**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КПІ»**

**Кафедра**

**автоматизованих систем обробки інформації та управління**

**Пояснювальна записка до розрахункової роботи**

з дисципліни

**“** **Технології Створення Програмних Продуктів”**

на тему

**"Система складання колендарного плану виготовлення товарів"**

|  |  |  |
| --- | --- | --- |
| Керівник : ас. Сперкач М. О.  Допущений до захисту  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2015 \_\_\_\_\_\_\_\_  підпис  Захистив з оцінкою  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_  оцінка підпис  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2015 |  | Виконав: Вереня О. І.  Залікова книжка № ІС-2102  Студент гр.. ІС-24, ФІОТ  3 курс |
| Керівник : ас. Сперкач М. О.  Допущений до захисту  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2015 \_\_\_\_\_\_\_\_  підпис  Захистив з оцінкою  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_  оцінка підпис  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2015 |  | Виконав: Кмець М.І.  Залікова книжка № ІС-2109  Студент гр.. ІС-24, ФІОТ  3 курс |
| Керівник : ас. Сперкач М. О.  Допущений до захисту  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2015 \_\_\_\_\_\_\_\_  підпис  Захистив з оцінкою  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_  оцінка підпис  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2015 |  | Виконав: Сидоров М.О.,  Залікова книжка № ІС-2123  Студент гр.. ІС-24, ФІОТ  3 курс |
| Керівник : ас. Сперкач М. О.  Допущений до захисту  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2015 \_\_\_\_\_\_\_\_  підпис  Захистив з оцінкою  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_  оцінка підпис  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2015 |  | Виконала: Тимчук А.О.,  Залікова книжка № ІС-2126  Студентка гр.. ІС-24, ФІОТ  3 курс |

Київ 2015

Національний технічний університет України “КПІ”

(назва вищого навчального закладу)

Кафедра автоматизованих систем обробки інформації і управління

Дисципліна Крос-платформене програмування

Напрям "Комп’ютерні науки", спеціальність "інформаційні управляючі системи та технології"

Курс 3 Група ІС-24 Семестр 6\_

**ЗАВДАННЯ**

**на розрахункову роботу студента**

Вереня Олександр Ігорович

(прізвище, ім’я, по батькові)

1. Тема проекту (роботи) Система складання колендарного плану виготовлення товарів

2. Строк здачі студентом закінченого проекту (роботи)

3. Вихідні дані до проекту (роботи) програмний код написаний мовою Java EE OS: Windows XP/7/8, база даних MySQL.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

1. Технічне завдання. 2.Пояснювальна записка 3. Інструкція програміста. 4. Інструкція системного програміста. 5. Інструкція користувача

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

Відео копії результатів, схема таблиць бази даних, діаграма варіантів використання,   
діаграма компонентів, DFD діаграма.

6. Дата видачі завдання 28. 02.2015 р.

 Національний технічний університет України “КПІ”

(назва вищого навчального закладу)

Кафедра автоматизованих систем обробки інформації і управління

Дисципліна Крос-платформене програмування

Напрям "Комп’ютерні науки", спеціальність "інформаційні управляючі системи та технології"

Курс 3 Група ІС-24 Семестр 6\_

**ЗАВДАННЯ**

**на розрахункову роботу студента**

Кмець Максим Ігорович

(прізвище, ім’я, по батькові)

1. Тема проекту (роботи) Система складання колендарного плану виготовлення товарів

2. Строк здачі студентом закінченого проекту (роботи)

3. Вихідні дані до проекту (роботи) програмний код написаний мовою Java EE OS: Windows XP/7/8, база даних MySQL.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

1. Технічне завдання. 2.Пояснювальна записка 3. Інструкція програміста. 4. Інструкція системного програміста. 5. Інструкція користувача

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

Відео копії результатів, схема таблиць бази даних, діаграма варіантів використання,   
діаграма компонентів, DFD діаграма.

6. Дата видачі завдання 28. 02.2015 р.

Національний технічний університет України “КПІ”

(назва вищого навчального закладу)

Кафедра автоматизованих систем обробки інформації і управління

Дисципліна Крос-платформене програмування

Напрям "Комп’ютерні науки", спеціальність "інформаційні управляючі системи та технології"

Курс 3 Група ІС-24 Семестр 6\_

**ЗАВДАННЯ**

**на розрахункову роботу студента**

Сидоров Максим Олегович

(прізвище, ім’я, по батькові)

1. Тема проекту (роботи) Система складання колендарного плану виготовлення товарів

2. Строк здачі студентом закінченого проекту (роботи)

3. Вихідні дані до проекту (роботи) програмний код написаний мовою Java EE OS: Windows XP/7/8, база даних MySQL.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

1. Технічне завдання. 2.Пояснювальна записка 3. Інструкція програміста. 4. Інструкція системного програміста. 5. Інструкція користувача

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

Відео копії результатів, схема таблиць бази даних, діаграма варіантів використання,   
діаграма компонентів, DFD діаграма.

6. Дата видачі завдання 28. 02.2015 р.

Національний технічний університет України “КПІ”

(назва вищого навчального закладу)

Кафедра автоматизованих систем обробки інформації і управління

Дисципліна Крос-платформене програмування

Напрям "Комп’ютерні науки", спеціальність "інформаційні управляючі системи та технології"

Курс 3 Група ІС-24 Семестр 6\_

**ЗАВДАННЯ**

**на розрахункову роботу студента**

Тимчук Андрій Олександрович

(прізвище, ім’я, по батькові)

1. Тема проекту (роботи) Система складання колендарного плану виготовлення товарів

2. Строк здачі студентом закінченого проекту (роботи)

3. Вихідні дані до проекту (роботи) програмний код написаний мовою Java EE OS: Windows XP/7/8, база даних MySQL.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

1. Технічне завдання. 2.Пояснювальна записка 3. Інструкція програміста. 4. Інструкція системного програміста. 5. Інструкція користувача

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

Відео копії результатів, схема таблиць бази даних, діаграма варіантів використання,   
діаграма компонентів, DFD діаграма.

6. Дата видачі завдання 28. 02.2015 р.

# Зміст

Зміст 7

1 Загальні положення 9

1.1 Опис предметного середовища 9

1.2 Опис процесу діяльності 10

1.3 Опис функціональної моделі 11

2 Рішення з інформаційного забезпечення 14

2.1 Вхідні дані 14

2.2 Вихідні дані 14

2.3 Опис інформаційного забезпечення 14

3 Рішення з програмного забезпечення 18

3.1 Опис вибраного програмного забезпечення 18

3.1.1 Короткий опис можливостей 18

3.1.2 Види діяльності, функції 18

3.2 Архітектура програмного забезпечення 19

3.2.1 Діаграми класів 19

3.2.2 Діаграма послідовності 21

3.2.3 Діаграма компонентів 23

3.3 Специфікація функцій 24

3.4 Вимоги до якості 24

3.4.1 Вимоги до надійності 24

3.4.2 Вимоги до програмного забезпечення системи 24

3.4.3 Вимоги до технічного забезпечення 25

4 Опис алгоритмів 26

4.1 Постановка задачі 26

4.2 Опис математичної моделі знаходження розкладу виготовлення товарів 26

4.3 Алгоритм рішення задачі знаходження розкладу виготовлення товарів 27

4.3.1 Уточнення та спрощення математичної моделі 27

4.3.2 Узагальнений алгоритм складання розкладу виготовлення товарів 28

4.3.3 Введення критеріїв. Створення альтернативних алгоритмів. 28

4.3.4 Опис критеріїв 29

4.4 Опис математичної моделі вибору оптимального складу компютеру 31

4.5 Алгоритм пошуку оптимального складу компютеру 32

4.5.1 Опис алгоритму 32

4.5.2 Тестування алгоритму. 33

5 Керівництво користувача 35

5.1 Інструкція користувача 35

5.2 Методика випробувань 38

5.3 Приклади модульних тестів 43

5.4 Приклади інтеграційних тестів 45

5 ВИСНОВОК 47

5.1 Результати роботи 47

5.2 Виконана функціональність відповідно до ТЗ 47

5.3 Невиконана функціональність відповідно до ТЗ 48

5.4 Перспективи розвитку програми 48

Додаток А 49

Календарний план 49

# 1 Загальні положення

## 1.1 Опис предметного середовища

Система «СКПВТ» призначена як для великих організацій, так і для звичайних користувачів, які хочуть зібрати собі комп’ютер, який відповідає усім їхнім потребам та можливостям. Цей ресурс дозволяє зареєстрованим користувачам створювати нові замовлення на збір копютерів (одного чи багатьох) та задавати строки на виконання цих замовлень. Також ця система дозволяє отримати своєчасну та швидку інформацію стосовно статусу нового замовлення, можливість чи неможливість його виконання, а також дату закінчення виконання замовлення. Кожен користувач, який створив нове замовлення може відмінити його, якщо воно ще не перейшло в стан виконуваного. У разі відмови у виконанні або успішному закінченню замовлення, користувач отримує своєчасне повідомлення на електронну пошту, зазначену при реєстрації.

Основними цілями створення системи «СКПВТ» є:

* Удосконалення роботи підприємства з виготовлення та збору комп’ютерів;
* Створення простого та інтуїтивно зрозумілого сервісу оформлення замовлень для клієнта заводу зі складання комп’ютерів за допомогою:
* Надання графічного конструктора складання типу комп’ютера;
* Надання критеріїв оцінки готового комп’ютера(вартість, потужність, якість);
* Підвищення продуктивності роботи планувальників заводу зі складання комп’ютерів за допомогою:
* Надання зручного інтерфейсу зворотнього зв’язку з замовником стосовно розглянутих замовлень;
* Надання автоматизованої системи розподілу працівників за прийнятими замовленнями;
* Подача розподілу замовлень у вигляді календарного плану;
* Створення інтерфейсу керування ресурсами(робітники, деталі) заводу зі складання комп’ютерів

## 1.2 Опис процесу діяльності

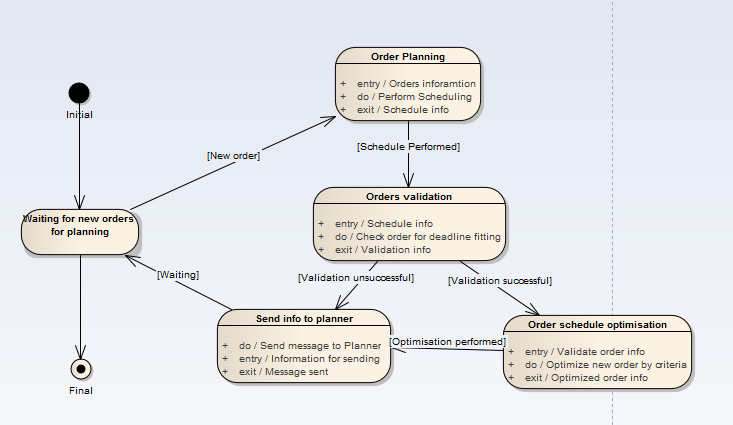
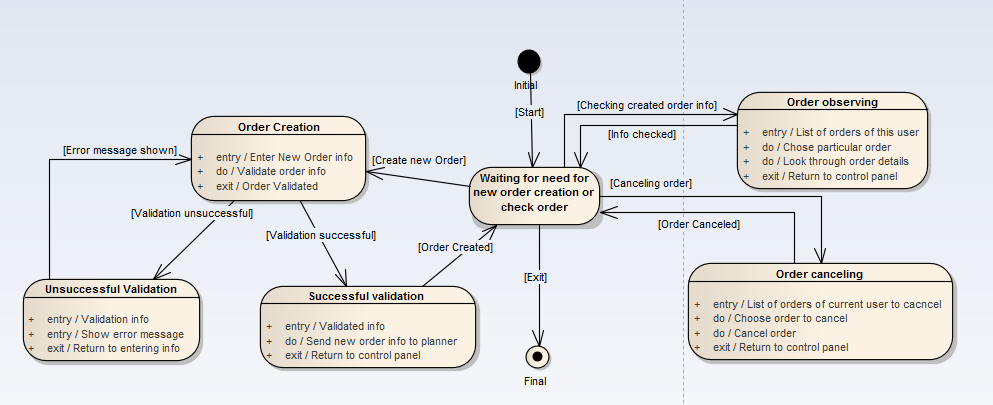


Рисунок 1.2.1 – Схема структурна станів підсистеми побудови розкладів

 Рисунок 1.2.2 – діаграма станів підсистеми створення замовлення

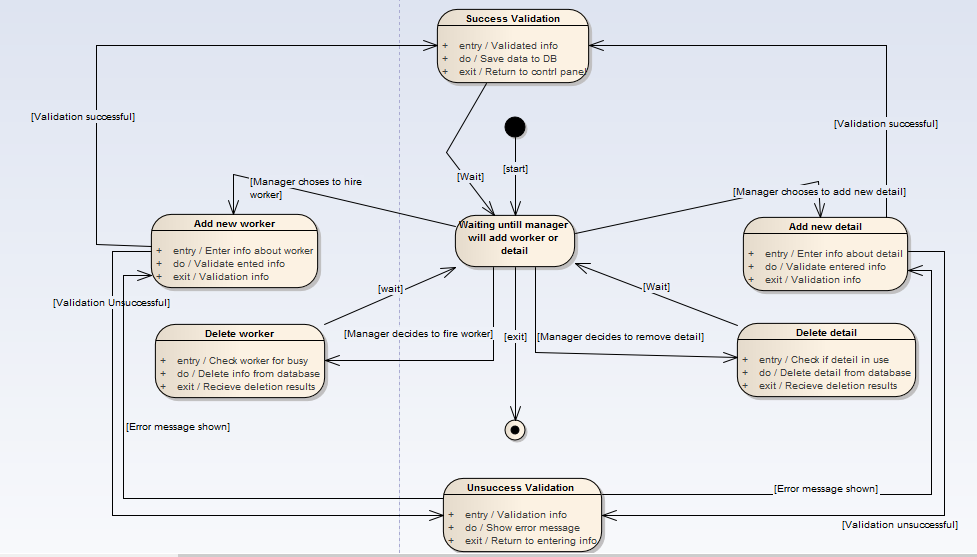


Рисунок 1.2.3 – діаграма станів підсистеми керування ресурсами

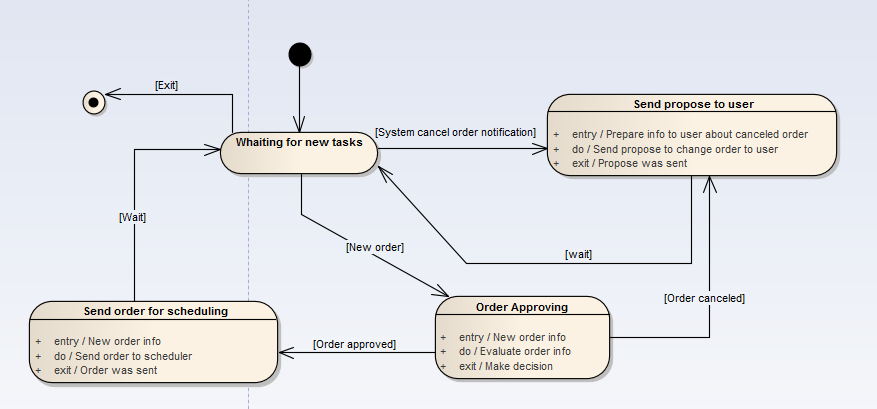


Рисунок 1.2.4 – діаграма станів підсистеми керування замовленнями

## 1.3 Опис функціональної моделі

Функціональна модель описана за допомогою діагрвми варіантів використання, яка зображена на рисунку 1.3.1.

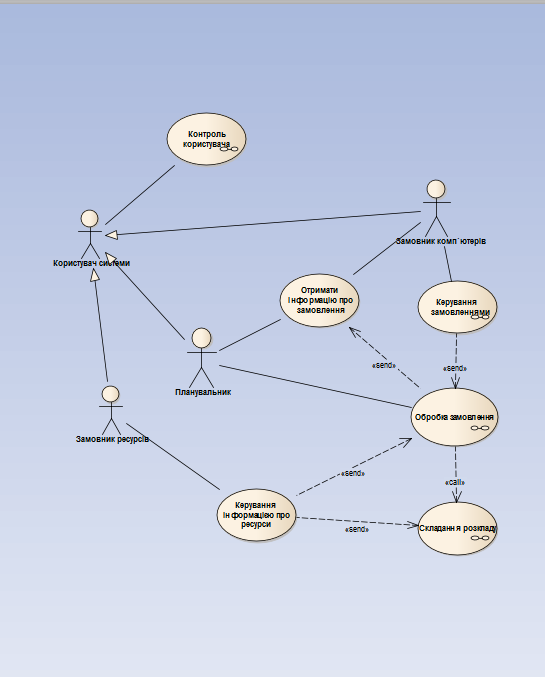


Рисунок 1.3.1 – Діаграма варіантів використання

Акторами системи є: Замовник комп’ютерів, планувальник, замовник деталей*.* Актори, варіанти використання та їх описи дій описані у таблиці 1.3.1

Таблиця 1.3.1

|  |  |  |
| --- | --- | --- |
| **Актор** | **Варіант використання** | **Опис дії варіанта використання** |
| Замовник комп’ютерів | Керування замовленнями | Конструювання типу комп’ютера з наданих компонент, вказання ватрості, потужності та якості комп’ютера, а також термін виконання замовлення;перегляд статусу замовлення; скасування замовлення |
| Отримати інформацію про замовлення | Висилання на пошту замовника інформації про статус замовлення та відображення на сторінці замовника про статус замовлення. |
| Планувальник | Обробка замовлення | Пошук потрібних деталей комп’ютера згідно з критеріями вартості потужності та якості, вказаними замовником, планування розкладів роботи; складання розкладу. |
| Отримати Інформацію про замовлення | Перегляд списоку запланованих замовлень, замовлень ,що виконуються, є виконаними, а також скасовані замовлення. |
| Замовник ресурсів | Керування інформацією про ресурси | Додавання, редагування та видалення записів про наявні виробничі ресурси(деталі, робітники) на підприємстві; найм робочих, звільнення робочих. |

# 2 Рішення з інформаційного забезпечення

## 2.1 Вхідні дані

Первісні дані вводяться в систему користувачем, через веб-інтерфейс системи.

**Дані, які надходять від користувача**. Першим чином користувач при реєстрації створює ***логін*** та ***пароль***. Далі, якщо користувач захоче створити нове замовлення, то він передає інформацію про бажане наповнення комп’ютера, якість, потужність, ціну та кількість таких комп’ютерів. Якщо користувач хоче відмінити замовлення, він надає інформацію про відмову від замовлення.

**Дані, які надходять від планувальника**. Планувальник відповідає за модерацію існуючих замовлень. Тому дані які надаватимуться планувальником такі:

* Відправлені на планування та репланування замовлення;
* Замовлення, які планувальник відмінив;
* Замовлення, які пропонуються змінити користувачем;
* та ін.

**Дані, які надходять від менеджера ресурсів**. Менеджер ресурсів відповідає за регулювання таких ресурсів як робітники та деталі, тому дані, які надає менеджер ресурсів наступні:

* інформація про деталь, яку необхідно додати до ситеми;
* інформація про деталь, яку необхідно видалити з ситеми;
* інформацію про нового робітника системи;
* інформацію про робітника, якого звільняють;

## 2.2 Вихідні дані

Вся вихідна інформація надається користувачеві через вікно браузера. Це може бути інформація про поточне замовлення, його статус. Також сплановані замовлення відображаються у календарі планувальника. У відділі ресурсів можна отримати інформацію про існуючих найнятих робітників та деталі, які може виготовляти система.

## 2.3 Опис інформаційного забезпечення

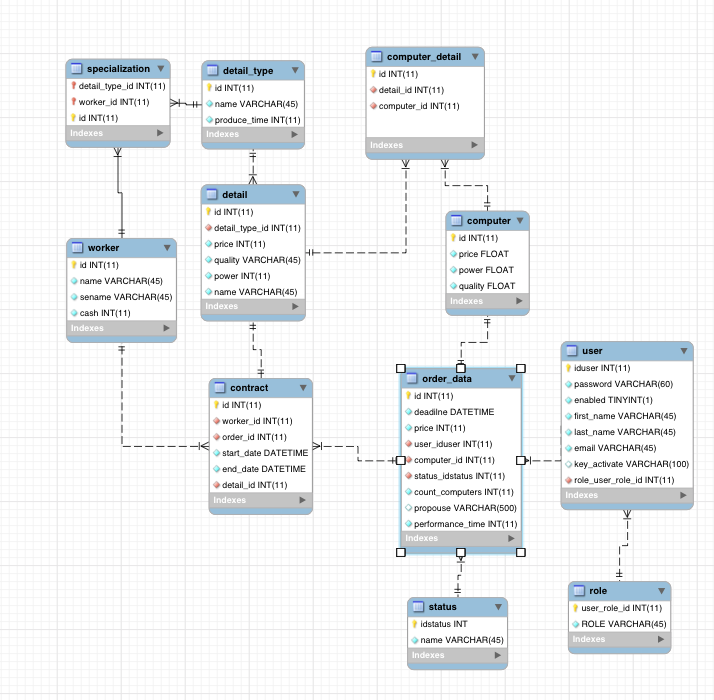
БД побудована таким чином, шо кожному об’єкту системи відповідає таблиця. Якщо користувач створює замовлення, то воно зберігається у відповідній таблиці. Інформація про склад комп’ютерів замовлення, робітників, та котракти на виконання замовлення з робітниками зберігаються у відповідних таблицях. ER діаграма бази данних наведена нижче.

Рисунок 2.3.1 – Структурна схема концептуальної моделі БД

У таблиці 2.3.1 наведено назви полів кожної з таблиць бази данних та їх опис.

Таблиця 2.3.1 – Опис таблиць БД

| **Назва таблиці** | **Поля таблиці** | **Тип даних поля** | **Логічний зміст поля** |
| --- | --- | --- | --- |
| user | Id | Integer | Id – який відповідає користувачу |
| first\_name | Characters(45) | Ім’я користувача |
| last\_name | Characters(45) | Прізвище користувача |
| enabled | tinyint(1) | Чи був користувач варифікован |
| password | Characters(60) | Пароль користувача |
| email | Characters(45) | Email користувача |
| role | id | Integer | Id ролі |
| role | varchar(45) | Назва ролі |
| order\_data | Id | INT(11) | Id замовлення |
| deadline | DATETIME | Дата дедлайну по замовленню |
| price | INT(11) | Загальна ціна замовлення |
| count\_computers | INT(11) | Кількість комп’ютерів у замовленні |
| performance\_time | INT(11) | Загальний час на виконання замовлення |
| propose | VARCHAR(500) | Якщо поле null, то поланування замовлення успішне, інакше містить пропозицію про зміну замовлення. |
| status | Id | INT | Id статусу замовлення |
| name | VARCHAR(45) | Назва статусу |
| computer | Id | Integer | Id комп’ютера |
| price | FLOAT | Ціна одного комп’ютера заданої конфігурації |
| power | FLOAT | Бажана потужність комп’ютера |
| quality | FLOAT | Бажана якість комп’ютера |
| detail | Id | Integer | Id – який відповідає статті |
| price | INT | Ціна деталі |
| quality | VARCHAR(45) | Якість деталі |
| power | Integer | Потужність деталі |
| name | VARCHAR(45) | Назва деталі |
| detail\_type | Id | Integer | Id типу деталі |
| name | VARCHAR(45) | Назва типу деталі |
| produce\_time | INT | Час виготовлення деталі |
| specialization | Id | INT | Id спеціалізації |
| detail\_type\_id | INT | id типу деталі |
| worker\_id | INT | id робітника |
| worker | Id | INT | Id робітника |
| name | VARCHAR(45) | Ім’я робітника |
| sename | VARCHAR(45) | Прізвище робітника |
| cash | INT | Заробітна плата робітника |
| contract | Id | INT | Id кнтракту |
| start\_date | DATETIME | Початок виконання завдання |
| end\_date | DATETIME | Кінець виконання завдання |
| order\_id | INT | id замовлення |
| worker\_id | INT | id робітника |

# 3 Рішення з програмного забезпечення

## 3.1 Опис вибраного програмного забезпечення

Система «СКПВТ» розроблена для використання у сфері виробництва. А саме для підприємств, які прагнуть автоматизувати систему планування виготовлення компьютерів.

Описати фреймворки

Відступи, продовження таблиць

Вказати рисунки у тексті

### 3.1.1 Короткий опис можливостей

Система «СКПВТ» передбачає такі можливості:

* Створення нових замовлень;
* Створення календарного плану для виготовлення товарів;
* Менеджмент нових замовлень;
* Менеджмент підприємства, шляхом заміни робітників та деталей;

### 3.1.2 Види діяльності, функції

Система «СКПВТ» призначена як для великих організацій, так і для звичайних користувачів, які хочуть зібрати собі комп’ютер, який відповідає усім їхнім потребам та можливостям. Цей ресурс дозволяє зареєстрованим користувачам створювати нові замовлення на збір копютерів (одного чи багатьох) та задавати строки на виконання цих замовлень. Також ця система дозволяє отримати своєчасну та швидку інформацію стосовно статусу нового замовлення, можливість чи неможливість його виконання, а також дату закінчення виконання замовлення. Кожен користувач, який створив нове замовлення може відмінити його, якщо воно ще не перейшло в стан виконуваного. У разі відмови у виконанні або успішному закінченню замовлення, користувач отримує своєчасне повідомлення на електронну пошту, зазначену при реєстрації.

Вимоги до технічного забезпечення серверу:

* Процесор – 1 Ггц, 1 ядро ЦП або краще
* Оперативна пам’ять не менш ніж 256 Мб
* Не менше ніж 5 ГБ ПЗУ
* 10 Мбіт/сек доступу до мережі Інтернет

Вимоги до технічного забезпечення серверу бази даних:

* Процесор – 1 Ггц, 1 ядро ЦП або краще
* Оперативна пам’ять не менш ніж 256 Мб
* Не менше ніж 5 ГБ ПЗУ
* 10 Мбіт/сек доступу до мережі Інтернет

Вимоги до програмного забезпечення серверу:

* Java EE 7;
* Tomcat 7 або вище;

Вимоги до програмного забезпечення серверу бази даних:

* СУБД MySQL

## 3.2 Архітектура програмного забезпечення

### 3.2.1 Діаграми класів

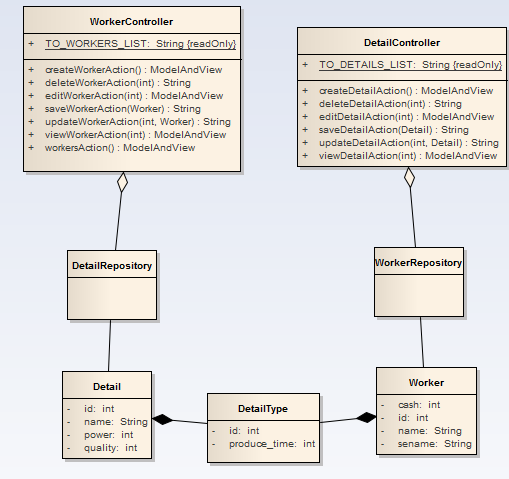


Рисунок 3.2.1.1 - Діаграмма класів підсистеми керування ресурсами

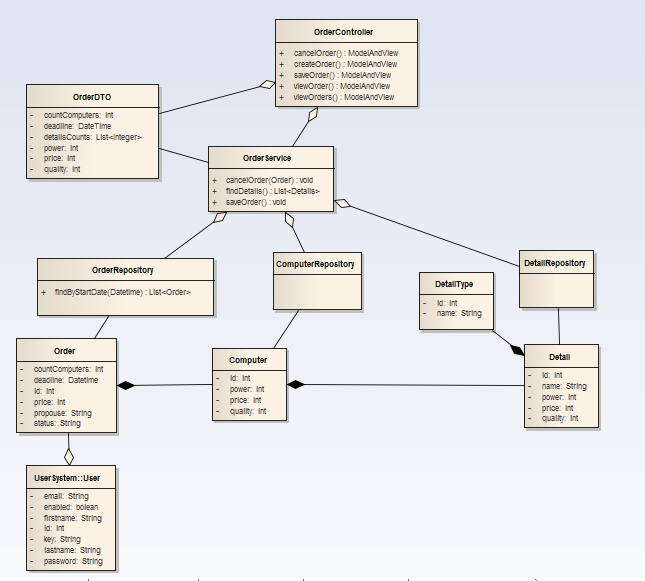


Рисунок 3.2.1.2 - Діаграмма класів підсистеми створення замовленнями

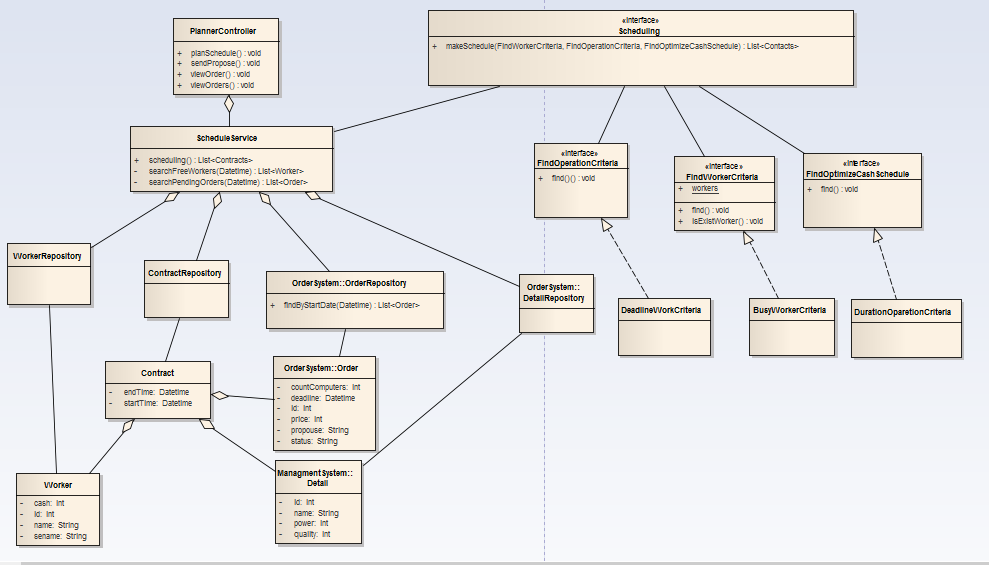


Рисунок 3.2.1.3 - Діаграмма класів підсистеми планувальника

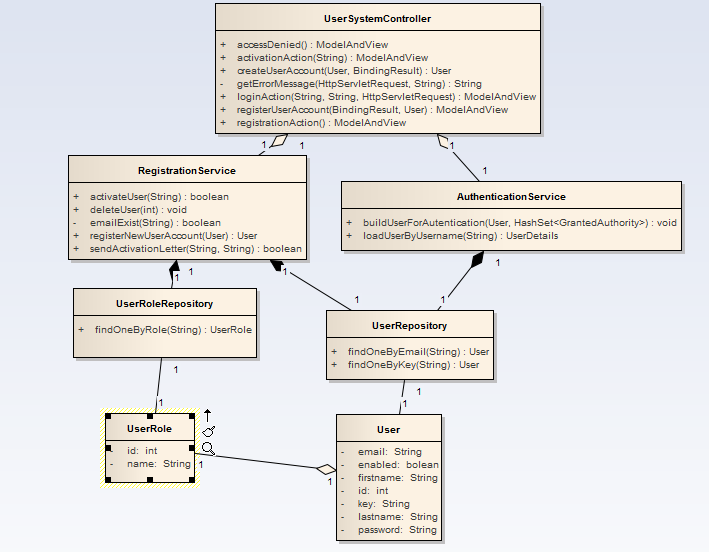


Рисунок 3.2.1.4 - Діаграмма класів реєістрації та аунтифікації

### 3.2.2 Діаграма послідовності

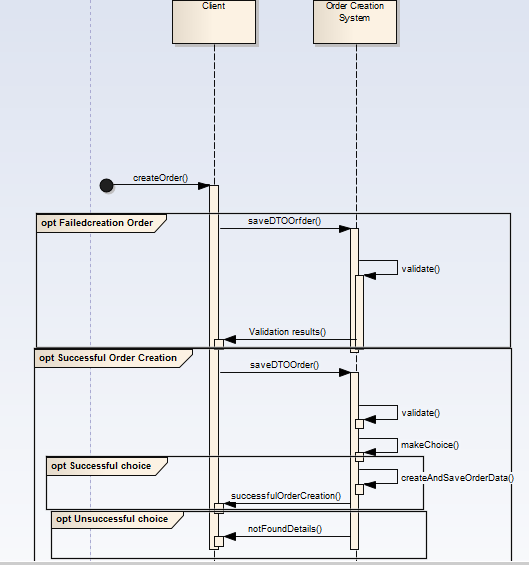


Рисунок 3.2.2.1 - Діаграмма послідовності підсистеми клієнта

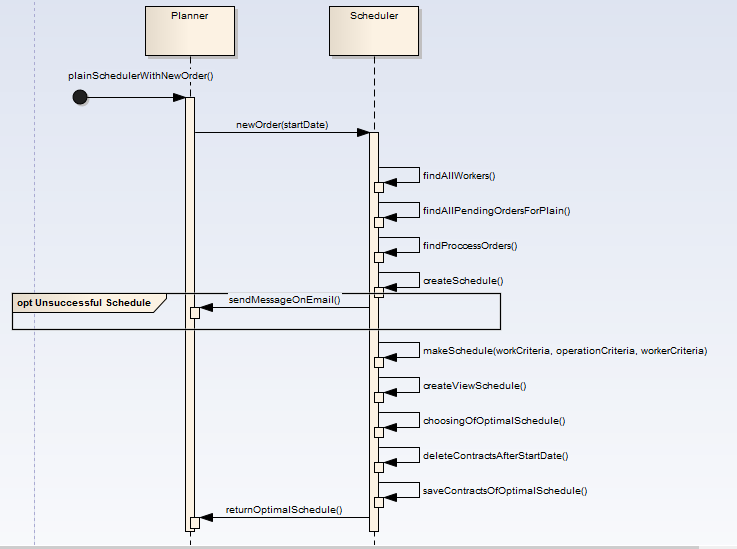


Рисунок 3.2.2.2 - Діаграмма послідовності підсистеми планувальника та створення розкладів

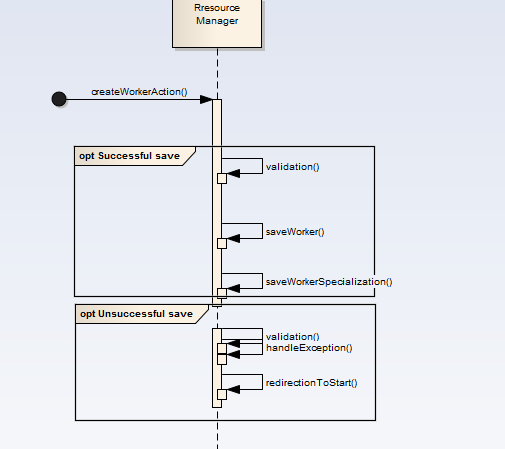


Рисунок 3.2.2.3 - Діаграмма послідовності підсистеми управління ресурсами

### 3.2.3 Діаграма компонентів

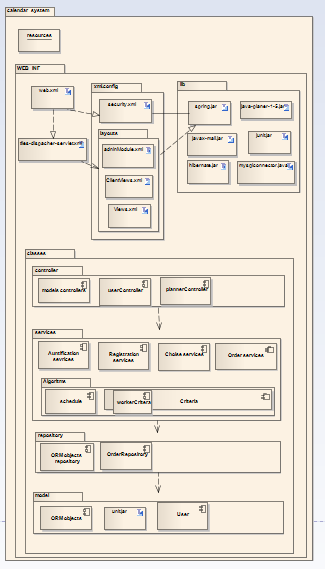


Рисунок 3.2.3.1 - Діаграмма компонентів

## 3.3 Специфікація функцій

Задача створення замовлення та отримання результату:

* Отримання від замовника критеріїв по створенню комп’ютерів (кількість, термін, якість, ціна, потужність);
* Відправлення замовлення до планувальника;
* Отримання результату планування;

Задача складання календарного плану:

* Перевірка можливості виконання замовлення;
* Перегляд альтернативних варіантів замовлення;
* Додавання в календарний план нового замовлення;
* Видалення існуючих замовлень;
* Перегляд календарного плану;

Задача керування інформацією про ресурси:

* Зміна, видалення, додавання інформації про наявних майстрів;
* Зміна, видалення, додавання інформації про деталі;

Задача аутентифікації в системі:

* Реєстрація замовників, планувальників та замовників ресурсів;
* Вхід в систему з правами відповідно до вищенаведених ролей;

## 3.4 Вимоги до якості

До програмного забеспечення були поставленні наступні вимоги:

### 3.4.1 Вимоги до надійності

* Реагування системи на введення некоректного діапазону вхідних даних:
  + Не здійснення відправлення замовлення;
  + Повідомлення про помилку.
* Вихідні дані являють собою календарний план виготовлення товарів, терміни виконання, в якому відповідають отриманим замовленням та не суперечить їм.
* Функціональні вимоги покриті тестовими сценаріями;

|  |
| --- |
| 3.4.2 Вимоги до програмного забезпечення системи |

Програмне забезпечення, що використовується при розробці, та бібліотеки програмних кодів повинні бути поширені та загальнодоступні.

Платформа: крос-браузерна платформа з підтримкою JVM.

Версія JVM 5.0 і вище .

Мова написання коду програми: Java.

### 3.4.3 Вимоги до технічного забезпечення

Вимоги до технічних характеристик ПК користувача:

* Процесор – Intel Pentium 1.5 ГГц;
* Обсяг оперативної памяти – 256 Мб;
* Дискова підсистема – 40 Гб;

# 4 Опис алгоритмів

## 4.1 Постановка задачі

Задана компанія, що займається складанням комп’ютерів. Кожне замовлення, що надходить до компанії, є набором деталей певного типу та кількості, збиранням яких займаються робітники, що мають спеціалізацію(перерахунок усіх типів деталей, що може зібрати заданий робітник). Необхідно знайти розклад виконання замовлень, що надходять до компанії, при чому після складання розкладу з замовником укладається контракт, котрий гарантує виконання даного замовлення( тобто наступні етапи знаходження розкладу, мають точно виконати всі ті замовлення, з якими укладено контракт).

Замовлення має наступні особливості:

* Порядок збору деталей є не важливим;
* З попереднього пункту випливає, що збір комп’ютера можна здійснювати паралельно;
* Збір деталі є неперервною операцією;
* Час збору деталі залежить лише від її типу;

Робітники мають наступні особливості:

* Робочий графік – з 10:00 години ранку до 18:00 години вечора;

## 4.2 Опис математичної моделі знаходження розкладу виготовлення товарів

Існує *n* замовлень збору комп’ютера, кожен з яких містить наступну інформацію:

*d i* – крайній термін виконання *i*-ого замовлення ;

*Li* – множина деталей, що необхідно зібрати в *і*-ому замовленні;

| *Li* | *= m –* кількість деталей, що необхідно зібрати в *і*-ому замовленні;

Кожна деталь, в свою чергу, має наступні параметри:

*u –* тип деталі;

*lu* – час збирання деталі *u*-ого типу(задано в хвилинах);

Збирання комп’ютерів здійснює *k* робітників, при чому задано:

*cs* – зарплатня *s*-ого робітника за 1 годину роботи;

*Ks* – множина типів деталей, що може збирати даний робітник( спеціалізація робітника);

Також для спрощення розгляду задачі введемо наступні параметри:

tij – час збирання *j*-ої деталі в *i*-ому замовленні;

cs’ - зарплатня *s*-ого робітника за 1 хвилину роботи;

*i∈[1,n]; j∈[1,m]; s∈[1,k]; u∈[1,p]*

Необхідно знайти наступні параметри:

T\_bijs – час початку збору *j*-ої деталі в *i*-ому замовленні *s*–им робітником;

T\_eijs – час кінця збору *j*-ої деталі в *i*-ому замовленні *s*–им робітником;

Також для спрощення введемо наступні параметри:

– фактичний час завершення виконання замовлення після знаходження розкладу; *(1)*

Для задовільного розкладу мають виконуватись наступні обмеження:

– фактичний час завершення виконання розкладу не перевищує крайнього терміну виконання замовлення; *(2)*

Цільовою функцією є:

– мінімізація витрат на заробітну плату робітників; (3)

## 4.3 Алгоритм рішення задачі знаходження розкладу виготовлення товарів

### 4.3.1 Уточнення та спрощення математичної моделі

Розглянувши постановку задачі очевидно, що дана система є динамічною, тобто в будь-який момент часу може прийти будь-яка кількість замовлень. Складання таких розкладів є вкрай складною задачею, тому перетворимо умову задачі наступним чином:

* Початок складання розкладу буде здійснено лише для одного замовлення, виконання якого ще не є спланованим. Надалі – нове замовлення;
* Час планування визначається як 10:00 година ранку наступного дня від початку складання розкладу. Надалі введемо термін – «точка відліку»;
* «Точка відліку» поділяє всі замовлення, що знаходяться в компанії, на 3 категорії: виконані(завершені) замовлення, не виконані(нове замовлення, частково завершені замовлення та замовлення, що очікують виконання) і не сплановані(замовлення, розгляд яких ще не відбувся);
* Складання нового розкладу буде відбуватися для всіх замовлень другої категорії;

Дані умови показані на наступному рисунку:

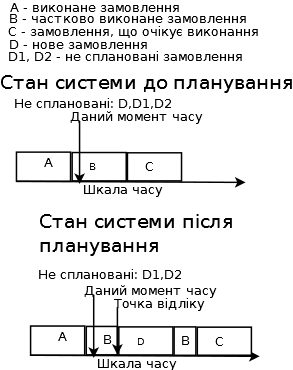


Рисунок 4.3.1.1 – статична система прийому замовлень

Таким чином, отримали статичний характер системи – в певний момент часу знаходження розкладу буде відбуватися для відомих та визначених замовлень.

### 4.3.2 Узагальнений алгоритм складання розкладу виготовлення товарів

На основі отриманої інформації складемо узагальнений алгоритм знаходження розкладу виготовлення комп’ютерів:

1. Знайдемо всі замовлення 2 категорії та запишемо їх до множини пошуку робіт(U)
2. Поки U не порожня: виберемо з U найкращу роботу та вилучимо її з множини пошуку
   1. Знайдемо всі не виконані деталі(частково виконані замовлення, містять в собі об’єм робіт, що вже був зроблений) в даній роботі та запишемо їх до множини пошуку операцій(V)
   2. Поки V не порожня: виберемо з V найкращу операцію та вилучимо її з множини пошуку
      1. Серед множини доступних робітників для даної операції виберемо найкращого робітника(X)
      2. До шкали робочого процесу X запишемо дану операцію
      3. Зі шкали робочого процесу X отримаємо T\_bijs та T\_eijs та запишемо до множини-результату(R)
3. Перевіримо чи виконується для R умова *(2)*

### 4.3.3 Введення критеріїв. Створення альтернативних алгоритмів.

Цільовою функцією для нашої моделі є отримання якомога «дешевого» розкладу, але при цьому не враховується ряд інших важливих факторів(серед яких, наприклад, – дотримання директивних термінів). Даний аспект вирішимо наступним чином: в узагальненому алгоритмі не визначено як саме відбувається пошук найкращої роботи(операції, робітника), тому можна скласти критерії пошуку, котрі будуть покращувати кінцевий розклад за своїми визначеними локальними параметрами(наприклад, мінімізація максимального запізнення виконання замовлень) та комбінувати їх для створення альтернативних розкладів, а глобальний критерій(цільова функція моделі) буде вибирати серед них найвигідніший.

### 4.3.4 Опис критеріїв

1. *Критерій вибору роботи*
   1. *Вибір роботи за зростанням директивного терміну(Автор: Кмець Максим)*

Даний критерій впорядковує роботи наступним чином: робота *i* передує роботі *j*, якщо *di<dj*. Таким чином забезпечується мінімізація максимального запізнення, що є ідеальним критерієм для сортування робіт, тому що задовільним вважається такий розклад, котрий виконує всі розклади вчасно. А отже, максимальне запізнення у такому розкладі рівне 0, тобто очевидно, що якщо скласти задовільний розклад не можливо в будь-якому випадку, то величина запізнення( максимальна, сумарна, середня) не грає ніякої ролі, але якщо розклад можна скласти задовільним, то мінімізація максимального запізнення знайде такий розклад гарантовано.

Доведення, що перестановка двох робіт за зростанням директивного терміну зменшує максимальний час запізнення:

Не порушуючи загальності припустимо, що в певному розкладі лише 2 роботи *i* та *j* не є впорядкованими за директивним терміном( *i* передує роботі *j*, але *dj<di*) і максимальне запізнення дорівнює *l*(очевидно, що відповідає запізненню роботи j). Тоді при перестановці *i* та *j* місцями, запізнення для *j* зменшиться, а запізнення для *i* буде на *di-dj* меншим від старого запізнення для *j.* Таким чином максимальне запізнення стане менше.

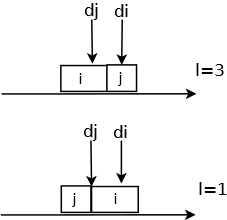


Рисунок 4.3.4.1 – впорядкування операцій за дерективним терміном

1. *Критерій вибору операції*

Критерії вибору операції є тісно пов’язаними з вибором робітника, що виконує дану операцію, тому представлені нижче критерії базуються на пункт 3 даного розділу

* 1. *Вибір операції за спаданням тривалості її виконання(Автор: Тимчук Андрій)*

Впорядкування для даного критерію відбувається так: операція *i* передує операції *j*(в даному пункті приналежність до замовлення немає значення), якщо *ti>tj*. Суть критерію полягає в наступному: нехай існує два робітника A та B та три операції *i,j,k* з тривалістю *ti > tj, ti > tk, tj~tk*. Можливі наступні варіанти розподілу операцій:

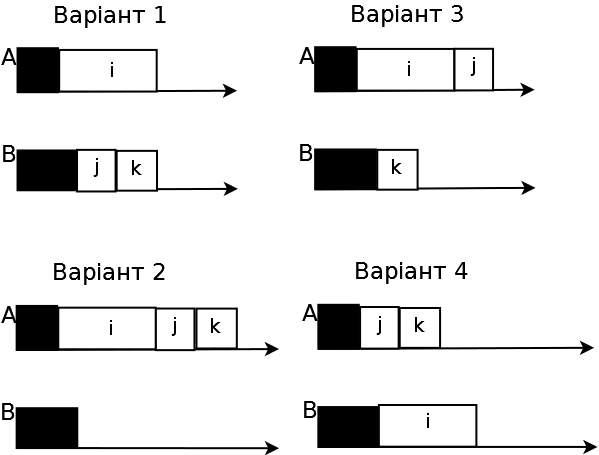


Рисунок 4.3.4.1 – варіанти розподілу операцій

Варіант 1 відповідає найкращому розподіленню операцій між робітниками і можливий лише за умови призначення найдовшої операції найбільш незайнятому робітнику( варіанти 2 та 3 надмірно навантажують робітника A, а варіант 4 призначає найдовшу операцію більш зайнятому робітнику. Є гіршим за 1 варіант як мінімум на початкову різницю зайнятості робітників A та B ).

1. *Критерій вибору робітника*
   1. *Вибір найменш зайнятого робітника(Автор: Андрій Тимчук)*

Для даного класу задач важливою властивістю є щільність розкладу, тобто максимальна завантаженість усіх робітників підприємства. Забезпечити таку властивість допомагає наступний критерій: операція *i* призначається такому з робітників *A* та *B(робітники можуть виконати операцію i),* у якого *cA<cB*, де *c* – функція зайнятості. Отже, на кожній ітерації робітники будуть рівномірно розподіляти операції між собою, забезпечуючи сумарне зменшення тривалості виконання замовлення.

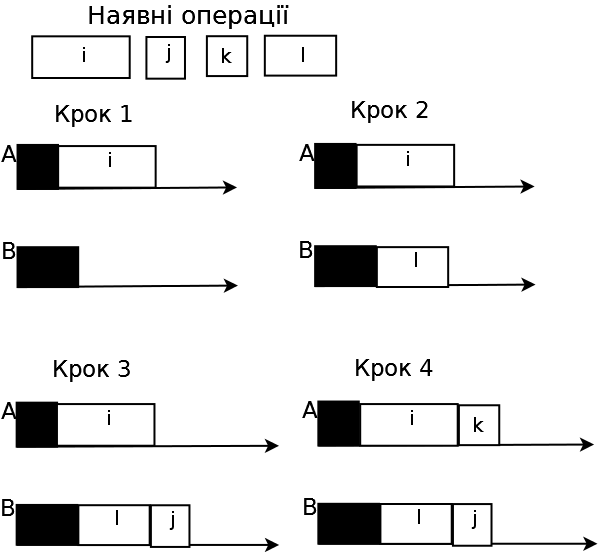


Рисунок 4.3.4.1 – розподіл операцій між робітниками

* 1. *Вибір найбільш дешевого робітника(Автор: Вереня Олександр)*

Цей критерій є модифікацією попереднього, а саме: якщо два робітника A та B є найбільш незайнятими, то вибір відбувається на основі погодинної заробітної плати, тобто вибирається «дешевший» робітник, що дозволяє покращити розклад з точки зору цільової функції вихідної математичної моделі.

## 4.4 Опис математичної моделі вибору оптимального складу компютеру

Існує *n* типів деталей та mi деталей кожного з n типів. Для кожної деталей відомі наступні параметри:

*s ij* – ціна *j*-ої деталі *i-ого* типу;

*q ij* –якість *j*-ої деталі *i-ого* типу;

*p ij* –потужність *j*-ої деталі *i-ого* типу;

*Li* – множина деталей для замовлення;

| *Li* | *= m –* кількість деталей, що необхідно зібрати;

Для кожного замовлення відомо:

S – максимальну ціну замовлення.

Q – середню якість деталей замовлення.

P – середню потужність деталі.

Тоді задача зводиться до знаходження min та маємо наступні обмеження:

## 4.5 Алгоритм пошуку оптимального складу компютеру

### 4.5.1 Опис алгоритму

Зважаючи на те, що для рішення цієї задачі методом повного перебору необхідно занадто багато часу, ( наприклад, для 5 видів деталей, по 120 деталей кожного типу = 24883200000 ) було вирішено реалізувати для цієї еврестичний алгоритм. А саме генетичний алгоритм.

Для опису генетичного алгоритму введемо наступні поняття:

Хромосома – певна деталь яка входить до складу кaопьютера.

Особина – набір деталей (хромосом) певного типу характерного для данного замовлення.

Популяція – набір компютерів (осіб) різного складу, які задовільняють початковим умовам.

Якщо особина задовольняє умовам поставленої задачі то будемо рахувати, що особина жива.

Можна виділити такі етапи генетичного алгоритму:

1. Створення початкової популяції:
2. Повторювання до виконання критерію зупинки алгоритму:
   1. Вибір індивідів із поточної популяції (селекція)
   2. Схрещення або/та мутація
   3. Обчислення функції пристосовуваності для всіх осіб
   4. Формування нового покоління

Для початку роботи алгоритму необхідно згенерувати початковий набір особин – початкову популяцію. Для цього згенеруємо 20 випадкових особин. Якщо за 500 спроб невдалося згенерувати жодної живої особини, то будемо рахувати, що задача не має ріщення.

Після створення початкової популяції будемо виконувати 1000 кроків еволюції для покращення популяції.

На кожному кроці єволюції обираємо дві особини з поточної популяції. Випадковим чином вибираємо точку схрещування та утворюємо масив з двох особин нащадків. Для кожного з нащадків перевіряємо чи живі назадки. Якщо так, то заносимо нащадка до популяції та видаляємо з популяції найгіршу особину. Якщо нащадок не задовольняє умовам популяції обираємо з його деталей (хромосом) таку *i-ту*, для якої:

Цю хромосому ми змінюємо на випадкову іншу деталь з набору відповідних хромосом. Якщо після реанімації особина задовольняє умовам, то повертаємо її у популяцію.

Для порівняя двох особин з однаковими значеннями цільової функції обираємо ту у якої сума критеріїв якості та потужності більша.

Умовою припинення еволюційних кроків є вичерпання числа поколінь, що відпущені на еволюцію. У нашому випадку 1000 кроків.

Після отримання кінцевої популяції обераємо з неї особину яка найбільше відповідає критеріям виживання (має найменший показник для цільової функції )

### 4.5.2 Тестування алгоритму.

Для тестування алгоритму була створена база данних деталей, налічувала 5 типів деталей та 10 деталей кожного типу та проведено 3 типи випробувань:

1. Задана ціна у 2000 одиниць та показники якості та потужності 2 та 1 відповідно.
2. Задана ціна у 4000 одиниць та показники якості та потужності 3 та 3 відповідно.
3. Задана ціна у 5000 одиниць на показники якості та потужності 4 та 4 відповідно.

Для кожного варіанту тестування тест проводився 10 разів.

Результати першого тесту наведені у таблиці 4.5.2.1

Таблиця 4.5.2.1

|  |  |  |  |
| --- | --- | --- | --- |
| № | Якість | Потужність | Ціна |
| 1 | 3.0 | 2.6 | 840 |
| 2 | 3.6 | 1.4 | 1100 |
| 3 | 3.0 | 2.6 | 840 |
| 4 | 2.2 | 3.4 | 840 |
| 5 | 3.6 | 2.2 | 1250 |
| 6 | 3.2 | 2.8 | 940 |
| 7 | 2.4 | 3.6 | 940 |
| 8 | 2.2 | 3.4 | 840 |
| 9 | 2.2 | 3.4 | 840 |
| 10 | 2.6 | 2.0 | 940 |

Методом повного перебору був знайдений найкращій результат: 840. Для цього випадку алгоритм в середньому давав відповідь на 11% менш вигідну ніж оптимальна ціна.

Для двох інших випадків було отримано результати на 33% та 10% відповідно. Результати зображені на рисунку 4.5.2.1.

Рисунок 4.5.2.1 – відношення абсолютних розвязків до розязків знайдених генетичним алгоритмом

У середньому наша реалізація генетичного алгоритму на тестових данних давала не більше 20% відхилення від результату знайденим методом повного перебору, при цьому зменшуючи кількість ітерацій порівняння з 105 до 103, тобто у 100 разів.

# 5 Керівництво користувача

## 5.1 Інструкція користувача

Інтерфейс системи «СКПВТ» є логічним та простим. Тому проблем з користуванням ресурсом виникнути не повинно.

Тому, проаналізувавши інтерфейс, було прийнято рішення у керівництві користувача описати і продемонструвати процес верифікації користувача. Робота з системою починається з вікна яке ми продемонстрували на рисунку 4.1.

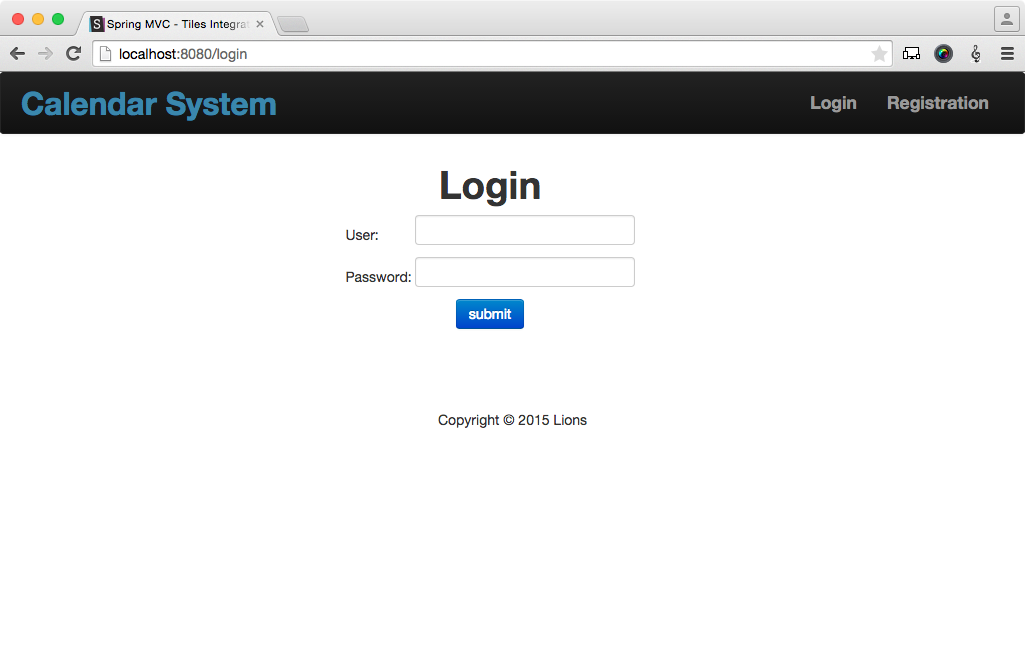


Рисунок 5.1.1 – Екранна форма реэстрації у «СКПВТ»

Зараєстрований користувач може виконати вхід в систему. Користувач, який ще не зареєструвався може це зробити перешовши до форми реєстрації.

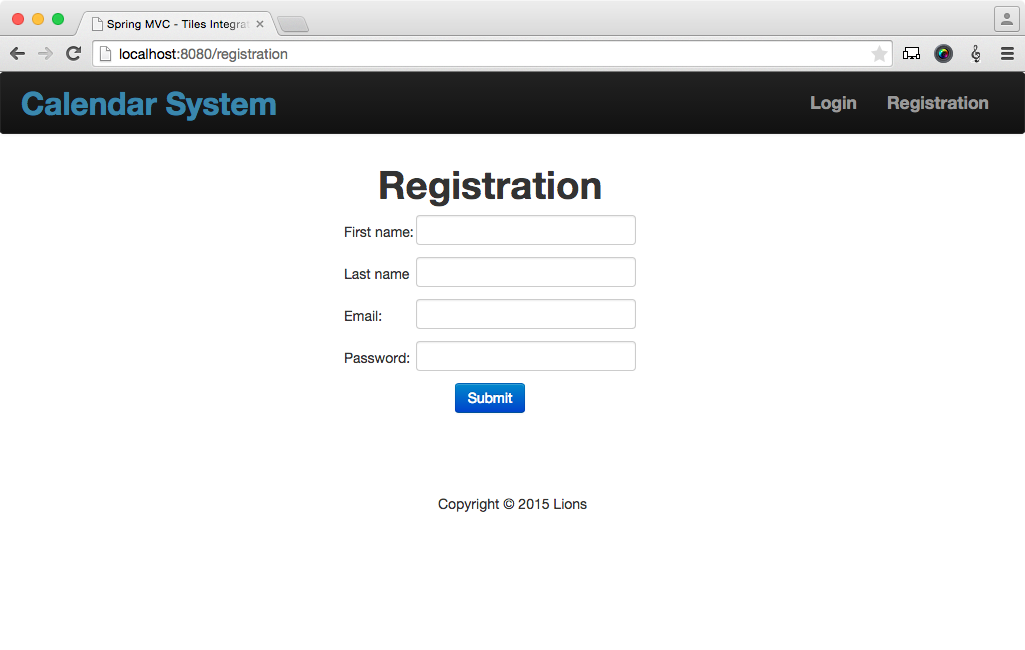


Рисунок 5.1.2 – Екранна копія форми реєстрації.

Після того як користувач залогінився він опиняється на сторінці списку своїх замовлень. За допомогою цієї сторінки користувач може скасувати своє замовлення або отримати змогу переглянути деталі свого замовлення.

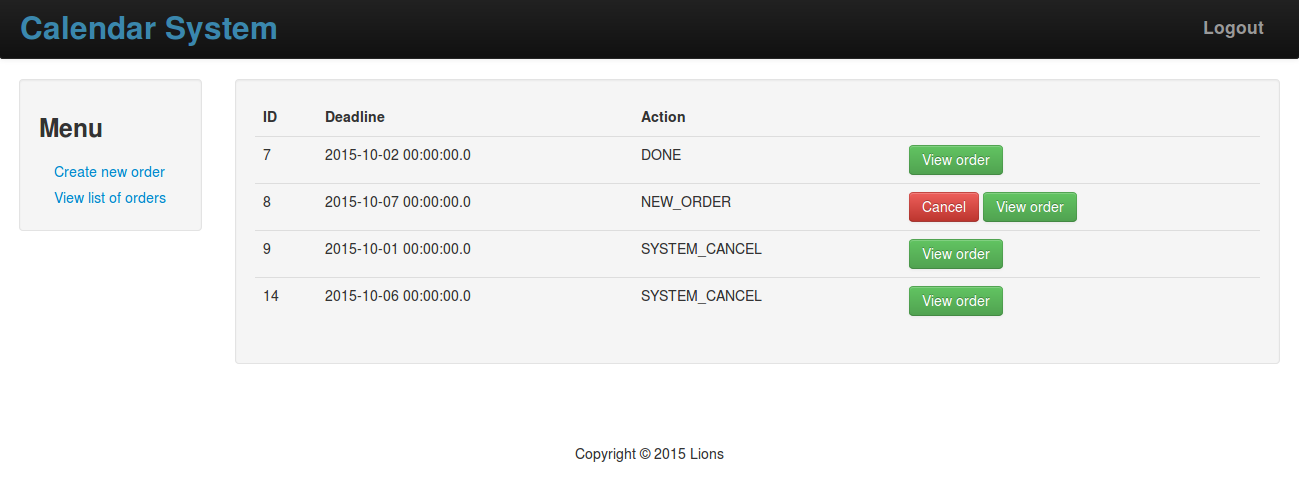


Рисунок 5.1.3 – Сторінка перегляду списку замовлень

Для того щоб переглянути детальну інформацію про замовлення необхіжно натиснути на кнопку “View order”.

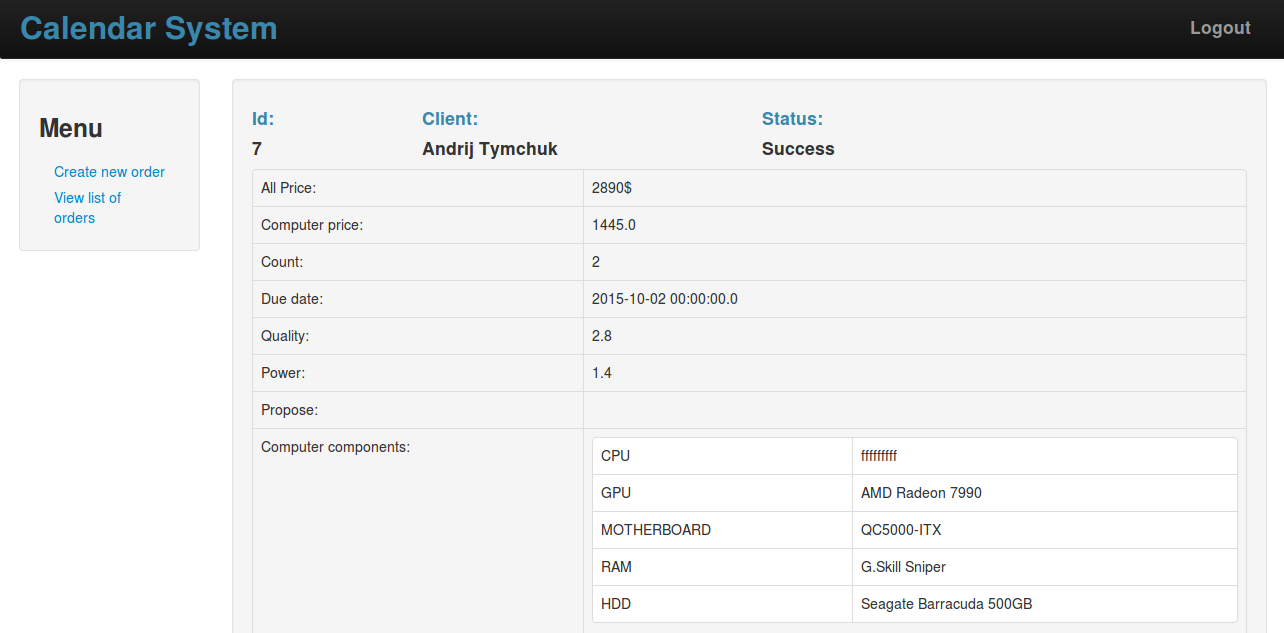


Рисунок 5.1.4 – Екранна копія сторінки перегляду інформації про замовлення

Для створення нового замовлення необхідно перейти за посиланням яке знаходиться у боковому меню та заповнити форму.

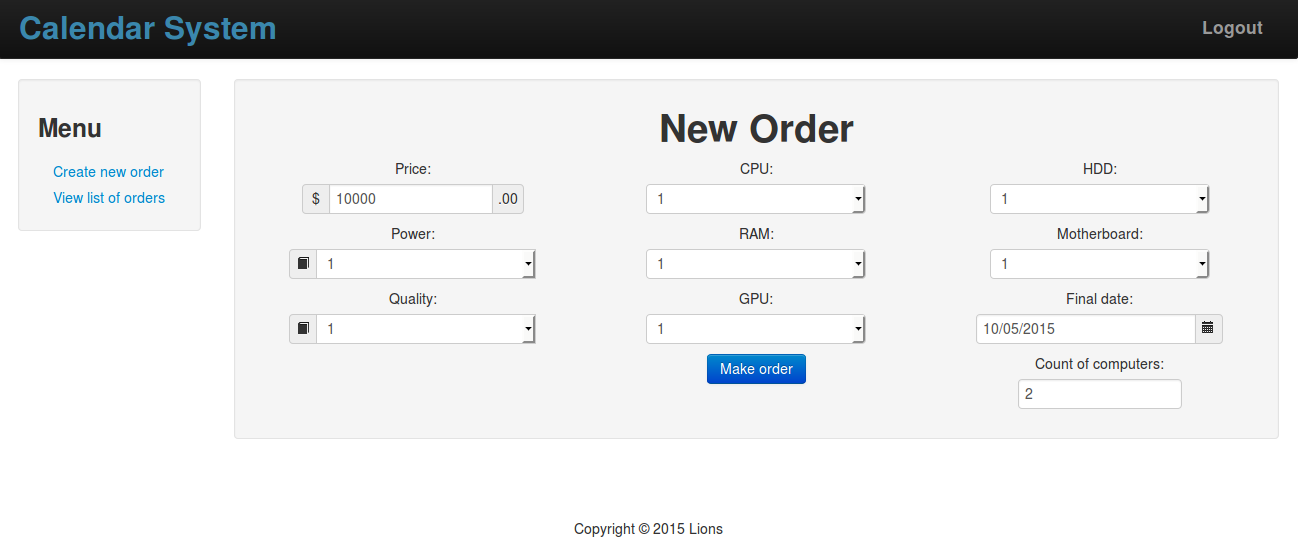


Рисунок 5.1.5 – Екранна копія процесу подачі заявки

Після натиснення кнопки «Надіслати», ми чекаємо на відповідь від планувальника, а заявка зявляється у списку заявок. Це продемонстровано на рисунку 5.6.

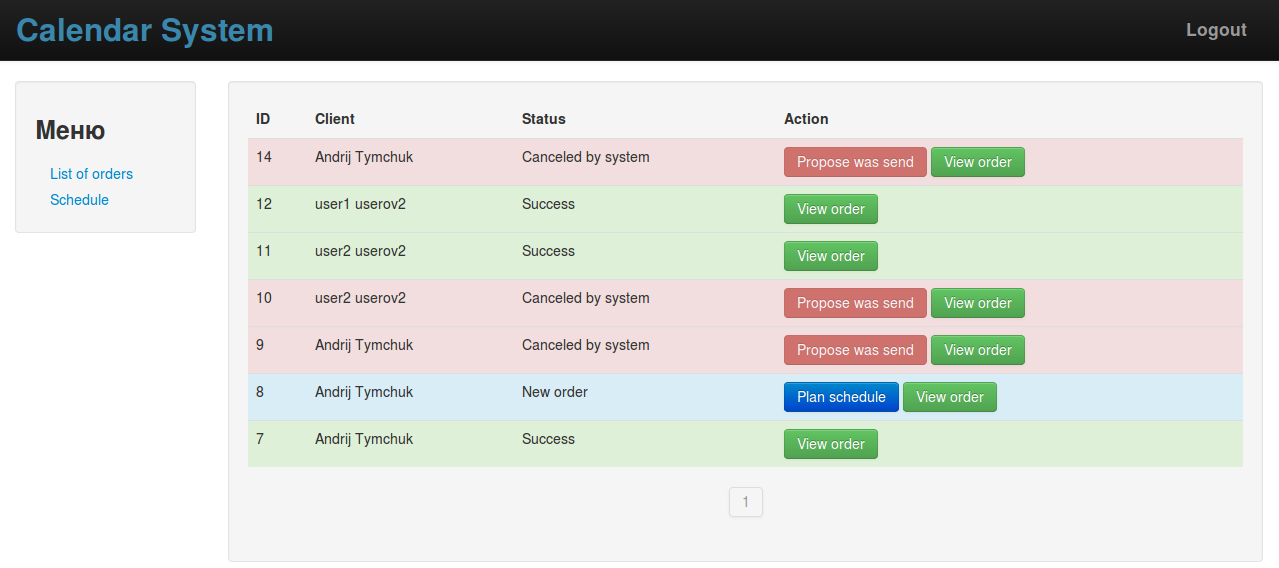


Рисунок 5.1.6 – Список замовлень планувальника

Коли планувальник спланує замовлення, то його статус у системі змінитьсяпро що користувач буде попереджений. Після виконання замовлення замовнику також прийде повідомлення на його електронну адресу яку він вказував при реєстрації.

## 5.2 Методика випробувань

Для тестування були розробленні тестові сценарії для наступних модулів:

* Реєстрація та авторизація
* Особистий кабінет користувача
* Особистий кабінет планувальника
* Особистий кабінет менеджера ресурсів

Приклади тестових сценаріїв:

* Реєстрація та авторизація

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Назва:** | Успішний тест на реєстрацію | | | |
| **Функція/Use Case:** | Реєстрація в систему | | | |
| **Дія:** | | **Очікуваний результат:** | | **Результат тесту:**   * пройдений * провалений * заблокований |
| **Передумова:** | |  | | |
| Користувач заходить на сторінку locallhost:/ | | Користувач може зареєструватися чи авторизуватися |  | |
| **Кроки тесту:** | |  | |  |
| Перейти на сторінку locallhost:/registration | | Відкрита сторінка реєстрації | |  |
| Користувач заповнює форму з наступними даними:  Ім’я : Олександр  Прізвище : Вереня  Пошта :vasya@gmail.com  Пароль : \*\*\*\*\*\* | | Користувач отримує повідомлення про успішну реєстрацію | |  |
| **Післяумова:** | |  | |  |
| Користувач повертається на locallhost:/ | | Тепер користувач зареєстрований | |  |
| **Назва:** | Успішний тест на авторизацію | | | |
| **Функція/Use Case:** | Реєстрація в систему | | | |
| **Дія:** | | **Очікуваний результат:** | | **Результат тесту:**   * пройдений * провалений * заблокований |
| **Передумова:** | |  | | |
| Користувач заходить на сторінку locallhost:/ | | Користувач може зареєструватися чи авторизуватися |  | |
| **Кроки тесту:** | |  | |  |
| Перейти на сторінку locallhost:/login | | Відкрита сторінка реєстрації | |  |
| Користувач заповнює форму з наступними даними:  Пошта :vasya@gmail.com  Пароль : \*\*\*\*\*\*\*\* | | Відкрита особистий кабінет ,що залежить від ролі користувача | |  |
| **Післяумова:** | |  | |  |
| Користувач повертається на locallhost:/ | | Тепер користувач зареєстрований | |  |

* Особистий кабінет користувача

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Назва:** | Успішний тест на перегляд замовлення | | | | |
| **Функція/Use Case:** | Перешлядання статусу замовлення | | | | |
| **Дія:** | | | **Очікуваний результат:** | | **Результат тесту:**   * пройдений * провалений * заблокований |
| **Передумова:** | | |  | | |
| Замовник комп’ютерів авторизується на сайті та знаходиться на сторінці localhost:/ | | | Замовник може передивится список замовлень та їх статус | |  |
| **Кроки тесту:** | | |  | |  |
| Перейти на сторінку localhost/orders | | | Відкрита сторінка яка містить список замовлень | |  |
|  | | |  | |  |
| **Післяумова:** | | |  | |  |
| Замовлення не зявляеться у планувальника | | | Замовлення не створено. | |  |
| **Назва:** | | Успішний тест на видалення замовлення | | | |
| **Функція/Use Case:** | | Видалення замовлення | | | |
| **Дія:** | | | **Очікуваний результат:** | | **Результат тесту:**   * пройдений * провалений * заблокований |
| **Передумова:** | | |  | | |
| Замовник комп’ютерів авторизується на сайті та знаходиться на сторінці localhost:/ | | | Замовник може видалити замовлення | |  |
| **Кроки тесту:** | | |  | |  |
| Перейти на сторінку localhost/orders | | | Відкрита сторінка яка містить список замовлень | |  |
| Натиснути на кнопку видалити навпроти потрібного замовлення | | | Статус замовлення змінюється на “Видалено” | |  |
| **Післяумова:** | | |  | |  |
| Замовлення видалено з черги замовлень | | | Замовлення не займає місця в черзі | |  |
| **Назва:** | | Успішний тест на друк замовлення | | | |
| **Функція/Use Case:** | | Друк замовлення | | | |
| **Дія:** | | | **Очікуваний результат:** | **Результат тесту:**   * пройдений * провалений * заблокований | |
| **Передумова:** | | |  | | |
| Замовник комп’ютерів авторизується на сайті та знаходиться на сторінці localhost:/ | | | Замовник може роздрукувати замовлення |  | |
| **Кроки тесту:** | | |  |  | |
| Перейти на сторінку localhost/orders | | | Відкрита сторінка яка містить список замовлень |  | |
| Натиснути на кнопку “роздрукувати“ навпроти потрібного замовлення | | | Почнеться друк замовлення |  | |
| **Післяумова:** | | |  |  | |
| Замовлення буде знаходитись на сторінці localhost/orders | | |  |  | |

* Особистий кабінет планувальника

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Назва:** | Планувальник переглядає список запланованих замовлень, замовлень, що виконуються, є виконаними, а також скасовані замовлення | | | |
| **Функція/Use Case:** | Обробка замовлень | | | |
| **Дія:** | | **Очікуваний результат:** | | **Результат тесту:**   * пройдений * провалений * заблокований |
| **Передумова:** | |  | | |
| Користувач авторизований як планувальник | | Користувачу доступні функції планувальника. |  | |
| **Кроки тесту:** | |  | |  |
| Планувальник перейшов за посиланням «Список замовлень» в боковому меню. | | Відкрита головна сторінка інтерфейсу планувальника, на якій зображено список(таблиця) виконаних/виконуваних/скасованих(системою/користувачем)/нових завдань. | |  |
| **Післяумова:** | |  | |  |
| Натиснути кнопку «Назад» в браузері | | Користувач опинився на попередній сторінці. | |  |
| **Назва:** | Планувальник переглядає календарний план | | | |
| **Функція/Use Case:** | Обробка замовлень | | | |
| **Дія:** | | **Очікуваний результат:** | | **Результат теста:**   * пройдений * провалений * заблокований |
| **Передумова:** | |  | | |
| Користувач авторизований як планувальник | | Користувачу доступні функції планувальника. |  | |
| **Кроки тесту:** | |  | |  |
| Користувач натиснув посилання «Переглянути календарний план» в боковому меню. | | Відкрита сторінка перегляду календарного плану з наявним у ній віджетом «Календар». | |  |
| **Післяумова:** | |  | |  |
| Натиснути кнопку «Назад» в браузері | | Користувач опинився на попередній сторінці. | |  |

Написання автоматизованих тестів виходить за рамки роботи, однак на базі тестових сценаріїв були створенні тести , що відносяться до таких основних груп:

* Інтеграційні — тести звязку системи з базою данних в важливих місцях обміну інформацією
* Модульні тести — тести на основних алгоритмах та обробках форм введення в систему інформації.

## 5.3 Приклади модульних тестів

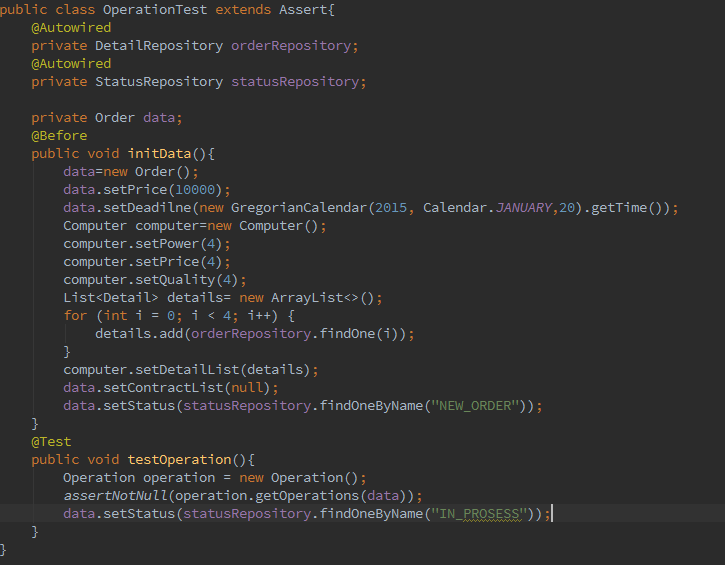


Рисунок 5.3.1 – приклад модульного тесту операційного компаратору



Рисунок 5.3.2 – модульний тест знаходження деталей замовлення для виконання плану

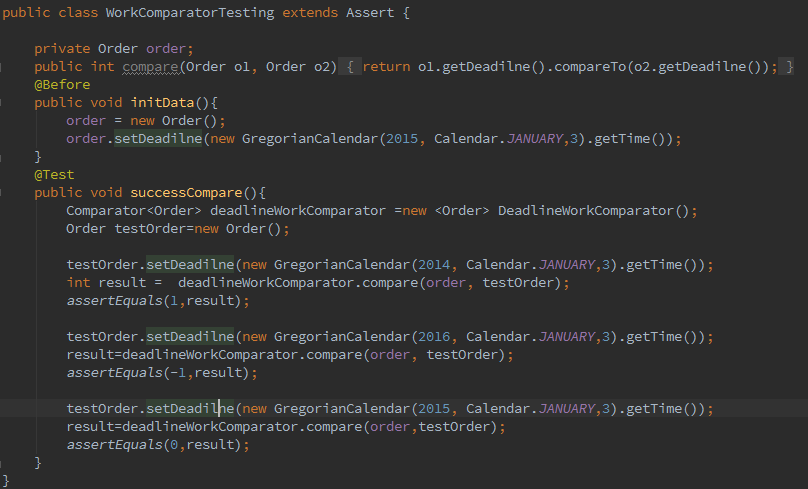


Рисунок 5.3.1 – приклад модульного компаратору по роботам

## 5.4 Приклади інтеграційних тестів

Загальна відношення покритття тестами до коду программи більше 30% . Такий низький показник можна пояснити використанням потужного Hibernate ORM та Spring JPA фреймворків ,що забеспечили безпечну роботу з базою данных. Крім цього використання JavaScript та регулярних виразів на стороні клієнта зменшили кількість тестових випадків через неможливість винекнення помилки.

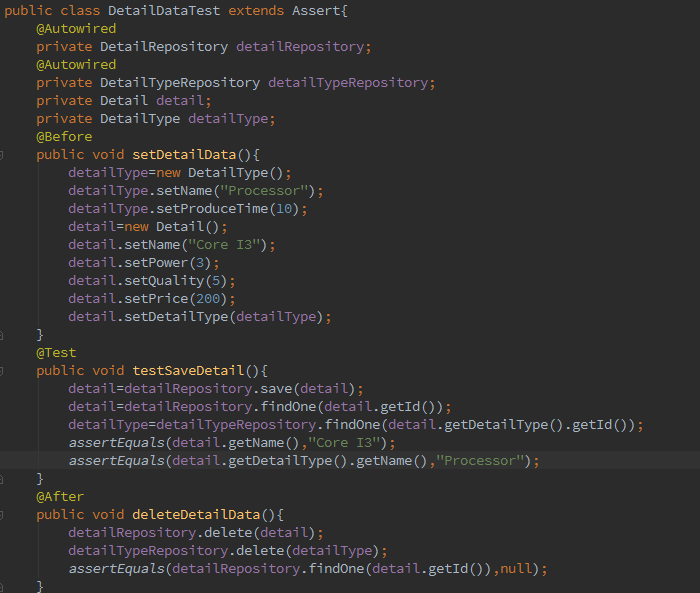


Рисунок 5.4.1 – приклад інтеграційного тесту CRUD операціями над деталями

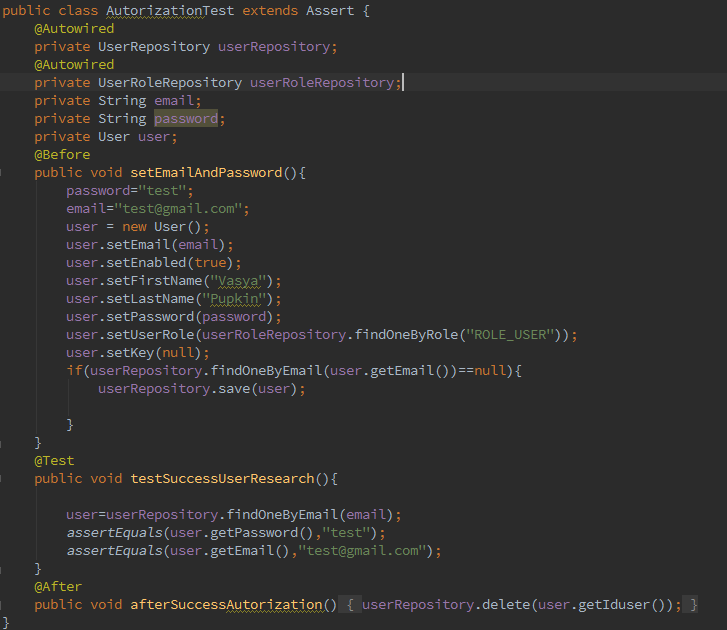


Рисунок 5.4.2 – приклад інтеграційного тесту авторизації

# 5 ВИСНОВОК

## 5.1 Результати роботи

По результатах проведених робіт було створено онлайн-систему для підприємств. Реалізовано майже повною мірою запланований функціонал. Тобто, повною мірою функціонують сторінки користувачів, планувальника та менеджера ресурсів. Коректно працюють алгоритми, що оптимізують планування нових замовлень та репланування старих. Коректно виконується надсилання повідомлень на електронну пошту, робота з інтерфейсами користувача, планувальника, менеджера ресурсів; додавання та скасування замовлень, робота з замовленнями. Також коректно проходить валідація користувача.

Основні проблеми з якими зіштовхнулася команда:

* мала кількість членів команди, на одну людину лягали завдання і менеджера, і тестувальника, і дизайнера;
* виникали проблеми при роботі з правами, а саме з виділенням унікальних прав на певний ресурс;
* проблеми були і з крос табличним пошуком по таблицях.

Загалом же команда задоволена результатом, тому що було отримано дуже велику кількість корисних навиків. Зокрема розробка на Java EE, робота в команді, розробка документації по проекту та ін.

Отже, роботу команди можна оцінювати позитивно, а проект досить якісним.

## 5.2 Виконана функціональність відповідно до ТЗ

Із задач, зазначених в ТЗ, були виконані наступні:

1. Задача створення замовлення та отримання результату:

* Отримання від замовника критеріїв по створенню комп’ютерів (кількість, термін, якість, ціна, потужність);
* Відправлення замовлення до планувальника;
* Отримання результату планування;

1. Задача складання календарного плану:

* Перевірка можливості виконання замовлення;
* Перегляд альтернативних варіантів замовлення;
* Додавання в календарний план нового замовлення;
* Видалення існуючих замовлень;
* Перегляд календарного плану;

1. Задача керування інформацією про ресурси:

* Зміна, видалення, додавання інформації про наявних майстрів;
* Зміна, видалення, додавання інформації про деталі;

1. Задача аутентифікації в системі:

* Реєстрація замовників, планувальників та замовників ресурсів;
* Вхід в систему з правами відповідно до вищенаведених ролей;

## 5.3 Невиконана функціональність відповідно до ТЗ

Задачі, які не було виконано в рамках розрахункової роботи:

* Друк детальної інформації про замовлення;

## 5.4 Перспективи розвитку програми

В майбутньому планується реалізувати ширший функціонал, розробити яскравіший дизайн, зробити можливість більш гнучкого керування процесами виробництва.

# Додаток А

## Календарний план

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ з/п** | **Назва етапів виконання розрахунково-графічної роботи** | **Строк виконання етапів проекту** | **Виконавець** | **Результат** |
|  | *Вивчення рекомендованої літератури* | *28.02.2014* | *Тимчук А.О.* |  |
|  | *Аналіз існуючих методів розв’язання задачі* | *28.02.2014* | *Кмець М.І.* |  |
|  | *Постановка та формалізація задачі* | *14.03.2014* | *Сидоров М.О.*  *Вереня О.І.* |  |
|  | *Розробка інформаційного забезпечення* | *10.05.2014* | *Тимчук А.О.* |  |
|  | *Модуль реєстрації, авторизації та верифікації користувачів* | *13.05.2014* | *Кмець М.І.* |  |
|  | *Модуль створення та керування замовленнями користувачів* | *13.05.2014* | *Сидоров М.О.* |  |
|  | *Модуль складання розкладів* | *13.05.2014* | *Тимчук А.О.* |  |
|  | *Модуль керування ресурсами* | *13.05.2014* | *Вереня О.І.* |  |
|  | *Налагодження програми* | *13.05.2014* | *Кмець М.І* |  |
|  | *Оформлення роботи* | *17.05.2014* | *Сидоров М.О.*  *Вереня О.І..* |  |
|  | *Подання РГР на захист* | *20.05.2014* | *Шевченко Н.В. Ткаченко В.Ю. Кальницький Р.І.* |  |

**Студент** \_\_\_\_\_\_\_\_\_\_Вереня О.І.

(підпис)

**Студент** \_\_\_\_\_\_\_\_\_\_Сидоров М.О.

(підпис)

**Студент** \_\_\_\_\_\_\_\_\_\_Кмець М.І.

(підпис)

**Студент** \_\_\_\_\_\_\_\_\_\_Тимчук А.А.

(підпис)

**Керівник роботи \_\_\_\_\_\_\_\_\_\_\_\_** Сперкач М.О