

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ

Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

**до виконання, оформлення та захисту курсової роботи
з дисципліни «Кросплатформені мови програмування»**
для студентів денної та заочної форми навчання
першого (бакалаврського) рівня вищої освіти
галузі «Інформаційні технології».

ЗАТВЕРДЖЕНО
на засіданні кафедри кібербезпеки та
програмного забезпечення, протокол
№ 1 від 26.08.2025 року

Кропивницький
2025

Кросплатформені мови програмування: методичні рекомендації до виконання, оформлення та захисту курсової роботи для студентів денної та заочної форми навчання, першого (бакалаврського) рівня вищої освіти, галузі «Інформаційні технології» / М-во освіти і науки України, Центральноукр. нац. техн. ун-т; [уклад. О.В. Коваленко, А.С. Коваленко] – Кропивницький: ЦНТУ, 2025. – 69 с.

Укладач:

Коваленко О.В., докт. техн. наук, доц.

Коваленко А.С., к.т.н., доцент.

Рецензенти:

Смірнов О. А., докт. техн. наук, професор, завідувач кафедри;

Якименко Н.М., к. ф.-м. наук, доцент.

ЗМІСТ

Вступ	4
ОСНОВНІ ЗАВДАННЯ КР.....	5
ФОРМА КОНТРОЛЮ КР	6
ОСНОВНІ ЕТАПИ КР	7
СТРУКТУРА І ОБСЯГ КР	14
АНАЛІЗ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ ТА СКЛАДАННЯ БІБЛІОГРАФІЇ	16
РОЗРОБКА ІНТЕРФЕЙСА КОРИСТУВАЧА.....	18
ЗМІСТ РОЗДІЛІВ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ.....	22
ВИМОГИ ДО ОФОРМЛЕННЯ ТЕКСТОВИХ ДОКУМЕНТІВ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ.....	27
ЗАХИСТ КУРСОВОЇ РОБОТИ ТА КРИТЕРІЇ ОЦІНЮВАННЯ.....	45
РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ	50
<i>Додаток А – Форма листа-завдання на курсову роботу та календарний план курсового проектування</i>	
<i>Додаток Б – Титульний аркуш курсової роботи</i>	
<i>Додаток В – Основний напис відповідно ГОСТ 2.104-2006</i>	
<i>Додаток Г – Приклад виконання блок-схеми алгоритма</i>	
<i>Додаток Д – Форма титульного аркушу програмного документу</i>	
<i>Додаток Е – Приклади представлення дизайну проекту</i>	

ВСТУП

Для полегшення виконання, оформлення та захисту курсової роботи (КР) студентам денної та заочної форми навчання необхідно ознайомитись з методичними рекомендаціями.

В методичних рекомендаціях висвітлено організаційні аспекти виконання КР з навчальної дисципліни «Кросплатформені мови програмування» для студентів денної та заочної форми навчання першого (бакалаврського) рівня вищої освіти галузі «Інформаційні технології», а також поради та вимоги щодо підбору, опрацювання та викладення теоретичного матеріалу, розробки КР та її програмної реалізації, порядку оформлення пояснювальної записки курсової роботи, її підготовки та представлення до захисту.

Навчальна дисципліна «Кросплатформені мови програмування» передбачає виконання курсової роботи, яка є складовою самостійної роботи здобувачів вищої освіти та спрямована на закріплення теоретичних знань і формування практичних навичок проектування, реалізації та використання мобільних додатків.

Курсова робота виконується на основі індивідуального завдання (тему пропонує здобувач освіти або на вибір з запропонованих тем) та полягає у розв'язанні інженерно прикладної задачі з проектування мобільного ПЗ.

Мета курсової роботи: закріплення та поглиблення теоретичних знань з дисципліни «Кросплатформені мови програмування», формування практичних навичок аналізу, проектування, реалізації, тестування та супроводу мобільного програмного забезпечення, а також набуття досвіду самостійного розв'язання інженерно прикладної задачі з розробки мобільного додатка (зокрема під ОС Android) на основі сучасних засобів і технологій програмування.

ОСНОВНІ ЗАВДАННЯ КУРСОВОЇ РОБОТИ

У процесі виконання курсової роботи здобувач повинен:

- Обрати та обґрунтувати тему курсової роботи відповідно до змісту дисципліни і поставленої інженерно прикладної задачі.
- Провести аналіз предметної області, визначити проблему, мету розробки та очікувані результати.
- Сформулювати функціональні та, за потреби, нефункціональні вимоги до мобільного додатка.
- Розробити структуру та логіку роботи програмного засобу, спроектувати основні модулі, сценарії взаємодії та інтерфейс користувача.
- Обґрунтувати вибір інструментальних засобів, мов програмування, бібліотек, фреймворків та середовища розробки.
- Реалізувати мобільний додаток відповідно до поставленого завдання з використанням засобів кросплатформеного програмування.
- Забезпечити коректну роботу додатка з даними, локальним збереженням, мережею або іншими компонентами відповідно до теми роботи.
- Провести тестування та налаштування ПЗ, усунути виявлені помилки.
- Оцінити працездатність і якість розробленого рішення на прикладах використання.
- Оформити пояснювальну записку відповідно до встановлених вимог та підготувати матеріали до захисту.
- Продемонструвати вміння аргументовано представляти результати роботи та відповідати на запитання під час захисту.
- Дотримуватися принципів академічної доброчесності під час виконання, оформлення та представлення курсової роботи.

ФОРМА КОНТРОЛЮ

Підсумковим етапом виконання курсової роботи є її захист. Захист курсової роботи є формою контролю з дисципліни «Кросплатформені мови програмування».

Захист курсової роботи проводиться в терміни, визначені кафедрою, до початку екзаменаційної сесії та є умовою допуску до заліку з дисципліни.

Курсова робота виконується здобувачем вищої освіти самостійно під керівництвом викладача протягом терміну встановленого навчальним планом.

КР носить творчий характер, кінцевим результатом якого може стати новий продукт (технологічний процес, механізм, технічні та програмні засоби, тощо) або пропозиції щодо вирішення сформульованої у курсовій роботі проблеми.

Форма реалізації – робота, виконана відповідно до Державних стандартів та вимог, визначених специфікою кафедри Кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету.

ОСНОВНІ ЕТАПИ КР

1. Отримати тему курсової роботи або затвердити власну тему у керівника КР (додаток А).

2. Скласти графік виконання курсової роботи, затвердити його у керівника КР та внести у календарний план завдання на курсову роботу (додаток А).

3. Розпочати збір і обробку інформації за темою курсової роботи. Виконати чітку постановку задачі на виконання курсової роботи.

4. Провести аналіз предметної області, наявних аналогів, програмних засобів і технологій, що можуть бути використані для розв'язання поставленої задачі.

5. Обґрунтувати вибір методів, засобів розроблення, середовища програмування, програмних платформ, бібліотек та інструментів, що використовуються під час виконання курсової роботи.

6. Виконати проектування структури мобільного програмного забезпечення, визначити склад основних модулів, їх функціональне призначення та взаємозв'язки між ними.

7. Оформити пояснювальну записку курсової роботи відповідно до встановлених вимог, підготувати схеми по результату:

7.1. Структурна схема (Показує основні модулі застосунку та зв'язки між ними).

7.2. Функціональна схема (Більш направлене представлення роботи системи, відокремлення основного функціоналу).

7.3. Дизайн проекту (Дизайн вікон мобільного додатка на вивір у форматі: LOW FIDELITY - WIREFRAMES, MEDIUM FIDELITY - MOCKUPS, HIGH FIDELITY - PROTOTYPES).

7.4. Блок-схема роботи мобільного програмного забезпечення (Наприклад: ініціалізація; перевірка дозволів; вибір підсистеми; запуск тесту;

зчитування параметрів; аналіз; збереження; відображення результату; завершення / повтор до завершення роботи).

8. Розробити інформаційну модель даних, за потреби спроектувати структуру локальної бази даних, формати збереження даних та механізми обміну даними із зовнішніми сервісами.

9. Розробити алгоритми функціонування мобільного програмного забезпечення, що забезпечують реалізацію основних функцій застосунку.

10. Виконати програмну реалізацію мобільного програмного забезпечення відповідно до поставленої задачі та затверджених проектних рішень.

11. Провести тестування програмного забезпечення, перевірити коректність роботи основних функцій, виконати налагодження та усунення виявлених недоліків.

12. Подати завершену курсову роботу керівнику КР на передзахист для перевірки, доопрацювати її за отриманими зауваженнями та підготувати до захисту.

13. виправити виявлені на передзахисті помилки та недоліки роботи, врахувати зауваження керівника й підготувати роботу до захисту.

14. Не пізніше встановленого календарним планом терміну подати курсову роботу до захисту.

За прийняті у КР рішення та достовірність даних відповідає автор курсової роботи, здобувач вищої освіти. Робота подається повністю оригінальна, якщо в роботі було застосовано роботи чи джерела інших авторів, то вони повинні бути вказані належним чином у вигляді цитат та зазначено у списку інформаційних джерел. Плагіат не допускається.

Студент, який без поважної причини не підготував КР або не захистив у зазначений термін, вважається таким, що має академічну заборгованість.

РЕКОМЕНДОВАНІ ТЕМИ КР

1. Мобільне програмне забезпечення для інженерної діагностики датчиків, комунікаційних модулів і базових підсистем мобільного телефону.
2. Мобільне програмне забезпечення для обліку та категоризації особистих витрат користувача.
3. Мобільне програмне забезпечення для планування навчальних завдань і контролю термінів їх виконання.
4. Мобільне програмне забезпечення для ведення електронного щоденника з локальним збереженням даних.
5. Мобільне програмне забезпечення для організації персонального розкладу занять і подій.
6. Мобільне програмне забезпечення для формування та супроводу списку покупок із сортуванням і фільтрацією.
7. Мобільне програмне забезпечення для обліку книжкової колекції користувача.
8. Мобільне програмне забезпечення для моніторингу фізичної активності на основі даних сенсорів смартфона.
9. Мобільне програмне забезпечення для підрахунку кроків і візуалізації добової активності користувача.
10. Мобільне програмне забезпечення для визначення параметрів переміщення користувача на основі GNSS даних.
11. Мобільне програмне забезпечення для реєстрації маршрутів пересування з відображенням ключових точок.
12. Мобільне програмне забезпечення для контролю рівня освітленості на основі вбудованих сенсорів пристрою.
13. Мобільне програмне забезпечення для моніторингу орієнтації та положення мобільного пристрою у просторі.
14. Мобільне програмне забезпечення для збору, збереження та аналізу показників датчиків смартфона.

15. Мобільне програмне забезпечення для ведення довідника контактної інформації з багатовіконним інтерфейсом.

16. Мобільне програмне забезпечення для обробки текстових даних із пошуком, сортуванням і фільтрацією записів.

17. Мобільне програмне забезпечення для ведення переліку академічних дисциплін і результатів навчання.

18. Мобільне програмне забезпечення для автоматизації обліку лабораторних робіт здобувача вищої освіти.

19. Мобільне програмне забезпечення для обліку відвідування занять і формування статистики активності.

20. Мобільне програмне забезпечення для створення та супроводу особистих нотаток із тегуванням записів.

21. Мобільне програмне забезпечення для керування переліком завдань з пріоритизацією та нагадуваннями.

22. Мобільне програмне забезпечення для формування персонального плану підготовки до заліків та іспитів.

23. Мобільне програмне забезпечення для ведення електронного каталогу мультимедійних матеріалів.

24. Мобільне програмне забезпечення для зчитування, парсингу та відображення даних у форматі JSON.

25. Мобільне програмне забезпечення для зчитування, парсингу та відображення даних у форматі XML.

26. Мобільне програмне забезпечення для агрегування даних із REST API та їх локального кешування.

27. Мобільне програмне забезпечення для отримання, обробки та візуалізації погодних даних із зовнішнього сервісу.

28. Мобільне програмне забезпечення для відображення новинного контенту з мережевим завантаженням і фільтрацією.

29. Мобільне програмне забезпечення для віддаленого перегляду довідкової інформації з серверного джерела даних.

30. Мобільне програмне забезпечення для збереження користувацьких налаштувань із використанням локального сховища.

31. Мобільне програмне забезпечення для роботи з базою даних SQLite у задачах персонального обліку.

32. Мобільне програмне забезпечення для обліку навчальної літератури з використанням локальної бази даних.

33. Мобільне програмне забезпечення для реєстрації та аналізу особистих показників часу виконання завдань.

34. Мобільне програмне забезпечення для контролю режиму дня та обліку щоденних звичок користувача.

35. Мобільне програмне забезпечення для ведення журналу споживання води та формування нагадувань.

36. Мобільне програмне забезпечення для обліку медичних показників користувача з візуалізацією динаміки змін.

37. Мобільне програмне забезпечення для підтримки користувача з обмеженими можливостями здоров'я з урахуванням вимог доступності.

38. Мобільне програмне забезпечення для реалізації доступного інтерфейсу користувача відповідно до принципів WCAG.

39. Мобільне програмне забезпечення для обробки подій життєвого циклу Activity та Fragment із збереженням стану інтерфейсу.

40. Мобільне програмне забезпечення для реалізації багатовіконної навігації з передаванням даних між екранами.

41. Мобільне програмне забезпечення для організації фонових задач із використанням сервісів та планувальників Android.

42. Мобільне програмне забезпечення для формування та керування системою сповіщень користувача.

43. Мобільне програмне забезпечення для реалізації безпечного доступу до геолокації, камери та пам'яті пристрою.

44. Мобільне програмне забезпечення для аналізу дозволів Android застосунку та підвищення безпеки обробки даних.

45. Мобільне програмне забезпечення для захищеного збереження конфіденційних даних користувача.

46. Мобільне програмне забезпечення для оцінювання продуктивності застосунку за показниками часу запуску та стабільності роботи.

47. Мобільне програмне забезпечення для аналізу ключових показників ефективності мобільного застосунку.

48. Мобільне програмне забезпечення для інформаційної підтримки користувача у задачах туристичної навігації.

49. Мобільне програмне забезпечення для обліку та класифікації фотофайлів із локальним сховищем метаданих.

50. Мобільне програмне забезпечення для довідково-інформаційного супроводу освітнього процесу здобувача вищої освіти.

51. Мобільне програмне забезпечення для ведення персонального календаря медичних процедур і прийому лікарських засобів.

52. Мобільне програмне забезпечення для обліку та аналізу показників якості сну користувача.

53. Мобільне програмне забезпечення для автоматизації обліку домашніх справ і контролю їх виконання.

54. Мобільне програмне забезпечення для супроводу індивідуальної програми тренувань із фіксацією результатів.

55. Мобільне програмне забезпечення для обліку енергоспоживання побутових пристроїв користувача.

56. Мобільне програмне забезпечення для ведення персонального архіву документів із класифікацією та пошуком.

57. Мобільне програмне забезпечення для обліку та аналізу персональних фінансових цілей користувача.

58. Мобільне програмне забезпечення для супроводу процесу вивчення іноземної мови з контролем індивідуального прогресу.

59. Мобільне програмне забезпечення для автоматизації обліку виконання курсових і самостійних робіт здобувача вищої освіти.

60. Мобільне програмне забезпечення для підтримки процесу самооцінювання навчальних досягнень здобувача вищої освіти.

61. Мобільне програмне забезпечення для обліку та аналізу витрат часу на виконання повсякденних дій.

62. Мобільне програмне забезпечення для реєстрації, класифікації та пошуку геолокаційних об'єктів користувача.

63. Мобільне програмне забезпечення для контролю використання ресурсів мобільного пристрою в різних режимах роботи.

64. Мобільне програмне забезпечення для обліку стану акумулятора мобільного пристрою та аналізу режимів енергоспоживання.

65. Мобільне програмне забезпечення для діагностики параметрів мережевого з'єднання мобільного пристрою.

66. Мобільне програмне забезпечення для виявлення, збереження та аналізу змін параметрів бездротових мереж.

67. Мобільне програмне забезпечення для автоматизації ведення переліку програмних проєктів користувача та контролю стану їх виконання.

68. Мобільне програмне забезпечення для формування персоналізованих нагадувань на основі подій, часу та місцезнаходження користувача.

69. Мобільне програмне забезпечення для обліку та аналізу індивідуальних показників навчальної активності здобувача вищої освіти.

70. Мобільне програмне забезпечення для формування, збереження та візуалізації результатів опитувань користувача.

СТРУКТУРА І ОБСЯГ КР

УВАГА! Текст підкреслений та виділений червоним коліром тут і далі по тексту це моменти на які в першу чергу необхідно звернути увагу здобувачам вищої освіти.

Документ курсової роботи повинен мати наступну структуру:

1 – Титульний аркуш.

2 – Завдання на курсову роботу та календарний план

3 – Пояснювальна записка (зміст і розділи):

- Перелік умовних позначень, символів, одиниць і термінів

- **Вступ** (1-2 сторінки).

- Розділ 1 **Призначення та область використання** (**10-18 сторінок**).

- Розділ 2 **Перегляд аналогічних існуючих систем** (**10-18 сторінок**).

- Розділ 3 **Опис і обґрунтування проектних рішень** (**15-20 сторінок**).

- Розділ 4 **Реалізація роботи. Розрахунки та експериментальні дані, що підтверджують вірність проектних і програмних рішень** (**12-20 сторінок**),

- Розділ 5 **Впровадження системи в промислову експлуатацію** (**2-5 сторінок**).

- Розділ 6 **Основні висновки** (**1-2 сторінки**).

- Список використаних джерел

4 Додатки:

4.1 **Додаток А** – Графічні матеріали

4.2 **Додаток Б** – Лістинг програми.

В залежності від особливостей теми курсової роботи, за погодженням з керівником КР можливе уточнення змісту розділів (текстові, графічні документи: функціональну схему системи та блок-схему алгоритму програми, тощо), введення нових або об'єднання деяких розділів пояснювальної записки КР.

Пояснювальна записка КР повинна мати не менше 45 сторінок друкованого тексту, який виконано виключно чорним кольором і чітким шрифтом. **Шрифт – Times New Roman, розмір шрифту – 14 пт, міжрядковий інтервал – 1.5 пт, абзацний відступ – 15 мм.**

Основна частина складається з розділів, підрозділів, пунктів та підпунктів. Кожний розділ починають з нової сторінки. Структура основної частини роботи повинна розкривати реалізацію завдань кваліфікаційної бакалаврської роботи.

Завдання та календарний план КР

Завдання на курсову роботу видається студентові перед початком роботи над КР і затверджується керівником курсової роботи (додаток А).

Календарний план (додаток А) складається студентом та затверджується керівником КР.

Рекомендовано наступний план курсового проектування (2 лист, з обратної сторони):

1. Вибір і затвердження теми курсової роботи.
2. Збір та аналіз матеріалів за темою роботи.
3. Постановка задачі та формування вимог.
4. Проектування структури й алгоритмів роботи ПЗ.
5. Програмна реалізація курсової роботи.
6. Тестування, налагодження та аналіз результатів.
7. Оформлення пояснювальної записки та підготовка до захисту.

Термін виконання етапів курсового проектування визначається студентом, погоджується з керівником, вноситься до календарного плану листа-завдання (додаток А) та підписується студентом і керівником.

Під час виконання КР кожен етап, відповідно до календарного плану, узгоджується з керівником, який робить відповідний запис у графі «Примітки» календарного плану.

АНАЛІЗ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ ТА СКЛАДАННЯ БІБЛІОГРАФІЇ

Під час виконання КР необхідно підібрати актуальну літературу та інші інформаційні джерела, які стосуються теми КР.

Після їх вивчення та аналізу для реалізації роботи, складається бібліографія, яка подається у розділі “Список використаних джерел” пояснювальної записки КР.

При самостійному пошуку фахової літератури особливу увагу слід звернути на нові публікації та періодичні фахові видання - газети, вісники, журнали, в яких можна знайти найновіші розробки в галузі інформаційних технологій, результати досліджень спеціалістів тощо. Також важливо опрацювати джерела сучасної зарубіжної літератури.

Окрім друкованих видань допускається використання електронних інформаційних джерел Internet.

Проте перевагу слід надавати офіційним (!) web-сайтам і базам даних. Найвідоміший серед них - електронний фонд Національної бібліотеки України імені В.І. Вернадського (НБУВ). Він доступний за адресою: <http://nbuv.gov.ua/>.

Спочатку необхідно ознайомитися з основною літературою (підручниками, журналами, збірниками наукових праць тощо), потім вивчаються електронні джерела інформації, розміщені в мережі або розповсюджуються на електронних носіях.

Бібліографічний опис у списку літератури повинен містити прізвище та ініціали автора, назву, видавництво, рік видання, кількість сторінок.

Нариклад:

Гайна Г.А. Основи проектування баз даних: Навчальний посібник. – К.: КНУБА, 2025. – 204 с.

Gradle. Gradle User Manual, документація системи збірки для Android проєктів [Електронний ресурс]. URL: <https://docs.gradle.org/>

Google. Налаштування Firebase для Android проєкту, підключення SDK та конфігурація [Електронний ресурс]. URL: <https://firebase.google.com/docs/android/setup?hl=en>

Google. Довідка Play Console, створення та первинне налаштування застосунку для публікації [Електронний ресурс]. URL: <https://support.google.com/googleplay/android-developer/answer/9859152?hl=en>

Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 с.

У тексті пояснювальної записки КР рекомендовано вказувати на використану літературу за допомогою квадратних дужок “[]”, в яких вказується номер джерела зі списку літератури.

Наприклад, [3] - посилання на джерело зі списку літератури під номером 3; [2-4] - посилання на джерела під номерами 2, 3 та 4; [1-3, 6] - посилання на джерела під номерами 1, 2, 3, 6.

РОЗРОБКА ІНТЕРФЕЙСА КОРИСТУВАЧА.

Інтерфейс повинен мати такі властивості: природність, узгодженість, дружність, принцип «зворотнього зв'язку», простота, гнучкість, естетична привабливість, чіткість.

Ефективність інтерфейса полягає у швидкому розвитку в користувачів простої концептуальної моделі взаємодії. Це досягається через узгодженість.

Інтерфейс може бути узгоджений в трьох аспектах: фізичному, синтаксичному і семантичному.

Фізична узгодженість відноситься до технічних засобів.

Синтаксична узгодженість відноситься до послідовності й порядку появи елементів на екрані (мова зображень), послідовності запитів (мова дій). Наприклад, синтаксична узгодженість матиме місце, якщо заголовок панелі завжди розміщується в центрі та вгорі панелі.

Семантична узгодженість відноситься до значення елементів, складових інтерфейсу. Наприклад, що означає «Вихід»? Де користувачі роблять запит на «Вихід» і що потім відбувається?

Природність інтерфейса - найважливіша його характеристика, оскільки він не змушує користувача істотно змінювати звичні для нього способи розв'язку задачі. Це, зокрема, означає, що повідомлення й результати, які видаються програмним продуктом, не повинні вимагати додаткових пояснень. Доцільно також зберегти систему позначень і термінологію, які застосовувались в даній предметній області.

Інтуїтивно зрозумілий інтерфейс –забезпечує використання знайомих користувачеві понять і образів (метафор). Користувачі зазвичай вивчають особливості роботи з новим програмним продуктом методом спроб і помилок. Ефективний інтерфейс повинен враховувати такий підхід. На кожному етапі роботи він повинен виконувати тільки відповідний набір дій і попереджати користувачів про ті ситуації, де вони можуть зашкодити системі або даним; ще краще, якщо у користувача є можливість відмінити або виправити виконані дії. Навіть за наявності добре спроектованого інтерфейсу користувачі можуть

робити помилки. Ці помилки можуть бути як «фізичного» типу (випадковий вибір неправильної команди або даних) так і «логічного» (ухвалення невірної рішення на вибір команди або даних).

Ефективний інтерфейс повинен дозволяти запобігати ситуаціям, які, ймовірно, закінчаться помилками. Він також повинен уміти адаптуватися до потенційних помилок користувача і полегшувати йому процес усунення наслідків таких помилок.

Завжди забезпечуйте зворотний зв'язок для дій користувача. Кожна дія користувача повинна отримувати візуальне, а іноді й звукове підтвердження того, що програмне забезпечення сприйняло введену команду; при цьому вид реакції, за можливістю, повинен враховувати природу виконаної дії.

Інтерфейс повинен бути простим. При цьому мають на увазі забезпечення легкості в його вивченні й у використанні. Крім того, він повинен надавати доступ до всього набору функціональних можливостей, передбачених даною програмою. Реалізація доступу до широких функціональних можливостей і забезпечення простоти роботи суперечать один одному. Розробка ефективного інтерфейса покликана збалансувати ці цілі.

Один з можливих шляхів підтримки простоти - зображення на екрані інформації, мінімально необхідної для виконання користувачем чергового кроку завдання.

Інший шлях до створення простого, але ефективного інтерфейсу - розміщення і представлення елементів на екрані з урахуванням їх смислового значення й логічного взаємозв'язку. Це дозволяє використовувати в процесі роботи асоціативне мислення користувача.

Під гнучкістю інтерфейсу розуміють здатність враховувати рівень підготовки та продуктивність праці користувача. Властивість гнучкості припускає можливість зміни структури діалогу і/або вхідних даних.

Проектування візуальних компонентів є найважливішою складовою розробки програмного інтерфейсу. Коректне візуальне представлення об'єктів забезпечує передачу дуже важливої додаткової інформації про поведінку і

взаємодію різних об'єктів. В той же час слід пам'ятати, що кожен візуальний елемент, який з'являється на екрані, потенційно вимагає уваги користувача, яка, як відомо, не безмежна. Слід забезпечити формування на екрані такого середовища, яке не тільки сприяло б розумінню користувачем представленої інформації, але і дозволяло б зосередитися на найбільш важливих її аспектах.

До візуальних атрибутів інформації, що відображається, відносяться:

- взаємне розташування і розмір об'єктів, що відображаються;
- палітра;
- засоби привертання уваги користувача.

Проектування розміщення даних на екрані передбачає виконання наступних дій:

- визначення складу інформації (елементів, компонентів, об'єктів тощо), яка повинна з'являтися на екрані;
- вибір формату представлення цієї інформації;
- визначення взаємного розташування даних (чи об'єктів) на екрані;
- вибір засобів привертання уваги користувача;
- розробка макету розміщення даних на екрані;
- оцінка ефективності розміщення інформації.

Загальні принципи розташування інформації на екрані повинні забезпечувати для користувача:

- можливість переглядання екрану в логічній послідовності;
- простоту вибору потрібної інформації;
- можливість ідентифікації зв'язаних груп інформації;
- помітність виключних ситуацій (повідомлень про помилки або попередження);
- можливість визначити, яка дія з боку користувача потрібна (і чи потрібна взагалі) для продовження виконання завдання.

Питання про те, яка інформація підлягає відображенню, вирішується залежно від специфіки теми курсової роботи. В даному випадку істотну роль

грає правильне розбиття завдання на операції (етапи), які не вимагають одночасної присутності великого об'єму даних на екрані.

Ця умова витікає з такої психофізіологічної особливості людини, як обмеженість його короткочасної пам'яті, здатної зберігати одночасно не більше п'яти-дев'яти об'єктів. Якщо вся інформація початкового документа не вміщується на одному екрані, деякі елементи даних можуть повторюватися на інших екранах для збереження цілісності й послідовності обробки. Як правило, повторювана інформація не повинна міняти свого розташування на всіх кроках виконання завдання.

Властивість природності інтерфейсу припускає, що інформація відображається на екрані у вигляді, придатному для безпосереднього використання. Виділення інформації - це використання таких атрибутів, які дозволяють привернути увагу користувача до певної області екрану. В якості подібних атрибутів можуть виступати: колір символів, колір фону, рівень яскравості, мерехтіння і застосування різних шрифтів для символів, що виводяться. Основна рекомендація: слід прагнути використовувати мінімально необхідну кількість атрибутів.

Програмна реалізація системи.

На цьому етапі за допомогою обраної мови програмування та на основі розробленого алгоритма здійснюється написання програми.

Важливою складовою лістинга програми є коментарі. Вони не тільки полегшать "читання" та відлагодження програми, але й є ознакою гарного стилю програмування.

Обов'язковим компонентом програмного забезпечення інструкція користувача (з детальним описом принципів роботи з програмою). Крім того, необхідно додати довідникову інформацію про розробника програмного забезпечення (прізвище, ім'я, по батькові, назва ЗВО, факультет, група, рік розробки тощо).

ЗМІСТ РОЗДІЛІВ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Титульний аркуш, завдання

Приклад оформлення титульного листа наведений у додатках, а оформлення завдання у додатку А.

Пояснювальна записка.

Зміст подають на початку роботи. Він містить найменування та номери початкових сторінок вступу, усіх розділів, підрозділів та пунктів.

Всі аркуші пояснювальної записки, оформлені в рамці з основним написом (додаток В) згідно ДСТУ.

У великому полі основного напису (додаток В) вказується шифр розробника роботи.

Приклад

КР-AA.25.00XX.00.00.ПЗ

Шифр розробника курсової роботи:

КР – курсова робота (без змін);

AA – Спеціальність (F7, F3, F5...);

25 – Рік проектування КР.

XX – Останні дві цифри номера індивідуального плану студента;

ПЗ – пояснювальна записка (без змін).

Перелік умовних позначень, символів, одиниць і термінів вводитьься, якщо в роботі вжита специфічна термінологія, маловідомі скорочення, нові символи тощо. Перелік друкується двома колонками, у яких, ліворуч за абеткою наводять скорочення, праворуч – їх детальну розшифровку. Якщо спеціальні терміни, символи, позначення і таке інше повторюється менше трьох разів, перелік не складають, а їх розшифрування наводять **у тексті при першому згадуванні у дужках.**

Вступ

У вступі розкривається сутність задачі та її значущість, підстави і вихідні дані для розробки теми. Вступ включає актуальність теми, мету і завдання роботи, практичне значення отриманих результатів.

Висвітлення актуальності не повинно бути багатослівним. Досить кількома реченнями висловити головне – сутність проблеми або науково-практичної задачі. та стислий висновок щодо розробки програмного забезпечення.

1 Призначення та область використання

Наводяться основні параметри (характеристики) системи, що розробляється, призначення системи й можливі області застосування.

Структура розділу:

1.1 Призначення системи

1.2 Область застосування

2 Перегляд аналогічних існуючих систем

Здійснюється огляд існуючого програмного забезпечення, систем, приладів, основних напрямків розвитку. Наводиться аналіз їх характеристик, властивостей, недоліків і переваг, на основі чого робиться висновок про доцільність (необхідність) проектування системи (розробки програмного забезпечення) відповідно до теми КР. Здійснюється короткий опис обраного інструментарію. Обґрунтовуються основні принципи розробки програмного забезпечення, його переваги над іншими для вирішення обраної задачі. Здійснюється чітка постановка задачі.

Структура розділу:

2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми КР

Аналіз переваг та недоліків існуючих рішень. Обґрунтування необхідності розробки системи за темою КР з урахуванням проведеного аналізу.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Коротка характеристика методів, апаратних засобів, середовища розробки. **Обґрунтування вибору по декільком параметрам. Обґрунтування вибору мови програмування.**

2.3 Розгорнута постановка завдання

Перерахування основних пунктів розробки, які визначають тему та склад КР (обсяг 1 сторінка).

3 Опис і обґрунтування проектних рішень

Обґрунтовуються основні принципи проектування системи, методика проектування.

Описується хід теоретичної побудови моделі проекту, приводиться її обґрунтування. Робляться стислі висновки.

При необхідності наводиться математична модель розробленої системи.

В цьому розділі наводяться наступні підрозділи, що містять відповідні схеми та їх детальний опис:

3.1 Опис функціонування системи

3.2 Розробка структурної схеми

Розробка структурної схеми пристрою, або системи (повна). **Опис побудови логічної структури системи.**

3.3 Розробка функціональної схеми

Розробка функціональної схеми системи. **Більш направлене представлення роботи системи нвж структурана схема, зазвичай описується одна група блоків функціональної схеми більш ретельно (відокремлено) для більш фокусного представлення основного контексту роботи.**

3.4 Розробка дизайну проекту

Представлено розроблені активіті (вікна програми), приклади представлено у додатку Е. На вибір здобувача необхідно представити дизайну

проекту в одну з трьох варіантів Low, Medium or High fidelity (НИЗЬКА, СЕРЕДНЯ, ВИСОКА ТОЧНІСТЬ)

4 Реалізація роботи. Розрахунки та експериментальні дані, що підтверджують вірність проектних і програмних рішень.

Наводяться розрахунки й експериментальні матеріали, які підтверджують вірність рішень, наведених у кваліфікаційній бакалаврській роботі.

Наводяться алгоритми, які реалізують функціональність системи.

Розробка класів, функцій, таблиць баз даних, взаємодії компонентів системи. Реалізація зв'язних, системних та програмних інтерфейсів. Компонувка частин системи.

В цьому розділі наводяться наступні підрозділи

4.1 Розробка блок-схем та опис алгоритмів функціонування системи

Опис блок-схем роботи системи, задіяних своїх чи використаних алгоритмів роботи системи, робота мобільного пристрою чи архітектури в цілому. Для представлення **необхідно показати що найменше 2 блок схеми** – Блок-схема роботи системи (робота системи від початку до завершення в цілому весь цикл), Блок схема роботи підпрограм(и) (показ роботи основного алгоритму реалізованої системи чи частини системи покроково). Крім того для підвищення оцінки опціонально рекомендується зробити схеми UML-діаграму класів та потоку даних у мобільному додатку (див. приклади у додатку Е).

4.2 Захист розробленого програмного забезпечення

В підрозділі «Захист розробленого програмного забезпечення» розглядаються механізми або алгоритми захисту розробленого програмного забезпечення (коротко наводиться конкретний алгоритм або метод). Захищатися може як сама прогшрамма так і дані у системі окремо, чи протокол передачі масиву даних, тощо.

5 Впровадження системи в промислову експлуатацію

Розробляється і описується методика інтеграції компонентів розробленого програмного забезпечення в існуючу апаратну систему. Описується інструкція користувача.

Наводяться скріншоти розробленого програмного забезпечення та форма авторського права.

6 Основні висновки.

Дається стисла характеристика розробленої системи за всіма основними параметрами. Містить найбільш важливі результати, одержані в ході виконання КР.

Список використаних джерел

Наводиться нумерований список літератури та інформаційних джерел, які використано для реалізації курсової роботи.

Кількість використаних джерел (літературних джерел, патентів, нормативно-технічних документів, адреси сайтів Інтернету), повинно бути не менше 30.

Список використаних джерел оформляється згідно Національного стандарту України ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання».

Додатки

За необхідності до додатків доцільно включити допоміжний матеріал, необхідний для повноти сприйняття роботи (інструкції, методики, опис алгоритмів дій); ілюстрації допоміжного характеру (таблиці, діаграми, схеми, звіти, запити тощо); формули і розрахунки, зразки анкет, тестів та ін. На кожний додаток повинно бути посилання в тексті.

ВИМОГИ ДО ОФОРМЛЕННЯ ТЕКСТОВИХ ДОКУМЕНТІВ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Текст пояснювальної записки оформляється у відповідності з вимогами ДСТУ 3008-95 та ДСТ 2.105-95.

Форми і правила виконання текстових документів (відомостей, пояснювальних записок) встановлені ДСТУ 3008-95. У відповідності з листом Міністра освіти України №1/9-73 від 01.03.99р. кваліфікаційні бакалаврські роботи повинні виконуватися лише державною мовою. Скорочення слів у тексті ПЗ здійснюється у відповідності з ДСТУ 3582–97. Інформація та документація. Скорочення слів в українській мові в бібліографічному описі. Загальні вимоги та правила.

ПЗ виконується на аркушах формату А4 (210x297 мм) ДСТ 2.301-68.

Титульний аркуш оформлюється по формі (Додаток Б).

Заголовний та наступні аркуші документа. Зміст оформлюються як заголовні аркуші. Оформлення рамок заголовних та наступних після них аркушів див. в Додатку В. Форма, розміри, зміст і порядок заповнення основного напису й додаткових граф до нього наведені в міжнародному стандарті ДСТ 2.104-2006:

- форма 2 (40 x 185 мм) – для заголовних аркушів текстових документів;
- форма 2а (15 x 185 мм) – для наступних аркушів текстових документів.

Вимоги до оформлення тексту

Текст пояснювальної записки друкується шрифтом **Times New Roman 14** розміру, відстань між рядками – 1.5 інтервал, вирівнювання тексту за шириною. Виключенням є приклади частин програмного коду, для них необхідно застосовувати шрифт **Courier New 10** розміру, полуторний інтервал та вирівнювання тексту по лівому краю.

Абзацний відступ повинен бути однаковим впродовж усього тексту і дорівнювати 1,5 см.

Відступи тексту від рамки: зверху і знизу не менше 10 мм; зліва і справа не менше 3-5 мм. Відступи повинні бути однаковими впродовж всього тексту.

Заголовки розділів ПЗ розміщуються по центру, друкуються великими буквами та виділяють жирним шрифтом. Структурні елементи "ЗМІСТ", "ВСТУП", "ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ ТА СПЕЦІАЛЬНИХ ТЕРМІНІВ" та "СПИСОК ЛІТЕРАТУРИ" не нумерують.

Заголовки підрозділів, пунктів та підпунктів розміщуються з абзацу малими буквами, крім першої великої, виділяються жирним шрифтом, та вирівнюються по ширині.

Відстань між заголовком розділу та подальшим і/або попереднім текстом (а також відстань між заголовком розділу та підрозділу) дорівнює двом рядкам (що також можна виставити інтервалом рівним 50 пт (пунктів)).

Відстань між заголовком підрозділу та подальшим і/або попереднім текстом дорівнює одному рядку (або ж інтервалу рівному 25 пт).

Всі інші заголовки, що знаходяться всередині підрозділів і не відображаються у змісті, розміщуються з абзацу малими буквами, крім першої великої, виділяються жирним шрифтом, та вирівнюються по ширині і не містять відступів між подальшим та попереднім текстом.

Заголовки слід друкувати без крапки в кінці. Якщо заголовок складається з двох і більше речень, їх розділяють крапкою. Перенесення слів у заголовку розділів не допускається.

Розділи і підрозділи повинні мати заголовки. Пункти і підпункти можуть мати заголовки.

Не допускається розміщувати назву розділу, підрозділу, а також пункту й підпункту в нижній частині сторінки, якщо після неї розміщено тільки один рядок тексту.

Нумерація розділів, підрозділів, пунктів, підпунктів. Розділи, підрозділи, пункти, підпункти звіту слід нумерувати арабськими цифрами.

Розділи звіту повинні мати порядкову нумерацію і позначатися арабськими цифрами без крапки, наприклад, 1, 2, 3 і т. д.

Наприклад: **1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ**

Підрозділи звіту повинні мати порядкову нумерацію в межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, відокремлених крапкою. Після номера підрозділу крапку не ставлять, наприклад 1.1, 1.2 і т. д.

Наприклад: **2.3 Розгорнута постановка завдання**

Пункти повинні мати порядкову нумерацію в межах кожного розділу або підрозділу. Номер пункту складається з номера розділу і порядкового номера пункту або з номера розділу, порядкового номера підрозділу та порядкового номера пункту, відокремлених крапкою. Після номера пункту крапку не ставлять, наприклад, 1.1, 1.2, або 1.1.1, 1.1.2 і т.д.

Переліки. Переліки, за потреби, можуть бути наведені всередині пунктів або підпунктів. Перед переліком ставлять двокрапку.

Перед кожною позицією переліку слід ставити малу літеру української абетки з дужкою, або, не нумеруючи – дефіс (перший рівень деталізації).

Для подальшої деталізації переліку слід використовувати арабські цифри з дужкою (другий рівень деталізації).

Приклад:

а) _____;

б) _____;

1) _____;

2) _____;

в) _____;

Переліки першого рівня деталізації друкують малими літерами з абзацного відступу, другого рівня – з відступом відносно місця розташування переліків першого рівня.

В тексті документу, за виключенням формул, таблиць та рисунків, не допускається:

– застосовувати математичний знак мінус(-) перед від’ємним значенням величин (слід писати слово “мінус”);

– застосовувати знак “ \emptyset ” для позначення діаметру (слід писати слово “діаметр”). При зазначенні розміру діаметру на кресленнях, які розташовані в тексті документу, перед розмірним числом слід писати знак “ \emptyset ”;

– застосовувати без числових значень математичні знаки, наприклад $>$ (більше), $<$ (менше), $=$ (дорівнює), \geq (більше або дорівнює), \leq (менше або дорівнює), а також знаки № (номер), % (процент);

– застосовувати індекси стандартів, технічних вимог та інших документів без реєстраційного номеру.

Написання чисел в тексті виконують в відповідності зі стандартом СТ РЕВ 543-73 “Числа. Правила запису та округлення”.

Числові значення величин в тексті слід вказувати зі ступенем точності, яка необхідна для забезпечення необхідних властивостей виробу, при цьому в ряді величин здійснюється вирівнювання числа знаків після коми. Округлення числових значень величин до першого, другого, третього і т.д. десяткового знаку для різних типорозмірів, марок і т.п. виробів одного найменування повинно бути однаковим. Дробові числа необхідно наводити у вигляді десяткових дробів, за виключенням розмірів в дюймах, які слід записувати $\frac{1}{4}$; $\frac{1}{2}$ ” (але не « $\frac{1}{4}$ », « $\frac{1}{2}$ »).

Якщо неможливо виразити числове значення у вигляді десяткового дроби, допускається записувати в вигляді простого дроби в один рядок через похилу риску, наприклад, $5/32$; $(50A-4C)/(40B+20)$.

Якість друку. Під час виконання ПЗ необхідно дотримуватись рівномірної щільності, контрастності й чіткості зображення впродовж усієї ПЗ. У звіті мають бути чіткі, не розпливчасті лінії, літери, цифри та інші знаки. **Помилки, описки та графічні неточності** допускається виправляти підчищенням або зафарбовуванням білою фарбою і нанесенням на тому ж місці

або між рядками виправленого зображення машинописним способом або від руки. Виправлення повинні здійснюватися ручкою чорного кольору.

Прізвища, назви установ, організацій, фірм та інші власні назви у звіті наводять мовою оригіналу. Допускається транслітерувати власні назви і наводити назви організацій у перекладі на мову звіту, додаючи (при першій згадці) оригінальну назву.

Скорочення слів і словосполучень – відповідно до чинних стандартів з бібліотечної та видавничої справи ДСТУ 3582-97.

Вимоги до оформлення ілюстрацій

Усі графічні матеріали (ескізи, діаграми, графіки, схеми, малюнки, креслення тощо) повинні мати однаковий підпис: «Рисунок».

Ілюстрації слід розміщувати у звіті безпосередньо після тексту, де вони згадуються вперше, або на наступній сторінці. На всі ілюстрації мають бути посилання у звіті.

Креслення, рисунки, графіки, схеми, діаграми, розміщені у звіті, мають відповідати вимогам стандартів.

Ілюстрації можуть мати назву, яку розміщують під ілюстрацією. За необхідності під ілюстрацією розміщують пояснювальні дані (підрисунковий текст).

Ілюстрація позначається словом "Рисунок ___", яке разом з назвою ілюстрації розміщують після пояснювальних даних, **наприклад перший рисунок розділу 3:**

Рисунок 3.1 – Схема роботи системи

Між назвою ілюстрації та подальшим текстом повинен бути один порожній рядок.

Ілюстрації слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу. Номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, відокремлених крапкою, наприклад, «рисунок

На всі таблиці мають бути посилання в тексті звіту.

Таблиці слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу. Номер таблиці складається з номера розділу і порядкового номера таблиці, відокремлених крапкою, наприклад, таблиця 2.1 – перша таблиця другого розділу. Таблиця може мати назву, яку друкують малими літерами (крім першої великої) і вміщують над таблицею. Назва має бути стислою і відображати зміст таблиці.

Висота рядків таблиці повинна бути не менше 8 мм.

Вимоги до оформлення формул та рівнянь

Формули та рівняння розташовують безпосередньо після тексту, в якому вони згадуються, посередині сторінки.

Вище і нижче кожної формули або рівняння повинно бути залишено не менше одного вільного рядка.

Формули і рівняння у звіті (за винятком формул і рівнянь, наведених у додатках) слід нумерувати порядковою нумерацією в межах розділу.

Номер формули або рівняння складається з номера розділу і порядкового номера формули або рівняння, відокремлених крапкою, наприклад, формула (1.3) – третя формула першого розділу.

Номер формули або рівняння зазначають на рівні формули або рівняння в дужках у крайньому правому положенні на рядку.

Пояснення значень символів і числових коефіцієнтів, що входять до формули чи рівняння, слід наводити безпосередньо під формулою у тій послідовності, в якій вони наведені у формулі чи рівнянні.

Пояснення значення кожного символу та числового коефіцієнта слід давати з нового рядка. Перший рядок пояснення починають з абзацу словом "де" без двокрапки.

Приклад оформлення першої формули розділу 3:

"Відомо, що:

$$S = F(T), \quad (3.1)$$

де T – базовий код вірусу;

S – зашифровані коди вірусу;

F – функція шифрування вірусу, що довільно вибирається з деякої множини перетворень {F}".

Переносити формули чи рівняння на наступний рядок допускається тільки на знаках виконуваних операцій, повторюючи знак операції на початку наступного рядка. Коли переносять формули чи рівняння на знакові операції множення, застосовують знак "x".

Формули, що йдуть одна за одною й не розділені текстом, відокремлюють комою.

Посилання в тексті пояснювальної записки на джерела слід зазначати порядковим номером за переліком посилань, виділеним двома квадратними дужками, наприклад, "у роботах [1 – 7] ...".

При посиланнях на розділи, підрозділи, пункти, підпункти, ілюстрації, таблиці, формули, рівняння, додатки зазначають їх номери.

При посиланнях слід писати: "... у розділі 4 ...", "... дивись 2.1...", "... за 3.3.4 ...", "... відповідно до 2.3.4.1...", "... на рис. 1.3 ...", або "...на рисунку 1.3 ...", "... у таблиці 3.2 ...", "... (див. 3.2) ...", "...за формулою (3.1)...", "... у рівняннях (1.23) – (1.-25) ...", "... у додатку Б...".

Оформлення додатків

КР обов'язково повинен містити додаток з вихідним кодом розробленої системи, все інше на вибір здобувача наприклад Інформаційно-логічна модель предметної області, Семантичне моделювання предметної області, тощо.

Кожен додаток слід починати з нової сторінки з вказівкою зверху посередині сторінки слова “Додаток” та його позначення.

Додаток повинен мати заголовок, який записують симетрично відносно тексту з прописної літери окремим рядком.

Додатки позначають великими літерами українського алфавіту, починаючи з А, окрім літер Г, Є, З, І, Ї, Й, О, Ч, Ь. Після слова “Додаток” йде літера, яка позначає його послідовність.

Оформлення лістинга програми

Формат тексту лістинга програми: шрифт **Courier New**, розмір шрифту **10 пт**, міжрядковий інтервал **1,0 пт**, вирівнювання - ліворуч.

В тексті програми необхідно у вигляді коментарів вказати прізвище та ініціали автора розробки, назву програмного забезпечення (тема курсової роботи) та ін. інформацію для забезпечення авторського права.

Якщо програмне забезпечення складається з декількох модулів, перед лістингом посередині сторінки жирним шрифтом Times New Roman, розмір 14 пт, необхідно вказати назву вихідного файлу (джерело програмного кода). Наприклад, “main.cpp”, “i_timer.h” тощо.

Коментарі є невід’ємною складовою тексту програмного забезпечення. Рекомендовано включати їх до лістингу безпосередньо під час розробки програмного забезпечення (а не після). Вони повинні коротко описувати кожен етап виконання програми (функції, процедури, модулі тощо). Мова коментарів - українська.

Оформлення графічних матеріалів. Умовні позначення

Структурна схема – це сукупність елементарних ланок об’єкта і зв’язків між ними. Під елементарною ланкою розуміють частину об’єкта, системи керування тощо, яка реалізовує елементарну функцію.

Елементарні ланки зображаються прямокутниками, а зв’язки між ними – суцільними лініями зі стрілками, що показують напрям дії ланки.

Функціональна схема – це схема, яка показує логіку роботи системи. Представляє собою схему пристрою, системи, апарату в якій основні вузли (блоки), що утворюють її, зображено прямокутниками та іншими фігурами, а зв’язок між ними показано лініями зі стрілками.

Функціональні схеми можуть виконуватися в менш деталізованому і в більш деталізованому вигляді. У першому випадку на схемі зображають найбільш важливі блоки системи і зв'язки між ними. У другому варіанті схема зображується більш детально, що полегшує її читання та більш повно ілюструє принцип роботи системи.

Блок-схема – опис алгоритму у вигляді блоків рішення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних фігур), які позначають такі елементи, як операції, потік, дані тощо.

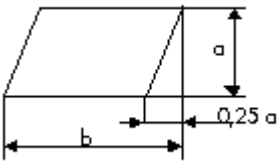
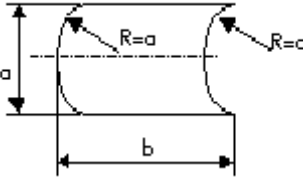
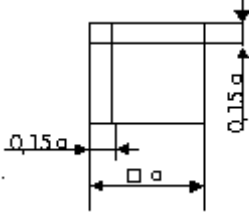
Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

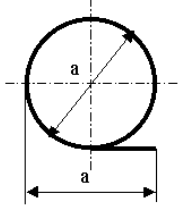
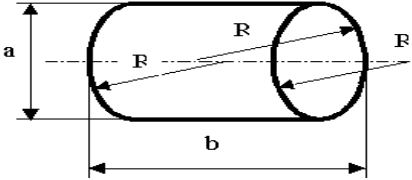
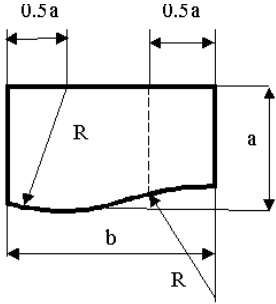
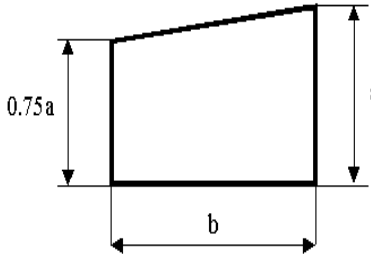
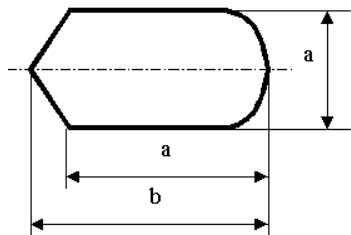
Приклад виконання блок-схеми наведено в Додатку Д.

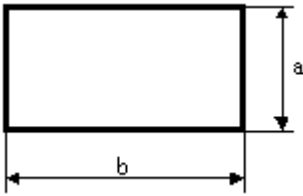
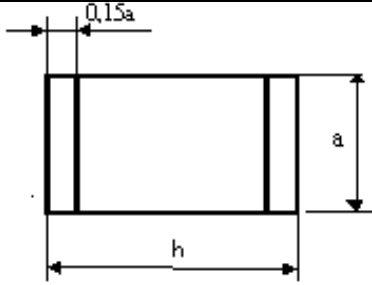
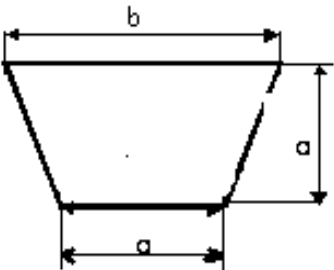
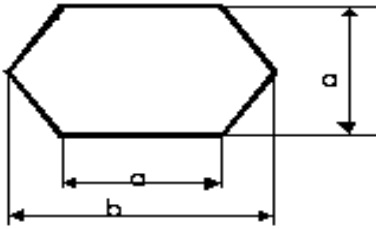
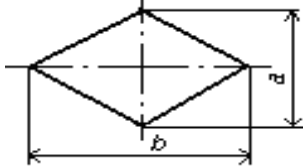
Правила виконання схем алгоритмів і програм


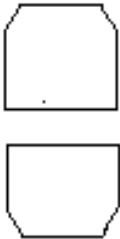

В таблиці 1 наведено умовні позначення регламентовані ДСТ 18.701-90.

Таблиця 1 – Умовні позначення встановлені ДСТ 18.701-90

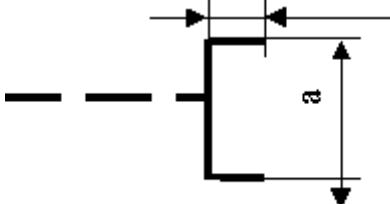
Назва	Позначення	Функції
<i>Символи даних</i> <i>Основні символи даних</i>		
Дані		Символ відображає дані, носій даних невизначений. (тут і далі $b = 2a$)
Дані, які запам'ятовуються		Символ відображає дані, що зберігаються у вигляді, придатному для обробки. Носій даних невизначений
<i>Специфічні символи даних</i>		
Оперативний запам'ятовуючий пристрій		Символ відображає дані, які зберігаються в оперативному запам'ятовуючому пристрої

<p>Запам'ятовуючий пристрій з послідовним доступом</p>		<p>Символ відображає дані, які зберігаються в запам'ятовуючому пристрої з послідовним доступом (магнітна стрічка, касета з магнітною стрічкою, магнітофонна касета)</p>
<p>Запам'ятовуючий пристрій з прямим доступом</p>		<p>Символ відображає дані, які зберігаються в запам'ятовуючому пристрої з прямим доступом (магнітний диск, магнітний барабан, гнучкий магнітний диск)</p>
<p>Документ</p>		<p>Символ відображає дані, які представлені на носії в формі, що зручно читається (машинограма, документ для оптичного або магнітного зчитування, мікрофільм, рулон стрічки, бланки вводу даних).</p>
<p>Ручне введення</p>		<p>Символ відображає дані, які вводяться вручну під час обробки з пристроїв будь-якого типу (клавіатура, перемикачі, кнопки, світлове перо, смуги з штриховим кодом).</p>
<p>Дисплей</p>		<p>Символ відображає дані, які представлені в зручній для людини формі на носії в вигляді пристрою, що відображається (екран для візуального спостереження, індикатори введення інформації).</p>

<i>Символи процесу</i> <i>Основні символи процесу</i>		
Процес		Символ відображає функцію обробки даних будь-якого виду (виконання визначеної операції або групи операцій, які приводять до зміни значення, форми або розміщення інформації або до визначення, за яким з декількох напрямків потоку слід рухатися).
<i>Специфічні символи процесу</i>		
Визначений процес		Символ відображає визначений процес, який складається з однієї або декількох операцій або кроків програми, які визначені в іншому місці (в підпрограмі, модулі).
Ручна операція		Символ відображає будь-який процес, що виконує людина
Підготовка		Символ відображає модифікацію команди або групи команд з метою впливу на деяку наступну функцію (встановлення перемикача, модифікація індексного реєстру або ініціалізація програми)
Рішення		Символ відображає рішення або функцію перемикаючого типу, який має один вхід та ряд альтеративних виходів,

		<p>один і тільки один з яких може бути активізований після обчислення вимог, визначених всередині цього символу. Відповідні результати обчислення можуть бути записані по сусідству з лініями, які відображають ці шляхи.</p>
Паралельні дії	 <p>Приклад.</p>	<p>Символ відображає синхронізацію двох або більше паралельних процесів</p>
Межа циклу		<p>Символ, складений з двох частин, відображає початок та кінець циклу. Обидві частини символу мають один і той же ідентифікатор. Вимоги для ініціалізації, приросту, закінчення і т.ін. розміщуються всередині символу на початку або в кінці в залежності від розміщення операції, яка перевіряє вимогу.</p>
<p><i>Символи ліній</i> <i>Основний символ ліній</i></p>		
Лінія		<p>Символ відображає потік даних або керування. При необхідності або для підвищення читання</p>

		можуть бути додані стрілки-показчики.
<i>Специфічні символи ліній</i>		
Передача керування		Символ відображає безпосередню передачу керування від одного процесу до іншого, іноді з можливістю прямого повернення до ініційованого процесу після того, як ініційований процес завершить свої функції. Тип передачі керування необхідно називати всередині символу (наприклад, запит, виклик, подія).
Канал зв'язку		Символ відображає передачу даних по каналу зв'язку
Пунктирна лінія		Символ відображає альтернативний зв'язок між двома або більше символами. Крім того, символ використовують для обведення анотованої ділянки.
<i>Спеціальні символи</i>		
З'єднувач		Символ відображає вихід в частину схеми з іншої частини цієї схеми і використовується для обриву лінії та продовження її в іншому місці. Відповідні символи-з'єднувачі повинні вмещувати одне і те ж унікальне

		позначення.
Термінатор		Символ відображає вихід в зовнішнє середовище та вхід із зовнішнього середовища (початок або кінець схеми програми, зовнішнє використання та джерело або пункт призначення даних).
Коментар		Символ використовують для описових коментарів або пояснювальних записів з метою пояснення або приміток. Пунктирні лінії в символі коментарю зв'язані з відповідним символом або можуть
	<p style="text-align: center;">Приклад.</p> 	обводити групу символів. Текст коментарів або приміток повинен бути розміщений біля обмежуючої фігури
Пропуск		Символ (три крапки) використовують в схемах для відображення пропуску символу або групи символів, в яких не визначені ні тип, ні число символів. Символ використовують тільки в символах лінії або між ними. Він застосовується головним чином в схемах, зображуючих загальне рішення з невідомим числом повторювань

Правила застосування символів та виконання схем

Символи в схемі повинні бути розміщені рівномірно. Слід дотримуватися розумної довжини з'єднувань і мінімальної кількості довгих ліній.

Символи повинні бути одного розміру. Не повинні змінюватися кути та інші параметри символів.

Мінімальну кількість тексту, необхідного для розуміння функції даного символу, слід розміщувати всередині даного символу. Текст для читання повинен записуватися зліва направо чи зверху донизу незалежно від напрямку потоку у відповідності з рисунком 2.

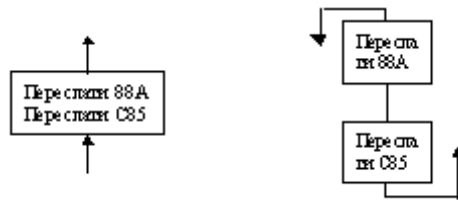


Рисунок 2 – Представлення тексту читання

Якщо об'єм тексту, який розміщується всередині символу, перевищує його розміри, слід використовувати символ коментарю. Якщо використання символів коментарю може заплутати або зруйнувати хід схеми, текст слід розмістити на окремому аркуші та давати перехресне посилання на символ.

В схемах може використовуватися ідентифікатор символів. Це зв'язаний з даним символом ідентифікатор, котрий визначає символ для використання в довідникових цілях в інших елементах документації (наприклад, в лістингу програми). Ідентифікатор символу повинен розміщуватися з лівого боку над символом у відповідності з рисунком 3.



Рисунок 3 – Приклад оформлення елементів блок-схеми

В схемах може використовуватися опис символів – будь-яка інша інформація, наприклад, для покращення розуміння функції як частини схеми. Опис символу повинен бути розміщений з правого боку над символом у відповідності з рисунком 4.



Рисунок 4 – Приклад оформлення елементів блок-схеми

Для використання в якості посилання на документацію текст на схемі для символів, що відображають способи виведення, повинен розміщуватися з правого боку над символом, а текст для символів, що відображають способи введення – з правого боку під символом у відповідності з рисунком 5.

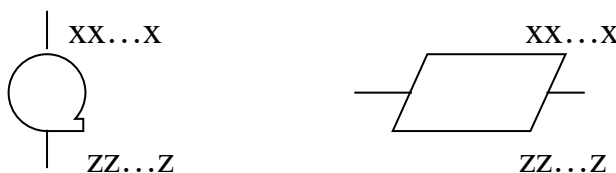


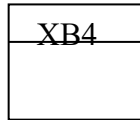
Рисунок 5 – Приклад оформлення елементів блок-схеми

В схемах може використовуватися докладний опис, який позначається за допомогою символу з смужкою для процесу або даних. Символ зі смугою вказує, що в цьому ж комплекті документації в іншому місці є більш докладний опис відміченого елемента блок-схеми.

Символ зі смугою являє собою будь-який символ, всередині якого в верхній частині проведена горизонтальна лінія. Між цією лінією та верхньою лінією символу розміщено ідентифікатор, що вказує на докладний опис даного символу у відповідності з рисунком 6.

В якості першого та останнього символу докладного опису повинен бути використаний символ вказівника кінця. Перший символ вказівника кінця повинен містити посилання, яке міститься також в символі зі смугою.

Символ зі смугою



Докладний опис

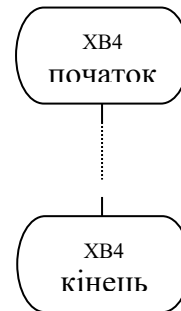


Рисунок 6 – Приклад оформлення елементів блок-схеми

Правила виконання з'єднань

Потоки даних або потоки керування в схемах показуються лініями. Напрямок потоку зліва направо та зверху донизу вважається стандартом. В випадках, коли необхідно ввести більшу ясність в схему (наприклад, при з'єднаннях), на лініях використовуються стрілки. Якщо потік має напрямок, відмінний від стандартного, стрілки повинні вказувати цей напрямок.

У схемах слід уникати перетину ліній. Лінії, які перетинаються, не мають логічного зв'язку між собою, тому зміна напрямку в точках перетину не допускається у відповідності з рисунком 7.



Рисунок 7 – Приклад оформлення елементів блок-схеми

Дві або більше вхідних ліній можуть об'єднуватися в одну вихідну лінію. Якщо дві або більше лінії об'єднуються в одну лінію, місце об'єднання повинно бути зсунуто у відповідності з рисунком 8.

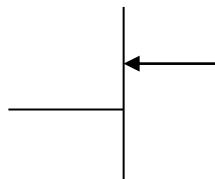


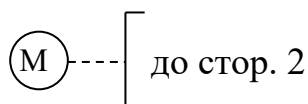
Рисунок 8 – Приклад оформлення елементів блок-схеми

Лінії в схемах повинні входити до символу або з лівого боку, або зверху, а виходити або з правого боку, або знизу. Лінії повинні бути спрямовані до центру символу.

При необхідності лінії в схемах слід розривати для уникнення зайвих перетинів або дуже довгих ліній, а також, якщо схема складається з декількох сторінок. З'єднувач на початку розриву називається зовнішнім з'єднувачем, а з'єднувач в кінці – внутрішнім.

Посилання до сторінок може бути приведені спільно з символом коментарю для їх з'єднувачів у відповідності з рисунком 9.

Зовнішній з'єднувач



Внутрішній з'єднувач

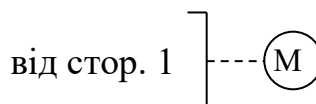


Рисунок 9 – Приклад оформлення елементів блок-схеми

Спеціальні умовні позначення

Декілька виходів

Декілька виходів із символу слід зображувати:

- 1) декількома лініями від даного символу до інших символів;
- 2) однією лінією від даного символу, яка потім розгалужується в відповідне число ліній у відповідності з рисунком 10.



Рисунок 10 – Приклад оформлення елементів блок-схеми

Кожен вихід із символу повинен супроводжуватися відповідними значеннями умов, щоб показати логічний шлях, який він представляє, з тим, щоб ці умови та відповідні посилання були ідентифіковані у відповідності з рисунком 11.

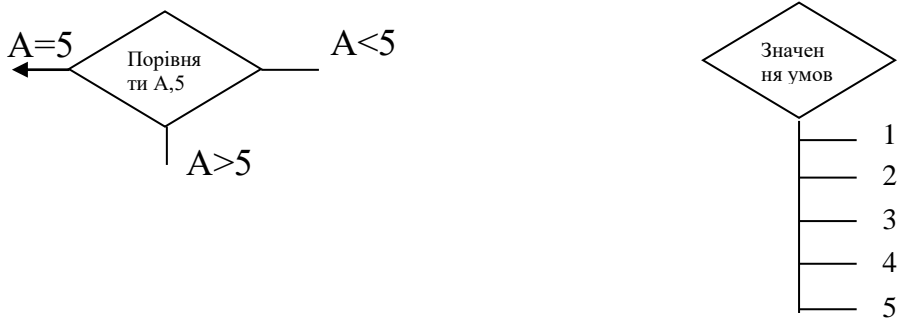


Рисунок 11 – Приклад оформлення елементів блок-схеми

Зображення, які повторюються

Замість одного символу з відповідним текстом може бути використано декілька символів з перекриттям зображення, кожен з яких містить описовий текст (використання або формування декількох носіїв даних або файлів, виробництво великої кількості копій друкарських звітів або форматів перфокарт) у відповідності з рисунком 12.

Коли декілька символів представляють упорядковану кількість, це упорядкування повинно розміщуватися від першого до останнього.

Лінії можуть входити або виходити з будь-якої точки перекритих символів, однак вищезгадані правила виконання з'єднань повинні виконуватися. Пріоритет або послідовний порядок декількох символів не змінюється за допомогою точки, в якій лінія входить або з якої виходить.



Рисунок 12 – Приклад оформлення елементів блок-схеми

ЗАХИСТ КУРСОВОЇ РОБОТИ ТА КРИТЕРІЇ ОЦІНЮВАННЯ

Захист курсової роботи здійснюється після її повного виконання (**див. розділ «ОСНОВНІ ЕТАПИ КР»**) й передзахисту у термін, встановлений календарним планом.

Оцінювання курсової роботи здійснюється за 100-бальною системою. Критерії оцінювання повноти виконання та захисту курсової роботи наведено в таблиці 3, оцінювання етапів (розділів) курсової роботи - таблиця 4.

Таблиця 3 - Критерії оцінювання повноти виконання й захисту КР

Кількість Балів	Вимоги до виконання та захисту курсової роботи
100-90	<p>Етапи курсового проектування виконано вчасно, курсову роботу подано на передзахист і захист у встановлені календарним планом терміни. Вчасно розроблено, погоджено і затверджено алгоритм та програмне забезпечення. Розробка має творчий підхід та оригінальні проектні рішення з елементами наукової новизни. Пояснювальну записку оформлено відповідно до встановлених вимог, теоретико-практичні аспекти розділів розкрито повністю, подано якісний графічний матеріал, структурні, функціональні схеми, використано вітчизняні та зарубіжні фахові видання.</p> <p>Захист курсової роботи передбачає чітко поставлену мету роботи, результати проведених досліджень і аналізу існуючого програмного забезпечення, опис і обґрунтування прийнятих проектних рішень щодо розробки програми, висновки; студент повинен мати ґрунтовні відповіді на всі додаткові питання.</p>
82-89	<p>Етапи курсового проектування виконано вчасно, курсову роботу подано на передзахист і захист у встановлені календарним планом терміни. Вчасно розроблено, погоджено і затверджено алгоритм та програмне забезпечення. Пояснювальна записка оформлена відповідно до вимог, теоретико-практичні аспекти розділів розкрито повністю. Захист курсової роботи передбачає чітко поставлену мету роботи, результати дослідження й аналізу існуючого програмного забезпечення, опис проектних рішень щодо розробки програми, висновки. Під час захисту курсової роботи студент в повному обсязі виклав теоретико-практичний матеріал, на поставлені додаткові питання мав ґрунтовні відповіді.</p>

75-81	<p>Етапи курсового проектування виконано вчасно, курсову роботу подано на передзахист і захист у встановлені календарним планом терміни. Алгоритм та програмне забезпечення є недосконалим та потребує подальшого вдосконалення, але відповідає темі курсової роботи. Пояснювальна записка оформлена відповідно до вимог, теоретико-практичні аспекти розділів розкрито повністю. Під час захисту проекта студент не в повному обсязі викладав матеріал, на поставлені питання мав нечіткі відповіді.</p>
67-74	<p>Курсову роботу подано до захисту з порушенням встановленого календарним планом термін. Алгоритм та програмне забезпечення є недосконалим, потребує подальшого вдосконалення, але відповідає темі курсової роботи. Пояснювальна записка оформлена відповідно до вимог, суть розділів розкрито не повністю. Під час захисту роботи студент відповів на більшість додаткових питань.</p>
60-66	<p>Етапи курсового проектування виконувались з систематичним запізненням, курсову роботу подано до захисту пізніше встановленого календарним планом терміну. Алгоритм та програмне забезпечення має суттєві недоліки, потребує подальшого вдосконалення, але відповідає темі курсової роботи. Пояснювальна записка оформлена з порушенням вимог, суть розділів розкрито не повністю. Під час захисту роботи студент не відповів на більшість додаткових питань.</p>
34-59	<p>Алгоритм, програмне забезпечення розроблено з грубими помилками та потребують суттєвого доопрацювання. Пояснювальна записка КР виконана з порушеннями вимог, матеріал викладено в неповному обсязі. (До захисту студент не допускається, курсову роботу - доопрацьовується.)</p>

Таблиця 4 - Критерії оцінювання курсової роботи

Назва критерії	Максимальна кількість балів
<i>Пояснювальна записка</i>	
Вступ	5
Розділ 1 Призначення та область використання	5
Розділ 2 Перегляд аналогічних існуючих систем	10
Розділ 3 Опис і обґрунтування проектних рішень	20
Розділ 4 Реалізація роботи вірність проектних і програмних рішень.	30
Розділ 5 Впровадження системи в промислову експлуатацію	10
Розділ 6 Основні висновки	5
Додатки	15
ВСЬОГО	100

За одержаними балами студент отримує оцінку за національною системою та ECTS відповідно до таблиці 5.

Таблиця 5 - Оцінки ECTS та національною системою відповідно до одержаних балів за КР

Кількість одержаних балів	ECTS	Національна система оцінювання
90-100	A	Відмінно
82-89	B	Добре
75-81	C	
67-74	D	Задовільно
60-66	E	
34-59	FX	незадовільно
1-34	F	

У разі непогодження з одержаною оцінкою за курсову роботу студент має право у п'ятиденний термін доопрацювати його та подати на повторний захист спеціально створеній екзаменаційній комісії.

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

Базова

1. Murphy M. L. Elements of Android Jetpack. [Електронний ресурс]. 2021. URL: <https://commonsware.com/Jetpack/Jetpack-FINAL.pdf>
2. Murphy M. L. Exploring Android. FINAL Version [Електронний ресурс]. 2021. URL: <https://commonsware.com/AndExplore/AndExplore-FINAL.pdf>
3. Murphy, M. L. Elements of Android Room. FINAL Version [Електронний ресурс]. 2021. URL: <https://commonsware.com/Room/Room-FINAL.pdf>
4. Google. Android Developers, офіційна документація та навчальні матеріали з розробки під Android [Електронний ресурс]. URL: <https://developer.android.com/?hl=en>
5. Google. Курси з Android розробки (включно з навчальними шляхами та codelabs) [Електронний ресурс]. URL: <https://developer.android.com/courses?hl=en>
6. Google. Курс Android Basics with Compose, базова розробка застосунків на Kotlin та Jetpack Compose [Електронний ресурс]. URL: <https://developer.android.com/courses/android-basics-compose/course?hl=en>
7. Google. Jetpack Compose, документація UI тулкіта та приклади [Електронний ресурс]. URL: <https://developer.android.com/develop/ui/compose/documentation?hl=en>
8. Google. Архітектура Android застосунку, рекомендації щодо шарів, ViewModel, repository та залежностей [Електронний ресурс]. URL: <https://developer.android.com/topic/architecture?hl=en>
9. Google. Android API Reference, довідник класів та API платформи [Електронний ресурс]. URL: <https://developer.android.com/reference?hl=en>
10. Kovalenko O., Popereshnyak S., Grinenko S., Grinenko O., Radivilova T. «Methods for Assessing the Maturity Levels of Software Ecosystems». *CEUR Workshop Proceedings Volume 2654*, 2019, Pages 251-261. Режим доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85091278920&origin=resultslist> (Scopus).
11. Коваленко О.В. Методи та засоби управління безпекою додатків. Інформаційно-керуючі системи на залізничному транспорті. №4, 2018. – С. 41-44. Режим доступу: http://nbuv.gov.ua/UJRN/Ikszt_2018_4_7 (Фахове видання. Категорія «Б»)
12. Коваленко О.В. Удосконалений метод управління ризиками розробки програмного забезпечення на основі напівмарковської моделі прийняття рішень. Сучасні інформаційні системи. – Випуск 2 (3). – Харків. – 2018. – С. 41-48. Режим доступу: <http://repository.kpi.kharkov.ua/handle/KhPI-Press/40480> (Фахове видання. Категорія «Б»)
13. Коваленко О.В. Методи якісного аналізу та кількісної оцінки ризиків розробки програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 3 (49). – Полтава: ПолтНТУ. – 2018. – С. 116-125. Режим доступу:

<http://journals.nupp.edu.ua/sunz/article/view/1146> (Фахове видання. Категорія «Б»)

14. Коваленко О.В. Моделі та методи розроблення програмного забезпечення комп'ютерних систем для підвищення безпеки даних: **монографія** / О.В. Коваленко // К.: Вид. «КОД» – 2019. – 295 с.

Допоміжна

15. Google. Тестування Android застосунків, підходи та інструменти (unit, instrumented, UI tests) [Електронний ресурс]. URL: <https://developer.android.com/training/testing?hl=en>

16. Google. Room, офіційна документація з роботи з локальною БД через ORM над SQLite [Електронний ресурс]. URL: <https://developer.android.com/training/data-storage/room?hl=en>

17. Google. DataStore, сучасне зберігання налаштувань та даних (Preferences, Proto) [Електронний ресурс]. URL: <https://developer.android.com/topic/libraries/architecture/datastore?hl=en>

18. Google. Практики безпеки Android застосунків, рекомендації щодо даних, дозволів і взаємодій [Електронний ресурс]. URL: <https://developer.android.com/privacy-and-security/security-best-practices?hl=en>

19. Google. Baseline Profiles, оптимізація продуктивності та швидкості запуску [Електронний ресурс]. URL: <https://developer.android.com/topic/performance/baselineprofiles/overview?hl=en>

20. Google. Профілювання в Android Studio, інструменти аналізу CPU, пам'яті, енергії та мережі [Електронний ресурс]. URL: <https://developer.android.com/studio/profile?hl=en>

21. Kotlin. Coroutines overview, основи корутин та асинхронності в Kotlin для Android [Електронний ресурс]. URL: <https://kotlinlang.org/docs/coroutines-overview.html>

22. Gradle. Gradle User Manual, документація системи збірки для Android проєктів [Електронний ресурс]. URL: <https://docs.gradle.org/>

23. Google. Налаштування Firebase для Android проєкту, підключення SDK та конфігурація [Електронний ресурс]. URL: <https://firebase.google.com/docs/android/setup?hl=en>

24. Google. Довідка Play Console, створення та первинне налаштування застосунку для публікації [Електронний ресурс]. URL: <https://support.google.com/googleplay/android-developer/answer/9859152?hl=en>

25. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 с.

26. Poul Klausen. JAVA 17: more about Java and Android software development. Bookboon. 2018. 257 с.

27. Peter Sommerhoff. Kotlin for Android App Development. Addison-Wesley Professional. 2019. 435 с.

28. Alexey Soshin. Kotlin Design Patterns and Best Practices. Packt Publishing. 2022. 513 с.
29. Pierre-Olivier Laurence, Amanda Hinchman-Dominguez. Programming Android with Kotlin. O'Reilly Media. 2022. 355 с.
30. Kevin D. Moore, Carlos Mota, Saeed Taheri. Kotlin Multiplatform by Tutorials. Razeware LLC. 2022. 400 с.
31. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 с.
32. Filip Babić, Luka Kordić, Nishant Srivastava. Kotlin Coroutines by Tutorials. Razeware LLC. 2022. 287 с.
33. Irina Galata, Victoria Gonda, Joe Howard, Ellen Shapiro. Kotlin Apprentice. Razeware LLC. 2021. 491 с.
34. René Cacheaux & Josh Berlin. Advanced iOS App Architecture. Razeware LLC. 2022. 334 с.
35. Ehab Amer, Alexis Gallagher, Matt Galloway, Eli Ganim, Ben Morrow, Cosmin Pupăză. Swift Apprentice. Razeware LLC. 2021. 555 с.
36. Neil Smyth. SwiftUI Essentials. Payload Media. 2022. 526 с.
37. Will Grant. 101 UX Principles. Packt Publishing. 2022. 432 с.

Інформаційні ресурси

38. Курс «Кросплатформені мови програмування» на сервері дистанційної освіти ЦНТУ. – URL: <https://moodle.kntu.kr.ua/course/view.php?id=1040>
39. Онлайн-курси UDEMY. – URL: <https://www.udemy.com/> – платформа онлайн-курсів різних ІТ тематик.
40. Онлайн-курси Prometheus. – URL: <https://prometheus.org.ua/> – українська платформа безкоштовних онлайн-курсів
41. Онлайн-курси Coursera. – URL: <https://www.coursera.org> – платформа онлайн-курсів різних ІТ тематик.
42. <https://biblprog.org.ua/ua/programming/> – каталог безкоштовних середовищ розроблення ПЗ.

Додаток А
(!!!Двосторонній друк документа!!!)

Центральноукраїнський національний технічний університет

Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Спеціальність (Вказується номер та назва)

ЗАТВЕРДЖУЮ
Керівник КР _____

«___» _____ 20__ р.

ЗАВДАННЯ
на курсову роботу студентів

Прізвище, Ім'я, По батькові

(прізвище, ім'я, по-батькові)

1. Тема роботи *Повна назва затвердженої теми*

2. Термін здачі студентом закінченої роботи XX травня 202X р.

3. Вихідні дані до роботи Технічне завдання

4. Вміст розрахунково – пояснювальної записки (перелік питань, що їх належить розробити)

1. Призначення та область застосування. б. Основні висновки.

2. Огляд існуючого програмного забезпечення.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Інструкція користувачу.

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

Структурна схема 1 аркуш

Функціональна схема 1 аркуш

Дизайн проекту 1 аркуш

Блок-схема роботи мобільного програмного забезпечення 2 аркуши

Дата видачі завдання « X » _____ 202X р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів курсової роботи	Строк виконання етапів роботи	Примітки
Вибір і затвердження теми курсової роботи.	__.__.202_ р.	
Збір та аналіз матеріалів за темою роботи.	__.__.202_ р.	
Постановка задачі та формування вимог.	__.__.202_ р.	
Проектування структури й алгоритмів роботи ПЗ.	__.__.202_ р.	
Програмна реалізація курсової роботи.	__.__.202_ р.	
Тестування, налагодження та аналіз результатів.	__.__.202_ р.	
Оформлення пояснювальної записки та підготовка до захисту	__.__.202_ р.	

Студент

(підпис)

Керівник

(підпис)

Додаток Б

Форма титульного аркушу пояснювальної записки для курсової роботи

ЗАТВЕРДЖЕНО

Наказ Міністерства освіти і науки,
молоді та спорту України
29 березня 2012 року № 384

Форма № Н-6.01

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Кафедра кібербезпеки та програмного забезпечення

КУРСОВА РОБОТА

з дисципліни: «Кросплатформені мови програмування»

на тему: Мобільне програмне забезпечення для...

Студента (ки) _____ курсу __-__ групи
спеціальності _____

(прізвище та ініціали)

Керівник _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS

Члени комісії _____
(підпис) (прізвище та ініціали)

_____ (підпис) (прізвище та ініціали)

_____ (підпис) (прізвище та ініціали)

м. Кропивницький 20 __рік

Додаток В
(довідниковий)

Основний напис для заголовних аркушів текстових документів
за ГОСТ 2.104-2006

<i>Вим.</i>	<i>Арк.</i>	<i>№ докум</i>	<i>Підп.</i>	<i>Дата</i>			
<i>Розроб.</i>					<i>Літ</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перев.</i>							
<i>Н. контр.</i>							
<i>Затв.</i>							

15

5

5

15

15

20

Основний напис для наступних аркушів текстових документів
за ГОСТ 2.104-2006

								<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум</i>	<i>Підп.</i>	<i>Дата</i>				

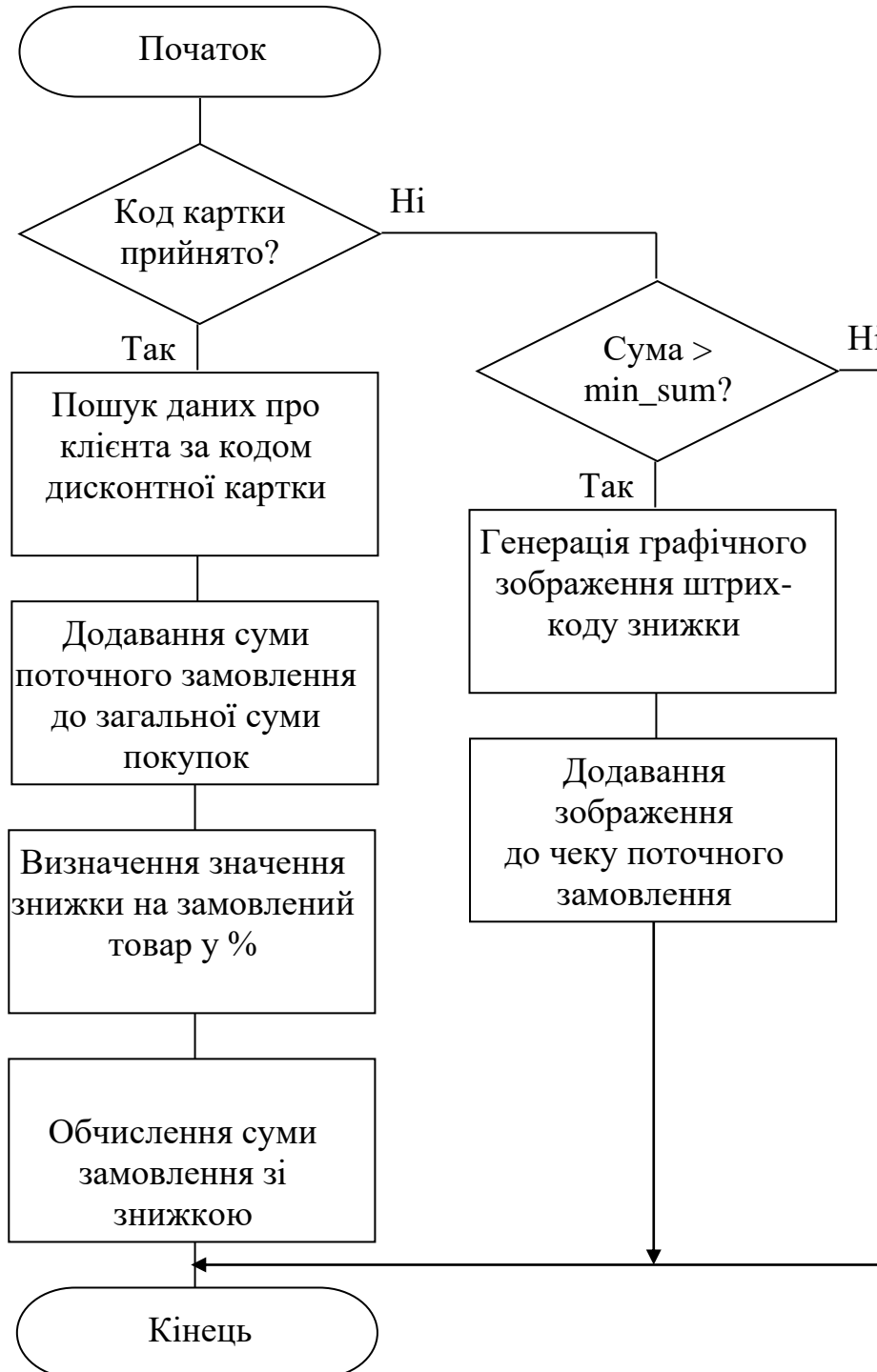
∞

10

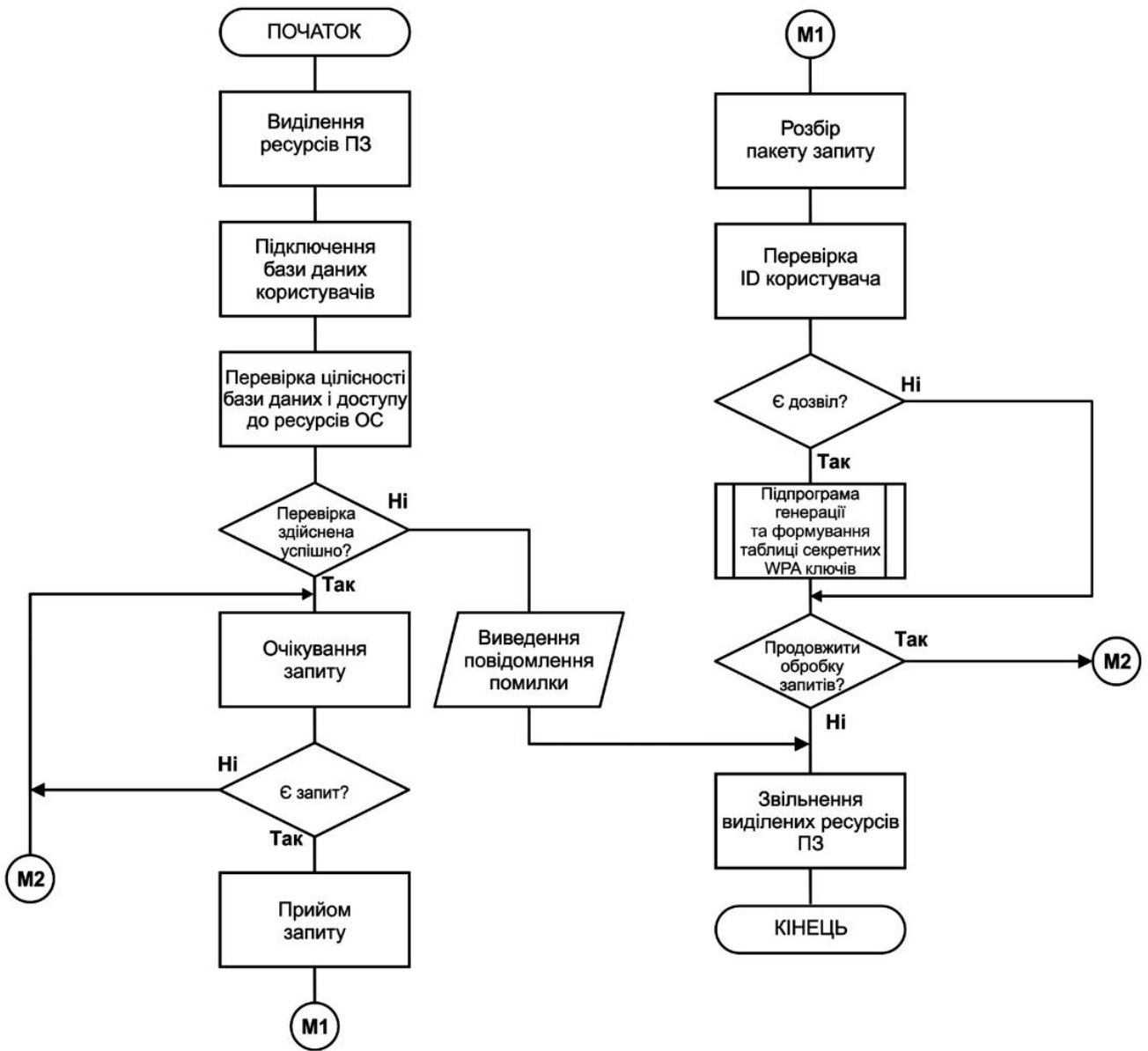
185

Додаток Г
(рекомендований)

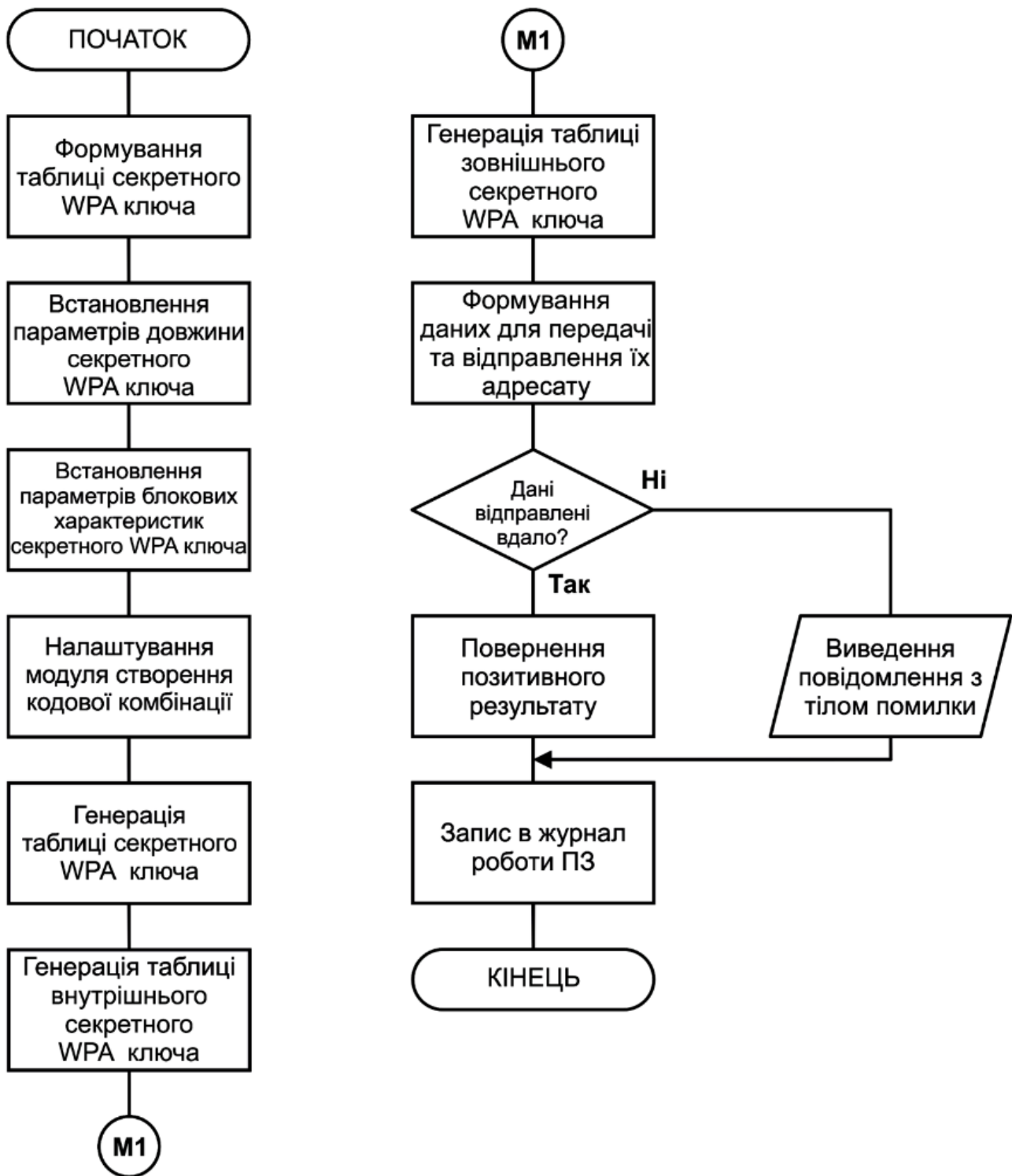
Блок-схема алгоритму роботи(приклад)
Часткова реалізація



Блок-схема роботи системи



Блок-схема роботи підпрограми



Додаток Д
Форма титульного аркушу програмного документу
Додаток Б

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ
Керівник курсової роботи
_____ Коваленко О.В.

Мобільне програмне забезпечення для ...

Лістинг програми

Код документу 12

Загальна кількість аркушів: ____

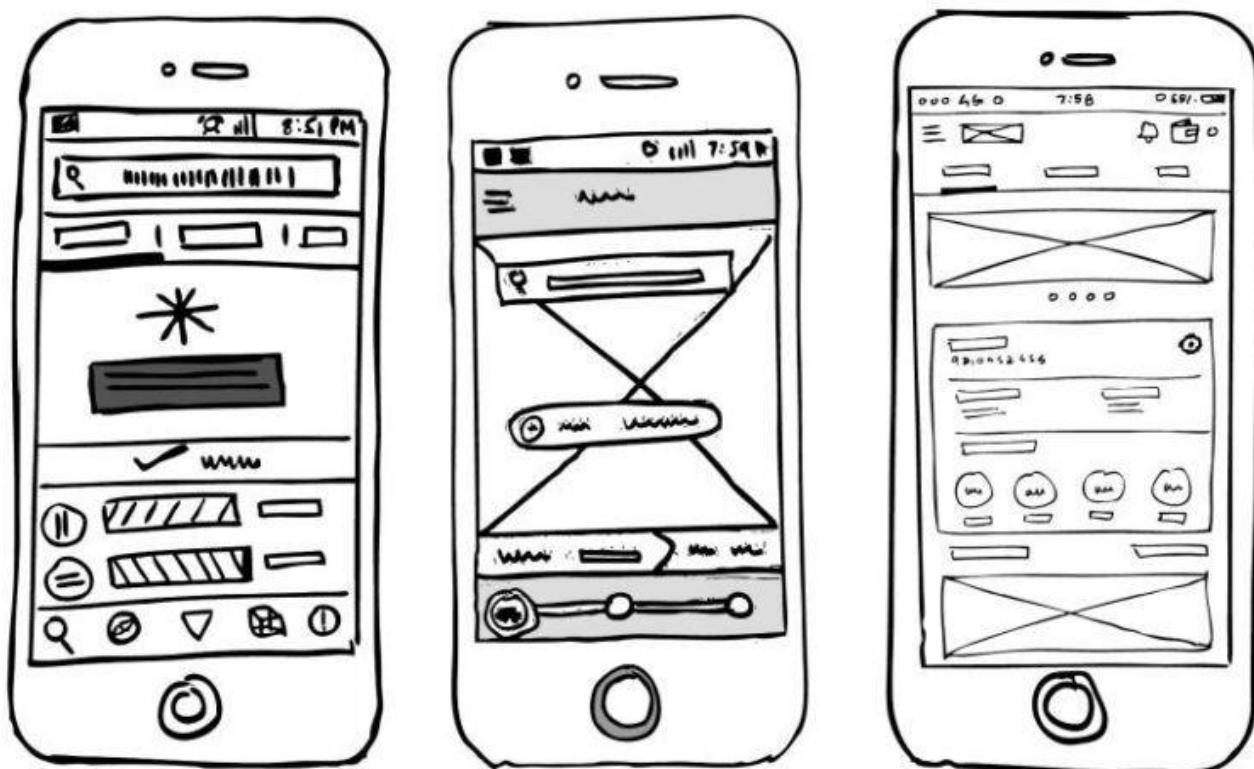
Літера: РП

Кропивницький 20__

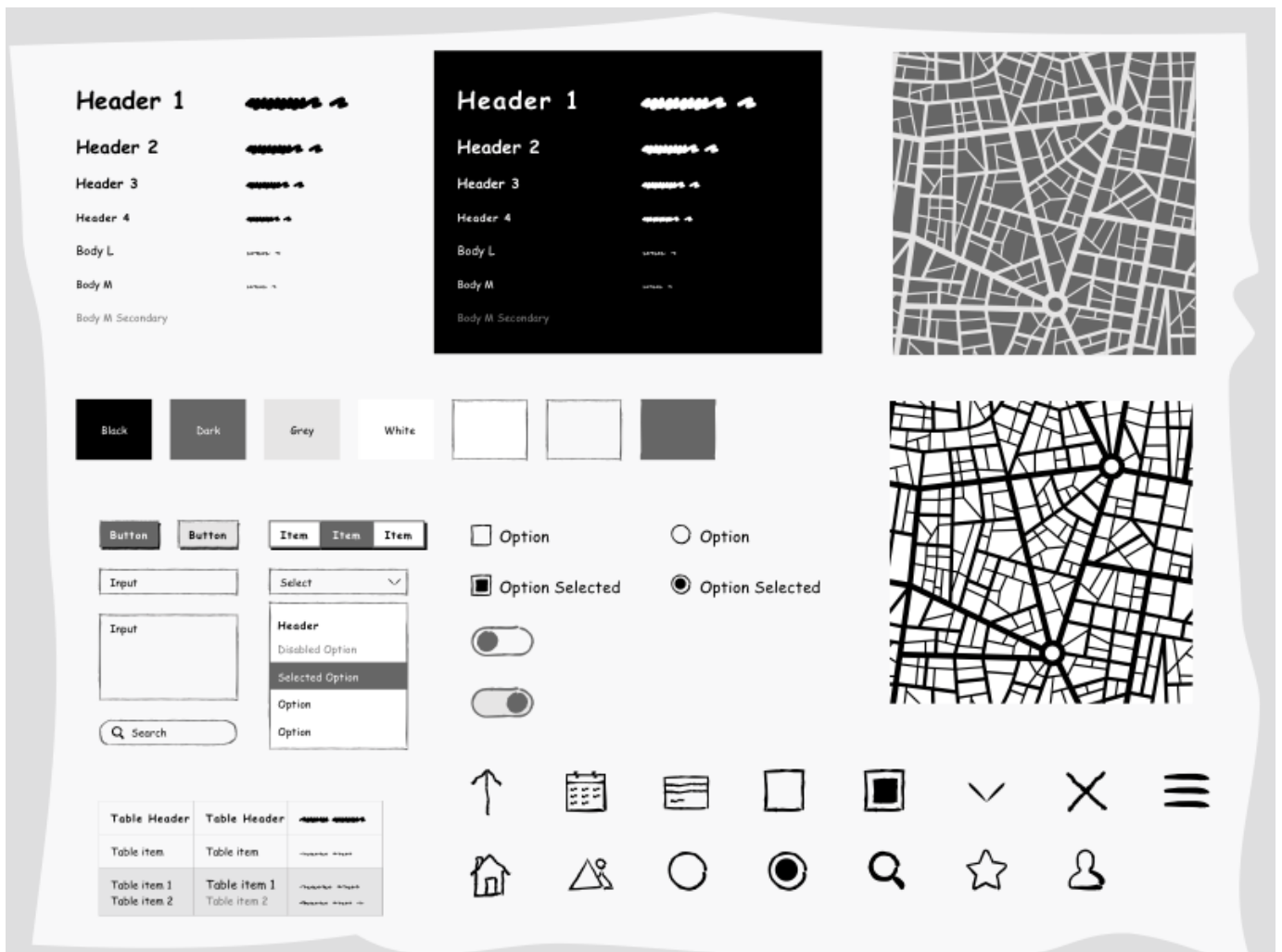
Додаток Е

Приклади представлення дизайну проекту в одну з трьох варіантів (обирається здобувачем на вибір) Low, Medium or High fidelity

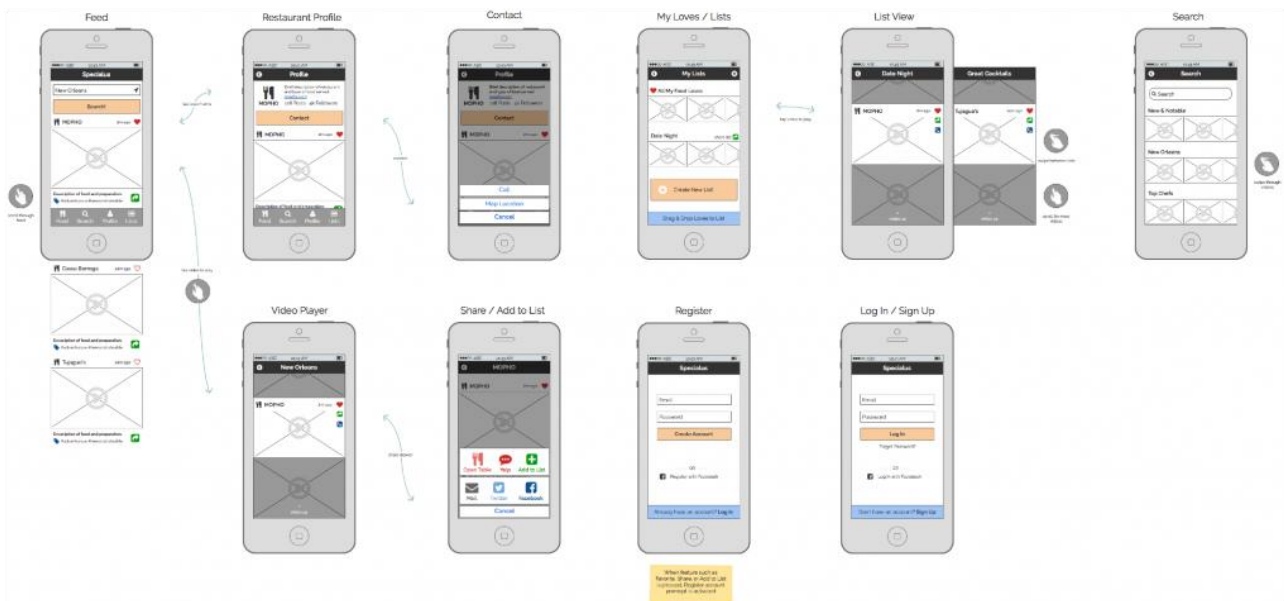
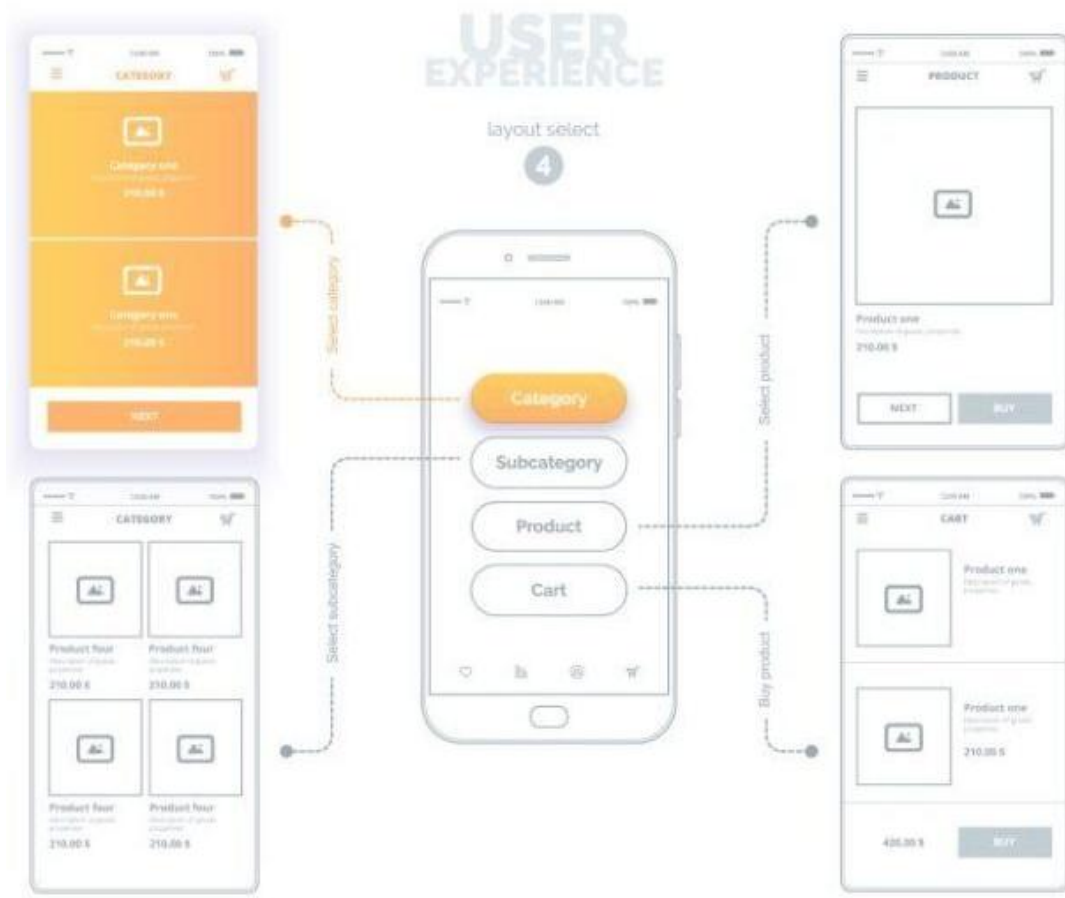
ПРИКЛАД РЕАЛІЗАЦІЇ LOW FIDELITY WIREFRAMES (НИЗЬКА ТОЧНІСТЬ)



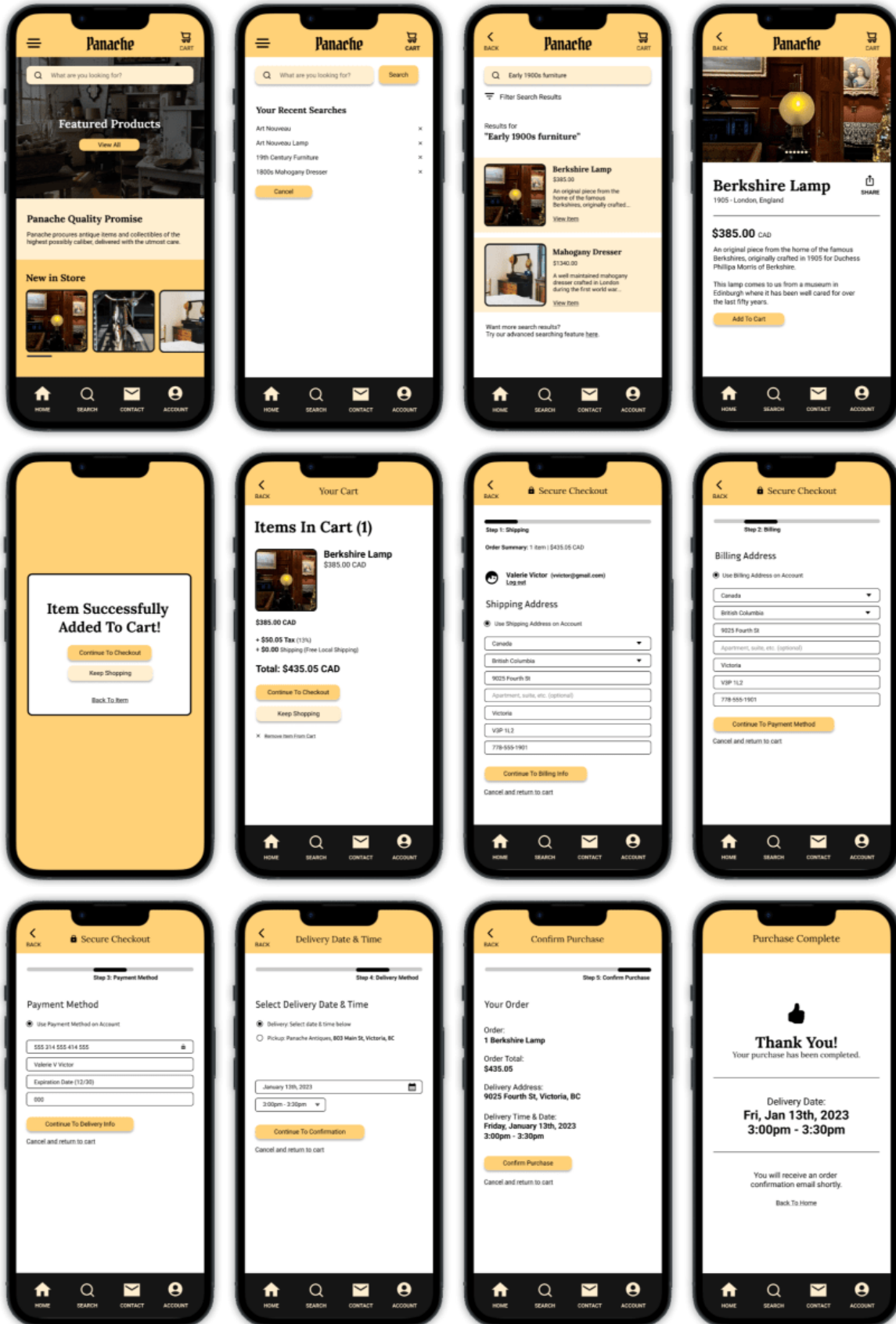
ПРИКЛАД ЕЛЕМЕНТІВ ЩО ВИКОРИСТОВУЮТЬСЯ ДЛЯ РЕАЛІЗАЦІЇ LOW FIDELITY WIREFRAMES (НИЗЬКА ТОЧНІСТЬ)



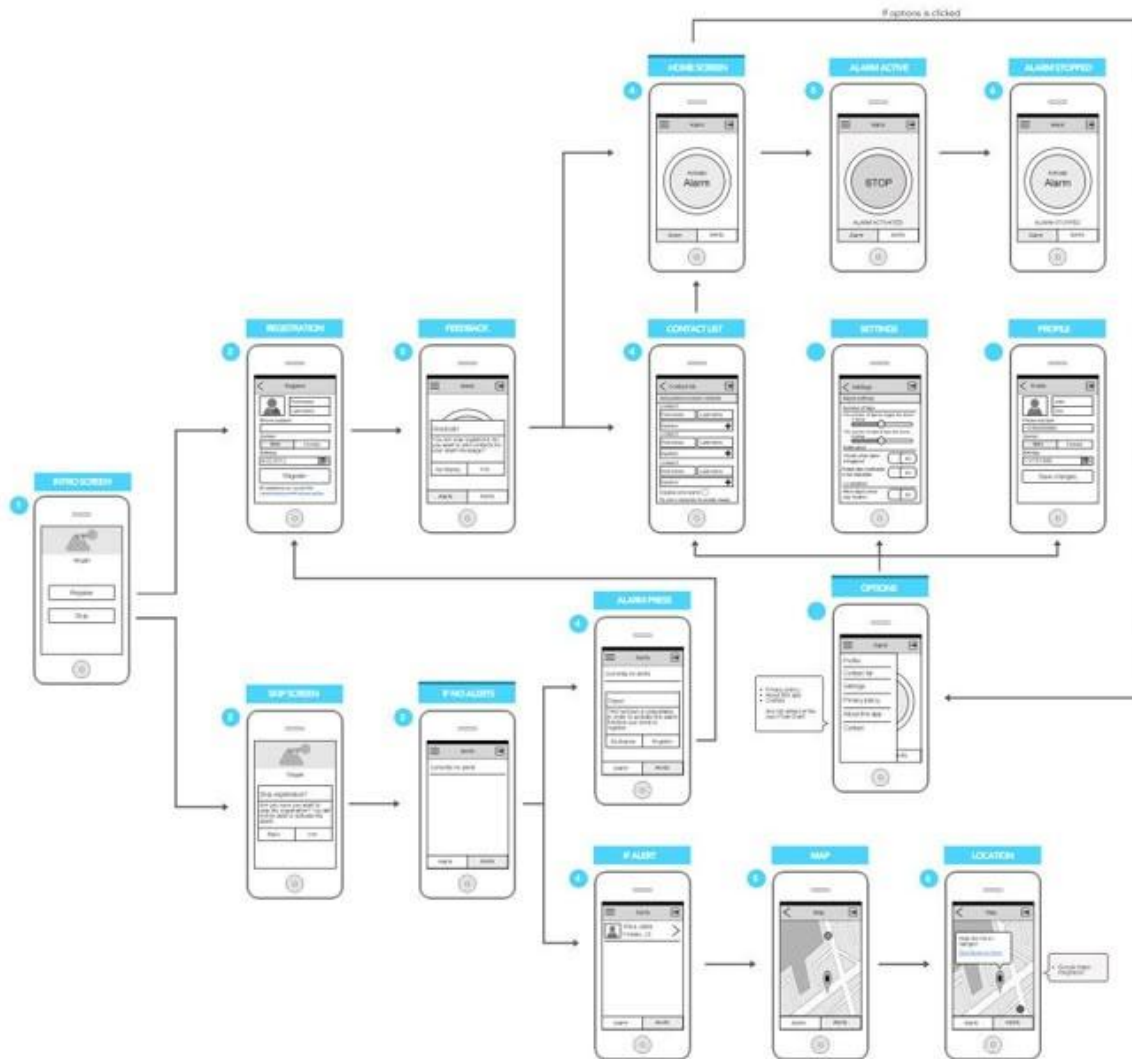
ПРИКЛАД ПЕРШОЇ РЕАЛІЗАЦІЇ MEDIUM FIDELITY MOCKUPS (СЕРЕДНЯ ТОЧНІСТЬ)



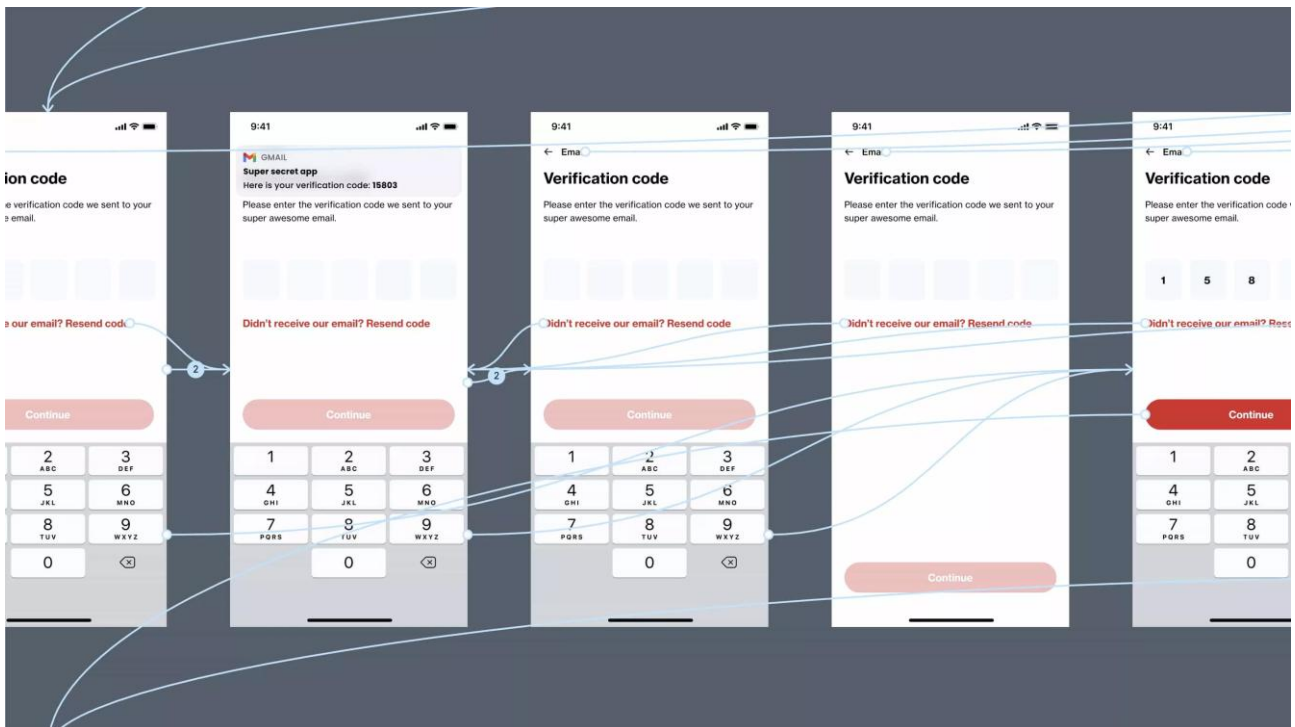
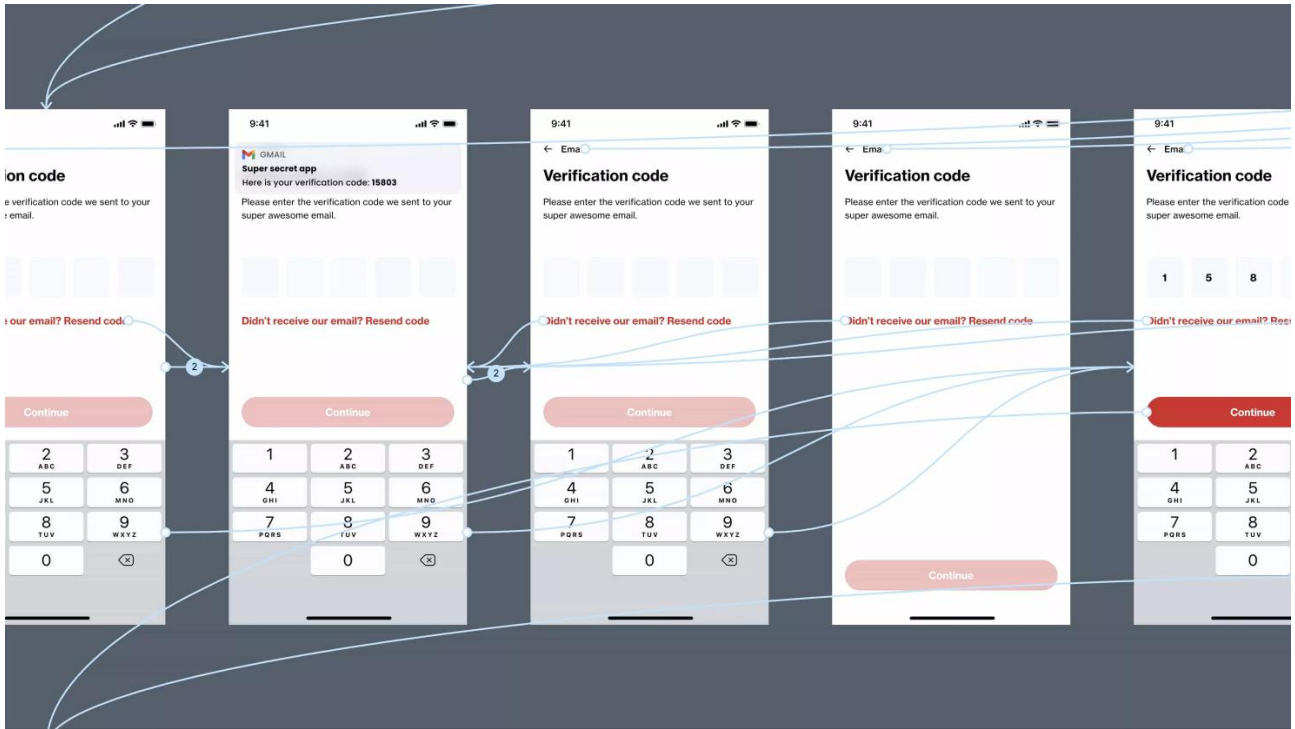
ПРИКЛАД ДРУГИЙ РЕАЛІЗАЦІЇ MEDIUM FIDELITY МОСКУПС (СЕРЕДНЯ ТОЧНІСТЬ)

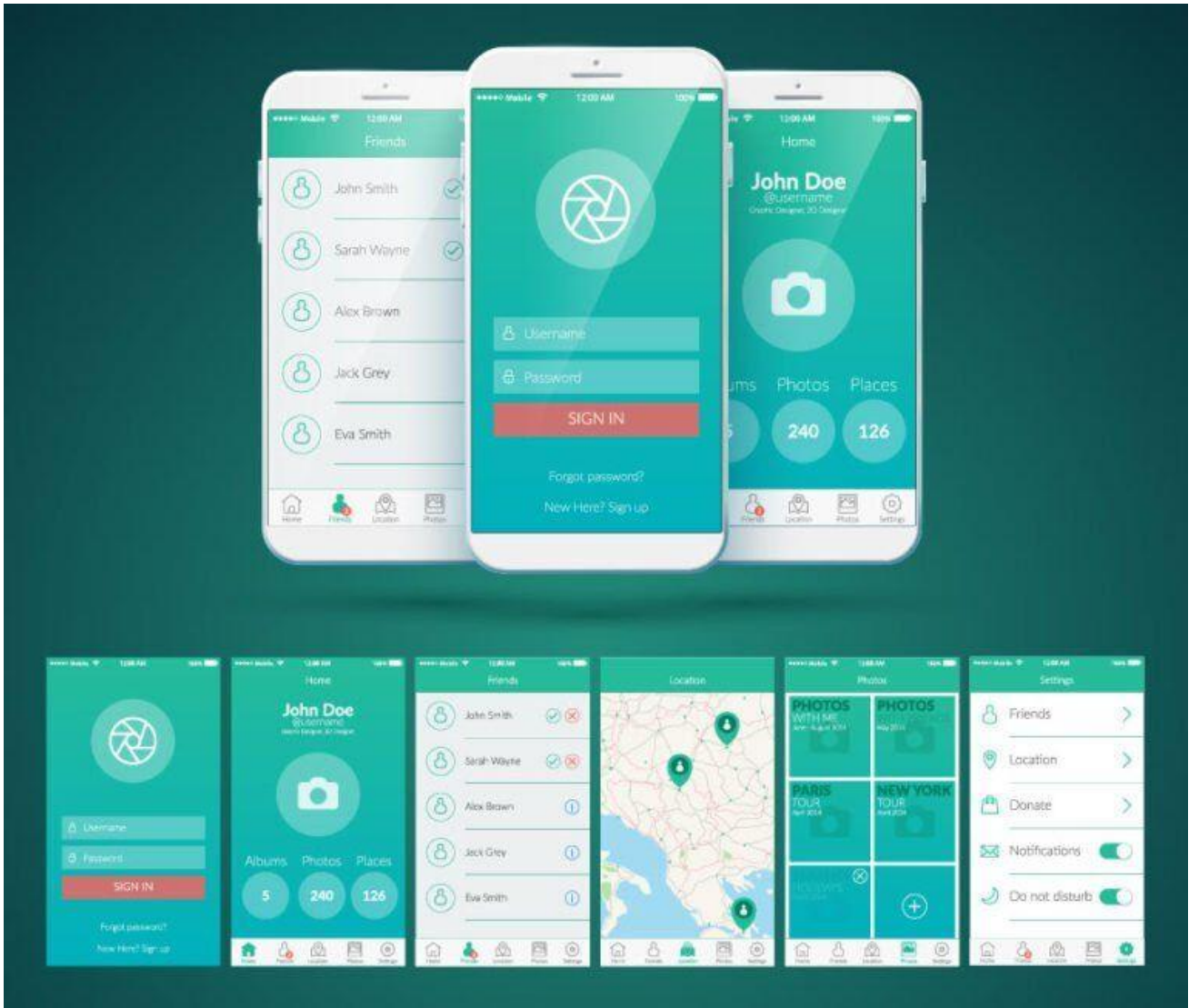


ПРИКЛАД ТРЕТІЙ РЕАЛІЗАЦІЇ MEDIUM FIDELITY MOCKUPS (СЕРЕДНЯ ТОЧНІСТЬ)



ПРИКЛАД РЕАЛІЗАЦІЇ HIGH FIDELITY PROTOTYPES (ВИСОКА ТОЧНІСТЬ) - Figma

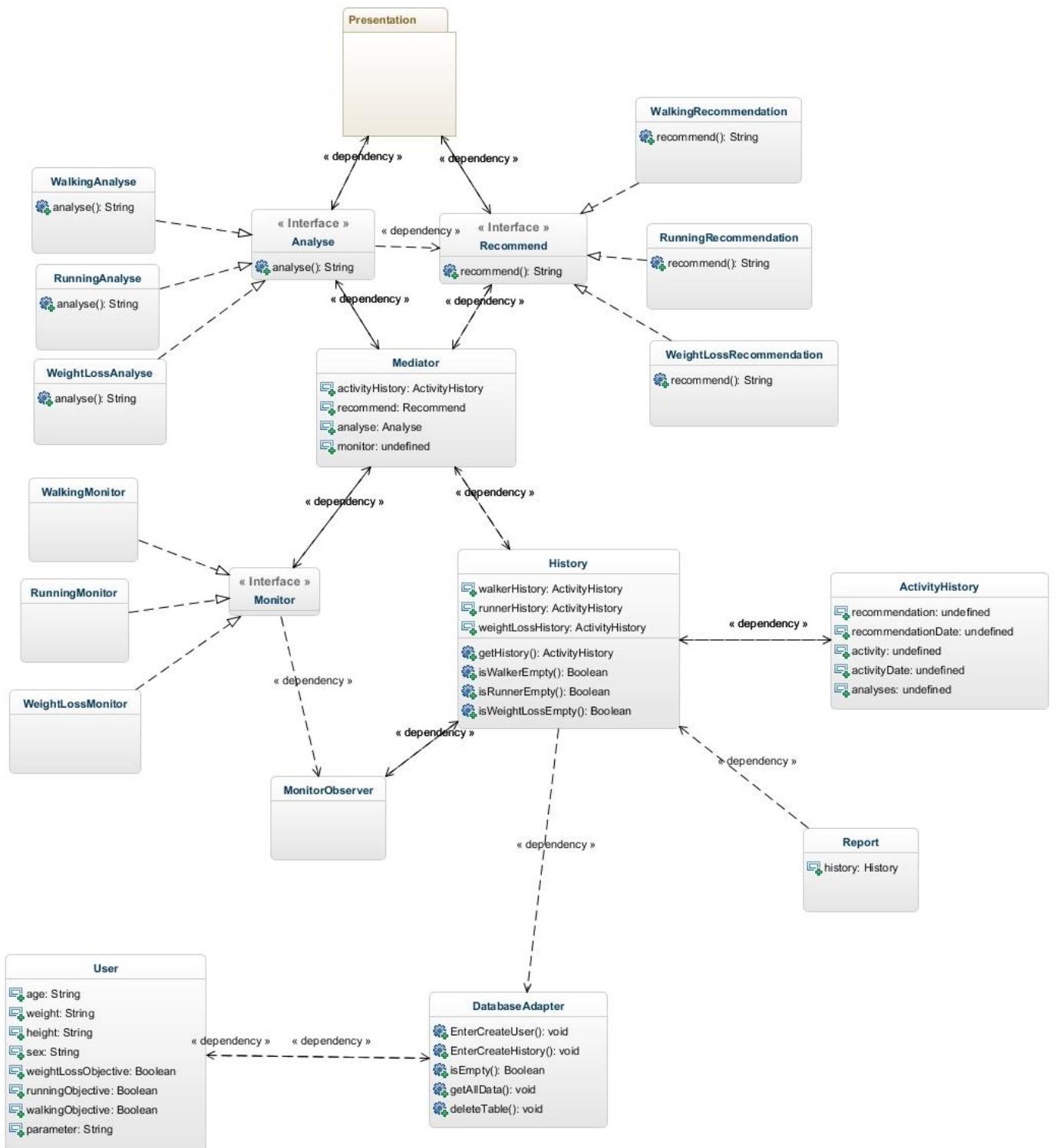




ПРИКЛАД РЕАЛІЗАЦІЇ

UML-діаграми класів. Нотації UML (Unified Modeling Language)

ОПЦІОНАЛЬНО



ПРИКЛАД ШАБЛОНУ ДЛЯ РЕАЛІЗАЦІЇ
Схеми потоку даних у мобільному додатку
ОПЦІОНАЛЬНО

Mobile App Data Flow Diagram Template

