# ВСТУП

Теперішній розвиток інформаційних технологій має великий вплив на всі сфери діяльності людини. Саме це стає рушійним фактором для їх впровадження у більшість галузей. Виключенням не стала і генеалогія та генетика, адже за останні роки рівень використання інформаційних систем у цих напрямках значно підвищився.

Майже у кожної особи нашого суспільства виникало прагнення до сімейної історії. Із приходом інформаційних технологій це прагнення стає значно доступнішим, адже з’являються можливості легкого збору, зберігання та аналізування історичних даних за допомогою існуючих програмних продуктів.

Питання використання інформаційних систем для створення генеалогічного дерева родини є цілком актуальним. Проведення всіх етапів збору та аналізу інформації без використання сучасної обчислювальної техніки вимагає кропіткої праці, а також великої кількості часу у відповідного спеціаліста. На практиці часто виникають помилки у неправильному знаходженні та зберіганні інформації про родовід. Тому на сьогоднішній день програмні продукти, що пов’язані з побудовою генеалогічних дерев, є досить важливими в практичному застосуванні, але вони через велику вартість є не зовсім доступними.

Для того щоб інформаційна система мала широке використання, в основу програмного продукту закладають велику базу даних та базу знань. За допомогою даної технології можна отримати змогу збирати та впорядковувати величезну кількість даних, а також проводити різноманітні операції з ними.

Темою дипломної роботи є розробка інформаційної системи генеалогічного дерева родини.

**Актуальність теми** полягає у тому, що дана інформаційна система дозволить покращити зберігання історичної інформації про сім’ю. Є можливість визначення ймовірності захворювання генетичними хворобами від пращурів до нащадків, побудова графіків за масивом інформаційних даних від усіх родичів. Також актуальним є реалізація онлайн мережі у якій є можливість додавання користувачів до друзів з функцією перегляду, та копіювання інформації про рідню. Створення бази знань генетичних захворювань, що можуть проявлятися у людей. Все це надасть можливість зберігати більш різноманітнішу інформацію про свій родовід.

**Об’єкт дослідження** – методи та засоби створення інформаційної системи генеалогічного дерева родини.

**Предмет дослідження** – це програмні засоби для автоматизації зберігання та аналізу інформації про родовід, а також можливість знаходження ймовірності передачі генетичних захворювань від пращурів до нащадків .

**Область застосування –** інформаційна система є корисною для користувачів, які поставили собі за мету сформувати своє генеалогічне дерево, що міститиме всю відому інформацію про предків та дає можливість передбачити генетичні захворювання у нащадків.

**Мета даної роботи** полягає у створенні інформаційної системи генеалогічного дерева родини, що полегшує формування загальної картини роду. Дана система розробляється для роботи з великим потоком даних, а також з метою скоротити витрати часових ресурсів користувачів у цілому.

**Практичне значення** одержаних результатів полягає в формуванні генеалогічного дерева родини, визначення ймовірності прояву генетичних захворювань у нащадків. Автоматизації розрахункових задач даної предметної області.

Дипломна робота складається із вступу, \_\_\_розділів, висновків, списку використаної літератури (\_\_\_ найменувань). Робота містить \_\_\_ таблиці, \_\_\_ рисунка. Роботу викладено на \_\_\_ сторінках друкованого тексту.

# 1 ОГЛЯД ТА порівняльний АНАЛІЗ аналогів ІНФОРМАЦІЙНОЇ СИСТЕМИ «генеалогічного дерева родини»

## 1.1 Призначення та область застосування інформаційної системи генеалогічного дерева родини

Генеалогічне (родовідне) дерево – схематичне представлення родинних зв'язків у вигляді умовно-символічного дерева, в коренях якого знаходиться родозасновник, а на кроні та вітках – представники різних ліній родоводу[Савелов Л. М. Лекции по русской генеалогии. — М.: Археографический центр. 1994. — Репринт издания 1909 года. ISBN 5-86169-015-4 (Т. 3); ISBN 5-86169-015-4 (рос.)]. Таке схематичне зображення надає користувачеві більш зручний вигляд представлення родоводу.

Користувач має можливість пошуку та зберігання різноманітної інформації про своїх предків, але робота із великою кількістю даних одночасно, не є досить зручною та займає велику кількість часу. Тому доцільно використовувати інформаційну систему генеалогічного дерева родини.

Інформаційна система – це сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів[Грицунов О. В. Інформаційні системи та технології. Навчальний посібник. — Х.: ХНАМГ, 2010. — 222 с.].

Інформаційна система генеалогічного дерева має можливість визначення ймовірності прояву генетичних захворювань, що передались від предків до нащадків, це стає можливим завдяки генетичним алгоритмам, зокрема за допомогою законів Менделя, що становлять основу класичної генетики спадковості. Генетичні захворювання це захворювання, обумовлені порушеннями в процесах збереження, передачі та реалізації генетичної інформації. В основі спадкових захворювань лежать мутації: генні, хромосомні та геномні. Відповідно до цього всі спадкові хвороби людини можна об’єднати в 4 великі групи:

1. генні (зміни на рівні окремих нуклеотидів),
2. геномні (зміни кількості цілих хромосом),
3. хромосомні (внутрішньо - і міжхромосомні перебудови),
4. мультифакторіальні (на розвиток хвороби впливають і гени, і фактори навколишнього середовища).

Спадкові захворювання і захворювання із спадковою схильністю є основною причиною смертності, особливо в економічно розвинутих країнах. На підставі великого статистичного матеріалу встановлено, що через генетичні порушення 1 із 130 зародків гине вже в перші дні, 25 % припиняють своє існування на пізніших строках вагітності. 40 % дитячої смертності зумовлено спадковими дефектами, 5-12 % новонароджених мають природжені генетичні дефекти, які зумовлені дефектом хромосом або мутаціями окремих генів. Кожна людина має 5-10 потенціально шкідливих генів, які передаються нащадкам разом з нормальними генами.[http://www.pharmencyclopedia.com.ua/article/591/spadkovi-xvorobi].

Основним призначенням продукту проектування є побудова генеалогічного дерева родини, яке представлене у класичній схемі, формування графіків на основі відомих даних та знаходження ймовірності прояву спадкових генетичних захворювань у майбутніх нащадків.

Основною областю застосування даної інформаційної системи генеалогічного дерева родини є діяльність користувачів які прагнуть дізнатися більше про власний родовід та про його подальший розвиток.

## 1.2 Огляд способів і засобів вирішення поставлених задач

### 1.2.1 Поняття «колективного розуму»

Вислів «колективний розум» поширюється вже кілька десятиліть, але став важливим і популярним з приходом нових комунікаційних технологій. Під цим поняттям розуміють витяг нового знання з об'єднаних переваг, поведінки і уявлень деякої групи людей.

Колективний розум був можливий і до появи Інтернету. Для того щоб збирати дані від розрізнених груп людей, об'єднувати їх і аналізувати, Всесвітня павутина не потрібна. До найважливіших форм подібних досліджень входять соціологічні опитування та переписи. Одержання відповідей від великого числа людей дозволяє робити про групу такі статистичні висновки, які на основі одиничних даних зробити неможливо. Породження нових знань, виходячи з даних, отриманих від незалежних респондентів, – це і є суть колективного розуму.

Прикладом можуть бути фінансові ринки, де ціна встановлюється не за бажанням індивідуума чи шляхом скоординованих зусиль, а в результаті поведінки на торгах безлічі незалежних людей, які діють у власних інтересах. Вважається, що ф'ючерсні ринки, де численні учасники укладають контракти виходячи зі своїх уявлень про майбутні ціни, здатні передбачати ціни більш точно, ніж незалежні експерти. Пояснюється це тим, що такі ринки акумулюють знання, досвід та інтуїцію тисяч людей, а експерт-одинак може покладатися лише на себе. Хоча методи колективного розуму існували і до появи Інтернету, можливість отримувати інформацію від тисяч і навіть мільйонів людей у Всесвітній павутині відкрила найширший спектр нових можливостей. У будь-який момент часу хтось користується Інтернетом для здійснення покупок, в дослідницьких цілях, в пошуках розваг або заради створення власного сайту. Ці дії можна відстежувати і вилучати з них інформацію, навіть не ставлячи користувачеві питань. Існує безліч способів обробити та інтерпретувати цю інформацію. Розглянемо кілька приклади, які ілюструють принципово різні підходи.

Вікіпедія – це онлайн енциклопедія, створювана виключно самими користувачами. Будь-яка людина може створити нову статтю або відредагувати вже існуючу, а повторювані спроби недозволеного використання відсікають не багаточисленні адміністратори. У Вікіпедії більше статей, ніж у будь-якій іншій енциклопедії, і, незважаючи на маніпуляцію деяких зловмисних користувачів, вважається, що за більшістю тем інформація є точною. Це приклад колективного розуму, оскільки кожна стаття підтримується великою групою людей, а в результаті виходить творіння, що набагато перевершує все, що могла б створити одна організована група. Програмне забезпечення Вікіпедії не піддає зібрану інформацію якийсь особливо інтелектуальній обробці; воно просто відстежує зміни і відображає останню версію.

Пошукова система Google – найпопулярніша в світі пошукова машина. У ній вперше для ранжирування веб-сторінок був застосований підхід, заснований на кількості посилань на дану сторінку. Для цього необхідно зібрати інформацію про те, що думають про дану сторінку тисячі людей, і скористатися нею для сортування результатів пошуку. Це зовсім інший приклад колективного розуму. Якщо Вікіпедія запрошує користувачів зробити свій внесок у створення сайту, то Google витягує інформацію з того контенту, який користувачі вже розмістили на своїх власних сайтах, і застосовує її для генерування оцінок від імені користувачів.

Термін «колективний розум» використовується у Всесвітній павутині, означаючи мережеві служби, що обробляють дані, зібрані з різних джерел і від різних учасників, та знаходять серед них найкориснішу інформацію для користувачів. Приклади реалізації: вікі-проекти за спільним накопиченням, вдосконалення та публікації знань (Вікіпедія); сервіси запитань і відповідей Google Answers; Habrahabr; групова розробка відкритого програмного забезпечення та ін[Говард Блум «Глобальный мозг: эволюция коллективного разума от Большого Взрыва до 21-го века» (Howard Bloom, Global Brain: The Evolution of Mass Mind from the Big Bang to the 21st Century, 2000)].

### 1.2.2 Особливості виконання генеалогічного дослідження

Генеалогічне дослідження передбачає пошук інформації у архівах щодо окремої родини на протязі кількох поколінь. Для виконання дослідження необхідна точна біографічна інформація: імена, прізвища та по батькові, місце та роки народження, місце проживання, здобутки відомих членів родини.

Повнота і чіткість відомостей є гарантією правильного пошуку необхідних даних.

Основними джерелами генеалогічних досліджень та історії родини є метричні книги, реєстрації актів громадянського стану. Іншими, також важливими матеріалами є документи посімейно-майнового обліку (для підросійської України це наприклад подвірні, посімейні (погосподарчі) списки, ревізькі «сказки», списки платників податку на прибуток, платників обов'язкового страхування, виборців та громадян, що були позбавлені виборчого права; справи про опікунство, межові книги, документація переписів 1917, 1921, 1926, 1939 рр.), джерела біографічного характеру (послужні (формулярні) списки та особові справи, переліки працівників підприємств, установ, та організацій, учасників релігійних спільнот, учнів, документація про стосунок до військової служби, фільтраційні і архівно-слідчі справи). Опис цих джерел подамо далі.

Виконання генеалогічного дослідження потрібно починати з наявної інформації, це такі дані як рік, дата народження одного з членів сім’ї. Спершу шукаємо актовий запис про народження особи в метричній книзі за місцем її народження.

У разі віднайдення цієї інформації можна дізнатися імена та по батькові батьків. Для того, щоби віднайти інших членів родини (сестер та братів) потрібний уважний перегляд рік за роком актових матеріалів про народження та смерті. Актові записи про шлюб потрібно переглянути для визначення років народження та місця приписки (можливого місця народження) батьків або дівочого прізвища матері.

Потрібно звертати увагу на всіх людей з таким самим прізвищем, оскільки вони можуть виявитися родичами).

Значно прискорить та полегшить виконання дослідження перегляд посімейних списків та ревізьких «сказок». За наведеними у цих документах роками смерті та народження можливо, уникаючи повного перегляду усіх наявних метричних книг, знайти потрібні актові записи. У разі, якщо деякі дані. потрібні для проведення дослідження відсутні, то можна звернутися до інших документів. Характеристику цих документів подамо у наступних статтях.

Для полегшення побудови родоводу рекомендується складати генеалогічні картки на кожну знайдену людину

[http://rodovody.com.ua/index.php/ua/genealogy].

## 1.3 Аналіз переваг та недоліків існуючих аналогів

Програмних продуктів які дозволяють будувати генеалогічні дерева досить багато. Коротко розглянемо переваги та недоліки деяких з них.

### 1.3.1 Програмний продукт «Family Tree Builder»

«Family Tree Builder» - проста зручна у використанні программа, російськомовна. Дозволяє вибудувати особисте генеалогічне дерево та має можливість збереження його в зручних графіках або звітах. Можна додавати до кожного учасника сімейного дерева дату народження або смерті, його фотографії, контактні дані, стать та інші важливі дані. До кожної персони можна додати чоловіка/дружину, дітей, батьків, братів або сестер. Створене дерево можна зберегти у вигляді графіка або звіту з подальшою можливістю друку для зручного перегляду. Інтерфейс програми показана на рисунку 1 та 2.

Можливості:

* експорт свого дерева в Excel;
* створення графіків і звітів;
* збереження графіків і звітів в зручних форматах і можливість друку;
* синхронізація з сайтом розробника для публікації родоводу;
* пошук родичів з понад 65 мільйонів зареєстрованих користувачів;
* додавання точної інформації до кожної персони (ПІБ, дата народження або смерті, інформація про подружжя або дітей, фотографії та ін.).

Плюси:

* потужна програма, що дозволяє створювати сімейне дерево;
* простий і зручний інтерфейс;
* можливість скачати безкоштовно Family Tree Builder;
* наявність російськомовної локалізації.

Мінуси:

* одномовний інтерфейс;
* безкоштовна версія має обмежений функціонал.

Інтерфейс програмного продукту зображено на рисунку 1.1 та рисунку 1.2.

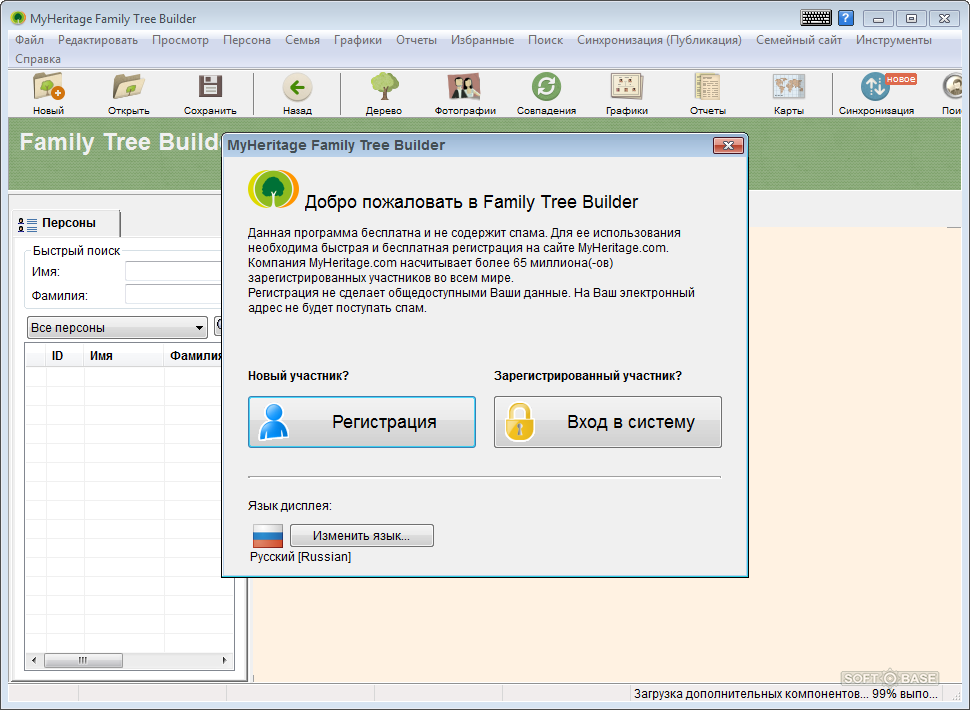


Рисунок 1.1 – Інтерфейс програми «Family Tree Builder»

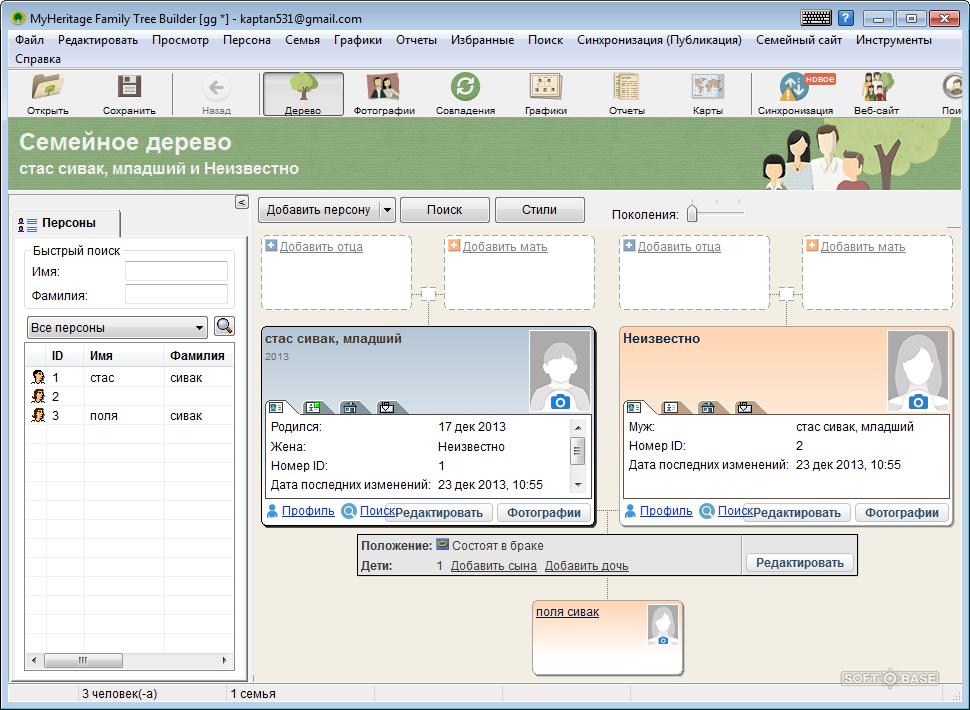


Рисунок 1.2 – Інтерфейс програми «Family Tree Builder»

### 1.3.2 Програмний продукт «SimTree»

Додаток допоможе візуально уявити генеалогічне дерево і внести коротку інформацію про кожну персони, включаючи фотографії. Інтерфейс простий і зрозумілий користувачам. Саме тому SimTree можна скачати і використовувати спільно з дітьми в навчальних цілях. Інтерфейс програми показана на рисунку 3 та 4.

Можливості:

* побудова генеалогічного дерева;
* табличне відображення інформації про кожну особу;
* додавати особисті дані до карток;
* автоматичне визначення родинних зв'язків при додаванні нових членів сім'ї.

Плюси:

* простий інтерфейс;
* зберігання практично необмеженого кількість даних;
* можливість прикріплювати файли.

Мінуси:

* обмежений функціонал;
* єдина схема візуального відображення спорідненості.

Інтерфейс програмного продукту представлено на рисунку 1.3.

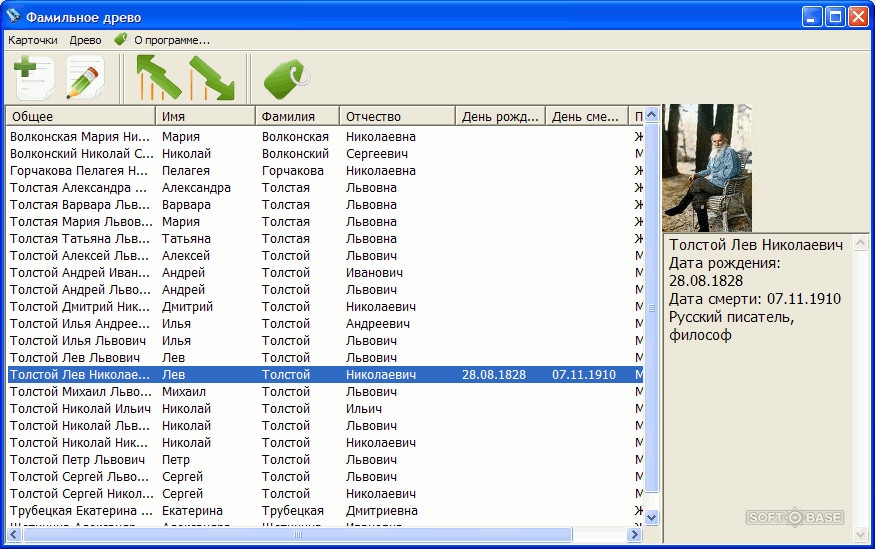


Рисунок 1.3 – Інтерфейс програми «SimTree»

### 1.3.3 Програмний продукт «Genopro»

Програма дозволяє швидко створити сімейне дерево навіть не дуже досвідченим користувачам. Завдяки наявності «гарячих клавіш» ви можете легко додати потрібних вам людей і інформацію про них. Програма умовно-безкоштовна: безкоштовна версія має обмежений термін дії (15 днів) і обмежена кількість персон для додавання (до 25 осіб). Щоб підвищити функціонал програми, необхідно буде купити ліцензійний ключ. Інтерфейс програми показана на рисунку 5 та 6.

Можливості

* наявність вбудованого генератора звітів;
* всі дані доступні для перегляду і редагування в табличному форматі;
* створення призначених для користувача вкладок даних для введення додаткової інформації;
* додавання інформації про відносини між людьми (наприклад, закоханість або конфлікт у відносинах);
* збереження резервної копії онлайн;
* відправка створеної родоводу іншим родичам по e-mail;
* друк створеної родоводу.

Плюси:

* програма має російськомовну локалізацію;
* наявність «гарячих клавіш»;
* можливість безкоштовного завантаження Genopro;
* експорт в різні формати, наприклад XML, Gedcom або TXT;
* імпорт сімейного дерева в форматі Gedcom, створеного в аналогічних програмах.

Мінуси:

* обмежений термін дії безкоштовної версії (15 днів);
* обмежена кількість створених персон в пробної версії (25 осіб).

Інтерфейс програмного продукту зображено на рисунках

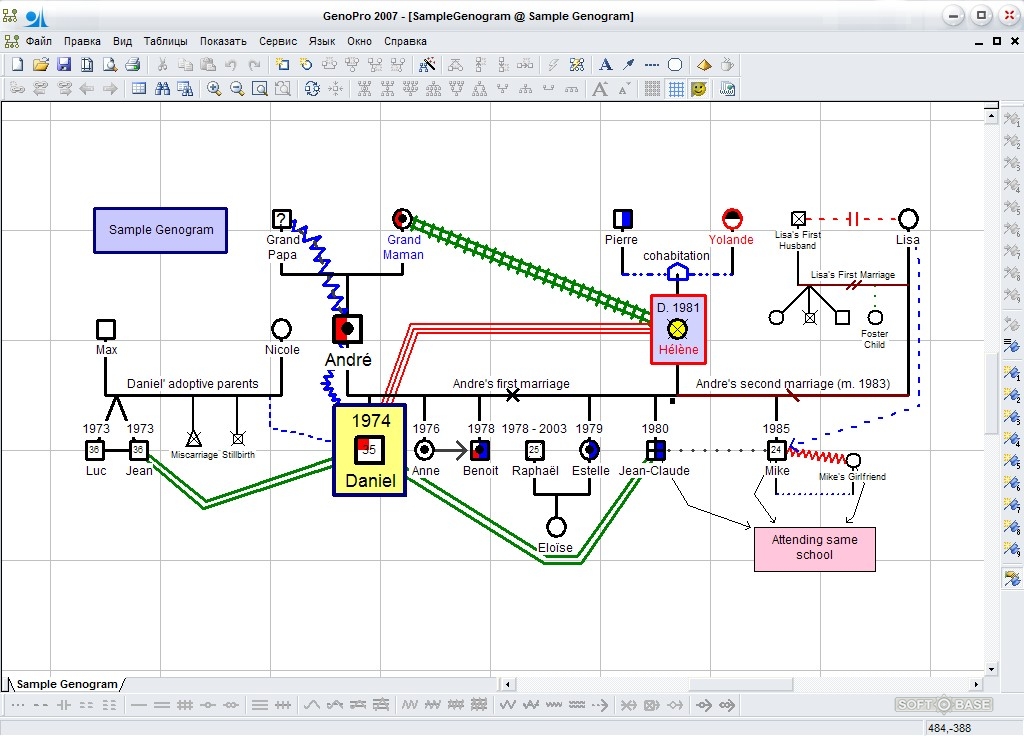


Рисунок 1.4 – Інтерфейс програми «Genopro»

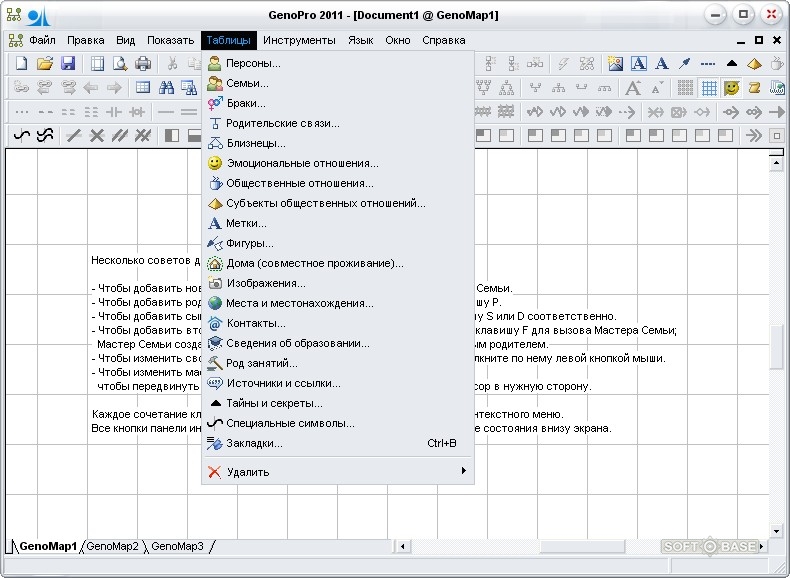


Рисунок 1.5 – Інтерфейс програми «Genopro»

### 1.3.4 Програмний продукт «RootsMagic Essentials»

Програма «RootsMagic Essentials» одна з найстаріших, її творець Bruce Buzbee написав першу версію ще в далекому 1987 році. З тих пір вона звичайно неодноразово змінювалася і допрацьовувалася і на даний момент представляє дуже симпатичну і функціональну платформу, що володіє масою переваг. Інтерфейс програми показана на рисунку 7 та 8.

Можливості:

* складання генеалогічного дерева за допомогою зручної візуальної схеми;
* дослідження предків за допомогою спеціальних алгоритмів;
* прив’язку людей до місцевості;
* пошук інформації про осіб у відкритих джерелах.

Плюси:

* простий, красивий і доступний інтерфейс;
* широкий функціонал;
* можливість публікувати своє дерево онлайн і пов'язувати його з іншими.
* Мінуси:
* програма не русифікована;
* багато важливих функцій доступні тільки в платній версії.

Інтерфейс програмного продукту представлено на рисунку 1.6 та рисунку 1.7.

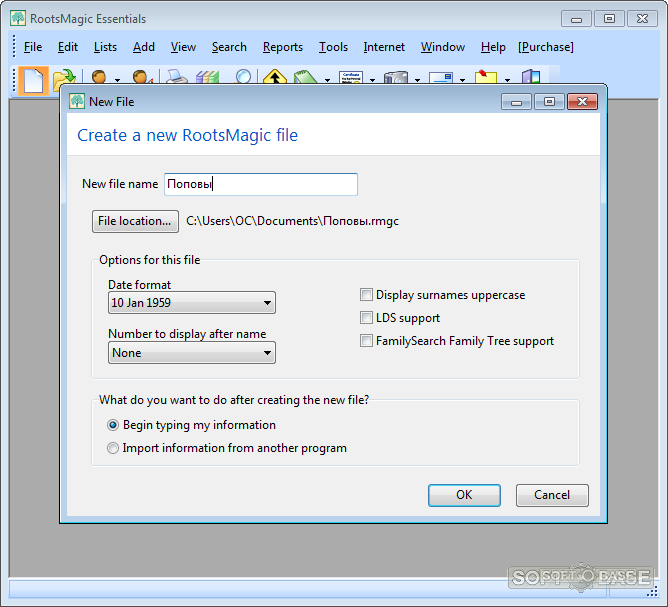


Рисунок 1.6 – Інтерфейс програми «RootsMagic Essentials»

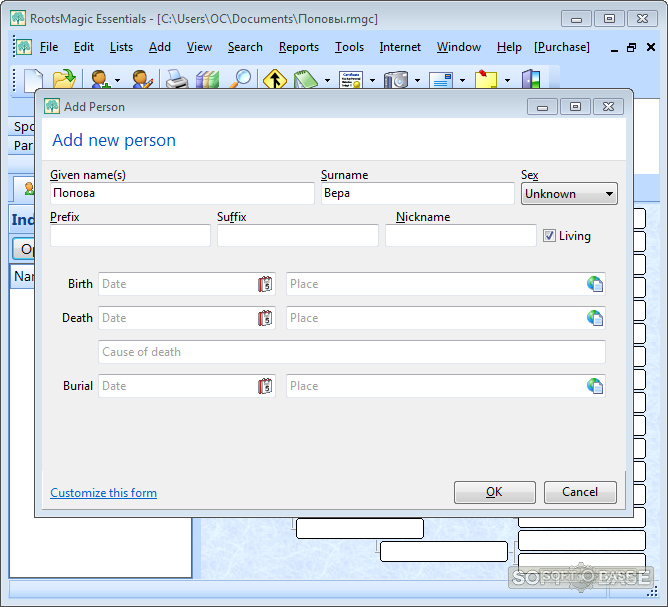


Рисунок 1.7 – Інтерфейс програми «RootsMagic Essentials»

## 1.4 Постановка задачі на розробку

Інформаційна система генеалогічного дерева родини повинна відповідати вимогам та виконувати такі функції, як:

* реєстрація нового користувача;
* вхід зареєстрованого користувача;
* створення генеалогічного дерева для користувача;
* додавання родичів до дерева користувача;
* можливість доручення до друзів;
* можливість дозволу перегляду дерева іншим користувачам;
* автоматичне формування генеалогічного дерева;
* ведення статистичного аналізу даних по родичам;
* можливість копіювання інформації із інших дерев;
* додавання нових генетичних захворювань;
* можливість знаходження ймовірності захворювання спадковою хворобою нащадка.

Також вся система повинна працювати завдяки Web-API, адже це дозволить поширити її на всі існуючі платформи. Всі дані які повертатимуться із сервера повинні бути у форматі JSON. Для забезпечення захисту особистих даних користувача, потрібно реалізувати шифрування його паролю для входу.

Для правильного визначення ймовірності прояву спадкових захворювань необхідно вказати достовірну інформацію про хворобу таку, як можливість передачі її жінкам або чоловікам, а також її форму прояву у пращура домінантна чи рецесивна.

## 1.5 Висновки до першого розділу

В даному розділі було проаналізовано методи та засоби для розробки якісного програмного продукту на тему «Інформаційна система «Генеалогічне дерево родини»». Було вивчено конкретну область, для врахування особливостей, якими буде володіти дана інформаційна система.

Проаналізувавши опрацьовану літературу, можна сказати, що генеалогія та генетика посідає досить значне місце у розвитку сімейної історії. Використання даного програмного продукту дозволить скласти умови для більш ефективного проведення збору інформацій про власну родину.

У першому розділі розглянуто аналоги для розроблюваної інформаційної системи. Існує велика кількість програмного забезпечення для побудови генеалогічного дерева родини, але немає такої системи , яка передбачає наявність у осіб певного спадкового захворювання. Тому розробка даної інформаційної системи є досить актуальна.

Також під час написання першого розділу було проаналізовано фактичний стан задачі, що розв’язується, на основі матеріалів, які характеризують об’єкт проектування; містить теоретичний виклад важливих аспектів проблеми, що досліджується, та критичний огляд наукової літератури і періодичних джерел з визначеної тематики.

Сформовано технічне завдання, в якому передбачено всі переваги та недоліки існуючих аналогів. Та включені ті функції, які не було знайдено у аналогів, що і робить даний програмний продукт унікальним.

# 2 Аналіз і проектування інформаційної системи «генеалогічне дерево родини»

У даному розділі розглядаються різновиди вимог, поданих для програмних систем. Обговорюються їх характеристики та пояснюються основні процеси для роботи з вимогами: формування вимог, аналіз вимог, управління змінами у вимогах.

Для того, щоб побудувати щось, потрібно зрозуміти мету завдання. Розуміння і документування цього називається аналізом вимог. Основна задача визначення вимог – знайти, обговорити та зафіксувати те, що дійсно потребується від системи в формі, зрозумілій і клієнтам, і членам команди розробників

## 2.1 Моделювання предметної області системи

Модель предметної області – це візуальне подання концептуальних класів або об’єктів реального світу в термінах предметної області. Такі моделі також називають концептуальними моделями, моделями об’єктів предметної області, або об’єктними моделями аналізу. Це свого роду словник основних абстракцій, тобто найважливіших іменників у просторі задачі. Іменники, які описують поняття з предметної області, називають доменними об’єктами. На самому початку аналізу і проектування необхідно створити модель предметної області, в якій всі доменні об’єкти будуть зображені на одній великій діаграмі класів.

У термінології UML модель предметної області – це, по суті справи, діаграма класів. Зазвичай в цій моделі опускається велика частина деталей, зокрема атрибути та операції класів. Ось чому можна вважати, що модель предметної області є зведеною діаграмою класів, як візуальний словник важливих абстракцій або словник предметної області [Методичні рекомендації щодо виконання дипломних робіт для студентів денної та заочної форм навчання напряму підготовки 6.050101 – «Комп’ютерні науки». / Авраменко В.С., Голуб С.В., Салапатов В.І. – Видання друге, перероблене і доповнене. – Черкаси: ЧНУ, 2016. – 108 с.].

На етапі моделювання предметної області можна створити діаграму класів, в якій будуть відображені основні ідея та логіка роботи системи, без поглибленого опису атрибутів і методів класів та їх взаємодії між собою. Це дозволить відобразити головну логічну концепцію системи, що посприяє подальшому проектуванню даної системи.

Основні класи програми представлено на рис. 2.1 та рис. 2.2.

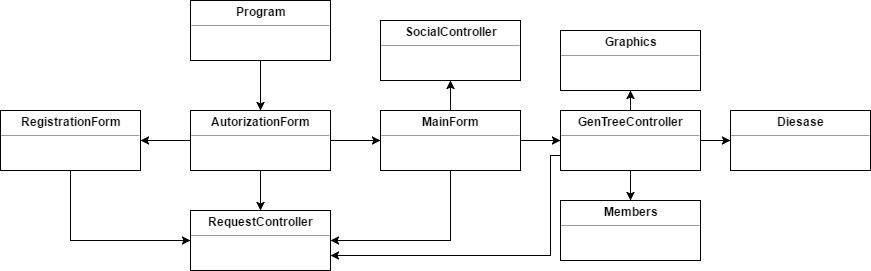


Рисунок 2.1 ­– Діаграма класів клієнтської частини інформаційної системи генеалогічного дерева родини

Класи «Program» та «MainForm», зображені на рис. 2.1 це головні класи програми, що відповідають за керування програмою та перегляд результатів її роботи.

Класи «AutorizationForm» та «RegistrationForm» – це класи, що відповідають за авторизацію та реєстрацію користувача відповідно. У класі «SocialController» реалізовані можливості зв’язку з іншими користувачами програми. Для генерації дерева та різноманітних графіків використовується класи «GenTreeController» та «Graphics», щоб це стало можливим їм потрібні дані, що знаходяться у класах «Disease» та «Members». Всі дані програма отримує з сервера за допомогою утиліти, що описана в класі «RequestController».

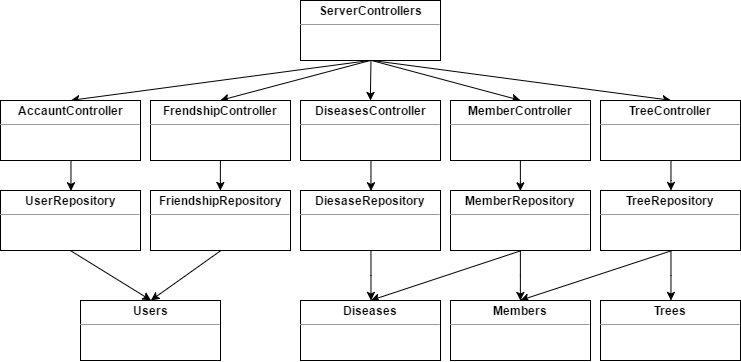


Рисунок 2.2 ­– Діаграма класів серверної частини інформаційної системи генеалогічного дерева родини

Класи, що зображені на рис 2.2 відповідають за збереження та обробку даних на сервері. Для реалізації API використовується клас «ServerControllers» у ньому описані всі методи зв’язку з клієнтською частиною.

Класи «AccauntController», «FriendshipController», «DiseasesController», «MemberController», «TreeController» ­– це класи для обробки запитів які надходять від клієнта. За зв’язок з даними, що знаходиться на сервері відповідають класи «UserRepository», «FriendshipRepository», «DiseaseReposito», «MemberRepository» та «TreeRepository».

## 2.2 Формулювання та аналіз вимог до інформаційної системи

Розширення сфери застосування інформаційних технологій і постійно зростаюча складність інформаційних систем привели до підвищення значущості перед проектного аналізу та робіт з бізнес-аналізу в цілому. Одним з ключових завдань у даній області є виявлення вимог до побудови систем.

Вимоги для інтелектуальних технологій є концентрованим відображенням потреб зацікавлених у створенні системи замовників, користувачів і розробників. Вони виникають у процесі збору первинної інформації, обговорення проекту системи, аналізу предметної області та мають тенденцію бути модифікованими під час виконання проекту [Литвинов В.В., Голуб С.В., Григор’єв К.М., Жигульська В.Ю. Об’єктно-орієнтоване моделювання при проектуванні вбудованих систем і систем реального часу. [Навчальний посібник з дисципліни: «Системний аналіз та проектування комп’ютерних інформаційних систем»]. – Черкаси: Вид. Від ЧНУ ім. Б. Хмельницького, 2011. – 376 с.].

Формування та аналіз вимог полягають у визначенні потреб та умов, які висуваються щодо нового чи зміненого продукту, враховуючи можливо конфліктні вимоги різних замовників, таких як користувачі.

Аналіз вимог є критичним для успішної розробки проекту. Вимоги мають бути задокументованими, вимірними, протестованими, пов’язаними з бізнес-потребами, і описаними з рівнем деталізації, достатнім для конструювання системи. Вимоги можуть бути архітектурними, структурними, поведінковими, функціональними, та не функціональними [Методичні рекомендації щодо виконання дипломних робіт для студентів денної та заочної форм навчання напряму підготовки 6.050101 – «Комп’ютерні науки». / Авраменко В.С., Голуб С.В., Салапатов В.І. – Видання друге, перероблене і доповнене. – Черкаси: ЧНУ, 2016. – 108 с.].

Основні вимоги для створення системи:

* можливість отримувати дані про генеалогічні дерева з серверу;
* наявність сервера, який дозволятиме отримувати всі дані;
* застосування розробленого алгоритму побудови генеалогічного дерева та прогнозування прояву генетичних захворювань;
* можливість графічного представлення генеалогічного дерева;
* можливість створення звітів та графіків за даними з генеалогічного дерева;
* наявність зручного інтерфейсу користувача;
* простота освоєння програми та роботи з нею;
* відкритість для модифікацій і доповнень наступними версіями і модулями;
* можливість додавання доручення до друзів та перегляд дерев друзів.

Аналіз вимог слугує мостом між підготовкою, плануванням і проектуванням програмного забезпечення.

Реалізація процедур цього етапу проектування відбувається в процесі побудови відповідних діаграм мовою UML або іншими мовами, що використовують системи автоматизованого проектування. Результати процедур із формулювання та аналізу вимог оформляються у вигляді діаграми прецедентів (варіантів використання, Use Case), діаграми діяльності, діаграми взаємодій (діаграми послідовностей і комунікації ) та їх описів.

### 2.2.1 Формування вимог за допомогою діаграми прецедентів

Діаграми прецедентів відіграють важливу роль не тільки у комунікації між збирачами вимог до проекту і потенційними користувачами. Діаграми прецедентів дописані бізнес логікою і детальними специфікаціями прецедентів, як джерельна інформація, успішно використовують учасники розробки проекту на всіх його фазах.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами [Диаграмма прецедентов. [Електронний ресурс]. – Режим доступу: [http://ru.wikipedia.org/wiki/ Диаграмма\_прецедентов](http://ru.wikipedia.org/wiki/%20Диаграмма_прецедентов)].

Візуальне моделювання в UML можна подати як деякий процес спуску від найбільш загальної і абстрактної концептуальної моделі початкової системи до логічної, а потім і до фізичної моделі відповідної програмної системи. Для досягнення таких цілей спочатку будується модель у формі так званої діаграми прецедентів, яка описує функціональне призначення системи або, іншими словами, те, що система робитиме в процесі свого функціонування.

Таким чином, діаграма прецедентів є вихідним концептуальним уявленням або концептуальною моделлю системи в процесі її проектування і розробки.

Основними елементами діаграми прецедентів є дійові особи, варіанти використання і відношення між ними. Дійова особа – це роль, яку користувач грає по відношенню до системи.

На рис. 2.3 відображено узагальнене представлення функціоналу системи з використанням таких прецедентів:

* вибір даних;
* вирахування ймовірності захворювань;
* перегляд результатів ймовірності;
* формування звітів та графіків;
* отримання даних з серверу;
* формування генеалогічного дерева.

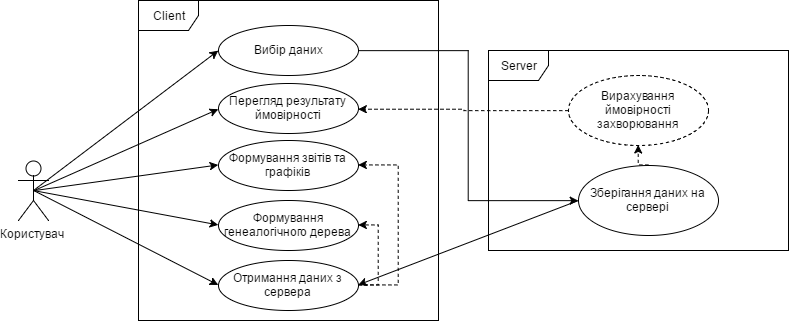


Рисунок 2.3 ­– Діаграма прецедентів інформаційної системи генеалогічного дерева родини

Розроблювана програма буде універсальною та матиме можливість отримувати дані із сервера. Всі головні операції відбуватимуться на сервері, в разі необхідності їх можна змінити, або дописати нові утиліти. Для зв’язку з сервером будь-яка клієнтська програма може використовувати API, що надає змогу даному продукту бути кросплатформенним.

Отже, цей прецедент – це узагальнене представлення функціоналу як сервера так і клієнта, завдяки ньому дана система зможе отримувати потрібні дані а також представляти їх у зручній для користувача формі.

З рисунку 2.3 видно, що проектована система представляється у вигляді множини сутностей, взаємодіючих із системою за допомогою так званих варіантів використання. Варіант використання служить для опису сервісів, що система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, чинений системою при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

З вище згаданих варіантів використання, користувач може обрати дані про родичів , які будуть розраховувати ймовірність захворювання у їх нащадків (якщо в цьому є потреба), та надсилати серверу запит обробляти ці дані. Після цього користувач може передивитись результати проведених обрахунків. Також він може створити різні графіки, що відображатимуть статистику його генеалогічного дерева.

Деякі прецеденти залежать від інших прецедентів. Наприклад, якщо необхідно відразу сформувати звіт та графіки , програма все одно має сформувати генеалогічне дерево, а для цього їй потрібні самі дані, які в свою чергу беруться з серверу.

### 2.2.2 Формування та аналіз вимог за допомогою діаграми діяльності

Діаграма діяльності – в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності.

Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку [Авраменко В.С., Голуб С.В., Салапатов В.І. Проектування інформаційних систем. [Навчальний посібник] – Черкаси: Вид. Від ЧНУ ім. Б. Хмельницького, 2015. – 501 с.].

Діаграма діяльності системи автоматизації аналізу господарської діяльності підприємства зображена на рис. 2.4.

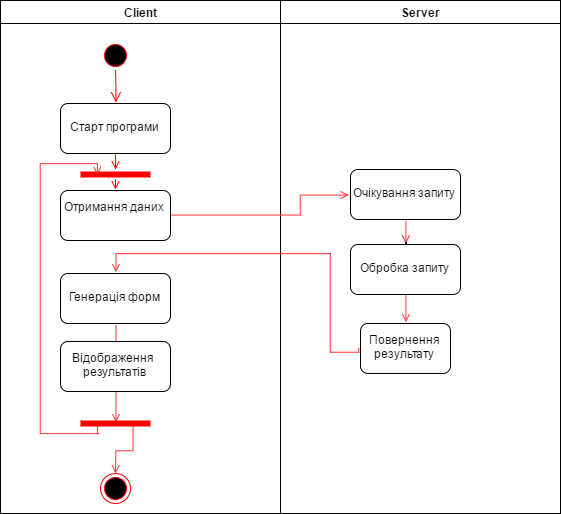


Рисунок 2.4 – Діаграма діяльності системи генеалогічного дерева родини

На рис. 2.4 зображений алгоритм, який виконується протягом роботи даного програмного продукту.

Застосування діаграм діяльності для реалізації варіантів використання не дуже наближує до появи цільового артефакту ­­– програмного коду, але може привести до більш глибокого розуміння даної задачі. Діаграма діяльності дозволяє будь-кому, хто виконує даний процес, обирати порядок дій. Іншими словами, діаграма тільки встановлює правила обов’язкової послідовності дій, за якими потрібно слідувати.

### 2.2.3 Аналіз вимог за допомогою діаграми послідовностей

У мові UML взаємодія елементів розглядається в інформаційному аспекті їх комунікації, тобто взаємодіючі об’єкти обмінюються між собою деякою інформацією. При цьому інформація приймає форму закінчених повідомлень. Іншими словами, хоча повідомлення і має інформаційний зміст, воно набуває додаткової властивості робити направлений вплив на свого одержувача. Будь-які види інформаційної взаємодії між елементами системи повинні бути зведені до відправки і прийому повідомлень між ними.

Для моделювання взаємодії об’єктів в мові UML використовуються відповідні діаграми взаємодії. Кажучи про ці діаграми, мають на увазі два аспекти взаємодії. По-перше, взаємодії об’єктів можна розглядати в часі, і тоді для представлення тимчасових особливостей передачі і прийому повідомлень між об’єктами використовується діаграма послідовності. По-друге, можна розглядати структурні особливості взаємодії об’єктів. Для представлення структурних особливостей передачі і прийому повідомлень між об’єктами використовується діаграма кооперації.

На діаграмі послідовності зображуються виключно ті об’єкти, які безпосередньо беруть участь у взаємодії і не показуються можливі статичні асоціації з іншими об’єктами. Для діаграми послідовності ключовим моментом є саме динаміка взаємодії об’єктів в часі.

Діаграма послідовностей інформаційної системи генеалогічного дерева родини зображена на рис. 2.5, з якого видно, як саме в даній системі об’єкти взаємодіють між собою у часі. Діаграма відображає роботу системи у випадку, коли користувач виконує основні дії, покладені за мету даної роботи, а саме – отримання з серверу даних та генерація генеалогічного дерева, вирахування ймовірності прояву генетичних захворювань, перегляд результату ймовірностей та генерування графіків відносно отриманих даних.

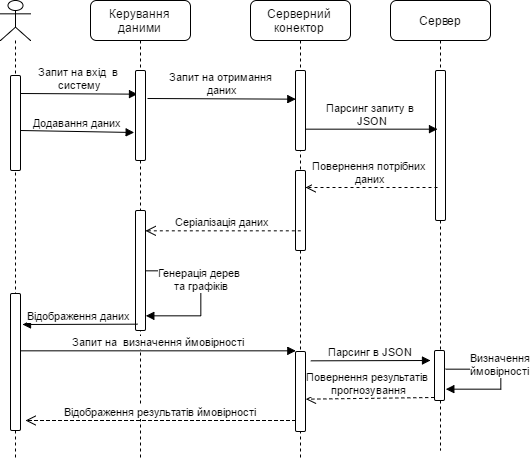


Рисунок 2.5 – Діаграма послідовностей інформаційної системи генеалогічного дерева родини

Окрім цього, на рисунку 2.5 відображена також робота з сервером через серверний «конектор», що слугує засобом обміну даними між клієнтською програмою та сервером. Дана логіка роботи через «конектор» надає програмі властивостей універсальності та в майбутньому дозволить працювати з багатьма програмами, що писатимуться на будь-якій мові програмування та під будь-яку платформу.

### 2.2.4 Функціонування системи за допомогою діаграми комунікації

Діаграма комунікації моделює взаємодії між об’єктами або частинами в термінах впорядкованих повідомлень. Комунікаційні діаграми представляють комбінацію інформації, взятої з діаграм класів, послідовності і варіантів використання, описуючи одночасно і статичну структуру і динамічну поведінку системи.

Діаграма комунікації – діаграма, на якій зображуються взаємодії між частинами композитної структури або ролями кооперації. На відміну від діаграми послідовності, на діаграмі комунікації явно вказуються відносини між об’єктами, а час як окремий вимір не використовується (застосовуються порядкові номери викликів).

Комунікаційні діаграми мають вільний формат упорядкування об’єктів і зв’язків як в діаграмі об’єктів. Щоб підтримувати порядок повідомлень при такому вільному форматі, їх хронологічно нумерують. Читання діаграми комунікації починається з повідомлення 1 і триває по напрямку пересилання повідомлень від об’єкта до об’єкта. Діаграма комунікацій даної системи зображена на рис. 2.6.

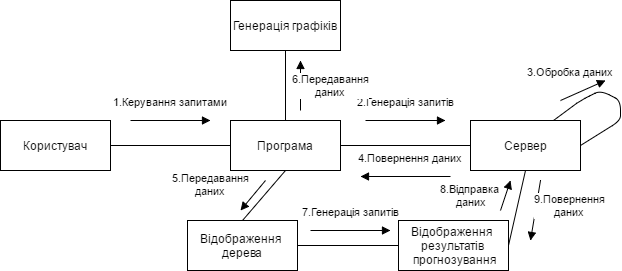


Рисунок 2.6 – Діаграма комунікацій системи генеалогічного дерева родини

Як видно з рис. 2.6, діаграма комунікації відображає взаємодію ролей або об’єктів у процесі функціонування системи, описують обмін даними (повідомленнями) між різними учасниками взаємодії. Такі діаграми моделюють сценарії поведінки системи. Замість того щоб малювати кожного учасника у вигляді лінії життя і показувати послідовність повідомлень, розташовуючи їх по вертикалі, як це робиться в діаграмах послідовності, комунікаційні діаграми допускають довільне розміщення учасників, дозволяючи малювати зв'язки, що показують відносини учасників, і використовувати нумерацію для представлення послідовності повідомлень.

Об’єкти взаємодіють один з одним за допомогою зв’язків – каналів для передачі повідомлень. Зв’язок між парою об’єктів розглядається як екземпляр асоціації між їх класами. Іншими словами, зв’язок між двома об’єктами існує тільки тоді, коли є асоціація між їх класами.

## 2.4. Об’єктно-орієнтоване проектування інформаційної системи

Для створення інформаційної системи була обрана об’єктно-орієнтована технологія проектування.

Метод об’єктно-орієнтованого проектування базується на: моделі побудови системи як множини об’єктів абстрактного типу даних, модульній структурі програм, спадному проектуванні, що використовується при виділенні об’єктів.

В об’єктно-орієнтованому підході в якості так званих блоків використовуються об’єкти, що містять свої власні коди та дані. Будова програм при об’єктно-орієнтованому підході показується графом взаємодії об’єктів, а не деревом ієрархії, як це відбувається в структурному проектуванні.

За основу структурного підходу взято структуризацію і декомпозицію навколишнього світу. Задача будь-якої складності розділяються на під задачі, а ті, у свою чергу, розділяються далі, аж поки кожна під задача не стане простою, відповідною підпрограмі, що виконує тільки одну дію. Методи структурного проектування застосовують модулі в якості блоків програми, а структура програми представлена ієрархією підпорядкованості модулів.

Можна сказати використання об’єктів при розробці систем фактично зробило можливим перехід від порівняно обмеженої бази, якою є алгоритм, до більш ширших структурних та поведінкових аспектів представлення системи, що і надало змогу наповнити змістом сам процес аналізу та розробки.

Виокремлюють такі основні плюси об’єктно-орієнтованого проектування:

– наявність різнопланових модельних поглядів;

– застосування абстракції предметної області;

– стабільність в представленні змін;

– покращення можливості для повторного використання;

– можливість управління масштабністю;

– краща підтримка аспектів надійності;

– успадковується підтримка паралелізму. [Литвинов В.В. Об’єктно-орієнтоване моделювання при проектуванні вбудованих систем і систем реального часу / В.В. Литвинов, С.В. Голуб. – Черкаси: ЧНУ ім. Богдана Хмельницького, 2011. – 376 с.].

### 2.4.1 Архітектурне проектування

Формування архітектури – перший і основний крок у вирішенні завдання проектування, що закладає фундамент уявлення програмної системи, здатної виконувати весь спектр детальних вимог.

Створення архітектури – це проектування на найвищому рівні (логічна архітектура). Логічна архітектура описує систему в термінах її принципової організації у вигляді пакетів, програмних класів і підсистем. Вона називається логічною, оскільки не визначає способи розгортання цих елементів у різних операційних системах або на фізичних комп’ютерах в мережі (це відноситься до архітектури розгортання).

Архітектурне проектування полягає у визначенні головних структурних особливостей системи, яку будують, а саме: складу компонент, способів їхньої композиції, обмежень на їхні з’єднання.

Етап архітектурного проектування інформаційної системи повинен бути відображений шляхом опису процесу об’єктно-орієнтованої декомпозиції системи до рівня переліку підсистем та їх зв’язку, опису її логічної структури у вигляді компонентів (діаграма компонентів UML) [Авраменко В.С., Голуб С.В., Салапатов В.І. Проектування інформаційних систем. [Навчальний посібник] – Черкаси: Вид. Від ЧНУ ім. Б. Хмельницького, 2015. – 501 с.].

#### 2.4.1.1 Діаграма пакетів

Діаграма пакетів – це структурна діаграма, в якій основними елементами є пакети і залежності між ними. Вона служить у першу чергу, для організації елементів у групи за певною ознакою з метою спрощення структури та організації роботи з моделлю системи. Класи складають структурний кістяк об'єктно-орієнтованої системи[Методичні рекомендації щодо виконання дипломних робіт для студентів денної та заочної форм навчання напряму підготовки 6.050101 – «Комп’ютерні науки». / Авраменко В.С., Голуб С.В., Салапатов В.І. – Видання друге, перероблене і доповнене. – Черкаси: ЧНУ, 2016. – 108 с.].

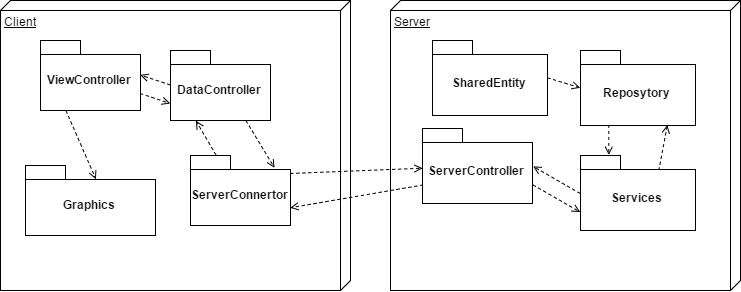
На рисунку 2.7 зображена загальна структура пакетів даної системи.

Рисунок 2.7 – Діаграма пакетів системи генеалогічного дерева родини

Як можна побачити з рис 2.7, даний програмний продук поділяється на дві програми «Client», «Server». У клієнті для зв’язку з сервером використовуватиметься пакет «ServerConnector». У сервері головну роль у зв’язку з клієнтом виконуватиме пакет «ServerController». Дана архітектура дозволить як на цьому етапі розробки, так і на майбутнє, ефективно використовувати модель роботи програми та створювати додаткові компоненти для логічної будови системи.

#### 2.4.1.2 Діаграма компонентів

Ще одним будівельним блоком для створення архітектури об'єктно-орієнтованої системи вважається компонент. Діаграма компонентів показує визначення, внутрішню структуру і залежності набору компонентів. В якості фізичних компонент можуть виступати файли, бібліотеки, модулі, виконувані файли, пакети.

Компонент – модульна і замінна частина системи, яка відповідає набору інтерфейсів і забезпечує реалізацію цього набору інтерфейсів. Реалізація компоненту завжди прихована. Компоненти системи з однаковими наборами інтерфейсів є взаємозамінними.

Той факт, що між двома компонентами завжди знаходиться інтерфейс, усуває їх пряму залежність. Компонент, що використовує забезпечений інтерфейс, функціонуватиме правильно незалежно від того, який компонент реалізує цей інтерфейс. Це дуже важливо і забезпечує гнучку заміну компонентів в інтересах розвитку системи.[Методичні рекомендації щодо виконання дипломних робіт для студентів денної та заочної форм навчання напряму підготовки 6.050101 – «Комп’ютерні науки». / Авраменко В.С., Голуб С.В., Салапатов В.І. – Видання друге, перероблене і доповнене. – Черкаси: ЧНУ, 2016. – 108 с.].

На рисунку 2.8 зображена діаграма компонентів даної інформаційної системи.

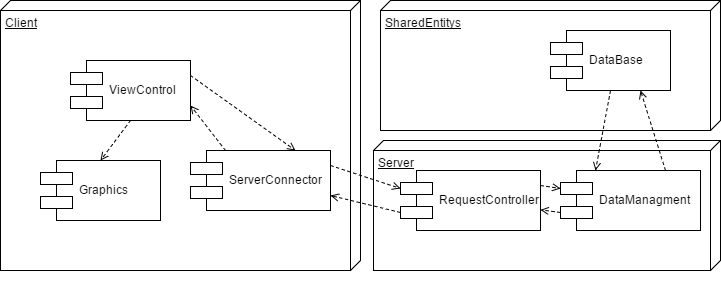


Рисунок 2.8 – Діаграма компонентів системи генеалогічного дерева родини

Діаграма компонентів, що зображена на рис. 2.8, показує компоненти в системі, тобто програмні підсистеми, з яких створюється архітектура, а також залежності між компонентами. Вона відображає модель роботи програми за тим самим принципом, що був описаний раніше, лише з додаванням логіки використання суспільних даних для роботи. Компонент «Data Management» виконує функцію так званої серіалізації даних, що надходять з серверу. Даний компонент дуже важливий, адже з будь-якого виду даних він може перетворити їх у такий вигляд, з яким зможе ідеально працювати наша система.

Також, компонент «SharedEntitys», що відповідає за збереження даних у базу даних, яка знаходиться на сервері.

### 2.4.2 Детальне проектування

Другим ступенем проектування, який слідує за створенням архітектури є детальне проектування. У ході цієї діяльності орієнтуються на максимальну підготовку до кодування програмної системи. Програмісти повинні отримати детальні проектні рішення, які забезпечать їх повною інформацією для створення програмного коду.

Основними будівельними блоками детального проектування єкласи. Результат етапу детального проектування повинен бути відображений діаграмами класів (що містять опис всіх операцій та атрибутів класів), які призначені для статичного моделювання об’єктів.

Діаграма класів описує типи об’єктів системи і різного роду статичні відносини, які існують між ними. На діаграмах класів відображаються також властивості класів, операції класів та обмеження, які накладаються на зв’язку між об’єктами. В UML термін функціональність застосовується в якості основного терміну, що описує і властивості, і операції класу.

Діаграми класів вважають основним засобом для представлення структури систем в термінах базових будівельних блоків і відносин між ними.[Методичні рекомендації щодо виконання дипломних робіт для студентів денної та заочної форм навчання напряму підготовки 6.050101 – «Комп’ютерні науки». / Авраменко В.С., Голуб С.В., Салапатов В.І. – Видання друге, перероблене і доповнене. – Черкаси: ЧНУ, 2016. – 108 с.].

#### 2.4.2.1 Діаграма класів

### 2.4.3 Розгортання програмної системи на апаратних засобах

#### 2.4.3.1 Діаграма розгортання

## 2.5 Модель поведінки інформаційної системи генеалогічного дерева родини

### 2.5.1 Діаграма кінцевого автомату

### 2.5.2 Діаграма діяльності?

### 2.5.3 Діаграма взаємодії(послідовності та комунікацій)?