

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
Завідувач кафедри  
\_\_\_\_\_ Скарга-Бандурова І.С.  
« \_\_\_\_ » \_\_\_\_\_ 2018 р.

**ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА**

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

НА ТЕМУ:

Розробка Android додатку для статистичної обробки інформації

Освітньо-кваліфікаційний рівень “бакалавр”  
Напрямок підготовки 6.050102 – “Комп’ютерна інженерія”

Керівник проекту:

\_\_\_\_\_

(підпис)

Сафонова С.О.

\_\_\_\_\_

(ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_

(підпис)

Критська Я.О.

\_\_\_\_\_

(ініціали, прізвище)

Здобувач вищої освіти:

\_\_\_\_\_

(підпис)

Бережний В.М.

\_\_\_\_\_

(ініціали, прізвище)

Група:

\_\_\_\_\_

КІ-14з

Севєродонецьк 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки  
Кафедра Комп'ютерних наук та інженерії  
Освітньо-кваліфікаційний рівень бакалавр  
Напрямок підготовки 6.050102 Комп'ютерна інженерія  
(шифр і назва)  
Спеціальність \_\_\_\_\_  
(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Завідувач кафедри \_\_\_\_\_  
І.С. Скарга-Бандурова  
« \_\_\_\_\_ » \_\_\_\_\_ 2018р.

**З А В Д А Н Н Я  
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Бережному Владиславу Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка Android додатку для статистичної обробки інформації

керівник проекту (роботи) Сафонова С.О., к.т.н., доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " 14 " 05 2018р. № \_\_\_\_\_

2. Термін подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної області і постановка задачі. Аналіз існуючих систем управління проектами. Опис основних функцій системи. Вибір середовища розробки. Програмна реалізація розроблених алгоритмів. Охорона праці та безпека в надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Електронні плакати

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст.викл. кафедри КНІ Критська Я.О.		

7. Дата видачі завдання \_\_\_\_\_

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Огляд літератури з теми ДП і постановка задачі	14.05.18-19.05.18	
2	Аналіз існуючих систем управління проектами	20.05.18-25.05.18	
3	Опис основних функцій системи	26.05.18-02.06.18	
4	Вибір середовища розробки	03.06.18-06.06.18	
5	Програмна реалізація розроблених алгоритмів	07.06.18-09.06.18	
6	Розробка розділу охорона праці	10.06.18-12.06.18	
7	Оформлення електронних плакатів	13.06.18-15.06.18	
8	Оформлення пояснювальної записки	14.05.18-15.05.18	

Здобувач вищої освіти

\_\_\_\_\_ ( підпис )

Керівник

\_\_\_\_\_ ( підпис )

Бережний В.М.

\_\_\_\_\_ (прізвище та ініціали)

Сафонова С.О.

\_\_\_\_\_ (прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту (роботи) бакалавра: 76 с., 36 рис., 5 табл., 29 бібліографічних джерел посилань, 3 додатки.

Об'єкт розробки: питання статистичної обробки інформації мобільним додатком для операційної системи Android.

Мета роботи: розробка мобільного додатку з використанням статистичних методів для слідкування за прогресом власної роботи над проектами та внесення у них незначних змін.

В проекті виконано:

1. Аналіз існуючих систем управління проектами та сформульоване технічне завдання дипломного проекту.
2. Опис основних функцій системи.
3. Вибір середовища розробки.
4. Програмну реалізацію розроблених алгоритмів.
5. Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкта, що впливають на персонал.

Отримано наступні результати: тестування показало задовільну якість та точність отриманих результатів, продемонструвало працездатність готового програмного продукту та його відповідність меті дипломної роботи. Мобільний додаток після незначних вдосконалень може бути у майбутньому задіяний у роботі над проектами.

Практичне значення, галузь застосування роботи: мобільний додаток дозволить досить зручно слідкувати за прогресом власної роботи над проектами та збільшити ефективність праці.

**Ключові слова:** SCRUM, СТАТИСТИКА, ДІАГРАМА, ANDROID, JAVA, ANDROID SDK, ECLIPSE

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А, м. Сєвєродонецьк, 93400.

## ЗМІСТ

Вступ .....	6
1 Аналіз предметної області .....	7
1.1 SCRUM.....	7
1.1.1 Визначення та історія виникнення .....	7
1.1.2 Scrum –процеси .....	8
1.1.3 Дійові особи .....	8
1.1.4 Основні поняття .....	10
1.1.5 Типи нарад .....	11
1.2 Існуючі системи управління проектами .....	13
1.2.1 Jira .....	13
1.2.2 TeamBridge .....	16
1.2.3 Zoho Projects .....	17
1.2.4 Comindware Project .....	19
1.2.5 Yodiz .....	21
1.3 Математична статистика як розділ статистики .....	23
1.4 Постановка завдання .....	24
2 Опис основних функцій системи .....	25
2.1 Побудова різних типів діаграм .....	25
2.1.1 Полігон частот .....	25
2.1.2 Кругова діаграма .....	26
2.1.3 Гістограма .....	27
2.1.4 Діаграма Ганта .....	28
2.2 Принципи побудови діаграм розроблюваного додатку .....	29
2.3 Майбутні функції .....	30
3 Програмна реалізація розроблених алгоритмів .....	31
3.1 Вибір середовища розробки .....	31

3.2 Архітектура системи .....	31
3.3 Основні технології .....	34
3.3.1 Програмний фреймворк .....	34
3.3.2 SQLite .....	36
3.3.3 Apache HttpClient .....	37
3.3.4 Багатопоточність у Android додатках .....	38
3.4 Існуючий аналог розроблюваної системи .....	39
3.5 Опис реалізації .....	41
3.6 Опис тестового матеріалу .....	42
3.7 Ілюстрації роботи програми .....	42
4 Охорона праці .....	46
4.1 Загальні питання з охорони праці .....	46
4.2 Аналіз стану умов праці .....	47
4.3 Виробнича санітарія .....	49
4.4 Гігієнічні вимоги до параметрів виробничого середовища .....	52
Висновки .....	61
Перелік джерел посилань.....	62
Додаток А Комп'ютерна презентація.....	65
Додаток Б Лістинг програми HibernateUtil.java.....	70
Додаток В Лістинг програми IndexBean.java.....	70

## ВСТУП

Обсяг світового ринку мобільних гаджетів постійно зростає. Однією з основних мобільних операційних систем є система Android. Мобільні додатки на цій операційній системі займають усе значніше місце у повсякденному житті людини, адже вони дозволяють досить зручно вирішувати різноманітні повсякденні завдання та можуть поліпшити трудову діяльність. Широке розповсюдження мобільного та бездротового інтернету лише сприяє зростанню користі від мобільних додатків, так як існує безліч сфер їх застосування.

Статистика вивчає методи кількісного охоплення і дослідження масових, зокрема суспільних явищ і процесів [1]. Збирання інформації про них сягає найдавніших часів. Її застосування може значною мірою покращити розуміння багатьох аспектів розглядуваної інформації, що у сучасному суспільстві набуває ще більшого значення. У поєднанні з можливостями, які надає користування мобільними додатками, статистична інформація стає дуже зручним інструментом.

Метою даної дипломної роботи є розробка мобільного додатку з використанням статистичних методів, за допомогою якого можна буде ефективно слідкувати за прогресом власної роботи над проектами та вносити у них незначні зміни.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 SCRUM

### 1.1.1 Визначення та історія виникнення

Scrum — методологія управління проектами для гнучкої розробки програмного забезпечення. Скрам чітко робить акцент на якісному контролі процесу розробки [2].

Підхід вперше описали Гіротака Такеучі та Ікуджіро Нонака в статті *The New New Product Development Game* (Гарвардський Діловий Огляд, 1986). Вони відзначили, що проекти, над якими працюють невеликі, крос-функціональні команди, зазвичай систематично продукують кращі результати, і пояснили це, як «підхід регбі». У 1991 році ДеГрейс та Шталь у книжці «Злі проблеми, справедливі рішення» [3] послалися на цей підхід, як на Scrum (сутичка навколо м'яча (у регбі)), спортивний термін, згаданий в статті Такеучі і Нонака. Кен Швабер на початку 1990-х використовував підхід, який привів Scrum в його компанію. Вперше метод Scrum було представлено на загальний огляд задокументованим, чітко сформульованим та описаним спільно Сазерлендом та Швабером на OOPSLA'96 в Остіні. Швабер та Сазерленд протягом наступних років працювали разом, щоб обробити та описати весь їхній досвід та найкращі практичні зразки для індустрії в одне ціле, в ту методологію, що відома сьогодні як Scrum. Швабер об'єднав зусилля з Майком Бідлом в 2001, щоб детально описати метод в книжці *Agile Software Development with SCRUM* [4]. Не зважаючи на те, що Scrum нарекли долю управління проектами з розробки ПЗ, він може також використовуватися в роботі команд обслуговувань програмного забезпечення (software maintenance teams) або як підхід управління розробкою і супроводом програм: Scrum of Scrums.



### 1.1.2 Scrum -процеси

Scrum — це кістяк процесу, який включає набір методів і попередньо визначених ролей. Головні дійові особи:

- ScrumMaster, той хто опікується процесами, веде їх і працює як керівник проекту.
- Власник Продукту, людина, що представляє інтереси кінцевих користувачів та інших зацікавлених в продукті сторін.
- Команда, яка включає розробників.

Протягом кожного спринту, 15-30 денного періоду (тривалість визначається командою), працівники створюють функціональний ріст програмного забезпечення.

Набір можливостей, які імплементуються кожного спринту, приходять з етапу, що має назву product backlog (документація запитів на виконання робіт). Він має найвищу пріоритетність за рівнем вимог до роботи, яка повинна бути виконана. Запити на виконання робіт (backlog items), що визначені протягом наради з планування спринту (sprint planning meeting), переміщуються в етап спринту. Протягом цієї наради Власник Продукту інформує про завдання, які він хоче, аби були виконані. Тоді Команда визначає скільки з бажаного вони можуть зробити, щоб завершити необхідні частини протягом наступного спринту [5]. Протягом спринту команда виконує визначений фіксований список завдань (backlog items). Впродовж цього періоду ніхто не має права змінювати перелік запитів на виконання робіт, що слід розуміти, як заморожування вимог (requirements) протягом спринту.

### 1.1.3 Дійові особи

За методикою Scrum у виробничому процесі є визначені ролі, що розбиті на дві групи — «свиней» та «курей». Ці назви використані через жарт про свиню та курку.

«Свині» використовуються для побудови продукту регулярно і часто (повністю задіяні), тоді як будь-які інші — «кури», ті, що зацікавлені (і задіяні) в проекті, але не мають прямого відношення до приготування страви. Потреби, бажання, ідеї та вплив курей беруться до уваги, але їм не завжди дозволяють прямо впливати, видозмінювати або включатися в хід Scrum проекту.

«Свині» цілком задіяні в проекті, до них відносяться:

- власник Продукту (Product Owner) — представляє інтереси кінцевих користувачів та інших зацікавлених у продукті сторін;
- керівник (ScrumMaster) — проводить наради (Scrum meetings) стежить за дотриманням усіх принципів скрам, вирішує протиріччя та захищає команду від відволікаючих факторів. Ця роль не припускає нічого іншого окрім коректного ведення скрам-процесу. Керівник проекту більш відноситься до власника проекту й не повинен фігурувати у якості скрам-мастера;
- команда (Scrum Team) — крос-функціональна команда розробників проекту, яка складається з спеціалістів різних профілей: тестувальників, архітекторів, аналітиків, програмістів и т. д. Розмір команди бажано повинен складати  $7 \pm 2$  чоловік. Команда є єдиним повністю задіяним учасником розробки й відповідає за результат як єдине ціле. Нихто крім команди не може втручатися у процес розробки за час спринту.

До «курей» відносяться:

- користувачі (Users);
- клієнти, продавці (Stakeholders) — особи, які ініціюють проект і для кого проект буде приносити вигоду. Вони задіяні в скрам тільки під час оглядової наради по спринту (Sprint Review);
- експерти-консультанти (Consulting Experts).

### 1.1.4 Основні поняття

Product backlog — це документ, який має список вимог до функціональності, які упорядковані згідно зі ступенем важливості. Product backlog представляє список того, що повинно бути реалізовано. Елементи цього списку називаються «історіями» (user story) або елементами backlog-у (backlog items). Product backlog відкритий для редагування усім учасникам Scrum-процесу [6].

Sprint backlog — містить функціональність, обрану Product Owner із Product Backlog. Всі функції розбиті по задачах, кожна з яких оцінюється командою. Кожен день команда оцінює об'єм роботи, який необхідно провести для завершення задачі [6].

Обов'язкові поля:

- ID — унікальний ідентифікатор, порядковий номер, який використовується для ідентифікації історій у разі їх перейменування;
- назва (Name) — стислий опис історії. Він повинен бути однозначним, щоб і розробники і product owner могли зрозуміти про що йдеться і відрізнити одну історію від іншої;
- важливість (Importance) — ступінь важливості даної історії на погляд product owner. Зазвичай являє собою натуральне число (іноді для цієї цілі використовуються числа Фібоначчі). Чим більше значення, тим більше пріоритет;
- попередня оцінка (initial estimate) — початкова оцінка об'єму робіт, необхідного для реалізації історії порівняно з іншими історіями. Вимірюється у story point. Приблизно відповідає числу «ідеальних людиноднів»;
- як продемонструвати (how to demo) — стисле пояснення того, як завершена задача буде продемонстрована у кінці спринта. Дане поле може являти собою код автоматизованого приймального тесту.

Іноді, також, використовуються додаткові поля у product backlog, в основному для того, щоб допомогти product owner визначитися з його пріоритетами.

Додаткові поля:

- категорія (track). Наприклад, «панель управління» чи «оптимізація». За допомогою цього поля product owner може легко вибрати усі пункти категорії «оптимізація» і задати їм низький пріоритет;
- компоненти (components) — указує, які компоненти (наприклад, база даних, сервер, клієнт) будуть зачеплені при реалізації історії. Дане поле складається з групи checkbox'ів, які відмічаються, якщо відповідні компоненти потребують змін;
- ініціатор запиту (requestor). Product owner може захотіти зберігати інформацію про усіх замовників, зацікавлених у даній задачі. Це потрібно для того, щоб тримати їх у курсі діла про хід виконання робіт;
- ID у системі обліку помилок (bug tracking ID) — якщо ви використовуєте окрему систему обліку помилок, тоді у описі історії корисно зберігати посилання на всі дефекти, які до неї відносяться.

### 1.1.5 Типи нарад

Планування спринта (Sprint Planning Meeting) – проходить на початку нової ітерації Спринта:

- Із Product Backlog обираються задачі, зобов'язання по виконанню яких за спринт приймає на себе команда.
- На основі обраних задач створюється Sprint Backlog. Кожна задача оцінюється у ідеальних людино-годинах.
- Рішення задачі не повинно займати більше 12 годин або одного дня. При необхідності задача розбивається на підзадачі.

- Обговорюється та визначається, яким чином буде реалізовано цей об'єм робіт.
- Тривалість наради обмежена зверху 4-8 годинами в залежності від тривалості ітерації, досвіду команди тощо.
- Перша частина наради. Беруть участь Product Owner + Команда: обирають задачі із Product Backlog;
- Друга частина наради. Бере участь лише команда: обговорюють технічні деталі реалізації, наповнюють Sprint Backlog.

Щоденна нарада (Daily Scrum Meeting) – відбувається кожен день протягом спринта. Є «пульсом» ходу спринта. Нараді властиві наступні обмеження:

- починається точно вчасно;
- всі можуть спостерігати, але говорять тільки «свині»;
- триває не більш ніж 15 хвилин;
- проводиться в одному і тому ж місці протягом одного спринта.

Протягом наради кожен член команди відповідає на 3 запитання: «Що зроблено з моменту попередньої щоденної наради?», «Що буде зроблено з моменту поточної наради до наступної?», «Які проблеми заважають досягненню цілей спринта?». Над рішенням цих проблем працює ScrumMaster. Зазвичай це рішення проходить за рамками щоденної наради і у складі осіб, що безпосередньо займаються даною перешкодою.

Демонстрація (Sprint Review Meeting) – проходить у кінці ітерації (спринта):

- команда демонструє внесок функціональності до продукту всім зацікавленим особам;
- залучається максимальна кількість глядачів;
- усі члени команди беруть участь у демонстрації (одна людина на демонстрацію або кожен показує, що зробив за спринт);
- обмежена 4-ма годинами в залежності від тривалості ітерації і змін у продукті.

Ретроспектива (Sprint Retrospective) – члени команди висловлюють свою думку про минулий спринт. Відповідають на два основних запитання: «Що було зроблено добре у минулому спринті?», «Що потрібно покращити в наступному?». Виконують покращення процесу розробки (вирішують питання та фіксують вдалі рішення). Обмежена 1-3 годинами.

## **1.2 Існуючі системи управління проектами**

Системи управління проектами — це комплексне програмне забезпечення, яке включає в себе додатки для планування задач, складання розкладу, розподілення ресурсів, сумісної роботи, спілкування, швидкого управління документуванням і адміністрування системи, яке використовується сумісно для управління великими проектами [7]. Коли кількість та масштаби проектів збільшуються, подібні системи стають необхідністю. Мобільні версії подібних програм також набувають усе більшого значення, оскільки дозволяють бути у курсі справ у будь-який час будь-де.

### **1.2.1 JIRA**

Web-орієнтована, повністю налаштовуєма issue-tracking система для управління проектами. Може бути використанна для підтримки клієнтів. Високий рівень безпеки. Наявність API та багатьох плагінів. Відмінна email-інтеграція. Можливий хостинг на стороні провайдера.

Система JIRA розроблена австралійською компанією Atlassian - це web-базований засіб для управління проектами та задачами. JIRA може бути застосована в усіх випадках, коли необхідно організувати роботу співробітників, ефективно назначати їм задачі, мати засоби миттєвого контролю виконання.

JIRA - це продукт, створений для організації процесу контролю запитів і задач, що має частину функціональності багатьох великих і дорогих систем управління проектами.

Ключовими поняттями в JIRA є проекти та задачі. Задачі створюються в проектах, для виконання задач призначаються виконавці. Задачі можуть бути різного типу та мати підзадачі, задачі можуть бути пов'язані з іншими задачами. Статус задач змінюється у процесі їх виконання.

Можливо організувати контроль розробки проектів, роздавши задачі виконавцям, зконфігурувати правила повідомлень про події всіх учасників процесу, керувати правами доступу користувачів та робити багато інших дій.

JIRA приносить велику користь будь-якій організації, діяльність якої можна інтерпретувати як виконання проектів і задач, що мають тематичні рамки та обмеження у часі. Має мобільну версію для платформ iOS та Android. Зображення даної системи управління проектами наведено на рисунках 1.1-1.3.

The screenshot displays the JIRA web interface. At the top, there's a navigation bar with 'Dashboards', 'Projects', 'Issues', 'Agile', 'Structure', and a 'Create issue' button. A search bar is on the right. Below the navigation, the 'Demo Dashboard' is shown. It contains two main panels: 'Structure: Theme Park Project Un-Structure-d' and 'Structure: Theme Park Project Structure'. Each panel shows a list of issues with columns for 'Key', 'Summary', 'Progress' (indicated by a bar), and 'TP' or 'Assign'. A dropdown menu is open over the 'RM-1' issue in the right structure, showing a list of suggestions for assignees: 'Ray', 'Bob - robert@aimwork...', 'Unassigned', and 'Automatic'. Below the issue lists, there is an 'Activity Stream' section showing recent updates.

Рисунок 1.1 – Приклад створення та призначення нової задачі

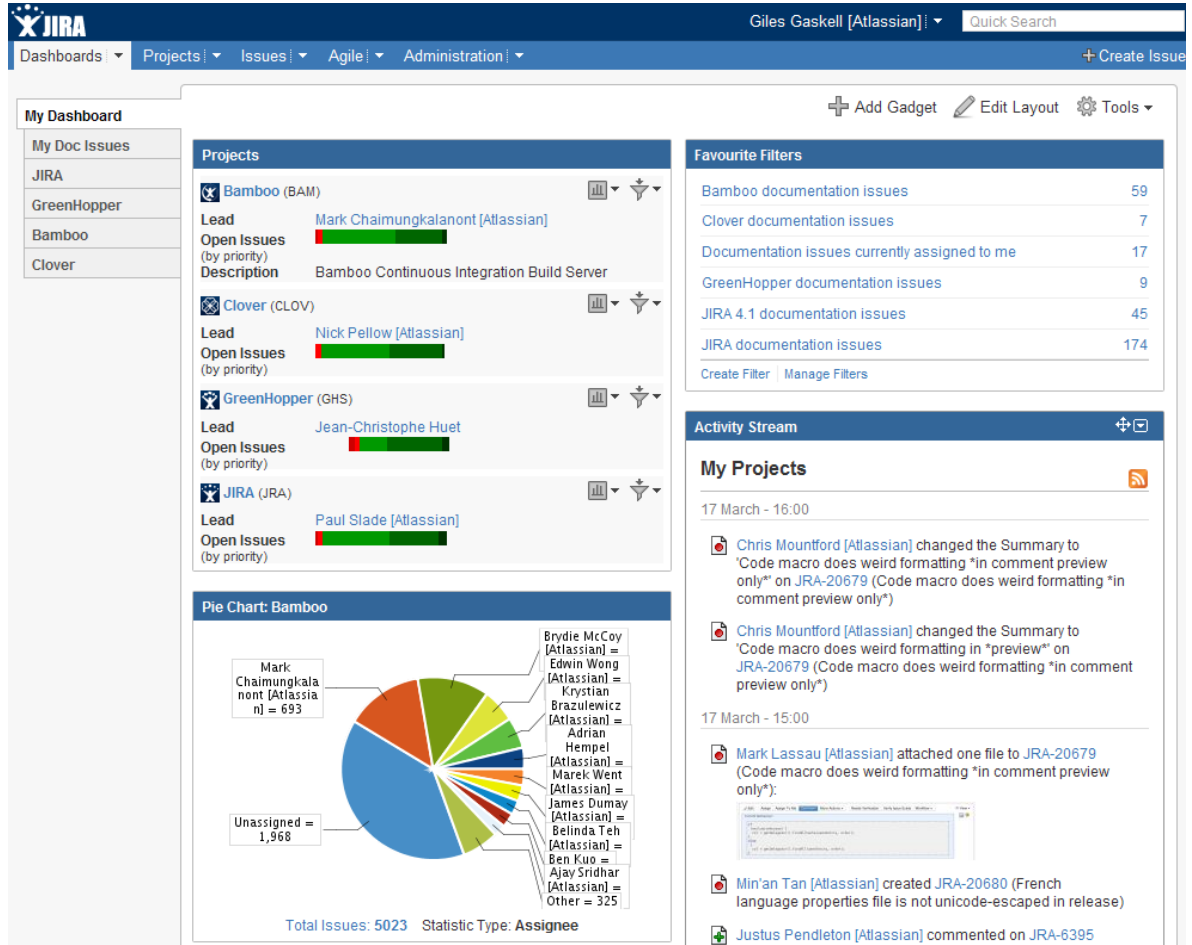


Рисунок 1.2 – Приклад панелі спостереження за проектами

Id	Requirement Summary	Depends On	Test Case	Bug	New Feature	Task
PA-R13 (1)	Functional Requirements	---	---	---	---	---
PA-R1 (2)	Functional Requirement 1 v2	---	PA-T16 (1), PA-T3 (1)	PA-1	---	PA-11
PA-R2 (3)	Functional Requirement 2 v2	Functional Requirement 2 Specs	PA-R1 (1), PA-R1 (2)	---	PA-5	---
PA-R3 (2)	Functional Requirement 3 v2	---	PA-T17 (1)	---	---	PA-10
PA-R4 (1)	Functional Requirement 4	---	PA-T18 (1)	PA-1, PA-3	---	PA-11
PA-R5 (1)	Functional Requirement 5	---	PA-T3 (1)	---	---	---
PA-R14 (1)	Usability Requirements	Usability Guidelines	PA-R4 (1)	PA-T4 (1) Version 1	PA-6	---
PA-R15 (2)	Usability Requirement 1 v2	---	PA-T19 (1)	---	---	PA-9
PA-R16 (2)	Usability Requirement 2 v2	---	PA-R1 (1)	PA-T20 (1)	---	---
PA-R17 (1)	Usability Requirement 3	---	PA-R1 (1)	PA-T20 (1)	---	---
PA-R19 (1)	Admin Screen	---	PA-R2 (2)	PA-T5 (1), PA-T3 (1)	PA-3, PA-4	---
PA-R21 (1)	Dashboard	---	---	---	---	PA-10
PA-R24 (1)	Marketing Requirements	---	---	---	---	---
PA-R25 (1)	Marketing Requirement 1	---	---	---	PA-7, PA-6	---
PA-R26 (1)	Marketing Requirement 2	---	PA-T9 (1)	---	---	---
PA-R27 (1)	Marketing Requirement 3	---	PA-T10 (1)	---	---	---
		PA-R26 (1)	PA-T11 (1)	---	---	---

Рисунок 1.3 – Приклад панелі відстеження зв'язку між вимогами



### 1.2.2 TeamBridge

Доступність TeamBridge забезпечується безкоштовністю стартового аккаунту та можливістю використовувати повний функціонал, незважаючи на наявність підписки. Платити потрібно лише за можливість підключити більшу кількість співробітників для роботи на порталі та розширити об'єм дискового простору для зберігання різної інформації. Другий плюс сервісу – це простота освоєння та однозначність функцій. Також корисною є наглядність робочого процесу та, беручи до уваги онлайн-спрямованість, швидке оновлення функцій.

До того ж TeamBridge – хмарний сервіс, а отже доступ до нього є звідусіль і завжди (при наявності інтернет-підключення). Зникає необхідність тримати штат адміністраторів для підтримки своїх серверів і інфраструктури мережі, а також значно зменшується час простою при переїзді з одного офісу в інший.

Але з таким підходом є декілька недоліків і найголовніший з них – це залежність від роботи серверів TeamBridge та інтернет-з'єднання.

Ще одна особливість – це неможливість реєструвати співробітників у системі при наявності тільки внутрішнього поштового сервері: потрібна робоча «зовнішня» постова скринька. Та третє зауваження – це неможливість задавати задачі за часом: формування задач та віх приходить на день, і точно планувати виконання задач за робочий день не можна.

Вцілому, сервіс може бути корисним не тільки для координації робіт організацій з декількома філіями у різних містах чи з задіянням фрілансерів, але і для компаній з великою кількістю проектів чи повсякденних задач і інтенсивними комунікаціями між співробітниками, підрозділами, замовниками та підрядчиками. До того ж абонентська плата дозволяє масштабувати витрати на організаційні питання в залежності від кількості проектів компанії на даний момент. Має мобільну версію для платформ iOS та Android. Зображення даної системи управління проектами наведено на рисунку 1.4.

Главная | Рабочие области | Задачи | Вехи | Обсуждения | Календарь | Лента Личный кабинет | Надежда Чередиченко

+ ДОБАВИТЬ
Рабочие области

Рабочие области

КОМПАНИЯ

Проекты

- Проект "Ликвидация новогодней продукции"
- Проект "Телемаркетинг"
- Разработка сайта "Текстиль"
- Участие в выставке "Технологии рекламы"

Отделы

- Администрация
- Отдел закупок
- Отдел логистики и доставки
- Отдел маркетинга и рекламы
- Отдел продаж

Наименование	Статус	Срок	Ответственный
Компания			Надежда Чередиченко
Отдел маркетинга и рекламы			Надежда Чередиченко
Промежуточный анализ работы в социальных сетях		01 апреля	Надежда Чередиченко
Прохождение испытательного срока!!		01 апреля	Дарья Петренко
Анализ эффективности проведенных рекламных кампаний за 1 квартал		10 апреля	Надежда Чередиченко
Готовность комплекта рекламных материалов по всем продуктам и услугам компании		12 апреля	Надежда Чередиченко
Контекстная реклама: промежуточный анализ		30 апреля	Надежда Чередиченко
SEO оптимизация сайта. Выход в топ 10.		31 мая	Надежда Чередиченко
PR активность: публикация в топ сми отрасли		28 июня	Надежда Чередиченко
Отдел продаж			Надежда Чередиченко
Отдел закупок			Юлия Паукова
Отдел логистики и доставки			Владимир Букатин
Администрация			Дмитрий Жабин
Проект "Телемаркетинг"		30 апреля	Дарья Петренко
Проект "Ликвидация новогодней продукции"		30 апреля	Надежда Чередиченко
Участие в выставке "Технологии рекламы"		30 апреля	Александр Чугреев
Разработка сайта "Текстиль"		08 мая	Надежда Чередиченко

Рисунок 1.4 – Приклад відображення існуючих завдань по категоріям

### 1.2.3 Zoho Projects

Zoho Project — це інструмент управління і контролю за проектами, який дозволяє створювати задачі, задавати терміни виконання і відстежувати етапи робіт; працювати з календарем, графіками Гантта, звітами; розподіляти файли підтримки — тобто в наявності всі стандартні функції, що мають бути в пакетах управління проектами. Zoho Projects безкоштовний для одного проекту, функціонал розміщений у мережі Internet. Має мобільну версію для платформ iOS та Android. Зображення даної системи управління проектами наведено на рисунках 1.5-1.7.

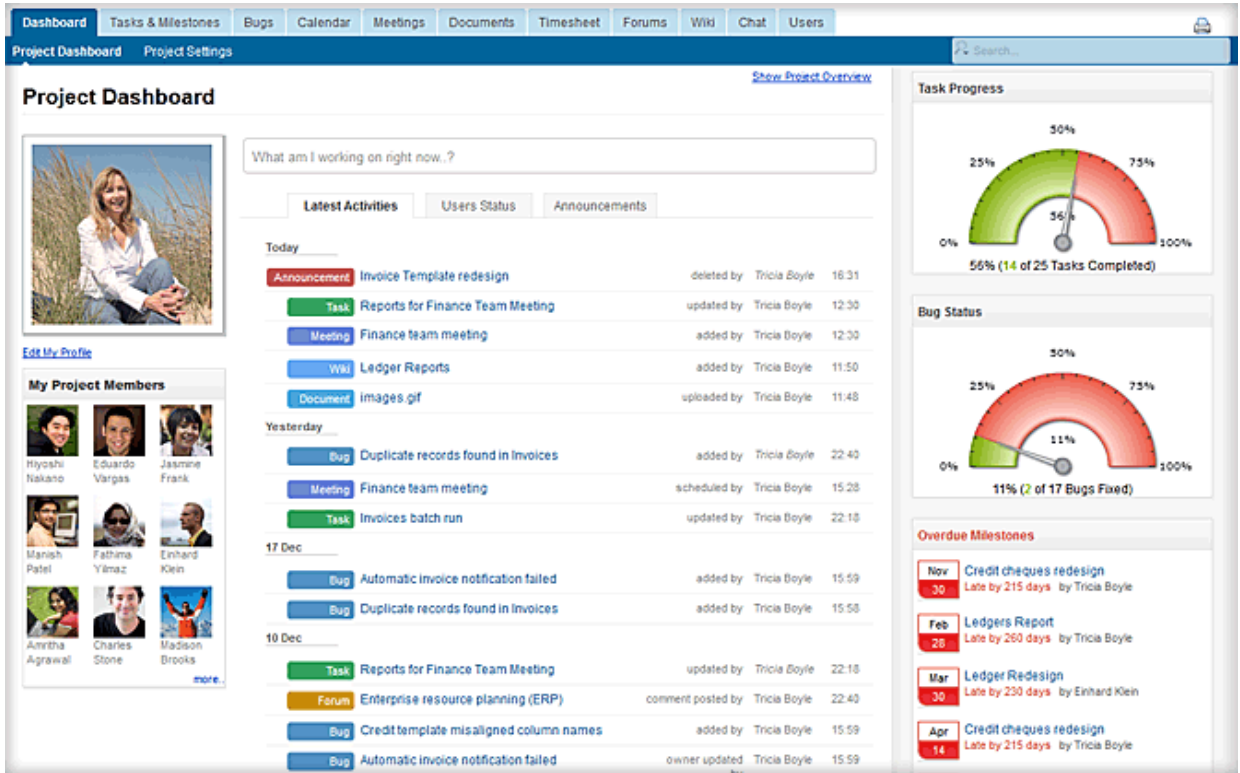


Рисунок 1.5 – Приклад панелі спостереження за проектами

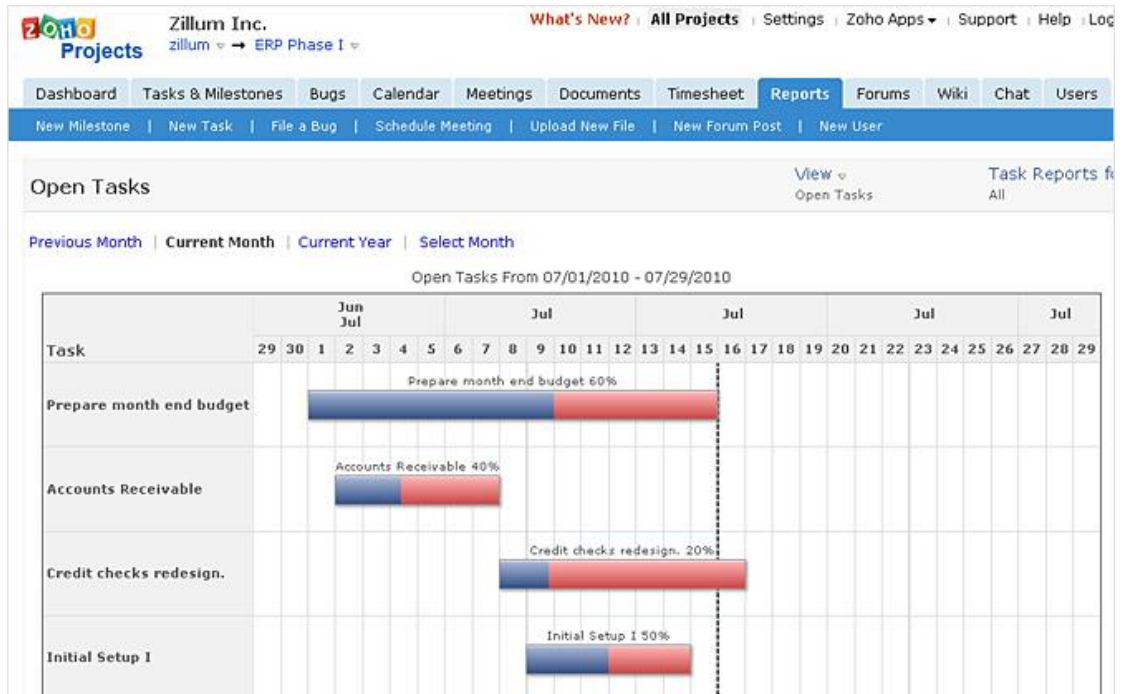


Рисунок 1.6 – Приклад діаграми прогресу виконання завдань

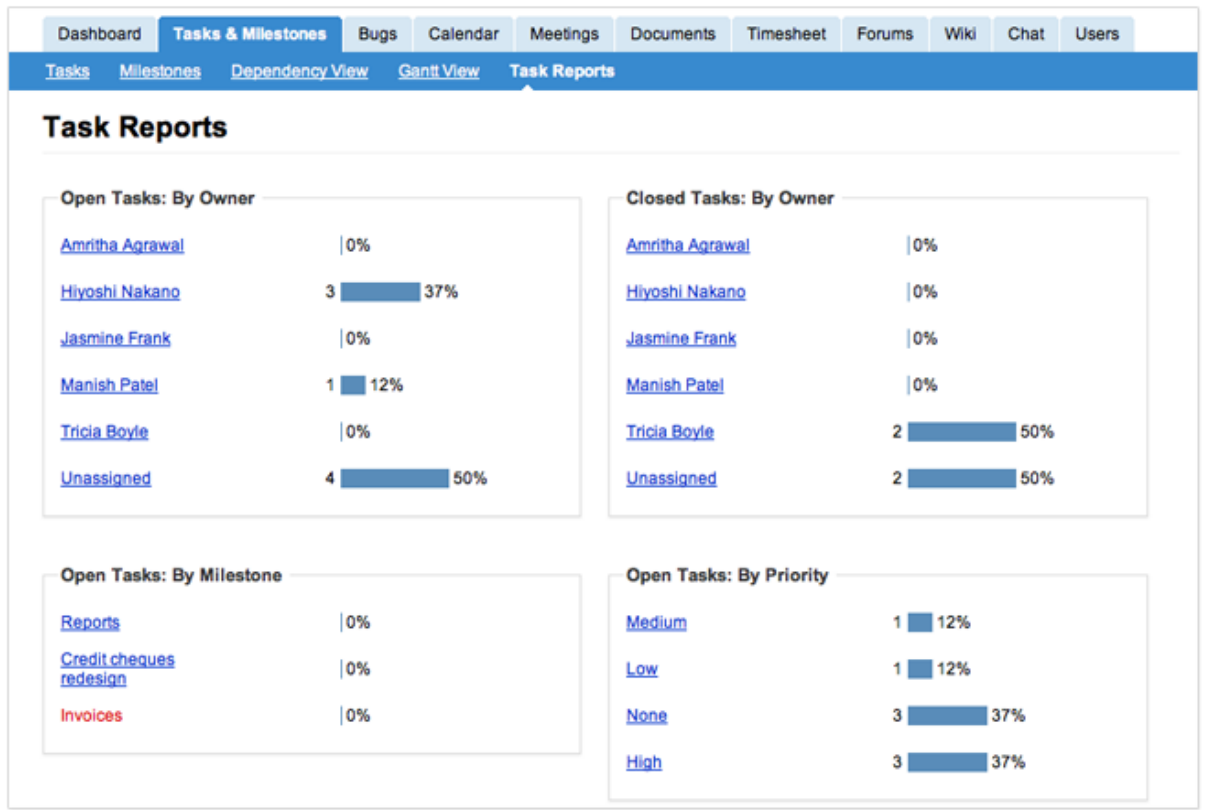


Рисунок 1.7 – Інформація стосовно завдань у даному проекті

## 1.2.4 Comindware Project

Comindware Project — онлайн-сервіс з мобільними додатками для проектного управління й командної взаємодії. Він допомагає полегшити процес планування для менеджерів й ефективно покращити виконання задач. Тут є функції установки пріоритетів у режимі реального часу, управління завантаженням ресурсів і автоспостереження за прогресом виконання пунктів.

Розробники внесли в систему оптимізовану навігацію з розділом «Обранного», нові функції архівації та копіювання проектів. Подібні нововведення збережуть час командам, які реалізують типові проекти. У той же час нова архівація дозволить зберігати завершені проекти з усією базою документів й обговореннями. Зображення даної системи управління проектами наведено на рисунках 1.8-1.9.

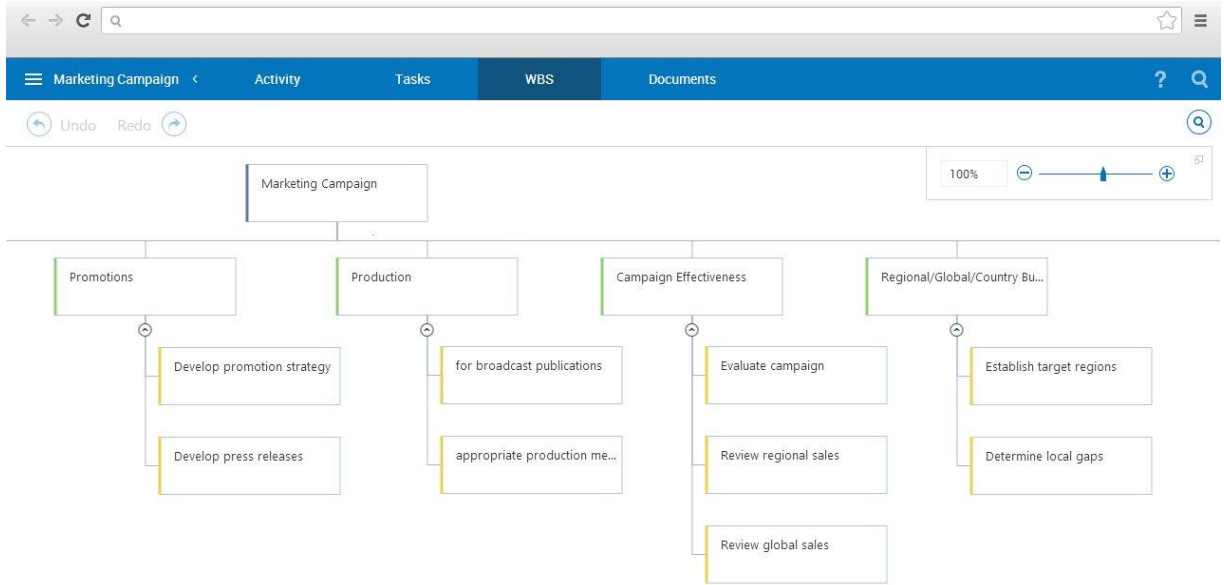


Рисунок 1.8 – Графічне пердставлення планування проекту

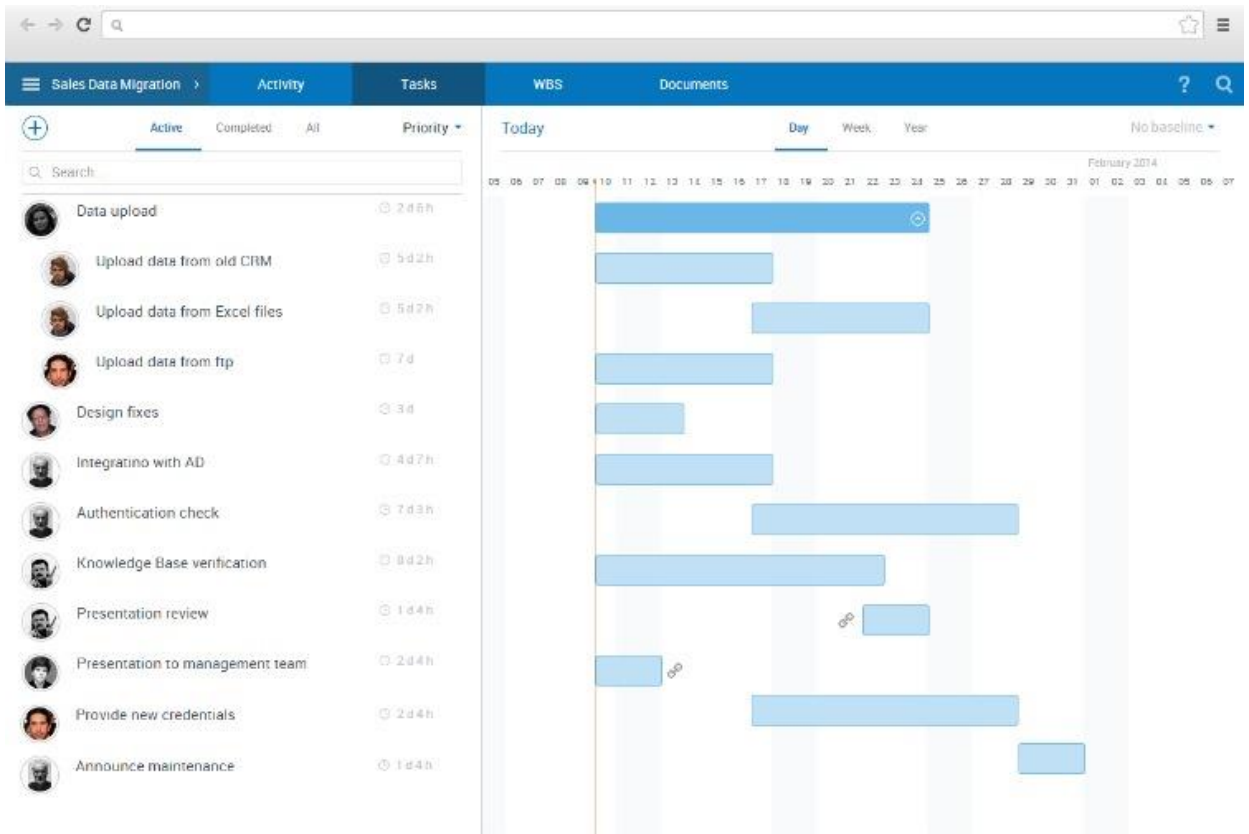


Рисунок 1.9 – Активні завдання, представлені у вигляді діаграми Ганта

## 1.2.5 Yodiz

Yodiz — веб-система управління проектами й відстеження задач для команд розробників програмного забезпечення. Вона може рухатись по спринтам і релізам з повідомленнями на електронну адресу, реалізуючи швидкий і простий робочий процес відстеження помилок. Має багато можливостей та має мобільні клієнти. Має мобільну версію для платформ iOS та Android. Зображення даної системи управління проектами наведено на рисунках 1.10-1.12.

The screenshot displays the 'Product Backlog' interface in Yodiz. At the top, there are summary statistics: 88 Unscheduled (666 Hours), 54 In Development (1276 Hours), and 9 Accepted (59 Hours). Below this is a search bar and a 'Filters' button. The main area is a table of user stories, sorted by priority (1 to 12). Each row includes a priority indicator, ID, title, epic, component, estimate in hours, status, and owner.

Priority	ID	Title	Priority	Epic	Component	Estimate (hr)	Status	Owner
1	US-3681	new user story for demo api browser	★★★★★			10	New	Mark
2	US-3668	new user story for demof ui browser	★★★★★	EP-112		13	New	Mark
3	US-3200	new user story for demo usability	★★★★★			10	In Progress	Mark
4	US-3588	new userstory , browerplugin firefox	★★★★★			15	New	Mark
8	US-3685	my new use story web api	★★★★★			43	New	Mark
9	US-1637	new user story	★★★★★			16	New	Mark
10	US-3659	my new US for demo api testing ui	★★★★★	EP-115		17	New	Mark
11	US-1609	dont send me notification when i comment	★★★★★			15	New	No User
12	US-3215	login and logout	★★★★★	EP-112		11	In Progress	Mark

Рисунок 1.10 – Перелік усіх активних завдань за їх пріорітетом

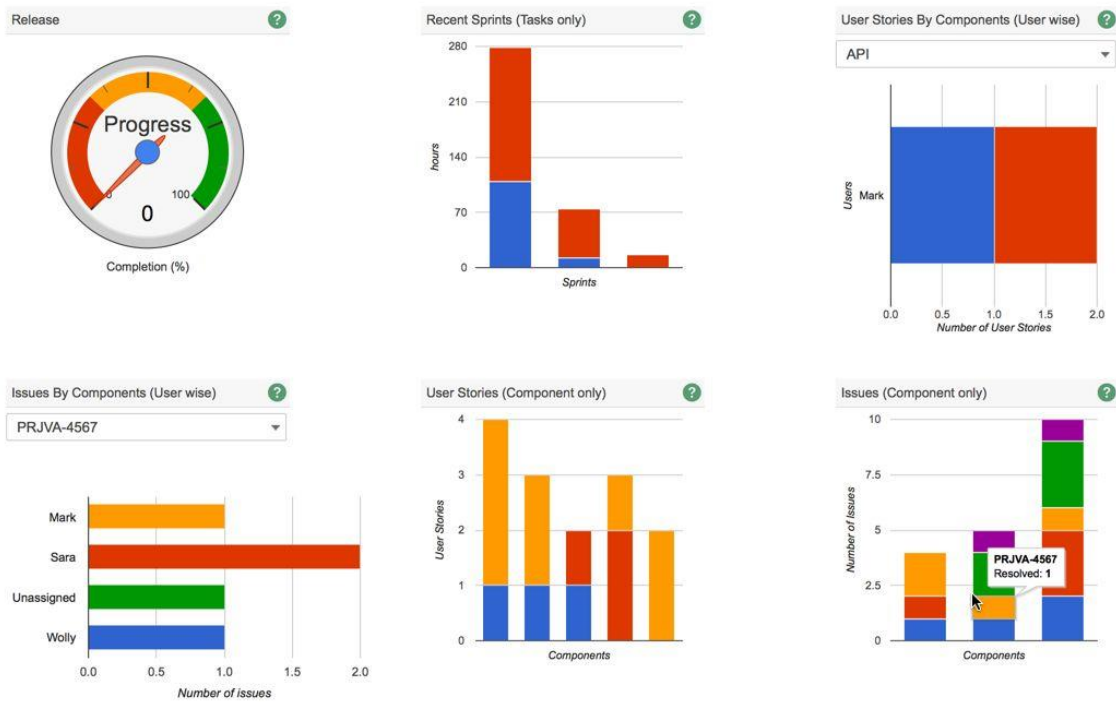


Рисунок 1.11 – Зображення доступних діаграм для більш зручного представлення інформації

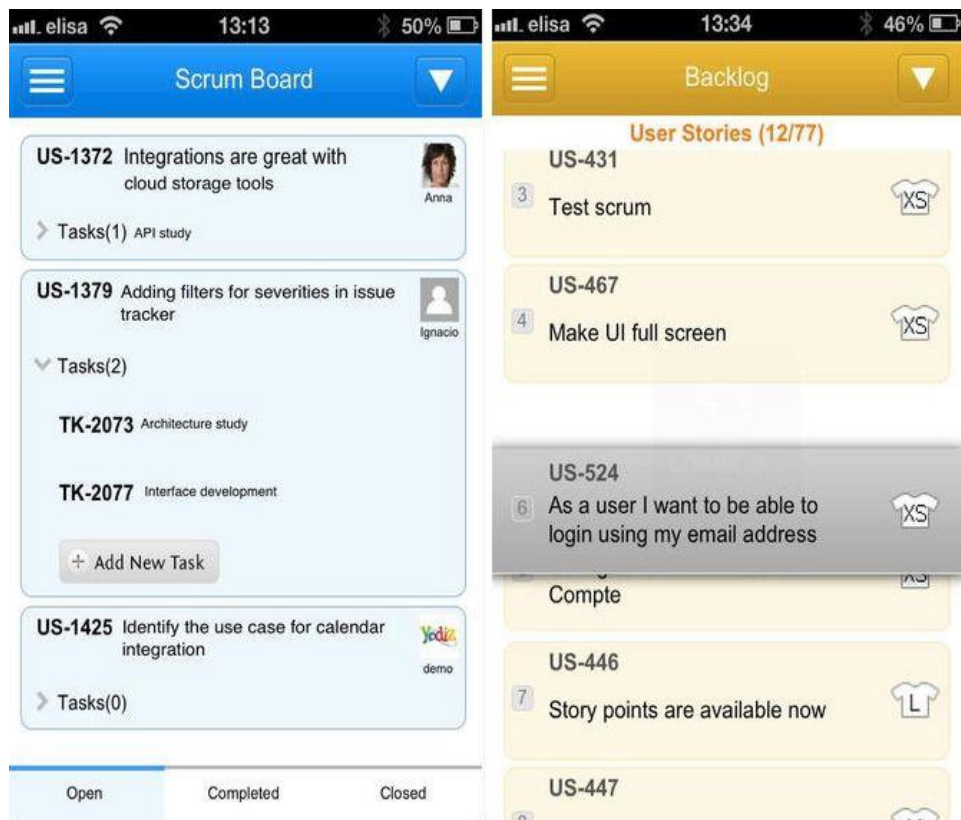


Рисунок 1.12 – Приклад мобільного додатку



### 1.3 Математична статистика як розділ статистики

Математична статистика - це розділ математики, присвячений вивченню закономірностей, що мають місце в масових явищах і математичних сукупностях. Зміст математичної статистики складають математичні методи систематизації, обробки і аналізу масових статистичних даних незалежно від їхнього якісного змісту [8].

Методи математичної статистики можуть бути застосовані для обробки і аналізу будь-яких статистичних даних. Математичні методи безпосередньо пов'язані з імовірнісною оцінкою результатів спостереження і визначенням математичної ймовірності. У зв'язку з цим висновки математичної статистики відносно масових явищ і процесів носять імовірнісний характер і спираються на апарат теорії ймовірностей.

Специфіка математичної статистики як особливої наукової дисципліни полягає в тому, що вона розглядає сукупності незалежно від їх конкретної природи і змісту, в яких варіація визначається випадковими причинами, розглядає їх як абстрактно-математичні сукупності [9].

Абстрактні сукупності і є предметом вивчення математичної статистики.

Теоретичною основою математичної статистики є теорія ймовірностей (як складова частина вищої математики), яка розглядає закономірності випадкових явищ.

Найважливішими розділами математичної статистики є статистичні ряди розподілу та їх характеристики, оцінка параметрів розподілу, перевірка статистичних гіпотез, дисперсійний і кореляційний аналіз. Останнім часом одержують широке застосування методи багатовимірної статистичного аналізу - факторний і кластерний аналіз, метод головних компонент та інші [8].



## 1.4 Постановка завдання

Метою даної дипломної роботи є розробка програмного засобу, що буде здійснювати статистичну обробку та відображення інформації шляхом застосування математичної статистики. Треба розробити мобільний додаток з використанням статистичних методів для слідкування за прогресом власної роботи над проектами та внесення у них незначних змін.

Для цього необхідно виконати наступні завдання:

- отримати необхідні для обробки статистичні дані;
- дослідити отримані статистичні дані для їх наступної обробки;
- розробити статистичну модель для роботи з даними;
- реалізувати отриману модель та отримати результати.

Як тестовий матеріал обрані дані з роботи програми, яка допомагає здійснювати контроль за роботою над проектами (інформація про наявні помилки, призначення співробітників, ліміт часу на завдання і т.д.).

## 2 ОПИС ОСНОВНИХ ФУНКЦІЙ СИСТЕМИ

### 2.1 Побудова різних типів діаграм

Основними функціями розроблюваної системи є отримання та графічне відображення даних користувача стосовно проектів, у яких він задіяний. Інформація подається у вигляді чотирьох видів діаграм.

#### 2.1.1 Полігон частот

Полігон частот – це один з найпростіших видів графічного зображення статистичної інформації. Має вигляд ламаної, що поєднує точки, які відповідають значенням інтервалів групування [10]. Як приклад, можна навести кількість створених запитів за кожен день на протязі деякого проміжку часу (тиждень, місяць тощо). При наявності існуючої вибірки будується дуже зручно та просто. Приклад графічного зображення полігону частот наведено на рисунку 2.1.

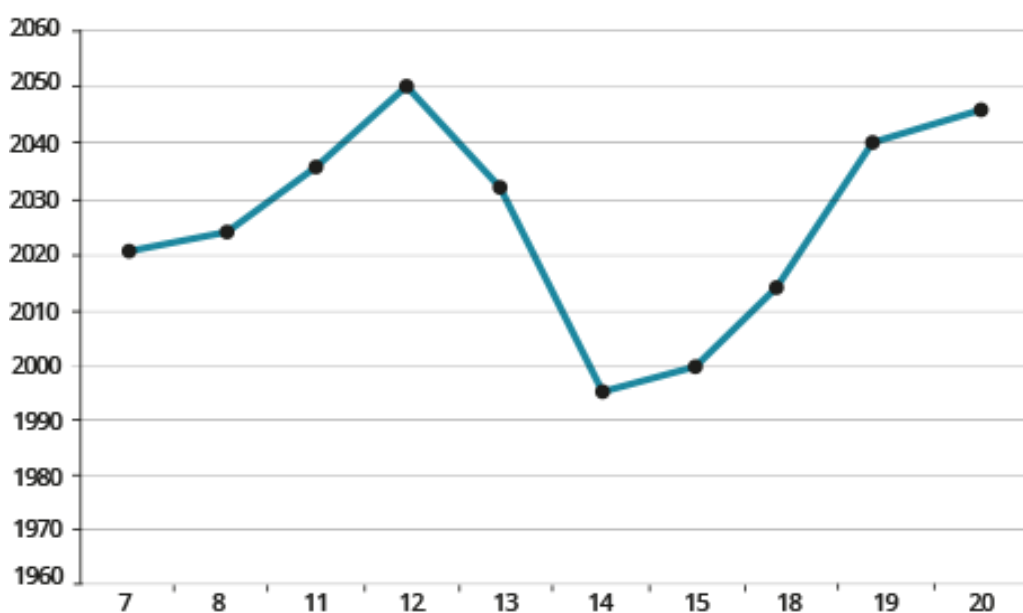


Рисунок 2.1 – Приклад зображення полігону частот

### 2.1.2 Кругова діаграма

Кругова діаграма графічно представляє кількісні дані у вигляді кола, поділеного на сектори. У колі відносні розміри секторів відповідають часткам кількісних показників [10]. По суті, цей графік демонструє процентне співвідношення частин порівняно з цілим. Зазвичай кожен сегмент відповідає найбільшим значенням параметра вибраного звіту. Менші значення групуються в сегмент "Інше".

Для того, щоб дізнатися, який кут матиме сектор, що відображає кожен показник, потрібно відношення значення цього показника до сумарного значення усіх показників помножити на 360. Початок наступного сектору завжди співпадає з кінцем попереднього. Побудова даного виду діаграм за наявної вибірки також не викликає проблем. Приклад графічного зображення кругової діаграми наведено на рисунку 2.2.

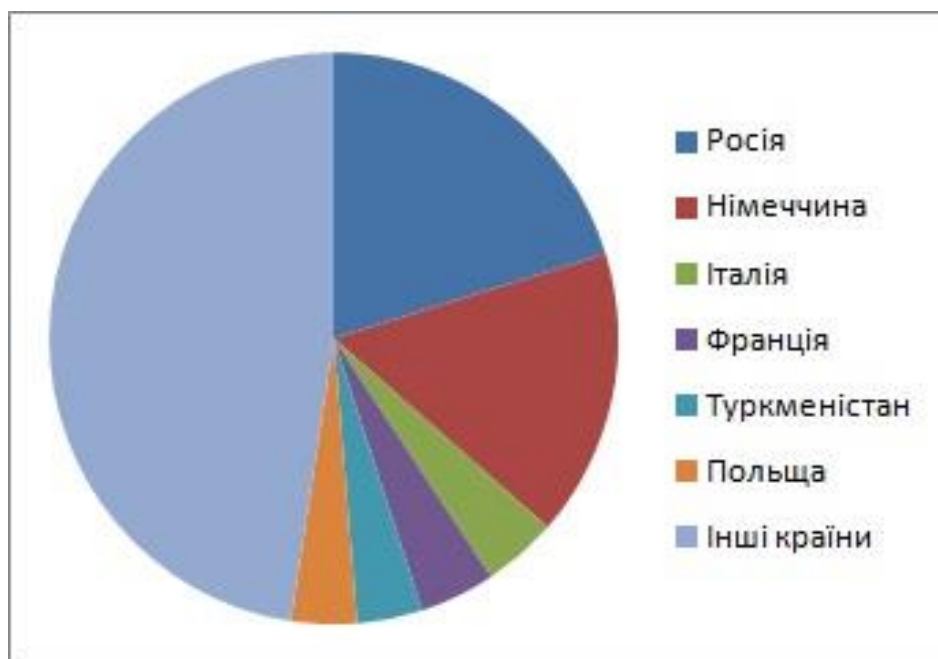


Рисунок 2.2 – Приклад зображення кругової діаграми

### 2.1.3 Гістограма

Гістограма – спосіб графічного представлення табличних даних. Являє собою діаграму, що складається з прямокутників без розривів між ними [10].

Горизонтальні межі прямокутника — інтервал групування статистичного ряду. Нижня межа прямокутника розміщена на осі  $Ox$ , а висота задається так:

$$\frac{m_i}{nh_i} \quad (2.1)$$

де  $m_i$  – значення даного інтервалу статистичного ряду;

$n$  – кількість інтервалів;

$h_i$  – ширина інтервалу.

Найчастіше для зручності беруть рівномірне розбиття статистичного ряду, тобто однакові розміри інтервалів  $h_i$  для будь-яких  $i$ . За характером інформації гістограма схожа на полігон частот, але відрізняється від останнього зовнішнім виглядом.

Якщо об'єднати одразу дві гістограми в одну, можна отримати гістограму з поповненням. Наприклад за допомогою гістограми з поповненням можна одразу відобразити кількість вирішених та невирішених завдань за проміжки часу на деякому часовому інтервалі. Приклад графічного зображення гістограми наведено на рисунку 2.3.

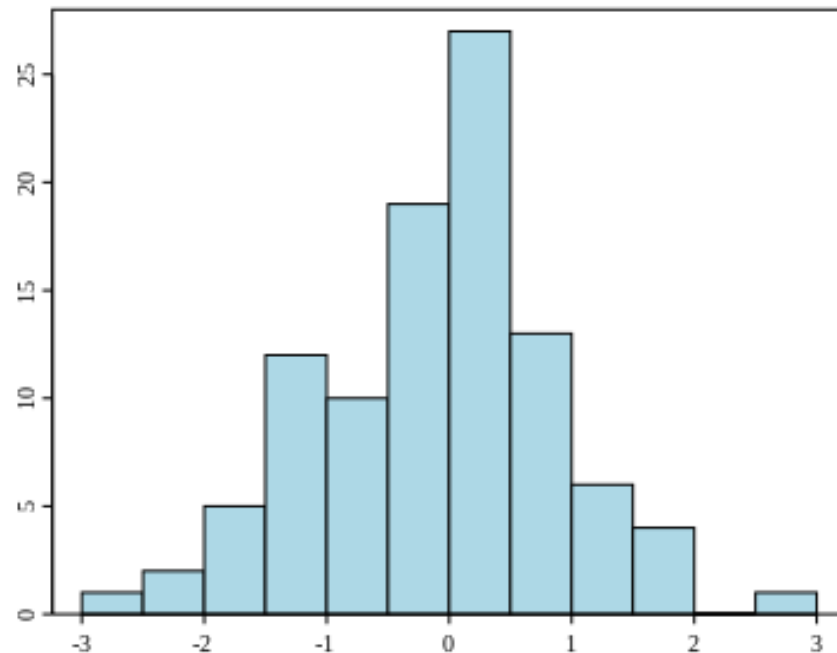


Рисунок 2.3 – Приклад зображення гістограми

#### 2.1.4 Діаграма Ганта або стрічкова діаграма

Діаграма Ганта – це популярний тип діаграм, який використовується для ілюстрації плану, графіка робіт за будь-яким проектом. Є одним з методів планування та управління проектами [10].

Перший формат діаграми був розроблений Генрі Л.Гантом у 1910 році.

Діаграма Ганта являє собою відрізки, розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню, заштриховується, відзначаючи відсоток виконання завдання; показується вертикальна лінія, що відповідає моменту «сьогодні».

Часто діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці містять додаткову інформацію про задачу. Приклад графічного зображення гістограми наведено на рисунку 2.4.

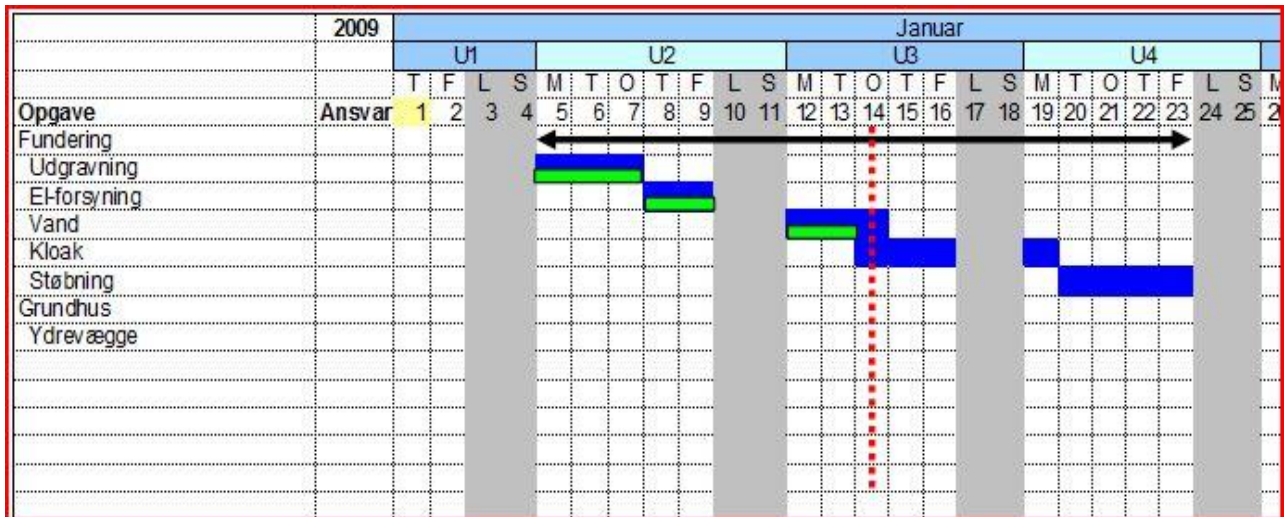


Рисунок 2.4 – Приклад зображення діаграми Ганта

## 2.2 Принципи побудови діаграм розроблюваного додатку

Мобільний додаток отримує необхідну для отримання статистики інформацію через мережу Internet, на її основі й здійснюється побудова діаграм, які графічно відображають користувачеві дані, що його цікавлять.

Полігон частот містить одразу дві ламані різного кольору, які відповідають за відображення створених та вирішених запитів за певний проміжок часу, який може складати до одного календарного місяця до дати на момент побудови діаграми. Вершини ламаних програмно розташовуються на перетині відповідної дати та кількості запитів (створених або вже вирішених).

Кругова діаграма буде відображати робоче навантаження команди, до якої відноситься користувач. Додаток отримує інформацію щодо кількості

відпрацьованих кожним членом команди годин над проектом та на основі цих даних вираховує відсоткові частки від загального (сумарного) робочого часу. Відповідно до цих часток загального часу будується кругова діаграма, кожен сектор якої відображає свого учасника. Радіус кола та кольори секторів встановлюються програмно. При кількості учасників команді більшій за 15 усі, хто відпрацював часу менше, ніж п'ятнадцятий учасник (за спаданням часу роботи) будуть віднесені до створеного сектору «Інші працівники». Сектори діаграми, що відповідають першим п'ятнадцятьом працівникам, будуть названі за їхніми іменами.

Гістограмою у розроблюваному додатку буде відображатися інформація про кількість створених запитів щоденно протягом останніх тридцяти днів. Для цього значення кількості існуючих запитів на дату, яка була 30 днів тому, будемо збільшувати щоразу, як будуть додаватися нові запити. Уся потрібна для цього інформація надходить у мобільний додаток через мережу Internet та повинна оновлюватися щоденно о сьомій ранку.

### **2.3 Майбутні функції**

З часом планується додати у розроблену систему алгоритм прогнозування часу, необхідного на вирішення задач різних типів. Даний прогноз буде базуватися на інформації про час, затрачений на подібні задачі користувачем та іншими співробітниками, задіяними над подібними завданнями. Ця інформація для прогнозу буде збиратися за термін, що дорівнює повним останнім шести календарним місяцям роботи користувача. Якщо для користувача подібний тип завдання зустрічається вперше, то замість прогнозу буде виводитись найменший та найбільший час серед співробітників, які вже знайомі з подібними задачами.

### 3. ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНИХ АЛГОРИТМІВ

#### 3.1 Вибір середовища розробки

Мова програмування Java найбільше підходить для написання мобільних додатків під операційну систему Android. Тому для розробки було обрано середовище розробки Eclipse, бо у ньому зручно працювати з Java. Окрім середовища розробки Eclipse для роботи знадобилося встановлення:

- ADT плагіну для Eclipse;
- JDK та JRE;
- Android SDK.

Тестування додатку доцільно провести за допомогою Android Virtual Device (AVD). AVD – це віртуальний смартфон з операційною системою Android, який дозволяє відтворювати багато моделей смартфонів, що позитивно впливає на кінцевий результат розробки та зберігає час.

#### 3.2 Архітектура системи

Розроблюваний додаток використовує шаблон проектування Model-view-controller (MVC).

Model-view-controller — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Цей шаблон поділяє систему на три частини: модель даних, вигляд даних та керування. Застосовується для відокремлення даних (модель) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.



Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах призводить до певної впорядкованості їх структури і робить їх зрозумілішими завдяки зменшенню складності.

Архітектурний шаблон Модель-Вид-Контролер (MVC) поділяє програму на три частини. У тріаді до обов'язків компоненту Модель (Model) входить зберігання даних і забезпечення інтерфейсу до них. Вид (View) відповідальний за представлення цих даних користувачеві. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача і повідомляє про зміни компоненту Модель. Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами.

MVC поділяє цю частину системи на три самостійні частини: введення даних, компонент обробки даних і виведення інформації. Модель, як вже було відмічено, інкапсулює ядро даних і основний функціонал з їх обробки. Також компонент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вид може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація. У функції Контролера входить моніторинг за подіями, що виникають в результаті дій користувача (зміна положення курсору миші, натиснення кнопки або введення даних в текстове поле).

Зареєстровані події транслуються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від виду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.

У розробленому додатку моделлю виступають java-класи, які відповідають за інформацію про обліковий запис користувача та необхідну для роботи інформацію

про проекти, у яких задіяний даний користувач. Вид представлений java-класами та xml-файлами, які відповідають за графічний інтерфейс входу у свій обліковий запис, головне меню та чотири види діаграм, що відображають інформацію про участь у проектах. Контролери представлені класами, які відповідають за отримання інформації та зв'язок між видами та моделями. Окрім класів обов'язковими складовими мобільного додатку є його маніфест та ресурси. Основні компоненти розробленого додатку наведено на рисунку 3.1.

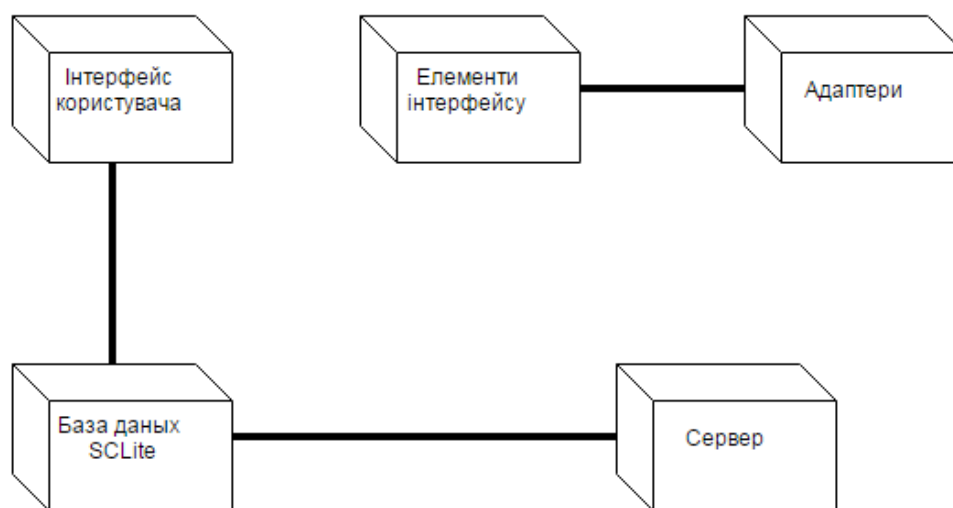


Рисунок 3.1 – Діаграма компонентів додатку

Приблизна структура проекту:

- Src — тут знаходиться вихідний код додатку (java-класи);
- Gen — папка, де знаходяться згенеровані системні файли;
- bin — скомпільовані файли проекту;
- libs — різноманітні бібліотеки, які використовуються додатком;
- res — різноманітні ресурси проекту (іконки, xml-файли, описи елементів меню, різні константи тощо);
- AndroidManifest.xml — файл опису проекту (містить підтримувані версії SDK, версії додатку тощо).

Структура проекту розробки Android додатку показана на рисунку 3.2.

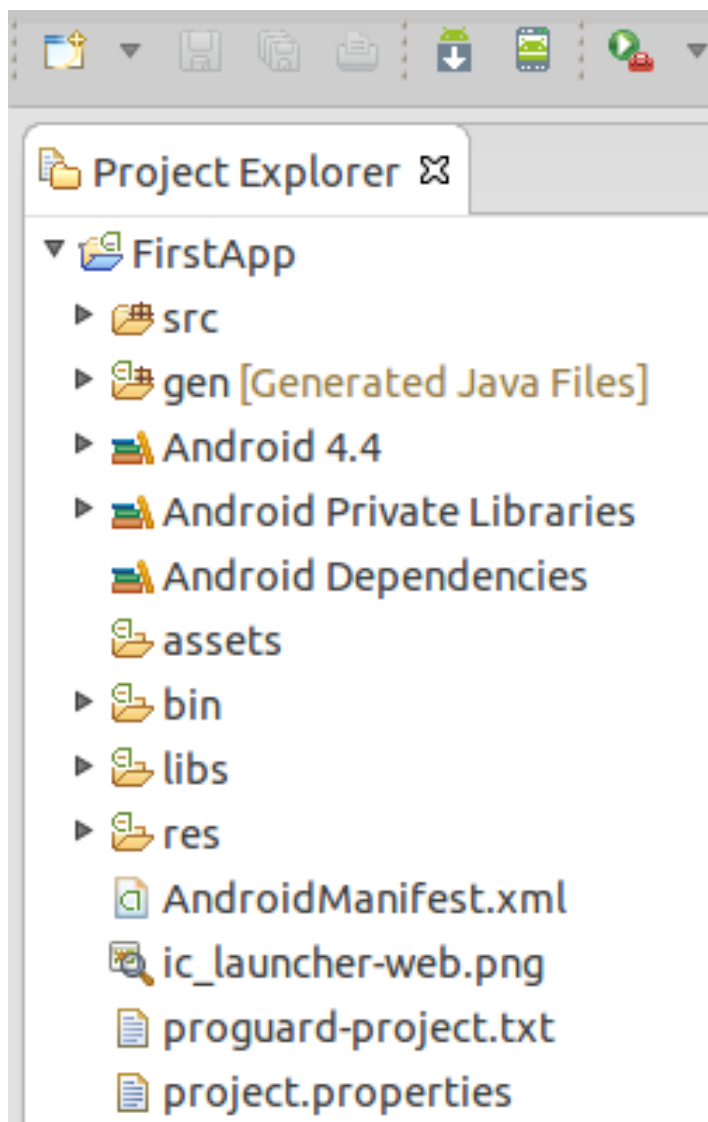


Рисунок 3.2 – Структура проекту

### 3.3 Основні технології, які використовуються при створенні Android-додатків

#### 3.3.1 Програмний фреймворк

Забезпечуючи відкриту платформу для розробки, Android пропонує розробникам можливість побудови інноваційних та різноманітних додатків.

Розробники легко можуть скористатися усіма апаратними перевагами Android пристрою, отримувати доступ до інформації о місцезнаходженні, запускати фонові сервіси, встановлювати оповіщення, додавати нотифікації на статус-панель тощо.

Розробники мають доступ до тих самих API фреймворку, що і базові додатки. Архітектура додатку спроектована таким чином, щоб спростити багаторазове використання його компонентів. Будь-який додаток може заявити про свої можливості, і будь-який інший додаток може цими можливостям скористатися (можливі проблеми безпеки опрацьовуються та контролюються фреймворком). Цей єдиний механізм дозволяє користувачу замінювати компоненти [12].

В основі усіх додатків знаходяться сервіси та системи, включаючи:

- різноманітний та розширюваний набір View, який може бути використаний під час побудови власних додатків, включаючи списки, текстові поля, кнопки та навіть вбудований веб браузер;
- контент провайдери, які дозволяють додаткам отримувати дані інших додатків (таких як Контакти), або ділитися власними даними;
- менеджер оповіщень, який дозволяє додаткам виводити власні оповіщення на статус-панель;
- менеджер ресурсів, який дозволяє додаткам отримувати доступ до позапрограмних ресурсів, таких як локалізовані строки, зображення та layout-файли;
- менеджер activity, який управляє життєвим циклом додатку та надає загальний стек зворотної навігації.

Android включає в себе набір бібліотек C/C++, які використовуються різними компонентами Android системи. Ці можливості представлені розробниками за допомогою програмного фреймворку Android. Деякі з найбільш важливих бібліотек представлені нижче:

- системна бібліотека C — BSD-похідна реалізація стандартної системної бібліотеки C (libc), налаштована спеціально на роботу на впроваджених Linux - пристроях;

- медіа бібліотеки — основані на PacketVideo's OpenCORE. Вони підтримують відтворення та запис багатьох популярних аудіо та відео форматів, а також файлів статичних зображень, таких як MPEG4, H.264, MP3, AAC, AMR, JPG, и PNG;
- Surface Manager — керує доступом до підсистеми відображення та непомітно для користувача поєднує графічні 2D і 3D слої декількох додатків;
- LibWebCore — сучасний движок веб браузера, який використовується в Android браузері;
- SGL — графічний 2D движок, що лежить в основі системи відображення;
- бібліотеки 3D графіки — реалізація заснована на API OpenGL ES 1.0. Бібліотеки використовують або апаратне 3D прискорення (де це можливо), або вбудовану, високо-оптимізовану систему 3D растеризації;
- FreeType — відображення растрових і векторних шрифтів;
- SQLite — потужний та легкий движок реляційних баз даних, доступний у всіх додатках.

### 3.3.2 SQLite

Особливістю SQLite є те, що вона не використовує парадигму клієнт-сервер, тобто движок SQLite не є окремим процесом, з яким взаємодіє додаток, а надає бібліотеку, з якою програма компілюється і движок стає складовою частиною програми. Таким чином, як протокол обміну використовуються виклики функцій бібліотеки SQLite. Такий підхід зменшує накладні витрати, час відгуку і спрощує програму. SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому пристрої, на якому виконується додаток. Простота реалізації досягається за рахунок того, що перед початком

виконання транзакції весь файл, що зберігає базу даних, блокується; ACID-функції досягаються зокрема за рахунок створення файлу-журналу.

Кілька процесів можуть одночасно без жодних проблем читати дані з однієї бази. Запис в базу можна здійснити тільки в тому випадку, коли жодних інших запитів у цей час не обслуговується; інакше спроба запису закінчується невдачею, і в програму повертається код помилки. Іншим варіантом є автоматичне повторення спроб запису протягом заданого інтервалу часу.

У комплекті постачання йде також функціональна клієнтська частина у вигляді виконуваного файлу `sqlite3`, за допомогою якого демонструється реалізація функцій основної бібліотеки. Клієнтська частина працює з командного рядка, і дозволяє звертатися до файлу БД на основі типових функцій ОС.

Завдяки архітектурі движка можливо використовувати `SQLite` як на вбудовуваних (`embedded`) системах, так і на виділених машинах з гігабайтними масивами даних.

### 3.3.3 Apache HttpClient

Протокол HTTP, можливо, є найзначнішим протоколом, що використовується у наш час у мережі Internet. Чисельні веб-сервіси, додатки, що працюють у мережі, та розвиток комп'ютерних мереж продовжують збільшувати роль протоколу HTTP та кількість додатків, які його використовують.

`Apache HttpClient` це бібліотека, яка значно розширює функціональність пакету класів `java.net`. Вона найбільш підходить для реалізації мережевих операцій на платформі Android.

Серед особливостей цієї бібліотеки можна назвати: повна реалізація усіх HTTP методів (`Get`, `Post`, `Put`, `Delete`, `Head`, `Options`, and `Trace`) у здатному розширюватися фреймворку; підтримка шифрування по протоколу HTTPS; тунельовані HTTPS з'єднання через HTTP проксі за допомогою методу `Connect`;

наявність багатьох аутентифікаційних схем (Basic, Digest, NTLMv1, NTLMv2, NTLM2 Session, SPNEGO, Kerberos); зручність у використанні сторонніх рішень; підтримка управління з'єднанням у багатопоточних додатках; прямий доступ до коду відповіді та заголовків, відправлених сервером.

Вихідний код знаходиться у вільному доступі згідно з ліцензією Apache.

### 3.3.4 Багатопоточність у Android додатках

В Android до потоку може бути прив'язана черга повідомлень. Можна відправляти туди повідомлення, а система буде спостерігати за чергою та відправляти повідомлення на відпрацювання. При цьому можна вказати, щоб повідомлення пішло на відпрацювання не одразу, а з деяким запізненням.

Handler — це механізм, який дозволяє працювати з чергою повідомлень. Він прив'язаний до конкретного потоку та працює з його чергою.

Handler вміє переміщати повідомлення в чергу. При цьому він ставить самого себе у якості отримувача цього повідомлення. Коли настає час, система дістає повідомлення з черги та відправляє його адресату на обробку.

Клас Handler дає дві корисні можливості:

- реалізувати відкладене у часі виконання коду;
- виконання коду не у своєму потоці.

Клас AsyncTask пропонує простий та зручний механізм для переміщення операцій, які потребують багато ресурсів, у фоновий потік. Завдяки цьому класу можна отримати зручну синхронізацію обробників подій з графічним потоком, що дозволяє оновлювати елементи інтерфейсу користувача для звіту про хід виконання завдання чи для виводу результатів при завершенні завдання.

За допомогою AsyncTask створюються, синхронізуються, а також керуються потоки, що дозволяє створювати асинхронні задачі, які складаються з операцій, що

виконуються у фоновому режимі, та оновлювати інтерфейс користувача, коли вони завершуються.

### **3.4 Існуючий аналог розроблюваної системи**

IT-Enterprise.SmartManager — це мобільний додаток, який дозволяє здійснювати контроль над проектом з мобільного пристрою з операційною системою Android. Цей додаток відображає усі поточні задачі, які впорядковані за пріоритетом і терміном виконання. Можна проаналізувати хто відповідальний за завдання, у який час їх потрібно виконати чи уточнити у керівника деякі аспекти. У план робіт співробітника автоматично додаються задачі, які є стадіями виконання будь-якого бізнес-процесу системи (перевірити, узгодити, затвердити документ тощо). При цьому спеціаліст має легкий доступ до усіх пов'язаних з задачею документів, що допомагає йому швидко та якісно виконувати свою роботу.

Додаток дозволяє працювати над задачами сумісно, обговорювати, делегувати завдання, уточнювати позиції перед узгодженням. Безпосередньо у додатку можна додавати, видаляти, переглядати вкладені файли (скановані документи, таблиці з розрахунками, тексти, фото тощо).

SmartManager для керівника — це, з одного боку, інструмент ефективного керування робочим часом і власною продуктивністю, а з іншого боку, це незамінний інструмент контролю усіх своїх підлеглих та підвищення загальної продуктивності підприємства. Зображення роботи даного мобільного додатку наведені на рисунках 3.3-3.5.



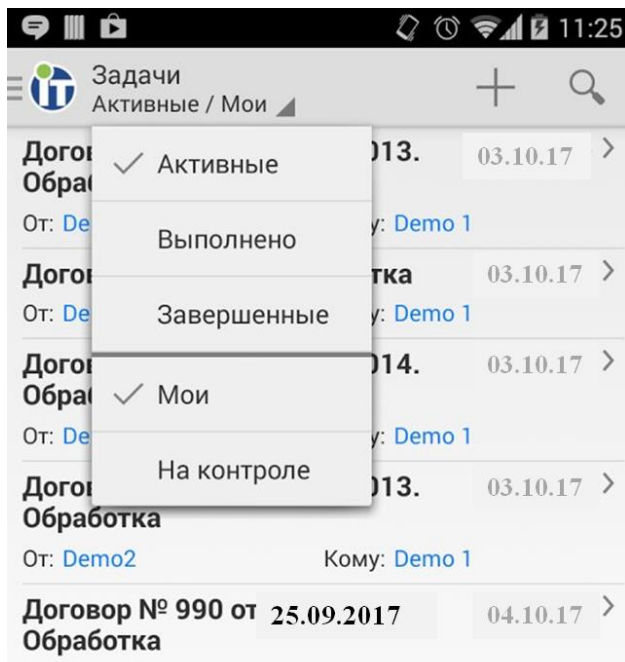


Рисунок 3.3 – Сортування завдань

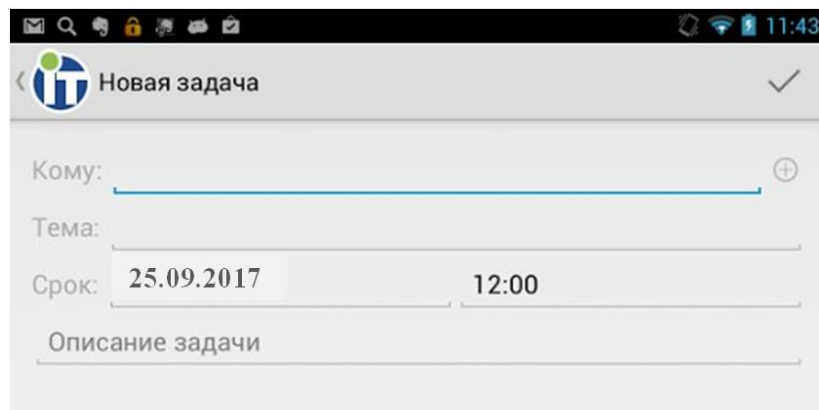


Рисунок 3.4 – Створення нової задачі

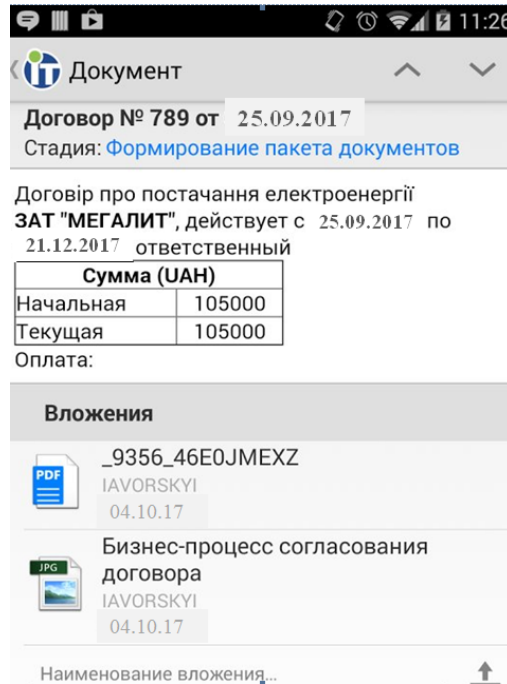


Рисунок 3.5 – Работа з вкладеними файлами

### 3.5 Опис реалізації

Для побудови діаграм була обрана бібліотека AChartEngine. Вона дає можливість досить зручно відображати інформацію щодо діяльності користувача. Для побудови, наприклад, кругової діаграми потрібно виконати наступні дії:

- Підключити бібліотеку.
- Створити та ініціювати три масиви, у першому будуть зберігатися кольори, у другому – підписи, у третьому – значення.
- Створити об'єкт CategorySeries і завантажити у нього масив строк з підписами та значення. Ці дані будуть використані при побудові діаграм.
- Створити об'єкт DefaultRenderer.
- Для кожного кольору у діаграмі створити SimpleSeriesRenderer і встановити його у цей колір. Потім кожен SimpleSeriesRenderer додати у об'єкт DefaultRenderer.

- Створити об'єкт `PieChartIntent` і передати йому `CategorySeries` та `DefaultRenderer`.

Подібним чином будуються усі інші діаграми.

### **3.6 Опис тестового матеріалу**

Під час написання дипломної роботи був розроблений мобільний додаток, який створено для проведення моніторингу прогресу проектів, у яких задіяний користувач. Додаток зберігає та регулярно оновлює певні дані стосовно роботи користувача, необхідні для функціонування, а саме для ведення статистики та її графічного відображення у вигляді ряду раніше перерахованих діаграм. Окрім відображення статистичних даних мобільний додаток надаватиме також деякі можливості з керування власною діяльністю, наприклад, підписатися на отримання оновлень щодо конкретних аспектів проекту, запропонувати або назначити завдання своїм колегам, перевірити статус власних завдань та переглянути їх.

### **3.7 Ілюстрації роботи програми**

Під час першого запуску додатку користувач має можливість ввести свої ім'я користувача та пароль, що забезпечить подальшу можливість автоматичного завантаження та регулярного оновлення даних (додаток Б). Дана функція показана на рисунку 3.7.

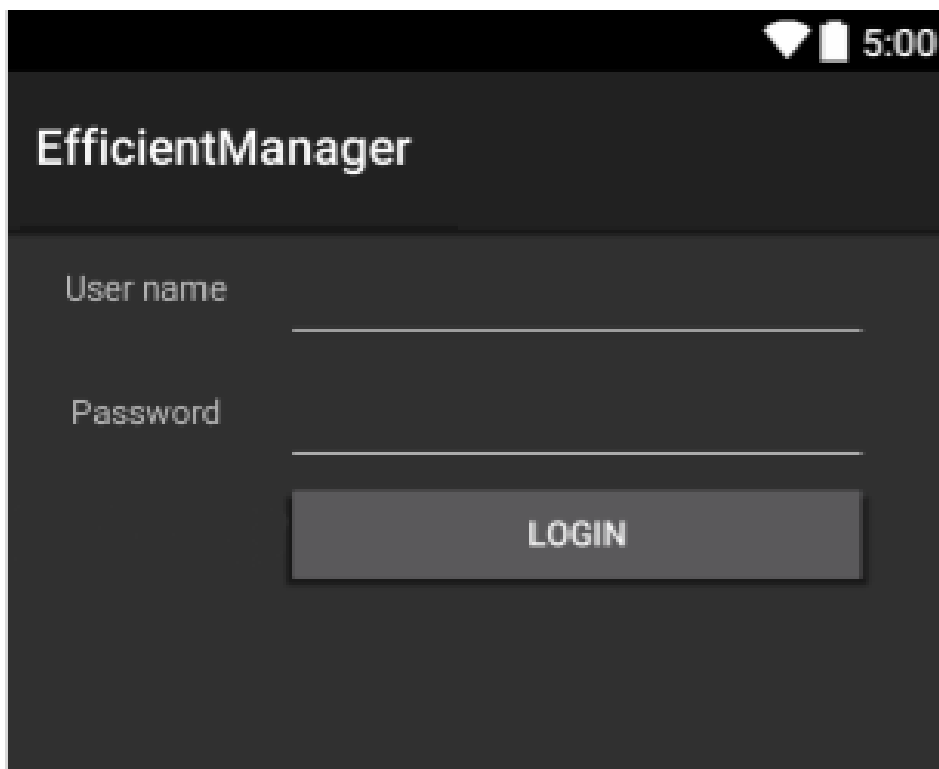


Рисунок 3.6 – Екран входу у свій обліковий запис

Після здійснення входу у свій обліковий запис одразу дозволяється обрати, яку інформацію потрібно відобразити графічно. Дана функція показана на рисунку 3.7.

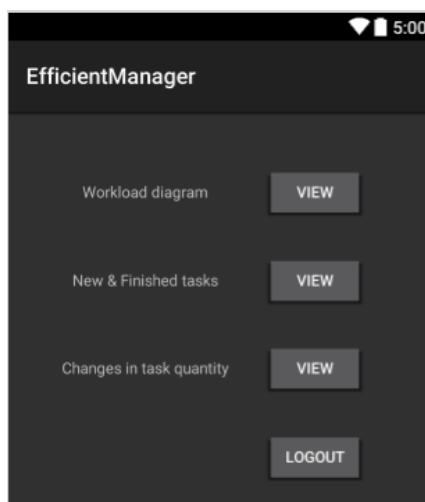


Рисунок 3.7 – Головний екран мобільного додатку

На рисунках 3.8 – 3.10 зображені відповідно кругова діаграма робочого навантаження, діаграма співвідношення створених та закінчених завдань та гістограма завдань під спостереженням користувача за останні 7 або 30 днів.

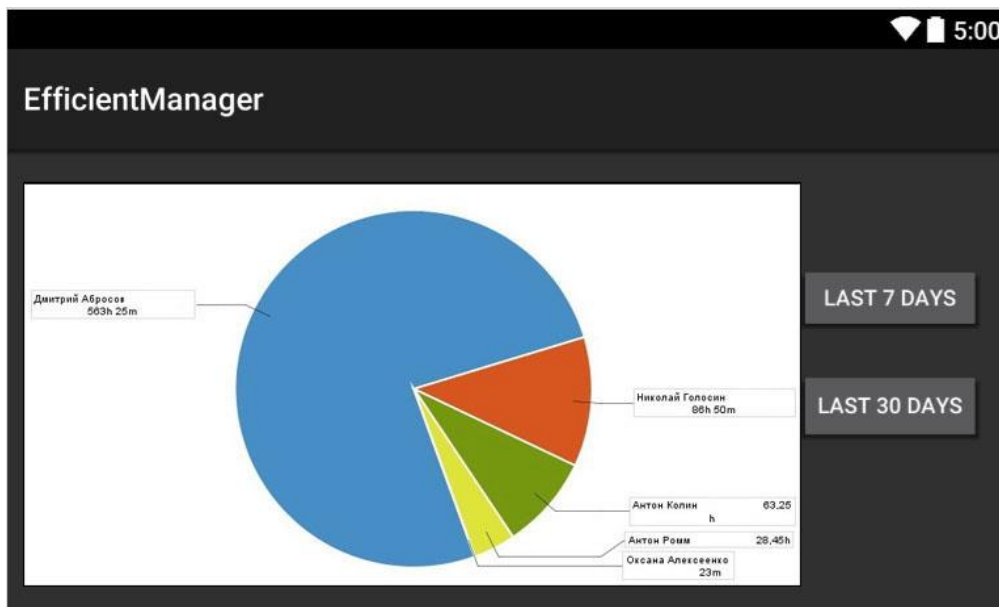


Рисунок 3.8 – Діаграма робочого навантаження

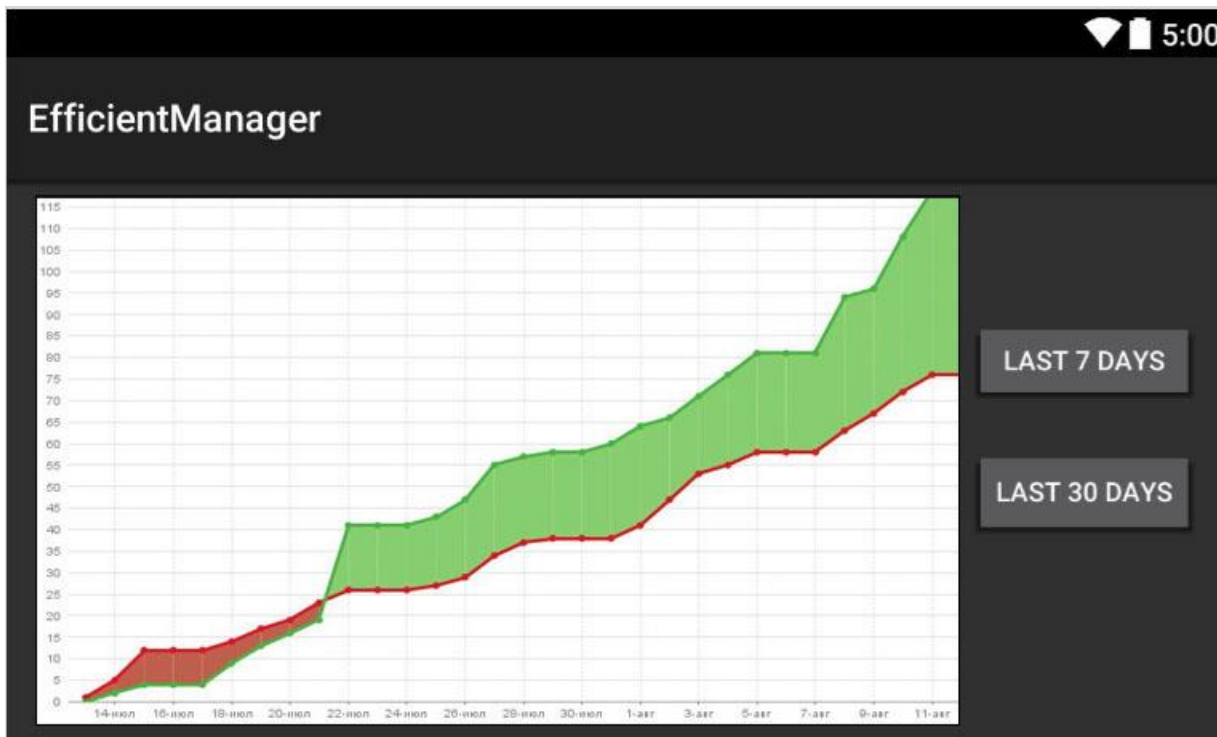


Рисунок 3.9 – Діаграма створених та закінчених завдань

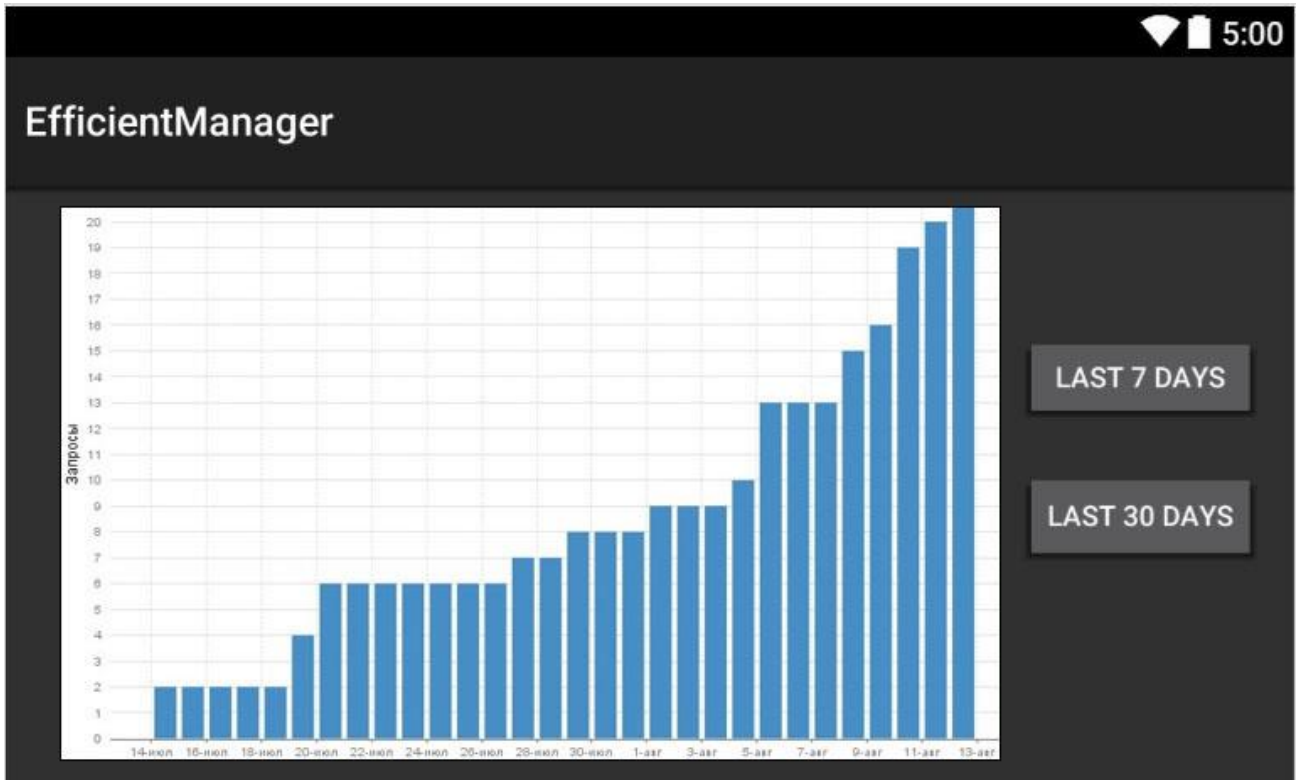


Рисунок 3.10 – Гістограма кількості завдань користувача

## **4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної роботи бакалавра була розробка мобільного додатку для проведення моніторингу прогресу проектів, у яких задіяний користувач. Аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера, на якому буде виконуватися розробка.

### **4.1 Загальні питання з охорони праці**

Санітарно-побутові умови, умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

#### **4.1.1 Правові та організаційні основи охорони праці**

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави, і особистої відповідальності працівників.

### 4.1.2 Організаційно-технічні заходи з безпеки праці

В організації/підприємстві проводиться навчання і перевірка знань з питань охорони праці відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 N 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за N 231/10511 НПАОП 0.00-4.12-05 [18].

## 4.2 Аналіз стану умов праці

Робота над створенням мобільного додатку проходитиме в приміщенні багатоквартирного будинку. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

### 4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1.

Таблиця 4.1 – Розміри приміщення

Найменування	Значення
Довжина, м	5
Ширина, м	5
Висота, м	2.7
Площа, м <sup>2</sup>	25
Об'єм, м <sup>3</sup>	67.5

Згідно з [17] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.



#### 4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [15] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 ÷ 800
Висота простору для ніг, мм	700	не менше 600
Ширина простору для ніг, мм	550	не менше 500
Глибина простору для ніг, мм	750	не менше 650
Висота поверхні сидіння, мм	450	400 ÷ 500
Ширина сидіння, мм	420	не менше 400
Глибина сидіння, мм	450	не менше 400
Висота поверхні спинки, мм	350	не менше 300
Ширина опорної поверхні спинки, мм	400	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	780	700 ÷ 800

Приміщення кабінету знаходиться на четвертому поверсі чотирьох поверхової будівлі і має об'єм 67.5 м<sup>3</sup>, площу – 25 м<sup>2</sup>. Температура в приміщенні протягом року коливається у межах 17–26°C, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум в лабораторії знаходиться на рівні 50 дБА. Система вентилявання приміщення — природна неорганізована, а опалення — централізоване.

#### 4.2.3 Навантаження та напруженість процесу праці

За фізичним навантаженням виконання випускної роботи бакалавра відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він

підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи наведені в [15].

Роботу над дипломним проектом визнано, такою, що займає 50% часу робочого дня та при восьмигодинній робочій зміні рекомендовано встановити додаткові регламентовані перерви - для розробників програм тривалістю 15 хв. через кожну годину роботи.

### **4.3 Виробнича санітарія**

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

#### **4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу**

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00.-1.28-10 [17], які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга  $U=+220\text{В} \pm 5\%$ ;
- робочий струм  $I=2\text{А}$ ;
- споживана потужність  $P=350\text{ Вт}$ .

Таблиця 4.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
<b>фізичні</b>			
- підвищена температура поверхонь обладнання	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	2	[17]
- підвищений рівень шуму на робочому місці	-//-	2	[16]
- підвищений рівень вібрації	-//-	2	[17]
- підвищена або знижена вологість повітря	-//-	2	[17]
- підвищена або знижена рухливість повітря	-//-	1	[17]
- підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	-//-	4	[28] [17]
- підвищений рівень статичної електрики	-//-	2	[28]
- підвищена напруженість електричного поля	-//-	2	[27]
- недостатність природного світла	порушення умов праці (вимог до приміщень)	2	[14]
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	[14]
- підвищена яскравість світла	порушення умов праці (організації місця праці-налагодження моніторів)	1	[15]
- понижена контрастність	-//-	1	[15]
<b>психофізіологічні:</b>			
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів;- виконання роботи за темою диплома, тестування; - оформлення роботи	4	[29] [15]
- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці - сидіння користувача) та організації робочого часу - безпервна робота)	2	[29] [15]

Робочі місця мають відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [15].

#### **4.3.2 Пожежна безпека**

Небезпека розвитку пожежі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, вентиляції і кондиціонування.

Передбачаються способи і засоби запобігання утворення (або внесення) в горюче середовище джерел запалювання, таких як:

- 1) застосування електроустаткування, відповідної пожежонебезпечної і вибухонебезпечної зонам відповідно до ПУЕ;
- 2) застосування в конструкції швидкодійних засобів захисного відключення можливих джерел запалення;
- 3) виключення можливості появи іскрового розряду в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол [13].

#### **4.3.3 Електробезпека**

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого

замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

#### **4.4 Гігієнічні вимоги до параметрів виробничого середовища**

##### **4.4.1 Мікроклімат**

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. Оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають [16] і наведені в табл. 4.4:

Таблиця 4.4 – Норми мікроклімату робочої зони об'єкту

<b>Період року</b>	<b>Категорія робіт</b>	<b>Температура С0</b>	<b>Відносна вологість %</b>	<b>Швидкість руху повітря, м/с</b>
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

Дане приміщення обладнане системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією. У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря у відповідності до [16]. Рівні позитивних і негативних іонів у повітрі мають відповідати [16].

Контроль параметрів мікроклімату в холодний і теплий період року здійснюється не менше 3-х разів на зміну (на початку, середині, в кінці).

#### 4.4.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає [14]. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення.

*Розрахунок освітлення.*

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше  $1/8$ , в побутових –  $1/10$ :

$$S_b = \left( \frac{1}{5} \div \frac{1}{10} \right) \cdot S_n, \quad (4.1)$$

де  $S_b$  – площа віконних прорізів,  $m^2$ ;

$S_n$  – площа підлоги,  $m^2$ .

$$S_n = a \cdot b = 5 \cdot 5 = 25 \text{ м}^2,$$

$$S = 1/8 \cdot 25 = 3,125 \text{ м}^2.$$

Приймаємо 2 вікна площею  $S=1,6 \text{ м}^2$  кожне.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні.

Розрахунок кількості світильників  $n$  виробляється по формулі (4.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \quad (4.2)$$

де  $E$  – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

$S$  – освітлювана площа,  $\text{м}^2$ ;  $S = 25 \text{ м}^2$ ;

$Z$  – поправочний коефіцієнт світильника ( $Z = 1,15$  для ламп розжарювання та ДРЛ;  $Z = 1,1$  для люмінесцентних ламп) приймаємо рівним 1,1;

$K$  – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

$U$  – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

$M$  – число люмінесцентних ламп в світильнику – 2;

$F$  – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 \cdot 25 \cdot 1,1 \cdot 1,5}{5400 \cdot 0,575 \cdot 2} \approx 2,0$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

### **4.4.3 Шум та вібрація, електромагнітне випромінювання**

Рівень шуму, що супроводжує роботу користувачів персональних комп'ютерів (зумовлений як роботою системних блоків, клавіатури, так і друкуванням на принтерах, а також зовнішніми чинниками), коливається у межах 50–65 дБА [16]. У залах опрацювання інформації та комп'ютерного набору рівні шуму не повинні перевищувати 65 дБА.

Віброізоляція можливо здійснювати за допомогою спеціальної прокладки під системний блок, який послаблює передачу вібрацій робочого столу. Вібрація на робочому місці в приміщенні, що розглядається, відповідає нормам [16].

### **4.4.4 Вентилювання**

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при  $V$  приміщення  $> 40 \text{ м}^3$  на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП.

Також має здійснюватися провітрювання приміщення в залежності від погодних умов. Тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

## **4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій**

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання наводимо приклади деяких заходів безпеки.

*1) Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:*



- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала  $2/3$  нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціонування повітря.

*2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:*

- постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів, а також мережевих кабелів живлення, за допомогою яких електроприлади під'єднують до електромережі;
- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;
- не тягнути за мережевий кабель, щоб витягти вилку з розетки;
- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;
- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;
- не залишати включені електроприлади без нагляду;
- не допускати потрапляння всередину електроприладів крізь вентиляційні отвори рідин або металевих предметів, а також не закривати їх та підтримувати в належній чистоті, щоб уникнути перегрівання та займання приладу;
- не ставити на електроприлади матеріали, які можуть під дією теплоти, що виділяється, загорітися (канцелярські товари, сувенірну продукцію тощо).

### Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом [22], приміщення в якому проводяться всі роботи відноситься до першого класу (без підвищеної небезпеки). Коефіцієнт використання вертикальних заземлювачів  $\eta_v$  в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4...0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги) враховують за допомогою коефіцієнта використання горизонтального заземлювача  $\eta_c$ .

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів  $R_{шт.з.}$ :

$$R_{шт.з.} = \frac{R_d \cdot R_{пр.з.}}{R_{пр.з.} - R_d}, \quad (4.3)$$

де  $R_{пр.з.}$  – опір природних заземлювачів;  $R_d$  – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то  $R_{шт.з.} = R_d$ .

Підставивши числові значення у формулу (А.3), отримуємо:

$$R_{шт.з.} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту  $\rho$ , Ом·м. Приблизне значення питомого опору глини приймаємо  $\rho = 40$  Ом·м (табличне значення).

3) Розрахунковий питомий опір ґрунту,  $\rho_{розр.}$ , Ом·м, визначається відповідно для вертикальних заземлювачів  $\rho_{розр.в.}$ , і горизонтальних  $\rho_{розр.г.}$ , Ом·м за формулою:

$$\rho_{розр.} = \psi \cdot \rho, \quad (4.4)$$

де  $\psi$  – коефіцієнт сезонності для вертикальних заземлювачів І кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів  $\rho_{розр.в.} = 1,7$  і горизонтальних  $\rho_{розр.г.} = 5,5$  Ом·м.

$$\rho_{\text{розр.в}} = 1,7 \cdot 40 = 68 \text{ Ом}\cdot\text{м}$$

$$\rho_{\text{розр.г}} = 5,5 \cdot 40 = 220 \text{ Ом}\cdot\text{м}$$

4) Розраховується опір розтікання струму вертикального заземлювача  $R_{\text{в}}$ , Ом, за формулою (4.5).

$$R_{\text{в}} = \frac{\rho_{\text{розр.в}}}{2 \cdot \pi \cdot l_{\text{в}}} \cdot \left( \ln \frac{2 \cdot l_{\text{в}}}{d_{\text{ст}}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_{\text{в}}}{4 \cdot t - l_{\text{в}}} \right), \quad (4.5)$$

де  $l_{\text{в}}$  – довжина вертикального заземлювача (для труб - 2–3 м;  $l_{\text{в}}=3$  м);

$d_{\text{ст}}$  – діаметр стержня (для труб - 0,03–0,05 м;  $d_{\text{ст}}=0,05$  м);

$t$  – відстань від поверхні землі до середини заземлювача, яка визначається за формулою (4.6):

$$t = h_{\text{в}} + \frac{l_{\text{в}}}{2}, \quad (4.6)$$

де  $h_{\text{в}}$  – глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м}$$

$$R_{\text{в}} = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left( \ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

5) Визначається теоретична кількість вертикальних заземлювачів  $n$  штук, без урахування коефіцієнта використання  $\eta_{\text{в}}$ :

$$n = \frac{2 \cdot R_{\text{в}}}{R_{\text{д}}} = \frac{2 \cdot 18,5}{4} = 9,25 \quad (4.7)$$

І визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки  $\eta_{\text{в}} = 0,57$  (табличне значення).

б) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання  $n_{\text{в}}$ , шт:

$$n_B = \frac{2 \cdot R_B}{R_d \cdot \eta_B} = \frac{2 \cdot 18,5}{4 \cdot 0,57} = 16,2 \approx 16 \quad (4.8)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача  $l_c$ , м:

$$l_c = 1,05 \cdot L_B \cdot (n_B - 1), \quad (4.9)$$

де  $L_B$  – відстань між вертикальними заземлювачами, (прийняти за  $L_B = 3$  м);  
 $n_B$  – необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м}$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки)  $R_r$ , Ом:

$$R_r = \frac{\rho_{\text{розр.г}}}{2 \cdot \pi \cdot l_c} \cdot \ln \frac{2 \cdot l_c^2}{d_{\text{см}} \cdot h_r}, \quad (4.10)$$

де  $d_{\text{см}}$  – еквівалентний діаметр смуги шириною  $b$ ,  $d_{\text{см}} = 0,95b$ ,  $b = 0,15$  м;

$h_r$  – глибина закладання горизонтальних заземлювачів (0,5 м);

$l_c$  – довжина з'єднувальної стрічки горизонтального заземлювача  $l_c$ , м

$$R_r = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом}$$

9) Визначається коефіцієнт використання горизонтального заземлювача  $\eta_c$  відповідно до необхідної кількості вертикальних заземлювачів  $n_B$ .

Коефіцієнт використання з'єднувальної смуги  $\eta_c = 0,3$  (табличне значення).

10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{зар}} = \frac{R_B \cdot R_r}{R_B \cdot \eta_c + R_r \cdot n_B \cdot \eta_B} \leq R_d. \quad (4.18)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова:  $R_{\text{зар}} < 4$  Ом, а саме:

$$R_{\text{зар}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_{\text{д}}$$

1) При виникненню пожеж при роботі на ПЕОМ від таких можливими джерел запалювання як:

- іскри і дуги коротких замикань;
- перегрів провідників, резисторів та інших радіодеталей ПЕОМ, від тривалої перевантаження та наявність перехідного опору;
- іскри при розмиканні і розмиканні ланцюгів;
- розряди статичної електрики;
- необережному поводженню з вогнем, а також вибухи газо-повітряних і паро-повітряних сумішей.

#### **Висновки до розділу 4**

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом; описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведено рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

## ВИСНОВКИ

У результаті виконання дипломної роботи був розроблений і програмно реалізований мобільний додаток для проведення моніторингу прогресу проектів, у яких задіяний користувач.

Для реалізації програми були отримані необхідні для обробки статистичні дані (кількість активних та закінчених завдань, загальний час роботи над проектом, інформація про проект та інших членів команди користувача); розроблено статистичну модель для роботи з даними; було реалізовано отриману модель.

Отримана інформація оброблюється за допомогою статистичних методів та представляється у вигляді зручних та зрозумілих діаграм, які відображають найважливіші аспекти проекту та прогрес роботи.

Подальше тестування показало задовільну якість та точність отриманих результатів, продемонструвало працездатність готового програмного продукту та його відповідність меті дипломної роботи. Отриманий мобільний додаток після незначних вдосконалень може бути у майбутньому задіяний у роботі над проектами.

Був проведений аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. За отриманими результатами описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним для робітника. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Карасёва Л. А. Статистика Всемирная история экономической мысли: В 6 томах [Текст]. Т. 1. От зарождения экономической мысли до первых теоретических систем политической жизни / Л. А. Карасёва — М.: Мысль, 1987 — 606 с.
2. Кон М. Scrum: гибкая разработка ПО [Текст] : пер. с англ. / М. Кон – К. : Ваклер — М.: «Вильямс», 2011. — 576 с.
3. ДеГрейс П, Злі проблеми, справедливі рішення [Текст] : пер. с англ. / П. ДеГрейс – С. : Воронин — М.: Series, 1990. – 344 с.
4. Бидл М. Разработка ПО применением Scrum [Текст] : пер. с англ. /М. Бидл – П. : Кислов — М.: Prentice Hall, 2002. – 380 с.
5. Schwaber К. Agile Project Management with Scrum [Текст]/ К. Schwaber — М.: Microsoft Press, 2004. – 163 с.
6. Книберг Х. Scrum и XP: заметки с передовой [Текст] : пер. с англ. / Х. Книберг — М.: С4Media, 2007. — 140 с.
7. Project Management Software [Электронный ресурс]/ Режим доступа: [www/ URL : http://www.techopedia.com/definition/13132/project-management-software](http://www.techopedia.com/definition/13132/project-management-software)
8. Орлов А. И. Прикладна статистика. Підручник. [Текст] / А. И. Орлов — М.: Іспит, 2006. — 671 с.
9. Миклашевский И.Н. Энциклопедический словарь Брокгауза и Ефрона : в 86 т. [Текст] Т.82 Статистика теоретическая. — М.: СПб., 1890—1907 — 527 с.
10. Желязны Д. Говори на языке диаграмм: Пособие по визуальным коммуникациям для руководителей [Текст]: пер. с англ./ Д. Желязны – К. : Ваклер — М.: Институт комплексных стратегических исследований, 2004. – 220 с.
11. Фримен А. ASP.NET MVC 4 с примерами на C# 5.0 для профессионалов, 4-е издание. [Текст] / А. Фримен — М.: «Вильямс», 2013. — 688 с.

12. About Android OS in few words [Электронный ресурс]/ Режим доступа: [www/URL : http://developer.android.com/guide/basics/what-is-android.html](http://developer.android.com/guide/basics/what-is-android.html)
13. ГОСТ 12.1.044-89 ССБТ. Пожаровзрывобезопасность веществ и материалов. Номенклатура показателей и методы их определения
14. ДБН В.2.5-28:2015 Природное и искусственное освещение
15. ДСанПіН 3.3.2.007-98 Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин
16. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку
17. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих
18. НПАОП 0.00-4.12-05 Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці
19. НПАОП 0.00-4.15-98 Про розробку інструкцій з охорони праці
20. НПАОП 0.00-6.03-93 Порядок опрацювання та затвердження власником нормативних актів про охорону праці
21. НАПБ Б.03.002-2007 Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою
22. НПАОП 40.1-1.01-97 Правила безопасной эксплуатации электроустановок
23. НПАОП 40.1-1.32-01 Правила устройства электроустановок. Электрооборудование специальных установок
24. ДСН 3.3.6.039-99 Санітарні норми виробничої загальної та локальної вібрації
25. ДСТУ ГОСТ 12.1.012-90 ССБТ. Вібраційна безпека. Загальні вимоги
26. ДБН В.2.5-67:2013 Опалення, вентиляція та кондиціонування
27. ГОСТ 12.1.006-84 ССБТ. Электромагнитные поля радиочастот. Общие требования безопасности. Допустимые уровни на рабочих местах и требования к проведению контроля



28. ГОСТ 12.1.030-81 ССБТ. Електробезпе́чність. Захи́сне заземле́ння.  
Зануле́ння
29. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації  
електронно-обчислювальних машин

## ДОДАТОК А

### Комп'ютерна презентація

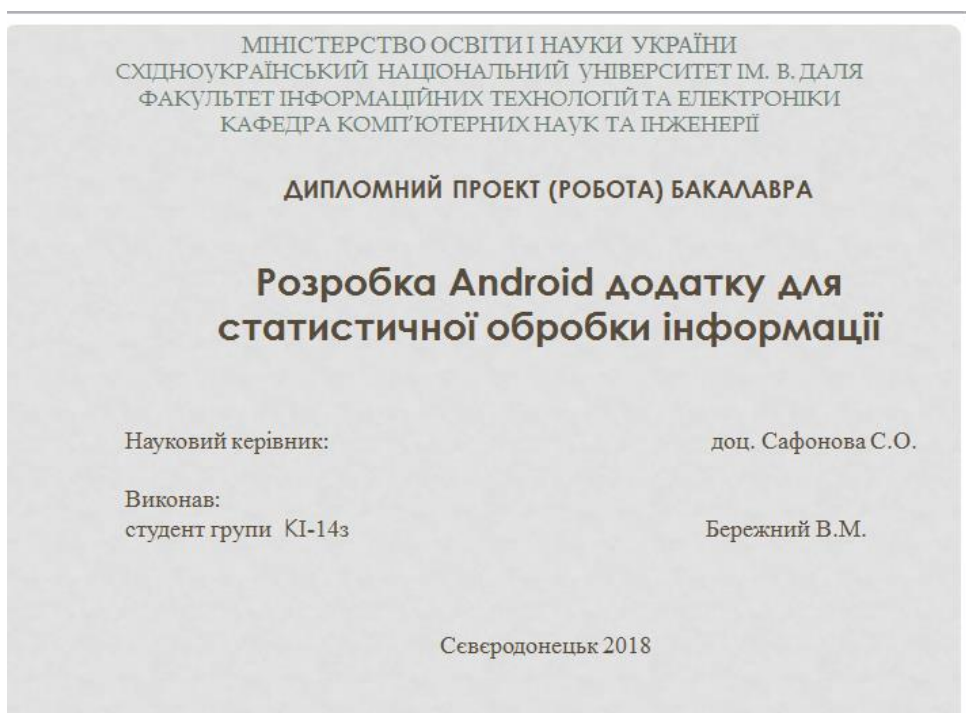


Рисунок А.1 – Слайд №1

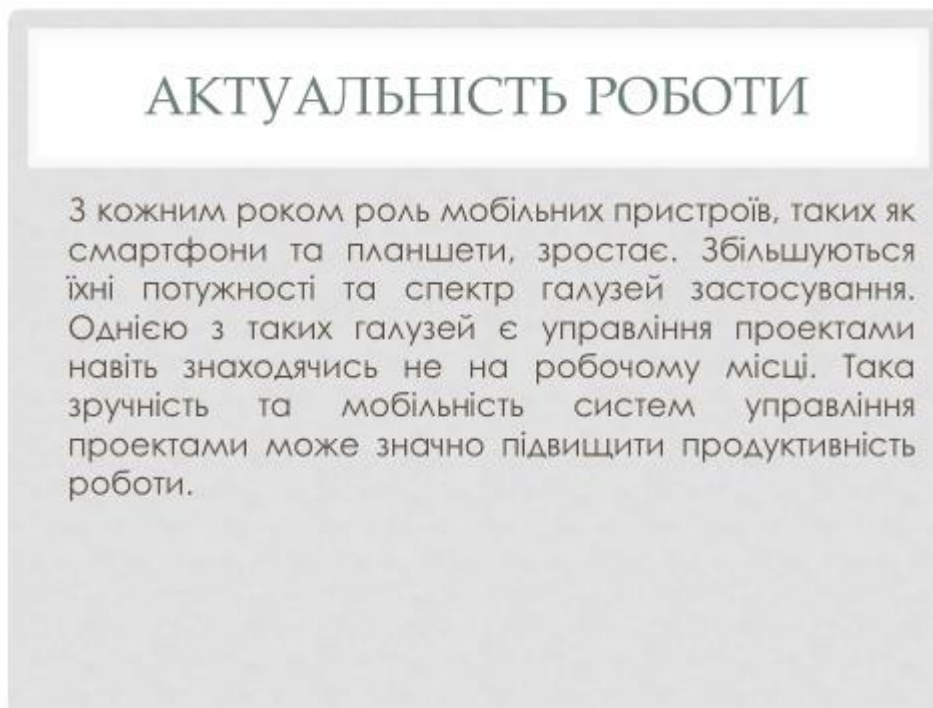


Рисунок А.2 – Слайд №2

## ПОСТАНОВКА ЗАДАЧІ

Метою даної дипломної роботи є розробка програмного засобу, що буде здійснювати статистичну обробку та графічне відображення інформації. Цю задачу можна розділити на наступні кроки:

- отримати необхідні для обробки статистичні дані;
- дослідити отримані статистичні дані для їх наступної обробки;
- розробити статистичну модель для роботи з даними;
- реалізувати отриману модель та отримати результати.



Рисунок А.3 – Слайд №3

## ПРОГРАМНА РЕАЛІЗАЦІЯ

Шаблон проектування MVC — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

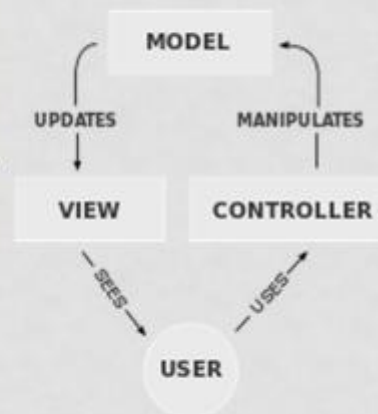


Рисунок А.4 – Слайд №4



Рисунок А.5 – Слайд №5

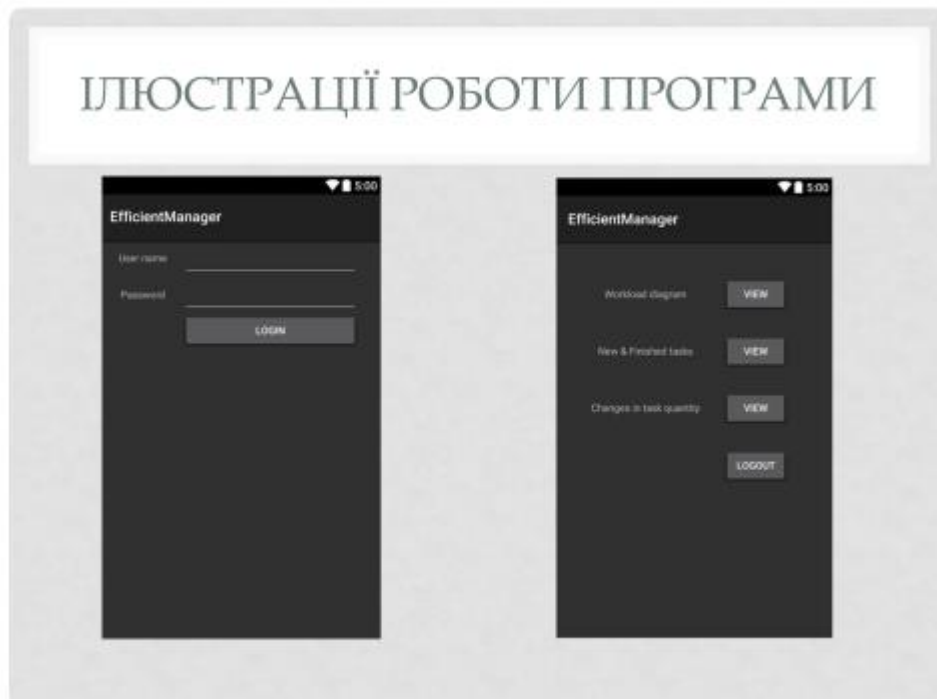


Рисунок А.6 – Слайд №6

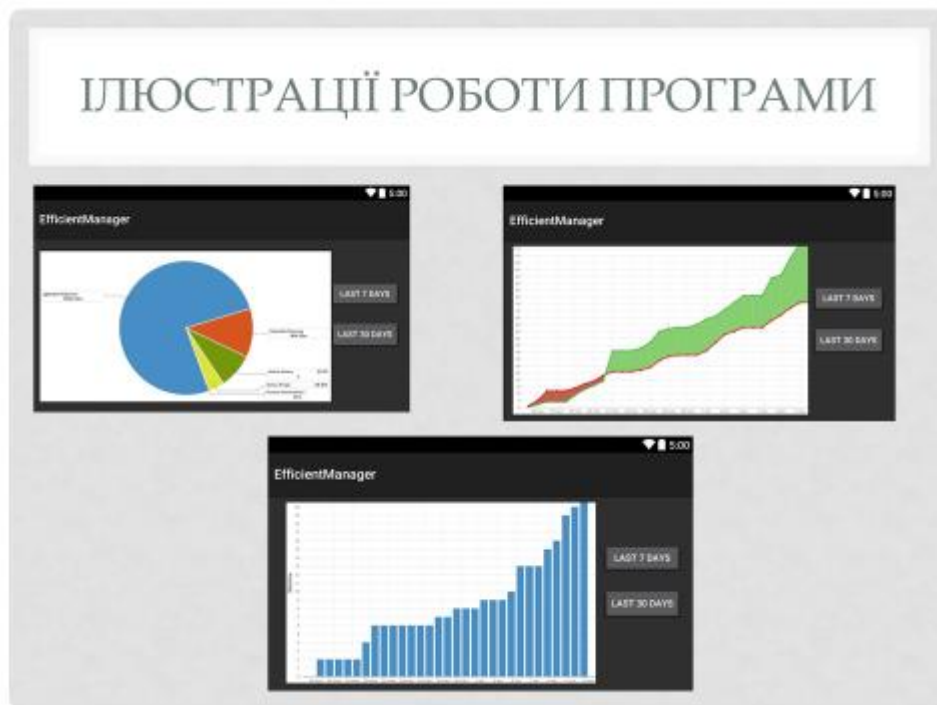


Рисунок А.7 – Слайд №7

## ОХОРОНА ПРАЦІ

- Був проведений аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. За отриманими результатами описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним для робітника. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Рисунок А.8 – Слайд №8

## ВИСНОВКИ

У результаті виконання дипломної роботи був розроблений і програмно реалізований мобільний додаток для проведення моніторингу прогресу проектів, у яких задіяний користувач.

Для реалізації програми були отримані необхідні для обробки статистичні дані (кількість активних та закінчених завдань, загальний час роботи над проектом, інформація про проект та інших членів команди користувача); розроблено статистичну модель для роботи з даними; було реалізовано отриману модель.

Отримана інформація оброблюється за допомогою статистичних методів та представляється у вигляді зручних та зрозумілих діаграм, які відображають найважливіші аспекти проекту та прогрес роботи.

Тестування показало задовільну якість та точність отриманих результатів, продемонструвало працездатність готового програмного продукту та його відповідність меті дипломної роботи. Мобільний додаток після незначних вдосконалень може бути у майбутньому задіяний у роботі над проектами.

Рисунок А.9 – Слайд №9

Дякую за увагу

Рисунок А.10 – Слайд №10

## ДОДАТОК Б

## Лістинг програми HibernateUtil.java

```

1     package home.dbms.persistance;
2     import home.dbms.persistance.entities.domain.Pencil;
3     import java.util.Date;
4     import java.util.HashMap;
5     import java.util.Map;
6     import org.apache.commons.logging.Log;
7     import org.apache.commons.logging.LogFactory;
8     import org.hibernate.Session;
9     import org.hibernate.SessionFactory;
10    import
11    org.hibernate.boot.registry.StandardServiceRegistryBuilder;
12    import org.hibernate.cfg.Configuration;
13    import org.hibernate.service.ServiceRegistry;
14    public class HibernateUtil {
15        public static final Log log =
16    LogFactory.getLog(HibernateUtil.class);
17        private static SessionFactory sessionFactory;
18        private static HibernateUtil instance;
19        private static ServiceRegistry serviceRegistry;
20        public static Map<String, String> connectionSettings =
21    new HashMap<String, String>();
22        public static void addAnnotatedClasses(Configuration
23    config) {
24            config.addAnnotatedClass(Pencil.class);
25        }
26        public static synchronized SessionFactory
27    getSessionFactory() {
28            if (sessionFactory == null) {
29                createSessionFactory();
30            }
31            return sessionFactory;
32        }
33        private static void createSessionFactory()
34            throws ExceptionInInitializerError {
35            try {
36                Date d1 = new Date();
37                System.out.println("HibernateUtil: Opening DB
38    connection.");
39                Configuration config = new Configuration();
40                HibernateUtil.addAnnotatedClasses(config);
41                config.configure();
42                connectionSettings.put("url",
43    config.getProperty("hibernate.connection.url"));

```

```

44         connectionSettings.put("username",
45         config.getProperty("hibernate.connection.username"));
46         connectionSettings.put("password",
47         config.getProperty("hibernate.connection.password"));
48         System.out.println("URL: " +
49         connectionSettings.get("url"));
50         System.out.println("USER: " +
51         connectionSettings.get("username"));
52         System.out.println("PASSWORD: "
53         + connectionSettings.get("password"));
54         serviceRegistry = new
55         StandardServiceRegistryBuilder()
56         .applySettings(config.getProperties()).build();
57         sessionFactory =
58         config.buildSessionFactory(serviceRegistry);
59         checkConnection(d1);
60     } catch (Exception ex) {
61         System.err.println("Failed to create
62         sessionFactory object." + ex);
63         log.error(ex.getMessage(), ex);
64         throw new ExceptionInInitializerError(ex);
65     }
66     public static void checkConnection() {
67         Session s = sessionFactory.openSession();
68         // s.beginTransaction();
69         s.getTransaction().begin();
70         s.close();
71         s = null;
72     private static void checkConnection(Date d1) {
73         try {
74             checkConnection();
75         } catch (Exception e) {
76             System.out.println("Opening a DB
77         connection... Failed");
78             System.out.println(e);
79             return;
80             System.out.println(String.format(
81             "Connection was successfully opened in
82             %d seconds",
83             (new Date().getTime() - d1.getTime()) /
84             1000));
85     public static HibernateUtil getInstance() {
86         if (instance == null) {
87             instance = new HibernateUtil();
88         }
89         return instance;
90     public static void shutdown() {
91         if (sessionFactory != null) {
92             sessionFactory.close();
93             sessionFactory = null;
94     }

```



## ДОДАТОК В

### Лістинг програми IndexBean.java

```

1  package home.dbms.web.bean.peges;
2  import home.dbms.persistence.HibernateUtil;
3  import home.dbms.persistence.entities.domain.Pensil;
4  import home.dbms.web.bean.PageBean;
5  import java.util.List;
6  import javax.faces.bean.ManagedBean;
7  import javax.faces.bean.ViewScoped;
8  import org.hibernate.Criteria;
9  import org.hibernate.Session;
10 import org.hibernate.criterion.Projections;
11 @ManagedBean
12 @ViewScoped
13 public class IndexBean extends PageBean {
14     private String hellow;
15     private Pensil pensil;
16     private Long count;
17     private List<Pensil> pensils;
18     @Override
19     protected void onConstructor() {
20         hellow = "Hellow";
21         initPensil();
22     }
23     private void initPensil(){
24         Criteria criteria =
25 HibernateUtil.getSessionFactory().openSession().createCriteria(P
26 ensil.class);
27         pensils = criteria.list();
28         criteria.setProjection(Projections.countDistinct("id"));
29         count =
30 Long.parseLong(criteria.uniqueResult().toString());
31         pensil = new Pensil();
32     }
33     public void save() {
34         Session session = null;
35         try {
36             session =
37 HibernateUtil.getSessionFactory().openSession();
38             session.beginTransaction();
39             session.save(pensil);
40             session.getTransaction().commit();
41         } catch (Exception e) {
42             e.printStackTrace();
43         } finally {
44             if (session != null && session.isOpen()) {

```

```
45         session.close();
46     }
47 }
48     initPencil();
49 }
50 public String getHellow() {
51     return hellow;
52 }
53 public void setHellow(String hellow) {
54     this.hellow = hellow;
55 }
56 public Pencil getPencil() {
57     return pencil;
58 }
59 public void setPencil(Pencil pencil) {
60     this.pencil = pencil;
61 }
62 public List<Pencil> getPencils() {
63     return pencils;
64 }
65 public void setPencils(List<Pencil> pencils) {
66     this.pencils = pencils;
67 }
68 public Long getCount() {
69     return count;
70 }
71 public void setCount(Long count) {
72     this.count = count;
73 }
```

