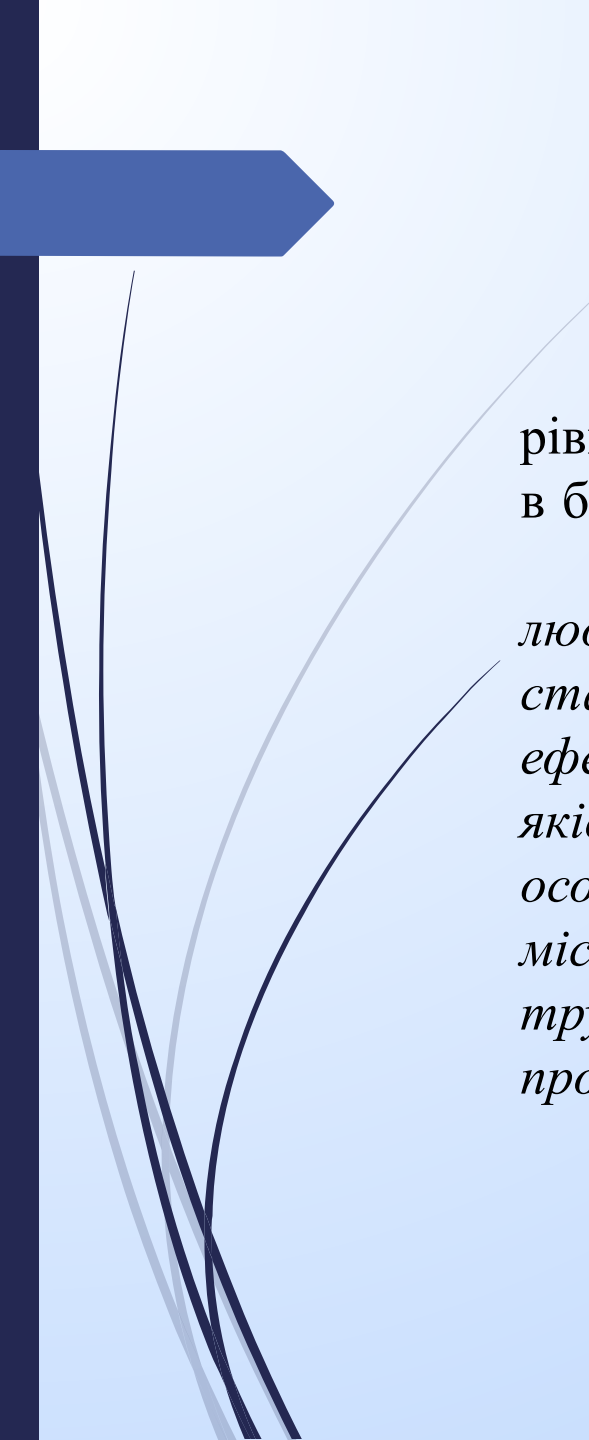


**Харківській національний університет
радіоелектроніки
Кафедра системотехніки**

**Курсовий проект на тему
Проектування CASE-засобу “ER-Builder”**

Виконав:
Студент групи ІТПм-21-2
Павленко М.Ю.

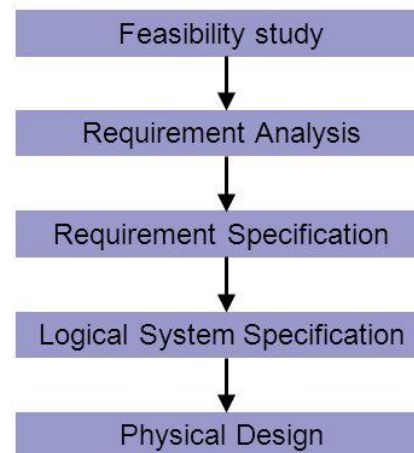


В даний час темпи розвитку суспільства багато в чому визначаються рівнем його інформатизації. Це передбачає широкомасштабне впровадження в багато сфер людської діяльності складних інформаційно-керуючих систем.

Процес проектування таких систем вимагає великих витрат часу, людських і матеріальних ресурсів. У той же час при відсутності стандартизованої технології проектування немає ніяких гарантій створення ефективної або навіть просто працездатною ІУС. Це пояснюється тим, що якість результатів проектування в цьому випадку в основному визначається особистим досвідом, знаннями і уподобаннями розробників, а також має місце невоспроизводимость результатів, відсутність наступності, труднощі сприйняття і розуміння концепцій розробки різними колективами проектувальників.

SSADM (Structured System Analysis and Design Method) є яскравим прикладом втілення принципу проектування "зверху вниз" в технології створення складних ІУС. Він поєднує в собі простоту застосування системними аналітиками середньої кваліфікації, точність визначення результатів проектування, узгодженість з сучасними стандартами і методологією управління проектними роботами (PRINCE), гнучкість в застосуванні до проектування широкого класу систем для різних типів об'єктів, гарантії якості результатів проектування та спадкоємність різних версій проектів.

SSADM Structure



Постановка задачі

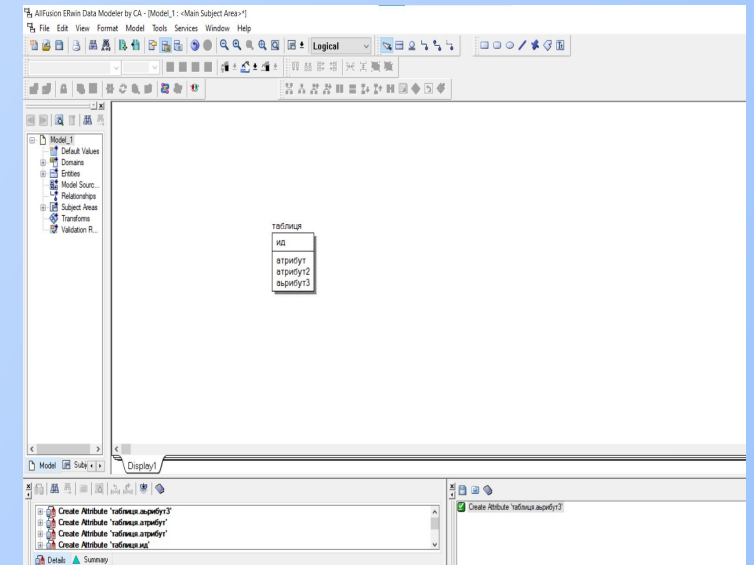
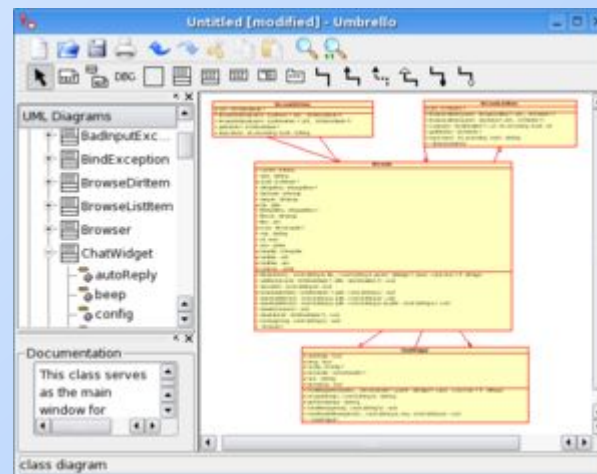
Завданням курсової роботи є аналіз, проектування і розробка системи логічного моделювання даних. Необхідно розглянути існуючі стандарти, а також спроектувати CASE-засіб, який дозволяє створювати ER-діаграми. *Спроектowana система має мати тривірневу архітектуру:*


- *рівень управління даними, що включає в себе репозитарій, а також забезпечує зберігання даних та доступ до них;*
- *прикладний рівень, який реалізує основну логіку (створення, валідація, збереження та завантаження схеми) і на якому здійснюється необхідна обробка інформації;*
- *рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;*

Вручну дуже важко розробити і графічно представити формальні специфікації системи, перевірити їх на повноту й несуперечність, і тим більше змінити. Якщо все ж таки вдається створити сувору систему проектних документів, то її переробка при появі серйозних змін складна та трудомістка.

Сучасні CASE-засоби охоплюють велику сферу підтримки численних технологій проектування інформаційних систем – від простих засобів аналізу і документування до повномасштабних засобів автоматизації, що покривають весь життєвий цикл програмного забезпечення (ПЗ).

CASE (англ. Computer-Aided Software Engineering) — набір інструментів і методів програмної інженерії для проектування програмного забезпечення, що допомагає забезпечити високу якість програм, відсутність помилок і простоту в обслуговуванні програмних продуктів.





Логічне моделювання даних призначено для побудови точної інформаційної моделі вимог до всієї проектованої системи в цілому або до окремих її частин.

Логічна модель даних:

- *допомагає аналітику зрозуміти предметну область системи за допомогою формалізації уявлень про неї;*
- *забезпечує досягнення взаєморозуміння між самими розробниками на ранніх стадіях проектування, що усуває багато проблем, які зазвичай виникають пізніше;*
- *служить основою для проектування файлів і бази даних, але при цьому інваріантна стосовно будь-яким спеціальним інструментальним засобам і методам фізичної організації даних;*
- *визначає термінологію, яка використовується згодом при складанні різних посібників для користувачів проектованої системи.*

Основними компонентами логічної моделі є:

- сутності;
- екземпляр сутності;
- атрибути сутності;
- ключ сутності;
- зв'язки між сутностями.

Сутність - це клас однотипних об'єктів, інформація про яких повинна бути врахована в моделі. *Кожна сутність повинна мати найменування, виражене іменником в однині.*



Екземпляр сутності - це конкретний представник даної сутності. *Наприклад, представником суті "Співробітник" може бути "Співробітник Іванов". Примірники сутностей повинні бути помітні, тобто суті повинні мати деякі властивості, унікальні для кожного екземпляра цієї сутності.*

Атрибут сутності - це іменована характеристика, що є деякою властивістю сутності. *Наприклад атрибут повинен бути вираженням іменником в однині (можливо, з характеризують прикметниками). Атрибути зображуються у межах прямокутника, що визначає сутність.*

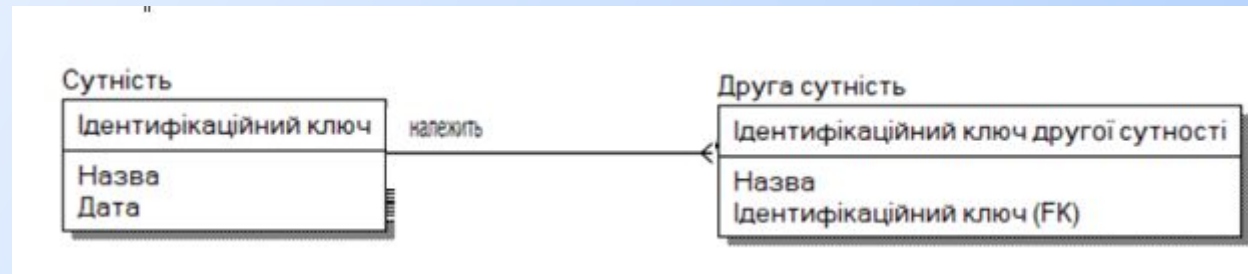
Ключ сутності - це набір атрибутів, значення яких є унікальними для кожного екземпляра сутності. *Сутність може мати кілька різних ключів (первинний, вторинні).*

Сутність

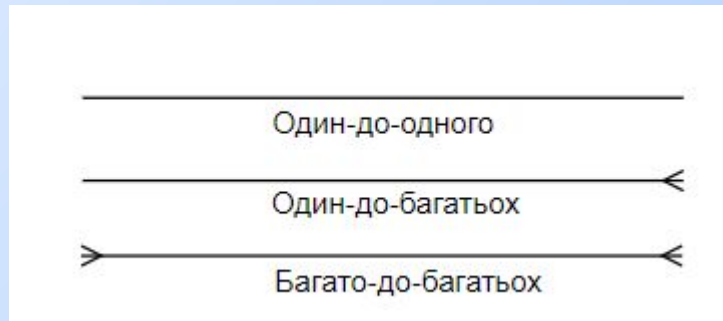
Ідентифікаційний ключ

Назва
Дата

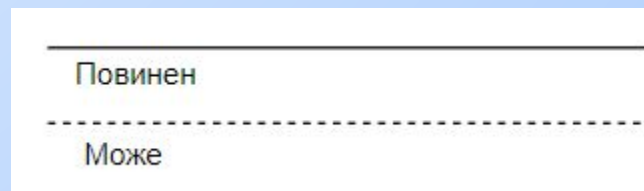
Зв'язок - це деяка асоціація між двома сутностями. Одна сутність може бути пов'язана з іншою сутністю або сама з собою. Зв'язки дозволяють по одній сутності знаходити інші сутності, пов'язані з нею. Графічно зв'язок зображується лінією, що з'єднує дві сутності.



Кожен зв'язок може мати тип зв'язку: один-до-одного, один-до-багатьох, багато-до-багатьох



Кожен зв'язок може мати одну з двох модальностей зв'язку – «може» або «повинен».



Нормалізація

У нормалізованій ЛМД об'єкти розглядаються як відносини, які повинні бути приведені до вигляду не нижче третьої нормальної форми (3NF).

Нормалізація бази даних — покроковий процес розбиття одного відношення відповідно до алгоритму нормалізації на декілька відношень на базі функціональних залежностей.

Фамилия и имя
Иванов Петр
Петров Иван
...



Фамилия	Имя
Иванов	Петр
Петров	Иван
	...

Код	Товар	Цена за тонну	Количество, тонн	Стоимость
1	Бананы	1200	10	12000
2	Киви	1500	20	30000
...				

Год	Бананы	Киви
2006	3200	1200
2007	1500	1500
...		



Год	Код товара	Кол-во
2006	1	1200
2007	2	1500
...		

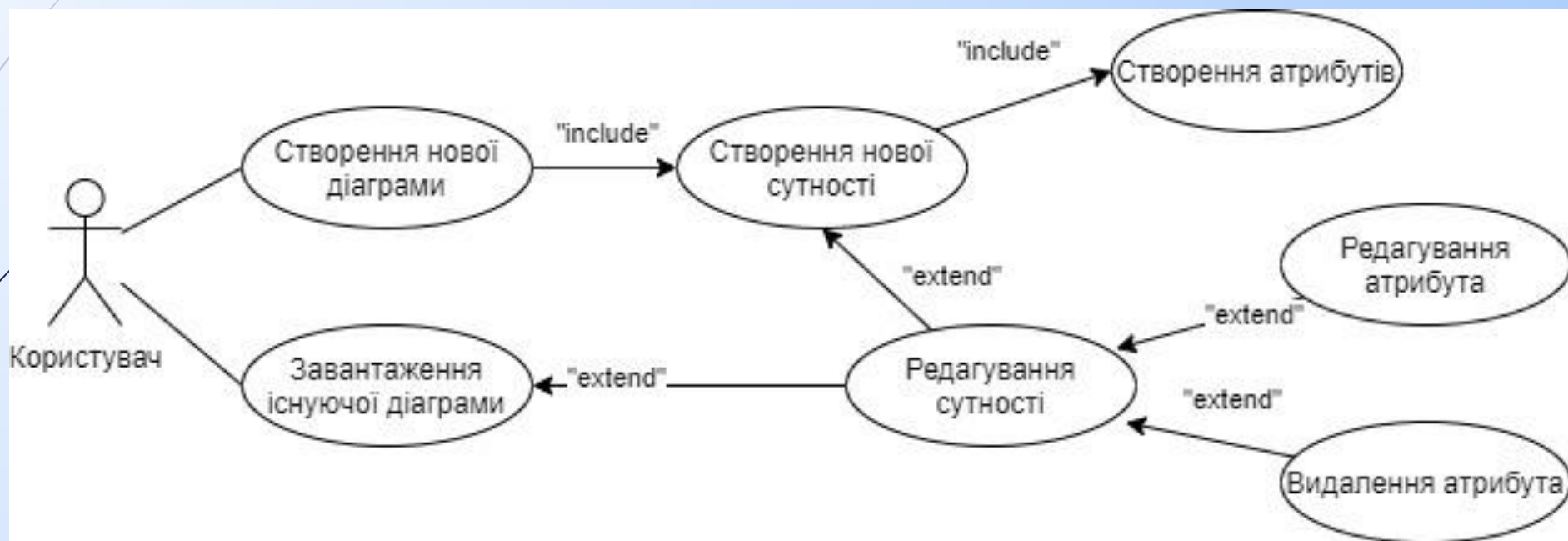
Код	Товар
1	Бананы
2	Киви
...	

∞ — 1 relationship line

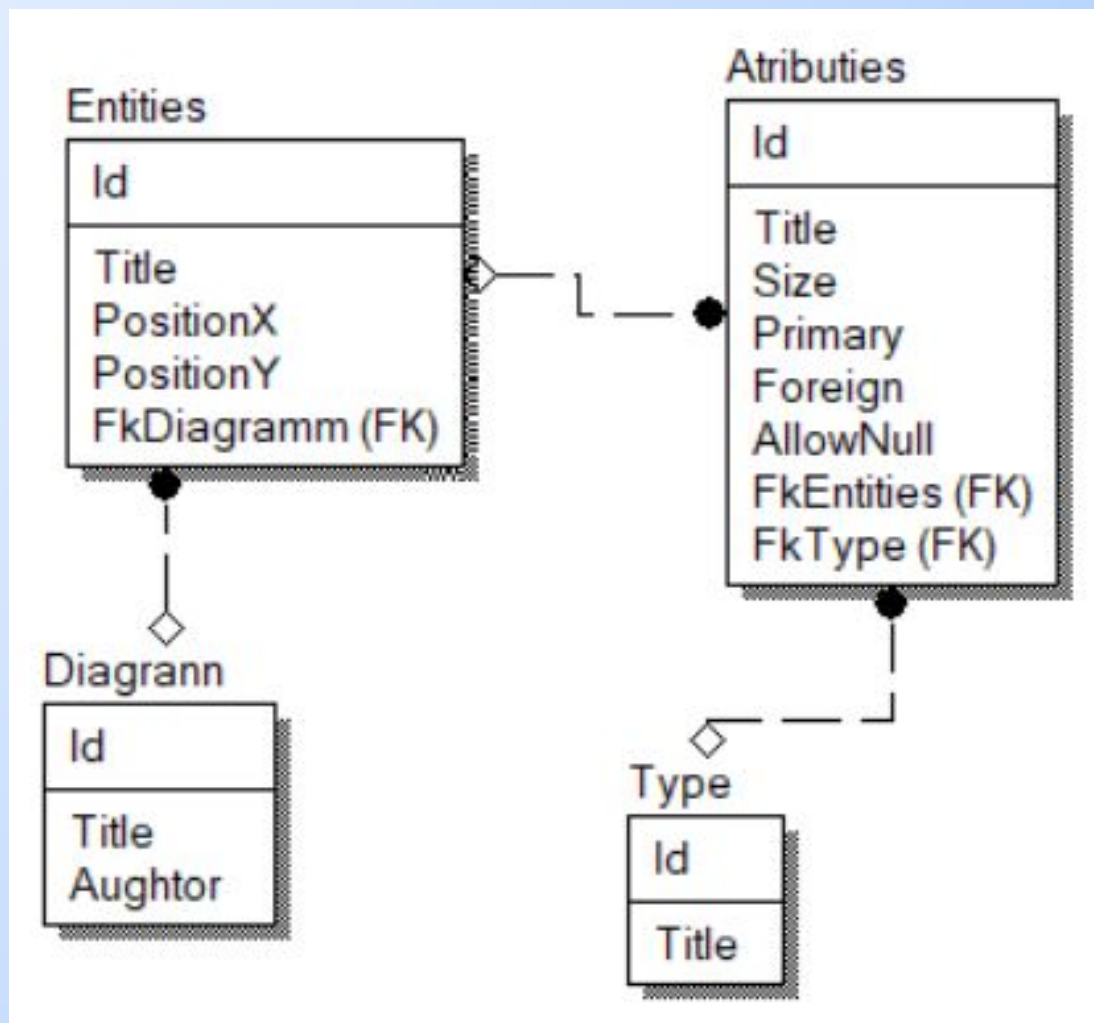
Розроблено
схему CASE- застосунку.



Загальна діаграма варіантів використання

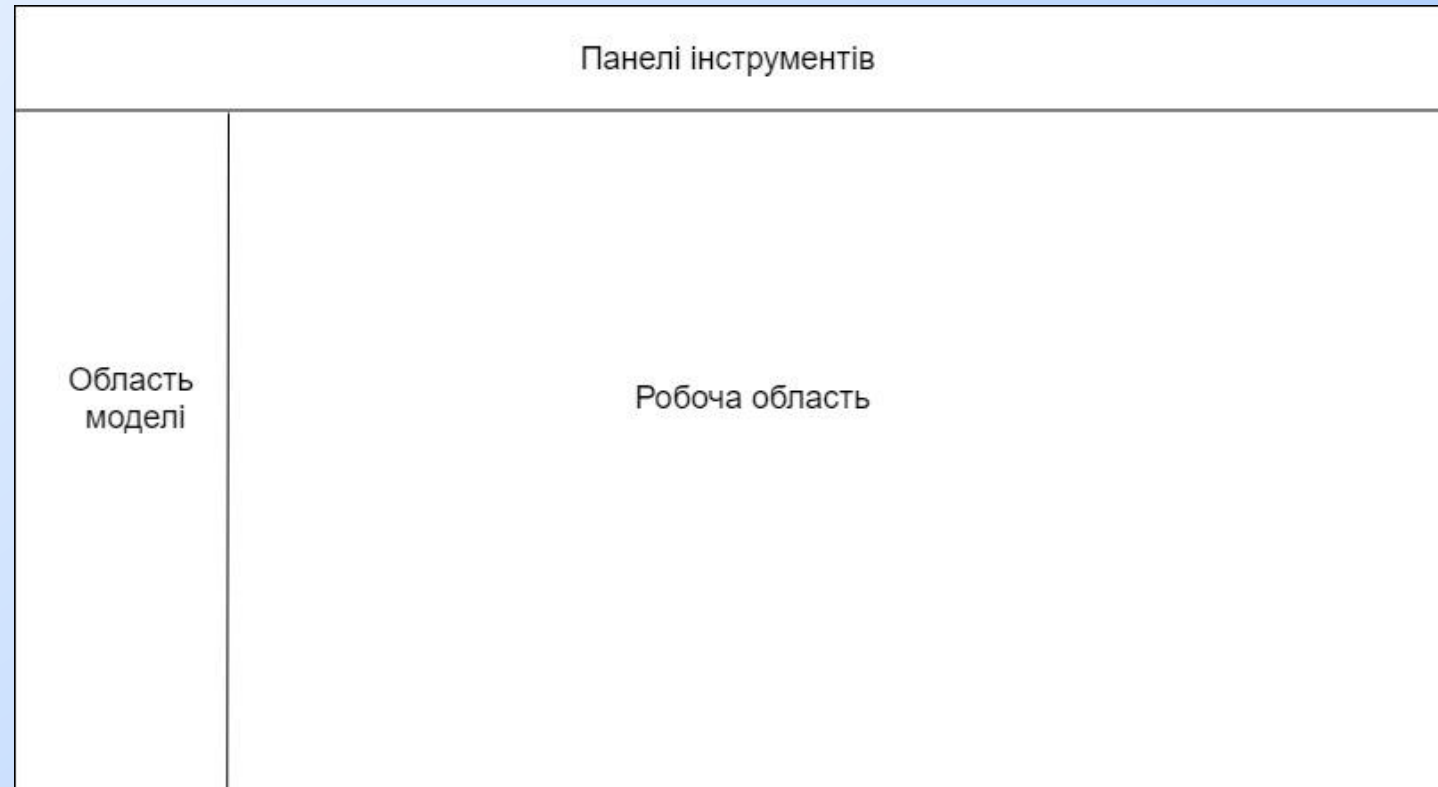


Відповідно до розробленої структури програми були виділені наступні основні сутності, які використовуються в середині програми



Дизайн інтерфейсу повинен починатися з людей. Його розроблюють для потреб людей, моделей поведінки і бажання. Є програми, бази даних, сервера, але користувач працює з інтерфейсом. Інтерфейс - це те, що бачить користувач, коли він працює з програмою.

Інтерфейс програми буде розроблений в інтуїтивно зрозумілій стилі.



Панель інструментів - містить кнопки для управління файлами (нова діаграма, відкрити, зберегти);

Область моделі містить назву моделі, всі сутності, їх атрибути.

Робоча область - містить власне розроблюєму модель. На ній є можливість створити нову сутність, а також редагувати створену сутність – додавати нові атрибути, редагувати або видаляти їх, видалити сутність.

Редагування кожного атрибута буде здійснюватися після натискання на відповідну кнопку праворуч від типу атрибута.

Назва діаграми

Сутність 1

Атрибут 1.1

Атрибут 1.2

Атрибут 1.3

Сутність 2

Атрибут 2.1

Атрибут 2.2

Атрибут 2.3

Додати сутність

Сутність			✕
?	Атрибут 1	int	☑ ✕
	Атрибут 2	string	☑ ✕
	Атрибут 3	int	☑ ✕
	Атрибут 4	bool	☑ ✕
Додати			

Атрибут 1

int

Назва

Тип

Розмір



Первинний ключ










Вторинний ключ


Нулеве значення

Оновити

Відмінити

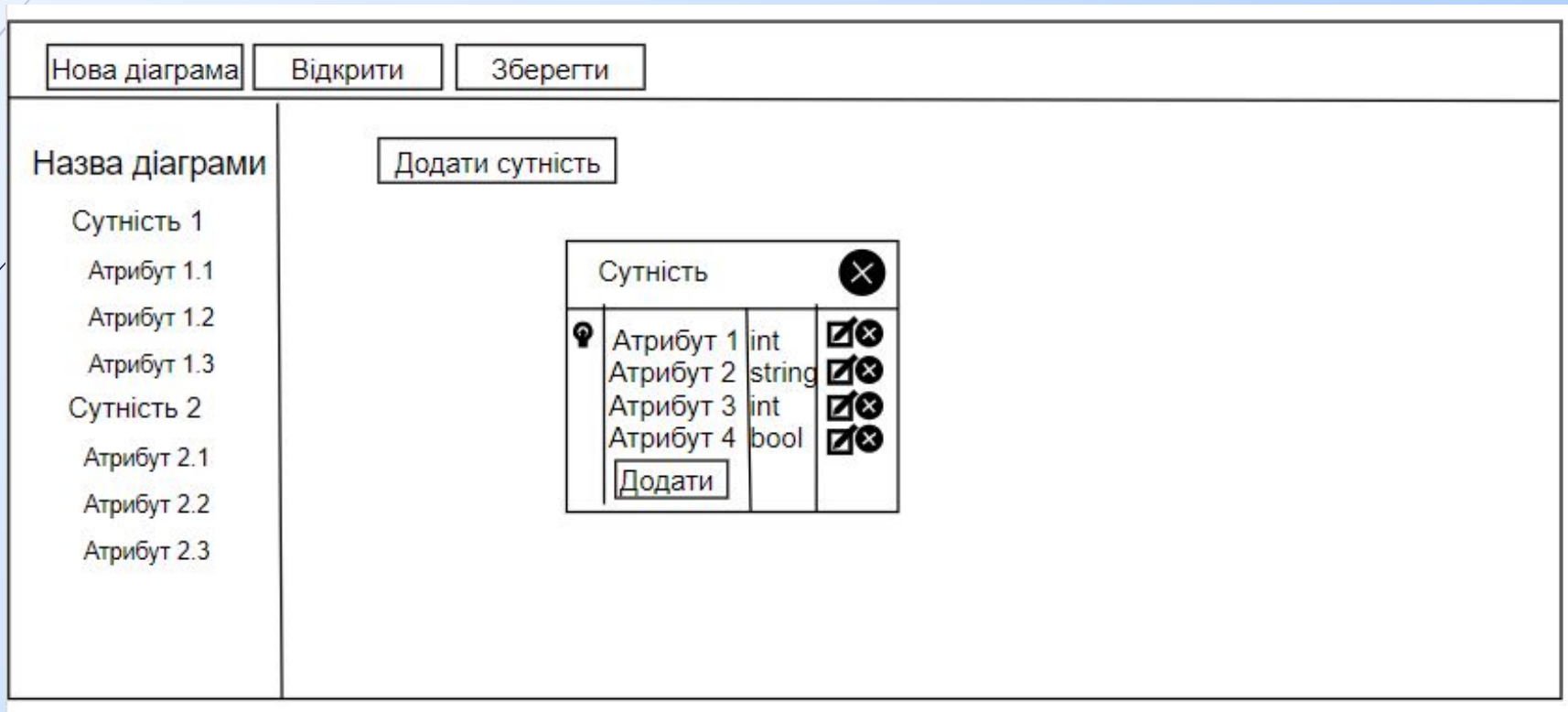
email	string		
Name	<input type="text" value="email"/>		
Type	<input type="text" value="string"/>		
Size	<input type="text"/>		
Default	<input type="text"/>		
<input type="checkbox"/> Primary Key	<input type="checkbox"/> Unique		
<input type="checkbox"/> Allow nulls	<input type="checkbox"/> Auto Increment		
<input type="checkbox"/> Foreign Key			
<input type="button" value="Update"/>	<input type="button" value="Cancel"/>	<input type="button" value="Delete"/>	

users			
id	integer		
email	string		
password	string		
name	string		
<input type="button" value="Add"/>			



Для того щоб додаток було незалежним від операційної системи і був доступним широкому колу користувачів, він буде реалізований у вигляді веб-додатку. Для реалізації серверної частини буде використаний мову програмування C#. Для реалізації клієнтської частини буде використаний HTML – мова гіпертекстової розмітки, CSS – мова опису стилів. Для забезпечення управління даними – СКБД MSSqlServer. Так, для використання CASE-застосунку необхідним є лише наявність сучасного браузера.

Результуючий прототип інтерфейсу програми.





Дякую за увагу!!