МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ САДІВНИЦТВА

Факультет економіки і підприємництва

Кафедра інформаційних технологій

Допущено до захисту
"_____" _____ 2023 р.
Зав. кафедри, к.т.н., доцент
_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ КОМПЛЕКСНОГО ЗАХИСТУ ІНФОРМАЦІЇ НА ПІДПРИЄМСТВІ

Виконав – студент 2 курсу, групи 21м-кн
освітньої програми "Комп'ютерні науки"
спеціальності 122 Комп'ютерні науки
Завінський Павло Віталійович

Науковий керівник – к.е.н., доцент
Концеба Сергій Михайлович

Умань – 2023

ЗМІСТ

ВСТУП 4

РОЗДІЛ 1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Загальний огляд комплексної системи захисту інформації

1.2 Основні критерії інформаційної безпеки та їх вплив на КСЗІ

1.3 Підвиди критеріїв та їх рівні

1.4 Методи забезпечення інформаційної безпеки та їх види

1.5 Вплив КСЗІ на загальну діяльність підприємства

РОЗДІЛ 2. МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Моделювання системи

2.2 Моделювання варіантів використання

2.3 Проектування моделі бази даних

2.4 Розробка дизайну

2.5 Вибір інструментарію

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Інформаційна технологія

3.2 Реалізація БД

3.3 Розробка додатку

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

ВСТУП

Інформація завжди була, є та буде неймовірно цінним ресурсом. В наш час інформаційні системи дуже щільно інтегруються у всі сфери людського життя, починаючи від закладів громадського харчування та закінчуючи великими компаніями зі світовими іменами, та навіть зараз не є можливим сказати що інформатизація досягла свого піку, вона швидше знаходиться на стадії розвитку.

Практично всі підприємства та державні структури використовують електронні системи для зберігання та передачі інформації. На даний момент схема роботи, яка була актуальна до цифрової ери, а саме робота виключно з фізичними носіями даних, є абсолютно нераціональною, а підприємства які її використовують, дуже швидко стають неконкурентоздатними. Навіть беручи до уваги всі переваги від використання інформаційних систем (пришвидшення роботи завдяки можливості швидкого обміну інформацією, відсутність необхідності в значній кількості фізичних носіїв та архівів для їх зберігання, можливість оптимізації роботи на основі даних якими оперує система і т. д.), не можна забувати про негативну сторону питання, а саме, електронний простір не є абсолютно безпечним. При роботі з ним також є певні проблеми, починаючи від технічних неполадок, що можуть привести до затримки в роботі підприємства, та закінчуючи зловмисниками, адже, як відомо, інформація в наш час вартує дорого, а там де є щось цінне, завжди є і ті, хто хоче отримати вигоду з нього незаконним шляхом.

Зловмисники є одночасно і фактором, який сприяє нестабільності при роботі з електронним простором, оскільки їх мотиви не завжди очевидні. Це може бути особа яка хоче отримати матеріальну вигоду шляхом продажу отриманої інформації, може бути колишній працівник, якого на його думку несправедливо звільнили та він хоче помститися, або просто хакер, який хоче довести самому собі, що він може обійти систему захисту. В той самий час протидія зловмисникам є доволі лінійною, якщо система безпеки налаштована

відповідно до всіх стандартів та вимог. Система захисту повинна бути універсальною, – їй байдуже яку ціль переслідує зловмисник.

Для забезпечення інформаційної безпеки підприємства необхідно використовувати комплексну систему захисту інформації. Така система – це сукупність всіх методів та засобів що використовуються для підвищення рівня захисту інформації. Те що називають комплексною системою захисту інформації (КСЗІ) насправді є поєднанням всіх засобів, які певним чином впливають на інформаційну безпеку підприємства. Важливо розуміти, що КСЗІ це не електронна система в звичному для всіх розумінні. Це є дещо більше. Це поєднання всіх чинників, які протидіють усім факторам, які можуть чинити загрозу для безпеки підприємства та додаткових засобів, які забезпечують працездатність, надійність тощо. Дана система не є тільки електронною, вона охоплює собою на однаковому рівні як антивірусні програми, так само і оф-лайн заходи для підвищення рівня інформаційної освіти серед персоналу. Той факт що вона працює не лише на інформаційному рівні, та організовує достатній рівень захисту для інформації компанії за допомогою технологічних засобів і програм, а також в цей момент увага уділяється також і фізичному рівню, робить дану систему захисту комплексною, оскільки у вас може бути відділена мережа, першокласні фахівці з інформаційної безпеки, дорого вартісні між мережеві екрани, інноваційне програмне забезпечення, але який в цьому сенс, якщо вашу серверну поставили біля кухні і на неї систематично проливають каву співробітники, або до офісу є вільний доступ не тільки у співробітників, а також і у кур’єрів зі служб доставки їжі, чи просто випадкових перехожих. На кожному комп’ютері в системі може стояти індивідуальний пароль, який гарантує що доступ до інформації на даному комп’ютері отримує лише ідентифікована особа, але в той самий час на стікері який приклеїли до рамки монітору написано цей самий пароль, що зводить на нуль його сенс.

Отже, саме комплексна система захисту інформації є неймовірно важливою складовою будь-якого підприємства, оскільки в наш час стрімкого розвитку технологій, вона дозволяє забезпечити безпечну та стабільну роботу

основних процесів підприємства, та захистити його як від внутрішніх, так і від зовнішніх загроз, при цьому беручи до уваги його специфіку та особливості.

РОЗДІЛ 1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1. Загальний огляд комплексної системи захисту інформації

Комплексна система захисту інформації(в подальшому КСЗІ) – це набір заходів фізичної та інформаційної безпеки, основною метою яких є мінімізація шансу несанкціонованого витоку даних за межі організації, а також забезпечення стабільної роботи основних систем підприємства, реалізація резервних схем роботи на випадок непередбачуваних обставин та підвищення рівню обізнаності співробітників щодо небезпеки нехтуванням захистом інформації.

Важливо розуміти, що незалежно від того, наскільки система є дорогою, продуманою або розробленою спеціалістами зі світовими іменами, вона не є ідеальною, та не зможе забезпечити 100 відсотковий захист, системи такого роду призначені для того, щоб мінімізувати можливий рівень шкоди, а також підняти вартість доступу до інформації на такий рівень, коли зловмисник витрачає більше ніж отримує. Тобто, для кращого розуміння наводимо наступний приклад. Існує невелика компанія, її клієнтська база зберігається в Excel файлі на старенькому робочому ноутбуці. Ми розуміємо, що ця база представляє певну цінність, особливо для таких самих маленьких компаній, які працюють в цій самій сфері, але ця цінність не є значною. Тепер розглянемо два варіанти розвитку подій. Перший, якщо в цій компанії реалізована елементарна система захисту інформації, та другий, якщо нею знехтували. Ми не розглядаємо варіанти при яких в компанії реалізована гарна система захисту, є відділ який нею керує та слідкує, оскільки для маленької компанії це занадто нераціональні витрати, система захисту яку ми розглядаємо включає в себе мінімальну роботу з персоналом (тобто працівникам пояснили що не можна зберігати собі на флешки цю базу, для роботи з нею вдома), перенесення бази в хмарне сховище з доступом через пароль та реалізація системи корпоративних T-mail. Всі наведені засоби, не є дорого вартісними, по правді, це все є

можливим організувати навіть безкоштовно, і не потрібно багато часу на реалізацію. В загальному все упирається в розмір бази, але в середньому за один робочий день, можливо навіть не цілий, цілком реально реалізувати такі засоби. При настільки невеликих укладеннях (декілька годин робочого часу), компанія отримує елементарну систему захисту, яка піднімає вартість доступу до інформації. Важливо також пам'ятати, що вартість самої інформації залишається однаковою, змінюється лише вартість доступу до неї [4]. Таким чином, в порівнянні цих двох варіантів, якщо з'являється зловмисник, ціль якого отримати цю базу та передати її конкурентам, то у другому варіанті він зможе отримати цю базу без особливих проблем: співробітники носять її на флешках, деколи забувають ці флешки на робочому столі, або дають друзям, відправляють її зі своїх персональних ноутбуків через незахищені мережі (наприклад мережа кав'ярні, чи кафе). Таким чином, отримати цю базу не є задачею для зловмисника, важче буде лише обрати спосіб, як її отримати, оскільки варіантів значна кількість. В першому варіанті, коли є елементарна система захисту, насправді також зловмисник може отримати цю базу, але для цього йому доведеться провести більше ретельну та детальну роботу та, наприклад, застосувати свої навички з соціальної інженерії, щоб отримати пароль та доступ до корпоративного акаунту, далі підключитись через мережу підприємства і отримати цю базу. І в першому, і в другому варіанті, зловмисник отримує базу, але в першому варіанті, йому необхідно витрати набагато більше часу, сил та, можливо, фінансів. Ймовірніше за все ще до початку роботи він це зрозуміє і не буде навіть намагатись, оскільки втрачає більше ніж отримує [1].

Основними елементами КСЗІ є організаційні та інженерно-технічні заходи. Повертаючись до визначення КСЗІ є доцільним нагадати, що організаційні заходи забезпечують фізичну безпеку, а інженерно-технічні – інформаційну безпеку.

Організаційні заходи – заходи забезпечення інформаційної безпеки в межах певної організації, які описують концепт ІБ конкретно для специфіки та особливостей даної організації, а також включають в себе регламенти та

інструкції щодо дій направлених на забезпечення ІБ підприємства [18]. Це такі як:

- інструкції для працівників в залежності від їх посад та обов'язків, щодо роботи з інформаційними системами (у випадках, коли їх обов'язки потребують такої роботи);

- регламент адміністрування елементів системи (тобто, яким чином ідентифікуються користувачі, як зберігається, змінюється та видаляється інформація, як проводиться робота з носіями інформації та їх знищення в випадку необхідності);

- організація заходів спрямованих на підвищення рівня інформаційної грамотності персоналу та навчанню їх правилам інформаційної безпеки;

- інструкції на випадок надзвичайних ситуацій (несанкціонований доступ до системи, поломка технічної складової системи).

Інженерно-технічні заходи – технічні заходи та методики їх використання спрямовані на забезпечення ІБ підприємства. Рівень інженерно-технічних заходів, які будуть використовуватися на підприємстві напряму залежить від цінності інформації, яка буде захищатись, її типу, рівня критичності та вимог самого підприємства.

Інженерно-технічними засобами забезпечення інформаційної безпеки є:

- міжмережеві екрани;

- засоби шифрування даних;

- сегментування мережі;

- захищені канали для передачі даних;

- засоби захисту аудіо-інформації (засоби, які створюють технічний шум, та таким чином блокують роботу пристроїв для запису аудіо або його трансляції, зазвичай такі засоби встановлюються в окремих приміщеннях, які використовуються для обговорення стратегічних питань) [2].

1.2. Основні критерії інформаційної безпеки та їх вплив на КСЗІ

Основними критеріями для КСЗІ є ключові принципи інформаційної безпеки. Вони відомі як КЦД – конфіденційність, цілісність, доступність. Ці принципи є фундаментом в сфері захисту інформації, на них будується все, не залежно від масштабів організації, її специфіки, вимог і т. д.. Всюди, де заходить мова про захист інформації, невід'ємно слідують ці принципи. Конфіденційність, цілісність та доступність, одночасно є основними принципами інформаційної безпеки, критеріями оцінки КСЗІ та властивостями інформації. Розглянемо їх, виходячи з цих трьох позицій, для отримання розуміння того, як вони зв'язані між собою та чому вони є настільки важливими [6].

Конфіденційність – властивість інформації, яка дає можливість обмеження доступу до неї, тобто певна інформація є відома лише певному кругу осіб, та ніхто за межами цього кругу не має доступу до цієї інформації. Відповідно виходячи з позиції конфіденційності, як принципу інформаційної безпеки, ми можемо сказати, що для забезпечення інформаційної безпеки підприємства є необхідним, щоб деяка інформація, позначена як конфіденційна, мала обмежений доступ до неї. Тобто вона не є загальнодоступною, а лише зареєстровані користувачі з правом доступу, мають можливість її використовувати. І з точки зору критерію оцінки КСЗІ, даний критерій оцінює відповідність реалізації цього обмеження доступу до вимог підприємства та інформації доступ до якої обмежується, і вже виходячи з цього робиться висновок, наскільки раціонально та доцільно був реалізований даний принцип.

Доступність – властивість інформації, яка гарантує, що користувач отримає доступ до необхідної йому інформації в той момент, коли йому це буде необхідно. Тобто, якщо користувач має можливість отримати доступ до інформації (цей доступ не обмежується конфіденційністю або посадою низького рівня в організації), то він отримає цю інформацію. Виходячи з позиції доступності, як принципу інформаційної безпеки, є можливим сказати,

що, по-перше, даний принцип розповсюджується не тільки на працівників організації, а також на клієнтів у випадку, якщо є інформація, яку вони можуть отримати, наприклад, якщо організація займається виробництвом певної продукції та продажом її через інтернет-магазин. Тоді з одного боку доступність гарантує, що працівники компанії будуть отримувати доступ до ресурсів та даних, які потрібні їм для роботи, і неважливо якого роду ця робота: чи розробка нової лінійки продукції, чи запуск рекламної компанії, і в той самий час доступність гарантує те, що клієнти будуть мати можливість зробити замовлення в будь-який зручний для себе час та зі сторони організації з цим не буде проблем. Відповідно доступність, як критерій оцінювання КСЗІ, означає, що система є відмово стійкою на достатньому рівні, вона забезпечує стабільний доступ до всіх ресурсів, а також має резервні елементи у випадку виникнення проблем з поточними.

Проаналізуємо таку властивість як цілісність. Дана властивість інформації забезпечує, що інформація в системі є цілісною та не підлягала несанкціонованим змінам, у випадку, якщо зміни були санкціонованими, то вже елемент даних після змін вважається цілісним. Таким чином, ця властивість дозволяє користувачам бути впевненими, що інформація, з якою вони працюють, є справжньою та до неї не було додано несанкціонованих змін, які її зіпсували. З точки зору цілісності, як принципу інформаційної безпеки, все доволі очевидно. Цей принцип гарантує, що інформація, внесена в систему, при подальшому використ анні є саме тією інформацією, яка вносилась раніше, та вона не підлягала змінам від третіх осіб. Відповідно оцінювання КСЗІ за цим критерієм відбувається в залежності від реалізації засобів, які забезпечують цілісність інформації. Зазвичай це засоби обмеження або розмежування доступу.

Виходячи з попередньо проведеного аналізу інформації щодо конфіденційності, доступності і цілісності, вже є можливим зробити висновок, що ці принципи є тісно пов’язаними між собою. По-перше, оскільки вони забезпечують інформаційну безпеку, то відсутність реалізації одного з них, або

ж його реалізація на низькому рівні, ставить під загрозу всю систему. По-друге, багато засобів спрямовані на реалізацію декількох принципів водночас, наприклад сегментація мережі, одночасно дозволяє підвищити рівень конфіденційності, а також використовуватися для забезпечення цілісності та захисту інформації від несанкціонованих змін. По-третє, ці критерії мають свої підрівні, кожен з яких описує дві основні деталі [3]: наскільки буде реалізований захист та яким чином він буде реалізований. Також в них є додаткова властивість, це підрівні, які повинні бути наявними для реалізації поточного рівня. Більше детально ці підрівні буде проаналізовано в наступному підрозділі.

Таким чином ми маємо три основи на яких будується КСЗІ, а також ми знаємо що ці основи залежать одна від одної. Також варто пам’ятати, що хоч всі основні принципи є важливими та пов’язані між собою в рамках одного проекту, їх відносна важливість між собою буде відрізнятись в залежності від потреб підприємства та інформації, якою воно оперує. Наприклад, підприємство може оперувати інформацією, більшість з якої є конфіденційною. Таким чином, при побудові КСЗІ варто звернути на це увагу та приділити конфіденційності максимально можливу суттєву увагу та ресурси, але при цьому також реалізувати заходи за рештою критеріїв на достатньому рівні.

1.3. Підвиди критеріїв та їх рівні

У попередньому підрозділі була наведено аналіз інформації щодо основних принципів захисту інформації, а також була згадано їх підвиди та рівні. У цьому підрозділі вони будуть проаналізовані більше детально.

Щодо конфіденційності в попередньому розділі було надано загальне визначення, яке в загальному можна спростити: конфіденційність захищає від несанкціонованого ознайомлення з інформацією та її розповсюдження. Це визначення описує суть, але воно є занадто узагальненим оскільки ми розуміємо, що конфіденційність захищає інформацію від несанкціонованого

розголосу, але не розуміємо в яких саме ситуаціях, на яких рівнях, та яким чином це робиться. Тому розглянемо цей аспект більше поглиблено частково спираючись на результати дослідження [5].

Конфіденційність ділиться на 5 підтипів.

- Довірча – дана послуга дає користувачу можливість передавати інформацію від захищених об’єктів системи до інших користувачів, дана послуга має 4 рівні реалізації, ранжування яких відбувається випливаючи з глибини та детальності реалізації послуги.

- Адміністративна – дана послуга дає можливість керування інформацією від захищених об’єктів системи для її адміністратора, або спеціалізованого користувача з особливим рівнем доступу до неї. Ця послуга має 4 рівні реалізація, ранжування відбується випливаючи з тих самих факторів, що і в попередньому випадку.

- Повторне використання об’єктів – послуга, яка працює з повторним, або одночасним використання об’єкта системи, забезпечує те, що у випадку повторного використання об’єкта, цей об’єкт не буде містити інформації, яка залишилась після попереднього користувача або процесу. Ця послуга містить лише один рівень її реалізація, в якому прописані основні тезиси, що повинні бути реалізовані для відповідності цьому рівню, а саме, політика повторного використання об’єктів повинна бути однаковою для всіх об’єктів системи, перед наданням доступу до об’єкту новому користувача, спочатку відбувається повне анулювання прав доступу до цього об’єкта попереднього користувача, а також інформація внесена ним в об’єкт повинна стати недосяжною для нового користувача.

- Аналіз прихованих об’єктів – дана послуга реалізує контроль каналів передачі інформації, які не контролюються за допомогою інших послуг чи елементів КСЗІ. Дана послуга має 3 рівні, ранжування відбувається за діями, які відбуваються у відношенні до прихованих каналів, на першому рівні ці канали просто виявляються, тобто ми знаємо що вони є, вони є задокументовані, але нічого з ними не робиться, на другому рівні відбувається контроль цих каналів,

ми знаємо що вони є, вони задокументовані, а також вони контролюються, а їх використання реєструється та задокументується, та останній рівень, це ліквідація, тобто ми знаємо що є приховані канали і просто їх ліквідуємо, відповідно логічним є що цей рівень забезпечує найвищий захист, оскільки при його реалізації підмножина прихованих каналів ліквідується та не чинить ніякої загрози для системи.

- Конфіденційність при обміні – дана послуга працює з забезпеченням конфіденційності інформації при її імпорті або експорті через незахищене середовище. Ця послуга має 4 рівні, які ранжуються відповідно від мінімальної конфіденційності при обміні до абсолютної конфіденційності при обміні, на останніх двох рівнях конфіденційність при передачі інформації через незахищене середовище зберігається найкращим чином, зокрема через те, що на цьому рівні реалізується приймання запитів на обмін, не лише на основі атрибутів системи, а також атрибутів приймальника та відправника інформації, що дає можливість мінімізувати шанс втручання в процес третіх осіб.

Ситуація з цілісністю, така сама, як і з конфіденційністю. В попередньому розділі ми розглянули її загальну картину, а зараз виконаємо детальний аналіз.

Цілісність ділиться на 4 підтипи

- Довірча – дана послуга забезпечує можливість користувачів передавати інформацію від захищених об’єктів до інших користувачів, при цьому зберігаючи цю інформацію цілісною, тобто завдяки достатньому рівню реалізації цієї послуги, користувач, який отримує інформацію від іншого користувача системи, може бути впевнений, що це саме та інформація, яку йому надсилали і по дорозі до нього, вона не була змінена третіми особами. Ця послуга має 4 рівні реалізації, які ранжируються у відповідності до глибини захисту який реалізується.

- Адміністративна – дана послуга забезпечує можливість передачі інформації та керування потоками з даних для адміністратора системи, або користувача зі спеціальними правами в рамках цієї системи, ця послуга має 4

рівні реалізації, ранжирування яких відбувається випливаючи з рівнем захисту цілісності, який забезпечується послугою.

- Відновлення – послуга, яка забезпечує цілісність інформації, шляхом відновлення її стану до попереднього стану, процес відновлення відбувається через відміну операції, або сукупності операцій направлених на зміну об’єкта. Дана послуга має 2 рівні, це обмежене відновлення і повне відновлення. Відмінність полягає у тому, що при реалізації обмеженого відновлення, відміняється певна обрана множина операції по відношенню до об’єкту в певний проміжок часу, а в ситуації з повним відновленням, проводиться відміна всіх операцій, які були виконані з об’єктом в певний проміжок часу.

- Цілісність при обміні – ця послуга реалізує засоби забезпечення збереження цілісності інформації при передачі її через незахищене середовище. Ця послуга має 3 рівні реалізації, та по аналогії з конфіденційністю, ця послуга на останньому рівні реалізації організовує запити при атрибутах зі сторони відправника та отримувача інформації, що в свою чергу забезпечує максимальний рівень захисту, тобто навіть при передачі через незахищене середовище отримувач може бути впевнений, що інформація яку він отримав не підлягала несанкціонованим змінам по дорозі до нього, а прийшла в тому самому вигляді, в якому її надіслав відправник. Для більшого розуміння, можна навести приклад використання елементів шифрування, коли атрибутом виступає ключ шифрування, при відправці інформації, вона шифрується відправником за допомогою певного ключа, таким чином по незахищеному середовищу передається зашифрована інформація, яка немає ніякої цінності та можливості до змін та лише коли вона доходить до одержувача, який також має ключ, за допомогою якого дешифрує її вона повертає свою цінність та значимість. Цей приклад також є справедливим для конфіденційності при обміні, оскільки засоби шифрування по перше не дають третім особам не маючим атрибута(ключа) прочитати інформацію, а по друге, зміни внесені в зашифровану інформацію, будуть відразу видні після дешифрації, таким чином реалізується і конфіденційність і цілісність інформації.

Та останній принцип, доступність, він ділиться на 4 підтипи, кожен з яких забезпечує дотримання цього принципу в окремих умовах. Оскільки доступність відрізняється від конфіденційності та цілісності в основних своїх догмах, то і послуги якими вона реалізується відрізняються, як можна бачити з інформації про конфіденційність та цілісність наведеної вище, в цих принципів є спільні послуги, такі як довірча, адміністративна, та при обміні, це зумовлено тим, що як конфіденційність так і цілісність направлені на збереження інформації, конфіденційність на збережені доступу до неї лише для авторизованих користувачів, а цілісність на збережені її незмінною. Тоді як принцип доступності, більше направлений на сам процес доступу, а не на інформацію. Тобто можна сказати що конфіденційність та цілісність працюють з інформацією, а доступність з тим щоб надати конкретно доступ до неї.

- Використання ресурсів – ця послуга дозволяє користувачу використовувати ресурси системи, тобто дає доступ до них. Ця послуга визначає реалізацію доступу до ресурсів для окремого користувача чи групи користувачів, визначає пріоритетність доступу та множини об’єктів до яких цей доступ може бути наданий. Дана послуга має три рівні реалізації.

- Стійкість до відмов – ця послуга реалізує можливість роботи системи при відмові її компонентів, тобто реалізація різних рівнів даної послуги дає можливість системі працювати навіть коли її елемент вийшов з ладу. Дана послуга має три рівні реалізації, перший рівень виділяється тим, що він не напрямлений на роботу з усіма елементами системи, а лише з обраною підмножиною, яка окремо регламентується. Цей рівень забезпечує роботу системи при відмові компонента з регламентованої підмножини. При цьому ймовірніше за все зі зниженням якості роботи до моменту усунення неполадки. Інші два рівні працюють охоплюючи всю систему зі всіма її елементами. Відмінність між ними полягає в тому, що другий рівень, як і перший, дає можливість системі працювати, але при цьому знижує якість цієї роботи до усунення неполадки, тоді як третій рівень є найвищим та навіть при відмові елементу системи дає її можливість працювати без зниження якості роботи.

Практично при реалізації цього рівня лише адміністратор буде знати про поломку, а рядові користувачі, навіть не помітять, що в системі є проблема.

- Гаряча заміна – ця послуга дає можливість адміністратору системи проводити заміну її елементів при цьому не перериваючи її роботу. Важливо також зауважити, що необхідність заміни може виникати, як через проблеми з елементом системи, наприклад його поломка, так і задля модернізації елементів системи, незалежно від обставин необхідності заміни. Адміністратор або технічний спеціаліст, повинен мати можливість зробити заміну, і при цьому не переривати роботу системи та зробити заміну таким чином, щоб після неї не довелось заново встановлювати програмне забезпечення. Послуга має 3 рівні реалізації. Перший рівень реалізує лише можливість модернізації системи, тобто він дає можливість виконати оновлення системи не впливаючи на його роботу, але при поломках елементів, будуть виникати проблеми і заміну провести без зупинки системи не вийде. Другий та третій рівні включають в себе перший, тобто дають можливість модифікації системи, а також дають можливість заміни проблемних елементів, різниця між ними в тому, що другий рівень не розрахований на всю систему, при його реалізації регламентується підмножина елементів, які можуть бути замінені без виникнення проблем в роботі системі, а третій рівень покриває всю систему і при його використанні є можливим замінити практично будь-який елемент не перериваючи роботу системи.

- Відновлення після збоїв – ця послуга дає можливість системі повернутись в робочий стан після збоїв. Послуга має три рівні: це ручне відновлення, автоматизоване відновлення та вибіркове відновлення. При реалізації першого рівня, в випадку, якщо в системі трапляється збій, який виводить її з ладу, адміністратор або технічний працівник, власними силами шукає проблему, її наслідки та усуває їх. Це рішення має право на існування завдяки своїй низькій складності та вартості, але при цьому воно зазвичай займає багато часу. Другий рівень реалізації спрямований на те, що система сама проводить самоаналіз після збою, визначає проблему та усуває її. Це рішення є доволі

швидким та надійним, але воно потребує великого обсягу попередньої роботи, зокрема з налаштування системи, визначення місць в ній з найбільшим шансом враження та підготування резервних елементів, які є підключені в систему в пасивному стані. Таким чином після виконання всіх цих дій у випадку збою система проводить самоаналіз, визначає, що проблема була в конкретному елементі, бачить що в неї є резервний елемент, який може покрити функціонал елементу, який був вражений та проводить відключення враженого елемента і підключає резервний. Також при автоматизованому відновленні адміністратор отримує звіт від системи, в якому вказано який елемент створив проблему, та як ця проблема була вирішена системою, і тоді адміністратор має достатньо часу щоб замінити чи виправити проблемний елемент, після чого переключити систему назад на нього. Для більшого розуміння можна навести приклад з генераторами електропостачання в будинках критичної інфраструктури. зазвичай вони підключені до мережі, але перебувають в пасивному стані до моменту коли стають необхідними. Якщо відбувається відключення електроенергії, то система електропостачання в будівлях критичної інфраструктури відключається від мережі, оскільки з неї не поступає енергія, та переключається на автономні генератори і працює на них до того моменту, поки в мережі знову не з’явиться електроенергія. Останній рівень реалізації схожий з попереднім, але відмінність полягає в тому, що він не потребує великого рівня попередньої підготовки та використовує наявні ресурси системи для продовження її функціонування з погіршеними характеристиками та при можливості повертає її в нормальний стан.

Узагальнений список всіх принципів та послуг та їх рівнів наведений на діаграмі нижче(рис. 1.1). Даний рисунок наведений для узагальнення та візуалізації інформації наведеної в цьому підрозділі попередньо.



Рис. 1.1. Дерево критеріїв інформаційної безпеки та послуг, які забезпечують їх

1.4. Методи забезпечення інформаційної безпеки та їх види

Для забезпечення інформаційної безпеки підприємства при застосуванні комплексної системи захисту інформації використовують два типи засобів:

- організаційні;
- інженерно-технічні.

В даному підрозділі ми більш детально проаналізуємо ці типи, зокрема інженерно-технічні засоби.

Інженерно–технічні засоби використовують спеціалізоване програмне забезпечення та технології для того, щоб організувати інформаційну безпеку на підприємстві. Існує три основних види цих засобів:

- технічні;
- інженерні;

- криптографічні.

Якщо провести пошук в мережі щодо інженерно-технічних засобів захисту інформації, то є можливість знайти дуже багато різноманітних типів цих засобів. Складається відчуття що їх набагато більше ніж три, але насправді ці три види, які наведені вище, є основними. Решта різновидів тим чи іншим чином відноситься до одного з них. Часто буває так, що ці види не використовуються в чистому вигляді. Тобто існує певний засіб, який використовується на підприємстві, і його не можна віднести до одного з цих трьох видів. Це тому що він є змішаним, тобто він одночасно може бути віднесений до декількох типів. Також варто зауважити, що інженерно-технічні засоби більше відносяться до інформаційного захисту, тоді як організаційні до фізичного. Насправді в сфері інформаційної безпеки вже не існує чіткого розділення, а кордон між інформаційною та фізичною безпекою є дуже розмитий. Зокрема через багатофункціональність, яка стала одним з основних надбань розвитку технологій, і тоді, коли раніше компанії зосереджувались на тому щоб засіб робив одну задачу, але максимально якісно, і в наш час з розвитком технологій існує багато інструментів, які при додаванні в систему покривають відразу декілька напрямків і при цьому роблять це на достойному рівні, нічим не поступаючи інструментам, які чітко спеціалізуються на одному напрямі.

Розглянемо більш детально типи інженерно-технічних засобів, які наведені в рамках даної роботи.

Технічні засоби – зазвичай це програмне забезпечення, спрямоване на захист електронного простору від сторонніх факторів.

Інженерні засоби – обмежуючі конструкції, які ускладнюють доступ до критичних елементів системи, а також технічні засоби, які виконують задачу схожу з технічними засобами, але в фізичному просторі.

Криптографічні засоби – роблять інформацію нечитабельною для неавторизованих осіб, дають можливість вільно передавати інформацію, навіть в незахищеному середовищі, завдяки тому, що вона буде в зашифрованому

вигляді, та не матиме абсолютно ніякої цінності без ключа, що дозволить її розшифрувати.

Також важливим фактором є те, що відповідно до законодавства України, а саме, закону України «Про захист інформації в інформаційно-телекомунікаційних системах», на юридичному рівні не існує розділення на технічні, інженерні та криптографічні засоби, оскільки в рамках законодавства всі засоби узагальнено називаються засобами технічного захисту інформації, і включають в себе всю множину інструментів. Комплексні системи захисту інформації повинні бути задокументованими, відповідати міжнародним стандартам та періодично проходити аудити сертифікованими органами [13], [15].

Щодо криптографічного захисту, ситуація складається таким чином, що криптографічні засоби захисту інформації по суті своїй були технічними, оскільки зазвичай криптографія це програмні засоби, які дозволяють шифрувати та дешифрувати данні використовуючи наявні алгоритми шифрування, а також генерувати ключі для цього шифрування. Але з розвитком технологій шифрування, як засіб захисту, відділилось від решти програмних інструментів через свою багатогранність, функціональність та корисність, і стало сприйматись як окремий вид засобів захисту інформації. Однією з особливостей даного виду захисту є те, що він працює не залежно від входження в систему, тобто, технічні засоби забезпечують безпеку системи, захищають її від зовнішніх загроз, та контролюють внутрішні, інженерні засоби працюють таким самим чином, але орієнтуються на фізичний простір системи, тоді як криптографічні заходи захищають інформацію, навіть якщо вона знаходиться за межами системи, в вільному незахищеному інформаційному просторі. Це є однією з причин, чому криптографічні засоби вважаються окремим видом, а не підвидом технологічних. Як було сказано раніше, незалежно від того, як гарно влаштована система захисту, інформацію рано чи пізно доведеться передавати, і зазвичай цей процес буде проходити через незахищений інформаційний простір, де вона стане дуже вразливою, але в

цьому випадку на допомогу приходить криптографія, яка забезпечить більш високий рівень захисту цієї інформації.

Тепер оперуючи інформацією наведеною в даному підрозділі, ми можемо вивести три твердження, які з неї випливають. Вони корелюють з результатами [8].

- Технічні, інженерні та криптографічні засоби мають одну мету (забезпечення інформаційної безпеки), але кардинально різні способи її досягнення

- З юридичної точки зору всі засоби підлягають під категорію “засоби технічного захисту інформації”.

- З технічної точки зору різні види захисту покривають різні області системи безпеки, тому спеціалісту, який займається проектуванням, або побудовою системи такого роду, необхідно розуміти відмінності між ними, їх особливості, переваги та недоліки в конкретних ситуаціях та на конкретних підприємствах.

1.5. Вплив КСЗІ на загальну діяльність підприємства

В загальному випадку вплив комплексної системи захисту інформації на діяльність підприємства повинен бути мінімальним з точки зору ускладнення робочих процесів чи додавання нових задач для персоналу, який ніяк не пов'язаний з інформаційними системами. КСЗІ та її введення до експлуатації на підприємстві несе за собою дві цілі:

- забезпечити захист інформації;

- оптимізувати робочий процес в інформаційному просторі.

З точки зору життєвого циклу, КСЗІ, не вважаючи на її комплексність та нелінійність, сприймається як стандартний програмний продукт. Життєвий цикл КСЗІ має 5 етапів[26]. Це наступні:

- Аналіз – проводиться аналітика організації, оцінка ризиків, визначення потреб та слабких місць, в загальному можна сказати що на цьому етапі

аналізується підприємство та складаються вимоги, яким повинна відповідати КСЗІ таким чином, щоб покривати бізнес-процеси організації та забезпечувати достатній рівень захисту інформації [17].

- Проектування – виходячи з вимог визначених в ході аналізу, виконаного в попередньому етапі, розробляється проектний план, обираються конкретні засоби забезпечення інформаційної безпеки та розробляються технології їх впровадження.

- Розробка – безпосередня побудова КСЗІ встановлення програмного та технічного забезпечення, внесення необхідних інженерних рішень, а також розробка регламентів та інструкцій необхідних для організації захисту інформації.

- Тестування – проведення тестування готової системи, її робото здатності та дотримання вимогам встановленим в ході виконання першого етапу, зазвичай на цьому етапі з’являються нюанси, які не були враховані при проектуванні і з’являється необхідність вносити зміни в систему. В більш рідкісних випадках, можуть виникати критичні помилки, через невдале проектування, що призводить до необхідності повернення на один етап назад і повернення до розробки. У випадку з КСЗІ, шанс на такий варіант розвитку подій є мінімальним, окаільки система є комплексною, а її елементи зазвичай не впливають напряму на роботу одне одного, і шанс вчинити настільки критичну помилку при проектуванні, щоб некоректно працювала вся система є дуже малим.

- Впровадження та підтримка – останній етап. Систему впроваджують на підприємство та вона починає виконувати свої функції. Технічний спеціаліст або адміністратор системи займається тим, що слідкує за роботою системи, виправляє поломки у випадку їх утворення, вносить модифікації, та загалом підтримує систему у стані, в якому вона може адекватно виконувати свої функції [9].

Звернемо увагу, на вплив розробки системи на організацію, тобто, в залежності від масштабів організації та її ресурсного забезпечення є 3 варіанти розвитку подій

- Перший, організаціє є масштабною, має свій відділ технічних спеціалістів, а можливо і відділ з інформаційної безпеки, а також достатню кількість ресурсів для того, щоб виділити їх на розробку системи. В такому випадку саме працівники організації будуть займатись розробкою системи, її тестуванням, впровадженням та подальшим супроводом. Цей варіант є позитивним в плані якості виконаної роботи, якості комунікації виконавця і замовника, оскільки систему захисту інформації будуть розробляти люди, які вже працюють з інформаційною системою підприємства, знають в яких моментах зазвичай виникають проблеми та які загрози є найсуттєвішими. А з іншого боку, цей варіант, не зважаючи на якість продукту отриманого на виході, є дуже дорого вартісним та ситуативним. Тому цей варіант підходить для великих корпорацій або державних структур, які можуть довірити задачу своїм працівникам та надати фінансування на її виконання і бути впевненими, що робота буде виконана якісно в реаліях компанії.

- Другий варіант, це використання аутсорсингових компаній, або замовлення системи, у компанії, яка спеціалізується конкретно на роботі з КСЗІ. Таким чином компанія отримає систему розроблену спеціалістами, які мають певний досвід саме в сфері розробки комплексних систем захисту інформації. З мінусів даного варіанта є можливим виокремити два пункти. Це, по-перше, можливі складності в комунікації, – класична проблема при роботі двох сторін, коли отриманий в результаті продукт в загальному є тим, що було замовлено, але в певних нюансах вийшло не так, як хотілось, і другий момент, це реалізація останнього етапу життєвого циклу, а саме підтримки, оскільки навіть після завершення розробки і введення системи в експлуатацію, буде необхідність наймати спеціаліста, який буде слідкувати за системою та лагодити її в випадках поломки.

- Третій варіант є найбільш бюджетним та простим, він підходить маленьким компаніям, які тільки починають свій шлях, їх сфера роботи може бути ніяк не пов’язана з інформаційними технологіями, але в час активної інформатизації, справу з ними мати доведеться в будь-якому випадку. Цей варіант полягає в найманні технічного спеціаліста, який буде займатись інформаційними системами, наприклад вести базу даних з клієнтами, надавати консультації в випадку виникнення проблем в роботі з ПК, а також введе елементарні заходи захисту інформації. Звісно, ці заходи будуть набагато простіші ніж в попередніх варіантах, та системою назвати їх також не є можливим, але як було сказано на початку даної роботи, задача КСЗІ не звести шанс втрати інформації до 0, а мінімізувати цей шанс та підняти вартість доступу до інформації таким чином, щоб зловмиснику це було не вигідно.

Отже вплив КСЗІ на підприємство буде мати лише позитивні наслідки, навіть не залежно від того, в якому форматі з представлених вище, була побудована ця система. Логічним є те, що більш якісна система забезпечить вищий рівень інформаційної безпеки підприємства, але в той же час, навіть елементарний набір засобів, який не буде коштувати організації практично нічого, значно підвищить рівень захисту в порівнянні з повною його відсутністю.

Також вартий уваги той факт, що введення в експлуатацію системи окрім явних наслідків, приведених вище, також підвищуєть рівень інформатизації підприємства, що приводить до введення нових електронних систем, які пришвидшують роботу організації, а працівники, при цьому сприймають ці зміни більш просто, оскільки вони вже мали змогу працювати з подібного роду системою, і ще одна не сприймається для них, як щось незвідане.

РОЗДІЛ 2. МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Моделювання системи

Першим кроком в моделюванні додатку є визначення функціональних вимог до нього, тобто необхідно створити перелік функцій, які повинні бути реалізовані в додатку. Додаток представляє з себе інструмент для візуалізації зв’язків між об’єктами та суб’єктами інформаційної системи, тобто є можливим сказати, що цей додаток, це свого роду панель адміністратора для спеціаліста, який займається інформаційною безпекою на підприємстві, він додає на цю панель елементи системи, виходячи з наявної в нього інформації, встановлює зв’язки між ними та отримує на виході структуру підприємства. Також варто звернути увагу на механізм допуску до додатку, оскільки додаток зберігає інформацію про працівників підприємства, то вільний доступ до нього є неприйнятним, таким чином необхідно реалізувати механізм авторизації користувача. Отже функціональні вимоги до додатку виглядають таким чином.

- Авторизація користувача
- Перегляд списку працівників та списку девайсів
- Можливість додавати нові записи до списку працівників та списку девайсів
- Можливість встановлювати зв'язок між працівником та девайсом
- Збереження всіх записів в БД
- Можливість переглянути які девайси є прив’язані до конкретного користувача та змінювати прив’язки за необхідності
- Редагування та видалення записів з бази через додаток.

Побудуємо діаграму IDF0 (рис. 2.1), яка узагальнює вимоги, та представляє візуальну картину додатку.



Рис. 2.1. Діаграма IDF0 системи підтримки інформаційної безпеки

Таким чином, виходячи з діаграми, ми бачимо загальну схему роботи додатку, а саме, користувач вносить інформацію про підприємство, а за допомогою таблиць БД та зв’язків між ними, формується вигляд інформаційної структури підприємства, завдяки якій користувач має можливість приймати рішення щодо безпеки підприємства та засобів, які варто організувати.

2.2. Моделювання варіантів використання

В контексті даного проекту, робота з додатком не передбачає неавторизованих користувачів, таким чином, для діаграми варіантів використання (рис. 2.2) буде представлений лише один актор, який представляється авторизованим користувачем.

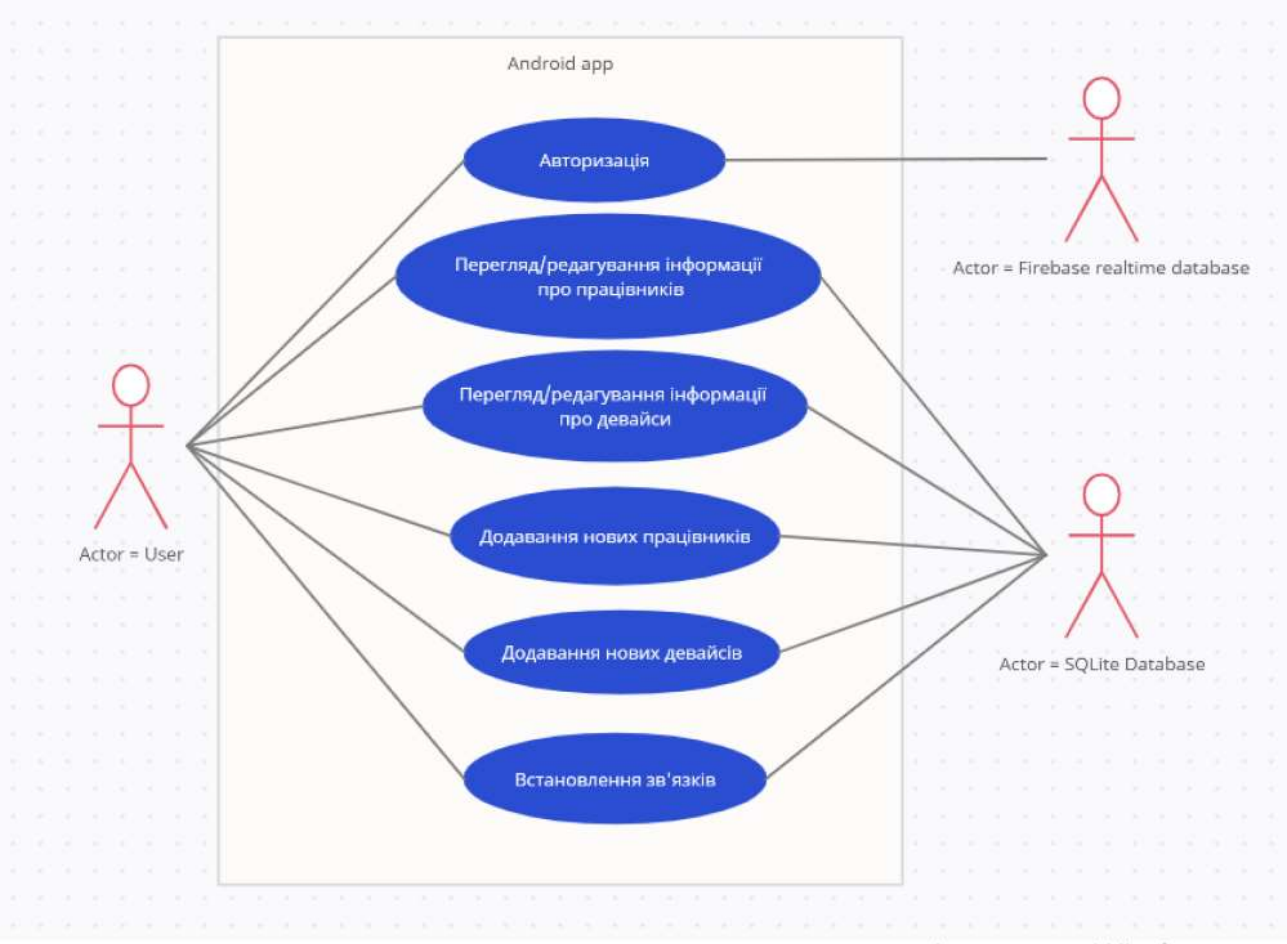


Рис. 2.2 Діаграма варіантів використання

Таким чином, на даний момент ми змоделювали внутрішні процеси додатку, а також варіанти його використання, одним з ключових моментів даного додатку є БД, яка зберігає всередині себе дані, а також дає користувачеві додавати в неї нові дані через інтерфейс додатку.

2.3. Проектування моделі бази даних

В контексті даного додатку, база даних складається з 4 таблиць:

- Workers – таблиця, яка зберігає інформацію про працівників організації (суб’єкти інформаційної безпеки)

- Devices – таблиця, яка зберігає інформацію про прилади для роботи з інформацією (об’єкти інформаційної безпеки)

- Link – таблиця, яка зберігає інформацію про встановлені зв’язки між таблицями Workers та Devices

- Users – таблиця, яка містить дані для авторизації.

Побудуємо ER діаграму (рис. 3.3) для представлення зв’язків між таблицями наведеними вище.

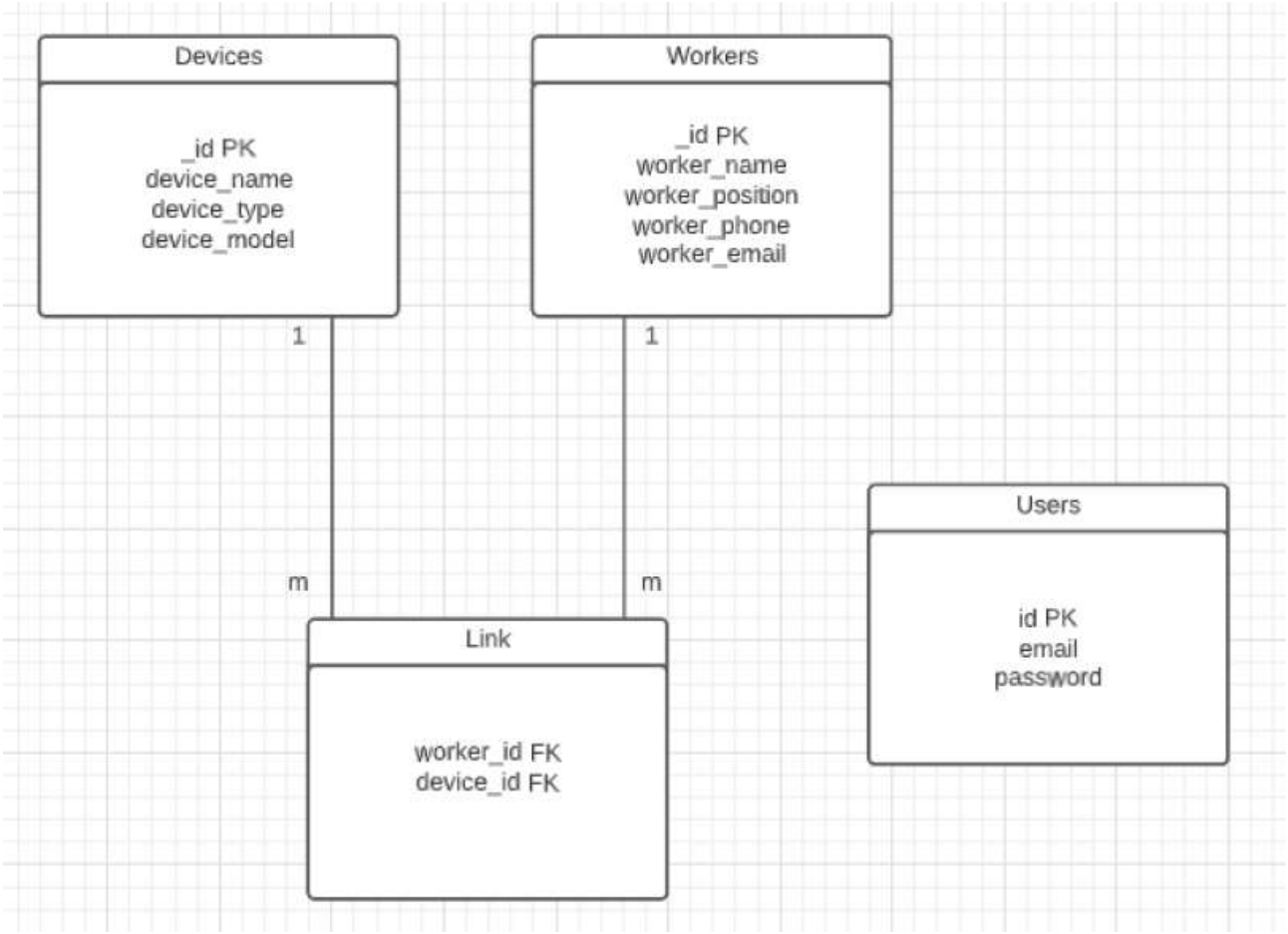


Рис. 2.3 ER діаграма

Розберемо детально кожну таблицю та її атрибути.

Таблиця Workers містить 5 полів, та представляє окремих працівників з їх персональними даними(таблиця 2.1).

Таблиця 2.1

Опис полів таблиці Workers

Назва	Тип	Опис
_id	Int	Ідентифікатор працівника, первинний ключ таблиці Workers
Worker name	Text	Ім’я працівника
Worker_position	Text	Посада працівника
Worker_phone	Text	Телефон працівника
Worker email	Text	Пошта працівника

Таблиця Devices містить 4 поля, та представляє девайси з їх основними характеристиками (таблиця 2.2).

Таблиця 2.2.

Опис полів таблиці Devices

Назва	Тип	Опис
_id	Int	Ідентифікатор девайса, первинний ключ в таблиці Devices.
Device name	Text	Назва девайсу
Device_type	Text	Тип девайсу(портативний чи стаціонарний)
Device model	Text	Модель девайсу

Таблиця Link містить два поля, які представлені зовнішніми ключами, вона виступає для зв’язку попередніх двох таблиць, та дає можливість поєднувати дані з них за ключами(таблиця 2.3).

Таблиця 2.3

Опис полів таблиці Link

Назва	Тип	Опис
Worker_id	Int	Ідентифікатор задачі, первинний ключ таблиці Tasks.
Device_id	Int	Ідентифікатор девайсу, зовнішній ключ, через який таблиця Devices пов’язується з таблицею Tasks.

Та останнім йде огляд таблиці Users, як видно на рисунку (рис. 2.4), ця таблиця не є пов’язаною з рештою, вона використовуються для авторизації конкретного користувача, і надає доступ до функціоналу додатку, також її реалізація відрізняється від решти, оскільки попередні таблиці реалізуються з використанням вбудованих в середовище розробки SQLite інструментів, а також окремого SQLite Browser, який використовується для створення запитів на створення та тестування таблиць. Авторизація в свою чергу реалізується за

допомогою Firebase, тобто дані з попередніх таблиць зберігаються напряму на девайсі з додатком, а таблиця Users для авторизації в хмарному сховищі.

Таблиця 2.4

Опис полів таблиці Users

Назва	Тип	Опис
id	Int	Ідентифікатор запису для авторизації, первинний ключ таблиці Users.
email	Text	Пошта користувача
password	Text	Пароль

Тепер ми маємо описаний функціонал, а також схему БД, яка необхідна для розробки додатку, таким чином виходячи з даних отриманих в цьому розділі є можливим приступити до розробки, залишається лише останній крок, це створення дизайну для додатку, маючи на руках його разом з структурою БД та функціоналом, залишиться лише вирішити технічні питання розробки.

2.4. Розробка дизайну

Першим кроком створимо схему, на яку нанесемо всі необхідні екрани додатку, які потребують розробки дизайну для них. Елементами цієї схеми стануть:

- Splash screen
- Екран авторизації
- Головний екран додатку
- Екран додавання працівника
- Екран девайсів
- Екран додавання девайсу
- Екран додавання девайса для працівника
- Екран редагування запису працівника

- Екран додавання запису девайса
- Екран перегляду інформації про працівника та його девайси
- Фрагмент для виводу статистики

В результаті ми маємо 11 екранів, для кожного з яких необхідно створити дизайн, а також розібрати взаємодію між цими екранами.

Для створення дизайну був використаний онлайн інструмент proto.io, який дозволяє розробляти прототипи для додатків різної направленості, та окрім створення дизайну, також дає можливість встановити зв’язки між екранами, та подивитись, яким чином вони будуть взаємодіяти між собою[22][23].

Першим екраном, який користувач бачить при запуску додатку є SplashScreen (рис. 2.5), на ньому наводиться назва додатку, логотип, фраза з описом та шкала завантаження, в контексті даного додатку ця шкала є лише візуальним ефектом, оскільки додаток не потребує часу на завантаження додаткових даних, чи зв’язку з мережевими ресурсами.

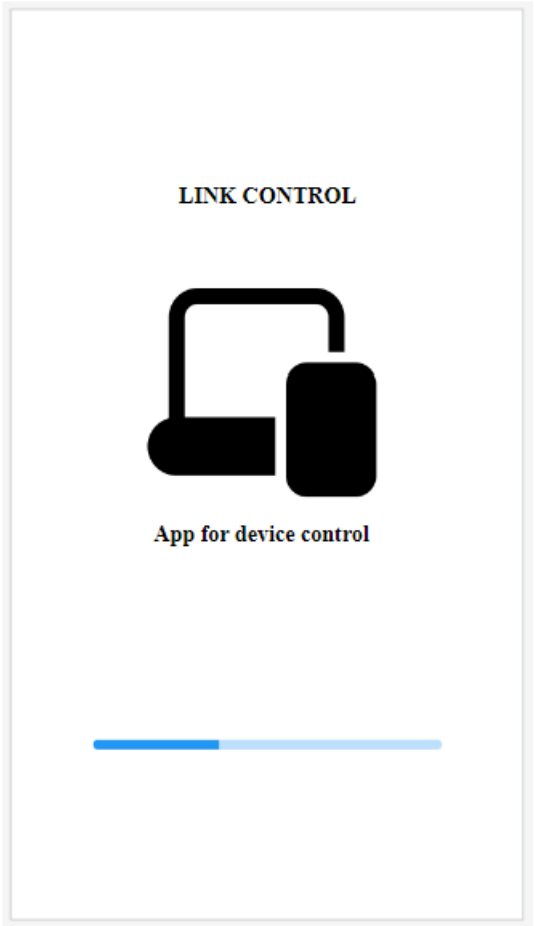


Рис. 2.5 Дизайн екрану SplashScreen

Наступним екраном, який бачить користувач э екран авторизації (рис. 2.6), на даному екрані він повинен ввести свої дані для авторизації, щоб отримати доступ до основного функціоналу додатку.

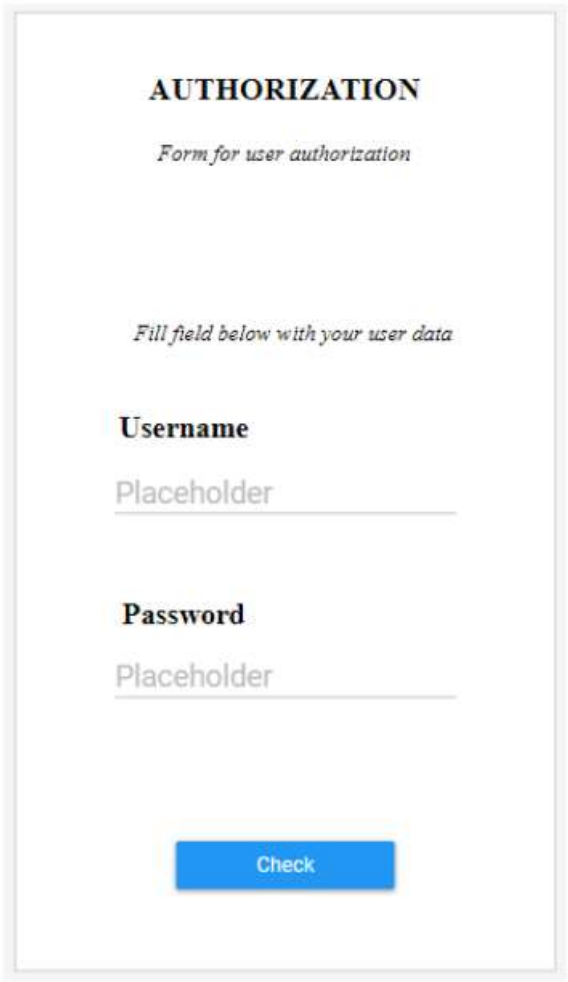


Рис. 2.6 Дизайн екрану авторизації

Після успішної авторизації в додатку, користувач потрапляє на основний екран (рис. 2.7), на цьому екрані наведений список всіх працівників та їх даних, які внесені в БД, також, окрім цього списку, користувач бачить фрагмент статистики в нижній частині екрану, даний фрагмент показує статистичні дані на даний момент, а саме, кількість працівників, кількість девайсів, а також кількість девайсів яким були назначені працівники. Також на цьому екрані є 4 інтерактивні елементи, це кнопки для редагування чи видалення інформації про працівника, ці кнопки прив’язані окремо до кожного запису, елемент для переходу на активність додавання нового працівника, у вигляді круглої кнопки в нижньому куті та елемент в для переходу до активності девайсів, у верхньому

куті, на рисунку дизайна він представлений іконкою макбука, в фінальній версії продукту дизайн іконки буде змінений.

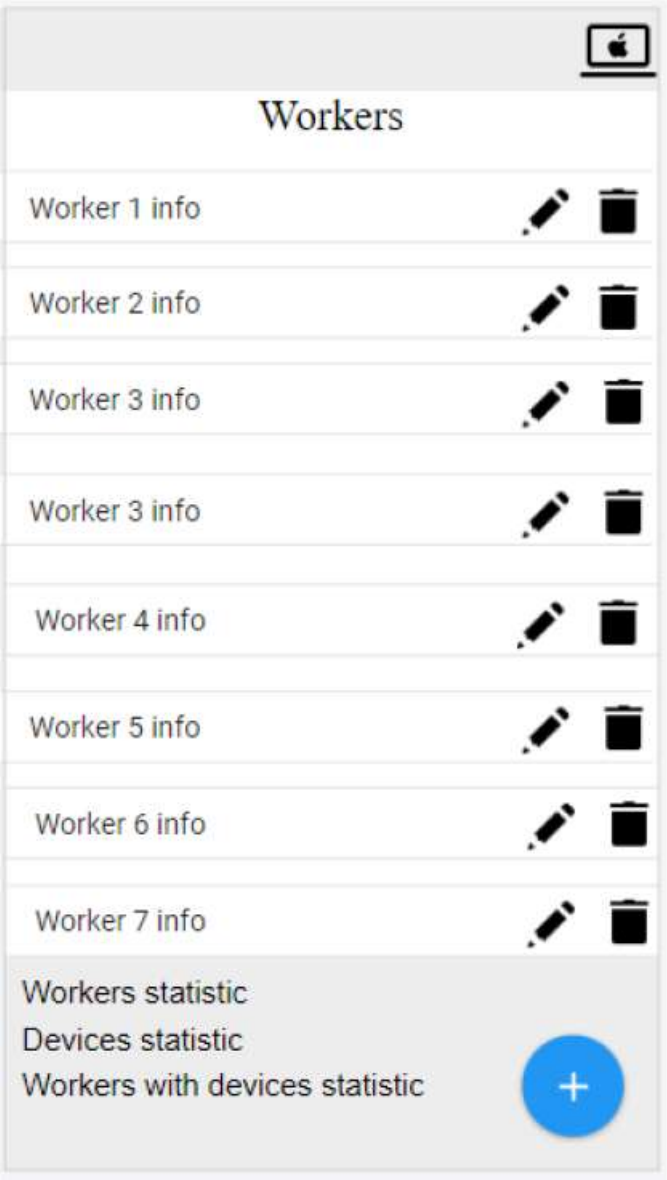


Рис.2.7 Дизайн головного екрану

При необхідності внесення змін в існуючий запис, користувач натискає на іконку олівця, і перед ним відкривається вікно внесення змін (рис. 2.8), в якому знаходяться поточні дані обраного запису, користувач вносить зміни та при натисканні на кнопку Upload зміни зберігаються в базу.

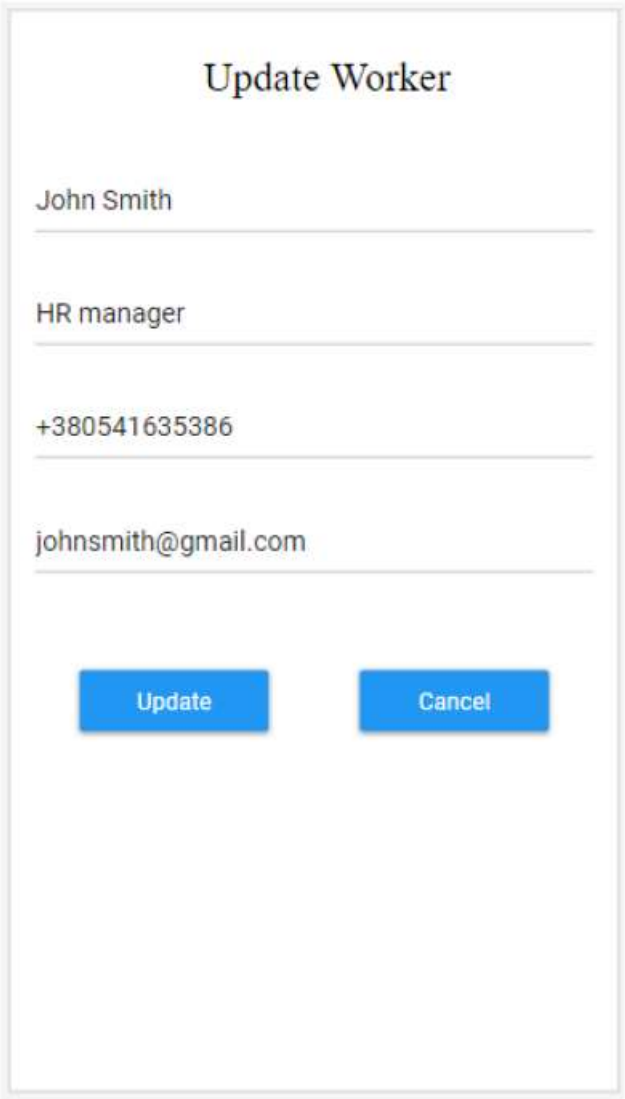


Рис. 2.8 Дизайн вікна редагування працівника

Якщо ж користувачу необхідно додати нового працівника до бази, він натискає на кнопку в нижньому куті та потрапляє до вікна створення нового працівника (рис. 2.9). В цьому вікні користувач бачить поля, які потрібно заповнити, а також заголовки, які показують, які саме дані потрібно ввести в конкретне поле.

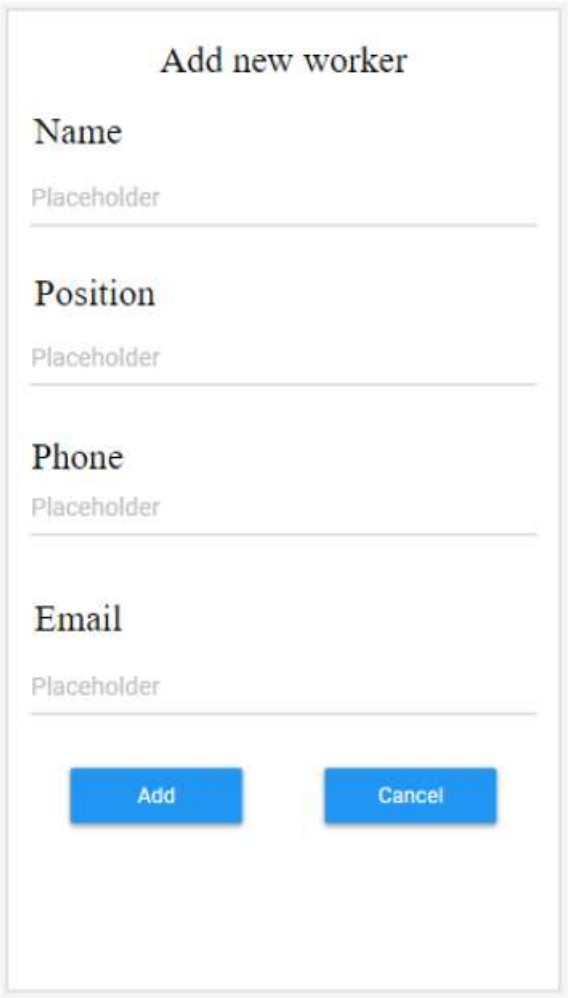


Рис. 2.9 Дизайн вікна додавання працівника

Активність видалення запису з бази не потребує освітлення, оскільки користувач натискає на відповідний елемент, додаток відкриває діалогове вікно з підтвердженням видалення, і якщо користувач підтверджує видалення, елемент видаляється.

При натисканні на елемент в верхньому куті, користувач переходить з головного екрану, на екран девайсів (рис 2.10), на цьому екрані наведений список існуючих девайсів, з кнопками редагування та видалення, статистичним фрагментом в нижній частині екрану, а також кнопкою додавання нового девайсу в нижньому правому куті, окрім цього в верхньому лівому куті наведена стрілка, яка дозволяє повернутись до головного екрану.



Рис. 2.10 Дизайн екрану девайсів

При потребі, користувач натискає на круглу кнопку в нижньому правому куті, та потрапляє до вікна створення нового девайсу (рис. 2.11), в цьому вікні наведені поля, які користувач має заповнити, та дві кнопки, одна з яких додає новий девайс до бази, а інша відміняє операцію створення та повертає користувача на екран девайсів.

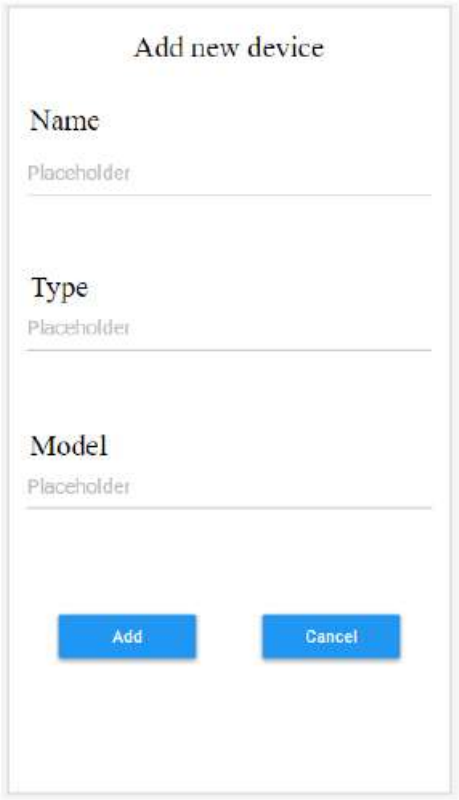


Рис. 2.11 Дизайн вікна додавання девайсу

Якщо користувачу необхідно відредагувати існуючий девайс, то він натискає на олівець, та потрапляє до вікна редагування інформації про девайс, функціонал такий самий, як з релагуванням даних працівника, користувач отримує вікно з введеними існуючими даними, вносить зміни та після натискання на кнопку ці зміни приймаються базою.

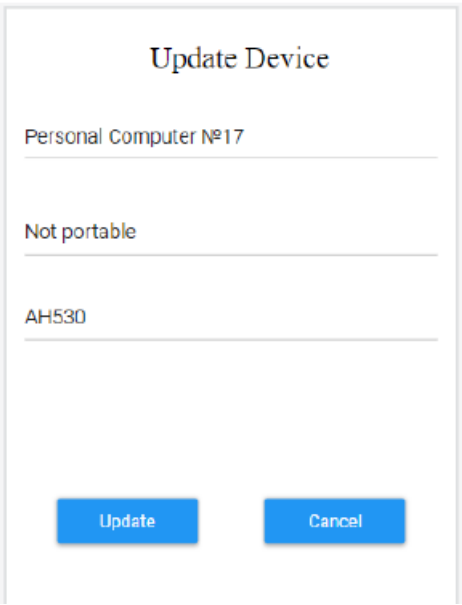


Рис.2.12 Дизайн вікна редагування девайсу

Тепер, коли були розглянутий дизайн базових можливостей, є можливим перейти до дизайну екранів, які дозволяють пов’язувати працівника з девайсом, для цього повертаємось до головного екрану, та натискаємо на одного з існуючих працівників, після натискання на працівника ми потрапляємо до екрану з інформацією про цього працівника (рис 2.13), тут дублюється інформація, яка наведена на головному екрані, а також наводиться список всіх девайсів, які є прив’язані до цього користувача, окрім цього, даний екран має три інтерактивних елемента, а саме це стрілка на тулбарі, яка дозволяє повернутись на головний екран, кнопка видалення девайсу для конкретного працівника, варто зауважити, що при натисканні на неї видаляється не девайс, а зв'язок між ним та конкретним працівником, та кнопка додавання девайсів для працівника, яка знаходиться з правого боку панелі з титулом області з девайсами, та в контексті дизайну виглядає як плюсик. А також в нижній частині розташований фрагмент зі статистикою.

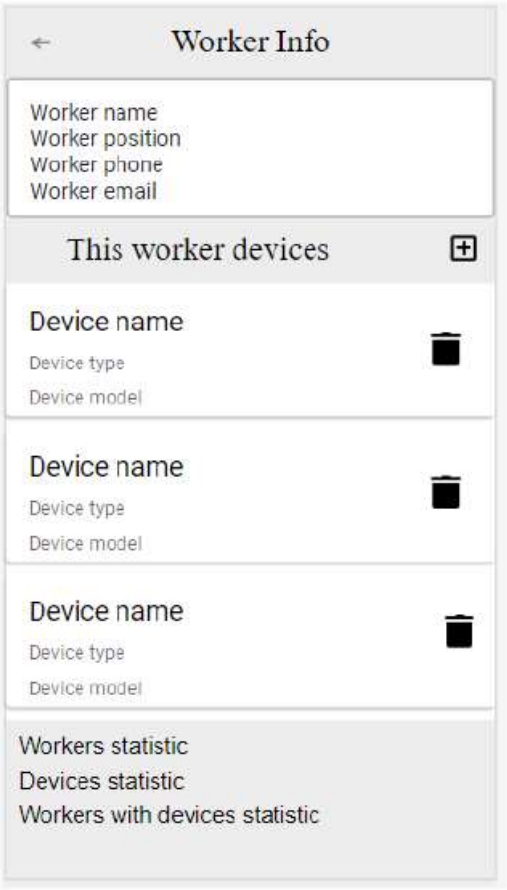


Рис. 2.13 Дизайн екрану інформації про працівника

При натисканні на плюсик, користувач має можливість додати девайси до працівника(рис 2.14), чия інформація була відкрита.

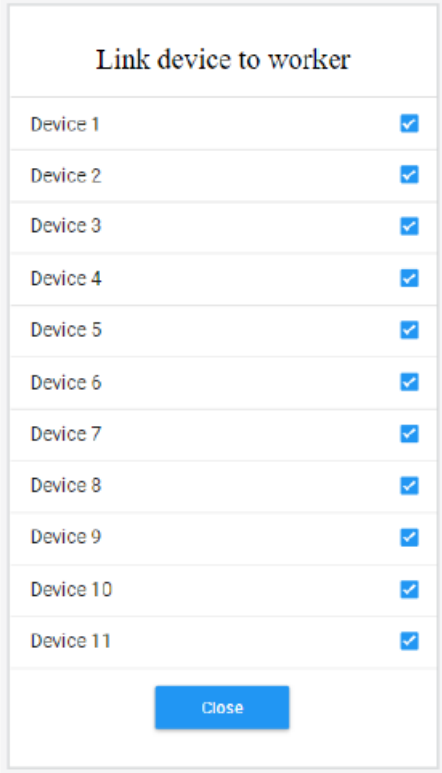


Рис. 2.14 Дизайн екрану додавання девайсів до користувача

Та фінальним екраном з дизайном є фрагмент статистики(рис 3.14), який наводився як частина багатьох екранів раніше.

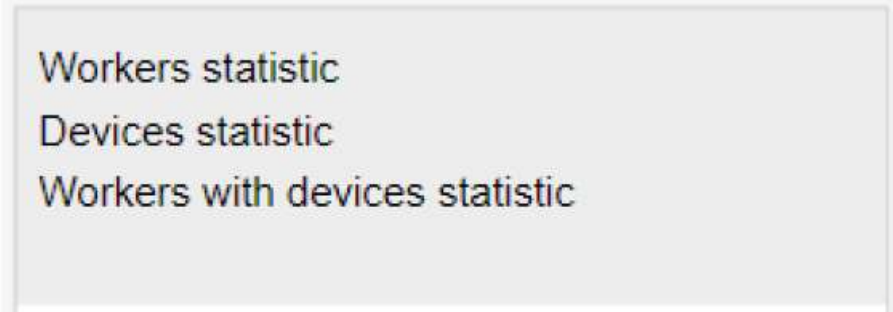


Рис. 2.15 Дизайн фрагменту статистики

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Інформаційна технологія

Комплексна система захисту інформації в рамках підприємства може набувати різних форм, та реалізовуватись з використанням широкого спектру різнопланових засобів, що є зрозумілим виходячи з матеріалів наведених в першому та другому розділах, але існують певні процеси, забезпечення яких є неймовірно важливою частиною інформаційної безпеки. По суті своїй ці процеси є ядром, навколо якого, та на основі результатів якого в подальшому додають нові модулі, нове програмне забезпечення, нові процеси, які покращують безпеку в напрямку необхідному конкретному підприємству.

В контексті даної роботи, було проведено виведення цих процесів та представлення їх у вигляді списку етапів, які є основними для подальшого проектування, та на основі яких і відбувається подальше вдосконалення системи, список цих етапів, в поєднанні з додатком, який покриває функціонально частину з них, є можливим назвати інформаційною технологією проектування комплексної системи захисту інформації на підприємстві.

Цей список етапів, виглядає таким чином:

Етап 1 – Аналіз підприємства

Даний етап проводиться в першу чергу, при виконанні практично будь-яких дій з інформаційною системою підприємства, починаючи від розробки маркетингової стратегії та закінчуючи побудовою системи захисту. На даному етапі проводиться аналіз бізнес-процесів, які відбуваються всередині підприємства, аналіз ієрархії, розділення обов’язків, дослідження діяльності компанії, тобто після виконання даного етапу, отримується повна модель підприємства та його діяльності, що в подальшому використовується для прийняття рішень з приводу доцільності, чи недоцільності інтеграції засобів в контексті конкретно цього підприємства.

Етап 2 – Визначення об’єктів інформаційної системи

Як є відомо, будь-яка інформаційна система складається з об’єктів, суб’єктів та зв’язків між ними, а їх контроль є не те що доцільним, він є необхідним для забезпечення інформаційної безпеки на підприємстві. В ході цього етапу виділяються саме об’єкти, об’єктом є саме інформація, але робота з інформацією не відбувається напряму, люди для цього використовують посередників у вигляді девайсів які мають можливість, записувати, передавати чи відображати інформацію, тому в контексті інформаційної безпеки об’єктами виступають саме ці девайси. В результаті даного етапу, отримується список всіх об’єктів, які працюють в рамках інформаційної системи підприємства.

Етап 3 – Визначення суб’єктів інформаційної системи

Суб’єктами інформаційної системи є люди, які працюють з об’єктами, визначеними в ході попереднього етапу. По суті на даному етапі створюється список всіх працівників компанії, які мають доступ до інформаційної системи, та мають можливість чинити на неї вплив. Варто зауважити, що наприклад в контексті будівельної компанії, існують працівники, які не взаємодіють з системою напряму, це можуть бути будівельники, які отримують задачу від керівництва та займаються її виконанням, а дані до системи про ці процеси заносяться відповідальними за це працівниками.

Етап 4 – Визначення зв’язків в інформаційній системі

На попередніх двох етапах, було визначено списки об’єктів та суб’єктів системи, що вже само по собі дає певний простір для контролю над ними, але без встановлених зв’язків, цей контроль не є повним, та залишається багато прогалин в роботі системи безпеки, наприклад за допомогою програмного забезпечення для моніторингу трафіку, є можливим помітити, що з конкретного девайсу було здійснено несанкціоновану передачу даних поза межі системи, таким чином спеціаліст відповідальний за безпеку знає що відбувся факт передачі, знає з якого девайсу це відбувалось, але не знає хто саме є прив’язаним до цього девайсу. Визначення зв’язків, в свою чергу пов’язує працівника та девайси з якими він працює, і робить ці девайси його зоною

відповідальності, оскільки у випадку ситуації, коли з його девайса було здійснено дії, які можуть зашкодити підприємству, він буде відповідати за це. Таким чином з одного боку зв’язки дозволяють сформувати цільну структуру підприємства, а з іншого підштовхують працівників до дотримання правил інформаційної безпеки при використанні своїх девайсів.

Етап 5 Проведення організаційних заходів для персоналу

Після реалізації попереднього етапу, працівники самі стають зацікавленими в тому, щоб їх девайси були захищені належним чином від неавторизованих дій з ними, але в той самий час, їм необхідна інформація про те, як їх захистити, чи яких правил потрібно дотримуватись при роботі з ними. Для цього був визначений даний етап, який полягає в проведенні організаційних заходів для працівників, на цих заходах працівникам пояснюють елементарні засоби захисту в інформаційному просторі, яких правил варто дотримуватись, а чого робити не варто. Дані заходи не спрямовані на надання працівникам спеціалізованих знань, це лише елементарний курс для того, щоб працівник мав можливість на мінімальному рівні забезпечити безпеку своєї зони відповідальності та не створював небезпеку для всієї системи необачними діями.

Етап 6 – Аналіз ризиків

Етап, який використовується коли базові етапи по забезпеченню захисту вже реалізовані, він полягає у визначенні можливих загрозу, прогнозуванню можливої шкоди та проведенню процесу оцінки. Таким чином отримується статистична інформація щодо ризиків та визначається найбільш небезпечні з них за відношенням шкода – ймовірність, після чого приймаються висновки щодо дій, які доцільно прийняти по відношенню до кожного ризику, спектр цих дій коливається від ігнорування до максимально можливої ліквідації ризику.

Етап 7 – Інтеграція засобів захисту

Після процесу оцінки ризиків та вибору з них найбільш небезпечних, приймаються відповідні дії, які зазвичай полягають в інтеграції певних засобів, програмних чи інженерних до системи, задля того, щоб знизити шанс

виникнення ризику, або ж для того, щоб зменшити потенційну шкоду, завдану ризиком.

В контексті даної роботи здійснюється розробка мобільного додатку, який дозволяє покривати функціонал наведений в 2,3 та 4 етапах даної інформаційної технології [29],[32].

3.2. Вибір інструментарію

Базою додатку, була обрана система android, через свій високий рівень популярності, а також той факт, що існує комфортне середовище для розробки додатків на базі цієї операційної системи [25]. Цим середовищем розробки є Android Studio, його комфортність забезпечується зручним інтерфейсом, а також вбудованим емулятором, завдяки цьому є можливість писати код і відразу без затримок перевіряти, яким чином його робота буде виглядати на телефоні. Також однією з безумовних переваг даного середовища є його поширеність, завдяки їй існує багато ресурсів в мережі Інтернет, на яких можна знайти корисну інформацію щодо розробки, а також безліч форумів на яких є можливим знайти готові відповідь на свої запитання, що в свою чергу дозволяє вирішувати практично будь-яку проблему, яка може виникнути при розробці. Нижче наведений інтерфейс Android Studio (рис. 3.1).

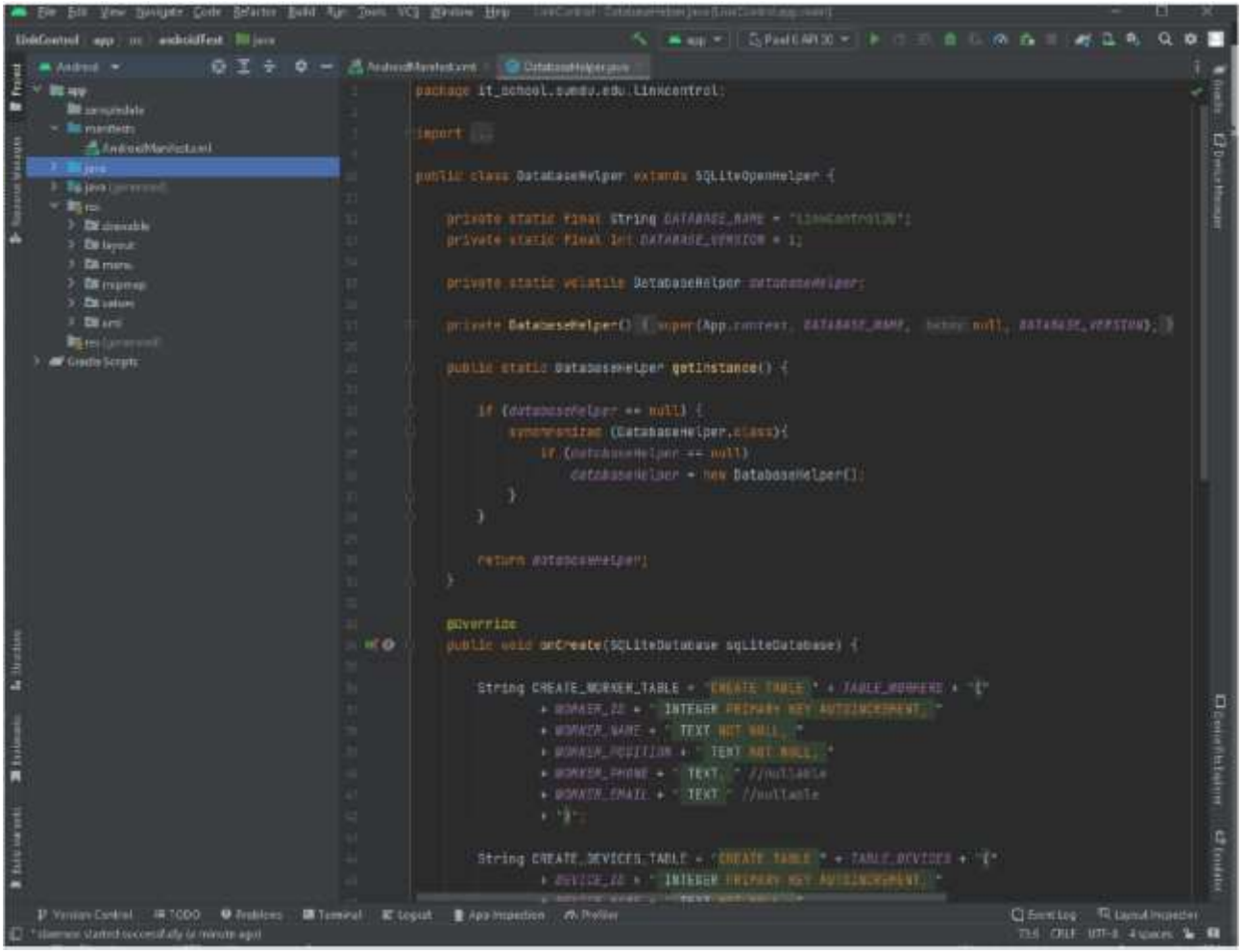


Рис. 3.1 Інтерфейс Android Studio

Важливим аспектом роботи в контексті розроблюваного додатку, є робота з базами даних, Android Studio має вбудований функціонал для роботи з SQLite [27], але при роботі з базами даних лише через Android Studio є доволі складно проводити тестування запитів та перевірку працездатності створених таблиць, тому для допомоги в цьому питанні, був використаний такий інструмент, як DB Browser SQLite (рис. 3.2), за допомогою цього інструменту було створено таблиці, відповідно до вимог поставлених в третьому розділі, після чого було проведено тестування роботи запитів до цих таблиць на коректність, та в результаті, SQL запити для створення таблиць були імпортовані безпосередньо в Android Studio для того щоб виконувати з ними подальшу роботу, вже знаючи що вони працюють саме таким чином, як необхідно в контексті даного додатку.

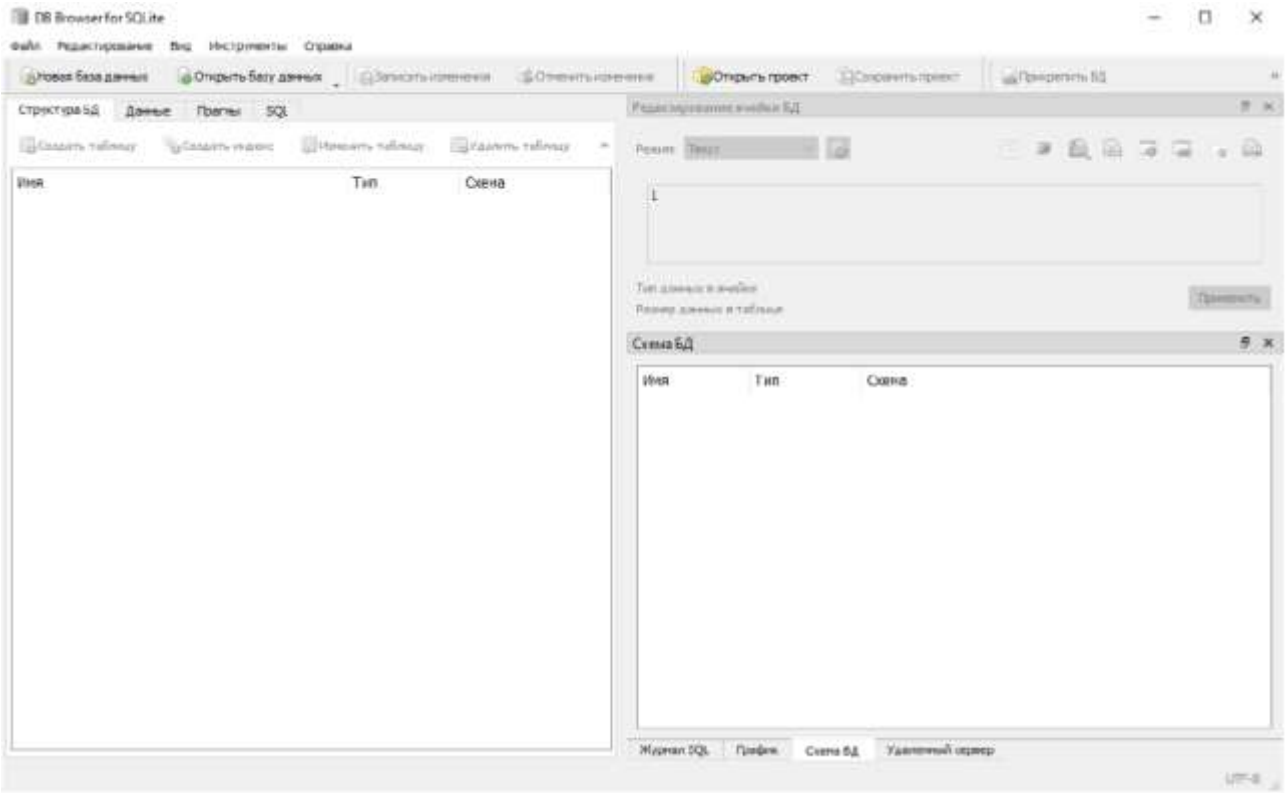


Рис. 3.2 Інтерфейс DB Browser SQLite

За допомогою представлених вище інструментів є можливо реалізувати весь функціонал системи, але при такій реалізації, функціонал авторизації буде дуже ненадійним, оскільки якщо реалізувати авторизацію через SQLite, то дані для неї будуть зберігатись напряму на девайсі користувача, що по суті зводить її користь на 0, оскільки отримавши доступ до телефону користувача, неавторизований користувач відразу має і базу даних для авторизації, через що отримати доступ до додатку не є важкою задачею. Тому для реалізації авторизації було обрано інструмент Firebase (рис. 3.3), цей інструмент дає можливість створити авторизацію користувача через NoSQL Realtime Database, дані для авторизації зберігаються в таблиці на Firebase, таким чином ризик отримання даних для авторизації несанкціонованим користувачем є мінімальним. На рисунку нижче, наведений інтерфейс Firebase, в контексті одного з минулих проектів [31].

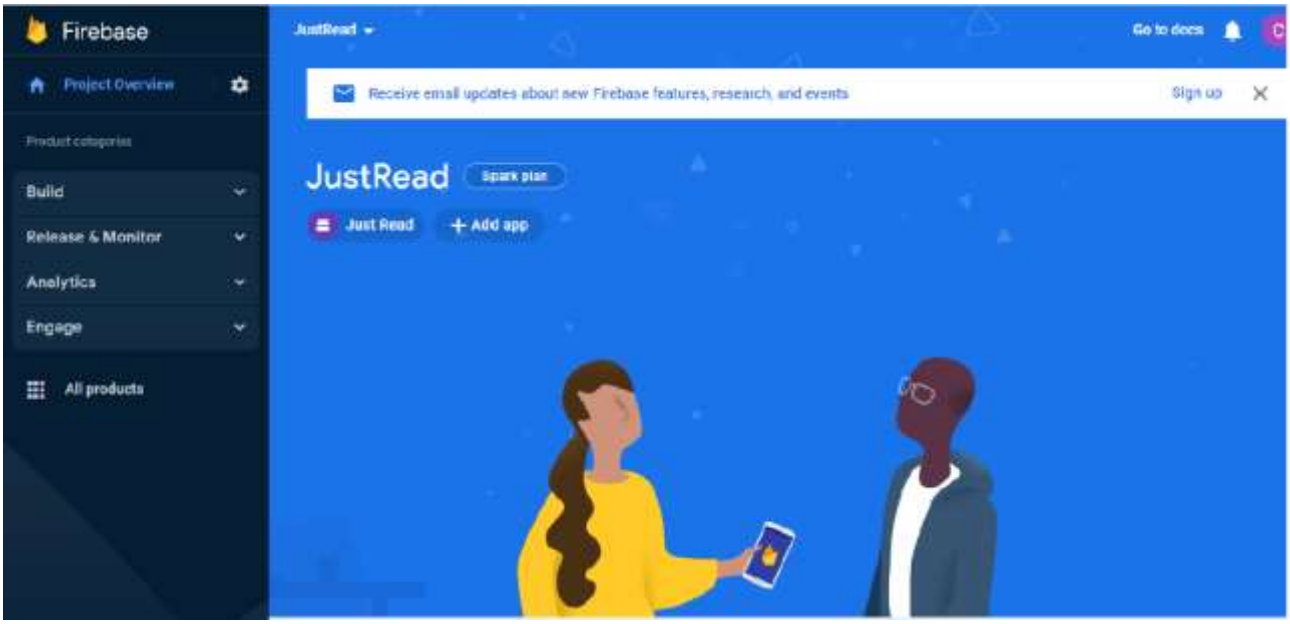


Рис. 3.3 Інтерфейс Firebase

Інструментарій для реалізації додатку обраний, тому є можливим перейти безпосередньо до розробки, а саме, реалізації БД, по моделі представленій в розділі номер три.

3.3. Реалізація БД

Як було сказано в попередньому розділі, в контексті даного проекту використовуються дві БД:

- SQLite для зберігання інформації та надання користувачеві можливості працювати з нею.

- Firebase для авторизації користувача

Для того, щоб реалізувати авторизацію користувача, для початку необхідно пов’язати проект, створений на Android Studio та Firebase, для цього в Firebase створюємо новий проект(рис. 4.4) та додаємо до нього додаток, після чого додаємо до нього авторизацію (рис. 4.5) , створений в середовищі Android Studio, на рисунках червоним контуром виділені основні моменти.

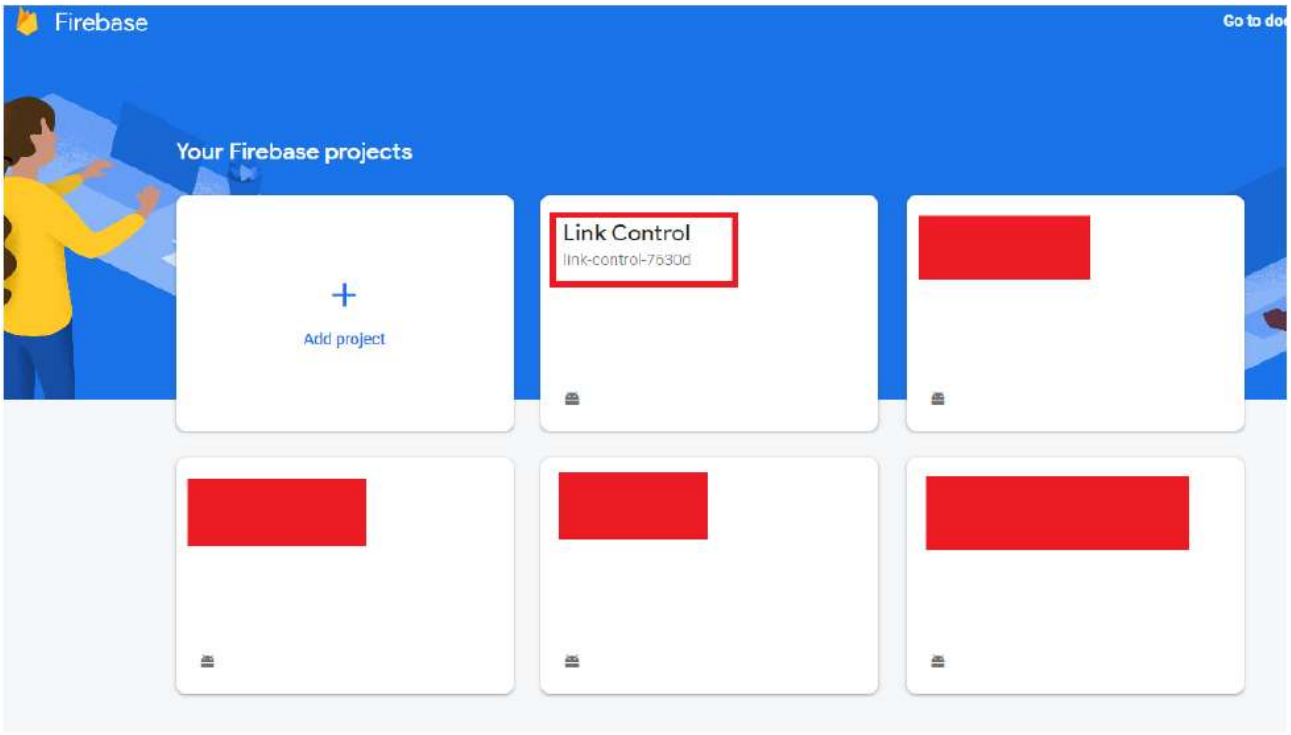


Рис. 3.4 Створення проекту в Firebase

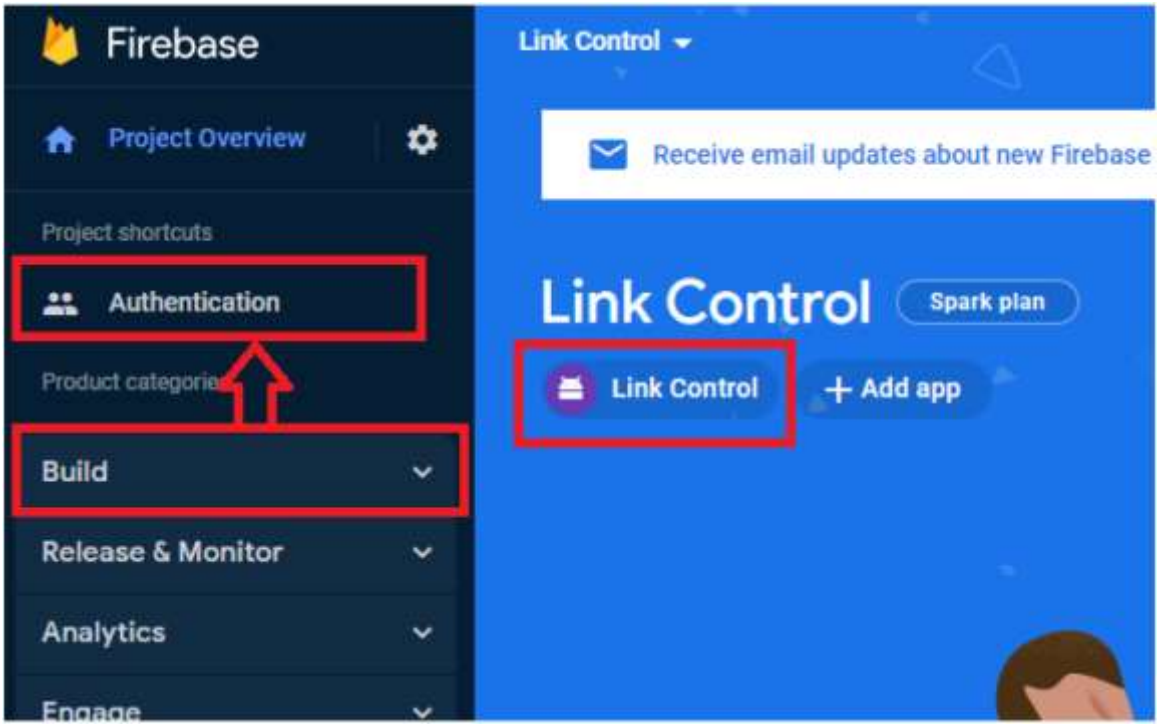


Рис. 3.5 Додавання додатку та авторизації

На вкладці Автентифікація (рис. 3.6), є наявна таблиця користувачів, з якою відбувається зв'язок, для авторизації чи реєстрації, в контексті даного проекту, відбувається лише перевірка ідентичності даних введених користувачем, з тими даними, які зберігаються в таблиці, та у випадку якщо ці дані є ідентичними, користувач допускається до наступної активності. Запис до

цієї таблиці був доданий вручну, оскільки в контексті додатку процес реєстрації, який заповнює таблицю, не буде реалізованим.

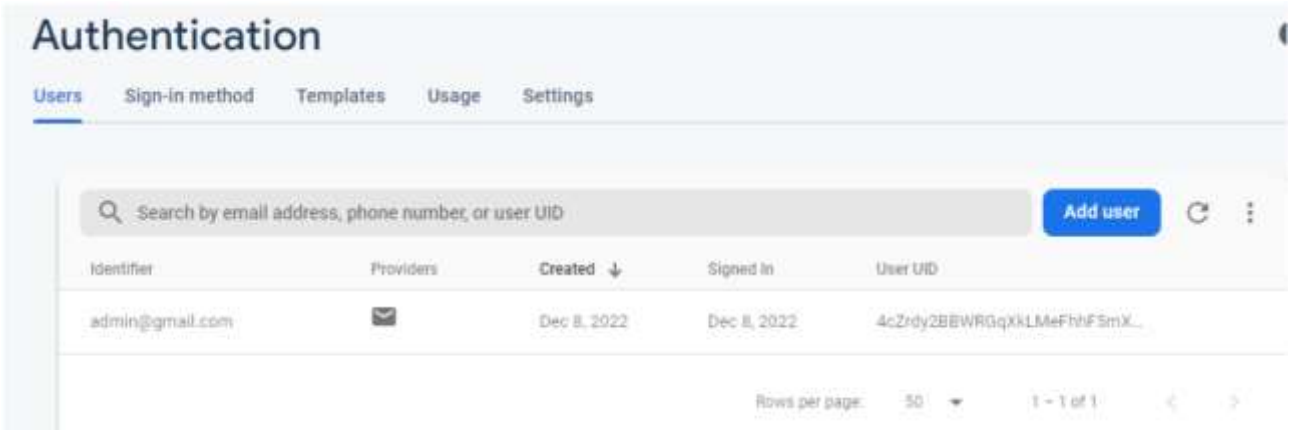


Рис. 3.6 Таблиця користувачів

На цьому етапі, робота з авторизацією зі сторони Firebase є закінченою, наступним кроком є додавання необхідних залежностей в Android Studio та синхронізація середовища з Firebase. Для цього, спочатку обираємо в інструментах Firebase та функціонал, який нам необхідний, в контексті даного проекту це автентифікація через користувальницьку систему(рис. 3.7), після чого ми отримуємо список етапів, які потрібно виконати для реалізації даного функціоналу (рис. 3.8).

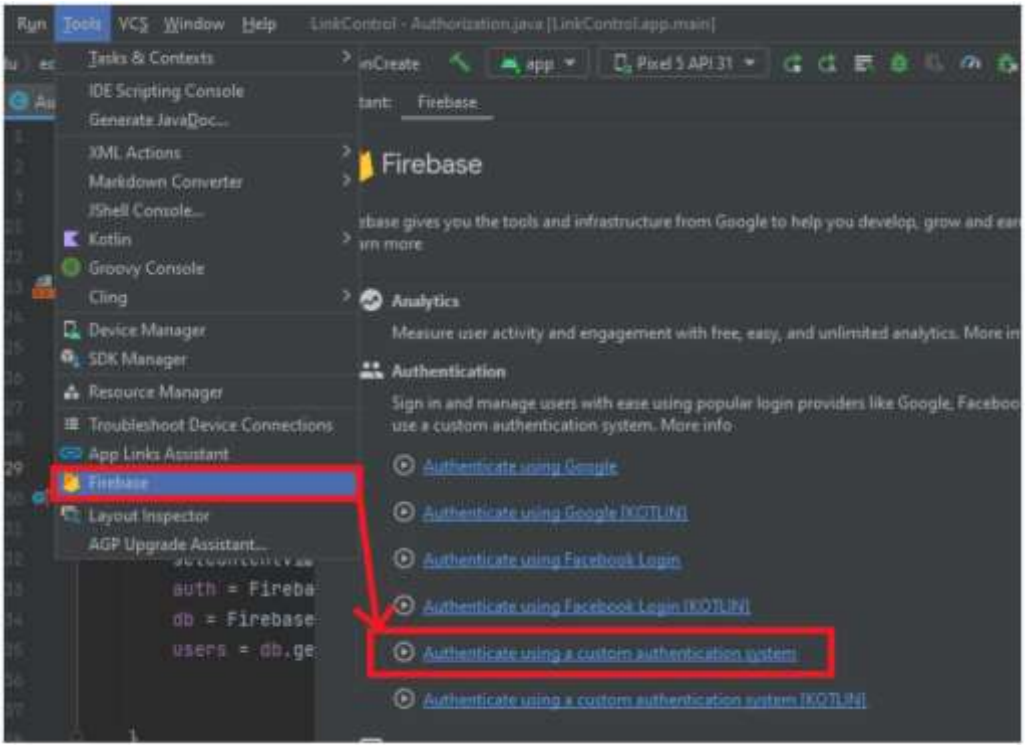


Рис. 3.7 Вибір функціоналу до реалізації

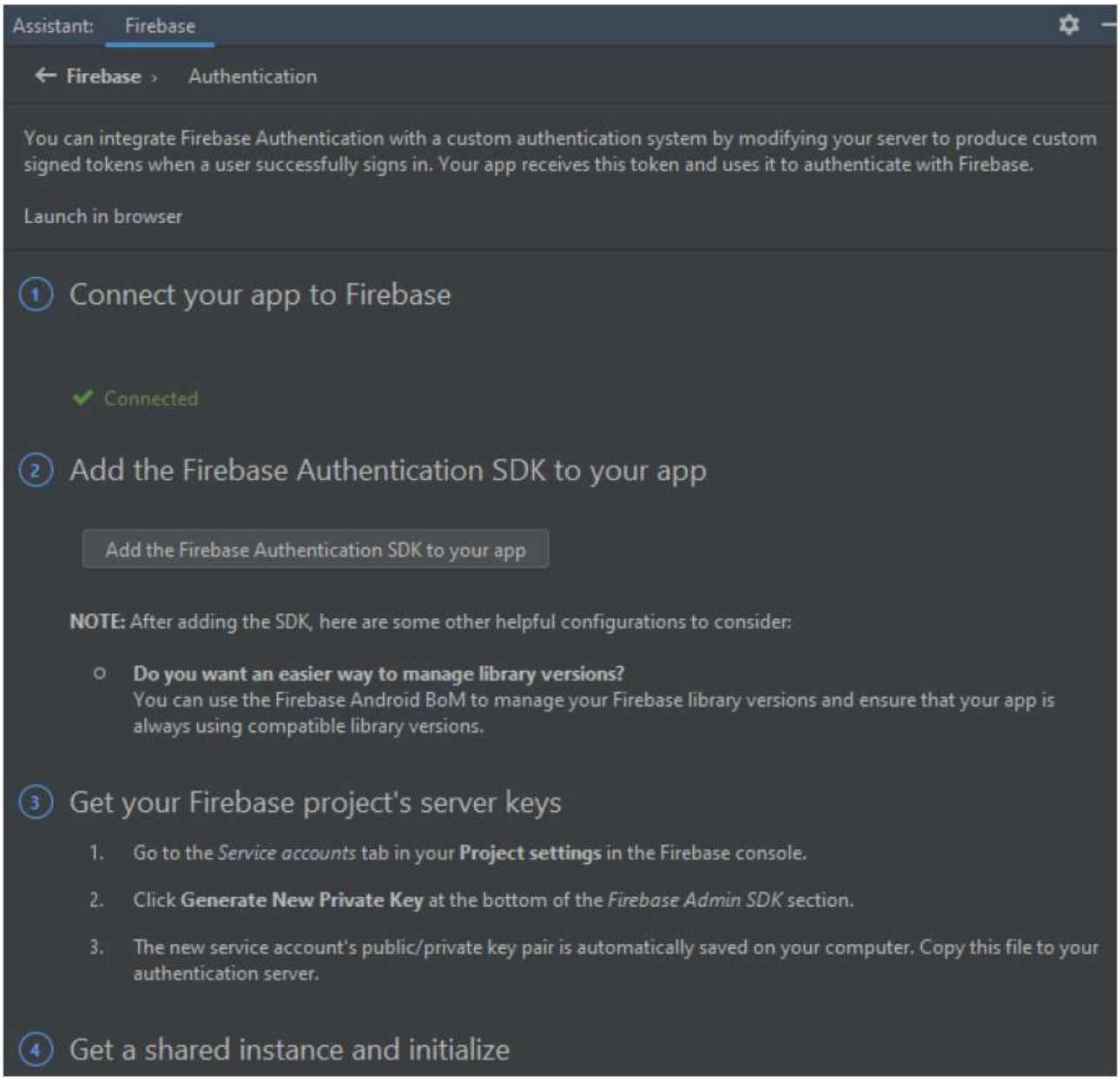


Рис. 3.8 Етапи реалізації

Після виконання всіх цих етапів, база є синхронізована з додатком та є можливим організовувати авторизацію користувача. Тобто робота з БД Firebase в контексті даного проекту є завершена, залишається лише реалізувати візуальний інтерфейс авторизації та викликати необхідні функції, які були наведені в етапах (рис. 3.8).

Далі йде організація роботи з SQLite DB, як було сказано в попередньому розділі, Android Studio підтримує SQLite та має вбудовані інструменти для роботи з ним, але робота за використанням лише цих інструментів, є доволі незручною, оскільки тестувати створення таблиць та запити до них через програмний код є доволі незручним, та може привести до помилок при

реалізації навіть добре спроектованої бази. Таким чином для вирішення цього питання було обрано допоміжний інструмент, SQLite DB Browser, даний інструмент дає можливість створювати таблиці та тестувати їх за допомогою зручного інтерфейсу, а також в подальшому експортувати запити в Android Studio, таким чином за допомогою цього інструмента є можливим бути впевненим що запити експортовані в середовище розробки працюють саме таким чином, який є необхідний для розробника. Використовуючи схему, побудовану в розділі моделювання створимо відповідні таблиці в середовищі SQLite DB Browser (рис. 3.9).

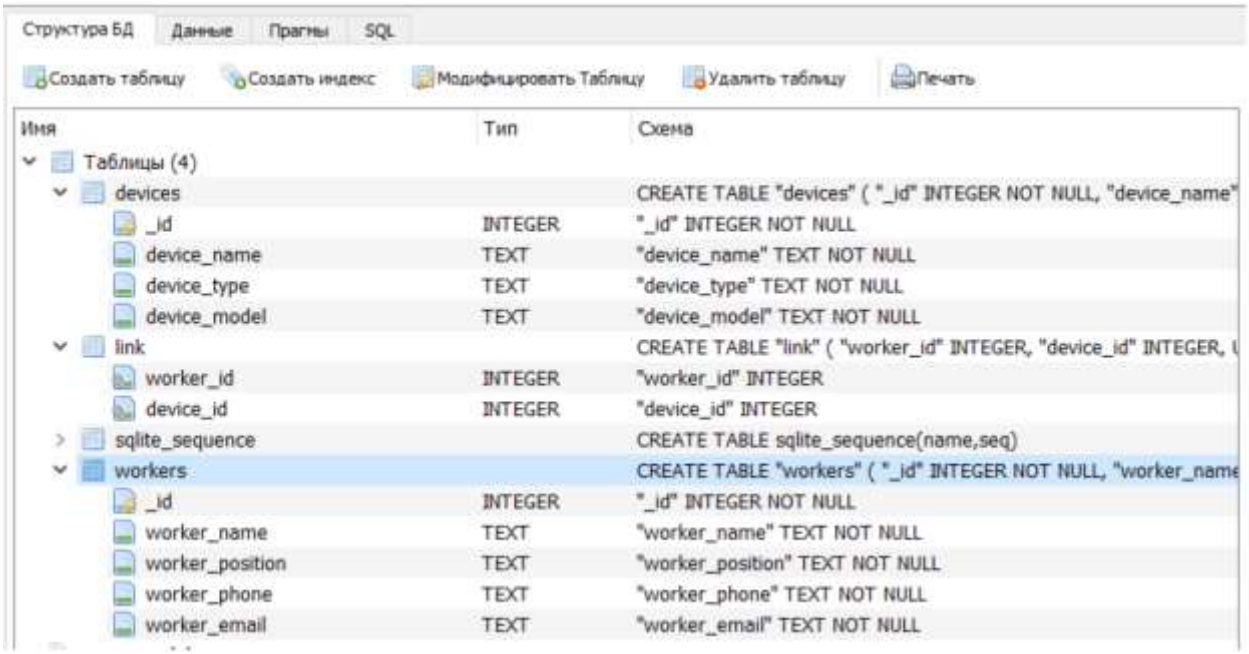


Рис. 3.9 Реалізація таблиць БД

Наступним кроком є наповнення таблиць даними для тестування та розробка запитів для перевірки коректності роботи БД. Для цього створюємо запит, який виводить інформацію про всі девайси пов’язані з конкретним працівником та тестуємо його працездатність (рис 4.10).

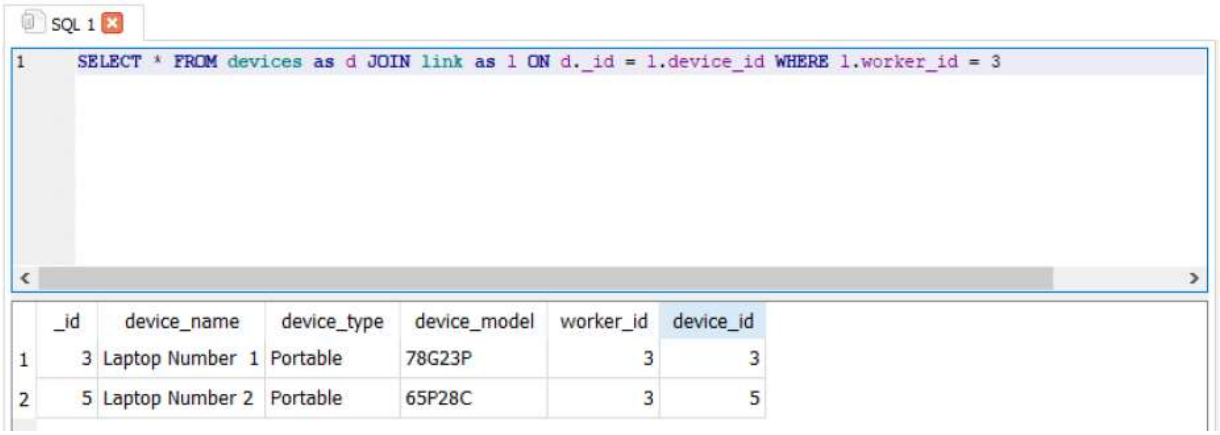


Рис. 3.10 Тестування запиту

Для більш зручного тестування, в БД була додана невелика кількість записів, таким чином є можливо вручну перевірити чи є результат виведений запитом коректним, в результаті перевірки було виявлено що так, результат є коректним, отже БД та запит працюють потрібним нам чином та є спроектовані відповідно до схеми представленої раніше, тому можливо переходити до етапу прямої розробки додатку в середовищі Android Studio, до якої експортувати запити на створення таблиць та на виконання дій з ними.

3.4. Розробка додатку

При роботі в середовищі Android Studio є два основних функціональних елемента, це java класи та layout елементи. Java клас відповідає за логіку додатку, прописує функціональну частину його роботи, а в контексті даного додатку використовується також для створення БД та установлення зв’язків з нею. Layout елемент, це елемент розмітки, тобто це представлення дизайну конкретного екрану, або фрагменту, яке складається з множини елементів розміщених та закріплених на макеті таким чином, яким це необхідно для розробника, доступ до цих елементів отримується за допомогою функції FindViewById, після чого стає можливим впливати з класу на макет. В контексті даного додатку було розроблено 35 Java класів та 19 layout елементів, повний код додатку, включаючи лістинг цих класів та елементів, а також лістинг файлу маніфесту, наведений в додатку Б.

В даному розділі розберемо основні моменти розробки додатку, такі як:

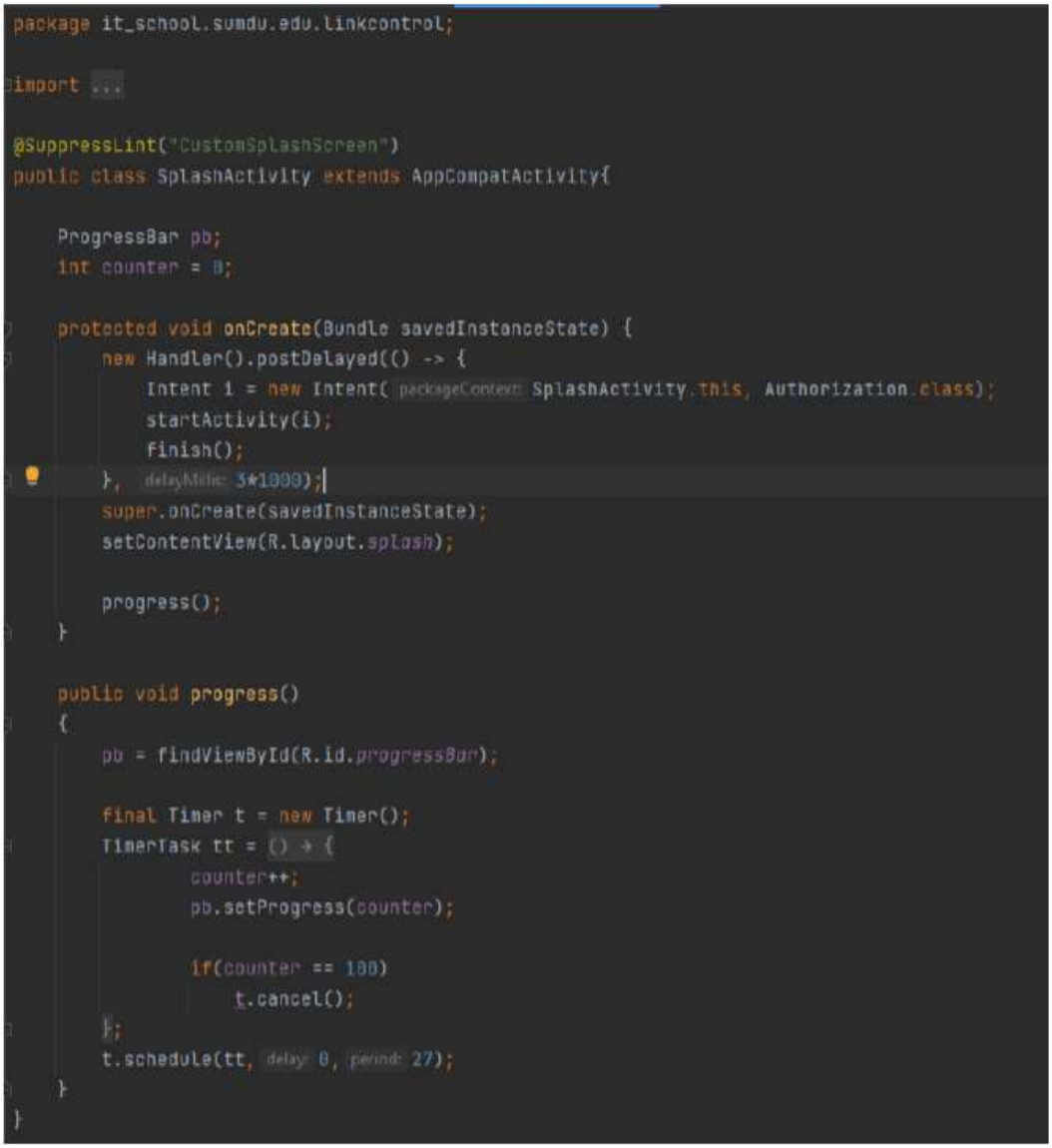
- Екран заставки
- Логіка авторизаціїта
- Створення БД
- Модель працівника
- Функція читання всіх працівників з БД
- Функція читання девайсів привязаних до конкретного працівника з БД

Для розробки екрана заставки, спочатку розробляється layout елемент, відповідно до дизайну розробленому в попередньому розділі (рис.4.11).



Рис. 3.11 Layout SplashScreen

Далі необхідно прописати логіку його роботи, а саме вказати час, на який заставка затримається на екрані, екран на який буде здійснено перехід після заставки, а також запустити ProgresBar елемент. Для цього був написаний код в класі Splash Activity (рис. 3.12).



```
package it_school.sumdu.edu.linkcontrol;

import ...

@SuppressLint("CustomSplashScreen")
public class SplashActivity extends AppCompatActivity{

    ProgressBar pb;
    int counter = 0;

    protected void onCreate(Bundle savedInstanceState) {
        new Handler().postDelayed(() -> {
            Intent i = new Intent( packageContext: SplashActivity.this, Authorization.class);
            startActivity(i);
            finish();
        }, delayMillis: 3*1000);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splash);

        progress();
    }

    public void progress()
    {
        pb = findViewById(R.id.progressBar);

        final Timer t = new Timer();
        TimerTask tt = () -> {
                counter++;
                pb.setProgress(counter);

                if(counter == 100)
                    t.cancel();
        };
        t.schedule(tt, delay: 0, period: 27);
    }
}
```

Рис. 3.12 Код SplashActivity

Як ми бачимо з рисунку наведеного вище, після затримки, в 3 секунди, заставка змінюється на екран Авторизація, а також за ці три секунди, за допомогою функції progress() здійснюється робота елементу типу progressbar.

Наступним кроком є огляд авторизації в рамках даного проекту, цей процес виглядає таким чином, користувач потрапляє на екран з короткою інструкцією та кнопкою авторизації (рис. 3.13), при натисканні на цю кнопку, відкривається діалогове вікно (рис. 3.14), з полями в які необхідно ввести інформацію користувача, та двома кнопками, одна з яких перевіряє інформацію та в випадку відповідності дає доступ до додатку, а інша скасовує операцію і закриває діалогове вікно.



Рис. 4.13 Layout авторизації



Рис. 3.14 Layout діалогового вікна

Логіка для цих двох layout прописана в класі Authorization (рис. 3.15-3.16).

```
{
    FirebaseAuth auth;
    FirebaseDatabase db;
    DatabaseReference users;

    @Override
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_auth);
        auth = FirebaseAuth.getInstance();
        db = FirebaseDatabase.getInstance();
        users = db.getReference("Users");


    }

    public void Sign(View view) {
        AlertDialog.Builder dialog = new AlertDialog.Builder( context: this);
        dialog.setTitle("Authorization");
        dialog.setMessage("Enter your details to log into your account");

        LayoutInflater inflater = LayoutInflater.from(this);
        View sign_in_window = inflater.inflate(R.layout.authorization, root: null);
        dialog.setView(sign_in_window);

        final EditText mail = sign_in_window.findViewById(R.id.mailFieldAuth);
        final EditText pass = sign_in_window.findViewById(R.id.passwordFieldAuth);

        dialog.setNegativeButton("Cancel", (dialogInterface, i) -> dialogInterface.dismiss());

        dialog.setPositiveButton("Authorization", (dialogInterface, i) -> {

            if(TextUtils.isEmpty(mail.getText().toString())||pass.getText().toString().length()<5)
            {
                Toast toast = Toast.makeText(getApplicationContext(),
                        "Incorrect filling of fields", Toast.LENGTH_SHORT);
                toast.show();
```

Рис. 3.15 Код класу авторизації, частина перша

```
                return;
            }
            auth.signInWithEmailAndPassword(mail.getText().toString(),pass.getText().toString())
                    .addOnSuccessListener(authResult -> {
                        Intent intent = new Intent( packageContext: this,WorkerListActivity.class);
                        startActivity(intent);
                    }).addOnFailureListener(e -> {
                        Toast toast = Toast.makeText(getApplicationContext(),
                                ""Error: "" + e.getMessage(), Toast.LENGTH_SHORT);
                        toast.show();
                    });
        });


        dialog.show();
    }
}
```

Рис. 3.16 Код класу авторизації, частина друга

Як ми бачимо, авторизація відбувається в методі Sign, цей метод включає в себе виклик діалогового вікна, макету для нього, а також подальшу звірку даних введених користувачем, з даними, які є збережені в Firebase.

Після того, як користувач здійснив авторизацію, він потрапляє до основного функціоналу додатку, та бачить список з користувачами виведеними з БД, як було сказано в попередньому підрозділі, ця БД створюється напряму в класі Android Studio DatabaseHelper, за допомогою запитів (рис. 3.17) сформованих в SQLite DB Browser.

```
blic void onCreate(SQLiteDatabase sqLiteDatabase) {

String CREATE_WORKER_TABLE = "CREATE TABLE " + TABLE_WORKERS + "("
        + WORKER_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "
        + WORKER_NAME + " TEXT NOT NULL, "
        + WORKER_POSITION + " TEXT NOT NULL, "
        + WORKER_PHONE + " TEXT, " //nullable
        + WORKER_EMAIL + " TEXT " //nullable
        + ")";

String CREATE_DEVICES_TABLE = "CREATE TABLE " + TABLE_DEVICES + "("
        + DEVICE_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "
        + DEVICE_NAME + " TEXT NOT NULL, "
        + DEVICE_TYPE + " TEXT NOT NULL , "
        + DEVICE_MODEL + " TEXT NOT NULL" //nullable
        + ")";

String CREATE_LINK_TABLE = "CREATE TABLE " + TABLE_LINK + "("
        + WORKER_ID_FK + " INTEGER NOT NULL, "
        + DEVICE_ID_FK + " INTEGER NOT NULL, "
        + "FOREIGN KEY (" + WORKER_ID_FK + ") REFERENCES " + TABLE_WORKERS + "(" + WORKER_ID + ") " +
        "ON UPDATE CASCADE ON DELETE CASCADE, "
        + "FOREIGN KEY (" + DEVICE_ID_FK + ") REFERENCES " + TABLE_DEVICES + "(" + DEVICE_ID + ") " +
        "ON UPDATE CASCADE ON DELETE CASCADE, "
        + "CONSTRAINT " + CONSTRAINT + " UNIQUE (" + WORKER_ID_FK + "," + DEVICE_ID_FK + ")"
        + ")";

sqLiteDatabase.execSQL(CREATE_WORKER_TABLE);
sqLiteDatabase.execSQL(CREATE_DEVICES_TABLE);
sqLiteDatabase.execSQL(CREATE_LINK_TABLE);
```

Рис. 3.17 Створення таблиць

Значення для назв таблиць, та їх колонок, зберігаються в окремому класі Constants (рис. 3.18),це зроблено для того, щоб не додавати багато зайвих змінних в DatabaseHelper та в подальшому зробити зручнішим механізм звертання до цих таблиць в рамках решти частини коду.

```
public class Constant {

    public static final String TABLE_WORKERS = "workers";
    public static final String WORKER_ID = "_id";
    public static final String WORKER_NAME = "name";
    public static final String WORKER_POSITION = "registration_no";
    public static final String WORKER_PHONE = "phone";
    public static final String WORKER_EMAIL = "email";
    public static final String TABLE_DEVICES = "devices";
    public static final String DEVICE_ID = "_id";
    public static final String DEVICE_NAME = "name";
    public static final String DEVICE_TYPE = "type";
    public static final String DEVICE_MODEL = "model";
    public static final String TABLE_LINK = "link";
    public static final String WORKER_ID_FK = "worker_id";
    public static final String DEVICE_ID_FK = "device_id";
    public static final String CONSTRAINT = "link_unique";
    public static final String TITLE = "title";
    public static final String CREATE_WORKER = "create_worker";
    public static final String UPDATE_WORKER = "update_worker";
    public static final String CREATE_DEVICE = "create_device";
    public static final String UPDATE_DEVICE = "update_device";
}
```

Рис. 3.18. Клас Constants

Також створюються моделі для елементів кожної з таблиць, які дають можливість користувачу встановлювати нові значення, або отримувати значення, які є в наявності в БД. Наприклад модель працівника (рис. 3.19)

```java
package it_school.sumdu.edu.linkcontrol.model;

public class Worker {
    private int id;
    private String name;
    private String position;
    private String phone;
    private String email;

    public Worker(int id, String name, String position, String phone, String email) {
        this.id = id;
        this.name = name;
        this.position = position;
        this.phone = phone;
        this.email = email;
    }

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public String getPosition() { return position; }

    public void setPosition(String position) { this.position = position; }

    public String getPhone() { return phone; }

    public void setPhone(String phone) { this.phone = phone; }

    public String getEmail() { return email; }

    public void setEmail(String email) { this.email = email; }
}
```

Рис. 3.19 Модель працівника

Такі моделі, відповідно створюються для кожної з наявних таблиць. Далі створюється функція readAllWorkers (рис. 3.20), яка дозволяє читати інформацію про всіх працівників з БД та додає її в ArrayList, а в подальшому, за допомогою WorkerListActivity, з підтримкою WorkerListAdapter та WorkerListViewHolder, інформація з цього ArrayList виводиться вдоситься в відповідні поля CardView та в подальшому ці елементи CardView(рис. 3.21.) додаються в RecycleView і показується користувачеві.

```
public void readAllWorker(Response<List<Worker>> response) {
    SQLiteDatabase sqLiteDatabase = databaseHelper.getReadableDatabase();

    List<Worker> workerList = new ArrayList<>();

    Cursor cursor = null;
    try {
        cursor = sqLiteDatabase.query(TABLE_WORKERS,  columns: null,  selection: null,  selectionArgs: null,
                groupBy: null,  having: null,  orderBy: null);

        if(cursor!=null && cursor.moveToFirst()){
            do {
                Worker worker = getWorkerFromCursor(cursor);
                workerList.add(worker);
            } while (cursor.moveToNext());

            response.onSuccess(workerList);
        } else
            response.onFailure("Worker list is empty");

    } catch (Exception e){
        response.onFailure(e.getMessage());
    } finally {
        sqLiteDatabase.close();
        if(cursor!=null)
            cursor.close();
    }
}
```

Рис. 3.20 Функція readAllWorkers

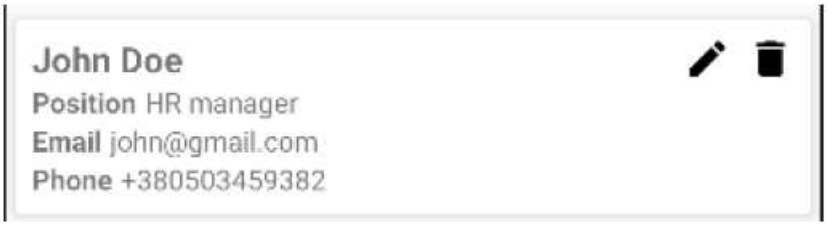


Рис. 3.21 Layout item_worker_card_view

Також в даному додатку існує функціонал, який дозволяє продивитись інформацію про працівника, та девайси, які є повязані з ним, для цього, кожен запис в RecycleView, вкладається в форматі CardView (рис. 3.21), таким чином існує можливість взаємодії з ним, і користувач, після натискання на конкретного працівника, потрапляє до активності WorkerInfo (рис. 3.22), де окрім інформації цього користувача, наведені девайси, які є пов’язані з ним, це реалізується за допомогою функції readAllLinkDeviceByWorkerId (рис.3.23), в рамках цієї функції є можливим бачити запит, за допомогою якого проводилось

тестування працездатності моделі БД, створеної в SQLite Database Browser, відмінність полягає в тому, що в запиті для тестування, змінна WorkerID обиралась вручну, та була рівна 3, а в контексті даної функції, ця змінна є динамічною, та додається в запит залежно від елементу, на який натиснув користувач, тобто, якщо користувач натиснув на на працівника, _id якого рівне 5, то запит буде виконуватись з workerId==5.

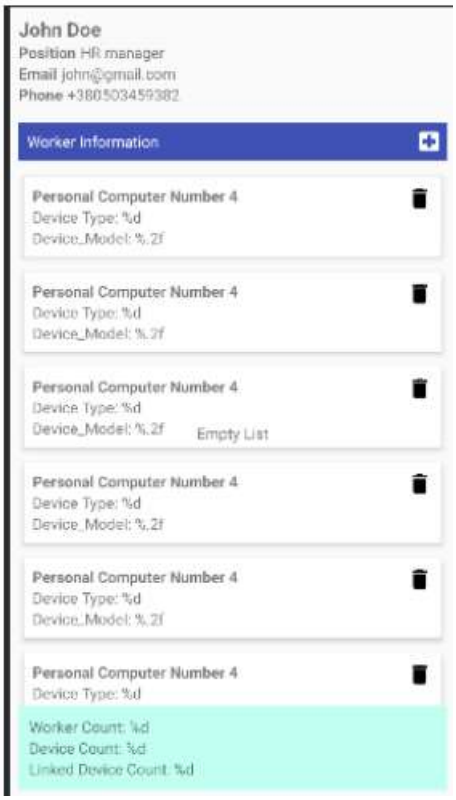


Рис. 3.22 Layout activity_worker_info

```
@Override
public void readAllLinkDeviceByWorkerID(int workerId, Response<List<Device>> response) {
    SQLiteDatabase sqLiteDatabase = databaseHelper.getReadableDatabase();

    String QUERY = "SELECT * FROM devices as d JOIN link as l ON d._id = l.device_id " +
            "WHERE l.worker_id = " + workerId;
    Cursor cursor = null;
    try {
        List<Device> deviceList = new ArrayList<>();
        cursor = sqLiteDatabase.rawQuery(QUERY, selectionArgs: null);

        if(cursor.moveToFirst()){
            do {
                @SuppressLint("Range") int id = cursor.getInt(cursor.getColumnIndex(DEVICE_ID));
                @SuppressLint("Range") String deviceName = cursor.getString
                        (cursor.getColumnIndex(DEVICE_NAME));
                @SuppressLint("Range") String deviceType = cursor.getString
                        (cursor.getColumnIndex(DEVICE_TYPE));
                @SuppressLint("Range") String deviceModel = cursor.getString
                        (cursor.getColumnIndex(DEVICE_MODEL));

                Device device = new Device(id, deviceName, deviceType, deviceModel);
                deviceList.add(device);

            } while (cursor.moveToNext());

            response.onSuccess(deviceList);
        } else
            response.onFailure("Link device list is empty");

    } catch (Exception e){
        response.onFailure(e.getMessage());
    } finally {
        sqLiteDatabase.close();
        if (cursor!=null)
            cursor.close();
    }
}
```

Рис.3.23 Функція readAllLinkDeviceByWorkerId

Варто зауважити, що в рамках даного підрозділу представлена реалізація базової частини функціоналу, повний код наведений в додатку Б, в ньому представлені всі класи та layout елементи, які були розроблені в рамках виконання даного проекту.

Таким чином результатом реалізації, за моделюванням описаним в третьому розділі, став мобільний додаток, який дозволяє авторизованому користувачеві отримувати доступ до БД та вносити в неї дані про працівників та девайси на підприємстві, а також встановлювати зв’язки між ними, що в свою чергу дозволяє організувати контроль за об’єктами та суб’єктами інформаційної системи на підприємстві, та функціонально покриває етапи під номерами 2,3,4 в технології описаній в рамках підрозділу 3.1 [24][25] .

ВИСНОВОК

Результатом виконання роботи, стала інформаційна технологія проектування базової системи захисту підприємства, описана в підрозділі 4.1., яка представляє з себе список етапів, які не є ресурсозатратними, а також більшість з них може бути реалізована особою без спеціалізованих навичок в сфері інформаційних технологій. Таким чином, реалізація цих етапів, дозволить сформувати захист інформації елементарного рівня на підприємстві, що в свою чергу, значно зменшує шанс виникнення проблем інформаційної безпеки.

В ході виконання даної роботи, також було розроблено програмний продукт, а саме додаток на основі Android “Link control”, даний додаток дозволяє контролювати зв’язки між об’єктами та суб’єктами в інформаційній системі підприємства. Основою для визначення функціональних вимог до додатку стала об'єктно-орієнтована методологія проектування комплексних систем захисту інформації. Таким чином, використання додатку, дозволить користувачу контролювати інформаційну систему підприємства та приймати заходи для підвищення рівня інформаційної безпеки цього підприємства. Також, окрім обліку, в додаток є влаштована область для контролю задач, таким чином користувач зможе ставити задачі, після чого позначати виконані, та ті які ще потребують виконання. Інформацію додаток отримує з бази даних, в яку користувач при потребі може додавати нові записи, використовуючи графічний інтерфейс додатку.

Окрім додатку, який є практичною складовою даної роботи, також була створена теоретична частина, яка висвітлює основні положення щодо створення комплексних систем захисту інформації, а також надає інформацію щодо грамотного проектування систем такого роду, та основних аспектів, які є ключовими в питанні комплексних систем захисту інформації. Були розглянуті основні критерії, відповідністю до яких проводиться оцінка якості системи захисту інформації, а також наведено приклад реальних загроз для кожного з

цих критеріїв, разом з прикладами того, як цих загроз можливо уникнути, чи мінімізувати.

Отже підсумовуючи всю роботу є можливим виділити основні результати, а саме:

- Було сформульовано інформаційну технологію проектування базової системи захисту інформації
- Було розроблено додаток на базі Android
- Спроектована та наповнена даними БД
- Створено теоретичний огляд питання КСЗІ
- Розглянуто основні засоби та методи захисту інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про затвердження Порядку проведення державної експертизи комплексних систем захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах Служби безпеки України : Наказ Служби безпеки України від 22.07.2019 р. № 1132. URL: https://zakon.rada.gov.ua/laws/show/z0910-19#Text (дата звернення: 05.10.2023).

2. Комплексна система захисту інформації (КСЗІ). URL: https://uss.gov.ua/service/kszi/(дата звернення: 05.10.2023).

3. Яремчук Ю.Є. Комплексні системи захисту інформації : навч. посіб. /Вінниця: Вид-во ВНТУ, 2018. 118 с.

4. Необхідність створення комплексної системи захисту інформації. URL: https://tzi.com.ua/neobxdnst-stvorennya-kompleksno-sistemi-zaxistu-nformacz-ksz.html (дата звернення: 05.10.2023).

5. Principles and order of developing complex information security systems in information and telecommunication systems. U.V. Zemlyanko, A.A. Zamula, A.A. Tkach, N.I. Litvinova, Y.A. *Peresechanskaya. Applied Radio Electronics: Sci. Mag.* 2010. Vol. 9. № 3. P. 460-469 (дата звернення: 05.10.2023).

6. ISO 27000. Wikipedia. URL: https://en.wikipedia.org/wiki/ISO_27000 (дата звернення: 05.10.2023).

7. ДСТУ ISO/IEC 27001:2015. Інформаційні технології. Методи захисту. Системи управління інформаційною безпекою. Вимоги. На заміну ISO/IEC 27001:2005 ; чинний від 2019-11-01. Вид. офіц.

8. ДСТУ ISO/IEC 27005:2019. Інформаційні технології. Методи захисту. Управління ризиками інформаційної безпеки. На заміну ДСТУ ISO/IEC 27005:2015 ; чинний від 2019-11-01. Вид. офіц.

9. ДСТУ ISO/IEC 27003:2018. Інформаційні технології. Методи захисту. Системи керування інформаційною безпекою. Настанова. Чинний від 2018-10-01. Вид. офіц.

10. ДСТУ ISO/IEC 27004:2018. Інформаційні технології. Методи захисту. Системи керування інформаційною безпекою. Моніторинг, вимірювання, аналізування та оцінювання. Чинний від 2018-10-01. Вид. офіц.

11. Security information and event management (SIEM) URL: https://www.imperva.com/learn/application-security/siem/

12. Andrzejewski, K. (2020). Security information management systems. Management Sciences, 24(4). URL: https://doi.org/10.15611/ms.2019.4.01 (дата звернення: 05.10.2023).

13. Про захист інформації в інформаційно-телекомунікаційних системах: Закон України від 05.07.1994 р. № 80/94-ВР: станом на 1 лип. 2022 р. URL: https://zakon.rada.gov.ua/laws/show/80/94-вр#Text (дата звернення: 09.10.2023).

14. Жураковський Б. Ю., Недашківський О. Л. Системазахисту інформації при передачі даних в радіоканалі. *Кібербезпека: освіта, наука, техніка.* 2022. No 3(15).

15. Перелік документів системи технічного захисту інформації (НД ТЗІ) URL: https://cip.gov.ua/ua/news/perelik-dokumentiv-sistemi-tekhnichnogo-zakhistu-informaciyi-nd-tzi (дата звернення: 05.10.2023).

16. Components of Mustonen-Ollila, E., Lehto, M., & Heikkonen, J. (2020). Components of defence strategies in society's information environment: a case study based on the grounded theory. Security and Defence Quarterly, 28(1). URL: https://doi.org/10.35467/sdq/118186 (дата звернення: 05.10.2023).

17. Делембовський, М., Шабала, Є., & Терентьєв, О. (2021). Аналіз методів та шляхів вирішення захисту інформації в інформаційно-телекомунікаційних системах. Грааль науки. URL: https://doi.org/10.36074/grail-of-science.19.02.2021.051 (дата звернення: 05.10.2023).

18. Сідлецький Є. В. Організаційні засоби захисту інформації в системах електронного документообігу. *Новітні технології у науковій діяльності і навчальному процесі: зб. тез Всеукр. наук.-практ. конф. студентів, аспірантів та молодих учених (м. Чернігів, 8-9 квіт. 2020 р.)* : збірник тез доп. Чернігів : НУ «Чернігівська політехніка», 2020. С. 144-146.

19. Криптографічні засоби захисту інформації URL: https://studfile.net/preview/5462915/ (дата звернення: 05.10.2023).

20. Shcheblanin, Y., & Rabchun, D. (2018). Mathematical model of information security's theat agent. *Cybersecurity: Education, Science, Technique*, 1. URL: https://doi.org/10.28925/2663-4023.2018.1.6372 (дата звернення: 05.10.2023).

21. S. Btoush, E., & M. Hammad, M. (2015). Generating ER Diagrams from Requirement Specifications Based On Natural Language Processing. *International Journal of Database Theory and Application*, 8(2). URL: https://doi.org/10.14257/ijdta.2015.8.2.07 (дата звернення: 05.10.2023).

22. Prototyping for all. URL: https://proto.io/ (дата звернення: 05.10.2023).

23. Icons for android studio. URL: https://icons8.com/icons/set/android-studio (дата звернення: 05.10.2023).

24. R. Sharma. Android Application Development. *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, no. VI, 2021.

25. A. Nugraha and F. Aminur Rahman. Android Application Development of Student Learning Skills in Era Society 5.0. *Journal of Physics: Conference Series*, 2021, vol. 1779, no. 1.

26. Á. Gunnarsson and M. Johnson. Application Lifecycle Management. *Pro Microsoft Power BI Administration*, 2020.

27. Meng, C., & Baier, H. (2019). Bring2lite: A Structural Concept and Tool for Forensic Data Analysis and Recovery of Deleted SQLite Records. *Digital Investigation*, 29. URL: https://doi.org/10.1016/j.diin.2019.04.017 (дата звернення: 08.10.2023).

28. Димова, Г. (2021). Аналіз методів оцінки ефективності систем фізичного захисту. *Computer-integrated technologies: education, science, production*, 45. URL: https://doi.org/10.36910/6775-2524-0560-2021-45-02 (дата звернення: 05.10.2023).

29. Легенчук, С. Ф., Царук, І. М., Назаренко, Т. П. (2021). Принципи захисту даних у системі обліку: управлінські аспекти. *Економіка, Управління Та Адміністрування*, 2(96). URL: https://doi.org/10.26642/ema-2021-2(96)-61-69

30. Равлюк, В. Ваврічен, О. (2021). Визначення рівня захищеності інформаційно-комунікаційної системи корпоративних мереж. *Збірник Наукових Праць Національної Академії Державної Прикордонної Служби України. Серія: Військові Та Технічні Науки,* 83(2). URL: https://doi.org/10.32453/3.v83i2.576 (дата звернення: 05.10.2023).

31. Khawas, C., & Shah, P. (2018). Application of Firebase in Android App Development-A Study. *International Journal of Computer Applications*, 179(46). URL: https://doi.org/10.5120/ijca2018917200 (дата звернення: 03.10.2023).

32. Degtyareva, L., Miroshnykova, M., & Voloshko, S. (2019). Аналіз структури системи захисту інформації. *Системи Управління, Навігації Та Зв'язку. Збірник наукових праць*, 2(54). URL: https://doi.org/10.26906/sunz.2019.2.078 (дата звернення: 03.10.2023).