

**КУПРЕЄВ ЮРІЙ ПАВЛОВИЧ**  
**"РОЗРОБКА ТЕМПЛЕЙТУ МОБІЛЬНОЇ ВЕБ-ВЕРСІЇ ДЛЯ**  
**ATUTOR "**

**керівник: д.т.н., проф. кафедри Ш Пастух Олег Анатолійович**

## АНОТАЦІЯ

**Купресв Ю.П. Розробка темплейту мобільної веб-версії для ATutor. – Рукопис.**

Кваліфікаційна робота здобуття освітнього ступеня «магістр» за спеціальністю 121 – Інженерія програмного забезпечення. – Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СПм-61//м. Тернопіль, 2020 //С. , рис. – , табл. – , додат. – , бібліогр. –

Мета кваліфікаційної роботи – використання сучасних інформаційних технологій для розробки шаблону, який дасть змогу з мобільного пристрою, без будь-яких змін та втрат на ідентичній версії, використовувати систему дистанційного навчання.

Практичне застосування – однозначно продукт є актуальним та має свою цільову аудиторію, а саме - всі зареєстровані в системі дистанційного навчання.

Технічні вимоги – базою шаблону запропоновано використати операційну систему Android. Інструментом для створення обрано високорівневу мову програмування Java.

**Ключові слова:** КВАЛІФІКАЦІЙНА РОБОТА, РОЗРОБКА, НОВИЗНА, ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, ТЕМПЛЕЙТ, ANDROID, JAVA, ОХОРОНА ПРАЦІ.

## ABSTRACT

### **Kupreyev Yu.P. Development of a mobile web version template for ATutor. – Manuscript.**

Qualifying work for obtaining a master's degree in specialty 121 — Software Engineering. – Ternopil Ivan Pul'ui National Technical University, Faculty of Computer Information Systems and Software Engineering, Software Engineering Department, group SPm-61 // Ternopil, 2020 // Pages. – , pictures. – , tables. – , supp. – , bibl.ref. –

The purpose of the qualification work is to use modern information technologies to develop a template. It will allow to use the identical version of the distance learning system by mobile device, without any changes or losses.

Practical application - the product is definitely relevant and has its target audience. There are all must be registered in the distance learning system.

Technical requirements - the template base is propose to use the Android operating system. The Java high-level programming language was chosen as the tool for creation.

**Key words:** QUALIFICATION WORK, DEVELOPMENT, NOVELTY, INFORMATION TECHNOLOGIES, TEMPLATE, ANDROID, JAVA, LABOR PROTECTION.

## ВСТУП

Актуальність теми. Вже ні в кого не викличе подиву чи сумнів той факт, що сучасні реалії все більше і більше заставляють нас всіх не просто замислюватись, а й переходити з реального світу у віртуальний. У сучасних дослідженнях останніх років, які я знайшов у вільному доступі, йдеться про те, що, для прикладу, у Північній Америці було понад 95 мільйонів користувачів мобільного зв'язку, і ця цифра майже наздоганяє показники щодо кількості Інтернет-користувачів присутніх на ринку. В інших же країнах, зокрема в країнах, які розвиваються, кількість користувачів мобільного зв'язку невинно зростає, в той час, як кількість Інтернет-користувачів залишалось досить незначним. Це говорить лише про те що сьогодні у світі поширена тенденція до збільшення використання мобільних пристосувань і це може бути зумовлено різними причинами.

В умовах не прогнозованої пандемії та карантинних обмежень, з метою збереження життя та здоров'я, людство сьогодні змушене не лише розважатись, а й комунікувати, влаштовуватись на роботу та працювати, здійснювати якісь побутові речі, покупки, консультуватись і навіть подекуди лікуватись лише віддалено. І сьогодні, як ніколи, на піку популярності сучасні інформаційні технології, оскільки саме вони використовуються не лише для підвищення ефективності діяльності, а й, без перебільшення, є можливістю взагалі вести будь-яку діяльність і залишитися в галузі у ці складні часи. Таким чином сьогодні вже самостійно формує задачі зі своїми вимогами та потребами. І зрозуміло, що осторонь не може залишитися жодна з галузей. Не виключенням є і освітній процес, як у закладах середньої освіти так і у спеціальних та вищих навчальних закладах.

Зрозуміло, що охопити роботою всі напрямки є неможливо та й не потрібно. Тому виокремлено окремий актуальний напрямок – вдосконалення системи дистанційного навчання через використання найпрогресивніших сучасних інформаційних технологій.

Напевно не варта нагадувати про достатньо очевидні проблеми дистанційного навчання. Це і недостатнє матеріально-технічне оснащення, очевидні проблеми у сімей, де є по декілька дітей і на всіх не вистачає техніки, та й зовсім тривіальні причини: проблеми в провайдері, технічні негаразди, тимчасова відсутність живлення в мережі та подібне, все це означає, що далеко не всі користувачі мають доступ до комп'ютерів та стаціонарного Інтернету. Але оскільки сьогодні все ж таки більшість учасників освітнього процесу змушені навчатись та виконувати вимоги освітнього процесу дистанційно і в більшості з дому, запропоновано продукт, який дасть змогу без будь-яких змін та втрат на ідентичній версії працювати і з мобільного пристрою.

Таким чином, запропонована система дасть можливість бути повноцінним учасником освітнього процесу використовуючи мобільні пристрої, зокрема, Android, оптимізувавши мобільну версію системи та використавши спеціальний додаток. Це дозволить будь-якому зареєстрованому користувачу дізнаватись все що стосується навчального процесу в цілому, обраного курсу зокрема, користуватись потрібною навігацією, користуватись скриньками та файлообмінником, складати модульні та підсумкові контролі, тобто дізнаватись всю корисну інформацію на екранах своїх пристосовань.

Таким чином, актуальність теми в розрізі розвитку відповідної прикладної галузі є очевидною.

Отже, метою дослідження є виявлення, опис та розробка шаблону як способу, інструменту і варіанту використання в

освітньому процесі гаджетів та якісна оцінка того, яку роль відіграє це використання з точки зору ефективності навчання на різних його етапах, ґрунтуючись на особистих оцінках а також інформації, отриманої в процесі спостережень.

Для досягнення поставленої мети потрібно пройти наступні етапи, які можуть уточнюватись та корегуватись в процесі виконання:

- сформулювати та погодити з керівником тему кваліфікаційної роботи;

- розробити та затвердити технічне завдання;

- проаналізувати технічне завдання;

- підібрати та проаналізувати бібліографічні матеріали українських та зарубіжних авторів, необхідні для виконання кваліфікаційної роботи;

- викласти загальні підходи та основні методи досліджень, основні методи проектування, розробки архітектури, підсистем і класів, обґрунтування засобів програмної реалізації;

- розробити допоміжну документацію;

- проаналізувати роботу щодо питань з дотримання положень про охорону праці та безпеку в надзвичайних ситуаціях;

- зробити відповідні висновки за результатами виконаної роботи.

Предметом дослідження є роль та використання гаджетів (чи інших портативних переносних електронних носіїв) на доступність, якість та ефективність освітнього процесу для студентів та інших зареєстрованих користувачів в системі дистанційного навчання вищого навчального закладу.

Об'єктом дослідження є розробка темплейту мобільної веб-версії для ATutor.

Саме ж вивчення ролі електронних гаджетів в освітньому процесі, зокрема, вищого навчального закладу буде здійснюватися за допомогою вирішення наступних дослідницьких завдань:

1) Виявити, яку роль відіграє постійне використання гаджетів: в процесі навчання студентів в розрізі якості, ефективності та зручності його; та для користувачів з викладацького складу.

2) Зрозуміти, що впливає та характер такого впливу при використанні гаджетів на ефективність, якість та зручність навчання:

- Від специфіки студентів;
- Від специфіки освітньої програми;
- Від особливостей використовуваних гаджетів.

3) Встановити, як самі користувачі оцінюють такий спосіб, його ефективність, якість та можливості безпосередньо на процес навчання і здачу іспитів.

# 1 АНАЛІТИЧНА ЧАСТИНА (Розробка програмної системи)

## 1.1. Аналіз вимог до програмної системи

### 1.1.1. Аналіз предметної області

Сьогодні тенденційно активно ростуть і збільшуються темпи продажів найрізноманітніших гаджетів. Це пов'язано і тим, що вони не лише просто впливають на наше повсякденне життя, а й слали його частиною. Слово «гаджет» з англійської мови перекладається, як «портативний технічний пристрій, основною функцією (функціоналом) якого є забезпечення діяльності в якійсь із галузей». Різноманітні комунікатори, планшетні комп'ютери, мобільні телефони, флешки – це одні з багатьох сьогодні вже самих звичних гаджети для нас.



Рисунок 1.1 – Приклад інструменту задіяного в освітньому процесі



Проте мене цікавлять ті пристрої, які використовуються студентами в процесі навчання, а користувачами з професорсько-викладацького складу та зацікавлених підрозділів вищого навчального закладу для роботи. Зазвичай до таких можна віднести ноутбуки, планшетні комп'ютери і смартфони. І сьогодні це питання набуло першочергової актуальності в розрізі складної епідеміологічної ситуації та суттєвих пандемічних обмежень одним із проявом яких є дистанційне навчання та загалом веденні освітнього процесу в дистанційній формі.

Однак, можливо й не варта нагадувати про достатньо очевидні проблеми дистанційного навчання. Це і недостатнє матеріально-технічне оснащення, очевидні проблеми у сімей, де є по декілька дітей і на всіх не вистачає техніки, тай зовсім тривіальні причини: проблеми в провайдера, технічні негаразди, тимчасова відсутність живлення в мережі та подібне, все це означає, що далеко не всі користувачі мають доступ до комп'ютерів та стаціонарного Інтернету. Але оскільки сьогодні все ж таки більшість учасників освітнього процесу змушені навчатись та виконувати вимоги освітнього процесу дистанційно і в більшості з дому, запропоновано продукт, який дасть змогу без будь-який змін та втрат на ідентичній версії працювати і з мобільного пристрою.

Очевидно що основними користувачами гаджетів є молоді люди, проте в одному випадку це призводить до того, що вони вчаться краще (додаткові можливості і т.п.), а в іншому випадку - використання гаджетів негативно чи навіть нейтрально впливає на процес навчання (відволікаються від головного, граються в ігри, не вміють шукати інформацію тощо). Мені також було цікаво ще й зрозуміти, чим це може бути обумовлено – специфікою самих студентів, специфікою освітньої програми або особливістю гаджетів.

Чому для одних використання гаджетів призводить до якісного стрибка в ефективності навчання, а для інших – воно є згубним і лише заважає освоєння навчальних дисциплін.

Ситуація на сьогодні з мобільним навчанням в Україні є дуже непростюю і в силу об'єктивних причин (фінансове становище сімей та вартість сучасних мобільних пристроїв, доступ до мережі, тощо) в тому числі. Оскільки для якісної освіти потрібно мобільний пристрій з великим екраном та сучасними можливостями. Більшість інформаційних ресурсів у сфері освіти - системи дистанційного навчання, електронні курси та портали - до використання на мобільних пристроях в своїй більшості просто не адаптовані. Тобто, щось може й буде працювати, але далеко не все і далеко не так, як зазвичай на ноутбучі або ПК.

Також складним питанням є й те, що дистанційне навчання, в своєму класичному вигляді, передбачає наявність різних медіакомпонентів у складі курсів. Для мобільного навчання частину з них потрібно переробляти, в більшості випадків - зменшити в об'ємі, оскільки при повністю мобільній освіті (з мобільного телефону через мобільний Інтернет) їх вивчення та робота з ними точно будуть неможливими.

На сьогодні, додатків для мобільного навчання, розроблених в Україні і для українських користувачів, ще дуже мало, в основному, вони дуже вузько направлені і не мають загальної доступності. Іншими словами, їх аналіз зробити є вкрай важкою задачею, оскільки доступ до ресурсів можуть отримати лише ті, хто зареєстрований в системі, а це професорсько-викладацький склад та студенти конкретного навчального закладу. А ті що вдалось переглянути часто викладачі не зовсім розуміють яким чином інтегрувати їх у навчальний процес при існуючих запропонованих можливостях.

Ще більше важливо пам'ятати про те, що мобільне навчання – це не просто освітні ресурси, доступні на сучасних мобільних пристроях. Це принципово нова філософія навчання. Разом з нею в навчальному процесі, повинні з'явитись абсолютно нові педагогічні підходи, які поєднують пристрої та образ життя сучасних студентів з навчанням і зроблять все це єдиним цілим. Потрібно зрозуміти, що саме буде і повинно працювати на мобільному пристрої, як саме воно повинно там працювати і для чого. Адже при дистанційному навчанні не йдеться про те, щоб перенести повністю все навчання на мобільний телефон або планшет. Мова йде про те, щоб можна було завжди цей доступний пристрій використовувати для підвищення якості освіти, та інтеграції навчального процесу в повсякденне життя студентів.

До речі, як тільки мова заходить про мобільне навчання, відразу варта згадати і про соціальне навчання, оскільки воно набуло найбільш інтенсивного розвитку саме тоді, коли мобільні пристрої так само як і постійне та регулярне перебування в соціальних мережах стали невід'ємною частиною життя великої кількості людей, що зробило абсолютно логічним і навчання в соціальних мережах. Таким чином, можна зробити висновок про те, що мобільне навчання та навчання в соціальних мережах – це певна єдина система доступного навчання, актуальна як ніколи саме сьогодні.

Ще одним користувачем мобільного навчання є компанії. І якщо навчальні заклади можуть мати якісь певні фінансові та методичні труднощі при його введенні, то їх бюджети в повному обсязі дозволяє спробувати щось нове, а компанії на ринку електронного навчання вже майже готові такі рішення надавати. І було б дуже раціонально, щоби саме тут, в цьому середовищі і почався (чи продовжився) бурхливий розвиток мобільного навчання.

Основною проблемою розвитку мобільного навчання в корпоративному секторі стали системи інформаційної безпеки, які не дозволяють ось так просто викладати (поширювати) навчальні матеріали компанії «назовні». На Заході цю проблему в основному вирішують за рахунок придбання спеціальних модулів системи дистанційного навчання, за допомогою яких здійснюється передача на мобільні пристрої в безпечному режимі. В українських системах дистанційного навчання, поки що таких напрацювання, як і реального стимулювання та фінансування таких наробок немає.

Цікавим є і те, що в процесі аналізу стан справ та тенденцій, я помітив, що навіть якщо виробники та заявляють, що курси та система на мобільних пристроях працюють, багато компаній все одно не готові через недостатньо безпечні з'єднання передавати настільки важливу інформацію.

### **1.1.2 Постановка задачі**

Освіту сміливо можна назвати одним із найдавніших і найбільш фундаментальних соціальних інститутів. Протягом багатьох століть люди вчилися і розвивалися, сформувавши певну систему засвоєння знань. Ця система росла і прогресувала, та розвивалась разом із людством, зайнявши основне місце в процесі соціалізації типового індивіда 21 століття. Свідомі молоді люди, ще не маючи дітей, вже думають про те, де і як їх майбутня дитина буде вчитися, розуміючи свою величезну відповідальність та важливість освіти в його житті. Кожен з нас є учасником освітнього процесу протягом свого життя: ми вчимося самі, потім вчимо і спостерігаємо за тим, як навчаються наші діти, хтось займається професійною викладацькою діяльністю. Освіта – це невід'ємна частина суспільства, його характерна ознака, на основі якої можна зрозуміти рівень його розвитку.

Зрозуміло, що сучасні тенденції тай ситуація в цілому не могли «не зачіпити» сферу освіти. Все більше і більше студентів по всьому світу не уявляють свого навчання без сучасних гаджетів.

Також не можна залишати поза увагою факт того, що використання інструментів інформаційних технологій у процесі освіти надає ще й додаткові можливості для якісного як вдосконалення так і здобування освіти, підвищення її інтенсивності та ефективності. Таким чином інформаційні технології, зокрема розроблюваний продукт надасть наступні основні можливості:

1 – комп'ютерна візуалізація навчальної інформації будь-якого змісту та характеру: як реальних так і віртуальних об'єктів вивчення;

2 – доступ та зберігання більших об'ємів інформації будь-якого характеру: графічна, текстова;

3 – можливість передачі великих об'ємів інформації;

4 – легка доступність інформації;

5 – можливість вирішення обчислювальних та інформаційно-пошукових завдань;

6 – можливість, у разі необхідності, контакту з викладачем у будь-який момент часу;

7 – миттєвий доступ до необхідного матеріалу чи ресурсу в рамках курсу матеріалів;

8 – спрощений доступ до контролю успішності та його ефективного та якісного профедення, як для викладача, так і для студента.

Таким чином, проаналізувавши стан справ та тенденції цього достатньо нового, але надзвичайно перспективного в умовах сьогоденнішніх реалій напрямку я виокремив найбільш часто впливаючі проблеми, по яким і буду формулювати конкретні

завдання, які можна буде вирішити за допомогою шаблону мобільної веб-версії для системи дистанційного навчання:

1 – в першу чергу потрібен сучасний пристрій (це ніби і не проблема для вирішення в даній роботі, але потрібно врахувати можливість середньо статистичного користувача, та присутній на ринку клас пристроїв, який би відповідав вимогам і одночасно не був надто дорогим, тобто був би доступним);

2 – фінансові труднощі, які часто унеможливають замовлення розробки будь-якого програмного продукту, в тому числі і шаблону (так само як і перша проблема, ніби і дотично стосується роботи, проте у моєму випадку, якраз і знімається, оскільки тема не потребує жовних капіталовкладень, оскільки розробка здійснюється в рамках дипломного проектування, для конкретного замовника);

3- проблеми в провайдера, технічні негаразди (так само як друге і третє – є усунє, однак потребує врахування, тобто можливість поновлення в системі дистанційного навчання без втрати попередніх дій чи проходжень);

4 – системи інформаційної безпеки (безпека забезпечуватиметься:

- індивідуальним логіном,
- паролем, який повинен буде бути комбінованим та складатися з не менше ніж 10 символів. Також пароль, якщо система «вважатиме» його запростим, на відповідний запит потрібно буде замінити. В іншому разі всі наступні дії будуть неможливими,
- блокування системи через певний проміжок часу при неактивності,
- різний рівень доступу до матеріалів та інструментів курсу (викладацький та навчально-користувацький);

5 – відволікаються від головного (оскільки передбачається для для використання потребуються сучасні гаджети, то зрозуміло, що їх можливості будуть використовуватись на повну: гра в ігри, не вміння чи не бажання пошуку, пошук зовсім у іншому напрямку, тому буде передбачено фіксацію факту неактивного використання програми чи перехід). Також мені було дуже цікаво ще й зрозуміти, чим це може бути обумовлено – специфікою самих студентів, незацікавленістю студентів, специфікою освітньої програми або особливістю гаджетів;

6 – медіакомпонент у складі курсів. Для мобільного навчання частину з них потрібно переробляти, в більшості випадків – зменшити в об'ємі, оскільки при повністю мобільній освіті (з мобільного телефону через мобільний Інтернет) їх вивчення та робота з ними точно будуть неможливими. Основна проблема тут – не втратити якість.

7 – вузько направленість (вони дуже вузько направлені і не мають загальної доступності. Іншими словами, їх аналіз зробити є вкрай важкою задачею, оскільки доступ до ресурсів можуть отримати лише ті, хто зареєстрований в системі, а це професорсько-викладацький склад та студенти конкретного навчального закладу. А ті що вдалось переглянути часто викладачі не зовсім розуміють яким чином інтегрувати їх у навчальний процес при існуючих запропонованих можливостях);

8 – ціленаправлена адаптованість (більшість інформаційних ресурсів у сфері освіти, а це: системи дистанційного навчання, електронні курси та портали, до використання на мобільних пристроях в своїй більшості просто не адаптовані. Тобто, щось може й буде працювати, але далеко не все і далеко не так, як зазвичай на ноутбуці або ПК);

9 – повна чи часткова відсутність методики, алгоритму дистанційного навчання (навіть якщо в більшості вищих навчальних закладах до мобільного навчання технічно готові – пристроїв та якості інтернету, а також можливостей сервера цілком достатньо, методика, на основі якої можна було зрозуміти, що таке мобільне навчання, як його використовувати та як для нього готувати навчальні матеріали та навчальний процес не поширюється. Це є наслідком того, що вважається таке явище тимчасовим. Однак, як тільки мобільне навчання буде введено як частина навчального процесу, і стане зрозумілою та загальноприйнятою на рівні відповідних міністерств та відомств методика тобто стане зрозуміло, що-коли-як-протягом якого терміну робити – розвиток в цьому напрямку піде наповну і в повному обсязі).

### **1.1.3 Пошук актантів та варіантів використання**

Сценарії поведінки, прецеденти, діаграми варіантів використання – все це вихідне концептуальне подання системи впродовж процесу як її проектування так і процесу розробки. До такої діаграми, в якості складових можуть входити: актори, варіанти використання, а також взаємозв'язки між ними. примітки і механізми розширення є загальними елементами нотації, які можуть використовуватись при побудові діаграм.

Якщо розглядати структуру, то така діаграма складатиметься з: проєктованої системи, яку представлено у вигляді якоїсь оптимальної кількості акторів, взаємодія яких із системою відбувається при допомозі варіантів використання. Дійова особа, актант, актор тут це як правило якийсь об'єкт, суб'єкт або ж система, яка може ззовні взаємодіяти з модельованою системою. Варіантом використання, в свою чергу, можна назвати специфікацію сервісів (або функцій), яка



надається актору системою. Таким чином, кожний варіант використання може визначати певну множину дій, які реалізуються за допомогою систем під час певної взаємодії із актором. Важливо відзначити, що тоді моделлю ніяк не відображається те, як буде реалізовуватись ця множина активностей(дій).

Якщо говорити про структурний підхід, то аналогом для діаграм варіантів використання може бути діаграма IDEF0 та DFD, варіантів використання – роботи (IDEF0) і процеси (DFD), акторів – зовнішні сутності (DFD).

Відомо, відповідно до UML актора (графічно) можна представити трьома способами (рисунок 1.2):

- 1 – чоловічком;
- 2 – класом із текстовим стереотипом «актор»;
- 3 – довільне графічне представлення.

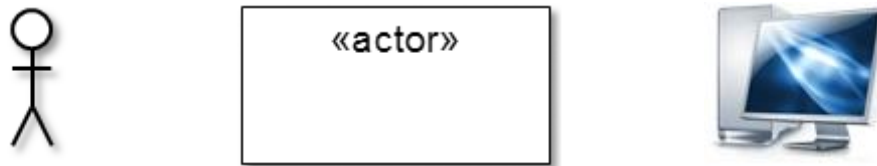


Рисунок 1.2 – Способи представлення актора

Внаслідок того, що діаграми варіантів використання є аналогом діаграм IDEF0 (англ. Function Modeling є методологією функціонального моделювання та графічного опису процесів) (рисунок 1.3) та DFD (англ. Data Flow Diagrams – діаграма потоків даних) (рисунок 1.4), методології їх розробки багато в чому збігаються. Тому вважає за необхідне описати алгоритм їх створення:

1 – спочатку варта побудувати контекстну діаграму, яка відобразить основні варіанти використання (функції) системи. Після чого для кожного з них побудувати та розглянути діаграми декомпозиції (деталізації).

Потрібно звернути увагу і на те, що контекстна діаграма може являти собою незв'язних граф (на відміну від IDEF0 і DFD).

2 – слід уникати надмірної деталізації діаграми варіантів використання. Цього не потрібно. Проте, слід пам'ятати, що варіант використання – це відносно великий блок функціональності системи, а в метю деталізації, в подальшому, будуть використовуватися інші види діаграм, що більше призначені для цієї цілі.

3 – з метою простішого та зрозумілішого сприйняття діаграма (контекстна чи декомпозиційна) не повинна бути перенасичуватись елементами. Рекомендовано на одній діаграмі відображати (представляти) не більше ніж 15 варіанти використання.

4 – розташовувати елементи потрібно таким чином, щоби була очевидною логічна послідовність виконання варіантів використання і було мінімальна кількість перетинів між відношеннями, тобто їх кількість не просто зменшено, а була б оптимальною.

5 – до початку побудови діаграми необхідно задокументувати потоки подій в системі. Потік подій – це процес обробки даних, реалізований в рамках одного або декількох варіантів використання. Опис потоку включає інформацію про те, які обов'язки покладаються на акторів, а які - на систему. Сюди включено:

- короткий опис поведінки, що реалізується у варіанті використання;

- передумови – умови, яких необхідно дотримуватись, до того як варіанти використання можуть бути задіяні. Для прикладу, такою умовою може бути завершення виконання іншого варіанту використання або наявність у користувача прав доступу;

- основний потік подій описує, що має відбуватися під час виконання варіанта використання в найбільш поширеному (типовому)

випадку. В цьому випадку дочірні варіанти використання пов'язані з базовим ставленням включення;

- альтернативні потоки подій описують виняткові ситуації (наприклад, введення неправильного пароля або необхідність виконання додаткових дій). Дочірні варіанти використання при розробці діаграми пов'язуються з базовим ставленням розширення;

- постумови – умови, які повинні бути виконані після завершення варіанту використання. Наприклад, такою умовою може бути обов'язкове збереження результатів розрахунку в базі даних на сервері.

б - на діаграмах не слід відображати особливості реалізації варіантів використання і внутрішньої організації системи, пов'язані зі специфікою використовуваних програмних і апаратних засобів. Дані діаграми в першу чергу призначені для спільного з замовником визначення функціональних вимог до системи. Тому розуміти (інтерпретувати) відображене на діаграмах і замовник і розробник повинні однаково.

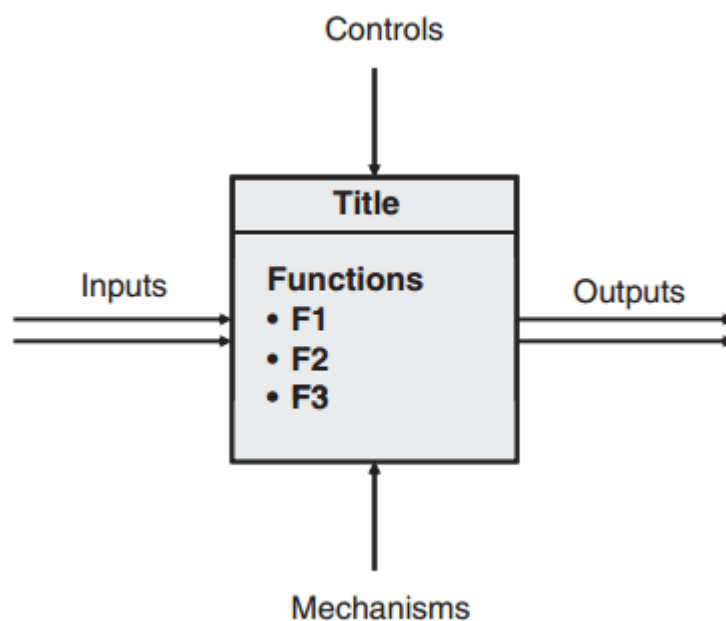


Рисунок 1.3 – Function Modeling

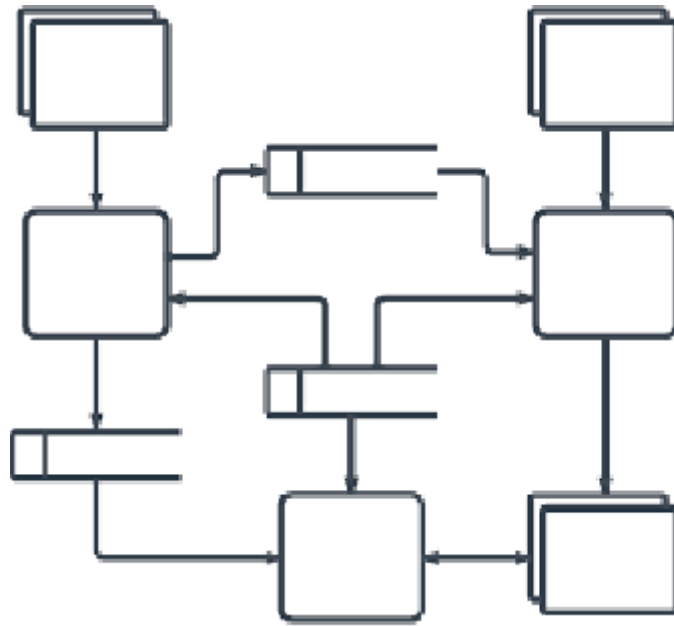


Рисунок 1.4 – Data Flow Diagrams

## **2 ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ**

Після визначення вимог до програмного забезпечення, я отримав вже варіант погодженого чіткого плану дій, графіку та термінів виконання. Це дозволить скоротити затрати часу на розробку та підвищить її якість, а також дозволить передбачити ще якісь інші нюанси, що можуть виникнути під час процесу розробки, для прикладу, вкрай важливі питання пов'язані з авторськими правами на програмне забезпечення та подібне.

В результаті попереднього проектування програмного забезпечення я отримав можливість:

- 1 - оцінки вартості та часу на розробку програмного продукту;
- 2 - виключення витрат часу та можливих фінансових затрат (якщо говорити про комерціалізацію розробки) і затрат на нераціональні дії, можливе вимушене допрацювання, тривалого погодження тощо,
- 3- уникнення розбіжностей та невдоволення замовника та користувачів.

### **2.1 Підготовчий етап**

Враховуючи особливості проєкту запропонований мною порядок розробки може відрізнитися від загально прийнятого. Його загальний вигляд представлено на рисунку 1.5



Рисунок 2.1 – Послідовність здійснення розробки

Під час підготовки було вирішено всі питання організаційного характеру:

- надання замовником матеріалів та документації (створення технічного завдання, затвердження дизайну). Також на цьому етапі я встановив наскільки достатньо є тих першоджерел і які моменти (етапи) вони «закривають» наскільки достатні вихідні та які етапи закривають. Таким чином я визначив та встановив орієнтовний об'єм робіт;

- фінансова сторона питання та терміни виконання: на основі наявних матеріалів затверджується приблизна вартість, кінцевий

термін усього проєкту, так само як і термін і точна собівартість найближчого з етапів.

Після цього, я погодився із технічним завданням, і отримав усі необхідні матеріали для здійснення розробки.

## **2.2 Етапи та результати проєктування**

До етапів проєктування слід віднести:

1 – описання – є спільною роботою замовника, результатом якої є обговорення користі програмного продукту, вимог до його працездатності та зовнішній вигляд. Обговорення технічних та алгоритмічних рішень.

2 – архітектура, етап затвердження запропонованої мови програмування, бази даних, серверів та фреймворків.

3 - технічне завдання, яке було складене мною, як архітектором, після обговорень із замовником на базі описів та його відповідей на мої запитання, які виникли в процесі вивчення питання, та які було повторно обговорено, погоджено та передано замовнику для внесення відповідних правок та уточнень в технічне завдання.

4 - макети (можуть додаватись і до технічного завдання). Йде мова про макети варіантів інтерфейсу, схеми влаштування, діаграм структур баз даних, схем взаємодії компонент і подібне.

5 – контроль. На цьому етапі я максимально намагався усунути зауваження.

6 – затвердження. Тут замовником було перевірено та внесено необхідні уточнення та зміни у технічному завданні самостійно. Після цього про всі зміни було оголошено мені, і після остаточного обговорення та погодження зауваження було усунуто, технічне завдання було затверджене і набуло вигляду остаточного до виконання документу.

В результаті проектування я отримав технічне завдання яке є зрозумілим та однозначним як для мене як виконавця (оскільки я є і керівником проєкту, і програмістом, і тестувальником, і дизайнером) так і для замовника і можливих інших учасників всього процесу розробки). Таким чином, встановлено що робити (описано продукт, функціонал, користувачів), як робити (описано архітектуру), як перевіряти, що завдання виконане (метод тестування, критерії оцінки).

### **2.3 Вимоги до розробки програмного продукту**

В результаті я отримав мінімально достатнє технічне завдання, яке дасть мені змогу:

- в повній мірі та достатньо чітко (у вигляді інструкції, « без води» та можливого «різночитання»), структуровано описати програмний продукт (якого кінцевого вигляду він повинен набути, яким чином та з чим співпрацювати, а також яким вимогам він повинен відповідати) так само як і його процес розробки в плані інформаційного процесу, щоб у мене як архітектора не виникло жодних питань чи сумнівів щодо реалізації,

- виключити всі суперечності та суперечливі відомості,
- відповідати вимогам ДСТУ та не викликати питань в плані невідповідності положенням законодавства України , тому що всі документи робота будуть оприлюднені.

Таким чином технічне містить:

- 1 – узагальнені відомості про розробку (назву продукту, ким і для чого його буде використано);

- 2 - загальні вимоги до програмного продукту (а саме структура, функції, архітектура, опис зв'язку із підсистемами, види інтерфейсів усіх складових, а також готовий дизайн чи його концепцію для кожної ролі кожного користувача);



- 3 – детальний та ґрунтовний план всіх робіт (етапи, строки);
- 4 - порядок тестування і прийомки(вид та склад випробувань продукту як в цілому так і окремих його частин);
- 5 – орієнтовний перелік усіх дій які дадуть змогу запусити продукт;
- 6 - вимоги щодо документування процесу а також і результату розробки.

При складанні технічного завдання особливу увагу я приділив т описав:

1. деталі:

- користувачів програмного продукту: увага їх ролям, правам і функціям;
- опису алгоритмів розроблених для опрацювання даних;
- переліку відкритих та закритих протоколів;
- вимогам щодо безпеки даних по всіх етапам життєвого циклу;
- список компонент(платних,вільних), які можуть бути використані в розробці.

2. приклад:

- при виявленні аналогів, обов'язково послатися на них;
- при опису роботи продукту представлено опис сценаріїв користувацької взаємодії з ним, які є типовими;
- приклади вхідних даних і форматів даних взаємодії підсистем (таблиця, база, сторінки тощо);
- вихідні дані, їх приклад (види звітів і експортованих файлів).

3. продуктивність та надійність:

- вказівка на моменти підвищеного робочого навантаження продукту (можливо це буде день чи місяць чи просто максимальне навантаження);

- вимоги до продуктивності, зберігання;
- обґрунтування щодо вибору устаткування для запуску програмного забезпечення;
- вказання хостингу серверної частини.

### 3 КОНСТРУЮВАННЯ ПРОГРАМНОЇ СИСТЕМИ

Наступний етапом після проектування є конструювання програмного продукту, яке займає одне з основних місць серед етапів життєвого циклу програмного забезпечення (рисунок 3.1)

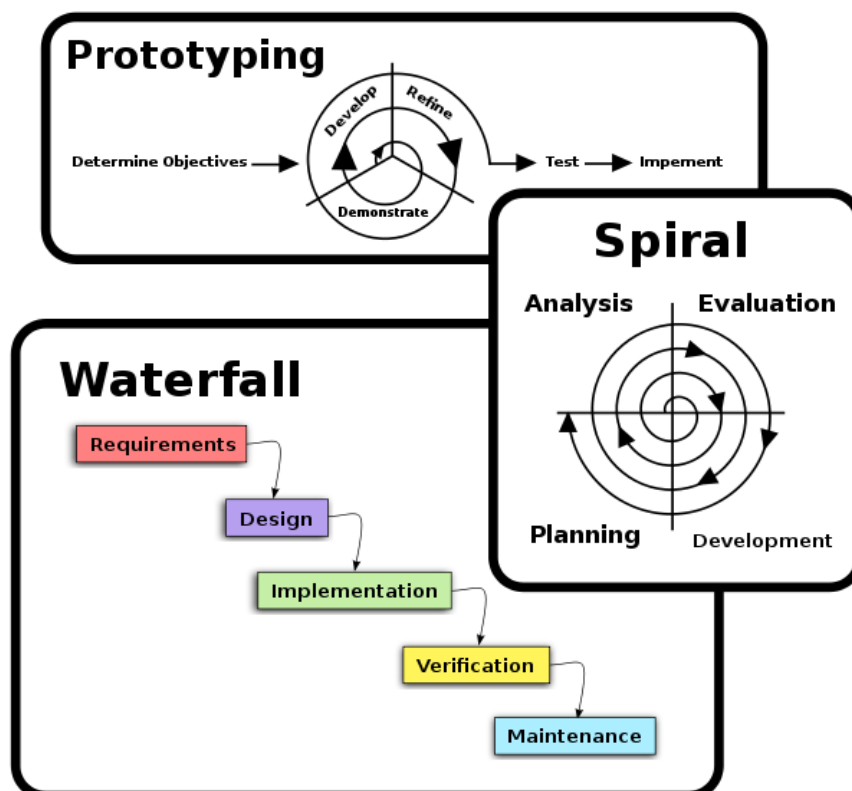


Рисунок 3.1 – Моделі життєвого циклу програмного забезпечення

#### 3.1 Вибір мови та середовища розробки

На цьому етапі було обрано та обгрунтовано мову програмування та середовище розробки. Базою шаблону запропоновано використати операційну систему Android. Інструментом для створення обрано високорівневу мову програмування Java. Далі я приведу пояснення, чому, що спонукало мене зробити такий вибір.



Рисунок 3.2 – Типовий вигляд робочого стола Android

По факту сьогодні відомо про декілька різновидів технологій , які дозволяють застосувати мобільні пристрої для навчання, сюди можна віднести окрім бездротових ноутбуків, портативних MP3-плеєрів, персональних цифрових помічників, електронних словників, так само як і мобільних телефонів всі ті пристрої, які містять третій же принцип роботи з використанням подібних технологій, яка сьогодні є об'єктом особливої уваги для тих науковців та вчених які займаються проблемами галузю освіти, зокрема питаннями та проблемами дистанційного навчання.

Одна із причин значного підвищення популярності мобільних пристроїв це звичайно сьогоднішня ситуація в світі, проблема ізоляції та інших неконтрольованих обмежень в умовах пандемії, це безумовно є питанням яке потребує негайного вирішення, адже освіта не може зупинятися і потрібно пристосовуватися до умов. Але крім цього фактом є й те, що використання мобільних пристроїв (в тому числі і телефонів) не потребує особливих додаткових, і я б сказав

навіть глобальних затрат. Йдеться про відсутність потреби в організації стаціонарного навчального місця не виникає потреби у створенні нового навчального приміщення, не потрібно закуповувати нове та сучасне обладнання, відсутня потреба фінансування окремого спеціаліста який би забезпечував успішне і максимально ефективно впровадження в навчальний процес такої нової компоненти. Крім того, мобільні пристрої (зокрема телефон) є значно дешевшими, якщо для порівняння з їх вартість взяти ціну ноутбуків, хоч володіють фактично тими ж функціями, що й персональні комп'ютери. Однак певні відмінності та особливості використання з такою метою все ж є, і про них чуть нижче.

Варта зазначити, що мобільні пристрої володіють багатьма можливостями комп'ютера: за допомогою них можуть демонструватися картинки, відео, електронні підручники, транслюватися звук. Їх також можна використовувати з метою розвитку усіх видів діяльності пов'язаної в мовленням, за допомогою них можна опановувати новий матеріал, вивчати, опановувати та повторювати в моделях, табличній формі, тренувальних вправах, використовувати інтерактивні матеріали, отримувати інформацію з електронних джерел глобальної мережі (використовуючи, для прикладу, браузер), проходити змістовні, модульні, підсумкові контролі, проводити атестаційні міроприємства, працювати з документацією, тобто повноцінно проводити освітній процес як зі сторони користувачів так і зі сторони організаторів (окремих людей та структур навчального закладу в цілому).

Крім вище зазначеного в більшості сучасних мобільних пристроїв присутня можливість використовувати електронну пошту, короткі повідомлення, які також додають додаткову можливість контролювати та корегувати проходження освітнього процесу.

Присутня можливість фотографувати з паперових носіїв та пересилати індивідуальні передбачені робочою програмою методичні завдання, а не переписувати їх, значно економлячи час.

Проте, звичайно є й сторони, які потрібно врахувати в процесі розробки шаблону мобільної веб-версії для системи дистанційного навчання. Про що саме йдеться:

- дороговизна мобільного зв'язку (потрібно врахувати що використовувати мобільний доступ до Інтернету це далеко не те саме що навчатись через персональний пристрій);

- розмір екрану, клавіатури, умови для навчання та подібне. Тут потрібно звернути увагу на вимоги до наступних технічних характеристик телефонів: їх розміри та маса, тип клавіатури, тип екрану, наявність аудіо функцій, об'єм пам'яті, пошукові потужності, швидкодія процесора, характерна моделі частота збоїв у роботі. Також не варта упускати момент, що користувачі часто не сприймають телефон як інструмент для навчання, можливе існування і психологічних бар'єрів як першочергових: якщо користувач не відчуває, того що мобільне середовище сприятиме навчанню, тоді це суттєво може вплинути на факт його застосування.

Оскільки базою розроблюваного шаблону запропоновано використати операційну систему Android, а інструментом для створення обрано високорівневу мову програмування Java, то далі в записці я обґрунтую чому.

Я спробував використати архітектурні шаблони model – view – controller (MVC), так і model – view – презентатор для розробки під Android. Мої висновки полягають у тому, що модель – вид – контролер працює нормально, але є декілька «issues». Все зводиться до того, як відтворювати клас Android Activity. Сприймати його як контролер, або це вид.

Фактичний клас Activity не розширює клас Android View, але він, однак, відображає відображення вікна користувача, а також обробляє події цього вікна (onCreate, onPause та інші).

Це означає, що коли я використовую шаблон MVC, то контролер фактично буде псевдо – контролером представлення. Оскільки за допомогою нього опрацьовується відображення вікна користувачеві з додатковими компонентами представлення, котрі я може додати до нього за допомогою setContentView, а також обробляє події, в крайньому разі для різних подій життєвого циклу активності.

В MVC контролер повинен бути головною точкою входу. Це трохи суперечливо, якщо це має місце при застосуванні його до розробки Android, оскільки діяльність є природною точкою входу більшості додатків.

Із-за цього я особисто вважаю, що шаблон типу model–view–presenter ідеально підходить для розробки Android. Оскільки роль представлення в цьому паттерні наступна:

- Служіння в якості точки входу
- Компоненти рендерингу
- Маршрутизація користувацьких подій до ведучого

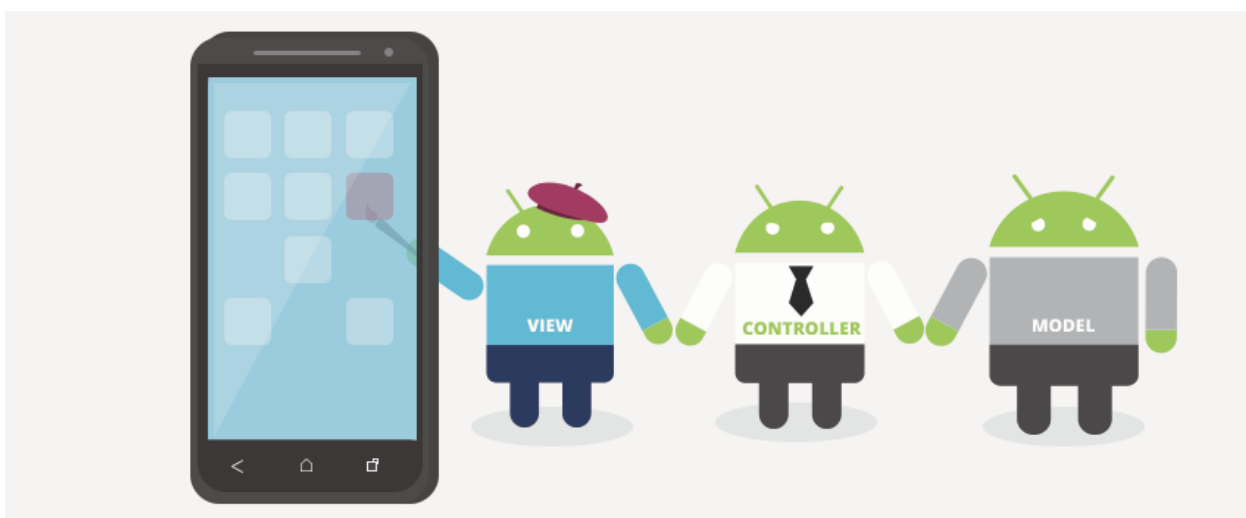


Рисунок 3 - Шаблон типу model–view–presenter

Це дозволить реалізувати мою модель наступним чином:

- Представлення - містить мої компоненти UI та обробляє події для них.

- Presenter - буде обробляти зв'язок між моєю моделлю та моїм представленням, «дивитися» на нього як на шлюз до моєї моделі. Це означає, що оскільки в мене складна модель предметної області, що представляє багато чого з чим я не стикався раніше, а моєму представленню потрібна тільки дуже невелика підмножина цієї моделі, моє завдання полягає лише в тому, щоби дати запит на модель, а потім оновити представлення. Наприклад, якщо у мене була б модель, що містить абзац тексту, заголовок та кількість слів. Але в даному доставлені мені потрібно відобразити лише заголовок в представленні. Після чого буде прочитано необхідні дані з моделі та відповідним чином оновлене представлення.

В основному модель - це моя повна доменна модель, яку потрібно зробити більш "tight", оскільки мені не знадобляться спеціальні методи для роботи з випадками, згаданими вище.

Завдяки виділенню моделі від представлення всіх разом (за допомогою презентатора) також стає досить більш інтуїтивно зрозумілим і тестування моєї моделі. Тут можливо варта використати модульні тести для моєї моделі домену та модульні тести для моїх докладчиків.

Таким чином власне я вважаю, що він відмінно підходить для розробки Android.



## 3 ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

### 3.1 ПЛАН ТЕСТУВАННЯ

Як відомо, сьогодні, при якісному тестуванні програмного забезпечення, витратна частина становить чи не найбільший відсоток, оскільки тестування є одним із найдорожчих етапів по всьому життєвому циклу продукту ІТ галузі. В цілому, якщо висловитись чисельно, то на цей етап зазвичай припадає 50-70% від суми повної вартості розробки. В сфері кодування програмних продуктів широке поширення отримали найрізноманітніші так звані CASE-засоби, за допомогою яких можна прискорити етапи та сам процес по створенню коду. Однак, в напрямку тестування суттєво відчутно нестачу таких засобів і здебільшого найбільше резервів та зусилля затрачається саме на так зване «ручне тестування».

Як правило, при здійсненні процесу тестування застосовують різноманітні методи. В цілому їх можна об'єднати в два основних - структурне («метод білої скриньки») та функціональне («метод чорної скриньки») тестування (рис.2.2). Оскільки тестування розроблюваного продукту буде проводитись одним із вище зазначених методів, варта їх порівняти (рис.2.1), з метою прийняття рішення про вибір одного із них .



Рисунок 2.1 – Порівняння методів тестування за методом «чорної скриньки» та «білої скриньки»

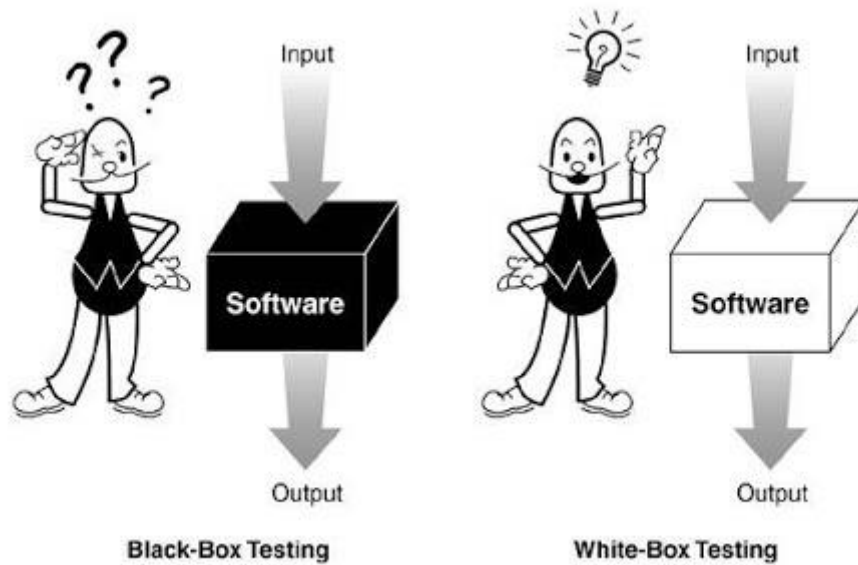


Рисунок 2.2 – Представлення тестування за методом «чорної скриньки» та «білої скриньки»

Для функціонального тестування характерна недоступність вихідного кода програми. Суть в тому, що здійснюється перевірка на відповідність поведінки безпосередньо програми до її зовнішньої специфікації. За критерій повноти тестування береться критерій повноти перебору вхідних даних на предмет усіх можливих їх значень. Здійснювати це практично є вкрай складною задачею.

Якщо ж говорити про структурне тестування, то для цього методу характерна відкритість тексту програми з метою її аналізу. Суть методу структурного тестування наступна: перевіряється внутрішня логіка програмного продукту. Можна говорити про повне тестування таким методом в разі якщо буде здійснено перебір на графі передачі програмного керування для кожного із усіх можливих шляхів. Їх кількість (чисельне значення) може сягати десятки тисяч. Крім такої складності, ще й повстає питання щодо можливості створити тести, які зможуть забезпечити данне покриття.

В цілому варта зазначити, що достатньо складною роботою є здійснення повного всеохоплюючого тестування навіть якщо говорити про відносно просту програму. Часом це може бути і неможливою задачею, якщо є обмеження по часу та ресурсам. Таким чином стає зрозуміло, що потрібно володіти певними критеріями, за допомогою яких можна визначитися та обрати так звані контрольні приклади, а також критерії за якими можна визначити момент припинення(завершення) тестувального процесу.

Якщо говорити про завдання функціонального тестування як такого, то його можна означити як здійснення перевірки на відповідність програми до своїх специфікацій. Для такого підходу, коли немає доступу до тексту програми вона (програма)розглядається як «чорна скринька». До найпоширеніших видів функціонального тестування відносять випадкове тестування, еквівалентна розбивка та аналіз по граничним умовам.

При випадковому (стохастичному) тестуванні мова йде про створення необхідної кількості незалежних тестів, для котрих генерування вхідних даних відбувається ран домно (випадково). У кожного метод є свої переваги та недоліки. До одного з недоліків цього можна віднести загальну кількість тестів, котрі потрібно згенерувати дотримуючись принципів та вимоги по надійності. Крім цього необхідно також забезпечити і момент незалежності цих тестів. Для прикладу при забезпеченні надійності програмного продукту при ймовірності відмови, яка б не перевищувала  $10^{-5}$  і з похибкою, яка б не перевищувала 5%, необхідно згенерувати 299 572 тести. Зрозуміло, що така кількість тестів вражає, і тягне за собою низку проблем, тому для того, щоб скоротити кількість необхідних тестів запропонували розглянути можливість розбити безліч вихідних даних на класи які є еквівалентними.

Тестування за еквівалентними класами передбачає можливість розбиття множини значень вхідних даних на підмножини число яких є кінцевим (вони називатимуться «класами еквівалентності»). Це робиться таким чином, щоби кожен тест, який представляє певний клас, був би еквівалентним для будь-якого іншого теста з цього ж класа. Два тести вважаються еквівалентними у випадку коли ними виявлено ті ж самі помилки. Проектування тестів за використання методу класів еквівалентності зазвичай проводять в два етапи: виділення класів еквівалентності (за специфікацією) та побудова множини тестів. Перший етап – в результаті здійснено вибору з специфікації для кожної з вхідних умов та «розбито» її на дві чи й більшу кількість груп, які повинні відповідати так званим «правильним класам еквівалентності» (ПКЕ) і «неправильним класам еквівалентності» (НКЕ). Таким чином утворюються множини елементами яких є допустимі і недопустимі для програми значення вхідних даних. Цей процес напряму може залежати від того як виглядає вхідна умова. Для прикладу, при описі вхідної умови множини ( $|x| \leq 0.5$ ), визначається один правильний ( $-0.5 \leq x \leq 0.5$ ) і два неправильних класи еквівалентності ( $x < -0.5$ ;  $x > 0.5$ ). На другому етапі МКЕ (метод класів еквівалентності) виділено класи, які використовують з метою побудувати тести. Для неправильних класів еквівалентності тести проектують так, щоби кожний тест покривав один і лише один неправильний клас еквівалентності, і так до тих пір аж поки всі вони не будуть покритими. МКЕ дає змогу досягнути суттєвого скорочення кількості тестів при порівнюванні із МВП (метод випадкового тестування). Одним із найсуттєвіших його недоліків є момент складності при виділенні класів еквівалентності, особливо для неправильних класів еквівалентності. Також до недоліків можна віднести ймовірність пропуску високоефективних

тестів певного типу (йдеться про тести, для яких характерна висока імовірність виявлення помилок). Для прикладу, мінімальні й максимальні допустимі значення для вхідних параметрів дозволяють знайти більшість помилок, які можуть бути пов'язані із певними відповідностями та переповненням типів даних. Для рішення цього питання було запропоновано МАГУ «метод аналізу граничних умов».

Метод аналізу граничних умов передбачає ситуації, які можуть виникати безпосередньо вище чи нижче, або ж на границі(межі) якоїсь певної вхідної або вихідної. Різниця між методом аналізу граничних умов та методу класів еквівалентності полягає в наступному:

- під вибором будь-якого з представників класу еквівалентності потрібно розуміти що його потрібно здійснювати так, щоб можна було здійснити перевірку за допомогою тесту кожену границю цього класу;
- в процесі побудови тестів розглядають не лише вхідні, а й вихідні умови (таким чином є деякі специфіковані обмежування на вхідні дані, зокрема їх значення).

Не слід нехтувати й загальними правилами для метода аналізу граничних умов. До них можна віднести побудову тестів для границь множини допустимого значення для вхідних даних і тести з використанням недопустимих значень, які можуть відповідати незначній кількості виходів за межу цієї множини. Для прикладу візьму тип даних INTEGER для бази даних SQLite з діапазоном значень  $[-9223372036854775808, -1, 0, 1, 9223372036854775807]$ . Тести будуються з гранично допустимими значеннями  $[-9223372036854775808, -1, 0, 1, 9223372036854775807]$ , та тести  $[-92233720368547759, -1, 0, 1, 92233720368547759]$ , значення яких можуть виходити за границі множини допустимих значень.

Варта також уваги і факт, що на практиці для того щоб локалізувати несправності створюються тести, які повинні бути в межах допустимого значення, йдеться про те, що вони є внутрішні для множин, а також тести, які можуть несуттєво відхилитись від граничних значень. Для дискретних множин можливих значень вхідних даних, будують тести як для мінімального так і для максимального значень для вхідних умов та для величин більшого чи меншого значення. Для прикладу, у випадку, коли вхідний файл містить від 1 до 255 запису, тоді обираються тести як для порожнього файлу так і для файлу, котрий включає 1, 254, 255 та 256 записів.

У випадку для впорядкованої множини вхідних і вихідних даних програми коли програми собою і послідовні файли і лінійні списки і подібне), під час проведення процедури тестування необхідно звернути увагу на перший та останній елемент такої множини.

Одним із найкорисніших методів проектування тестів є метод аналізу граничних умов. Однак досить часто його ефективність є недостатньою оскільки часом граничні умови є ледве вловимими, їх визначення є достатньо складним процесом.

При структурному тестуванні, або тестуванні типу «біла скринька», використовується аналіз вихідного коду програми. Відомо про три види тестування за принципом «біла скринька»: тестування, в основі якого лежить потік керування програми, на основі потоку даних та мутаційне тестування.

При застосуванні тестування потоку керування програми відбувається тестування логіки програми. Зазвичай програмна логіка представлена графом керування, в якому оператори – це вершини, а дуги – це переходи.

Тестування, в основі якого лежить потік даних найбільше уваги «приділяє» взаємозв'язку між змінними. При цьому виді тестування

виділяють вершини, в яких відбувається ініціалізація змінної Тут також увага приділяється вивченню переходів та взаємозв'язків між вершинами.

В основу ж мутаційного тестування покладено внесення несправностей безпосередньо в вихідний код програми та порівняння як між роботою вихідної і програми-мутанта.

Так як здійснення вичерпного структурного тестування являється вкрай нелегкою задачею, варта зосередитись на виборі таких критеріїв його повноти, якими б допускалась їх проста перевірка, та які б могли полегшити ціле направлене підбирання тестів.

Для деталізації і прийняття рішення не варта упускати і можливості класичних методів як структурного так і функціонального тестування. Ці методи також мають як переваги так і недоліки, а також і обмеження, зокрема, якщо говорити про функціональне тестування – то його характеристикою можна назвати велику кількість необхідних та потрібних тестів. Тут одразу ж потрібно розглянути запитання яке полягає в можливості забезпечити незалежність цих тестів. При обмеженні кількості тестів все одно виникнуть складності і під час виділення класів еквівалентності, так і під час виявлення границь. Важливо також зазначити ,що в цьому випадку можна пропустити цілий ряд тестів високої ефективності, при чому «шкідлива» логіка зазвичай не визначається. Також варта відзначити те, що функціональне тестування (його методи звичайно) не надають можливості локалізації несправностей.

Застосування функціонального тестування виправлане у випадку, коли програмний продукт вже створено, тобто йдеться про останні етапи життєвого циклу програмного забезпечення. Якщо під час функціонального тестування якість створення програмного

забезпечення піддається сумніву, то доведеться повертатись до попередніх етапів розробки. Як правило, це може призвести до незапланованих перевитрат, зокрема в плані фінансових та часових збитків.

За допомогою структурного тестування можна здійснити перевірку внутрішньої логіки програми. Це дозволить вчасно локалізувати фактично усі несправності. Однак тестування структурного типу все далі і далі складніше виконати якісно. Це пов'язано із збільшенням розміра вихідного коду програми. Таким чином його повноцінність ставиться під сумнів. До втрати по якості також може призвести і спуск по наданій ієрархічній структурі критерійних взаємозв'язків. Це може призвести до упущення помилок певного типу, що безсумнівів повпливає на якість.

Також варта зазначити про певних обмеженнях можливостей для застосування тестування структурного типу для різних фаз. Якщо ж при модульному тестуванні, із-за вихідного коду малого розміру, можливе застосування представлених методів, хоча і з тими чи іншими складнощами, частина з яких наведена вище, при інтеграційному та системному тестуванні прикладна область для представлених класичних методів є вкрай обмеженою. Це спричинено різким зростанням розміру вихідного коду або ж і взагалі його відсутністю.

Крім всього сказаного варта також зазначити і те, що наскільки є можливо застосувати структурне тестування залежить і від того, яку було обрано парадигму програмування. Зокрема для програмування за процедурним підходом можна застосувати методи структурного тестування; а вже при об'єктно-орієнтованому підході воно різко обмежується в силу збільшення розмірів вихідного коду; у випадку ж



підходу компонентно-базованого програмування – такі методи зазвичай взагалі вважаються незастосовними, в силу тієї ж причини.

Для компонентно-базованого програмування, компоненти в своїй більшості представлено у вигляді «чорних скриньок» і доступними є тільки їх автомати станів(на UML). Отже, для класичних методів структурного тестування, коли рівнем абстракції є рівень операторів, такі методи є не застосовними.

При тестуванні програмного забезпечення розробленого за принципами компонентно-базованого підходу основною задачею є перевірка якості на взаємодію поміж компонентами, тобто на їх інтеграційне тестування. Тут не йдеться про перевірку на правильність функціонування самих таких компонент, оскільки їх вже в більшості випадків було протестовано.

Отже, я прийшов до висновку, що для розроблюваного мною програмного продукту потрібні якісь специфічні та спеціалізовані критерії інтеграційного тестування, що працюватимуть на зовсім іншому рівні абстракції. Таким чином при врахуванні пропонованих автоматів станів компонент варта сконцентрувати увагу власне на між модульній взаємодії, а не зосереджуватись на особливостях їх внутрішньої роботи.

### **3.2 Розробка алгоритму тестування**

Для початку пропоную почати розробку тестів в основі яких лежить все таки структурне тестування на базі потоку управління, на реальному програмному кодї. Для тестування була обрана частина програми, яка відповідає за підключення до бази даних.

Тест №1. Нехай існує правильно написаний sql-запит до бази даних, Оголошений об'єкт класу SQLiteDatabase, але не створений

об'єкт класу `SQLiteOpenHelper` і `SQLiteDatabase` ніяк не ініціалізований (тобто немає підключення до бази даних). Тоді виконати скрипт можна викликавши метод класу `SQLiteDatabase`, але через відсутню ініціалізація виникне виняткова ситуація, а саме помилка зв'язана з `NullPointerException`. Крім того виникає проблема з відсутнім підключенням до БД і неможливістю у результаті виконати скрипт. Виникне виняткова ситуація `SQLiteException`.

Тест №2. Нехай оголошені об'єкти класів `SQLiteDatabase` і `SQLiteOpenHelper`. Як і в попередньому тесті `sql` запит оформлений правильно і доступ до методу `rawQuery()` в нього є. Але в результаті будуть створені об'єкти класів `NullPointerException` і `SQLiteException`, подібно попередньому тесту.

Тест №3. Нехай оголошені і ініціалізовані об'єкти і відкрито підключення до бази даних. `Sql` запит оформлений правильно і доступ до методу `rawQuery()` в нього є. В результаті тест буде успішним, запит до БД виконається успішно.

Тест №4. Нехай оголошені і ініціалізовані об'єкти і відкрито підключення до бази даних. `Sql` запит містить синтаксичну помилку. Доступ до методу `rawQuery()` в нього є. В результаті підключення до БД буде успішним, але через помилку в запиті виникне `SQLiteException` і зміни в базу даних не будуть внесені.

Виконання сформованих тестів можна зобразити за допомогою блок-схеми (рисунок 2.3). На ній події зображені за допомогою геометричних фігур. Прямокутник на блок схемі – це дії, які виконуються на даному етапі(наприклад створення об'єктів чи арифметичні операції). Ромб – це зображення умовного оператора (аналог `if` в мовах програмування). Від нього йде дві лінії, які характеризують виконання чи невиконання умови. Лінії тут

показують зв'язки між елементами блок схеми. Коло на блок схемах означає початок і закінчення алгоритму.

В програмному продукті підключення до бази даних у всіх класах успішне, виконується без помилок. Sql запити записані в програмний код і не можуть бути сформовані динамічно. Це в свою чергу вберігає від виняткових ситуацій і багів зв'язаних з помилками в запиті до бази даних.

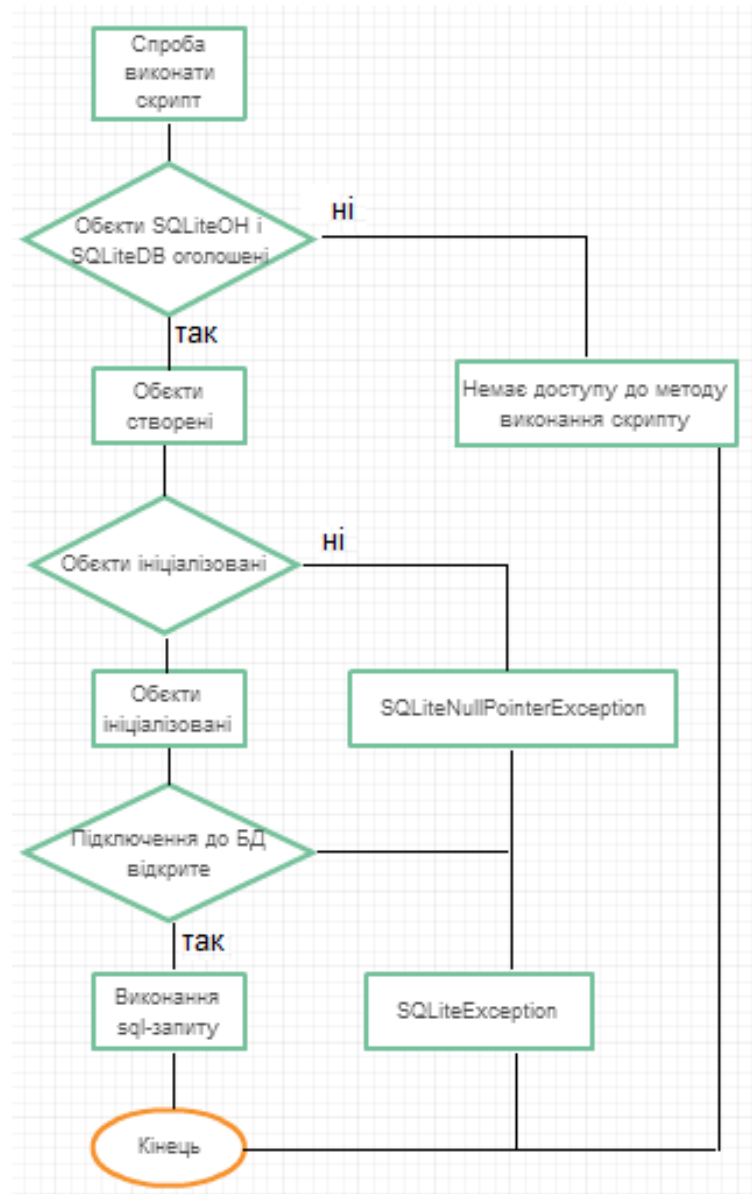


Рисунок 2.3 – Блок схема структурованого тестування

Тестування списків виконується за допомогою методів граничних умов. Наприклад маємо список функцій до яких має доступ користувач. Однак однією з особливостей розроблюваного шаблону є різний рівень доступу до певного функціоналу. Тепер видалимо функцію, до якої користувач не повинен мати доступу. Для цього виділяємо потрібний пункт списку і натискаємо кнопку «видалити». Після натискання спрацьовує метод, в якому в шаблон sql-запиту вставляються дані з пункту списку. База даних отримує запит, прирівнює передані дані до існуючих і видаляє ті записи з таблиці, поля якої відповідають переданим в запиті.

Як вчить досвід програмістів – потрібно писати частину функціоналу і тестувати його на роботоздатність. Це правильний підхід, хоча на перший погляд значно сповільнює процес написання коду. Але навпроти написання сотні стрічок коду і при запуску може виникнути багато помилок як синтаксичних так і логічних. Під час такого підходу відомо, що при введенні існуючих в БД логіну і паролю авторизація проходить успішно. А якщо ввести неправильно логін чи пароль, то поле паролю очиститься і виведеться спливаюче повідомлення про введення неправильних даних.

Розроблений програмний продукт успішно працює, дефектів і помилок на тестуванні не виявлено. Функціонал повністю відповідає побажанням замовника і технічному завданню.

## ВИСНОВОК

Як сказав один з відомих у одній із своїх праць «Життя в місті реального часу: мобільні телефони і міський метаболізм». І це не просто слова, це говорить й про те, що з появою вільного масового доступу до всесвітньої павутини повністю змінилося й життя людства. Мобільні телефони стали свого роду першопрохідцями в цьому процесі. Більше того, можна говорити про те, що вони так би мовити очолили вихід портативних засобів комунікації не просто на світовий ринок на лідерські позиції, вони стали інструментом на сьогодні та засобом як здобуття освіти, в тому числі і вищої, так і, без сумніву, спричинили за собою фундаментальні трансформації в процесах сприйняття окремих людей.

У даній кваліфікаційній роботі на здобуття освітнього ступеня «магістр» було чітко визначено як місце так і роль мобільних пристосувань, зокрема інструментів робота яких основана на нових технологій мобільних комунікацій в процесі сучасного освітнього процесу.

Таким чином, говорячи про сучасних Інтернет-користувачів мобільних пристроїв можна говоримо про те, що користувачі смартфонів і мережі Інтернет повинні бути мінімально освічені в цьому питанні, щоб мати можливість максимально ефективно використовувати можливості нових технологій, і крім цього при використанні додаткових можливостей, зокрема розпізнавання голосу, вони можуть навіть не використовувати письмо і читання, що є неоціненним в наш час, із за можливості не лишній раз торкатись та наражати себе на інфікування.

Робота складається із титульного аркуша, завдання на кваліфікаційну роботу, анотації українською та англійською мовами, змісту, вступу, основної частини (яка складається з аналітичної (аналітичний огляд в області досліджень) та теоретичної (виклад загальних підходів і основних методів досліджень, опис теоретичних, що включає основні методи проектування, розробки архітектури, підсистем і класів, обґрунтування платформ і засобів програмної реалізації) частин та стислого обґрунтування доцільності впровадження розробки), висновку, списку використаних джерел та додатків.

Отже, метою дослідження є виявлення, опис та розробка шаблону як способу, інструменту і варіанту використання в освітньому процесі гаджетів та якісна оцінка того, яку роль відіграє це використання з точки зору ефективності навчання на різних його етапах, ґрунтуючись на особистих оцінках а також інформації, отриманої в процесі спостережень. Для досягнення поставленої мети пройдено наступні етапи:

- сформульовано та погоджено з керівником тему кваліфікаційної роботи;
- розроблено та затверджено технічне завдання;
- проаналізовано технічне завдання, підібрано та проаналізовано бібліографічні матеріали необхідні для виконання кваліфікаційної роботи;
- проведено аналітичний огляд області досліджень;
- викладено загальні підходи та основні методи досліджень, основні методи проектування, розробки архітектури, підсистем і класів, обґрунтування платформ і засобів програмної реалізації;
- розроблено допоміжну документацію;

- проаналізовано роботу щодо питань з дотримання положень про охорону праці та безпеку в надзвичайних ситуаціях;

- зроблено відповідні висновки за результатами виконаної роботи.

Результати за темою кваліфікаційної роботи апробовано та представлено на VIII Міжнародній науково-технічній конференції «Інформаційні моделі, системи і технології», м. Тернопіль, 2020 р.