МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ

Розрахунково-графічна робота

з дисципліни: «*Основи системного аналізу*»

на тему: «**Система для тайм-менеджменту та сумісної роботи із вбудованим чатом**»

Виконав:
студент ІІІ курсу
групи ДА-91
Макрушин А.М.

Київ – 2022

Зміст

Вступ

Сучасні робочі процеси роботи над будь-якими проектами мають високу необхідність у структуризації, візуалізації та можливості декомпозувати глобальні задачі проекту, а особоливо коли робота ведеться певною групою працівників. Це є необхідним тому, що це фактично збільшує ефективність роботи над проектом та продуктивність співробітників. Саме тому на сьогоднішній день сучасні тікет-системи, або ж, якщо більш узагальнити, системи управління проектами, є незамінними в роботі.

Варто зазначити що така необіхдність йде разом із популярністю таких систем. І на це є велика кількість причин:

- візуалізації дає краще сприйняття процесів і, як наслідок, краще бачення загальної картини;
- забезпечує менеджерам та директорам можливість відслідковувати статистику виконання задач, таким чином оцінюючи ефективність команди, що може впливати на багато бізнес-аспектів;
- полегшує формування звітностей за виконаною роботою;
- на осноі отриманої інорфмації легше проводити аналітику та роботу над помилками, покращувати робочі процеси;
- декомпозиція задач керується давнім, але ефективним принципом “розділяй та володарюй” яка збільшує ефективність роботи;
- полегшує кооперацію в команді;
- полегшує передачу інформації між різними етапами роботи між колегами та ін.

Тим не менш, системи управління проектами є лише частиною необхідного набору програмного забезпечення дя повноцінної роботи, і особливо в умовах

віддаленої роботи. Для цього потрібні наступні види програмного забезпечення, за винятком вищезгаданого:

- мессенджери;
- програми для відео- та/або голосових онлайн коференцій;
- системи ведення документації;
- системи для відслідковування та створення подій та їх розкладу;
- більш специфічне ПО, залежно від галузі роботи компанії;

Якщо останній пункт є особливим для кожного випадку, то інші є небхідними для усіх випадків. І при цьому, в сучасному світі для кожного із перелічених видів програмного забезпечення часто використовуються різні сервіси, або системи. І це накладає додактове фінансове навантаження на компанції, оскільки легальна робота із продуктами вимагає покупку ліцензій цих продуктів.

Ідея представленої досліджуваної системи в даній розрахунково-графічній роботі є об’єднання багатьох видів програмного забезпечення в один. Таким чином, усі необхідні будь-якій компнії робочі процеси будуть виконуватись в одній системі. Плюси від такого ПО:

- збереження життєвоважливої інформації для роботи в одному додатку, що зменшує ймовірність її втрати при перемиканні між багатьма сервісами;
- збільшує кофморт роботи;
- дозволяє кастомізувати роботу;
- заощаджує кошти компаній на покупці багатьох сервісів.

В даній роботі буде не тільки представлено систему для ефективного тайм-менеджменту та сумісної роботи із вбудованим чатом, а ще її проаналізано та дано можливі прогнози розвитку.

Предмет дослідження в даній роботі

Опис предмету дослідження

Предметом дослідження є система — веб-додаток, для для сумісної роботи та особистого тайм-менеджменту, з відслідковуванням прогресу виконання задач, визначенням важливості завдання, його складності та відповідальної людини за його виконання, можливістю кастомізації робочого процесу, змінюючи кількість рядків, їх колір і назву, що дозволяє користувачам самостійно визначати свій робочий процес, використовуючи можливості нашої системи.

Досліджувана система може бути частиною корпоративних робочих процесів і бути використана як система для управління проектами всередині бізнесу, зовнішніми проектами, або ж задання вектору розвитку бізнесу, та його особистих, більш глобальних задач, із можливістю безпосередньої і швидкої комінукації за допомогою вбудованого чату.

Поняття досліджуваного предмету

Веб-додаток — система, що може бути відображена в будь-якому веб-бразуері, через який буде вестись взаємодія з користувачем, яка побудована на принципах веб-програмування та використання технологій мережі Інтернет та мережевих протоколів обміну інформацією.

Тайм-менеджмент — грамотне розподілення часу на певні види діяльності, вирішення задач, для його структурування та ефективного використання.

Кастомізація робочого процесу — налаштування середовища, залежно від власних потреб, необхідних речей та побажань, для забезпечення комфортної роботи на різних етапах виконання задач.

Чат — тип мережвеих додатків для миттєвого обміну повідомленнями, файлами для швидкої комунікації.

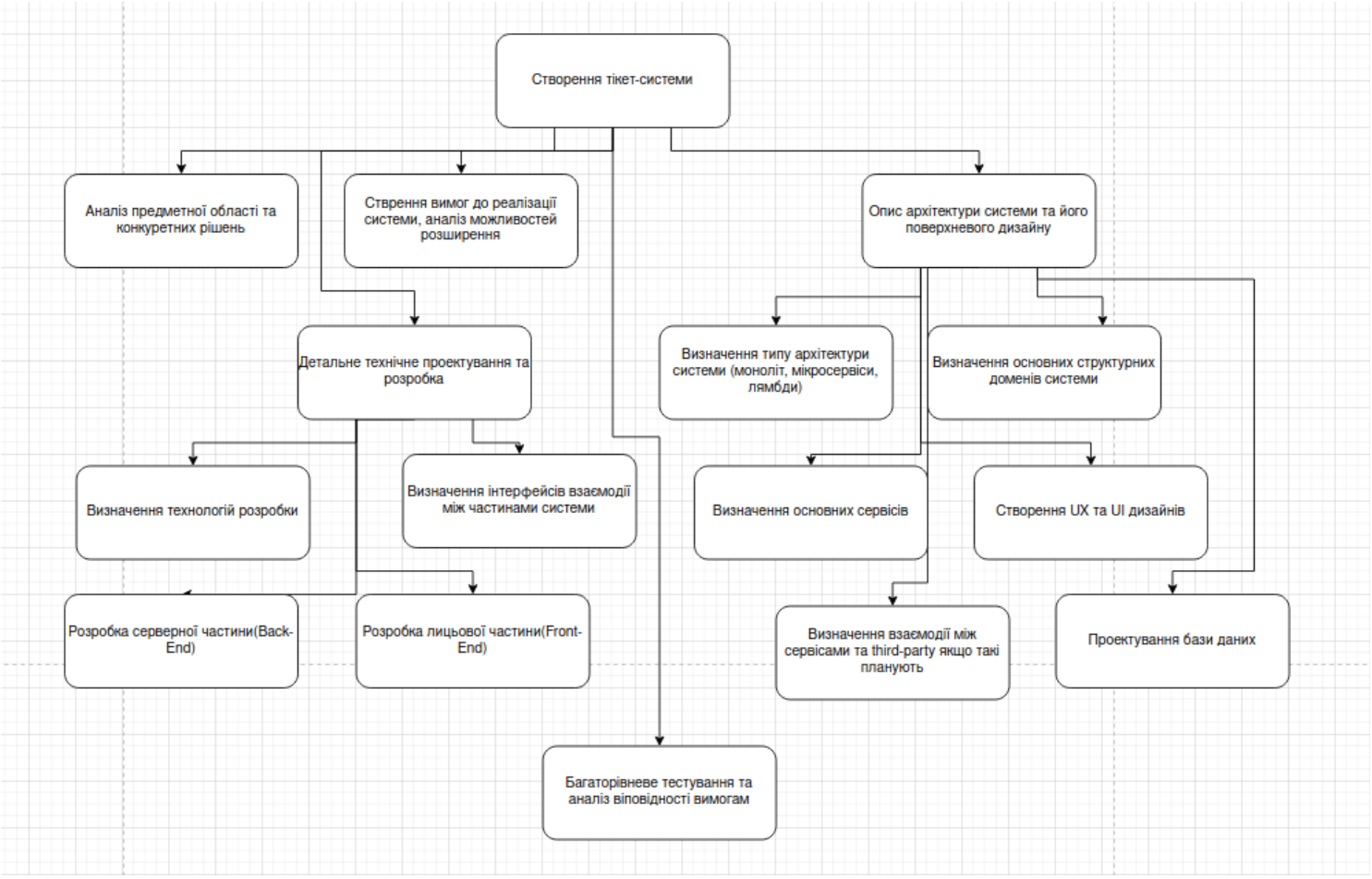
Показники досліджувального предмету

Відповідно до даного визначення досліджуваного предмету, його показниками будуть:

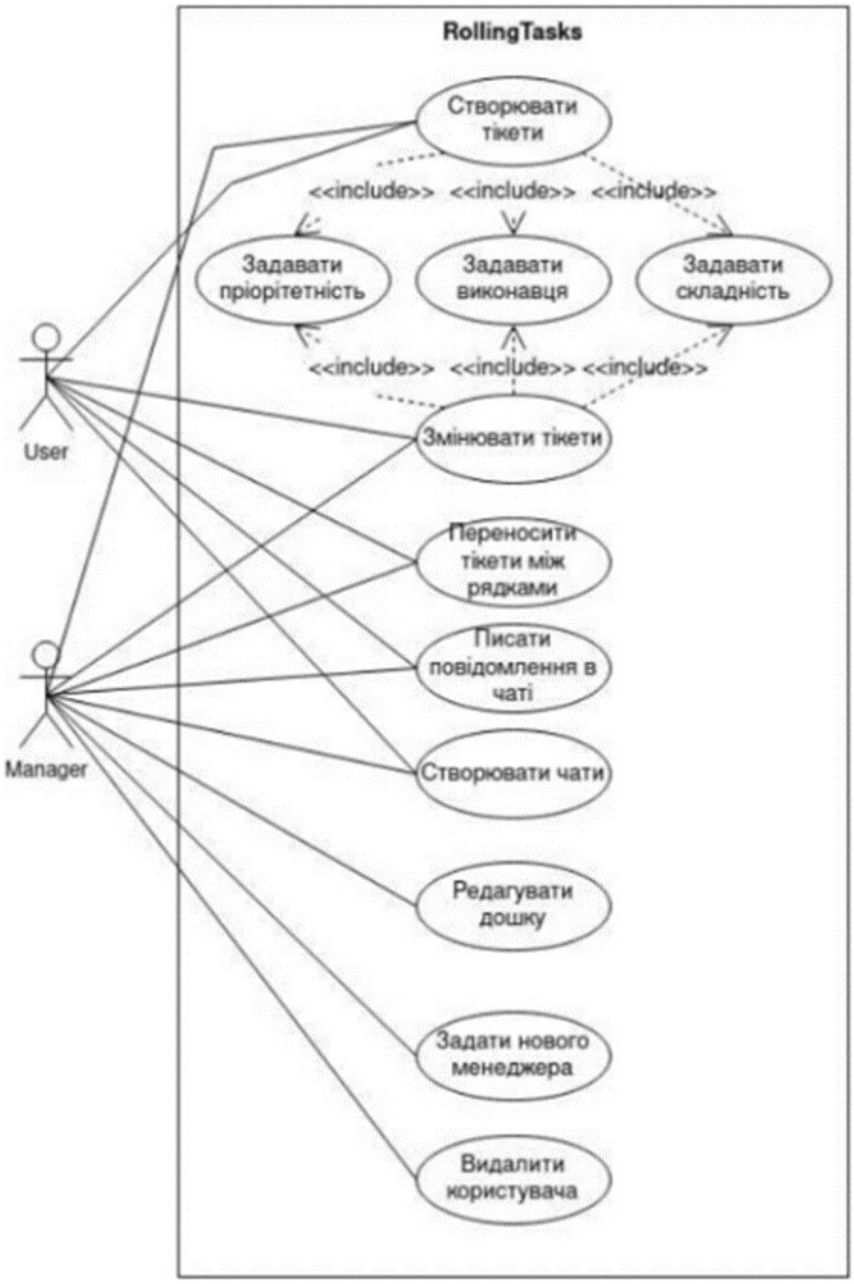
- комфорт робочих процесів;
- ступінь кастомізації;
- якість і швидкість комунікаці;
- ефективність управління задачами проектів;
- ступінь економії бізнесом коштів на зекономлених ліцензіях.

Дерево проблем

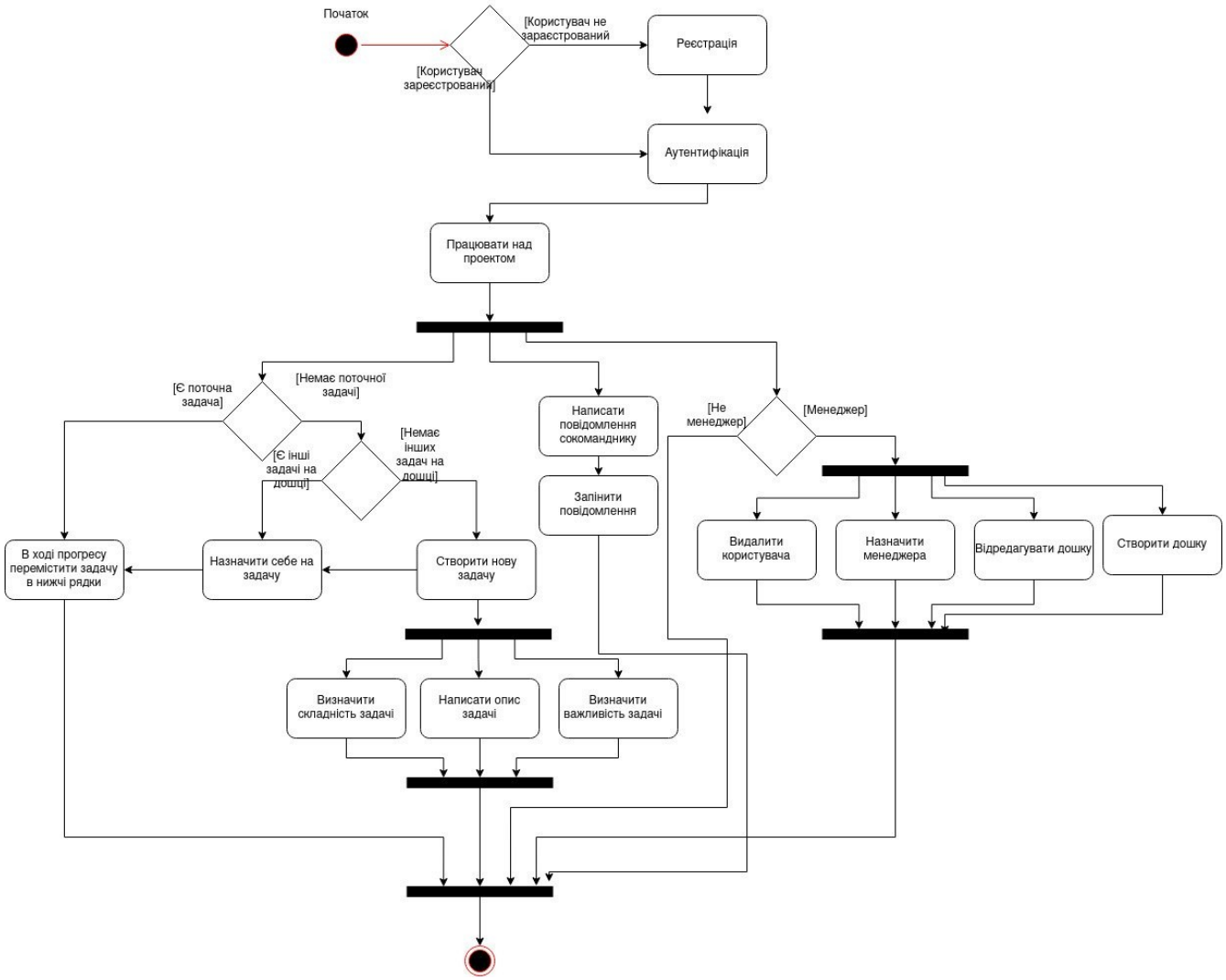
Дерево задач



Діаграма прецедентів

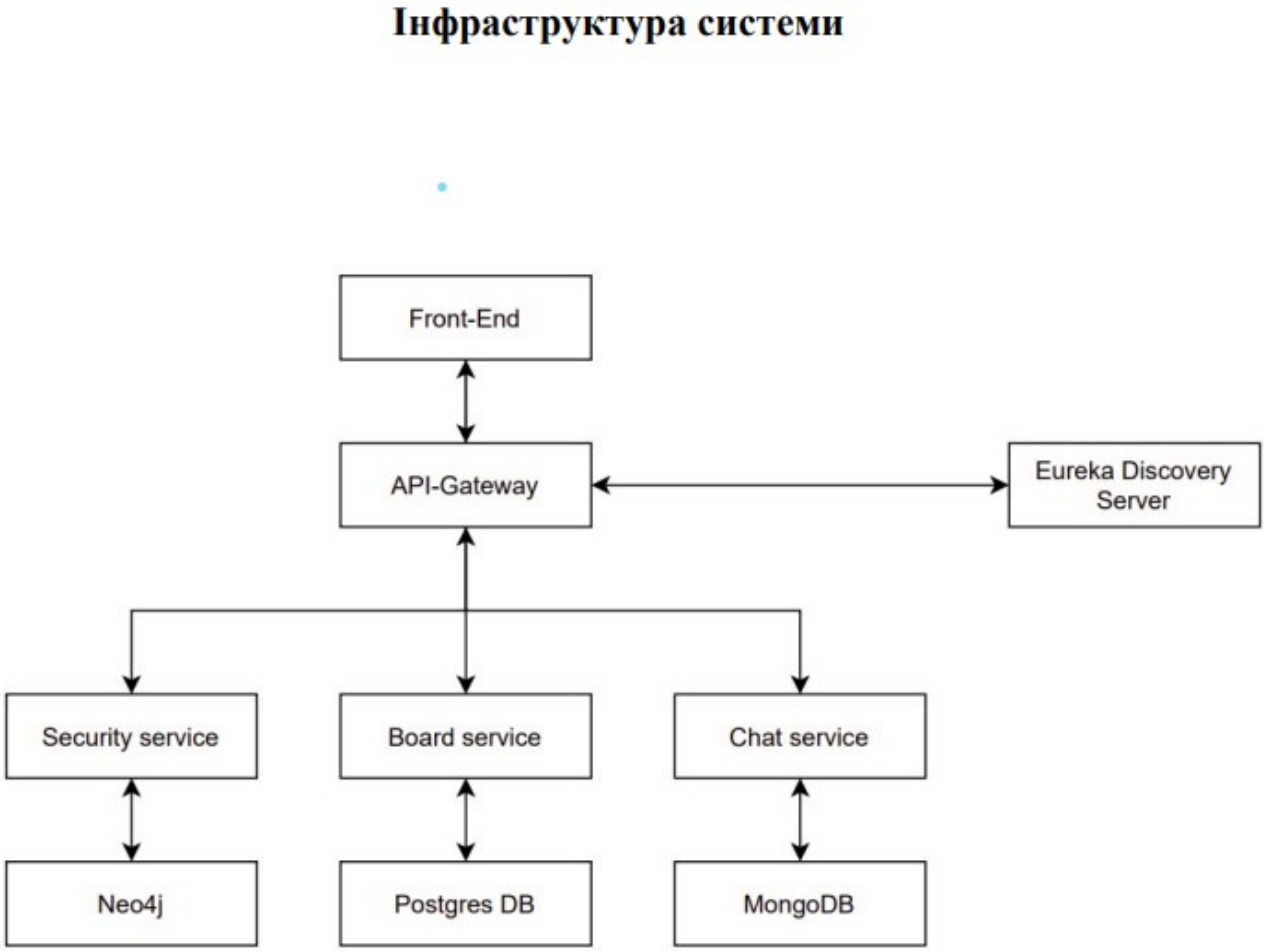


Діаграма діяльності



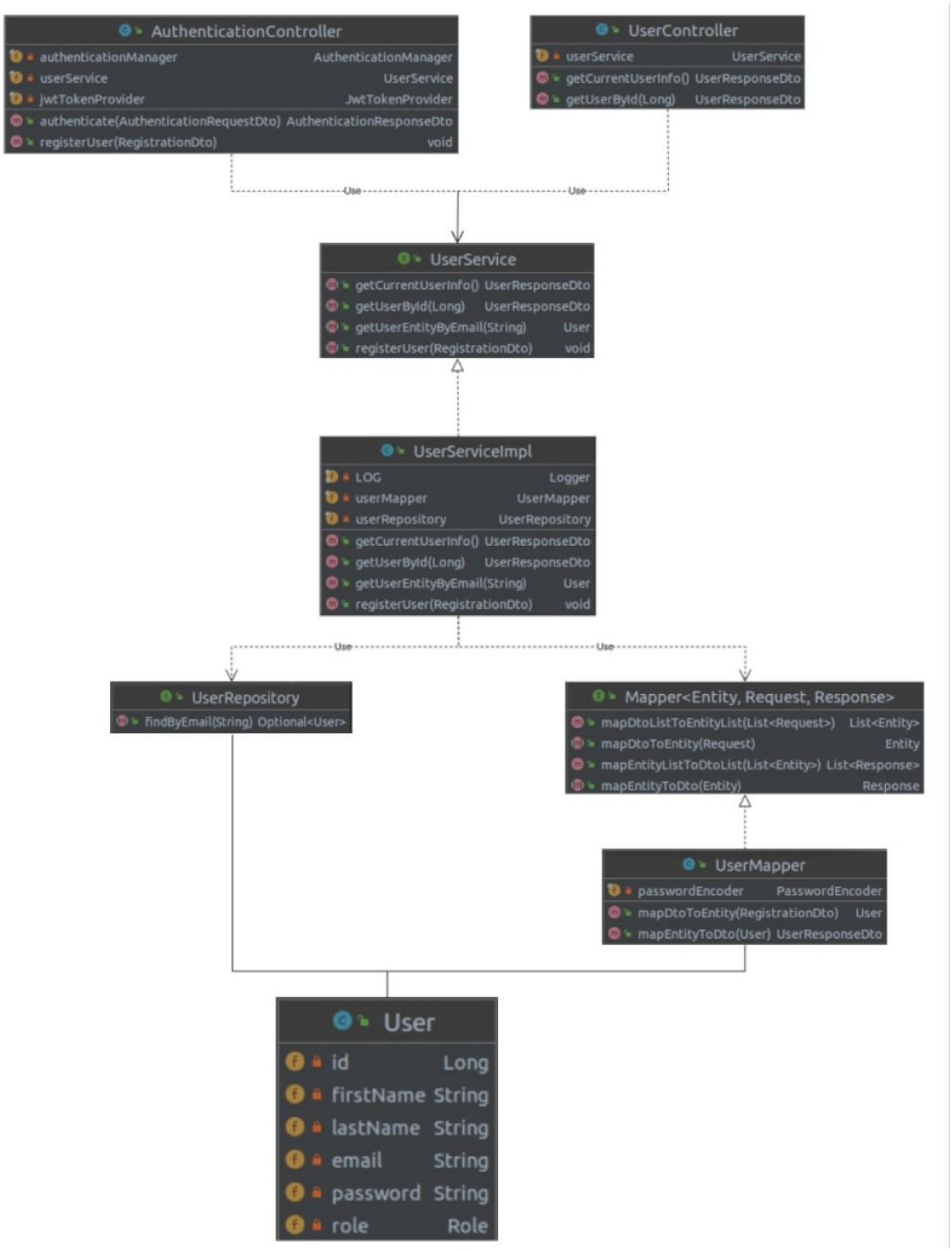
Діаграми класів

Побудована система має мікросервісну архітектуру. Відповідно до цього, система розділена по доменним вагомим частинам на User-service, Board-service та Chat-service. На наступному малюнку можна побачити детальніше структуру системи:

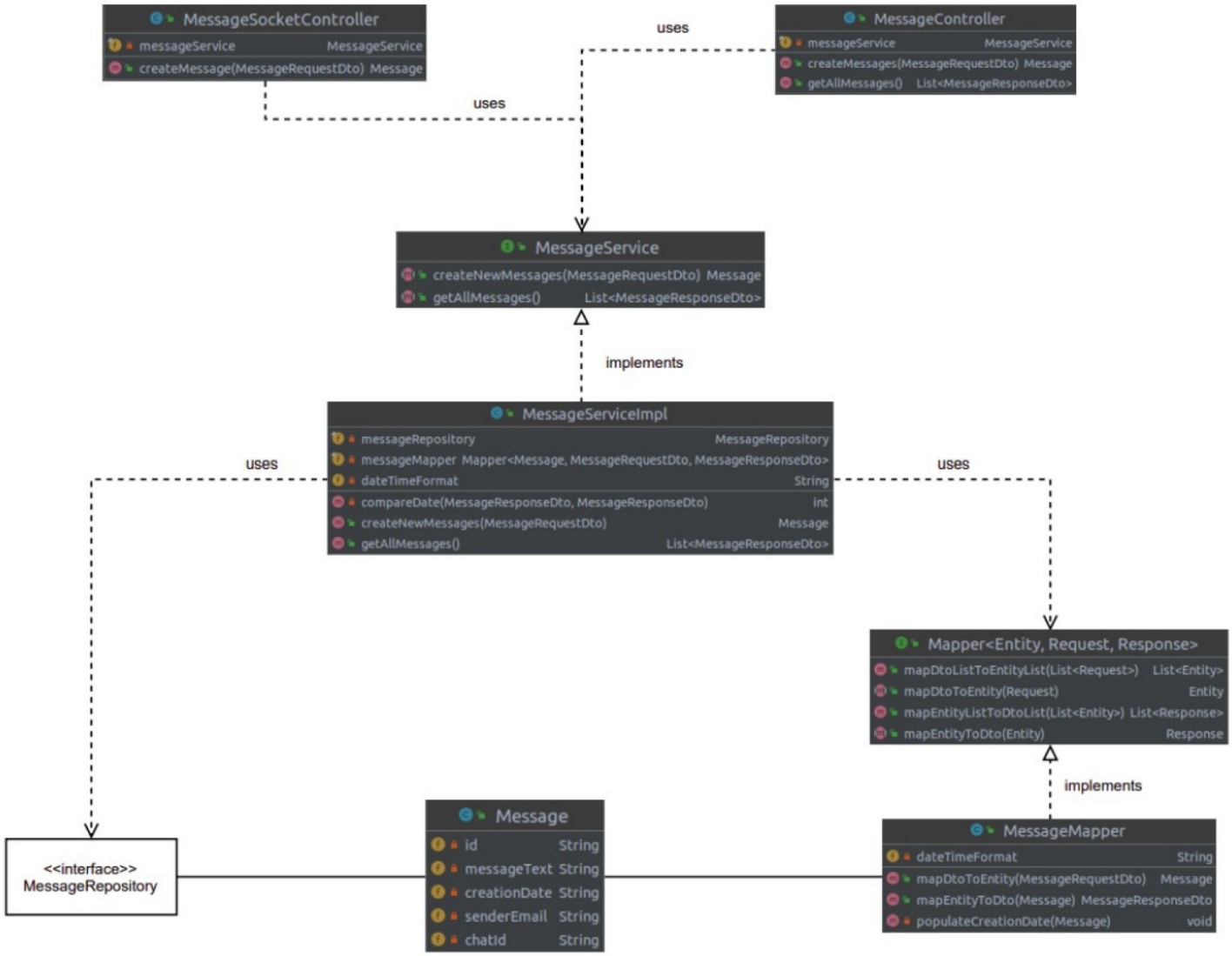


Діаграма класів Board-service

Діаграма класів Security-service



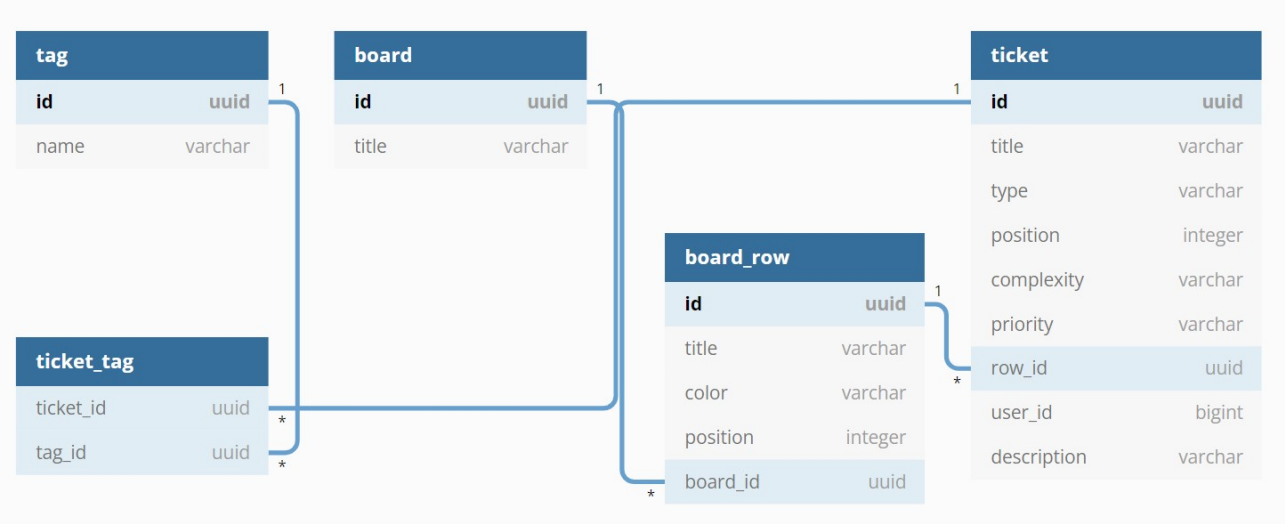
Діаграма класів Chat-service



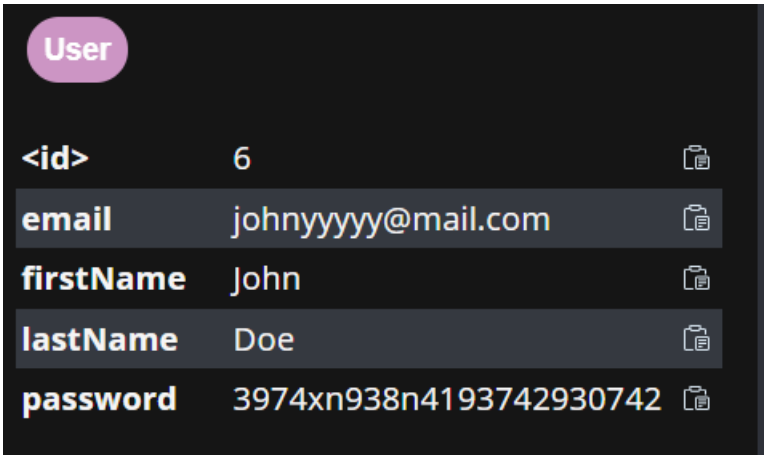
Структура баз даних

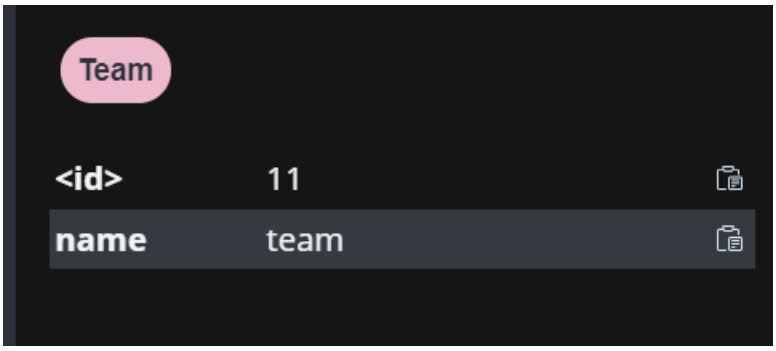
Оскільки система побудована в мікросервісній архітектурі, система бла спроектована відповідно до кращих практик мікросервісної архітектури, одна з яких каже: 1 база даних на 1 мікросервіс. Відповідно для кожного мікросервісу є окрема база даних з окремою структурою.

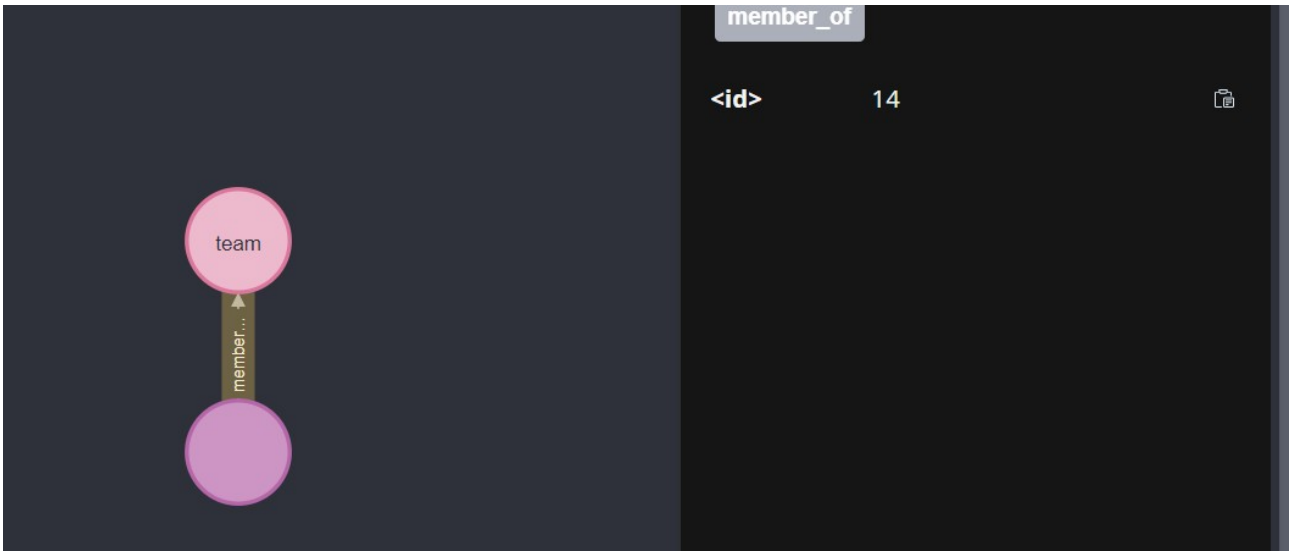
База даних Board-service — PostgreSQL



База даних Security-service — Neo4j







База даних Chat-service — MongoDB

```
{
  "id" : "string",
  "messageText" : "string",
  "creationDate" : "string",
  "email" : "string",
  "fullName" : "string"
}
```

db.message.find()

	{} _id	{} _class	{} creationDate	{} email	{} fullName	{} messageText
1	61a91f312458ab384d612965	com.webmu…	07/12/2021 19:48:43.394	akhmedovmagomed2001@gmail.com	Mahomed Akhmedov	Update messsage 2.0!
2	61a921abc36b14732babeaa5	com.webmu…	<unset>	akhmedovmagomed2001@gmail.com	Mahomed Akhmedov	Update message v.2
3	61a937b762039e4629db604f	com.webmu…	02/12/2021 23:16:39.904	<unset>	<unset>	Update message v.4.6
4	61a937cf62039e4629db6050	com.webmu…	02/12/2021 23:17:03.623	<unset>	<unset>	less go
5	61a937d662039e4629db6051	com.webmu…	02/12/2021 23:17:10.160	<unset>	<unset>	сщвштп
6	61a937df62039e4629db6052	com.webmu…	02/12/2021 23:17:19.611	<unset>	<unset>	coding*
7	61a937df62039e4629db6053	com.webmu…	02/12/2021 23:17:19.838	<unset>	<unset>	I don't get it
8	61a937e662039e4629db6054	com.webmu…	02/12/2021 23:17:26.008	<unset>	<unset>	and I
9	61a937fa62039e4629db6055	com.webmu…	02/12/2021 23:17:46.802	<unset>	<unset>	How was your day?
10	61a9380962039e4629db6056	com.webmu…	02/12/2021 23:18:01.540	<unset>	<unset>	It was so hard.... ;(
11	61a9381262039e4629db6057	com.webmu…	02/12/2021 23:18:10.141	<unset>	<unset>	And your?
12	61a9381962039e4629db6058	com.webmu…	02/12/2021 23:18:17.718	<unset>	<unset>	so so
13	61aa2294e0c2eb1187b45294	com.webmu…	03/12/2021 15:58:44.098	<unset>	<unset>	ssecurity
14	61ae1eb4e26499333bf0e5f3	com.webmu…	06/12/2021 16:31:16.230	test@email.com	<unset>	Auth! updated
15	61ae8a54f6a0486f506805a1	com.webmu…	07/12/2021 00:10:28.529	akhmedovmagomed2001@gmail.com	Mahomed Akhmedov	Auth! Username!
16	61af9dde5c3a52324fd67691	com.webmu…	07/12/2021 19:46:06.802	akhmedovmagomed2001@gmail.com	Mahomed Akhmedov	7 December message
17	61afa6864e1b033c4a3bc223	com.webmu…	07/12/2021 20:23:02.761	akhmedovmagomed2001@gmail.com	Mahomed Akhmedov	7 December message: re…
18	61be656fef634e10cb8bba0c	com.webmu…	19/12/2021 00:49:19.441	test@email.com	Test User	Hi
19	61be656fef634e10cb8bba0d	com.webmu…	19/12/2021 00:49:19.501	test@email.com	Test User	Hi
20	61be656fef634e10cb8bba0e	com.webmu…	19/12/2021 00:49:19.526	test@email.com	Test User	Hi

Можливі альтернативні рішення

В процесі проектування системи було вирішено будувати архітектуру веб-додатку за принципами мікросервісної архітектури з окремою базою даних для кожного мікросервісу. Тому альтернативами можна вважати інші види архітектури. Введемо 3 різі альтернативи архітектури систем:

- моноліт
- мікросервіси
- лямбди;

Також для оцінки альтернативних рішень будемо звертати увагу на швидкість розробки, а також необхідний час для доставки нової функціональності в систему при заданій архітектурі. Також введему необхідну суму для обслуговування сервера(ів) для кожної архітектури.

У випадку моноліта необхідно оплачувати постійний час роботи досить потужного сервера, оскільки усі операції в системі будуть виконуватись на одній машині. У випадку мікросервіса ми матимемо 3 маленьких не потужних сервера які будуть працювати постійно. У випадку лямбд, ми матимемо біля 10 дуже леггкихсерверів, які будуть працювати лише тоді, коли буде до них необхідне звернення.

Час на розробку на 1 місяць = 5 розробників * сер. з.п. 3000$ = 15000$

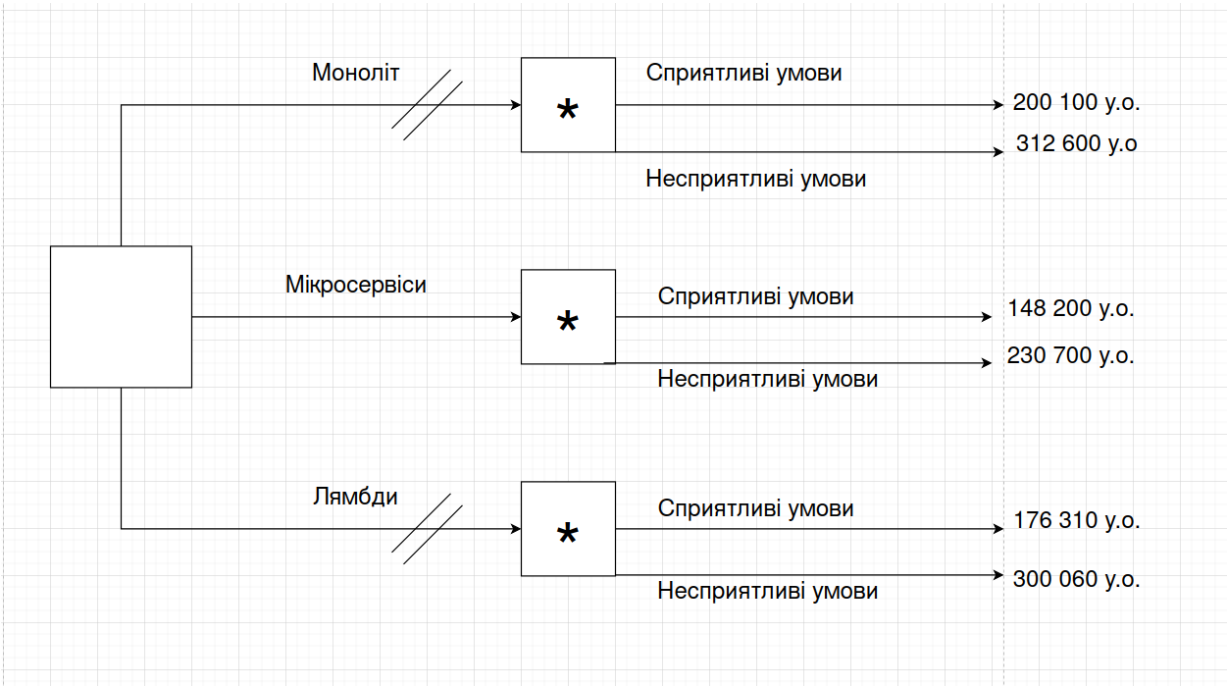
Оцінимо 1$ як 1 у.о

За несприятливих умов час на розробку збільшиться вдвічі.

Умова	Моноліт	Мікросервіси	Лямбди
Розробка в місяцях	3	4	6
Час роботи сервера	24	24	16

в день			
Ціна роботи сервера за годину	10$	2.5$ * 3	1.5$ * 6
Час на доставку нової функціональності	1.5 місяць	2 тижні	3 тижні

Оцінимо за даними параметрами розробку системи, а також її підтримку протягом року, при тому, що протягом року потрібно доставити 3 нові можливості в систему:



Можна побачити, що обраний варіант архітектури виявився найефективнішим з матеріальної точки зору

Прогноз розвитку

Ще з етапу проектування досліджуваної системи, її головною метою було об’єднання багатьох типів програмних додатків, або ж окремих систем, в одне ціле. Це є дуже корисним для того, щоб проводити усі робоч процеси в одному місці. Таким чином можна легко зберігати та обмініюватись інформацією з колегами, яка не зможе загубитися в численній кількості пересилань та копіювань з одного програмного додатку в інший. І одним із визначних аргументів того, чому дана система має право на життя і, більш того, може стати популярною, є те, що дана система не лише замінить кілька додатків, а ще й зменшить суму грошей, які компанії витрачають на число ліцензій цієї купи додатків. Саме те, що дана система є джерелом економії може привернути увагу, в першу чергу, тих компаній, які шукають способи зекономити, але при цьому не втрачаючи комфорт в забезпеченні необхідних бізнесу робочих процесів. Це можуть бути невеликі компанії, або ж стартапи які тільки починають свій шлях. З іншого боку, дана система може бути корисна для компаній, що відчувають брак інвестицій або недостатню кількість доходів. В такому випадку, економія на ліцензіях буде першим, що зумовить розповсюдження досліджувалної системи.

При цьому робота над покращенням системи не буде зупинятись, будуть додаватись нові функціональності, а за рахунок отримної певної кількості клієнтів, і відповідно, користувачів, можна збирати відгуки по тому що подобається, і що можна покращити. При швидкому регуванні на критику, можна заслужити певну довіру в клієнтів та їх задоволення системою. А один задоволений клієнт може бути причиною додавання більшої кількості клінтів за рахунок рекомендацій та появи системи в інформаційному просторі.

В такому випадку потрібно буде лише збільшувати темпи розробки на модифікації системи в тому ж шляху — заміна необхідних для робочих процесів додатків. На черзі можуть стати програми для відео конфереції, які за останні кілька років стали просто незамінними для роботи; можна буде рощвинутись в сторону кастомізації системи, що лише збільшить клієнтську базу, так як таким чином користувачі зможуть налаштовувати все під себе. В роботі важливим є створення подій та їх відслідковування в календарі, або ж ведення документації. Усе це можна буде об’єднати в еко-систему яка буде на льоту інтегруватись між собою. Таким чином можна буде дуже швидко розростися в клієнтській базі компаній, що зможуть замість 3-5 ліцензій мати лише 1, яка буде покривати усі необхідні випадки користування та необіхдні робочи процеси.

Висновок

В ході даного дослідження було детально розглянуто та проаналізовано систему Rolling Tasks, яка є тікет системою для тайм менеджменту, кооперації та комунікації.

Під час проведеного аналізу було визначено поняття та зміст досліджуваної системи та предметної області, її показники та сформовано її загальний опис. На основі встановлених знань, бли побудовано кілька графіних моделей відповідно до функціонального сенсу досліджуваної системи. Було встановлено для чого потрібна ця система, які проблеми вирішує та яка мотивація для її створення. Цю інформацію було зображено на дереві проблем. Відповідно нього було сформовано задачі, які має виконувати дана система, як вирішення вище згаданих проблем. Кроки для їх забезпечення були сформовані на основі аналізу предметної області, а також існуючих конкуретних рішень, виявляючи недоліки та необхідні аспекти, які треба імлементувати в систему. Після цього було сформовано дерево задач, в якому описано необхідні кроки, або ж задачі, що необіхдно було виконати для створення системи, для задоволення і розв’язання знайдених проблем.

Важливим етапом в аналізі системи було створення UML діаграми послідовностей (use case diagram) в якій можна побачити усі можливі дії користувачів (як рядового користувача, так і менеджера) в наведеній системі, а також було спроектовано діаграму діяльності, де вищезгадані дії були описані вигляді кроків та послідовностей дій в цій системі. Таким чином, на основі даних діаграм, можна отримати загальне представлення функціональності системи та к нею користуватись, які проблеми вирішує.

Окрім цього, було проведено доменний аналіз системи, на основі якого сформовано діаграму класів, в яких зображено принципи взаємодії всередині

системи та їх відповідальностей в ній, в тому числі із базами даних, структури яких також було наведено.

Так як під час проектування систем необіхдно виконувати і приймати багато різного роду рішень, сиестема може бути побудована по іншому, якщо на певних етапах проектування прийти до альтернативних рішень. Ця проблема також було проаналізована в даній роботі на прикладі аналізу альтернативи представленій мікросервісній архітектурі. В результаті чого було отримано, що рішення будувати архітектуру системи виявилось не тільки технологічно ефективним, а ще й матеріально вигіднішим.

Наприкінці було сформовано власний прогноз для розвитку даної системи і , на мою думку, він є дуже втішним та позитивним.