**Практичне завдання № 1.**

**Визначення бізнес-вимог і вимог користувача**

1.1 Визначення бізнес-вимог

1.1.1 Опис предметної області автоматизації

Вся інформація, яка нас оточує, надзвичайно велика. С часом ця інформація збільшує свої розміри. Тому має місце проблема обробки інформації. Деякі організації зберігають інформацію в звичайних папках для документів, але більша частина все ж використовує комп’ютеризований спосіб. І вже сьогодні без баз даних неможливо представити роботу не однієї програми, тому зараз набагато доцільніше зберігати інформацію в електронному вигляді. Вартість зберігання інформації в файлах дешевше, ніж на папері. Бази даних допомагають зберігати, структурувати інформацію і витягувати оптимальним для користувача чином.

Використання клієнт/серверних технологій дозволяють зберегти значні кошти, а головне і час для отримання необхідної інформації, а також спрощують доступ і ведення.

Будь-яка система, навіть невелика, потребує постійного контролю. Незалежно від якості її налаштування та якості апаратного забезпечення в її складі не можна покладатися на можливість її самостійної стабільної роботи. Навіть наявність відповідальної особи, не може гарантувати стабільність її роботи, особливо у випадку значної складності або масштабності. Оскільки людина не може постійно шукати проблему тої чи іншої проблеми, є виправданим застосування інформаційних систем які дозволяють виявляти неполадки на ранніх етапах та отримувати про них якомога детальнішу інформацію. При цьому, обробка та аналіз отриманої інформації покладається на відповідальну особу.

Одним із способів автоматизації пошуку помилки є відтворення роботи людини-експерта. Це стало можливим з використанням експертних систем, які оперуються на базу знань про можливі неполадки. Але корисність систем на основі експертних знань, напряму залежить від якості баз знань.

Головна мета цієї дипломної роботи є поліпшення безпеки роботи з електрообладнанням. Дана система, моніторить вимоги з електробезпеки і приймає рішення про допустимість проведення робіт. Рішення програма буде приймати виходячи з правил, які будуть описані на основі експертної системи.

Експертна система акумулює знання технічних фахівців про виявлення причин аномальної роботи мереж і можливі способи приведення мережі в працездатний стан. Найпростішим варіантом експертної системи є контекстно-залежна система допомоги. Більш складні експертні системи, бази знань, що володіють елементами штучного інтелекту.

 Об’єктом автоматизації даної роботи є поліпшення пошуку інформації з електробезпеки. Суб’єктами є персонал який відповідає за електробезпеку на підприємстві, або будь-який користувач якому необхідно знайти данні та отримати рекомендації з електробезпеки.

Рисунок 1 – Ментальна карта

1.1.2 Визначення проблем, що виникають у суб'єктів автоматизації при

взаємодії з об'єктом

Мабуть найбільш проблемними функціями системи є **Пошук інформації** і **Консультація користувача**. Вони обидва призводять до виникнення проблем у суб'єктів автоматизації при взаємодії, але ці проблеми незначні. **Пошук інформації** як і **Консультація користувача** можуть виконуватися дуже довго. Наприклад: Користувач запустив консультацію, при виконанні користувачеві будуть задані різного роду питання, що стосуються електробезпеки. Залежно від відповіді, виконання консультації буде варіюватися від 30 сек. до 10+ хв.

1.1.3 Проблемний аналіз існуючих програмних продуктів

Було вирішено виділити наступні інструменті розробки баз знань:

* JESS
* OpenL Tablets
* JBoss Drools
* IBM JRules

СИСТЕМА Jess

Jess – движок правил, написаний мовою Java та призначений для використання у програмних продуктах, написаних також мовою Java. Розроблений Ернестом Фрідманом-Хілом у національній лабораторії Сандія (Sandia National Labs, США) у 1995 році.

Jess був створений як реалізація CLIPS мовою програмування Java. У подальшому при розробці були реалізовані нові можливості, але Jess досі зберігає зворотну сумісність із CLIPS. Основі принципи його роботи лежать у порівнянні набору правил з набором фактів. Порівняння фактів та правил здійснюється за алгоритмом Rete. База знань при використанні Jess задається у текстовому вигляді. Код Jess не є відкритим. Для використання Jess у комерційних продуктах потрібно придбати ліцензію, для учбових закладів та лабораторій ліцензія є безкоштовною.

СИСТЕМА OpenL Tablets

OpenL Tablets – система управління бізнес-правилами на основі таблиць прийняття рішень. Ця система розробляється компанією Exigen Services та написана мовою Java.

Дана система підтримує деякі додаткові типи подання знань. Окрім таблиць прийняття рішень підтримуються також дерева.

OpenL Tablets не надає жодних засобів візуалізації знань, засобів для перевірки якості знань, обмежені перевіркою синтаксичних помилок та типізації.

СИСТЕМА JBoss Drools

JBoss Drools – система управління бізнес-правилами з відкритим вихідним кодом розробляється компанією Red Hat і розповсюджується за ліцензією ASL 2 (Apache License 2.0.)

Даний продукт передбачає подання знань у вигляді фактів та правил, як і в Jess виконання відбуваються за допомогою алгоритму Rete. Правила описуються за допомогою спеціального діалекту мови XML.

СИСТЕМА IBM JRules

IBM JRules – система управління бізнес-правилами розробляється французькою компанією ILOG. Дана система підтримує подання знань як у вигляді фактів та правил, так і у вигляді таблиць прийняття рішень. Цей продукт має закритий вихідний код, для використання потрібна ліцензія.

В таблиці 1 наведена порівняльна характеристика розглянутих засобів розробки баз знань.

Таблиця 1 Порівняльна характеристика інструментів розробки баз знань

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Jess | OpenL Tablets | JBoss  Drools | IBM  JRules |
| Форма подання знань | Правила та факти | Таблиця рішень | Правила та факти | Правила та факти |
| Відкритий вихідний код | Ні | Ні | Так | Ні |
| Інструменти редагування бази знань | Середовище Eclipse, текстовий редактор | Microsoft Word, Excel | Середовище Eclipse, текстовий редактор | Середовище Eclipse,  Microsoft Word, Excel |
| Контроль якості знань | Ні | Обмеж. | Обмеж. | Обмеж. |
| Візуалізація знань | Ні | Ні | Так | Так |
| Перевірка суперечливості знань | Ні | Так | Так | Так |
| Перевірка над достатності знань | Ні | Ні | Ні | Ні |
| Перевірка повноти знань | Ні | Ні | Ні | Так |

1.1.4 Визначення цілей

Розробка полягає у створенні клієнт/серверної системи, яка поліпшує безпеку роботи з електрообладнанням. Дана система, розглядає вимоги з електробезпеки і приймає рішення про допустимість проведення робіт.

Програмний продукт повинен мати функції для роботи з базою правил, виконувати консультацію користувача, вміти розмежувати доступ за допомогою авторизації та реєстрації.

Система прийняття рішень повинна підвищити безпеку шляхом надання рекомендацій користувачу. За допомогою цієї програми користувачам буде зрозуміло, чи можна допускати працівників до роботи.

Для створення бази знань необхідно провести добування знань на основі відповідей експерта.

1.1.5 Визначення унікального назви програмного продукту

Виходячи з того що програмний продукт спрямований на рішення проблем з електробезпеки, було вирішено назвати продукт ZEVS, як відомо Зевс в давньогрецькій міфології бог неба, грому і блискавок.

На рис. 2 зображено тестовий логотип програмного продукту.

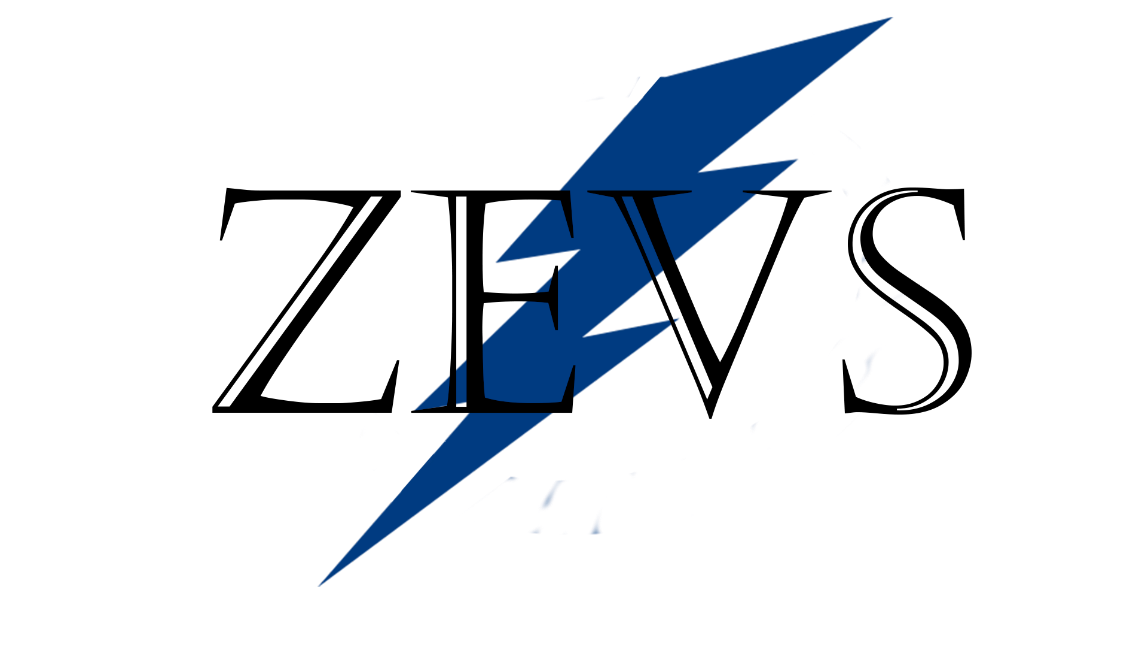


Рисунок 2 – Логотип програмного продукту

Що таке ZEVS ?

ZEVS - electrosecurity god (бог електробезпеки)

Програма ZEVS - це інформаційна система яка допомагає користувачу вирішувати проблеми з електробезпеки. Ця система допоможе людям які не настільки добре знають електробезпеку, допоможе знайти проблему с обладнанням в найкоротші строки. Також дана система стане в нагоді для автоматизації робочого місця, для своїх потреб з електробезпеки.

1.2 Визначення вимог користувача

1.2.1 Опис діаграми прецедентів

Користувач даної системи може виконувати наступні дії:

* Пошук інформації з електробезпеки
* Отримання консультації
* Додавання, видалення та редагування користувачів
* Додавання, видалення та редагування правил експертної системи
* Додавання, видалення та редагування інформаційної бази
* Отримання статистики
* Авторизація
* Реєстрація

На рис. 3 представлені формалізовані вимоги до програмної системи, що розробляється у вигляді діаграми прецедентів.

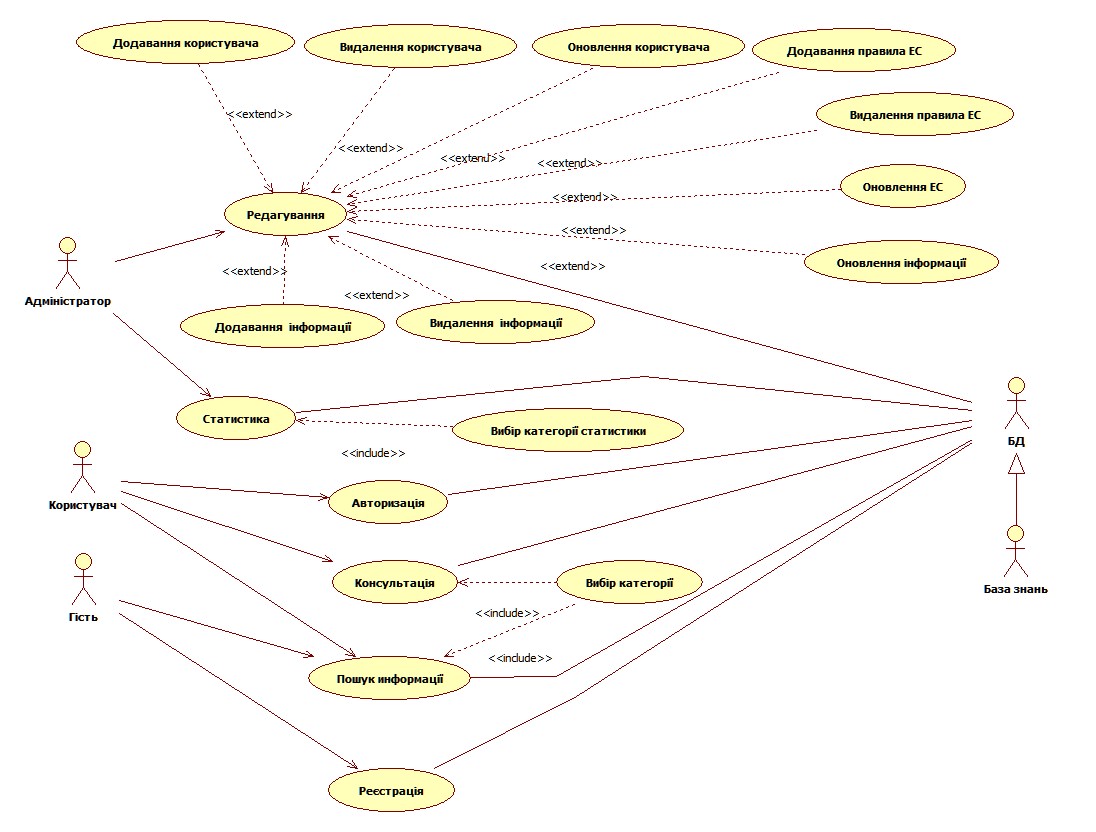


Рисунок 3 - Діаграма прецедентів

Адміністратор – актор який раніше зареєстрований в системі і має особливі права доступу, він відповідає за супровід системи, в його арсеналі всі необхідні функція для роботи с даними.

Користувач – актор який вже раніше був зареєстрований в системі, він може шукати інформацію та отримувати консультацію.

Гість – актор який не зареєстрований в системі, його функціонал обмежений і складається лише з пошуку необхідної інформації.

Редагування – прецедент к якому має доступ лише Адміністратор, він відповідає за додавання, змінення та оновлення даних системи, дані дії Адміністратор може виконати в разі виявлення якихось недоліків.

Статистика – прецедент к якому має доступ лише Адміністратор, відповідає за відображення інформативно статистичних даних системи.

Консультація – у разі виникнення питань з електробезпеки, користувач може скористатися цією функцією, яка в ході інтерв'ю відповість на питання, що цікавлять користувача, і в результаті система допоможе вирішити якусь проблему.

Реєстрація – цей прецедент відповідає за реєстрацію нового користувача в системі.

Авторизація – прецедент який відповідає за вхід в систему раніше зареєстрованого користувача.

1.2.2 Опис сценаріїв використання прецедентів

1. **Прецедент «Реєстрація»**

**Актори:** Гість, база даних.

**Передумова:** Користувач має доступ до локальної або глобальної мережі, з'єднання стабільне без розривів. Сервер запущений і готовий до виконання поставлених завдань.

**Основний успішний сценарій:**

1. Гість запускає програму «ZEVS».
2. Система ZEVS відображає вікно (форму) з проханням авторизуватися або зареєструватися.
3. Гість вибирає пункт **«Зареєструватися»**
4. Система ZEVS відображає вікно **Реєстрації** з полями **Логін**, **Пароль, Прізвище, Ім’я**  та поле повторного вводу паролю.
5. Гість вводить бажаний **Логін**. Система аналізує введенні данні та підтверджує.
6. Гість вводить бажаний **Пароль.** Система аналізує введенні данні та підтверджує.
7. Гість вводить бажане **Прізвище** та **Ім’я**  **.** Система аналізує введенні данні та підтверджує.
8. Гість повторно вводить пароль для підтвердження. Система аналізує введенні данні та підтверджує.
9. Система підтверджує реєстрацію нового користувача.

**Альтернативний сценарій:**

5a. Гість ввів **Логін** який вже зайнятий другим користувачем. Повідомлення про помилку. Повернення до пункту 5, основного сценарію.

5b.Гість використовував неприпустимі символи при введенні **Логіна**. Повідомлення про помилку. Повернення до пункту 5, основного сценарію.

6a. Гість ввів дуже короткий **Пароль**. Повідомлення про помилку. Повернення до пункту 6, основного сценарію.

6b. Гість використовував неприпустимі символи при введенні **Паролю.** Повідомлення про помилку. Повернення до пункту 6, основного сценарію.

7a. Гість ввів неправильний **Пароль**. Повідомлення про помилку. Повернення до пункту 7, основного сценарію.

8a.Система не може додати нового користувача до бази. Повідомлення про помилку. Повернення до пункту 3, основного сценарію.

1. **Прецедент «Авторизація»**

**Актори:** Гість, база даних.

**Передумова:** Гість був раніше зареєстрований, має доступ до локальної або глобальної мережі.

**Основний успішний сценарій:**

1. Користувач запускає програму «ZEVS».
2. Система ZEVS відображає вікно (форму) з проханням авторизуватися або зареєструватися.
3. Гість вводить свій **Логін** та **Пароль**.
4. Система аналізує введенні данні та підтверджує вхід.

**Альтернативний сценарій:**

4a. Система знаходить помилку в введених даних. Повідомлення про помилку. Повернення до пункту 3, основного сценарію.

1. **Прецедент «Консультація»**

**Актори:** користувач, база даних.

**Передумова:** Користувач виконав вхід у систему та має доступ до локальної або глобальної мережі.

**Основний успішний сценарій:**

1. Користувач вибирає пункт меню **«Консультація»**
2. Система ZEVSвідображає вікно (форму) з проханням вибрати тему консультації зі списку.
3. Користувач вибирає потрібний йому варіант. Система аналізує та виводить інформацію.
4. Користувач вибирає потрібний пункт. Система аналізує та підтверджує вибір.
5. Система починає опитування користувача для вирішення проблеми.
6. Користувач відповідає на питання Системи. Система аналізує та підтверджує.
7. Система виводить кінцевій результат опитування.

**Альтернативний сценарій:**

6a.Користувач неправильно відповідає на питання системи. Повідомлення про помилку. Повернення до пункту 5, основного сценарію.

7a. Система не виводить кінцевій результат. Повідомлення про помилку. Повернення до пункту 5, основного сценарію.

1. **Прецедент «Редагування»**

**Актори:** Адміністратор, база даних.

**Передумова:** Адміністратор виконав вхід у систему, та має права доступу до бази даних, також доступ до локальної або глобальної мережі.

**Основний успішний сценарій:**

1. Адміністратор вибирає пункт меню **«Редагування правил»**
2. Система ZEVSвідображає вікно (форму) з переліком функцій для роботи з даними.
3. Користувач вибирає потрібний пункт. Система аналізує та підтверджує вибір.
4. Користувач вводить необхідні данні. Система аналізує та підтверджує.
5. Система виводить повідомлення про успішне редагування даних.

**Альтернативний сценарій:**

4a.Користувач вводить некоректні дані. Повідомлення про помилку. Повернення до пункту 3, основного сценарію.

5a.Система не відредагувала данні. Повідомлення про помилку. Повернення до пункту 3, основного сценарію.

1. **Прецедент «Статистика»**

**Актори:** Адміністратор, база даних.

**Передумова:** Адміністратор виконав вхід у систему, та має права доступу до бази даних, також доступ до локальної або глобальної мережі.

**Основний успішний сценарій:**

1. Адміністратор вибирає пункт меню **«Статистика»**
2. Система ZEVSвідображає вікно (форму) з переліком статистичних даних.
3. Адміністратор вибирає потрібний пункт. Система аналізує та підтверджує вибір.
4. Система виводить необхідні данні .

**Альтернативний сценарій:**

3a.Адміністратор вводить некоректні дані. Повідомлення про помилку. Повернення до пункту 3, основного сценарію.

4a.Система не виводить данні. Повідомлення про помилку. Повернення до пункту 3, основного сценарію.

1. **Прецедент «Пошук інформації»**

**Актори:** Користувач, база даних.

**Передумова:** Користувач виконав вхід у систему, та має права доступу, також доступ до локальної або глобальної мережі.

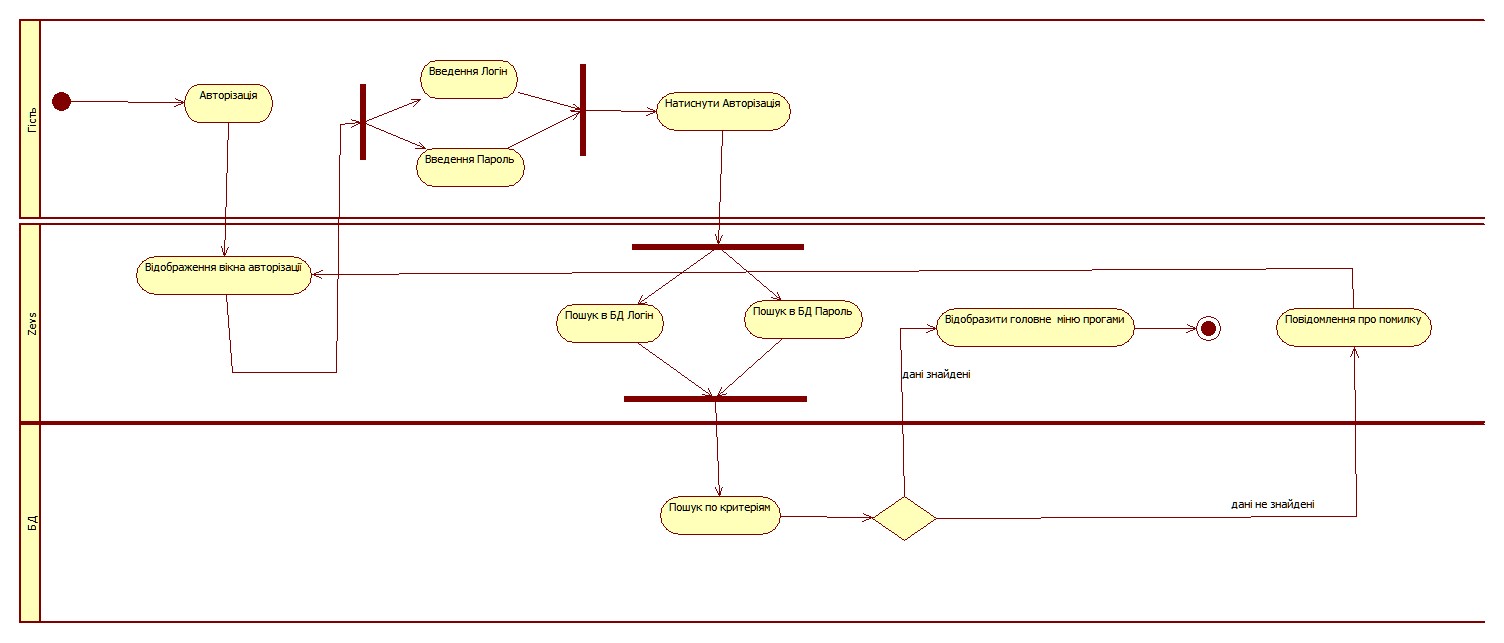
**Основний успішний сценарій:**

1. Користувач вибирає пункт меню **«Пошук інформації»**
2. Система ZEVSвідображає вікно (форму) з полем для вводу даних пошуку.
3. Користувач вводить потрібні данні. Система аналізує та підтверджує .
4. Система шукає та виводить необхідні данні .

**Альтернативний сценарій:**

3a.Адміністратор вводить некоректні дані. Повідомлення про помилку. Повернення до пункту 3, основного сценарію.

4a.Система не виводить данні. Повідомлення про помилку. Повернення до пункту 3, основного сценарію.

1.2.3 Специфікація призначених для користувача вимог

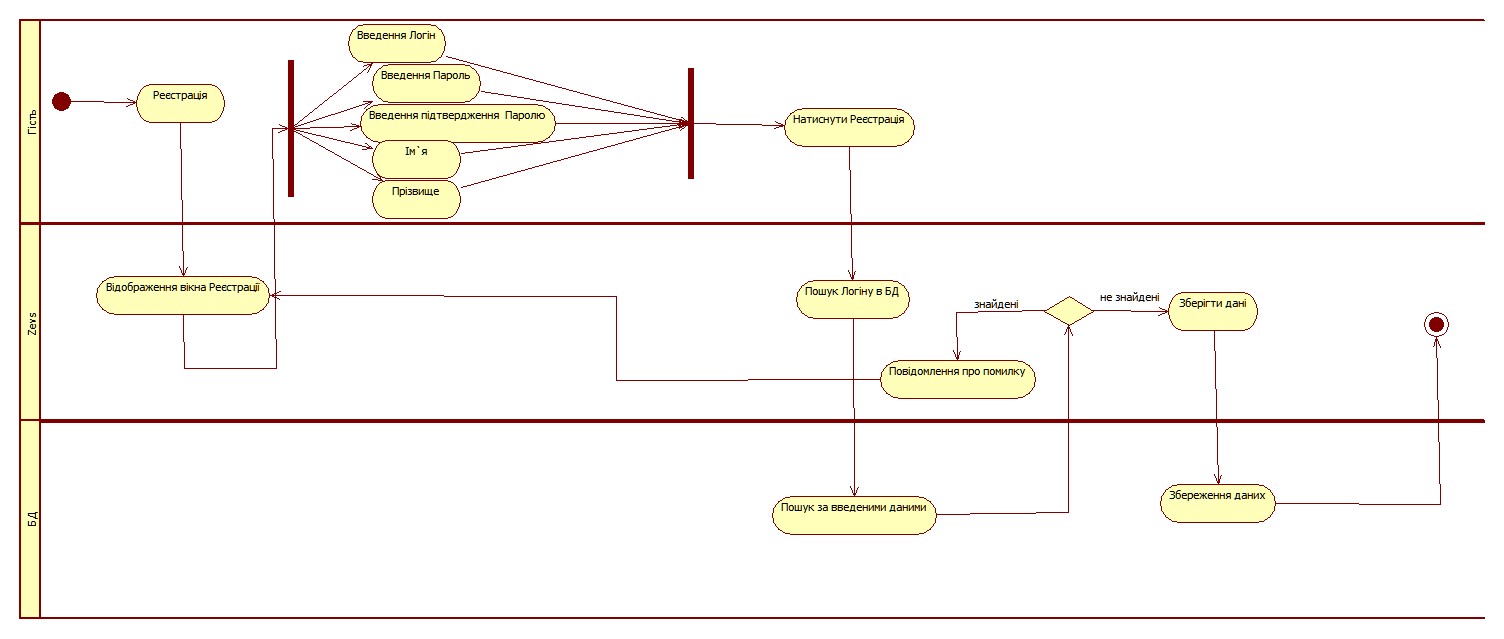
Рисунок 4 - Діаграма діяльності «Авторизація»

Рисунок 5 - Діаграма діяльності «Реєстрація»

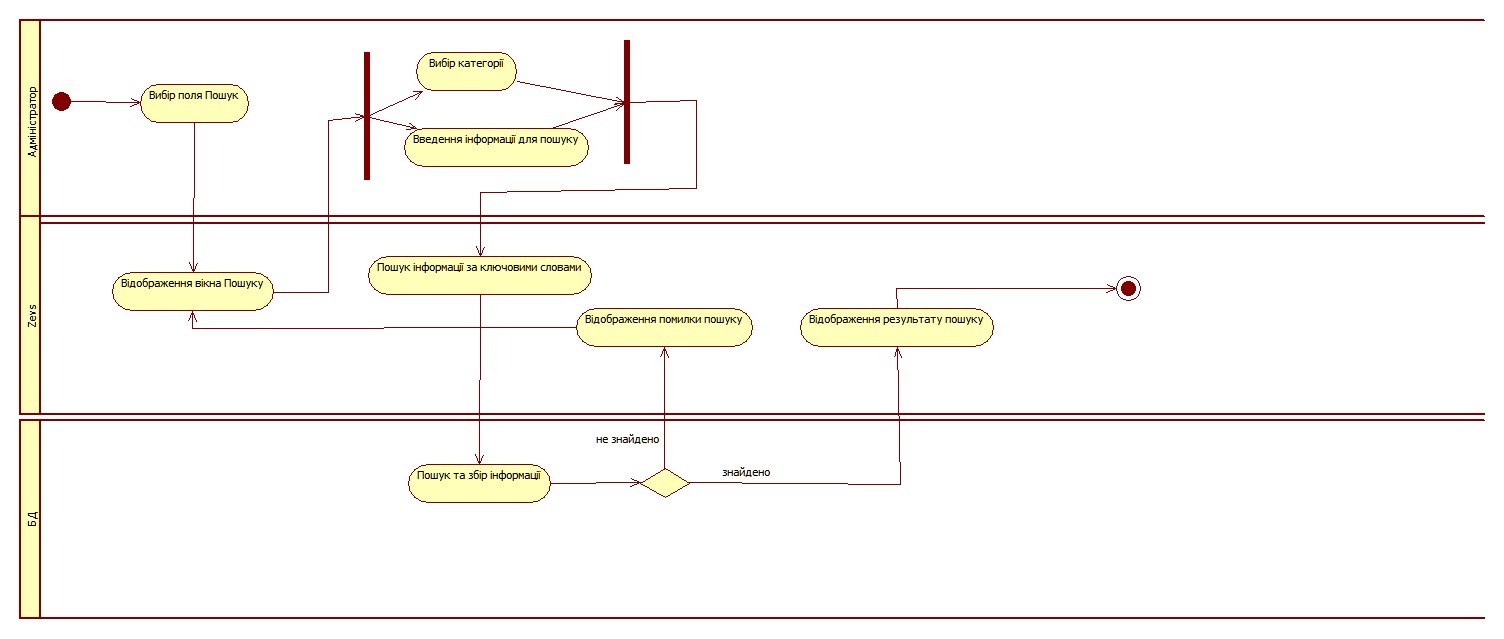
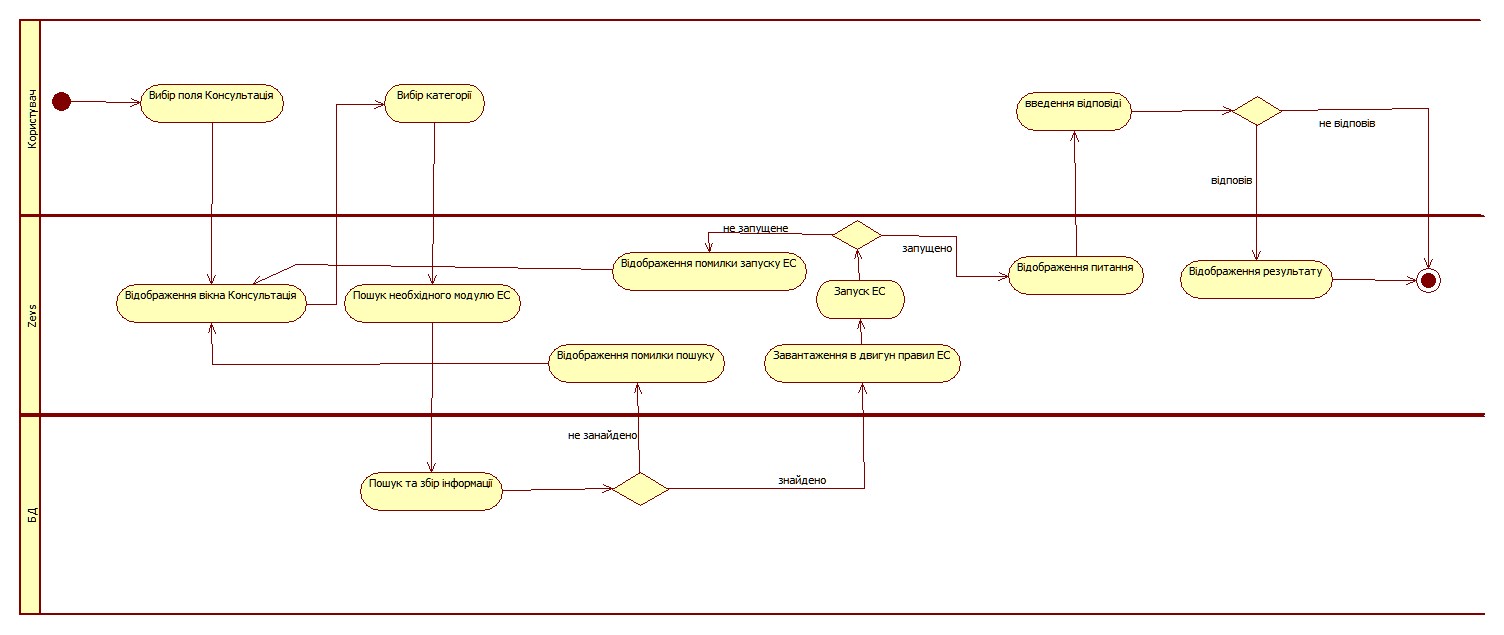
Рисунок 6 - Діаграма діяльності «Консультація»

Рисунок 7 - Діаграма діяльності «Пошук інформації»

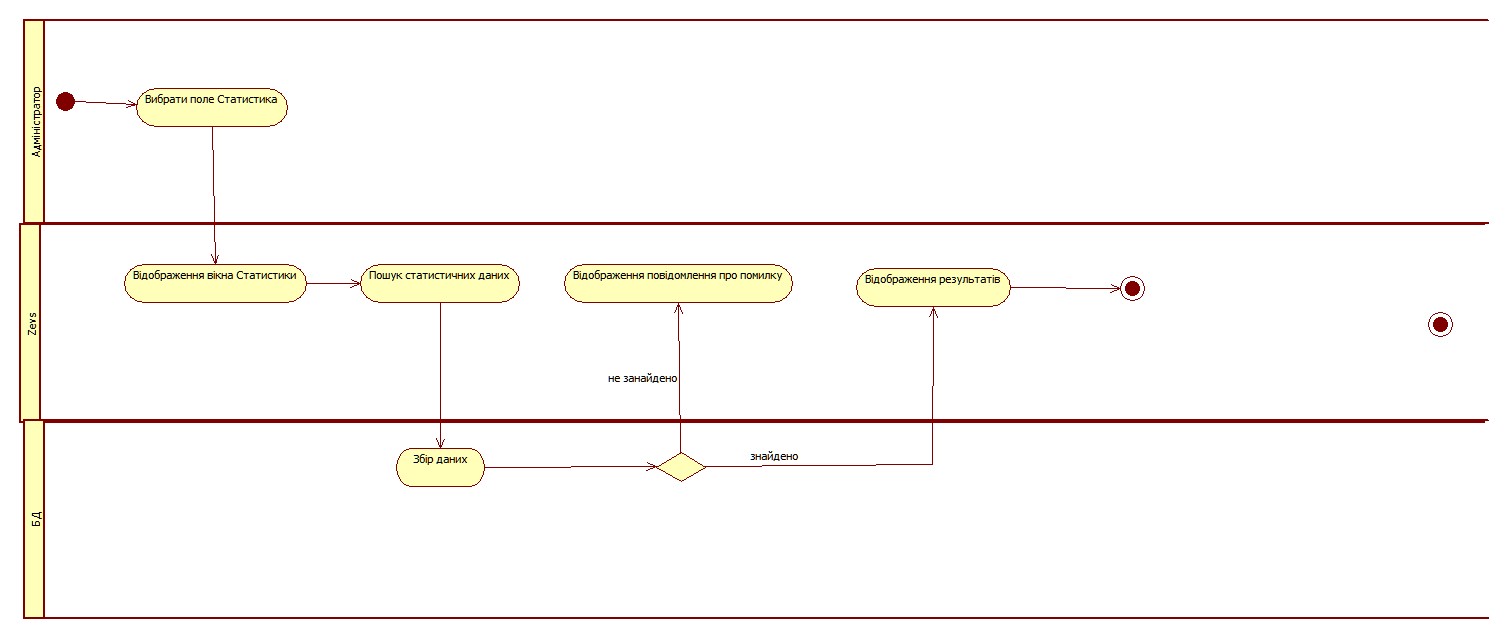
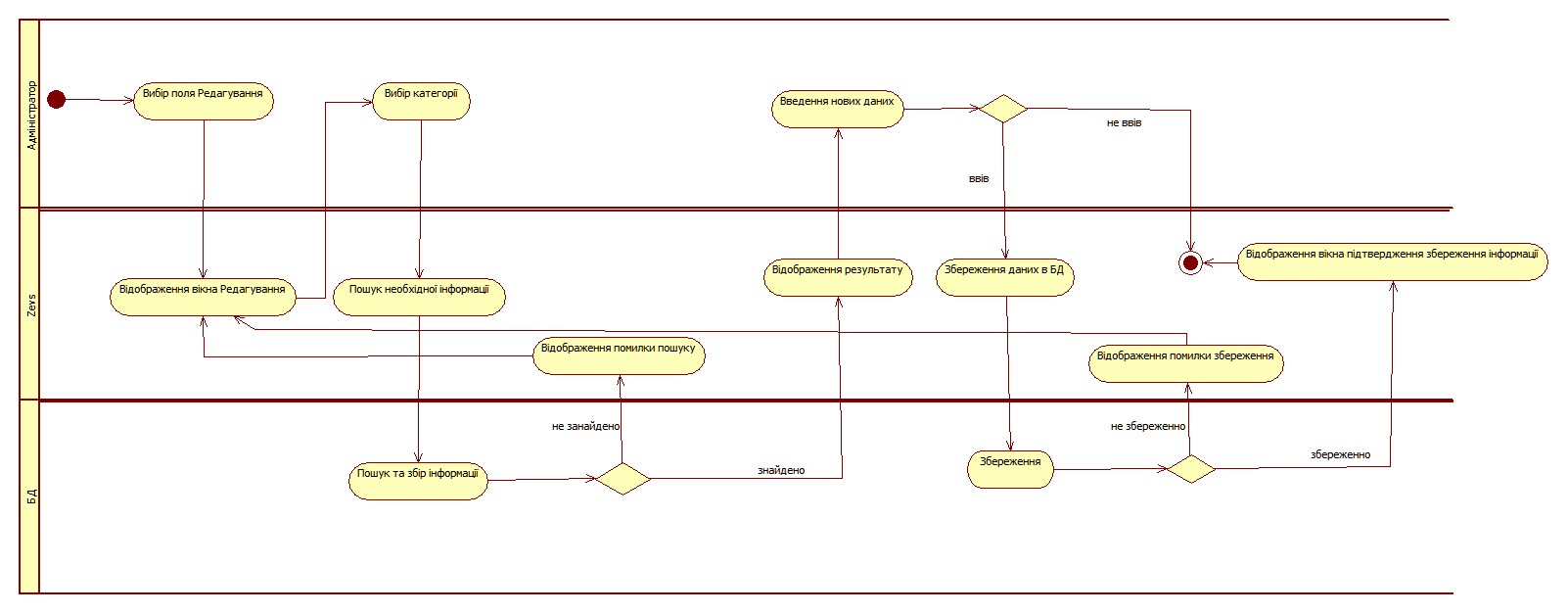
Рисунок 8 - Діаграма діяльності «Редагування»

Рисунок 9 - Діаграма діяльності «Статистика»