Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний  інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації

і управління

Звіт

з лабораторної роботи № 2 з дисципліни

«Компоненти інженерії програмного забезпечення»

«Дослідження процесу розробки програмного забезпечення.  Специфікування вимог»

Виконали студенти ІП-01

Паршиков Анатолій Олегович

Пасальський Олександр Русланович  
Шпилька Владислав Сергійович

Перевірив

Київ 2021

**Лабораторна робота 2**

**Дослідження процесу розробки програмного забезпечення.  Специфікування вимог.**

**Мета лабораторної роботи** – отримати навички специфікування вимог до програмного забезпечення.

**АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

**Інформаційна система житлового агентства** – це система яка дозволяє наймачам орендувати квартиру, а власникам здати свою нерухомість, вона містить базу даних з усіма потрібними даними про своїх клієнтів і списки вільних квартир з усіма характеристиками. Клієнти системи взаємодіють з нею через спеціальне програмне забезпечення.

Обслуговування клієнта системою починається з реєстрації, створення персонального облікового запису, вказавши логін, пароль, повне ім’я, дату народження, контактний телефон. Після реєстрації відбувається вхід в систему вказавши логін і пароль. Далі система пропонує клієнту дві дії розмістити оголошення про здачу житла або знайти житло.

Якщо клієнт обирає розмістити оголошення про здачу житла, він повинен вказати наступні дані: ціна (грн. в місяць), кількість кімнат, площа (метри квадратні), адрес (місто, вулиця, номер будинку, номер квартири), особливості, фотографії помешкання. Після всіх вказаних даних клієнт нажимає клавішу готово. Система опрацьовує його оголошення, зберігаючи його в своїй базі даних.

Якщо клієнт обирає знайти житло він повинен вказати наступні дані, характеристики підходящого йому житла: орієнтовну ціну, кількість кімнат, площу, адресу. Після вказаних даних клієнт нажимає пошук. Система опрацьовує його запит і надсилає список підходящих по його критерієм помешкань (список може бути порожнім). Клієнт може переглянути запропоновані оголошення, вибрати найбільш підходящий варіант і зателефонувати власнику, якщо ніщо йому не підходить змінити параметри пошуку.

**Глосарій**

**Користувач -** людина яка використовує нашу інформаційну систему житлового агентства (наймач, власник житла).

**Наймач**– це особа, якабажає орендуватижитло у власника житла для проживання в ньому.

**Власник житла** – це особа, у володінні якої є житло і вона бажає здавати його в оренду наймачу.

**Оренда** — це підтверджене договором строкове платне користування майном, необхідним орендареві для здійснення підприємницької або іншої діяльності.

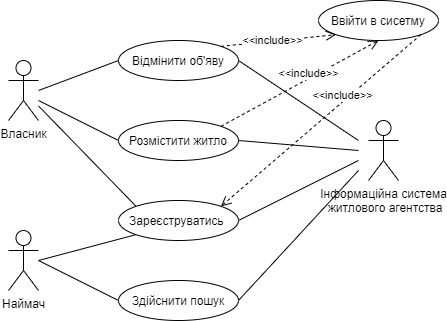
**Агентство з нерухомості** — це спеціалізована комерційна організація, яка здійснює посередницькі послуги між власником і наймачем.

**Персональний обліковий запис** - запис, що містить набір відомостей, які користувач передає будь-якої комп'ютерній системі.

Предметна область може бути представлена у вигляді описів користувача:

* Як користувач я хочу, щоб веб-сторінки завантажувались протягом 5 секунд, щоб я міг швидко працювати.
* Як наймач я хочу, щоб всі оголошення квартир були з фотографіями, щоб я міг точно зрозуміти чи це житло мені підходить.
* Як власник я хочу, щоб мої оголошення мали більшу активність, щоб мою квартиру подивилося більше людей.

**ДІАГРАМА ПРЕЦЕДЕНТІВ**



**СПЕЦИФІКАЦІЯ УМОВ**

1.

|  |  |
| --- | --- |
| Прецедент | Реєстрація |
| Description | Створення персонального облікового запису |
| Actor | Користувач(Власник, Наймач) |
| Pre-Conditions | У користувача ще немає облікових записів |
| Main Flow | 1. Користувач визиває функцію реєстрації 2. Система дозволяє ввести персональну інформацію, яка потрібна для реєстрації 3. Користувач заповнює всю необхідну інформацію 4. Система перевіряє коректність інформації 5. Система створює запис 6. Система повідомляє про успішність створення нового облікового запису. |
| AlternativeFlow | 1. Користувач відмовляється від заповнення інформації 2. Система відображає попередження 3. Якщо користувач підтверджує свій вибір   а.) Система завершує основний сценарій   1. Якщо користувач не підтверджує свій вибір   а.) Система переходить до кроку 3 основного сценарію.   1. Інформація введена не вірно 2. Система повідомлю про помилки 3. Система переходить до кроку 3 основного сценарію. |
| Post-Conditions | 1. Обліковий запис створений 2. Користувачу прийшло відповідне повідомлення |

**Функціональні вимоги:**

Ф1. Система повинна перевіряти дійсність введеної інформації

Ф2. Система повинна додавати інформацію про користувача до бази даних, для подальшого використання

Ф3. Система повинна присилати повідомлення про успішну реєстрацію на електронну пошту користувача

**Нефункціональні вимоги:**

НФ1. Система повинна бути написана на мові С++

НФ2. Система повинна перевіряти дійсність інформація протягом 5 секунд.

НФ3. Система повинна додавати інформацію про користувача протягом 5 секунд.

НФ4. Система повинна відправити відповідне повідомлення протягом однієї хвилини.

2.

|  |  |
| --- | --- |
| Прецедент | Вхід в систему |
| Description | Вхід у свій обліковий засіб |
| Actor | Користувач(Власник, Наймач) |
| Pre-Conditions | У користувача є обліковий запис |
| Main Flow | 1. Користувач визиває функцію входу в систему 2. Система дозволяє ввести персональну інформацію (логін і пароль), яка потрібна для входу в обліковий запис 3. Користувач заповнює всю необхідну інформацію 4. Система перевіряє коректність інформації 5. Система повідомляє про успішність входу в обліковий запис. |
| AlternativeFlow | 1. Користувач відмовляється від заповнення інформації 2. Система відображає попередження 3. Якщо користувач підтверджує свій вибір   а.) Система завершує основний сценарій   1. Якщо користувач не підтверджує свій вибір   а.) Система переходить до кроку 3 основного сценарію.   1. Інформація введена не вірно 2. Система повідомлю про помилки 3. Система переходить до кроку 3 основного сценарію. |
| Post-Conditions | 1. Користувач ввійшов в обліковий запис |

**Функціональні вимоги:**

Ф1. Система повинна перевіряти дійсність введеної інформації

**Нефункціональні вимоги:**

НФ1. Система повинна бути написана на мові С++

НФ2. Система повинна перевіряти дійсність інформація протягом 5 секунд.

**3.**

|  |  |
| --- | --- |
| Прецедент | Розмістити житло |
| Description | Розміщення об’яви про вільне житло |
| Actor | Власник(Користувач) |
| Pre-Conditions | 1. Власник ввійшов в обліковий запис, який зареєстрований в системі 2. Житло належить власнику 3. В цьому житлі ніхто не живе |
| Main Flow | 1. Власник визиває функцію Розмістити житло 2. Система дозволяє ввести інформацію про будинок 3. Власник заповнює всю необхідну інформацію 4. Система перевіряє коректність інформації 5. Система створює нову об’яву і розміщує її для пошуку 6. Система повідомляє про успішність створення нової об‘яви |
| AlternativeFlow | 1. Власник відмовляється від заповнення інформації 2. Система відображає попередження 3. Якщо власник підтверджує свій вибір   а.) Система завершує основний сценарій   1. Якщо власник не підтверджує свій вибір   а.) Система переходить до кроку 3 основного сценарію.   1. Інформація введена не вірно 2. Система повідомлю про помилки 3. Система переходить до кроку 3 основного сценарію. |
| Post-Conditions | 1. Об’ява створена 2. Власнику прийшло відповідне повідомлення |

**Функціональні вимоги:**

Ф1. Система повинна перевіряти дійсність введеної інформації

Ф2. Система повинна додавати інформацію про житло та об’яву до бази даних

Ф3. Система повинна зробити об’яву доступною для пошуку

Ф4. Система повинна присилати повідомлення про успішне розміщення на електронну пошту власника

**Нефункціональні вимоги:**

НФ1. Система повинна бути написана на мові С++

НФ2. Система повинна перевіряти дійсність інформація протягом 5 секунд.

НФ3. Система повинна додавати інформацію про житло протягом 5 секунд.

НФ4. Система повинна відправити відповідне повідомлення протягом однієї хвилини.

НФ5. Система повинна розмістити об’яву протягом однієї години

**4.**

|  |  |
| --- | --- |
| Прецедент | Відмінити об’яву |
| Description | Власник хоче відмінити об’яву |
| Actor | Власник (Користувач) |
| Pre-Conditions | 1. Власник ввійшов обліковий запис, який зареєстрований в системі 2. У власника є активна об’ява про оренду житла 3. Житло належить власнику 4. Хтось вже проживає цьому будинку або власник більше не хоче здавати це житло |
| Main Flow | 1. Власник визиває функцію Відмінити об’яву 2. Система відміняє об’яву, тобто вона недоступна в пошуку 3. Система повідомляє про успішність операція |
| AlternativeFlow | 1. Власник відмовляється від операції 2. Система відображає попередження 3. Якщо власник підтверджує свій вибір   а.) Система завершує основний сценарій   1. Якщо власник не підтверджує свій вибір   а.) Система переходить до кроку 1 основного сценарію. |
| Post-Conditions | 1. Об’ява більше недоступна 2. Власнику прийшло відповідне повідомлення |

**Функціональні вимоги:**

Ф1. Система повинна видалити інформацію про житло з бази даних

Ф2. Система повинна видалити об’яву з пошуку

**Нефункціональні вимоги:**

НФ1. Система повинна бути написана на мові С++

НФ2. Система повинна видалити інформацію про житло протягом 5 секунд.

НФ4. Система повинна відправити відповідне повідомлення протягом однієї хвилини.

НФ5. Система повинна зняти об’яву протягом однієї години

**5.**

|  |  |
| --- | --- |
| Прецедент | Здійснити пошук |
| Description | Здійснення пошуку вільної квартири згідно з заданими фільтрами |
| Actor | Наймач (Користувач) |
| Pre-Conditions |  |
| Main Flow | 1. Наймач визиває функцію Здійснити пошук 2. Система дозволяє встановити фільтр пошуку 3. Система знаходить всі вільні квартири згідно з заданими фільтрами 4. Система виводить на екран знайдені квартири |
| AlternativeFlow | 1. Наймач відмовляється від заповнення фільтрів 2. Система встановлює, що фільтрів немає 3. Система переходить до кроку 3 основного сценарію. 4. Користувач не знайшов підходящого житла 5. Система переходить до кроку 2 основного сценарію |
| Post-Conditions | 1. На екрані зображено вільні квартири згідно з пошуком, та показано інформацію про них. |

**Функціональні вимоги:**

Ф1. Система повинна обробити фільтри пошуку

Ф2. Система повинна шукати об’яви згідно з пошуком

**Нефункціональні вимоги:**

НФ1. Система повинна бути написана на мові С++

НФ2. Система повинна видавати результат протягом 30 секунд

**МОДЕЛЬ ЖИТТЄВОГО ЦИКЛУ**

Для проектування даної системи було обрано Водоспадну модель життєвого циклу. Ми обрали цю модель, тому що вона має зрозумілу і просту структуру процесу розробки. По-друге зручна звітність, дозволяє легко відстежити ресурси і витрачений час завдяки суворій етапності процесу розробки та детальній документації проекту на кожному кроці. По-третє дана модель є стабільною, тобто задачі, які стоять перед продуктом, ясні команді з самого початку розробки, і залишаються незмінними протягом усього процесу. Також ця модель не потребує високого рівня підготовленості команди розробників, що є плюсом.

Ця модель складається з таких пунктів:

1. **Системні і програмні вимоги.** Для нашого проекту вимогами є побудова системи, яка дозволяє виконувати наступні дії: квартиронаймачам дібрати і зняти житло, а власникам - запропонувати і здати.
2. **Проектування.** Проведення попереднього аналізу, визначення функцій системи, а також побудова діаграми прецедентів.
3. **Реалізація.** В цій частині безпосередньо пишеться код програми**,** йде інтеграція програмного забезпечення.
4. **Тестування.** Перевірка фінального продукту. У разі виявлення помилок, слідує їх виправлення.
5. **Верифікація.** Це процес посвідчення, що наша система відповідає всім висунутим вимогам.
6. **Експлуатація і супровід.** Випуск продукту та його підтримка після вводу до експлуатації.

**Висновок:** виконавши лабораторну роботу, миотримати навички специфікування вимог до програмного забезпечення. Навчилася організовувати роздільну роботу між учасниками проекту. Розробили модель інформаційної системи житлового агентства. Провели попередній аналіз предметної області програмного забезпечення, що проектуємо. Побудували діаграму прецедентів на основі попереднього аналізу. Специфікували вимоги і обрали модель життєвого циклу ПЗ підготували захист у вигляді мітингу з замовником.