



Приватний заклад вищої освіти
Одеський технологічний університет «ШАГ»
Кафедра інформаційних технологій та фундаментальної
підготовки

Випускна кваліфікаційна робота бакалавра

«Інформаційна система для управління завданнями у команді»

на здобуття бакалаврського рівня вищої освіти
зі спеціальності «122 Комп'ютерні науки»

Виконавці проекту

№	П.І.Б.	Група
1	Соломончук Станіслав Ігорович	КН-П-202
2	Романенко Ярослав Юрійович	КН-П-202
3	Белялов Омар Наріманович	КН-П-202
4	Калиніна Любов Андріївна	Outsource

Автор звіту

_____ Белялов Омар Наріманович

АНОТАЦІЯ

УДК 004.9

Д-30

Беялов О.Н. , Соломончук С. І., Романенко Я. Ю. - Система управління задачами для мобільних пристроїв

: Автореферат випускної кваліфікаційної роботи на здобуття бакалаврського рівня вищої освіти зі спеціальності «122 Комп'ютерні науки». - Одеса: ОТУШ, 2024 - 20 с.

Робота присвячена дослідженням, розвиненню та впровадженню інформаційної системи, головним призначенням якої є розробка мобільного додатку для планування та синхронізації командної роботи над проектами. Додаткові функції системи включають реєстрацію, авторизацію та автентифікацію, можливість створення та управління дошками, що представляють собою великі області та проекти, а також функції системи включають можливість створення та управління картками, які представляють собою окремі задачі.

Актуальність роботи визначається поширеністю інструментів для управління задачами. Зі збільшенням використання мобільних телефонів ми звертаємо особливу увагу саме на мобільний додаток, а для зручності користування та ефективності – на зручний та зрозумілий інтерфейс. Додатково, актуальність диктується у реалізації покращених засобів захисту інформації у системі, зокрема, персональних даних.

Об'єктом дослідження у рамках даної роботи виступає процес розробки мобільного додатку для управління задачами. Предметом дослідження обрано реалізація функціональності мобільного додатку, аналогічного до Trello. У якості задач дослідження було встановлене наступне:

- Провести огляд існуючих мобільних додатків для управління задачами та проектами, визначити їх основні характеристики, функціональність, можливості та недосконалості для подальшого розвитку.
- Визначити програмні засоби (інтерфейси API), які дозволять реалізувати необхідний функціонал для управління задачами на

- мобільній платформі, визначити їх особливості та ступінь вживання.
- Розробити програмне забезпечення, реалізуючи функціонал додатку для управління задачами, провести тестування за різних умов та для різних пристроїв.
 - Реалізувати візуальну та технічну структуру для організації та управління задачами у вигляді дощок та карток, за допомогою яких користувачі отримають змогу ефективно планувати виконання проекту, відслідковувати прогрес та виконувати завдання.

При складанні звіту використано 9 посилань на друковані та електронні джерела інформації.

ЗМІСТ

ВСТУП	4
Технічне завдання до проекту	5
РОЗДІЛ 1. Загальна характеристика системи	8
РОЗДІЛ 2. Реалізація клієнтської частини	9

РОЗДІЛ 3. Результати випробувань_____12

ВИСНОВКИ_____19

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ_____20

ВСТУП

Майже будь яка робота пов'язана з проектами. Через розвиток та поширення ІТ-технологій постає задача перенесення великої кількості послуг у цифровий простір, Стратегія розвитку інформаційного суспільства в Україні визначає, що *«комунікаційні технології є необхідним інструментом соціально-економічного прогресу, одним з основних чинників інноваційного розвитку економіки»* [1].

Завдяки своїм технічним можливостям, сучасні мобільні технології - смартфони включно, значно спрощують управління та організацію задачами. Системи, які дозволяють користувачам ефективно планувати та координувати виконання задач є дуже поширеними та знаходять своє існування у мобільних додатках [2].

В якості предмету даної роботи було обрано мобільний додаток для управління та координування задачами, який дає змогу користувачам створювати та керувати дошками, представляючими собою певні проекти, картками, представляючими собою певні задачі, встановлювати терміни виконання задач, відстеження прогресу та системи балів для відслідковування активності виконавців.

Переважаюча кількість подібних систем передбачає можливість збереження історії задач, можливість редагування задачі, та відстеження факта виконання. Важливо також впровадити базові заходи безпеки для мобільних додатків, які включають в себе регулярні оновлення ПЗ, контроль доступу та мінімізацію використання конфіденційних даних [3].

Детальне технічне завдання до проекту наводиться далі. У рамках даного звіту розглядається архітектура проекту, яка представляє собою як клієнтську, так серверну частини. Клієнтська частина являє собою мобільний додаток створений для користувачів операційної системи Android. Серверна частина представляє компіляцію технологій, яка виконує функцію збереження даних та управління реєстрацією, авторизацією, тощо.

Звіт складається з чотирьох розділів: опис загальної характеристики системи, деталі створення клієнтської частини, принцип роботи серверної частини та результати розгортання та тестування. Висновки відображають результати, що були досягнуті у рамках даної частини проекту. Загальний висновок представляє собою сукупність звітів усіх учасників проекту.

ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОЕКТУ

Загальний опис.

Проект представляє собою інформаційну систему за зразком програми управління задачами "Trello". Проект створено для організації та управління задачами, відстеження прогресу виконання задач.

Призначення:

Ціль проекту полягає в наданні користувачам інструмента для організації задач, підвищення продуктивності та покращення взаємодії між виконавцями задач. Мобільний додаток дозволяє створювати простір для організації та планування задач за допомогою реалізування парадигми управління задачами Kanban.

Вимоги до функціоналу:

Реєстрація користувачів:

Користувач додатку може зареєструватися за допомогою авторизаційних даних (e-mail адреси, логіну та паролю).

Авторизація користувачів:

Користувач додатку може авторизуватися у додаток за допомогою введених при реєстрації даних (логіну та паролю) та увійти у систему.

Дошки:

Дошки є однією з основних функцій додатку. Дошка реалізує собою проект, до якого прив'язаний користувач. На ній відображаються усі задачі у рамках проекту, включаючи задачі закріпленні за користувачем. Користувач має мати можливість зручно створювати нові дошки, переглядати відображення дошок учасником яких він є, створювати та редагувати колонки та картки, переміщати картки між колонками, редагувати та видаляти дошку за допомогою панелі налаштувань.

Створення та управління картками:

Картки реалізують собою окремі задачі, які закріплені за користувачем. Користувач має мати можливість переглядати картки, залишати коментарі та переглядати матеріали, прикріплені до картки. Меню картки має опис, відображення списку учасників (та можливість додавання нових та видалення існуючих, при умові що користувач є адміністратором дошки), встановлені дати виконання задачі, контрольні списки, які представляють собою перелік необхідних задач, які потрібно виконати.

Архітектура

ПЗ має бути спроектоване за клієнт-серверною архітектурою:

- **Backend** - серверне програмне забезпечення, що забезпечує збереження та оброблення поточних даних, взаємодію з базою даних та сховищами. Надає програмний інтерфейс взаємодії (API) з іншими вузлами.
- **Mobile App** - додаток, що встановлюється на мобільних пристроях, отримує та відправляє дані з Backend через взаємодію з API.

Технології

Зберігання коду ПЗ повинно здійснюватись за допомогою системи контролю версіями (VCS), таких як Git. Структура файлів та каталогів повинна відбивати архітектуру ПЗ, на зразок:

- root
 - Backend
 - Web
 - Mobile

Для кожного з модулів рекомендується (з можливістю часткової зміни) наступні технології створення:

Backend - Node.js (для створення серверної логіки, обробки запитів та управління базами даних).

Mobile - Java або Kotlin (для розробки під Android), або Flutter (для кросплатформної розробки).

Вихідний код ПЗ повинен оформлятися відповідно до загальноприйнятих стандартів, таких як **GNU**. Ці стандарти полегшують підтримку коду та забезпечують його якість [4].

РОЗДІЛ 1

ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМИ

У відповідності до технічного завдання, система була спроектована за клієнт-серверною розподіленою архітектурою. У складі системи було відокремлено наступні архітектурні частини:

- «Backend»,
- «Frontend».

Частина «Backend» реалізована як централізований вузол для збереження даних користувача, що створюються та модифікуються протягом життєвого циклу інформаційної системи. Дана частина представляє собою програмне забезпечення, яке реалізує підключення до бази даних та сховища, отримує, обробляє та відправляє дані використовуючи API для взаємодії з клієнтом.

Частина «Backend» виконана на базі програмного забезпечення NodeJS, яке підключається до MongoDB для збереження даних, а Firebase Storage для збереження файлів.

Клієнтська частина системи «Frontend», призначена для надання користувачеві доступу до функцій системи через засоби візуального інтерфейсу. До цієї функції належить можливість управління задачами, відображення дошок та карток та їх налаштування. Вихідні коди та ресурси цієї частини розміщені на репозиторії [<https://github.com/FacePunch1337/TaskChecker/tree/main>]. Подальший звіт присвячений детальному опису цієї частини.

РОЗДІЛ 2

РЕАЛІЗАЦІЯ КЛІЄНТСЬКОЇ ЧАСТИНИ

Клієнтська частина інформаційної системи реалізована з використанням наступних технологій:

- Операційна система: Android, гарантується підтримка систем з версіями API від 23 до 31 (поточна на час початку створення проєкту).
- Засоби розроблення: мова програмування Java (OpenJDK 64-Bit Server VM by Oracle Corporation) та середовище Android Studio.

Зі сторони клієнтської частини мобільного додатку реалізовано наступні функції:

- Авторизація користувача, реєстрація нових користувачів
- Створення та редагування дошок
- Створення та редагування карток
- Керування учасниками

Попередньо до реєстрації введені дані проходять попередню валідацію для перевірки коректності на формат введених даних. Після успішної валідації користувач може продовжити процес реєстрації. Після вдалої реєстрації користувач має змогу авторизуватися у додатку та почати користуватися їм.

На наступному скріншоті продемонстровано реалізацію деяких аспектів валідації даних на стороні клієнта, а саме - чи заповнено поле та чи відображає воно відповідний індикатор.

```
private void validateField(EditText field, TextView label) {
    String fieldValue = field.getText().toString();
    field.setBackgroundResource(R.drawable.rounded_border_shadow);
    field.setPadding(70, 0, 35, 10);
    field.setCompoundDrawablesRelativeWithIntrinsicBounds(null, null, null, null);
    label.setTextColor(ContextCompat.getColor(requireContext(), R.color.black));
    if (!TextUtils.isEmpty(fieldValue)) {
        if ((field == emailEditText && !isValidEmail(fieldValue)) || (field == repeatPasswordEditText &&
            !isValidPasswordLength(fieldValue)))
        {
            field.setCompoundDrawablesRelativeWithIntrinsicBounds(null,
                                                                    null,
                                                                    ContextCompat.getDrawable(requireContext(), R.drawable.error),
                                                                    null);
            field.setBackgroundResource(R.drawable.rounded_error_border_shadow);
            label.setTextColor(ContextCompat.getColor(requireContext(), R.color.red));
            field.setPadding(70, 0, 35, 10);
        } else {
            field.setCompoundDrawablesRelativeWithIntrinsicBounds(null,
                                                                    null,
                                                                    ContextCompat.getDrawable(requireContext(),
                                                                        R.drawable.correct), null);
            field.setBackgroundResource(R.drawable.rounded_border_shadow);
            label.setTextColor(ContextCompat.getColor(requireContext(), R.color.black));
            field.setPadding(70, 0, 35, 10);
        }
    }
}
```

Авторизація користувача здійснюється за допомогою логіну та паролю, введених на етапі реєстрації.

При реєстрації або авторизації здійснюється обмін даними з частиною «Backend» у результаті чого авторизація підтверджується (у разі правильно ведених даних) або відхиляється. Передача даних користувача проходить з використанням шифрування за допомогою бібліотеки Bcrypt [5],[6] .

```
// Хеширование пароля перед сохранением в базу данных
const hashedPassword = await bcrypt.hash(password, 10);

const user = await User.create({
    username,
    email,
    password: hashedPassword,
    avatarURL: req.body.avatarURL || defaultAvatarURL,
    avatarFilename: req.body.avatarFilename || avatarFileName,
});
```

Для реалізації методу real-time пошуку учасників використовувався ChatGPT. Цей метод, подібно до функціоналу пошуку Google, оновлює список користувачів, які видаються при пошуку за введеною інформацією [7].

Як вже зазначалось вище, з повним кодом проекту можна ознайомитись у репозиторії <https://github.com/FacePunch1337/TaskChecker/>

РОЗДІЛ 3 РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

Результатом компіляції коду клієнтської частини являється APK файл додатку, який може бути встановлений на смартфон або планшет з операційною системою «Android». Головна серія випробувань, звіт з якої наведено далі, проведена на смартфоні «Xiaomi Redmi 9T» з операційною системою «Android 12».

Початок роботи користувача з додатком починається з реєстрації користувача у системі. Під час реєстрації користувач повинен заповнити поля реєстрації, такі як ім'я користувача, пошта, пароль, як це наведено на рис. 1.

Мінімальна кількість інформації необхідної для реєстрації забезпечує зручність та спрощує процес реєстрації для користувачів [8].

The image displays two versions of a mobile application's registration screen. Both screens feature a header with 'Sign up' and 'Sign In' buttons. The main heading is 'Sign Up'. The form includes four input fields: 'Username', 'E-Mail', 'Password', and 'Repeat password'. The 'Password' field has a toggle icon for visibility. Below the form is a 'Sign up' button and a link 'I'm already registered'. The right-hand version of the screen shows the form filled with the following data: Username: John Smith, E-Mail: johnsmith12@gmail.com, Password: passwordExample, and Repeat password: masked with dots. Green checkmarks are visible next to the Username, E-Mail, and Password fields, indicating successful validation.

Рисунок 1 - Реєстрація користувача

Після реєстрації, або Якщо користувач вже зареєстрований у системі, то він може увійти до системи за допомогою даних, зазначених при реєстрації, як це наведено на рис. 2

The image displays two versions of a 'Sign In' login form side-by-side. Both forms have a header with 'Sign up' and 'Sign In' links, where 'Sign In' is highlighted. The main title 'Sign In' is centered above the input fields. Each form contains two input fields: 'Username' and 'Password'. The 'Username' field on the right is filled with 'John Smith' and has a green checkmark icon on its right side. The 'Password' field on the right is filled with masked characters '.....' and has a red X icon on its right side. Both forms have a dark blue 'Sign In' button at the bottom.

Рисунок 2 - Авторизація користувача

Авторизовавшись у додатку, у користувача є можливість бачити дошки, учасником яких він є. Також у користувача є змога створювати свої власні дошки, як це наведено на рис. 3.

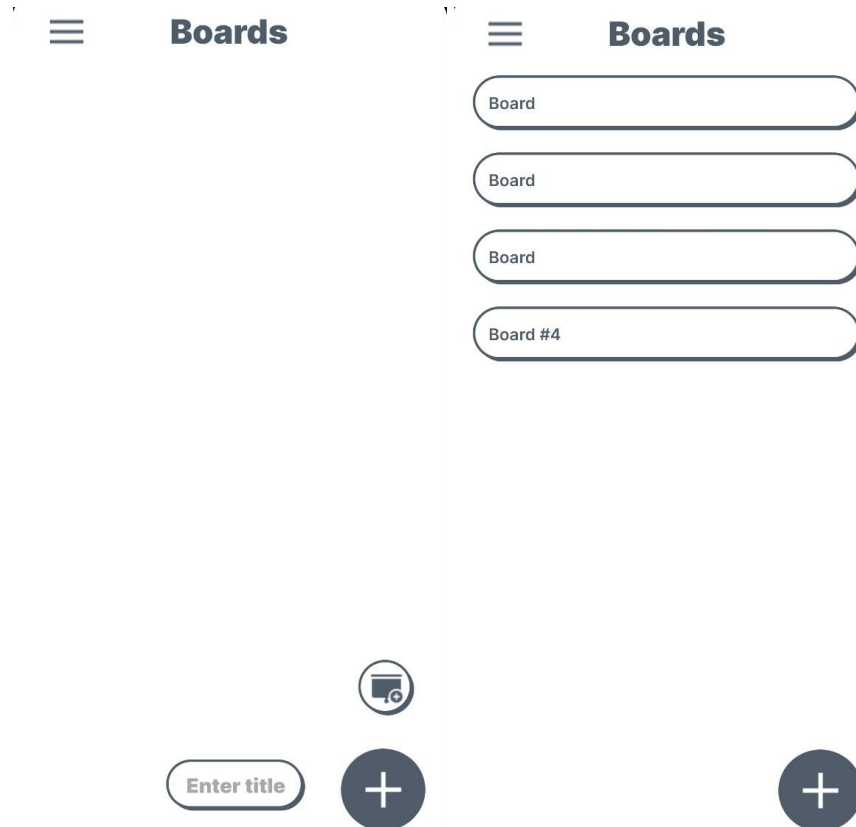


Рисунок 3 - Список закріплених за користувачем дошок та меню створення нової дошки

Після створення дошки у користувача є змога оновлювати список задач у розділі Backlog – список ще не розпочатих задач. Користувачі додають усі завдання у Backlog, де вони чекають своєї черги для початку роботи. З початком роботи над завданням, його можна перетягнути в інші розділи, такі як In Progress та Release [9]. На рис. 4 наведено наявні категорії.

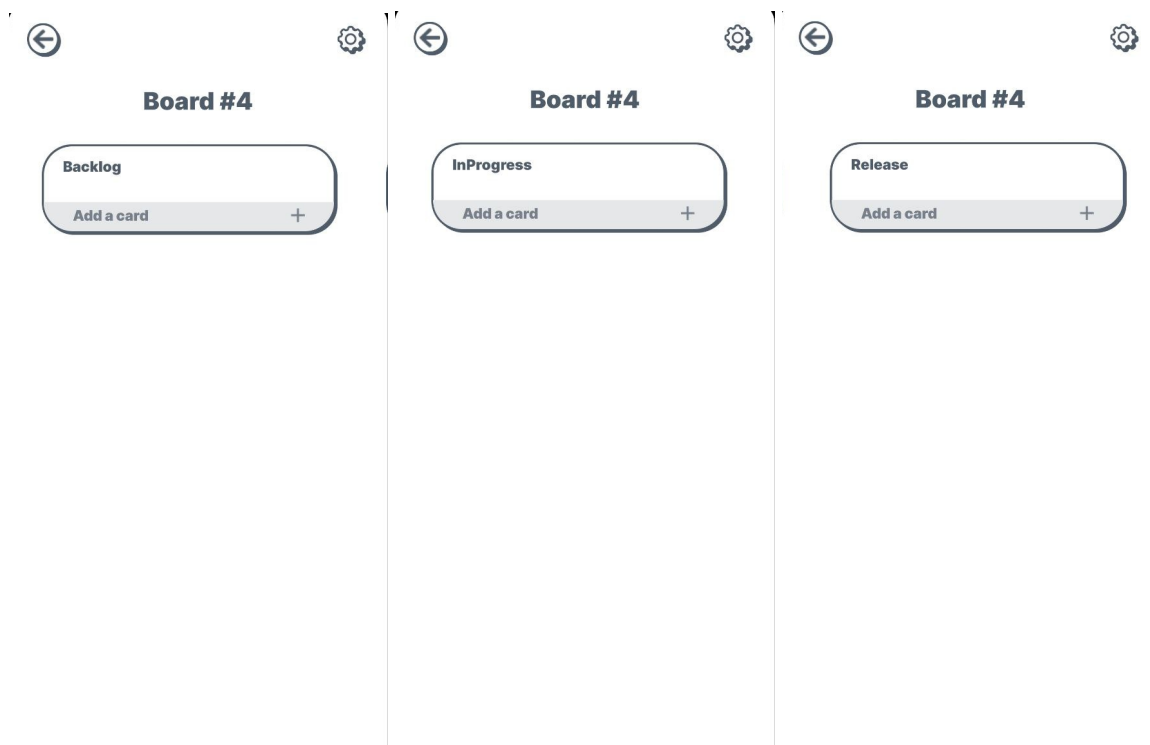


Рисунок 4 - Список наявних категорій для розміщення задач

Після обрання дошки у користувача є можливість створювати "картки" – конкретні задачі які пов'язані з цим проектом, як це наведено на рис. 5

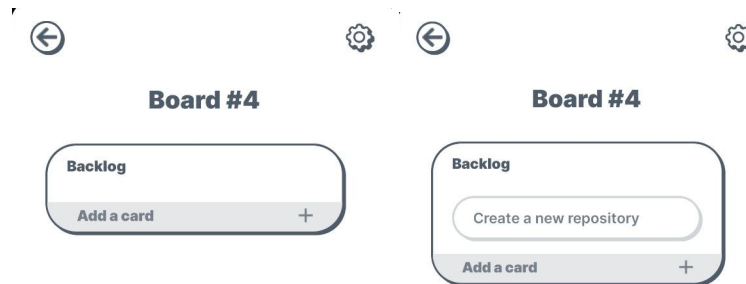


Рисунок 5 - Початковий вигляд категорії Backlog та оновлений вигляд після додавання картки з назвою "Create a new repository"

Кожна карта має свій розширений функціонал. При натисканні на картку відкривається меню, у якому користувач може змінити назву картки (задачі), задати виконавця, додавати та/або видаляти учасників, задати строки виконання, конкретні підзадачі та залишати коментарі. На рис. 6 продемонстровано певні складові цього функціоналу.

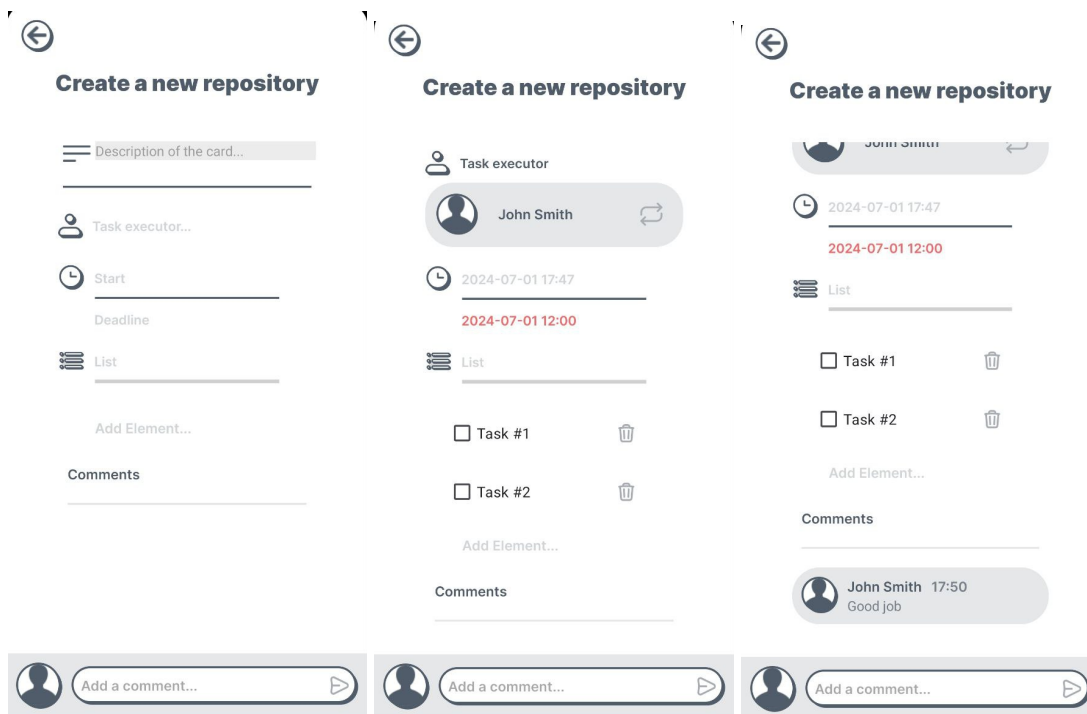


Рисунок 6 - Заповнення категорій картки, таких як строки виконання, список завдань, демонстрація наявності коментарів та їх відображення.

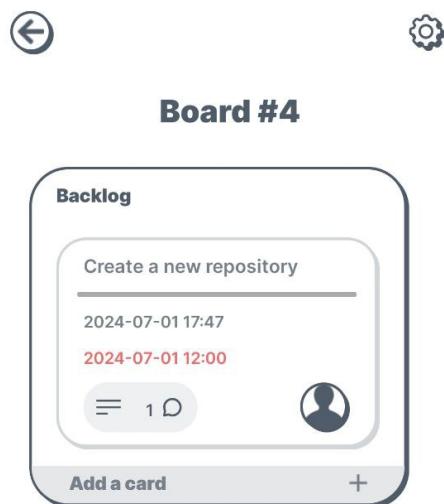


Рисунок 7 - демонстрація картки після заповнення окремих категорій.

ВИСНОВКИ

У ході проектування системи було виконано огляд існуючих мобільних додатків для управління задачами. Для подальшого розвитку системи було визначено основні задачі та переваги аналогічних застосунків.

Для реалізації функціоналу системи було використано інтерфейси API, що забезпечують інтеграцію та обробку даних. Основними компонентами Backend-складової системи стали NodeJS з використанням MongoDB, на основі яких и було розроблено серверну частину проекту, та Firebase Storage для збереження даних. Для Frontend-складової системи, який забезпечує зручний візуальний інтерфейс для користувача, було використано Android Studio та Java.

Безпечне шифрування даних користувачів забезпечується завдяки використанню бібліотеки Bcrypt, яка підвищує рівень захисту персональних даних користувачів та запобігає несанкціонованому доступу до інформації.

Працездатність додатку та відповідність до технічного завдання була підтверджена у ході тестування. Користувачі можуть реєструватися, авторизуватися, створювати дошки та картки, даючи собі змогу керувати проектами та задачами.

Усі вихідні коди розміщені на репозиторії Github за посиланням – <https://github.com/FacePunch1337/TaskChecker/tree/main>.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Стратегія розвитку інформаційного суспільства в Україні.
Розпорядження Кабінету Міністрів України від 15 травня 2013 р. № 386-р. // Урядовий портал. URL: <https://www.kmu.gov.ua/npas/246420577>
2. 50+ Trello Statistics: Revenue, Usage And Templates // MarketSplash.
URL: <https://marketsplash.com/trello-statistics/>
3. Guidelines for Managing the Security of Mobile Devices // NIST. URL: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-124r2.pdf>
4. GNU Coding Standards. URL: <http://www.gnu.org/prep/standards/>
5. Node.js Bcrypt - Wcorder. URL: <https://wcoder.github.io/notes/nodejs-bcrypt>
6. Bcrypt - npm. URL: <https://www.npmjs.com/package/bcrypt>
7. ChatGPT - OpenAI. URL: <https://chat.openai.com/>
8. Mobile app form design: best practices for happy users - Justinmind. URL: <https://www.justinmind.com/blog/mobile-app-form-design-best-practices/>
9. "What is Trello and How To Use It?" - Simplilearn. URL: <https://www.simplilearn.com/tutorials/project-management-tutorial/what-is-trello>