д.* Розробка десктопного 3 45

Реценз.

застосунку для організації

Н.Контр. ПеревіривОНПУ, каф. ІТ, гр. АД-171

справ при командній роботі Утверд.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ВСТУП    Потоки інформації, що циркулюють у світі, який нас оточує, величезні. З часом вони мають тенденцію до збільшення. У будь-якій компанії, як великий, так і малій, виникає проблема організації управління даними, яка забезпечила б найбільш еффективну роботу. В сучасних умовах однією з найбільш розповсюджених задач, що виникає при користуванні комп'ютером, є використання інформаційних систем, заснованих на програмних застосунках, які стали вірними помічниками майже у будь-якій справі.  Особливо актуальною проблема організації задач та справ постає в компаніях, де спеціалісти сумісно виконують різні проекти, розподіляючи завдання. Тому створення застосунку, який би допомагав командам в таких умовах працювати найбільш ефективно є надзвичайно доцільним.  Метою даної роботи є створення десктопного застосунку, розрахованого на управління заздалегідь певною структурою інформації і рішення цілком певного і обмеженого кола завдань для організації, робота якої заснована на колективній праці.  Програмні застосунки можна реалізовувати на різних мовах програмування. Однією із таких є об’єктно-орієнтована мова програмування  Java.  Мета роботи полягає в розширенні, закріпленні та систематизації знань з дисципліни, що вивчається, шляхом проектування та розробки програмного забезпечення, а саме десктопного застосунку для організації командної роботи.  Застосунок розробляється для персональних комп’ютерів під операційною системою Windows. Середовище розробки IntelliJ IDEA, мова програмування  Java. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 4 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. АНАЛІЗ СПЕЦИФІКИ ПРЕДМЕТНОЇ ОБЛАСТІ      * 1. Особливості розробки десктопного застосунку     Десктопні програми - це повнофункціональні програми, які працюють незалежно від інших застосунків і вимагають наявність оператора. Для їх роботи необхідні достатні апаратні ресурси комп'ютера, сам додаток і набір функцій для роботи з застосунком [1].  Такі застосунки розміщуються на комп'ютері користувача. Вони не вимагають для роботи підключення до інтернету, взаємодіють з користувачем за допомогою стандартного інтерфейсу, мають більш високу швидкодію, залежать від операційної системи і вимагають установку на кожен комп'ютер користувача, який бажає працювати з ним. Це текстові редактори, медіа-плеєри, програми розрахунку, обчислення, вивчення – в загальному все програми, які встановлені у нас на комп'ютерах, є десктопними застосунками. Так як ми маємо доступ до системних файлів програми, даний тип додатків більш вразливий, і повністю залежить від дій користувача.     * 1. Аналіз аналогів на ринку     Для аналізу розробляємого програмного забезпечення перевіримо його унікальність, розглянувши декілька подібних рішень. Одними з найбільш популярних програм для ефективної організації роботи в командах є:  Trello;  Asana.  Розглянемо ії детальніше, проаналізуємо основні переваги та недоліки.  Trello – це програмний продукт Fog Creek Software, який популярний серед невеликих команд (рис. 1.1). Якщо в світі систем управління проектами можна визначити мінімалізм, це однозначно Trello. Freemium-модель дозволяє не турбуватися щодо оплати непотрібних в маленьких проектах додаткових | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 5 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| можливостей, маючи можливість завжди при необхідності «розширитися» до роботи побільше. Розробники надають доступ до API, доступні мобільні застосунки на iOS, Android.      Рисунок 1.1 – Головне вікно сервісу Trello    В основі сервісу закладена популярна парадігма управління канбан. Вона полягає у тому, що за правильно заданих вихідних даних і правильному розподілі ресурсів робота буде виконана вчасно. Аудиторія сайту у вересні 2014 року склала 5 млн користувачів – в честь цього компанія надавала місяць безкоштовного Gold-доступу.  З переваг Trello варто виділити:   * легкість і доступність; * інтеграція в хмарні сховища для управління файлами; * зручні дошки та чати; * наявність повідомлень.   Головний мінус є комплексним – невеликий функціонал. Це робить «Дошку» зручною тільки для скромних проектів, а до більш великих завдань доведеться доглядати щось серйозніше. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 6 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Asana – простий та надзвичайно зручний сервіс для управляння задачами та проектами (рис. 1.2). «Командна робота без e-mail» - свідчить слоган на офіційному сайті проекту. І це дійсно так: за рахунок зручних списків завдань сервіс дозволяє організувати роботу команди, не виходячи за межі системи.  Розроблений співзасновником Facebook сервіс став доступний в 2011 році.  Сьогодні їм користуються майже 100 000 команд розробників.      Рисунок 1.2 – Головне вікно сервісу Asana    Система дає можливість користуватися API, доступні програми для iOS і  Android. Основною одиницею тут є робочий стіл – віддалений родич дошки з Trello. При цьому є можливість сортувати завдання за пріоритетністю, присутня зручна система коментування та зв'язку з колегами. Схема підписки – Freemium,  а за додатковий функціонал команді з 5 осіб доведеться заплатити $ 21.  Відрізняють Asana:   * відмінна функціональність, кілька «рівнів» постановки завдання. Як наслідок – хороша деталізація; * зручний інтерфейс робить продукт схожим на повноцінне десктопних програм | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 7 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * відсутня необхідність комунікації поза сервісу; * система підходить як для планування серйозних завдань у великій команді розробників, так і для складання списку покупок батьком-одинаком. З недоліків користувачі відзначають відсутність локалізованої версії продукту. Крім того, якщо порівнювати сервіс з Trello, другий дає більше свободи в налаштуванні дощок-списків за рахунок меншої регламентованості.     1.3 Використані засоби розробки    В якості середовища розробки було обране IntelliJ IDEA. Це інтегроване  середовище розробки програмного забезпечення для багатьох мов програмування, зокрема Java [2], JavaScript, Python, розроблена компанією JetBrains. Дизайн середовища орієнтований на продуктивність роботи програмістів, дозволяючи сконцентруватися на функціональних завданнях, в той час як IntelliJ IDEA бере на себе виконання рутинних операцій.  Мовою програмування десктопного застосунку стала Java – об'єктноорієнтована мова програмування, випущена компанією Sun Microsystems в 1995 році як основний компонент платформи Java. Зараз розробкою мови займається компанія Oracle, яка придбала Sun Microsystems в 2009 році. Синтаксис мови багато в чому схожий на C і C ++. В офіційній реалізації, Java додатки компілюються в байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи [6].  Для реалізації бази даних була обрана СУБД PostgreSQL – [об'єктнореляційна](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) [система керування базами даних.](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) Вона є альтернативою як комерційним СУБД [(Oracle Database,](https://uk.wikipedia.org/wiki/Oracle_Database) [Microsoft SQL Server,](https://uk.wikipedia.org/wiki/Microsoft_SQL_Server) [IBM DB2](https://uk.wikipedia.org/wiki/DB2) та інші), так і СУБД з відкритим кодом ([MySQL,](https://uk.wikipedia.org/wiki/MySQL) [Firebird,](https://uk.wikipedia.org/wiki/Firebird) [SQLite)](https://uk.wikipedia.org/wiki/SQLite).  Для управління та складання проекту будо обрано «Apache Maven» — це засіб автоматизації роботи з програмними проектами, який спочатку використовувався для Java проектів. Maven забезпечує підтримку побудови не просто перебираючи файли з цього репозиторію, але й завантажуючи назад | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 8 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| артефакти у кінці побудови. Локальний кеш звантажених артефактів діє як первісний засіб синхронізації виходу проектів на локальній системі.  Для опису програмного проекту який потрібно побудувати (build), Maven використовує конструкцію відому як Project Object Model (POM), залежності від зовнішніх модулів, компонентів та порядку побудови. Виконання певних, чітко визначених задач — таких, як компіляція коду та пакетування відбувається шляхом досягнення заздалегідь визначених цілей (targets). Ключовою особливістю Maven є його мережева готовність (network-ready).  Maven базується на плаґін-архітектурі, що дозволяє зробити використання будь-якої програми контрольованим через стандартний вхід. Теоретично, це могло б дозволити будь-кому писати плаґіни для інтерфейсу з інструментами для побудови (компілятори, тестери тощо) для будь-якої мови. В дійсності, підтримка і використання для мов відмінних від Java були мінімальною. Тепер існують плаґіни для .NET та C/C++.  Для роботи з графічним інструментом було використано JavaFX. Це  платформа та набір інструментів для створення насичених інтернет-застосунків з можливістю підвантаження медіа та змісту. Вперше продемонстровано Sun Microsystems на Міжнародній конференції Java-розробників JavaOne у травні 2007. JavaFX включає в себе набір утиліт, за допомогою яких веб-розробники та дизайнери можуть швидко створювати та надавати розвинуті інтернетзастосунки для десктопів, мобільних пристроїв, телебачення та інших платформ. JavaFX Scene Builder – це візуальний інструмент макета, який дозволяє користувачам швидко створювати користувальницькі інтерфейси JavaFX без кодування. Користувачі можуть перетягувати компоненти інтерфейсу  користувача до робочої області, змінювати їх властивості, застосовувати таблиці стилів, а кодова FXML для макета, який вони створюють, автоматично генерується у фоновому режимі. Результатом є файл FXML, який потім можна об'єднати з проектом Java, зв'язавши інтерфейс користувача з логікою програми.  Усі засоби розробки вказані у таблиці 1.1. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 9 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблиця 1.1 Використані програмні засоби розробки   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Назва засобу/ технології** | |  | **Версія** | |  | **Посилання для завантаження** | **Короткий опис**  **призначення** | | Java | |  | 9 | |  | https://java.com/ru/download/ | Мова написання коду | | JDBC Drive  for  PostgreSQL | | r | 9.19011.jdbc4 | |  | https://mvnrepository.com/artif act/postgresql/postgresql | Драйвер для підключення до бази даних | |  | PostgreSQL |  |  | 9.5.13 |  | https://www.postgresql.org/ | СУБД | | IntelliJ IDEA | |  | 2017. | | 2 | https://www.jetbrains.com/idea | Середовище розробки | | JavaFX | |  | 2.2.7 | |  | http://www.oracle.com/technet  work/java/javase/overview/java fx-overview-2158620.html | Розробка графічного інтерфейсу | | SceneBilder | |  | 2.0 | |  | http://www.oracle.com/technet  work/java/javase/downloads/sb  2download-2177776.html | Візуальний інструмент для дизайну | | Maven | |  | 3.5.2 | |  | http://maven.apache.org/ | Управління та складання проекту | | Git | |  | 2.17 | |  | https://git-scm.com/downloads | Контроль версій | | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 10 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. ПРОЕКТУВАННЯ     2.1 Написання user stories та формування backlog`у продукту    Бачення системи  Планується реалізувати систему, яка б дозволила користувачам вміло організовувати свої завдання та відстежувати іх виконання в рамках сумісного виконання спільних проектів.  Користувацькі ролі:  Гість – користувач, який не має облікового запису, тому не може користуватися системою. Може зареєструватися.  Користувач - може авторизуватися, член певної команди. Може переглядати завдання проектів, частиною команди яких він є, виконувати свої завдання та відправляти адміністратору на перевірку.  Адміністратор – користувач, який здійснює контроль над проектами, в яких він назначений головним. Може додавати, коригувати та видаляти проекти та завдання.  Користувацькі історії:   1. Як гість я могу зареєструватися в системі для подальшої роботи.   Тест 1: натиснути кнопку «*Реєстрація*». Відкриється вікно реєстрації, ввести необхідні дані та отримати доступ до системи.   1. Як гість я могу увійти до системи під своїм обліковим записом. Для цього необхідно ввести логін та пароль.   Тест 1: ввести дані, натиснути кнопку «*Увійти*». Відкриється головне вікно програми.   1. Як адміністратор я можу додати проект для подальшої роботи з ним.   Тест 1: натиснути кнопку «*Додати проект*». Відкриється вікно для додавання інформації. Ввести необхідні дані та зберегти їх.   1. Як користувач я можу зробити відмітку про завершення завдання та відправили його на перевірку адміністратору. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 11 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Тест 1: натиснути кнопку «*Завершено*» при перегляді додаткової інформації про задачу.   1. Як адміністратор я можу додати завдання до певного проекту.   Тест 1: необхідно натиснути на кнопку "*Додати завдання*". Відкриється вікно для додавання інформації. Ввести необхідні дані та зберегти їх.   1. Як користувач я можу переглянути усі подробиці певного завдання.   Тест 1: із списка завдань вибрати необхідне та нажали на нього.  Відкриється вікно із усією додатковою інформацією.   1. Як користувач я можу переглянути усі свої задачі.   Тест 1: натиснути на кнопку "*Мої завдання*". Відкриється список завдань певного користувача.   1. Як користувач я можу подивитися усі завдання проектів, в яких я приймаю участь.   Тест 1: у вкладці *«Проекти»* натиснути на потрібний. Відкриється вікно із усіма подробицями проекту. Натиснути на кнопку *«Усі завдання»*. У новому вікні з’явиться таблиця із завданнями проекту.   1. Як адміністратор я можу внести зміни до завдань.   Тест1: у вкладці *«Завдання»* натиснути на потрібне завдання.  Відкриється вікно із усіма подробицями завдань. Внести необхідні зміни.  Натиснути на кнопку *«Зберегти»*.   1. Як адміністратор я можу видалити завдання.   Тест1: у вкладці *«Завдання»* натиснути на потрібне завдання. Відкриється вікно із усіма подробицями завдань. Натиснути на кнопку *«Видалити»*.   1. Як адміністратор я можу внести зміни до проектів.   Тест1: у вкладці *«Проекти»* натиснути на потрібний проект. Відкриється вікно із усіма подробицями завдань. Внести необхідні зміни. Натиснути на кнопку *«Зберегти»*.   1. Як адміністратор я можу видалити проект. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 12 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Тест1: у вкладці *«Проекти»* натиснути на потрібний. Відкриється вікно із усіма подробицями завдань. Натиснути на кнопку *«Видалити»*.   1. Як адміністратор я можу підтвердити виконання користувачем завдання.   Тест1: натиснути на кнопку "*Виконані завдання*". Відкриється вікно з усіми завданнями, відпраленими на перевірку, вибрати потрібне та підтвердити його виконання.   1. Як користувач я можу переглядати свій профіль.   Тест1: натиснути на кнопку "*Профіль*". Відкриються вікно з інформацією про користувача.   1. Як користувач я можу змінити дані в своєму профелі.   Тест1: натиснути на кнопку "*Профіль*". Відкриються вікно з інформацією про користувача. Внести необхідні зміни, натиснути на кнопку *«Зберегти»*.   1. Як користувач я можу відсортувати завдання за наближенням дедлайну.   Тест1: у вкладці *«Завдання»* натиснути на кнопку "*Відсортувати за датою*».   1. Як користувач я можу відсортувати завдання за приорітетом.   Тест1: у вкладці *«Завдання»* натиснути на кнопку "*Відсортувати за приорітетом*».   1. Як користувач я можу відсортувати проекта за наближенням дедлайну.   Тест1: у вкладці *«Проекти»* натиснути на кнопку "*Відсортувати за датою*».  Backlog продукту - єдиний упорядкований список вимог до продукту, тобто перелік усіх можливостей, які зацікавлені люди хочуть побачити в продукті.  Backlog продукту продемонстровано в таблиці 2.1. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 13 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблиця 2.1 Backlog продукту   |  |  | | --- | --- | | Пріоритет | Користувацька історія | | 10 | Як гість я могу зареєструватися в системі для подальшої роботи. | | 10 | Як гість я могу увійти до системи під своїм обліковим записом. | | 9 | Як адміністратор я можу додати проект для подальшої роботи з ним. | | 9 | Як користувач я можу зробити відмітку про завершення завдання та відправили його на перевірку адміністратору. | | 9 | Як адміністратор я можу додати завдання до певного проекту. | | 8 | Як користувач я можу переглянути усі подробиці певного завдання. | | 8 | Як користувач я можу переглянути усі свої задачі. | | 8 | Як користувач я можу подивитися усі завдання проектів, в яких я приймаю участь. | | 7 | Як адміністратор я можу внести зміни до завдань. | | 7 | Як адміністратор я можу видалити завдання. | | 7 | Як адміністратор я можу внести зміни до проектів. | | 7 | Як адміністратор я можу видалити проект. | | 7 | Як адміністратор я можу підтвердити виконання користувачем завдання. | | 6 | Як користувач я можу відсортувати проекта за наближенням дедлайну. | | 6 | Як користувач я можу відсортувати завдання за наближенням дедлайну. | | 6 | Як користувач я можу відсортувати завдання за приорітетом. | | 5 | Як користувач я можу переглядати свій профіль. | | 5 | Як користувач я можу змінити дані в своєму профелі. |     2.2 Розробка прототипів графічного інтерфейсу    Storyboard являє собою звичайну діаграму, де прямокутниками позначені вікна, а стрілками - можливість переходу з цього вікна в інше (поруч зі стрілкою, як правило, вказують тригер переходу - кнопку, вкладку і т.д.). | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 14 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| На рис. 2.1 представлений Storyboard проекту в нотації UML (Activity diagram).    Рисунок 2.1 – Storyboard проекту    Макети основних вікон.  На рис. 2.2 зображено вікно при вході до системи.      Рисунок 2.2 – Макет вікна авторизації | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 15 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

Змн

*.*

Арк

*.*

№ докум.

Підпис

Дата

Арк.

16

І

Т

КР 12

6

АД

1

71

ПЗ

На рис. 2.3 зображено в

ікно при створенні нового облікового запису.

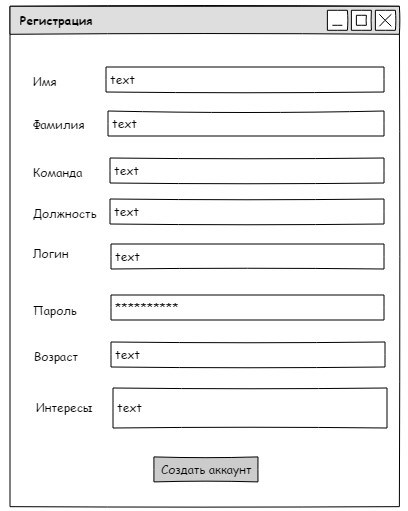


Рисунок 2.3

–

Макет вікна регістрації

На рис. 2.4 зображено головне вікно програми.

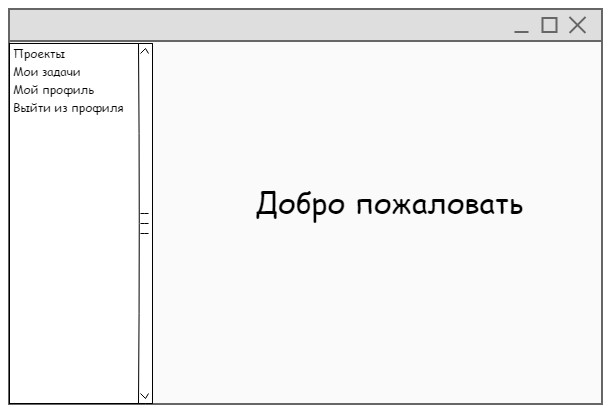


Рисунок 2.4

–

Макет головного вікна

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| На рис. 2.5 зображено вікно із усіма проектами команди, в якій працює користувач.      Рисунок 2.5 – Макет вікна проектів    На рис. 2.6 зображено вікно із усіма завданнями, які має виконати користувач.      Рисунок 2.6 – Макет вікна задач | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 17 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| На рис. 2.7 зображено вікно, в якому відображені усі відомості про певний проект з можливістю видалення та коригування.      Рисунок 2.7 – Макет вікна додаткової інформації про проекти    На рис. 2.8 зображено вікно, в якому відображені усі відомості про певне завдання з можливістю видалення та коригування.      Рисунок 2.8 – Макет вікна задач | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 18 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2.3 Виявлення класів і зв’язків за допомогою CRC-карток    CRC-картка (англ. Class-responsibility-collaboration card) - це інструмент  мозкового штурму що використовується при проектуванні об'єктноорієнтованого ПЗ. Спершу вони були запропоновані Вордом Каннінгемом та Кентом Беком як навчальний інструмент, але також були популярними і серед спеціалістів і рекомендуються прихильниками екстремального програмування. Мартін Фаулер описав CRC-картки як життєздатну альтернативу діаграмам послідовностей в UML щоб проектувати динаміку взаємодії між об'єктами.  Основні CRC-картки зображені на рисунках 2.9 – 2.10.      Рисунок 2.9 – CRC-картки (1) | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 19 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 2.10 – CRC-картки (2)    2.4 База даних    [База даних](http://lib.mdpu.org.ua/e-book/vstup/glos.htm#База даних) - упорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 20 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| потреб користувачів. В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти[4].  Для опису бази даних для десктопного застосунку, відповідно до вхідних та вихідних даних, а також необхідного функціоналу були створені такі сутності, зв’язок між якими продемонстровано на рис. 2.11:   * користувач; * задача; * команда; * проект; * виконані завдання; * профіль;     Рисунок 2.11 – Діаграма бази даних    Сутність «Користувач» містить інформацію про юзерів (рис. 2.12), які мають доступ до системи:   * код користувача; * логін; * пароль; | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 21 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * роль.     Рисунок 2.12 – Сутність «Користувач» в MS Access    Сутність «Профіль» (рис. 2.13) містить додаткову інформацію про користувачів:   * код профіля; * код користувача; * ім’я; * фамілія; * вік; * місце роботи; * код команди; * інтереси.       Рисунок 2.13 – Сутність «Профіль» в MS Access    Сутність «Проект» (рис. 2.14) містить усю необхідну інформацію про проекти відповідної команди:   * код проекта; * назва; * дедлайн; | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 22 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * опис;  закінчений; * код команди.     Рисунок 2.14 – Сутність «Проект» в MS Access    Сутність «Завдання» містить відомості про усі завдання користувачів та відповідних проектів (рис. 2.15):   * код завдання; * код проекту; * назва; * дедлайн; * опис; * закінчено; * приорітет; * відповідальний.       Рисунок 2.15 – Сутність «Завдання» в MS Access | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 23 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Сутність «Виконані завдання» містить відомості про завдання, які були відправлнені на перевірку до адміністратора (рис. 2.16):   * код; * код завдання.       Рисунок 2.16 – Сутність «Виконані завдання» в MS Access    Сутність «Команда» містить відомості про команди, в яких працюють користувачі (рис. 2.17):   * код команди;  адміністратор; * назва.       Рисунок 2.17 – Сутність «Команда» в MS Access    2.5 UML – діаграми    UML (англ. Unified Modeling Language) — уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 24 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація [7].    2.5.1 Діаграма класів    Діаграма класів — статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.  Діаграма класів проекту зображена на рисунку 2.18.    2.5.2 Діаграма послідовностей    Діаграма послідовності (англ. sequence diagram) — різновид діаграми в UML. Діаграма послідовності відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень.  Для проекту було розроблено 5 основних діаграм послідовностей.  Діаграма послідовностей для додавання завдання зображена на рис. 2.20, для додавання проекту – на рис. 2.21.  Діаграма послідовносей для входу до системи зареєстрованого  користувача зображена на рис. 2.22. Діаграми для перегляду завдань та проектів  – на рисунках 2.23 та 2.19. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 25 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 2.18 – Діаграма класів      Рисунок 2.19 – Діаграма послідовностей для перегляду проектів | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 26 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 2.20 – Діаграма послідовностей для додавання задачі      Рисунок 2.21 – Діаграма послідовностей для додавання проекту | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 27 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Рисунок 2.22 – Діаграма послідовностей для входу до системи        Рисунок 2.23 – Діаграма послідовностей для перегляду завдань | | | |  |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 28 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 ОПИС ПРОЦЕСУ РОБОТИ    Програмний застосунок слугує помічником для людей, які разом працюють над сумісними проектами. За допомогою нього можна швидко та якісно координувати роботу, розподіляти задачі та слідкувати за їх виконанням. Невід’ємною частиною застосунку є база даних, в якій зберігається вся необхідна для роботи інформація.  При запуску програми відкривається вікно входу до системи (рис. 3.1), де користивуч може ввести свій логін та пароль, якщо у нього вже є обліковий запис. Якщо поля не були заповнені (рис. 3.2) або при введенні одного з параметрів була допущена помилка (рис. 3.3), буде отримано повідомлення про помилку.      Рисунок 3.1 – Вікно авторизацій      Рисунок 3.2 – Повідомлення про помилку | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 29 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 3.3 – Повідомлення при неправильному введенні даних    Якщо користувач програми ще не має облікового запису, він може перейти до вікна реєстрації, натиснувши на кнопку «SignUp» (рис. 3.4). Якщо користувач не заповнює необхідні поля, які помічені зірочкою, то з’явиться повідомлення про помилку (рис. 3.5). Якщо користувач із таким логіном вже існує, буде отримано повідомлення про це (рис. 3.6). Якщо вказаної команди немає в базі, то з’явиться повідомлення із відповідним посиланням (рис. 3.7).      Рисунок 3.4 – Вікно реєстрації | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 30 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 3.5 – Повідомлення, якщо не усі дані введені      Рисунок 3.6 – Повідомлення, якщо користувач із таким логіном існує      Рисунок 3.7 – Повідомлення, якщо команди не існує    Після авторизація залежно від того, яка роль користувача в команді  (звичайний користувач чи адміністратор), відкриється головне вікно в звичайному режимі(рис. 3.8) чи режимі адміністратора (рис. 3.9).      Рисунок 3.8 – Головне вікно в звичайному режимі | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 31 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 3.9 – Головне вікно в режимі адміністратора    В лівій частині вікна можна побачити основне меню, яке простежується в усіх вікнах відповідного режима. Меню складається із таких пунктів:   * «Проекти» – у цьому вікні буде можливість переглянути усі проекти команди, в якій працює користувач; * «Мої задачі» – відкриється вікно із усіма завданнями користувача; * «Мій профіль» – перехід до вікна, де користувач зможе переглянути, змінити та зберегти усі відомості про себе; * «Вихід із системи» – повернення до вікна авторизації.   У режимі адміністратора модна побачити ще один пункт – «Виконані завдання» – відкриється вікно із завданнями, які користувачі команди відправили адміністратору на перевірку.  При натисканні на пункт «Проекти» відкриється відповідне вікно в звичайному режимі (рис. 3.10) чи в режимі адміністратора (рис. 3.11).  У цьому вікні з’являється можливість відсортувати проекти за наближенням до дедлайну (рис. 3.12). | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 32 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 3.10 – пункт «Проекти»      Рисунок 3.11 – «Проекти» в режимі адміністратора      Рисунок 3.12 – «Проекти» після сортування | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 33 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| При натисканні на відповідному проекті, відкриється вікно із відомостями про нього. В звичайному режимі є можливість лише передивитися їх та подивитися усі завдання, які входять до проекту (рис. 3.13, 3.15). У режимі адмінистратора (рис. 3.14) можливо також змінювати та видаляти проекти.  Видалити проект, доки не виконані усі завдання, які до нього входили неможливо (рис. 3.16).    Рисунок 3.13 – Подробиці певного проекту      Рисунок 3.14 – Подробиці певного проекту в режимі адміністратора | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 34 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 3.15 – Всі завдання певного проекту      Рисунок 3.16 – Повідомлення про помилку    У вікні «Усі завдання» певного проекту можна обрати будь-яке завдання, про яке хочеться дізнатися подробиці та натиснути на нього. З’явиться вікно із описанням завдання.  У режимі адміністратора є також можливість додати новий проект. Після натискання на відповідний пункт відкриється вікно (рис. 3.17), де користувач може ввести усі необхідні параметри та зберегти їх.  При натисканні на пункт «Мої завдання» відкриється відповідне вікно в звичайному режимі (рис. 3.18) чи в режимі адміністратора (рис. 3.19).  У цьому вікні з’являється можливість відсортувати завдання за наближенням до дедлайну (рис. 3.20) чи за приорітетом (рис. 3.21). | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 35 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 3.17 – Додавання проекту      Рисунок 3.18 – пункт «Мої завдання» | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 36 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 3.19 – «Мої завдання» в режимі адміністратора      Рисунок 3.20 – «Мої завдання» після сортування за приорітетом      Рисунок 3.21 – «Мої завдання» після сортування за дедлайном    При натисканні на відповідному завданні, відкриється вікно із відомостями про нього. В звичайному режимі є можливість лише передивитися їх (рис. 3.22). У режимі адмінистратора (рис. 3.23) можливо також змінювати та видаляти завдання. Видалити невиконане завдання неможливо (рис. 3.16). | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 37 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

Змн

*.*

Арк

*.*

№ докум.

Підпис

Дата

Арк.

38

І

Т

КР 12

6

АД

1

71

ПЗ

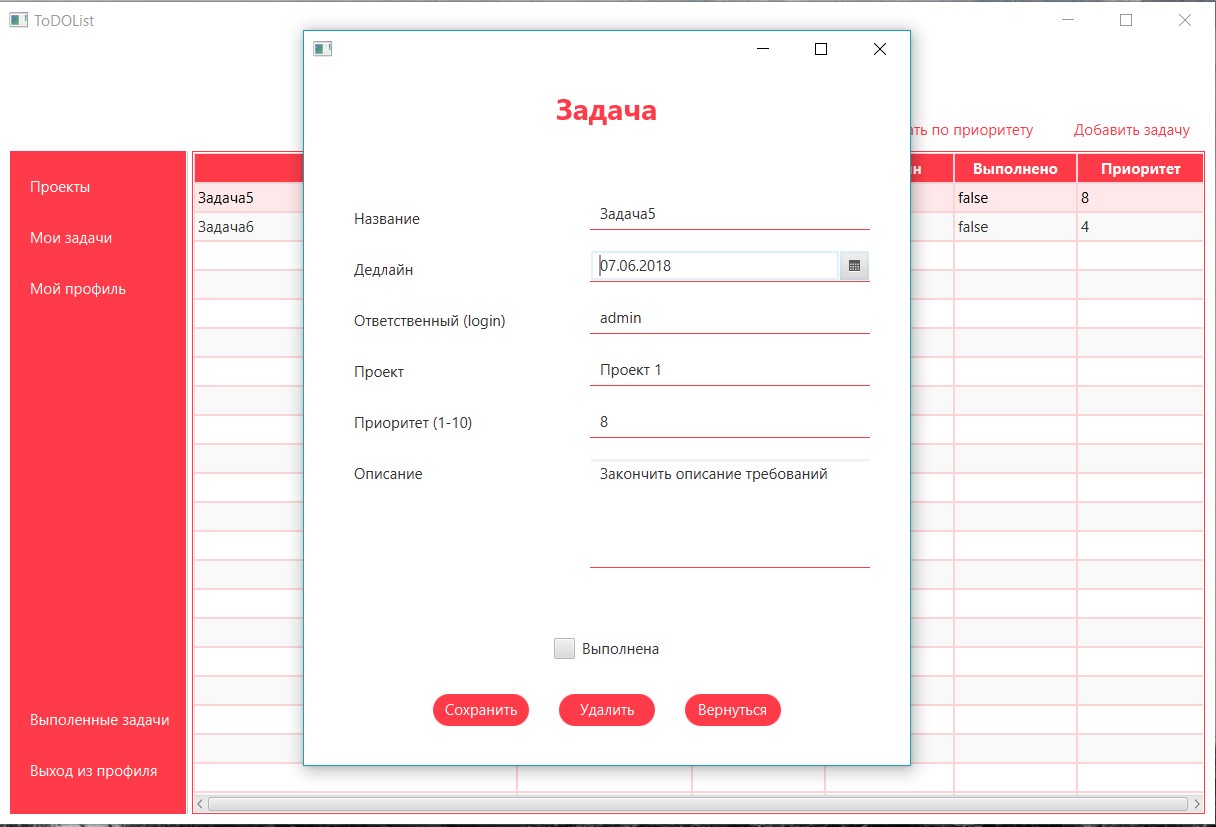


Рисунок 3.22

–

Подробиці певного завдання

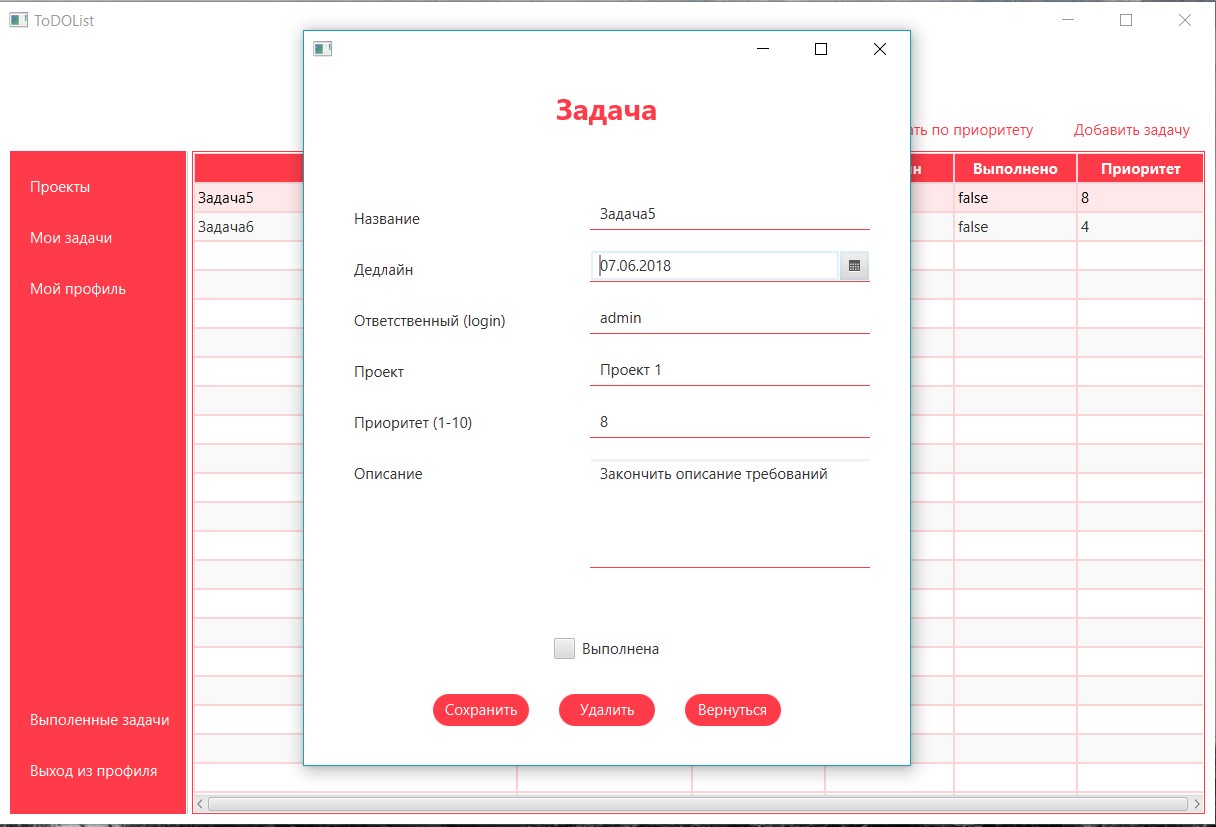


Рисунок 3.23

–

Подробиці певного завдання в режимі

адміністратора

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| У режимі адміністратора є також можливість додати нове завдання. Після натискання на відповідний пункт відкриється вікно (рис. 3.24), де користувач може ввести усі необхідні параметри та зберегти їх.  При введенні дедлайну, який перевищує дедлайн відповідного проекту буде показано повідомлення про помилку (рис. 3.25).      Рисунок 3.24 – Додавання завдання        Рисунок 3.25 – Повідомлення при неправильному дедлайні | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 39 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Користувач може помітити певне завдання виконаним (рис. 3.26), тоді воно буде відправлене адміністратору команди на перевірку. Якщо завдання вже було відправлено, то користувач отримає про це повідомлення (рис. 3.27).      Рисунок 3.26 – Відправлення завдання на перевірку      Рисунок 3.27 – Повідомлення про помилку    При натисканні на пункт «Мій профіль» відкриється відповідне вікно з усією інформацією користувача (рис. 3.27). В цьому вікні можна продивитися чи змінити необхідні дані. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 40 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 3.27 – Вікно пункта «Мій профіль»    При натисканні в режимі адміністратора на пункт «Виконані завдання» з’явиться вікно, де адміністратор команди зможе подивитися усі виконані завдання (рис. 3.28), перевірити їх та підтвердити їх виконання.      Рисунок 3.28 – Вікно пункта «Виконані завдання»    Код програмної реалізації наведено в додатку А. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 41 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ВИСНОВКИ    В наш час володіння та користування інформацією є дуже важливим, тому треба вміти надзвичайно швидко пристосовуватися до змін науково-технічного прогресу. Важливою є саме здатність самостійно працювати із новою необхідною інформацією. На сьогоднішній день програмні застосунки дозволяють ефективно зберігати, структурувати і систематизувати великі обсяги даних. І без них неможливо уявити роботу більшості фінансових, промислових, торгівельних та інших організацій. Використання клієнт-серверних технологій дозволяє зберегти значні кошти, а головне і час для отримання необхідної інформації, а також спрощують доступ і ведення, оскільки вони грунтуються на комплексній обробці даних і централізації їх зберігання.  Виконання даної курсової роботи дозволило мені:   * розширити, закріпити і систематизувати теоретичні та практичні знання в рамках дисципліни, що вивчається, стосовно специфіки розробки сучасного програмного забезпечення; * розвинути навички ведення самостійних теоретичних і практичних досліджень в напрямку проектування програмного забезпечення; * набути досвіду обробки, аналізу та систематизації результатів практичних досліджень; * сформувати навички роботи з науковою літературою та правильного оформлення дослідницьких робіт; * закріпити навички науково оформляти і викладати свої думки, висновки і результати дослідження.   В ході виконання першого розділу був проведений аналіз розробки застосунку для персонального комп’ютера. Була представлена структура застосунку. Проведено аналіз аналогів на ринку. Були вибрані та обґрунтовані програмні засоби розробки, які використовувалися, а також наведені приклади їх використання. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 42 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| В другому розділі представлено проектування бази даних. Були представлені логічна та фізична модель бази даних. Була обрана СУБД.  В ході виконання третього розділу була спроектована архітектура застосунку. Було описано бачення системи, користувацькі ролі, користувацькі історії, сформований backlog продукту. Розроблені прототипи графічного інтерфейсу, виявлені класи та зв’язки між ними за допомогою CRC-карток. Були написані діаграми класів та послідовностей.  В ході виконання четвертого розділу був написаний програмний код продукту. Описаний процес роботи програмного застосунку. Наведені скриншоти інтерфейсу.  Завдяки розробці програми мною було вдосконалено власний стиль програмування. Я отримала досвід роботи з мовою програмування Java, а також практичні навички розробки десктопного застосунку. Я переконана у тому, що отриманий досвід безперечно знадобиться мені у професії. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 43 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ПЕРЕЛІК ПОСИЛАНЬ     1. Басюк Т. М. Основи інформаційних технологій: навч. посібник / Т.М.   Басюк, Н.О. Думанський, О.В. Пасічник; за наук. ред. В.В. Пасічника.– Л. :  Новий Світ-2000, 2011.– 390 с.   1. Васильєв А.Н. Об’єктно-орієнтоване програмування. Java; Навчальний посібник – СПб.: Пітер, 2011. – 400с. 2. Вирт Н. Алгоритмы и структуры данных: Пер. с англ./ Н. Вирт. – СПб.:   Невский диалект, 2008. – 352 с.   1. Глава М.Г. Організація баз даних та знань: Конспект лекцій.   [Електронний ресурс] – Режим доступу: [http://library.opu.ua.](http://library.opu.ua/)   1. Гіберт Ш.Б. Java. Полное руководство, 8-е издание; Посібник   програміста. – СПб.: Київ, 2013. – 175с.   1. Копитко М.Ф., Іванків К.С. Основи програмування мовою Java: Тексти лекцій. – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002. – 83 с. 2. Фаулер Мартін. UML. Основи. Третє видання. Короткий посібник по стандартній мові об’єктного моделювання. — Символ Плюс, 2006. – 501 с. 3. Шилдт Г., Холмс Д. Искусство программирования на Java. - М.:   Издательский дом "Вильямс", 2005. – 336 стр. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 44 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ДОДАТОК А  Програмний код можна переглянути за посиланням: [https://github.com/.](https://github.com/angelinagramatik/toDoList).. | | | | | | |
|  |  |  |  |  | ІТ КР 126 АД171 ПЗ | Арк. |
|  |  |  |  |  | 45 |
| Змн*.* | Арк*.* | № докум. | Підпис | Дата |